A performance model of a telecommunications network structured according to Intelligent Network principles

Adrian Newcombe B. Eng.

A thesis submitted as a requirement for the degree of Master of Engineering in Electronic Engineering.

Dublin City University

Supervisor : Dr. T. Curran.

School of Electronic Engineering.

September 1997

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Engineering in Electronic Engineering is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Signed Adrian Newcombe ID No.: 92700390

Adrian Newcombe

Date : 5/9/97

Abstract

Title : A performance model of a telecommunications network structured according to Intelligent Network principles

Author : Adrian Newcombe

The *Intelligent Network* (IN) is an telecommunications services architecture which enables the rapid creation and deployment of supplementary telecommunications services. However, this flexibility makes the management of the network performance critical in ensuring that customers receive their expected Quality of Service. This thesis describes a model which has been developed to predict the delays in the network due to IN-specific service processing.

The model proposed is a queuing network which builds on the existing state of the art as follows. The characteristics of the flows between the IN physical entities are considered and general service time distributions are assumed at each entity. Additionally, the model allows the reservation of capacities as the SCP for each service type. An analytic formulation was developed using the decomposition approximate method. The model was also simulated in order to ascertain its' accuracy.

The results of the analytic solution and the simulation were compared for different scenarios and the results are presented in this thesis. The analytic approximation was found to be a very good solution for situations where network utilisation was low or medium. As the utilisation of the network increases to higher utilisation factors, the accuracy of the analytic solution decreases.

ACKNOWLEDEMENTS

I would like to take this opportunity to thank the following people for their help and assistance during the course of this work.

To my supervisor Dr. Tommy Curran, for the opportunity to undertake the work, and the help and guidance throughout. I would also like to thank Dr. Dmitri Botvich for his help, advice and comments.

To Fiona Lodge and Jimmy McGibney, for the comments and proof reading.

To my colleagues at Broadcom and in particular, Michael Slevin, Mark Tierney and Declan O'Sullivan.

To my friends and colleagues at DCU, for their help.

To my parents and family, for their support throughout my education.

To my fiancée Michelle for her help, support, encouragement and understanding, without which, I could not have completed this work.

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1

1. Introduction

The Intelligent Network (IN) is a telecommunications network service control architecture defined by ITU-TS. It provides a framework for the fast and efficient provisioning of new services through the reuse of service capabilities and network resources. The IN aims to facilitate service/network implementation-independent provisioning of telecommunications services in a multi-vendor environment. This is achieved by removing the dependence of operators on service developments by equipment vendors and by providing reusable, service independent functions, which may be used to ease the specification and design of new services. The processing of an IN service occurs at distinct and specialised functional components in the network - in other words, the service logic is distributed over each of the functional components. These components may be collocated in the same physical equipment, or may be distributed across physically separate Network Elements.

This situation is quite different to the traditional implementation of service logic in a telephony network. In such cases, while the call itself makes use of many different network resources (e.g. switching), the service logic resides at the end points of the network. This means that the processing of value added services such as Freephone, or Call Forward is performed at the local exchange, requiring the appropriate service logic at each of the local exchanges in the network.

Due to the different nature of the service processing in an IN structured network, it is important to consider what impact this new paradigm will have upon existing performance models of networks. The need to model the performance of the network has been apparent since the development of the first telephone networks. As networks become more complex and their usage grows further, the need to model the performance also increases. There are many situations in which it is important to model the performance of a telecoms network, including :

- During network planning, it is important that the network planner is able to predict the performance of the planned network. By modelling the performance of a planned network, the operator can determine a-priori what the expected performance of such a network should be.
- During network and resource dimensioning. In Telecoms networks, resource dimensioning is generally done by estimating the amount of resources required to meet the real or projected service usage and their associated Quality of Service (QoS) requirements.
- During performance optimisation of the network. This allows the analysis of planned changes to the network configuration, without committing the expenditure of implementing the new configuration.
- During service provisioning. When the operator is introducing a new service to the network or providing an existing service, to a new region, or large corporate customer, a large load may be added to the network. It is essential that the operator determines beforehand, what the effect on the network, of such a significant service load will be.
- During the drafting of service level agreements. It is increasingly common for large customers to enter into service level agreements about the costing and the QoS which the customer can expect. It is important for the service provider to be able to determine the nature of the performance related QoS parameters, before the provider can commit to such an agreement.
- During the development of '*crash plans*' for the network. A crash plan is a plan of action to be taken when a certain overload or failure scenario occurs in the network (e.g. a television phone-in causes an unprecedented load on certain network resources, congesting them or causing a network resource to fail). A performance model can be used to evaluate the performance of the crash plan.

Given these motivations for performance modelling, it is important to address some specific motivations which occur in an IN structured network. These arise from the manner in which services are processed in the IN. The processing of an IN service is

distributed across the IN functional components. The advantage of this is that service data and logic are in centralised locations and provisioning is more efficient. The operator does not have to deploy the service in every switch or network equipment, but instead deploys in a set of centralised nodes.

However, due to the distributed and shared nature of IN service execution, it is necessary to consider the impact that a particular service will have upon the performance of other services and on the performance of the network resources. A problem occurring at one of these resources can affect all of the services at that service resource [1]. Additionally, because the overall processing of a service is performed at several nodes, a failure at one node may have a knock-on effect on other nodes and affect a larger number of subscribers [1]. Thus, the processing of the service is analogous to a chain, which is only as strong as its weakest link - if a link were to break, the chain is broken, while if an IN resource were to fail or become congested, the knock-on effect would be felt by the other IN resources with which it is associated.

The operator will wish to avoid these problems as :

- A violation of the QoS parameters associated with a service is a breach of the contract for that service.
- In a competitive environment, customer-operator contracts may contain penalty clauses for each time QoS is violated. In such a case, the operator may have to pay the customer 'compensation' for each breach of contract.
- The customer may become tired of these breaches and may move to an operator whom they perceive as being better able to meet their needs.
- Where such congestion occurs, calls may be lost in the network due to unavailability of resources or the calling party getting impatient with the delay.

Each of these represent a loss of revenue to the operator, a situation that the operator must rectify to become efficient and remain competitive. However, if an operator reduces these problems by ensuring that there are always more than enough resources in the network to meet the demand, then the operator introduces another problem. If resources are under utilised, then there are resources in the network for which the operator is not getting any

return. As these resources cost money, then it is effectively money invested for which the operator is not gaining any benefit. Obviously the operator will also want to minimise the occurrence of this situation, so that all resources in the network are utilised as much as possible. So the challenge for the operator is to minimise the QoS problems experienced by service users while attempting to utilise scarce network resources to their full extent. This means that a performance model of the network is required, which allows an operator to predict the performance of a given network configuration under a particular service load.

This thesis describes a performance model of an IN structured network which has been defined. The model is a queuing network which enables the prediction of the mean delays experienced by services in the network. An analytic formulation has been developed for the queuing network which allows the formulation to be readily implemented on a computer. A simulation was also developed to validate the results of the formulation.

In the remainder of this report, a performance model of an IN structured network is proposed. The aim of the work is to define a tractable analytic model of the performance of the resources and services in the network. The model is then compared with a simulation of the network, in order to determine the efficacy of the solution.

The report is organised in the following manner.

- Chapter Two presents the IN concepts and a state of the art survey of IN performance models.
- Chapter Three provides a short overview of queuing theory.
- Chapter Four describes the proposed queuing model of the network, the associated analytic solution, and the simulation used to compare the results.
- Chapter Five details the results of the work.
- Chapter Six documents the conclusions of the work and the recommendations for further study.

Chapter 2

2. The Intelligent Network

Traditionally, services offered by Telecom operators were implemented by the switching equipment in the network. This caused several problems for a Telecom operator. The reliance of the operator on services to be developed by the equipment vendor restricted the operator in the services in which it could offer, particularly given the fact that few operators' networks contain equipment from a single vendor. As the equipment varied from vendor to vendor, so also did the services implementations and the services supplied by the different equipment in the network. This led to a situation where, the range of services available or service behaviour varied from area to area depending on the equipment on which they were implemented.

This situation also made the task of service creation and provisioning in the network more difficult. The provisioning of a new service required the updating of service logic in every switch. For example, to provision a network-wide Freephone number required the updating of routing tables in every switch in the network. In a large network, this is non-trivial and also introduces a greater potential for mistakes to be made.

In the competitive Telecoms market which is evolving at the moment, an operator must be able to provision services to customers quickly, and develop new services rapidly, in order to maintain a competitive edge. The IN is an architectural concept defined by ITU-TS, which addresses these issues. It provides a framework for the fast and efficient provisioning of new services through the reuse of service capabilities and network resources. The IN aims to facilitate service/network implementation independent provisioning of services in a multi-vendor environment. This is achieved by removing the dependence of the operator upon service developments by equipment vendors and to provide reusable service independent functions, which may be used to ease the specification and design of new services.

The core aims of the IN standardisation process can be summarised as follows :

- To enable the efficient use of network resources through the use of specialised service resources, which are then reused across the network and for different services;
- To promote integrated and rapid service creation and implementation through use of modular reusable network functions and
- To define standardised communication between network functions, via service independent interfaces.

The IN architecture is defined by the ITU [2] in terms of the IN Conceptual Model (INCM), which defines a multi-layered architecture for representing different aspects of an IN structured network.

2.1 The IN Conceptual Model.

The IN Conceptual Model (INCM) provides a framework that relates the different models and concepts within the IN to each other. The INCM consists of four distinct planes each describing a different view of the capabilities of the network. These views are :

1. service aspects;
2. global functionality;
3. distributed functionality and
4. physical aspects.

These views are represented in the INCM by the following planes:

1. *The Service Plane,* which provides an implementation independent view of the services in the network;
2. *The Global Functional Plane,* which provides a view of the network functionality as a single entity;
3. *The Distributed Functional Plane,* which models a distributed view of the functionality within the IN;
4. *The Physical Plane,* which models the physical aspects of the network.

2.1.1 The Service Plane.

In the *Service Plane* (SP), IN supported services are described to the end user as service features and services. ITU recommendation Q.1202 [3] defines a service feature as *'a specific aspect of a service that can also be used in conjunction with other services and service features'*. A service feature may be a core or optional component of a service that is an atomic component of a service as seen by the service user. A service feature may be used as a building block in the design of new services. The recommendation defines a service as being *'a stand alone commercial offering, characterised by one or more service features'*.

ITU recommendation Q.1211 [4] defines a set of 25 target service and 38 service features for Capability Set 1 (CS1). As one of the stated aims of the IN is to promote service creation, the operator is not precluded from creating new services from the set of service features or with additional service features.

2.1.2 The Global Functional Plane.

In the *Global Functional Plane* (GFP) [5], the functionality of an IN structured network is modelled as a single entity. The GFP models the functions of the IN in a service and service feature independent manner. These network functions are represented as *Service Independent Building-blocks* (SIBs), which are standard, reusable, network-wide capabilities, used to create services and service features. Included among the set of SIBs, is the *Basic Call Process* (BCP) SIB. This is a special SIB which identifies the call process from which IN services are invoked. Services and service features are created by chaining SIBs together with the BCP. This description of the service/service feature is known as the *Global Service Logic* (GSL).

2.1.2.1 Service Independent Building Blocks

A SIB, as defined above, is a standard, reusable, network wide capability, used to create services and service features. A SIB is independent of the services and service features, within which it is used and also of the technologies on which it is implemented. Service features can be created by combining one or more SIBs together. However as SIBs are

service independent, the service dependence is introduced during service creation by the data upon which the SIBs work.

There are two types of data which are input to a SIB :

1. *Call Instance Data* (CID), are data specific to the call instance in question (e.g. the dialled number). Such data might come from the BCP SIB, the user, or from another SIB in the SIB chain.
2. *Service Support Data* (SSD), are data specific to the service in question (e.g. an announcement to be played to the user). The values of SSD are set at service creation time.

2.1.2.1.1 SIBs in Capability Set 1

Capability Set 1 (CS1) defines thirteen SIBs which can be supplied by a CS1 IN structured network. The service provider is not precluded from adding new SIBs, where necessary to provide new service capabilities, but these SIBs should meet the guidelines laid down for a SIB[1]. The CS1 SIBs are :

- *Algorithm,* which applies a mathematical algorithm to data to produce a data result;
- *Charge,* which defines a special charging treatment for a call (e.g. reverse charging);
- *Compare,* which performs a comparison of an identifier against a specified value;
- *Distribution,* which distributes calls to different points based on user specified parameters;
- *Limit,* which limits the number of calls related to IN provided service features subject to user parameters[2];
- *Log,* which logs call information for each call into a file which may be used for management purposes;

---

[1]The SIB should be network and service /service feature independent, perform a standard reusable network function, and provide a standard, stable interface to other SIBs.

[2]This is not intended as a congestion control mechanism.

- *Queue*, which allows incoming calls to be queued to a called party who is busy;
- *Screen*, which performs a comparison of an identifier against a list to determine whether the identifier can be found in that list;
- *Service Data Management*, which allows end user data to be replaced or retrieved;
- *Status*, which enables the service instance to check the status and/or status changes of network resources;
- *Translate*, which determines output information from input information (e.g. number translation);
- *User Interaction*, which allows information to be exchanged between the network and a call party (e.g. play announcements, collect digits, etc.);
- *Verify*, which confirms that the information received is syntactically consistent with the expected form of such information.

2.1.2.2 Basic Call Process SIB

The BCP special SIB provides basic call connectivity between parties in the network. The BCP is the starting and ending point for the SIB chains, which are used to represent IN services/ service features. These points in the BCP are known as :

A. *Points of Initiation* (POIs), which are the BCP functional launching points for the SIB chains. CS1 defines nine POIs, which are :
   1. *call originated* - identifies that the user has made a service request, without yet specifying a destination address (i.e. user OFF HOOK).
   2. *address collected* - address input has been received from the user.
   3. *address analyzed* - address input analysed to determine its characteristics.
   4. *prepared to complete call* - network is prepared to attempt completion of the call to the called party.
   5. *busy* - call destined for busy user.
   6. *no answer* - call offered to user who hasn't answered.

7. *call acceptance* - call is active, but connection has not yet been established.
8. *active state* - call is active and connection has been established.
9. *end of call* - a call party has disconnected.

B. *Points Of Return* (PORs), which are the BCP functional points where the SIB chains terminate. CS1 defines six PORs, which are :

1. *continue with existing data* - indicates that the BCP should continue call processing without modification.
2. *proceed with new data* - indicates that the BCP should proceed with call processing with a modification of call data.
3. *handle as transit* - indicates that the BCP should treat the call as if it had just arrived.
4. *clear call* - identifies that the BCP should clear the call.
5. *enable call party handling* - identifies that the BCP should perform functions to enable call control for individual call parties.
6. *initiate call* - identifies that a call should be initiated. This may be independent of an existing call or may be in the context of an existing call.

It is the POIs and PORs in the BCP, together with the SIB chains which start and terminate upon them, that determine the service /service feature to be created (see Figure 2-1). The same SIB chains launching from and/or returning to different POIs/PORs may describe different services / service features. The SIB chains, POIs, PORs and SIB data when viewed together are known as the Global Service Logic (GSL).

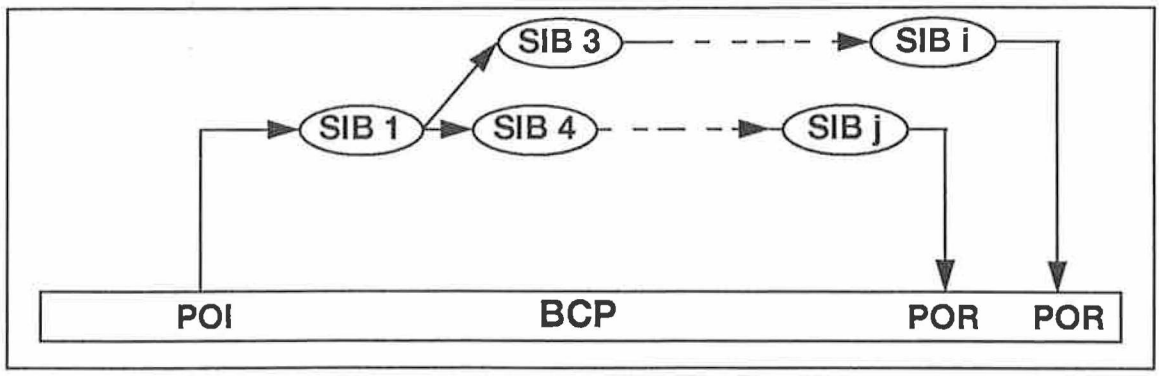

Figure 2-1 - Global Service Logic Example.

2.1.2.3 Global Service Logic

The GSL defines the order in which SIBs must be chained, in order to provide a service / service feature. The GSL comprises (see Figure 2-1):

- BCP POIs;
- BCP PORs;
- SIBs;
- Connections between SIBs and between SIBs and POIs/PORs;
- Input/output data to each SIB;
- SSD, CID for each SIB.

2.1.3 Distributed Functional Plane.

The *Distributed Functional Plane* (DFP) provides a distributed view of the IN structured network. The DFP contains *Functional Entities* (FEs), which are a unique subset of the total set of functions required to provide a service. One or more functional entities may be located in a *Physical Entity* (PE) and while different PEs may contain different functions, certain functions may be contained in more than one PE. Interactions between FEs are called information flows and the relationship between any two FEs is defined by the set of information flows between them. FEs may be classified in two groups - those related to IN service execution, and those related to IN service creation/management. The full set of FEs and their relationships are shown in Figure 2-2.

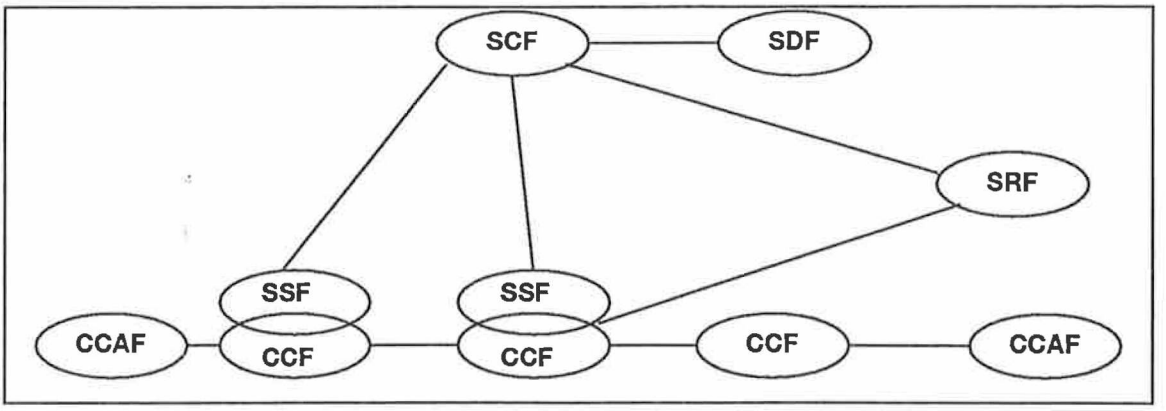

Figure 2-2 - Service Execution Entities in the Distributed Functional Plane.

The IN service execution FEs are :

- The *Call Control Agent Function* (CCAF), provides access for users. It is the interface between users and network call control functions.
- The *Call Control Function* (CCF), which provides call/connection processing and control.
- The *Service Switching Function* (SSF), which is associated with the CCF. It provides the set of functions required for the interaction between the CCF and a Service Control Function.
- The *Service Control Function* (SCF), which commands call control functions in the processing of IN service requests. The SCF may interact with other FEs to access additional logic, or to obtain information required to process a call/service logic instance.
- The *Service Data Function* (SDF), contains customer and network data for real time access by the SCF in the execution of an IN provided service.
- The *Service Resource Function* (SRF), provides specialised resources required for the execution of IN provided services and is used, for example, in facilitating interactions with the service user.

The IN service creation/management FEs are :

- The *Service Creation Environment Function* (SCEF), allows service to be defined, developed, tested and input to the Service Management Function.

- The *Service Management Agent Function* (SMAF), provides an interface between service managers and the Service Management Function allowing service managers to manage their services.
- The *Service Management Function* (SMF), allows deployment, provisioning and ongoing support of IN services.

2.1.3.1 The Basic Call State Model

The *Basic Call State Model* (BCSM) is defined by ITU [6], as *'a finite state machine description of CCF activities required to establish and maintain communication paths for users'*. It identifies the points in basic call and connection processing at which IN service logic instances are permitted to interact with call and connection control capabilities. It also provides a framework for describing basic call and connection events that can lead to the invocation of IN service logic instances and the points in call and connection processing when the transfer of control can occur. A BCSM is defined in terms of (see Figure 2-3) :

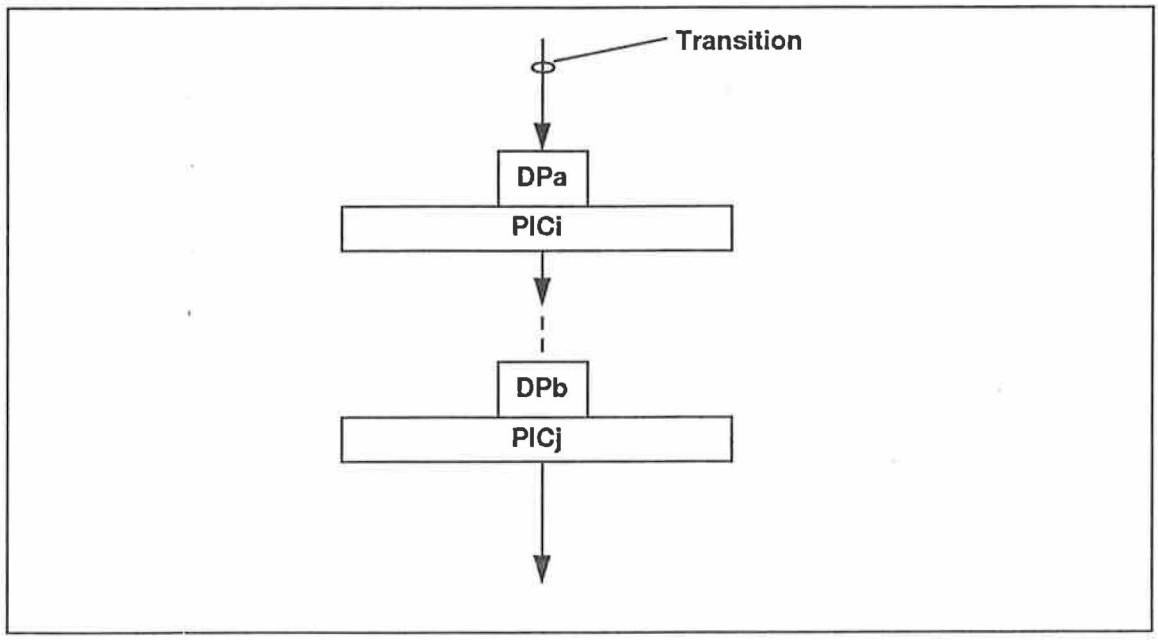

Figure 2-3 - Basic Call State Model Components.

- *Points in Call* (PICs), which identify CCF activities required to complete one or more call/connection states of interest to IN service logic instances.

- *Detection Points* (DPs), which indicate points in basic call/connection processing at which transfer of control to the SCF can occur.
- *Transitions,* which indicate the normal flow of basic call/connection processing from one PIC to another.
- *Events,* which cause the transition into or out of a PIC.

In CS1 the BCSM models switch processing of a two party call and reflects the separation between the originating and terminating portions of call. To this end, the BCSM is divided into two halves - the *Originating BCSM* (OBCSM) to model the originating portion of call/connection processing and the *Terminating BCSM* (TBCSM) to model the terminating end of call/connection processing.

2.1.3.2 Originating Basic Call State Model

The OBCSM [6] defines six PICs - these being:

1. *O_NULL&Authorize_Origination_Attempt.* At this PIC the originating party has attempted to make a call and the authority of the calling party to make the call is verified.

   Entry Events:

   - The disconnection or clearing of a previous call.

   Exit Events:

   - Indication of a desire to place an outgoing call;
   - Authority to place call has been denied.

2. *Collect_information.* At this PIC information is collected from the calling party (e.g. Dialling string).

   Entry Events:

   - Indication of a desire to place an outgoing call.

   Exit Events:

   - Dialling information has been collected;
   - Calling party abandons call;
   - An error occurs in collection.

3. *Analyse_information*. At this PIC the information collected is examined to determine routing address and call type.

   Entry Event:

   - Dialling information has been collected.

   Exit Events:

   - Availability of routing and call address;
   - Calling party abandons call;
   - An error occurs in call analysis.

4. *Routing_and_alerting*. At this PIC the next route is being selected, the authority of the calling party to place this type of call is verified and the call is also being processed by the TBCSM.

   Entry Event:

   - Availability of routing and call address.

   Exit Events:

   - Indication from the TBCSM that call is accepted;
   - Unable to select a route or call unable to be presented to called party (e.g. network congestion);
   - Indication that called party is busy;
   - Indication from TBCSM that called party has not answered;
   - Calling party abandons call;
   - Calling party does not have authority to place call.

5. *O_Active*. At this PIC connection is established between called and calling parties.

   Entry Event:

   - An indication from TBCSM that call is accepted.

   Exit Events:

   - A service /service feature request received from calling party;
   - A party disconnects;
   - An error occurs.

6. *O_Exception*. At this PIC an exception (error) has occurred.

   Entry Event:

- An error occurs.

Exit Events:

- Default handling of the error condition is performed.

The OBCSM is shown in Figure 2-4.

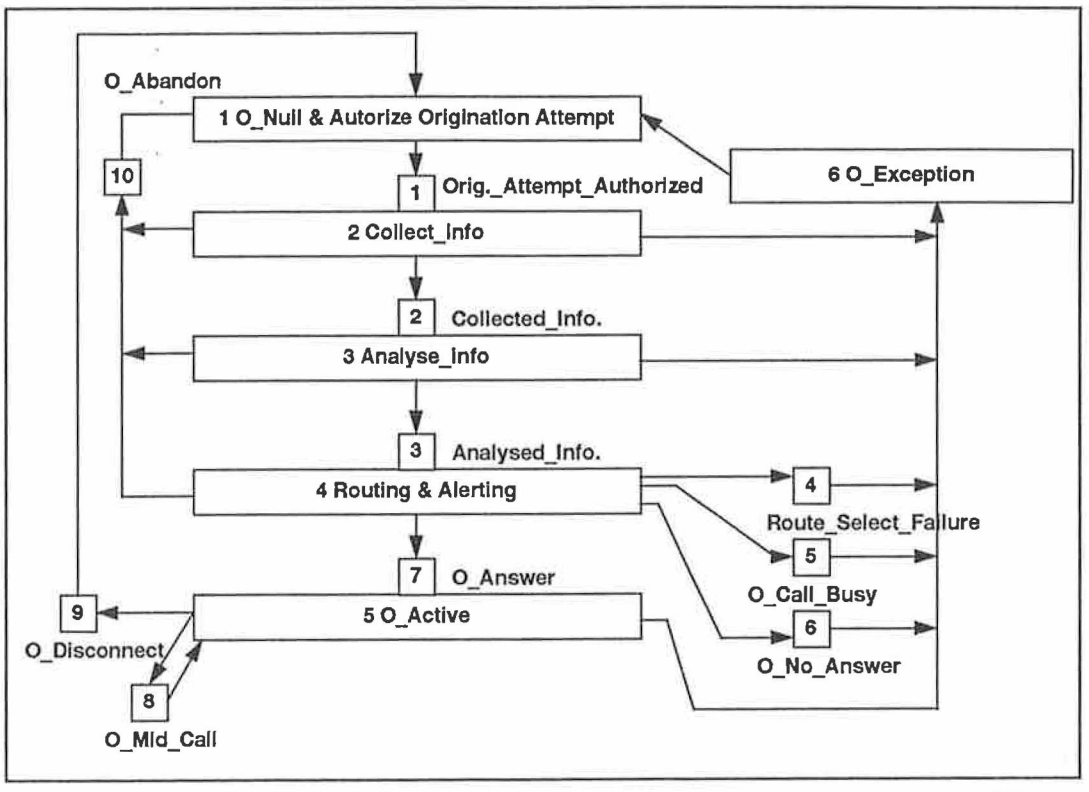

Figure 2-4 - Originating Basic Call State Model for CS1.

2.1.3.3 The Terminating BCSM.

The PICs for the TBCSM defined in [6] are (see Figure 2-5):

7. *T_Null&Authorize_Termination_Attempt.* At this PIC the interface is idle and an indication of an incoming call is received from the OBCSM. The authority to route this call to the called party is verified.

   Entry Event:

   - Disconnection of a previous call.

   Exit Event:

   - Incoming call received and authority verified.

8. *Select_Facility&Present_call.* At this PIC a resource to use for the call is chosen and the resource is informed of the incoming call.

   Entry Event:

   - Incoming call received & authority verified.

   Exit Events:

   - Called party is alerted;
   - All resources are busy;
   - Call is accepted;
   - Calling party abandons call;
   - An error occurs.

9. *T_Alerting.* At this PIC the called party is informed of the incoming call.

   Entry Event:

   - Calling party being alerted.

   Exit events:

   - No answer;
   - Call accepted;
   - Calling party abandons call.

10. *T_Active.* At this PIC the connection between calling and called parties is established.

    Entry Event:

    - The call is accepted.

    Exit Events :

    - The called party invokes a service/service feature;
    - The call party abandons call;
    - An error occurs.

11. *T_Exception.* At this PIC an error has occurred.

    Entry Event:

    - An error is encountered.

    Exit Event:

- The default error handling of error condition is performed.

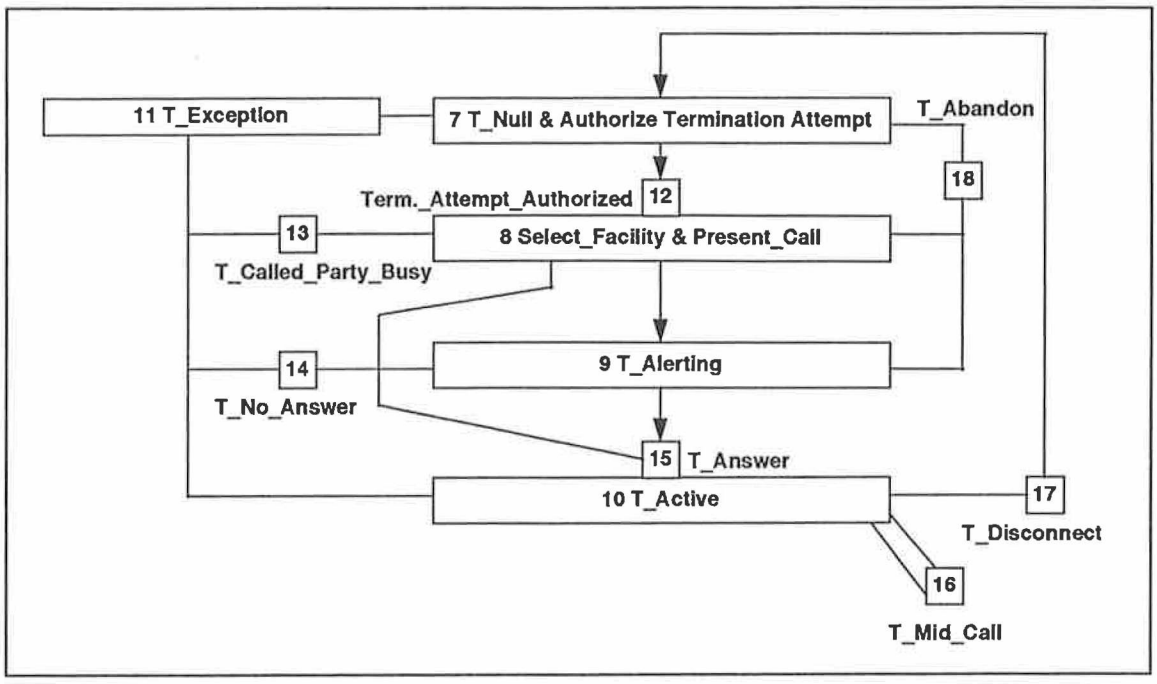

Figure 2-5 - Terminating Basic Call State Model for CS1.

2.1.3.4 BCSM Detection Points

A *Detection Point* (DP) is a point in the call processing at which an event is detected and transfer of control may occur. DPs may be armed to indicate to an IN Service Logic Program Instance (SLPI) that the DP has been encountered. Then the service logic is used to influence call processing. If a DP is not armed, the SCF does not get involved in call processing. DP may be characterised by four attributes :

1. The *Arming Mechanism,* which describes how the DP is armed. This may be either static or dynamic. A DP may be *statically* armed via SMF service feature provisioning and the DP remains armed until disarmed by the SMF. A DP is *dynamically* armed by the SCF during the processing of an IN call and the DP remains armed until it is detected.
2. The *Detection Criteria,* which are the conditions that must be met at the armed DP before the SCF is notified.

3. The *relationship*. When the conditions for an armed DP are met, the SSF may provide an information flow via a relationship. This relation may be an IN service control relationship which is one of two types :
    - A control relationship, where the SCF can influence call processing;
    - A monitor relationship, where the SCF cannot influence call processing.
4. The *Call Processing Suspension*. When criteria for an armed DP are met and the DP encountered, the CCF may suspend call processing and allow the SCF to influence call processing. Alternatively, the CCF may continue call processing and notify the SCF that the DP was encountered. In this case, it does not expect a response from the SCF. Table 2-1 shows a breakdown of the DPs based on the four criteria above.

DP Type	Arming Mechanism	Criteria	IN Service Control Relationship	Suspension	Service Feature Examples
TDP-R	Static	Specific to DP	Initiates Control Relationship	Yes	All
TDP-N	Static	Specific to DP	Initiates and terminates monitor Relationship	No	Televote
EDP-R	Dynamic	None	Within context of existing control relationship	Yes	Call Distribution
EDP-N	Dynamic	None	Within context of existing control or monitor relationship	No	Call Queuing

Table 2-1 - BCSM Detection Types.

2.1.4 The Physical Plane

The *Physical Plane* (PP) describes the physical IN structured network in terms of Physical Entities and the interfaces between them. A physical entity is piece of equipment as which one or more FEs are implemented. The ITU defined ten PEs for CS1 [7], but in this discussion we will limit ourselves to a subset of the ten, which provide all the

functions necessary to support service execution[3].

This subset is (see Figure 2-6) :

- The Service Switching Point (SSP), which provides access to the network, performs any required switching and allows access to the set of IN capabilities. Functionally a SSP contains a CCF, SSF and, in some cases,[4] a CCAF.
- The Service Control Point (SCP), which contains the service logic programs (SLPs) that are used to provide IN services. The SCP contains a SCF and in some cases[5] may also contain a SDF.
- The Service Data Point (SDP), contains data used by SCPs to provide services. It contains the SDF.
- The Intelligent Peripheral (IP), provides access to specialised service resources and allows for interaction with a user during the course of a call. It contains the SRF.

---

[3]All the other PEs are either a combination of FEs which are already contained in the subset identified or are related to management functions.

[4]In the case where the SSP is a local exchange.

[5]For example, in a case where performance may be an issue.

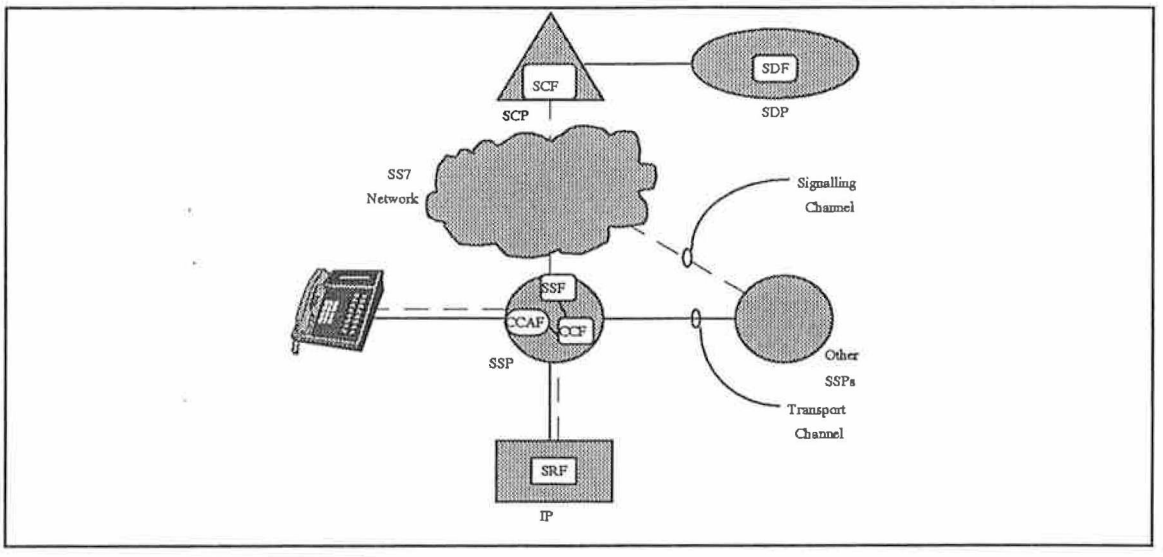

Figure 2-6 - Physical Entities in the Physical Plane.

2.1.5 A Service Example - Abbreviated Dialling.

Having given an overview of the INCM and its constituent parts, it is useful to consider how a service is actually described in each layer of the model and the different views of the service in each. As an illustrative example, we consider a very simple service - Abbreviated Dialling. The Abbreviated Dialling service is one which allows the user to assign a short code to a longer number which they dial frequently. For example, a user might assign the two digit code 12 to the number (012) 3456789. This service has two aspects :

1. The user assigns codes to numbers;
2. The actual use of the service, where the user makes the call by indicating the abbreviated service and entering the code. Let us assume that an abbreviated number is indicated by a two digit code, preceded by a '#' character.

For the purposes of illustration, the representation of the second aspect of the service, in the different planes of the INCM, is presented here.

2.1.5.1 Abbreviated Dialling in the Service Plane

The abbreviated dialling service appears in the Service Plane as it would to the service user. It consists of a single core service feature - abbreviated dialling. The service allows the user to assign a short code to a longer number.

2.1.5.2 Abbreviated Dialling in the Global Functional Plane

The abbreviated dialling service is represented in the GFP, by its GSL (see Figure 2-7). The SIB chain is launched from the BCP at the *'Address Analysed'* POI. This indicates that the dialled number has been analysed and identified as being of interest to the service logic. The SIB chain itself contains the *verify* SIB to check the syntax of the address, the *translate* SIB to translate the abbreviated number into the appropriate network address and *the user interaction* SIB is used if either of the previous two SIBs encounters an error. In this case, an announcement is played to the user to indicate the error and the call is cleared. The PORs are *'Proceed with New Data'* in the case where the translation has been successful, or *'Clear Call'* where an error has been encountered. This GSL description of the service describes the functions of the service but does not describe where in the network the functions are implemented. As such, the view presented is a network-wide functional view of the service.

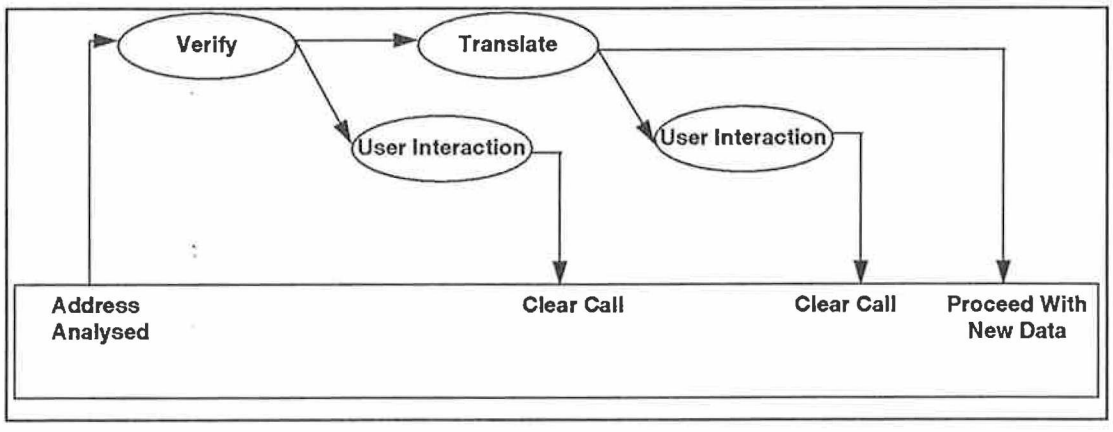

Figure 2-7 - Global Service Logic for Abbreviated Dialling Service.

2.1.5.3 Abbreviated Dialling in the Distributed Functional Plane

In the DFP, the functionality of the service is distributed across a number of functional entities. One of the most important aspects of the service view in the DFP, is the set of information flows which are exchanged between the different FEs and the sequence in which they occur. These are (see Figure 2-8) :

- The User goes off-hook and a dial tone is provided to the user;
- The User dials '#' followed by two digits;

- The digits typed in by the user are collected and examined by CCF;
- A service request is recognised by the CCF and call processing is suspended;
- The SSF builds a message containing the set of dialled digits;
- This message is sent to the SCF via the SS7 network;
- The SCF creates a new instance of the abbreviated dialling SLP;
- The SLPI sends the dialled digits as the information key to the SDF and the actual Destination Number (DN) associated with the abbreviated dial code is found;
- The SDF passes the result back to the SCF;
- The SCF tells the CCF (via the SSF) to route the call to this DN;
- The SLPI terminates and CCF once again takes over call processing;
- The call is now processed as a normal call.

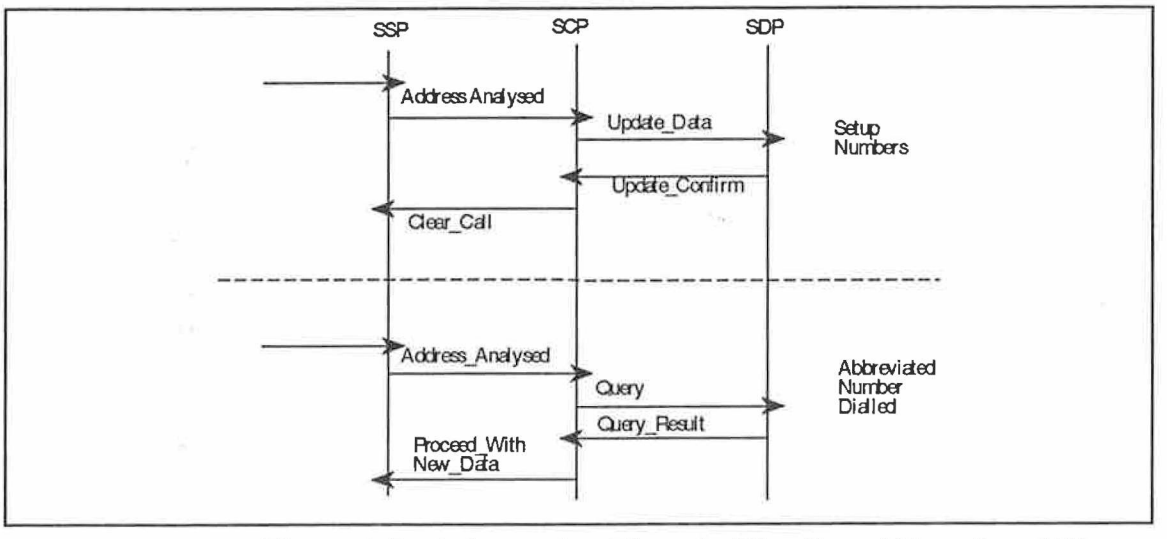

Figure 2-8 - Information Flows in Distributed Functional Plane for Abbreviated Dialling Service.

2.1.5.4 Abbreviated Dialling in the Physical Plane

The service view provided by the physical plane may or may not be, similar to that provided by the DFP. This is dependent on whether the DFP FEs reside in a single PE, or if they are combined in one PE. Assuming that the FEs above each reside in the following PEs as shown in Table 2-2, then we get the view as shown in Figure 2-9.

Functional Entity	Containing Physical Entity
Service Switching Function / Call Control Function	Service Switching Point
Service Control Function	Service Control Point
Service Data Function	Service Data Point

Table 2-2 - Mapping of Functional Entities to Physical Entities.

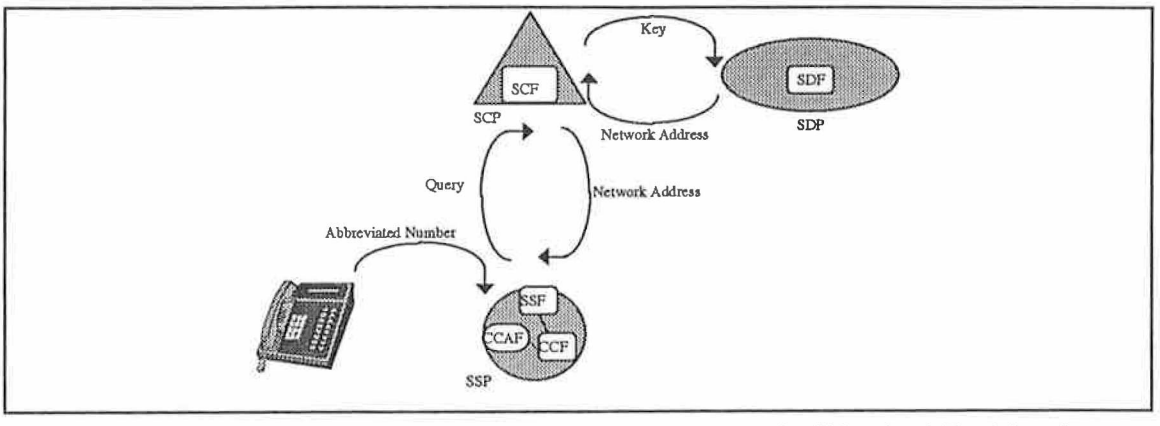

Figure 2-9 - Interactions between the Physical Entities for Abbreviated Dialling Service.

2.2 The Advanced Intelligent Network.

The Advanced Intelligent Network (AIN) is an IN architecture proposed by Bellcore [8], which is similar in concept to the ITU recommendation. While it does not have the conceptual planes of the INCM, the functional and physical architectures are quite similar albeit using different terminology. One of the major differences between the two recommendations is how services are created. IN (as defined by ITU), provides a set of reusable network functions in the form of SIBs which can then linked together and customised with data in order to create a service.

In AIN, the service designer is offered an API in the form of functional components, which are low level functions which can be used to create a service. So while the IN has a data centred approach to service creation, through the linking of SIBs together with the appropriate data, the AIN takes a functional approach in which the service is realised using a combination of primitive functions.

2.3 Teletraffic Implications of Intelligent Networks

The implications of IN upon the teletraffic characteristics of a network can be classified into two areas :

1. *User Traffic Characterisation*, in which the traffic characteristics of users are analysed and predicted;
2. *Performance Criteria* which should be measured and analysed in order to ensure that the network is operating efficiently and for network planning purposes.

These are now discussed in detail.

2.3.1.1.1 User Traffic Characterisation

User traffic characterisation and forecasting, which will be more difficult for two reasons [9] :

1. There will be limited historical data (in some cases, there may be none) from which to make a characterisation or prediction of a user load;
2. The number of traffic parameters is likely to increase rather than decrease.

The former assumption is quite easy to justify as, for example, the forecasting might be related to a new service to be deployed. Thus, there will be no historical data from which to base an assumption and so other methods of forecasting such as market research must be used instead. The latter assumption is more difficult to understand at a first glance. However, as the IN services become more complex, requiring more intricate interactions between the IN entities and with the service user(s), more parameters may be needed to represent the different types of interactions.

Jensen, [10] discusses other issues in forecasting traffic demands on an IN structured network. These include :

a) Often it is assumed that the service demands are fixed and not influenced by the resulting quality. This does not take the users behaviour into account and the behaviour of the user can have a significant effect on the incoming load [11],[12]. However, Jensen proposes that for network planning and dimensioning, it is still valid to assume that the service demands are fixed.

b) The characteristics of IN have a bearing on predicting the load a service will place upon the IN resources. Different services can be implemented in different ways and can have different complex interactions with several of the IN resources. Thus, the number and nature of the interactions of the service with the resources is an important factor in determining the loads placed upon the IN resources.

2.3.1.1.2 Performance Criteria.

A survey of the current literature shows that there are three main characteristics of an IN structured network which must be modelled. These are :

1. Post Dialling Delay;
2. Impaired Call Rate;
3. Lost Call Rate.

Post Dialling Delay

Post Dialling Delay (PDD), is the delay between the user dialling a number (or interacting with the network in some manner) and receiving a response. While PDD has been decreasing in the Public Switched Telephone Network (PSTN) due to developments such as common channel signalling, it is likely to increase in the IN due to the increased processing of the value added aspects of each call (e.g. time taken for database lookups). This means that an increase in processing time at any of the IN nodes may result in a net increase in PDD, which causes the QoS to be violated. While the source of the congestion may be a single service, it can have a serious effect on the whole network. This point is reinforced by Pierce et al [13], who point out that such congestion in the network will affect all services which share the congested resources (i.e. not just the newly added services, but also previously existing services). The problems may be further worsened by subscriber behaviour [12] - due to the increased delay, subscribers will tend to abandon calls and retry resulting in further call set-up attempts, placing a larger load in the already overloaded resources.

According to Yan and MacDonald [9], this additional call processing could push the users expectations of PDD beyond the users level of acceptance. This is further compounded,

according to the authors, by the evidence that users will tend to expect PDD targets lower than have previously been accepted.

MacDonald and Archambault [14] present the results of experiments which were carried out on users perceptions of PDD. The standard thresholds for PDD as defined by the ITU [15], define a PDD threshold of three seconds for a local call, five seconds for a trunk call and eight seconds for an international call. According to MacDonald and Archambault, these targets for PDD are easily met in a PSTN/ISDN with efficient routing and in such a network, values would average between 1.3 and 1.5 seconds. The authors define a target of 4 seconds for the setting up of a local call (the local call is the worst case as it has the lowest PDD threshold and the most stringent user expectations) and allot 1.5 seconds of this time to non-IN call set up and processing. This leaves a maximum of 2.5 seconds for the processing of the IN aspects of the call which, as the authors state, is a very strict target and one which may have a significant effect on the planning and operation of IN structured networks.

Impaired Call Rate

Traditionally in the PSTN a call was either connected to the user or it was blocked. In IN more complex scenarios can occur, as some IN calls could be completed without all of the desired service features being provided. An example of such a scenario, is where a call to a mobile roaming service is connected to a voice mail service because the terminal location information was not supplied in time. Such a call can have been said to have been completed as the caller was connected to the voice mail. Because all of the required service features were not available, the call is said to have been *impaired* [1]. The Impaired Call rate is thus, an important parameter of the IN.

Blocking Probability

The blocking probability is a common metric used throughout teletraffic modelling and describes the probability that a call request will be lost due to unavailability of resources for the request. However, a survey of the literature shows that the blocking probability is seen as being of lesser importance than the previous two parameters. The reason for this may be that the service logic implementations are delay systems rather than loss systems, and that in the case of a resource being blocked, the service logic could re-route the call to other backup resources (e.g. a voice mail). In such a scenario, as presented previously, the call is said to be impaired.

In summary, the important parameters of a model of the IN will be the PDD and the Impaired Call and Lost Call rates.

2.4 State of the Art in IN Performance Models

This section describes the results of a state of the art survey into existing approaches and solutions to performance modelling of IN structured networks. Compared to performance modelling research in other areas of telecommunications such as high speed networks (e.g. ATM) there has been very little work conducted in the area of performance modelling of the IN. Rumsewicz [16] proposes that this is due to the perception that the relatively low throughput rates means that the interesting issues have all been solved or that the lack of bandwidth is seen as not being "sexy" enough.

Most of the work which has been done in this area is aimed at modelling overload control strategies to prevent IN resources from becoming overloaded and to help them recover from such situations. When congestion occurs at a resource, then the network should be able to recover from the problem gracefully and with the minimum of disruption to customers. Congestion control strategies such as call gapping and throttling aim to do this. In order to develop and evaluate such strategies, it is important to have a model of the network. The models examine the performance of the network under overload conditions and while network resources are subject to traffic controls. This means that the model may not be as valid when the network is in a steady state.

A further point to note in relation to models for overload is that they tend to concentrate on modelling the SCP rather then the network as a whole and for simplicity often focus on the behaviour of one service [17].

The remainder of the work in modelling IN performance addresses the steady state operation of an IN structured network mostly for planning or dimensioning purposes. In some cases, the models are purely simulation, in others, analytic models have been used, while in other cases, a combined approach has been taken. The approach taken to solve the model will be presented here as each model is discussed. Also, some models focus on the behaviour of the network with one service deployed, while others deal with multi service networks. Again the number of services modelled will be discussed for each model presented.

Leever et al [18] propose an analytic model for the prediction of resource utilisation, resource response times and service response times. The model makes the following assumptions :

- Each PE contains one or more CPUs.
- The SSP is modelled as a single server queue with limited buffer size, a general arrival process and Poisson service distribution (i.e. a G/M/1/m queue). The service discipline is First Come First Served (FCFS).
- The IP is modelled as a single server queue with general arrival process, Poisson service time distribution and a FCFS service discipline (i.e. a G/M/1 queue).
- The SCP is modelled as a number of parallel CPUs in series with the SDP. The CPUs are modelled similarly to those in the SSP.
- The SDP is modelled as a single server queue with general arrival rate, deterministic service time and a FCFS service discipline (i.e. a G/D/1 queue).

Three services are modelled - Number Translation, Credit Card calling and Virtual Private Network (VPN). The analytic formulation of the model allows the estimation of the following three model parameters :

1. The utilisation of the PE. The assumption is made that that jobs are uniformly distributed over the PEs.
2. The PE response times, which are calculated using the formula for the mean response time of the M/M/m queue. This assumes that the arrival process has a Poisson distribution, rather than a general distribution.

3. The service response time, which is found by summing the individual response times at each of the physical entities.

This analytic model was used to determine the response times of the three services which were considered and a simulation was used to validate the results. Unfortunately, the authors do not publish the network configuration for which these results were gathered or any details of the simulation which was used. However, the work does provide an analytic formulation of a queuing model for the IN, allowing the estimation of the service response time, the response time at each PE and the utilisation of each PE.

An issue which is not addressed in this model, is the nature and the interdependencies of the message streams between the physical entities in the network. In this queuing model, requests arriving at a particular station are often leaving another station in the network. Such flows cannot always be assumed to be Poisson as the time spent in the queue and in processing at the originating entity may affect the distribution of the departures from that station (see chapter 3). While the queues are denoted as having general arrival time distributions, the formula used for estimation of the response times is that of the M/M/1 queue. This means that Poisson arrivals are being considered.

Kwiatkowski and Northcote [19] use an analytic queuing model to calculate mean delays of services in an IN. The focus of this work is on the delays experienced by signalling messages in the SS7 network, and when the SCP is in overload condition. The model is used to evaluate the mean waiting time experienced by SS7 *Message Signalling Units* (MSUs) and the end-to-end delays experienced by each type of service. The following assumptions are made :

- Each processor, within each of the SS7 network elements, is modelled as a single server with non-pre-emptive priority service, general service time and Poisson arrivals (i.e. a M/G/1 queue with priority service discipline). The mean wait times at each of these processors is estimated.
- The behaviour of the transmission links can be approximated by an M/G/1 system. The mean waiting time for transmission of MSUs on SS7 links can be estimated using the Pollaczek-Khinchin formula (see chapter 3).

- The jobs[6] within a call are performed sequentially so that the overall mean delay is the sum of the response times for each of the jobs within the call.

The model was used to analyse the overload behaviour of a network of eight Signalling Points and one SCP with two services - a POTS service (which doesn't use IN resources) and a Freephone service. The analytic model was validated by a simulation of the network and although not published the authors claim the difference between the simulated results and estimated results was no greater than three per cent.

Kühn, Schopp and Bafutto [20], [21] propose a two tier model for the performance analysis of IN structured networks. In one tier, the detailed aspects of the SS7 network are modelled and analysed, while in the second tier, the IN resources are modelled and the details of the SS7 network are abstracted away. Each of the IN PEs are modelled as a single server queue with Poisson arrivals and general service time distribution (i.e. M/G/1 queue). For the purposes of the IN analysis, the SS7 network is modelled as an infinite server queue. The model is then developed as an open queuing network where requests take one of multiple routes depending on the service types. The decomposition approximate method (see chapter 3) was used in order to solve this model analytically. The authors do not provide detail on the analysis which was performed on the IN aspects of the network and concentrate on the signalling aspects. However, the model was used to analyse the load on the IN resources and to determine the response time of the individual resources. The service response time can then be estimated. The model considers two services Freephone and Credit Card Calling. The authors do not present a comparison of the analytic results with simulated results and it is unclear how accurate the model is.

Lodge et al [22],[23] propose a detailed network model of an IN for both simulation and analytic formulation. An important focus of this work is congestion control schemes used at the SSP. For this reason, the SSP is modelled in more detail than the rest of the network. The network queuing model consists of the following :

- The SSP is modelled by four queues. The first queue receives all call requests and its' service time models call authorisation within the SSP. The second

---

[6] A job is defined as a particular transaction with a PE.

queue's service time, represents the time taken for number analysis. The third queue service time, represents the wait time for routing of calls. Non-IN calls enter this queue after completing number analysis, while IN calls enter after service specific processing at SCP, SDP, IP is complete. The fourth queue represents the wait time for the transmission of messages to the IP. All of the SSP queues are served by the same processor according to a priority based processor sharing service discipline. The second queue has three service time distributions one each representing call rejection time, non IN acceptance time, and IN acceptance time. All of the other SSP queues have a single exponential service time distribution.

- The SCP is represented as a single server queue with exponential service time distribution.
- The SDP is represented as a single server queue with deterministic service time distribution.
- The IP is represented as an Erlang-C delay server with uniformly distributed service time.
- The distribution of inter arrival times at each of the queues is assumed to be general.

The parameters of interest in the model are the mean response time at each of the queues and the mean response time experienced by a service type. The model does not consider the signalling network.

This model was simulated using the OPNET simulation package and was also developed analytically using the decomposition approximate method (see chapter 3). The author found that as the utilisation of the resources in the model increased, the margin of error between the simulation and approximation also increased [24]. An earlier incarnation of the model described later in this thesis, and of the model used by Lodge is described in [25].

Gulyani [12] developed a simulation, in which the performance of an Advanced Intelligent Network (AIN) SCP is analysed under overload conditions. The objective was to examine the performance of the system where congestion control strategies are used. In

particular, the aim of the work was to contrast the performance of the SCP during overload, when traffic controls are present and when such controls are not present. In order to analyse this behaviour, a simulation model was developed of an AIN network, for a set of four services :

1. Call Forwarding;
2. Call Transfer;
3. Call Back;
4. Automatic Call Distribution.

The simulation was developed as a discrete event simulation, using the OPNET simulation package and modelled the behaviour of the AIN SSP, SCP, IP and Service Management System (SMS)[7,8]. The simulation did not implement any underlying queuing model of the AIN, but instead modelled, in a simplified manner, the actual processing of the service in the network. In particular, the simulation modelled the call and service processing of a service in the network by considering the passage of the (signalling) messages throughout the network. The processing times of the services at each resource were not modelled in a probabilistic manner, but the distributions of the arrival of calls into the network was considered.

Kihl and Rumsewicz [17] propose and analyse a congestion control strategy for IP voice circuits in an IN structured network containing multiple SCPs and multiple services. The authors contend that there are two issues in relation to overload at an IP - overload of the IP processor and congestion on the voice circuits between the SSP and IP. The work conducted here concentrates on these issues and a fluid flow model of a multi SCP network is proposed which calculates the overflow on the SSP-IP circuits and the congestion levels at the SCP. The model allows the analysis of the behaviour of the network under the control strategies. This model was also simulated in order to validate the model.

---

[7] The SMS is important in this scenario as it sets the thresholds for the traffic controls.

[8] The SDP does not feature in this simulation. It is instead, assumed to be combined with the SCP.

Arvidsson, Petersson and Angelin [26], [11] take a novel approach of defining the network profit and customer satisfaction as performance metrics, in which successful calls, rejected calls and blocked calls all have a particular cost function associated with them. The authors use a prediction method of estimating future processing and transmission delays based on the previous historical delays. These predictions are then combined with the profit/satisfaction metrics, using Bayesian decision theory, in order to determine the most profitable action to take for a particular call. However, this model, which was used for simulations, concentrates on the algorithms for admission control algorithms and is not intended for an a-priori analysis of the overall behaviour of the network.

Ahlfors [1] proposes a queuing model for analysis of overload control strategies, but this model is tied up in the analysis of the operation of the control strategies themselves.

Wohlin and Nyberg [27] present a set of reusable simulation models for the performance analysis of an IN structured network. The motivation for this work is the development of a tool for the examination of the effect of the introduction of new services to the network. The outputs of the tool, are the loads upon the resources in the network and the delays experienced by service users. The inputs to the tool, are a description of the new service, the service usage profiles associated with it and the configuration of the network and the services. The authors describe a general simulation model for teletraffic analysis but do not provide any detail in terms of how this model is developed for an IN structured network.

Becker et al [28] describe the development of a simulation tool for service creation and deployment. This tool analyses the execution of value added services in an IN, in order to predict the services behaviour in the real network. The services are analysed through use of animation and simulation. Animation is useful in understanding how the service behaves, but it does not predict anything about performance. The performance of the service may be predicted through simulation of the behaviour of the network under certain load conditions. The simulation yields results, such as, the mean response time (i.e. the time needed to process a service request). However, the authors do not provide a

detailed description of the simulation model which was used in the tool, nor are any detailed results presented.

Galletti and Grossini [29],developed a simulation of the operation of traffic controls in the IN. The work focused on the comparison of the operation of two overload controls (Automatic Call Gapping and Focused Destination Overload Control), with the case in which there were no traffic controls at all. The simulation reproduced the behaviour of the network for POTS and Mass Calling services for the traffic controls, in order to analyse the mean response time of the digital exchange and the loss probability of a call due to congestion. The simulation model used a network of queues, in which each class of service at each node, was handled by a separate queue. The authors do not describe the particular queue types which were used at each node to model the service processing.

Folkestad and Emstad [30] propose a token based load control strategy for the SDP. In order to analyse this scheme the authors propose a model of a network with a SCP and multiple SDPs in which the resources are modelled as follows :

- The SDP is modelled as a M/M/1 queue.
- The SCP and the network delay between the SCP and the SDP are modelled using an infinite server with exponentially distributed service time and Poisson arrivals.

This forms a network queuing model which is solved using Bard-Schweitzer approximate methods. The result of this approximation, is the number of tokens which must be allocated to each of the SCPs and services in the network.

To conclude, it can be seen that much of the work in IN performance modelling has concentrated on the analysis of overload conditions and the modelling of congestion control strategies to counter such situations. However, some of the models proposed for overload analysis may also be applied to steady state analysis of the network. There has been much less modelling of IN structured networks in steady state conditions. Of the published work, that of Leever et al [18], Lodge et al [22],[23], and Schopp et al

[20],[21], provide detailed models of IN specific processing in order to estimate mean resource response times and mean service response times[9].

2.5 The Future of the Intelligent Network.

IN structured networks are being increasingly deployed across the world as a means of developing flexible new services, faster and of moving towards a vendor independent service platform. This trend is likely to continue as further global deregulation causes competition to increase, and in doing so, requires the operator to deliver better and more innovative services in a shorter time frame.

Increasing customer sophistication will also create requirements for newer and more advanced services. Future developments in the network may result in most calls in the network requiring specialised call processing. For example, *Personal Communications Network* (PCN) services may require that each call undergoes specific call processing on call set-up. This means, that in such a network, every call is an IN call. Such call processing could easily be performed by an IN structured network. As mobile services evolve towards *Personal Communication Services* (PCS) and *Universal Personal Telephony* (UPT), there will also be a requirement for intelligence in the network to support these services. Already, IN structured networks are being used in conjunction with GSM networks to provide additional services to the plain old mobile telephony service.

Thus, it can be seen that there is a requirement for intelligence in the network and the IN is well placed to be a solution to these requirements. However, there are other factors which may militate against it. One of these factors is the increasing importance of Internet based services, in which the service logic is not centralised in the network, but rather distributed towards the end points of the network and into the terminals. Examples of this trend are, the proliferation of services using the *World-Wide Web* (WWW), and the growing incidence of service applications written in java, downloaded from the network and run upon the terminal. In fact, the network terminal itself may undergo radical change

---

[9] Service Response Time in the context of the IN models is the same as PDD.

in the near future and many future network terminals will be low cost computers, with very little data storage and a network connection. Such computers will act as intelligent terminals and would be quite capable of implementing some of the services which are currently identified as part of IN Capability Set 1 (e.g. Abbreviated Dialling, Televote).

Another problem with IN is the pace of standardisation efforts in this area. Capability set 1 [2] defines a very narrow set of services which are confined to single ended narrowband circuit switched services. Capability Set 2 is still in the development process by standards bodies such as ITU-TS and ETSI. In the meantime, many vendors are offering proprietary extensions to CS1 which allow the development of a wider set of services then would otherwise be possible with CS1. The problem with this scenario is that once an operator uses such proprietary service logic in the network, then it is tied to that vendor while it continues to use the service and, in doing so, loses one of the stated benefits of an IN structured network in the first place. Additionally, the pace of development of services in the Telecoms sector means that many service providers are unable to wait for new standards to come on stream, but must develop new advanced services which standard IN is unable to provide now, particularly in the area of data and multimedia services.

Another concept, which will have an influence on the future of IN, is the *Telecoms Information Networking Architecture* (TINA) [31]. The IN concept enabled the removal of service logic from local switches into specialised nodes in the network and signalled a coexistence between traditional Telecoms equipment and computing equipment in the network. TINA defines an architecture for a service platform in which service logic is distributed throughout the network and service processing occurs in a distributed computing framework. The TINA architecture allows for service logic to be distributed as it is in an IN structured network, but in TINA it need not necessarily be structured thus. In a TINA network, the distribution of the service logic across the network resources will be more flexible and transparent. For this reason, it may be that an IN structured network will provide a migration path towards a full TINA structured network.

In summary, IN is well placed to take advantage of the expected advanced service requirements which should accompany the increased competition between operators and increased customer requirements. However, much work must be done in the near future if

IN is to compete with other technologies such as Internet based services. In the longer term, IN could provide a migration path for operators towards networks based upon the TINA architecture.

One of the major advantages with the IN, is the flexibility which it offers the operator in creating new services and provisioning them in the network at a much faster rate. However, as will be shown in Chapter 4 this also means that the operator is faced with greater challenges in managing the performance of the IN structured network and it is this motivation which leads us to define a performance model of an IN structured network.

Chapter 3

3. Queuing Theory

Queuing theory developed from the need to develop models to predict behaviour of physical systems that service randomly occurring demands. Queuing models are widely used for physical systems in order to measure characteristics of the system, and some of the earliest applications were in the area of teletraffic modelling. The characteristics being modelled might, for example, be a measure of the mean time that a customer will spend waiting in the queue, an estimate of the mean length of the queue, or a measure of the utilisation of the service resource.

In simple terms, a queuing system is one in which customers arrive at a service facility, wait some period of time for service and depart after being served. The queuing system may be made up of more than one queue and may also serve different categories (or classes) of customer. In the case where there is only one server, the system is referred to as a single server queue and where there is more than one queue, the system is referred to as a queuing network. In general, a queue is characterised by [32]:

- The *arrival pattern* of customers to the system, which is described by the distribution of the arrivals and the mean number of arrivals in some period of time.
- The *service pattern* of servers, which is described by the distribution of the service times and by the mean number of customers served in some period of time, or as the mean time required to serve a customer (the service time).
- The *service discipline*, which describes how customers in the queue are selected for service.
- The *queue capacity* or *queue length*, which is important where only a finite number of customers are allowed wait for service.
- The *number of servers*, which is the number of parallel servers which may serve customers from the queue simultaneously.
- The *number of service stages*, which may occur in cases where a customer requires service sequentially from cascaded servers.

The remainder of this chapter will define some fundamentals of queuing theory, before discussing single server queues, queuing networks and approximations.

3.1 Queuing Theory Fundamentals.

3.1.1 Stochastic Processes

A *stochastic* or *random process* is defined by Kleinrock [33] as being "*a family of random variables* $X(t)$ *where the random variables are indexed by the time parameter* t ". Random processes may be classified according to:

- Their *index* parameter, which may be finite and discrete, or continuous.
- The *statistical dependencies* among the random variables $X(t)$, for different values of the index parameter t. This may be specified by the *joint distribution* function $f_x(x,t)$, of the random variables $X = [X(t_1), X(t_2),\ldots,X(t_n)]$: $f_x(x,t) = P[X(t_1) \leq x_1,..,X(t_n) \leq x_n]$ for all $x = (x_1, x_2,..,x_n)$ and $t = (t_1, t_2,..t_n)$.

The joint distribution function, describes the dependencies among the random variables and is used to characterise different types of stochastic processes. Some of these processes are :

- The stationary process, in which $f_x(x,t)$ is invariant to shifts in time for all values of x,t;
- The independent process, where the set of random variables $\{X_n\}$ are independent of each other;
- The Markov process, in which the next state (value) of the process, depends only on the current state and not on the history of the previous states;
- The birth death process, in which state transitions may only occur between neighbouring states.

3.1.1.1 The Markov Process

The Markov process [34], provides a means for modelling physical systems, where the state changes at arbitrary instants in time. A Markov process is a stochastic process, with a discrete number of states. The future state of the process is dependent only on

the current state of the process. The past history of states does not have any impact upon the future of the process. This is known as the *memoryless property* and is found in several processes, the most important and common of which is the Poisson process.

3.1.1.2 The Poisson Process

The *Poisson Process* is a counting process, for a number of randomly occurring events, in a given interval of time. The events are referred to as arrivals and could, for example, represent the number of telephone calls arriving at a switch. In a Poisson process, the mean number of arrivals in time t is $\lambda * t$, where λ is the mean arrival rate. The characteristics of a Poisson process are :

- Stationarity. The number of arrivals in intervals of equal length are identically distributed. That is, the number of arrivals in interval 1 is the same as the number of arrivals in interval 2 if the intervals are of the same length. $X(t_2 - t_1) = X(t_3 - t_2) \Leftrightarrow |t_2 - t_1| = |t_3 - t_2|$.
- The mean time between arrivals is $1/\lambda$. So if there are λ arrivals in one second, then $1/\lambda$ is the mean time between arrivals.
- The interarrival times are independent and have an exponential distribution.

The last property is an important one. Assuming the number of arrivals in an interval to be a Poisson random variable, is equivalent to assuming that the time between two successive arrivals is an exponentially distributed random variable $= 1 - e^{-\lambda t}, t \geq 0$, where λ is the mean arrival rate for customers. The mean interarrival time is $1/\lambda$ and its variance is $\sigma^2 = 1/\lambda^2$.

Conversely, exponential arrival times in a process imply that the number of arrivals in an interval is a Poisson random variable and that the process is Poisson. This is particularly useful as the exponential distribution is memoryless i.e. the time to the next arrival is independent of the time elapsed since the last arrival.

3.1.2 Little's Law

Little's law is a very useful relation which applies to any queue or contention system in a steady state [34]. It relates the mean number of customers competing for a

resource $\overline{L}$, with the mean time that a customer waits in the system to complete service $\overline{W}$ and the mean arrival rate of customers λ. Little's law is stated as $\overline{L} = \lambda \overline{W}$.

3.1.3 Kendal's Notation

Kendal's notation is a shorthand notation used to describe a queue in the form of A/B/C/D/E, where :

- A denotes the arrival process;
- B the service process;
- C the number of parallel servers and may take a value between 1 and ∞;
- D the queue length and may take a values between 1 and ∞ and
- E the service discipline.

The symbols of interest in this report and their values are presented in Table 3-1.

Symbol	Meaning
M	Markov
D	Deterministic
G	General

Table 3-1 - Symbol Meanings.

3.2 Single Server Queues

3.2.1 The M/G/1 queue

The M/G/1 queue has a Poisson arrival process and general service times and has a single server with *First In-First Out* (FIFO) service discipline. From Little's law, the probability that there is at least one customer in the queue is λ/μ (i.e. $P[X_n \geq 1] = \lambda/\mu$. This can be explained intuitively, as the arrival rate is λ and the service rate is μ. If the service rate of the server is much larger than the arrival rate, λ/μ is much less than 1. This is to be expected when the capacity of the server is much greater than the arrival rate of the customers. When the arrival rate and the service rate are of the same order of magnitude, then λ/μ is close to one, which is also to be expected in a heavily loaded server. When the arrival rate is greater than the service

rate, the server cannot cope with the demand. If the queue length is unbounded, the number of customers will tend towards infinity. If the queue length is bounded, the queue will fill up and customers may be lost. The queue is in the steady state (or equilibrium) when $\lambda/\mu < 1$. Thus, the mean arrival rate must be less than the mean service rate for the queue, or the number of customers will keep increasing. The mean queue length can be found using the *Pollaczek-Khinchin* formula :

$$\overline{L} = E[X_n] = \rho\left[1 + \frac{\rho(1+Ks)}{2(1-\rho)}\right],$$

where :

- ρ is the utilisation of the server and is equal to λ/μ;
- λ is the mean arrival rate;
- μ is the mean service rate;
- Ks is the *square of the coefficient of variation* (SVC) of service time $Ks = \sigma/(\overline{x})^2$ (i.e. the ratio of the square of the variation in service time to the square of the mean number of customers in the queue).

Pollaczek-Khinchin allows the mean queue length to be found when only the utilisation factor ρ, and the first two moments ($\overline{X}$ and $\overline{X}^2$) of the service time distribution are known. Note that when the utilisation of the service ρ is small, the mean queue length is small, which is as expected. Once the mean queue length $\overline{L}$ has been found, then the waiting time can be found using Little's law $\overline{L} = \lambda\overline{W}$. As both $\overline{L}$ and λ are known $\overline{W}$ may be found as $\overline{L}/\lambda$.

Because the distribution of the service time is represented, the formula will model the situations where the service time is deterministic (constant service time, which is the M/D/1 queue) and where the service time is exponentially distributed (i.e. M/M/1 queue). This can be shown as follows :

- In the M/D/1 queue, the service time distribution is a constant $\overline{X}$. Thus, the variation of the service time is zero, which means that Ks is also zero. The mean queue length is found using the Pollaczek-Khinchin formula:

$$\bar{L} = \rho + \rho\left[\frac{1+0}{2(1+\rho)}\right] = \rho + \frac{\rho^2}{2(1-\rho)}, \text{ or } \bar{L} = \frac{\rho}{(1-\rho)} - \frac{\rho^2}{2(1-\rho)}, \text{ which is as}$$

expected for the M/D/1 queue [33].

- In the M/M/1 queue, the service time distribution is exponential, which means that Ks is one. Using the Pollaczek-Khinchin formula, the mean queue length is $\bar{L} = \rho + \rho\left[\frac{1+1}{2(1-\rho)}\right] = \frac{\rho}{(1-\rho)}$, which is as expected for the M/M/1 queue [33].

Note that the queue length for the M/M/1 queue is greater than that for the M/D/1 queue by a factor of $\frac{\rho^2}{2(1-\rho)}$.

In the Pollaczek-Khinchin formula, it can be seen that the queue length increases with the coefficient of service time.

3.2.2 The G/G/1 Queue

The G/G/1 queue is a further generalisation of the single server queue. It allows an arbitrary arrival rate, an arbitrary service distribution and has a single server. As the arrival process is not Markovian, then many of the simplifications which may be applied when Markovian arrivals occur, are lost. This means that it is much more difficult to determine the characteristics of such queues. Many of the techniques used to analyse such queues are based on spectral analysis methods [33]. These methods are very complex and out of the scope of this discussion. However, *Kingman's formula* provides a relatively simple means of estimating the upper bound on the mean queue length. As before, once the mean queue length is found, the mean waiting time may also be estimated through application of Little's Law. Kingman's formula is a generalisation of the Pollaczek-Khinchin formula and is formulated as :

$$\bar{L} = E[L] = \rho\left[1 + \frac{\rho(ka + ks)}{2(1-\rho)}\right],$$

where :

- $\bar{L}$ is the mean queue length;
- ρ is the utilisation of the server and is equal to λ/μ;

- λ is the mean arrival rate;
- μ is the mean service rate;
- Ks is the square of the coefficient of variation of the service time;
- Ka is the square of the coefficient of variation of interarrival time.

This generalisation may be demonstrated by considering the application of Kingman's formula for exponential arrivals in a single server queue (i.e. the M/G/1 queue for which Pollaczek-Khinchin estimates the mean queue length). For an exponential distribution, the SVC of the interarrival time is unity, thus $Ka = 1$. Putting this into Kingman's formula results in $\bar{L} = \rho\left(1 + \frac{\rho(Ka + Ks)}{2(1-\rho)}\right)$, which is equivalent to the Pollaczek-Khinchin formula.

3.3 Queuing Networks

In many situations, the physical system to be represented by the queuing model is made up of several (i.e. more than one) resources interconnected with each other in some way (e.g. a telephone network or a computer.) Some such systems may be modelled by a single queue which views the system like a 'black box'. The external characteristics of the system are modelled by this queue, but no information about the dynamics of the individual resources can be found from the model. Alternatively, a network of queues may be used to model the system. This queuing network contains a queue to model each physical resource (or group of resources) in the system. Each queue is interconnected in the same manner as the resources in the physical system and represent the possible paths of messages or customers in the network. It is assumed that the transition times for a customer between queues is zero.

Queuing networks may be categorised into three classes, based on whether the network exchanges customers with the exterior :

1. An *open network* has at least one arc entering and one arc leaving[10] the graph of the network. In other words, customers may arrive from the exterior into the network and pass through the network to the exterior.

[10]The open network must have both an arc entering and leaving. It cannot have one without the other or the network will either contain an infinite number of customers or zero customers.

2. A *closed network* has no external arcs. This means that the network has a fixed number of customers which move around the network continually.
3. A *mixed network* contains several different *classes of customer*. The network is mixed if it is open for some classes of customers and closed for other classes of customer.

As the physical systems of concern (i.e. telecoms networks) interact with the exterior, the remainder of this discussion concentrates solely on open queuing networks. Section 3.4.1 discusses the simplest possible queuing network - a Jackson Network. Section 3.4.2 discusses BCMP networks, while section 3.5 details approximation methods which may be used to determine characteristics of the queuing network.

3.3.1 Jackson Networks

A *Jackson Network* is the simplest possible form of queuing network and has a stationary solution in product form, for the joint probabilities of the lengths of the queues (i.e. $P[X_1 = k_1] \cdot P[X_2 = k_2] \ldots P[X_N = k_N]$. It is written in the form of a product of the marginal distributions of each queue (i.e. $P(k_1, k_2, \ldots, k_N) = P_1(k_1) \cdot P_2(k_2) \ldots P_N(k_N)$. Jackson networks have the following characteristics :

- The queuing network contains N service stations with a queue of unlimited size at each station. Each station i, contains m_i exponential servers with parameter μ_i.
- The movement of a customer from one station to another is represented by a Markov chain and there is no distinction between the stochastic characteristics of the different customers. This means that the probability of being in state Y_{i+1} at instant $i+1$, is dependent only on the state at instant i.
- Customers arrive according to a Poisson process and pass towards the exterior.
- In the stochastic case, the service times will always be independent of each other and are distributed exponentially, having parameters which depend on the length of the queue.

- A customer completing service at a station makes a probabilistic choice of either leaving the network or entering another station. This choice is dependent only on the current state and not on the past history.

It should be noted that while the external arrivals to a queue follow a Poisson distribution, arrivals from other stations in the network may not be Poisson.

Jackson's Theorem [33] shows that for all stations in the network, the i^{th} station in the network behaves as if it were an independent *M/M/m* system, with Poisson input rate λ_i. The state vector for the network is $(k_1, k_2, ..., k_N)$, where k_i is the number of customers in the i th station. The equilibrium probability associated with this state is denoted by $P(k_1, k_2, ..., k_N)$. The marginal distribution of finding k_i customers at station i is denoted by $P_i(k_i)$. The joint distribution for all the station can be factored into the product of each of the marginal distributions: $P(k_1, k_2, ..., k_N) = P_1(k_1)P_2(k_2)...P_N(k_N)$. $P_i(k_i)$ may be found from the solution of the *M/M/m* system for each station i. This is not to say that the network can be decomposed into individual *M/M/m* queues, each with a Poisson input stream with mean rate λ_i. It can be shown, that the flows between stations in the network are not Poisson[32], particularly where there is feedback. However, Jackson's theorem does hold and the network appears as if it's nodes are independent *M/M/m* queues.

3.3.2 BCMP Networks

Jackson networks were the only queuing network models to be used until the mid 1970s for modelling and evaluating computer systems. This was because of the simplicity of the network models and the applicability of the exponential distribution. In some cases, distributions similar to the exponential distribution can be replaced by the exponential distribution without significant change to the solution [35]. However, the need for modelling of computer networks started research into simple solutions for new networks. As a result of this work, the BCMP network models were introduced. *Baskett, Chandy, Muntz and Palacios* (BCMP) networks retain the solution in product form, in the equilibrium state, while introducing different classes of customer and different service disciplines.

3.3.2.1 Service disciplines

In Jackson networks, the queue at each station is ordered by the sequence in which the customer arrived. This is known as the *First Come - First Served* (FCFS) service discipline of *First in - First Out* (FIFO). Other service disciplines are :

- *Processor Sharing* (PS), where a customer receives $1/k$ seconds of service/second, if there are k customers at the station. All customers receive equal share of the processor time and processing quanta are assigned to each customer in turn.
- *Infinite Server* (IS), where the number of servers is infinite (or is such that there is always a server empty). This means that a customer entering a station is served immediately.
- *Last Come - First Served* (LCFS) *with pre-emption.* Here there is a single server with absolute priority assigned to the newly arriving customer. The service is pre-emptive - each time a new customer arrives at the station the currently served customers service is interrupted and that customer is placed at the head of the queue. When the new customer has been served the interrupted customers service resumes where it was left off.

3.3.2.2 BCMP Theorem

The *BCMP Theorem* [35],[34] states that queuing networks with service disciplines which are *FIFO, PS, IS, LCFS* and with several classes of customer have a product form solution for the steady state, joint probability distribution, of the station states. In BCMP networks, it is only necessary to know the means of the service time distributions in order to find the steady state joint probability distribution. Because the joint probability is the product of the marginal probabilities the network may be studied station by station. The BCMP network has the advantage over a Jackson network of allowing different classes of customer and different service disciplines. However, as in Jackson networks, where large networks with multi-stage stations the normalisation constant may be difficult to calculate.

3.4 Approximation Methods

The only networks for which a solution is actually known, in an explicit form, are Jackson and BCMP networks. However, as mentioned in the previous section, where

large networks are concerned, the calculation of the normalisation constant can be difficult and may in some cases be impossible. Where only networks with the FIFO service discipline are of interest, then networks of more than three queues for which the solution is known have the following characteristics [35] :

- The service time distributions are negative exponentials;
- External arrivals are Poisson;
- There is only one class of customer or where there is more than one class, the service times are independent of the class;
- All queues have unlimited capacity.

The number of networks possessing these characteristics is relatively small and the solutions are in the form of joint probabilities. It is for this reason that approximation methods must be used.

Several approximation methods exist to obtain solutions for queuing networks. Amongst them are [35] :

- The *Decomposition Method,* which considers each queue within the network in turn, with the external and internal (i.e. those emanating from other stations in the network) streams approximated.
- The *Aggregation Method,* which considers the network on a group by group basis, where each group has weak interaction with the exterior.
- The *Mean Value Analysis Method,* which examines the first moments of the mean numbers in each queue.
- The *Isolation Method,* in which each queue is isolated in turn and examined while taking into account the effects of interaction with the exterior.

Other numerical and iterative methods also exist, but they suffer the disadvantages of only being able to study small networks, or of having uncertainty about how close the approximation is to the actual solution. They generally require long computation times. The remainder of this discussion will focus upon the decomposition method.

3.4.1 The Decomposition Method

The *Decomposition Method* is one of the most widely used approximate methods [34]. It is particularly useful when isolated features of the network (such as, contention for passive resources or blocking) cannot be represented in product form. The

decomposition method provides an exact solution for product form networks (a network whose solution may be found in product form e.g. a Jackson or BCMP network). It provides a good solution for 'almost product form networks' and produces good results even when many of the criteria for a product form solution are violated by the network. The use of the method involves four main steps [34] :

- Decompose the network into appropriate sub-networks;
- Solve each sub network independently;
- Aggregate the sub-network solutions to form an approximate solution of the network and
- Validate the aggregate solution with a simulation model.

There are various forms of the decomposition method which may be classified by the approach taken to decomposition, these vary from:

- Methods which solve each sub-network individually and approximate the solutions. In this case, the sub-network should not in general contain either a single station, or all the stations in the network, as in either case, no progress towards a solution will have been gained from the decomposition. Instead, a sub-network contains *'clusters'* of related stations, which have a high rate of transition with other stations in the cluster and a relatively low rate of transitions to stations outside the cluster.
- *Hierarchical Decomposition,* in which decomposition is performed at a number of levels. At the lowest levels, sub-networks are further decomposed into sub-networks, which are represented by single servers. These are then combined to form the sub-networks which form the decomposition at the higher levels.
- *Input/output flow decomposition,* decomposes the network into sub-networks, but rather than solving each of the sub-networks in isolation and then approximating the overall solution, the flows between a particular sub-network and each of the other sub-networks are considered when solving for that sub-network. In this case, because the sub-networks are not viewed in isolation, there may be significant flows between the sub-networks and the sub-networks may only contain one station. This provides a very flexible and powerful means of approximating the

solution for the network. It is this particular variation of the method which will be discussed further.

Harrison and Patel [34] define the steps to be taken, in the input/output decomposition of a network as :

1. Decompose the network into sub-networks, some of which can be single server centres;
2. Analyse each sub-network in isolation in terms of arrival and departure processes of sub-network;
3. Approximate all non renewal processes by renewal processes;
4. Consider the first two moments of the interarrival times of all processes (i.e. the mean and coefficient of variation of interarrival time) and
5. Produce a computational algorithm to solve the model.

As before, the solution would be validated through use of a simulation. This model will simulate the queuing model which was decomposed and will validate the assumptions made while decomposing the network.

Step three above, involves the modelling of inter-sub-network flows as renewal processes. This is based upon the assumption that the departure process from any station i is a renewal process (i.e. the time interval between two successive departures does not depend upon the preceding intervals). This assumption is valid in the case where the arrivals are Poisson and the services are distributed exponentially or when the station is saturated [35].

As an intuitive justification of this assumption, we have already seen the memoryless property of the exponential distribution - thus independence of the inter-arrivals is consistent with this property. In a saturated server, there are always more customers than there are servers. This means the mean interval between successive departures from the station, is the mean service time for the customer when all servers are busy.

It should be noted that, as in Jackson and BCMP networks, the internal flows between stations do not, in general, have Poisson or renewal processes. However, the decomposition method does provide a good approximation, particularly if the subnet partitioning is performed well.

This is an important issue in the application of the decomposition method and to this end Harrison and Patel [34] provide a set of assumptions, which in general allow the solution to be accurate :

- Transitions between sub-networks are single step transitions - they only involve single customers. Thus, if a station supplies bulk service (more than one at an instant of time) to another station, then both stations should be in the same sub-network.
- The sub-network behaviour depends only on the sub-network state and is independent of the state of the other sub-networks. This means that if a station is blocking, then all the nodes which it blocks must share the same sub-network.
- The input/output behaviour of a customer through each sub-network is independent of the state dependent routing decisions made during its time in the sub-network. This allows the sub-network to be viewed from the outside as a black box which has consistent behaviour towards a customer.
- Transitions between sub-networks involving a customer do not depend on the behaviour of another customer.

3.4.1.1 Formulation of the Decomposition Method.

To present the formulation of the Decomposition Method, the following notation is introduced (taken from [35]):

- The *Square of the Variation Coefficient* (SVC) of the interarrivals between two successive departures from station i is denoted C_i;
- The time interval between two arrivals at station i is denoted A_i;
- The service time at station i is denoted S_i;
- A customer departing station i moves to station j with probability p_{ij};
- The time interval between two successive departures from station i is denoted τ_i and may be approximated by

$$\tau_j = \begin{cases} S_j, \textit{with probability } \rho_j = \lambda_j / \mu_j \\ \mathrm{A}_j + S_j \textit{ with probability } 1-\rho_j \end{cases}.$$

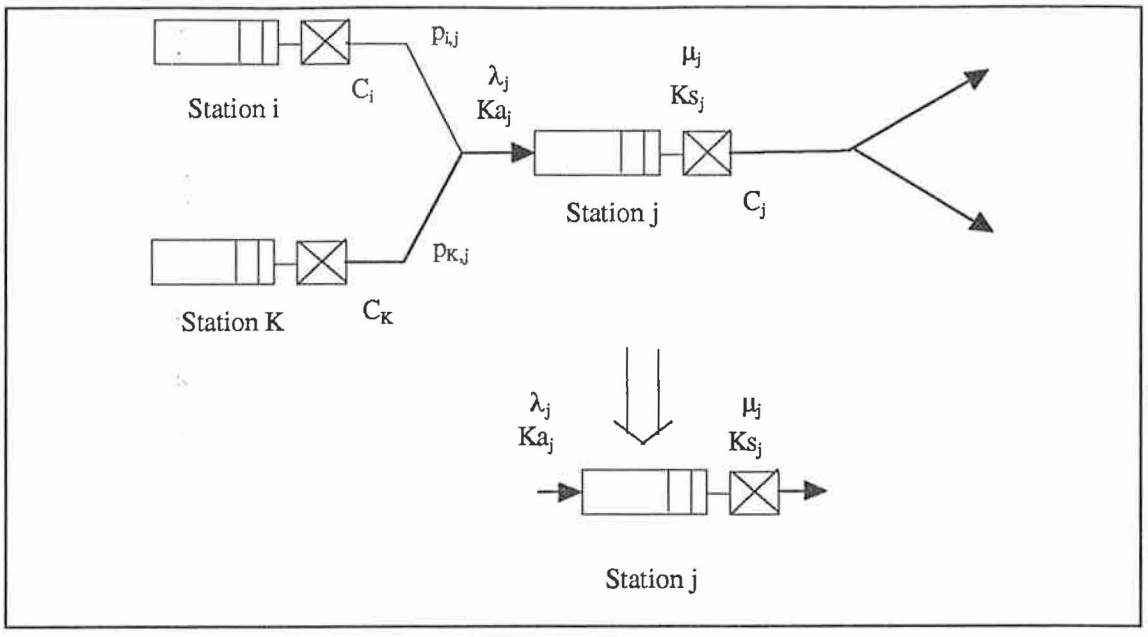

Figure 3-1 - Decomposition of Queuing Network

Taking expectations, a relation is obtained between C_j and $E\left[A_j^2\right]$. This is expressed in terms of the SVC of the number of customers entering the station j at a time t, Ka_j [35]. This allows C_j to be written in the form :

$$Cj = -1 + \rho_j^2\left(Ks_j + 1\right) + \left(1 - \rho_j\right)\left(2\rho_j + 1 + \frac{1}{\lambda_j}\sum_{i=0}^{N}\left[\left(C_i - 1\right)p_{ij} + 1\right]\lambda_i p_{ij}\right) \text{ for } j = 1..N$$

C_j can be seen to depend on the SVC of the service time at station j and on the variations in the arrivals from the other stations in the network, from which station i receives its input. This in turn depends on the SVC of the departures from each of these stations. C_0 is the SVC of the interarrivals from the outside world to the network and Ks_j is the SVC of the service time at station j. The queue at each station j can be studied separately, using the values of the first two moments of the arrival and service processes. The mean length of the i^{th} queue $\overline{L_i}$ can be approximated for each station by Kingman's formula:

$$\overline{L_i} = \rho_j\left(1 + \frac{\rho_j\left(Ka_j + Ks_j\right)}{2\left(1 - \rho_j\right)}\right)$$

The response time $\overline{R_i}$ may then be found through Little's law $\overline{R_i} = \overline{L_i}/\lambda_i$.

In summary, the Decomposition Method studies each queue in the network on a queue by queue decomposition of the network. In this method, the distributions of the

service times and interarrival times at each station i are approximated by the total mean rate of arrivals λ_i and the *square of their variation coefficients* (i.e. the variation multiplied by the rate squared). In this way, the arrival streams from other station in the network and from the exterior can be approximated. The mean queue length is then found using Kingman's formula, which is a generalisation of the Pollaczek-Khinchin formula.

3.5 Application of Queuing Models

Having presented a general overview of queuing theory, it now remains to mention how such techniques may be applied to real physical systems. Gelenbe and Pujolle [35] identify a methodology for developing queuing models. It consists of three phases:

1. The *analysis* of the physical system to determine its characteristics and behaviour. This leads to a model of the physical system in terms of queues (i.e. a queuing network).
2. Formulation of the set of equations which govern the model. The parameters of the model are identified through observation of the physical system or through domain knowledge.
3. Solution of the system of equations.

There are no hard and fast rules for the first two phases. In particular, phase two is often very difficult. It is necessary to determine the set of system states from the model and calculate the relationships which exist between the probabilities of these states. This is generally only possible where the model is Markovian (i.e. it possesses the memoryless property). Where the model is not Markovian, there are four alternatives which may be taken [35] :

1. To define embedded Markov sub-chains within the model;
2. To make the model Markovian by the addition of virtual states;
3. To simplify the model. For example, a model might be simplified by assuming that a server is exponential, a stream is Poisson, or a queue length is unlimited. In this case, it is necessary to determine, where possible, the error introduced through simplification.
4. To modify the model so that it may be used, by either adding or deleting elements of the model.

3.5.1 Simulation

In cases where it is difficult to formulate the set of equations governing the model, simulation may be a viable alternative. Simulation is also very useful in verifying that an analytic approximation is a good approximation. Also, many analytic results provide information about the steady state behaviour. If the transient behaviour, or behaviour with respect to time is of interest then simulation may be a useful tool. However, simulation also has its problems. These include the fact that a simulation is run on a particular set of data and the results are particular to that data. For a different set of data the simulation must be re-run. Also, it is necessary to consider the duration of the simulation and how many times it must be run to gather the results required. Despite these drawbacks, simulation is a useful tool, in both verifying an analytic approximation and in modelling the behaviour of a system for which an analytic solution could not be found.

Chapter 4

4. A Model of the Intelligent Network

4.1 Introduction

The aim of the work described in this report is to produce a model of the performance of an IN structured network. In Chapter Two, both the requirements of such a model and the existing models have been presented. In this chapter, the new performance model of the IN which has been developed is described. This new model builds upon the work of Leever et al [18] (as the state of the art at the time the model was being developed). In the model proposed by Leever et al, the inter arrivals at each of the IN resources are exponentially distributed (i.e. a Poisson process). However, this assumption may not always hold as a service message arriving at a particular resource (e.g. SCP) will typically have come from another service resource (e.g. SSP). This means the inter arrival times of service messages will depend not only on the inter arrival times of calls to the network, but also on the nature of the service times at the other resources and how heavily loaded these resources are. The model proposed here considers the nature of the flows between the network resources, which are important in determining the characteristics of the inter arrival times at each of the IN physical entities. By representing these flows as having general distributions, a truer picture of the inter arrival time distribution can be found.

The model also extends that of Leever et al, by considering general service times rather then just exponential service time distributions. The simulation model implementation and formulation of the analytic model are such that the distribution of the service time can be chosen for each resource. This allows the customisation of the model to cater for the characteristics of different services and IN PE implementations. In the presentation of the model here, the scenario chosen is one in which the service time distributions are deterministic.

At the SCP, the model enables the allocation of capacities to each of the services supported by the SCP. This models SCP implementations, in which the SLPIs for each

service type run concurrently and receive a certain amount of processor cycles. IT also models the situation where congestion control strategies such as throttling are in place at the SCP.

The model was developed in parallel to that of Schopp et al [20],[21] and Lodge et al [22],[23]. The model proposed by Schopp et al, allows consideration of general service times at each of the network resources but in common with Leever et al the authors make the assumption that the inter arrival times are exponentially distributed. Lodge et al have defined a model of the IN which provides significant detail of the behaviours in the SSP. This is because the work is aimed at analysis of overload control strategies implemented in the SCP and SSP. The model presented here overlaps with that of Lodge et al and this author was involved in the development of earlier incarnations of that model as described in [25],[36]. Both Lodge et al and Schopp et al, use the decomposition approximate method in order to formulate an analytic solution to their queuing models. The same method is also used in this model and the characteristics of this formulation are described in Chapter 5.

The remainder of this chapter describes in detail the performance model of the IN, which is divided into two aspects :

1. The services which are implemented upon the network and
2. The actual network resources existing in the network, their respective inter-connections and the configuration of these resources.

4.2 Services upon the Network

It is assumed that there are three services available on the network. The choice of three services allows the examination of the effect of mixing services, while avoiding the overhead of considering how more than three services might be implemented. The model may be easily extended with more services by following the analysis methodology used here. The services concerned are now discussed in detail.

4.2.1 Abbreviated Dialling.

Abbreviated dialling allows the service user to assign a short numeric code to numbers which they use often. By dialling the code the user is connected to the appropriate directory number. The service has two aspects :

- The assignment of directory numbers to codes;
- The use of the service where the user dials in the abbreviated code to initiate a call.

The SIB chains for abbreviated dialling are shown in Figure 4-1 and the corresponding information flows are shown in Figure 4-2. There are two chains for the service, corresponding to the two aspects of the service described above.

- In the first chain, the input data is first verified by the verify SIB, and then the user's table of abbreviated numbers in the SDP is updated using SDM SIB.
- In the second chain, the input data is first verified to ensure it is complete, before being passed onto the Translate SIB, which translates the abbreviated number dialled into the DN it corresponds to.

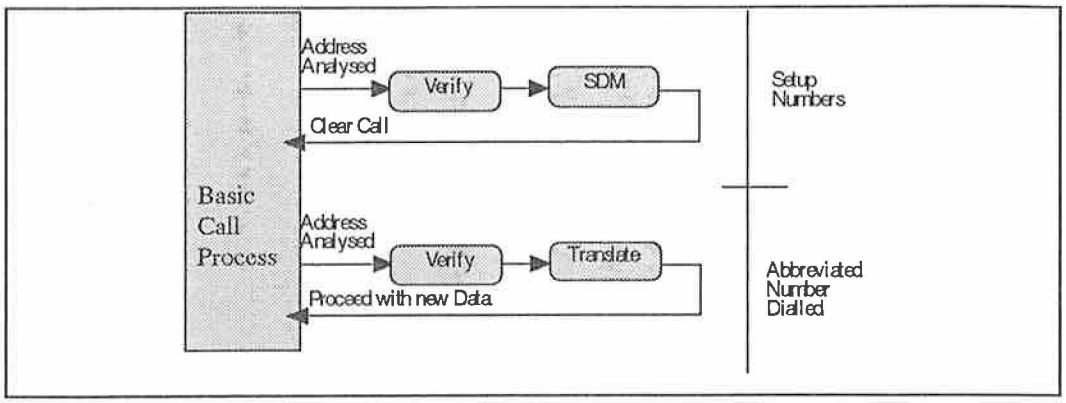

Figure 4-1 - SIB Chain for Abbreviated Dialling Service.

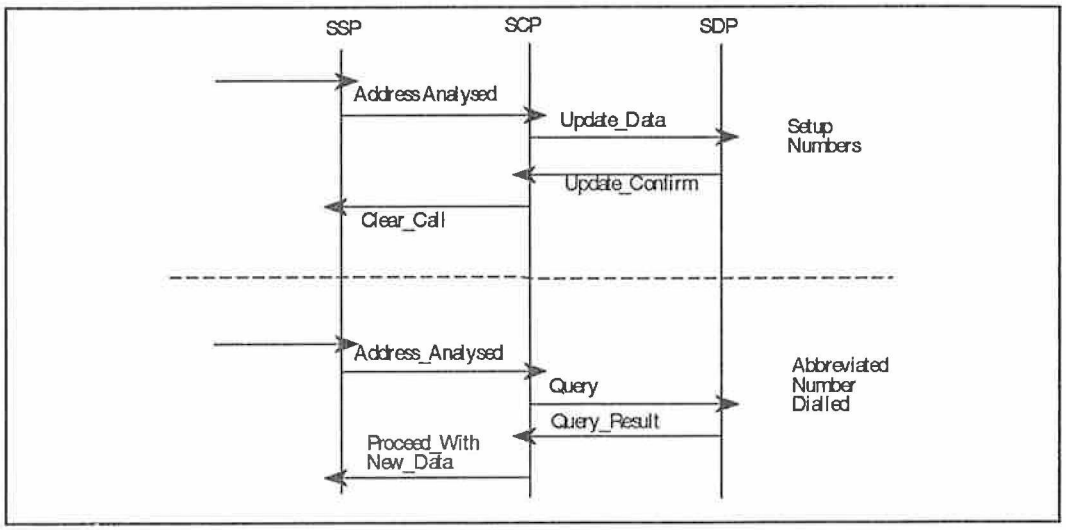

Figure 4-2 - Information Flows for Abbreviated Dialling Service.

4.2.2 Call Forwarding.

Call forwarding allows the service user to unconditionally forward all incoming calls to a particular destination, specified by the user. The service user can then receive all incoming calls at the forwarded address. The calling party is not aware of the fact that the call has been forwarded. The destination to which the call is forwarded can be anywhere within the network, rather than a local area, as is the case in some non-IN implementations.

The service has three aspects :

a) Service activation. Here the service user invokes the forward service and indicates the DN to which it is to forwarded to. All incoming calls are now forwarded.

b) Service operation. An incoming call to the users DN is intercepted and forwarded to the forwarded number.

c) Service Deactivation. The user cancels the forwarding. No incoming calls are now forwarded.

The SIB chains for the Call Forwarding service are shown in Figure 4-3, while the information flows are shown in Figure 4-4. There are three chains for the service, corresponding to the three aspects of the service:

a) The first chain corresponds to service activation. The input from the user is passed to the verify SIB to ensure it is syntactically correct. It is then input to the SDM SIB which updates the SDP with the directory number to which the calls are to be forwarded.

b) The second chain corresponds to the operation of the forward service. The dialled number, towards which the network is attempting to terminate the call, is passed through the *verify* SIB to ensure it is correct. It is then input to the translate SIB, so that the number to which the calls are to be forwarded is retrieved. The call then terminates to this number.

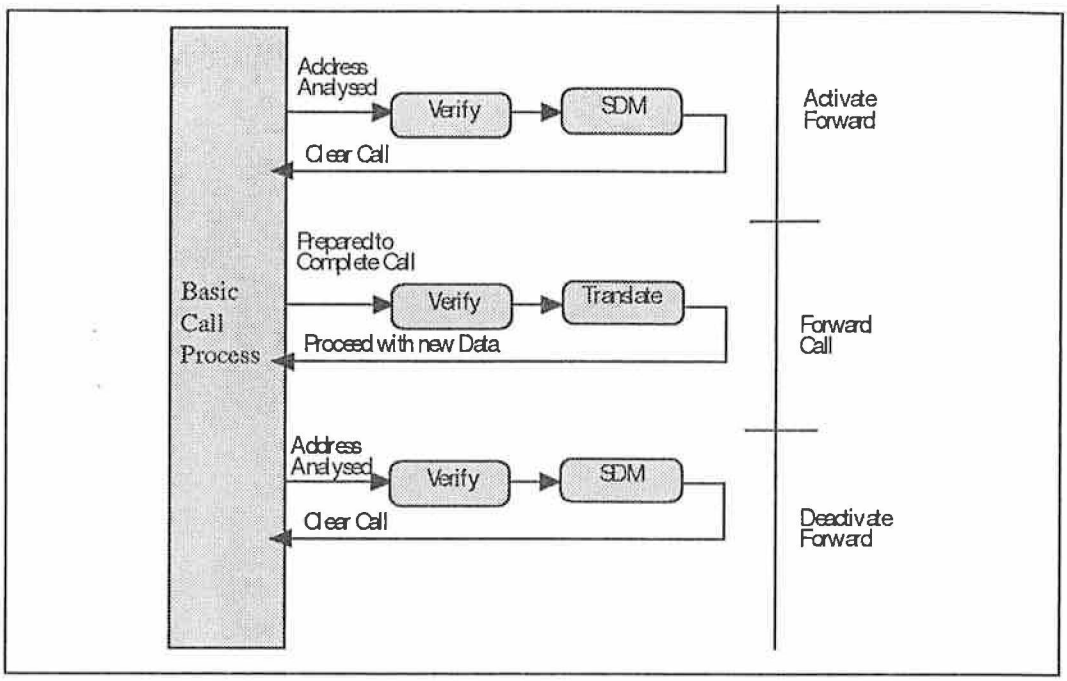

Figure 4-3 - SIB Chains for Call Forwarding Service.

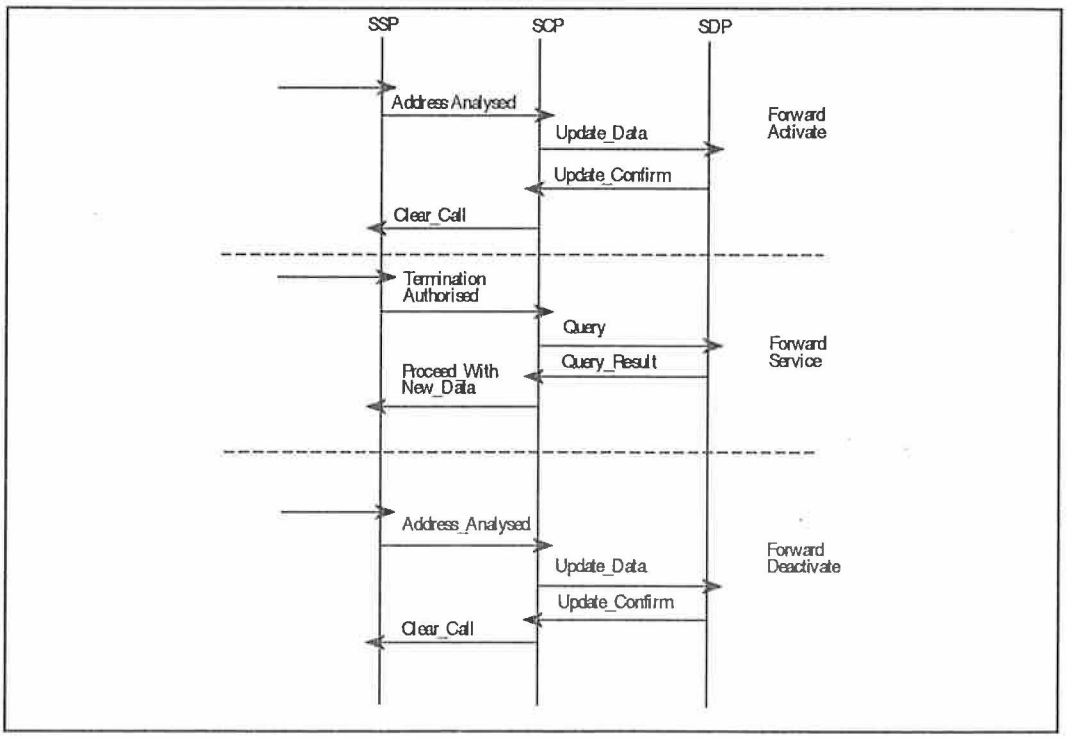

Figure 4-4 - Information Flows for Call Forwarding Service.

4.2.3 Televote.

This service allows the user to take part in a telephone poll. The calling party dials the number and is played a recorded announcement, detailing the alternatives which he/she may vote for. The calling party chooses from the 'menu' and the corresponding vote count is updated. The calling party is then played an announcement to confirm that the vote was successful and the call ends. The SIB chains for Televote are shown in Figure 4-5, while the information flows are shown in Figure 4-6. It is assumed in this implementation that the peg counts (or vote counts) are stored in the SDP. This may not always be the case - in fact it may be much quicker for each SLP instance to keep an internal count of the peg counts, all of which can be added together at the end of the poll to determine the result. This could be done to avoid potentially costly database accesses, particularly, in such a highly concurrent environment. In that situation, the database writes by many SLPIs would result in long periods waiting for locks to be relinquished on the database. This implementation allows us to exercise the SDP more, which is interesting for the model. In the design of the service, it is assumed that such wait times for database access are not a problem.

There is only one SIB chain for the Televote service. On dialling the Televote number, the service data is passed onto the UI SIB, which ensures that the user is played an announcement to ask the user to indicate their preference using the keypad. This data is passed through the verify SIB to ensure that it is correct. From here, it is passed onto the Algorithm SIB, which increments the particular peg count before passing it onto the SDM SIB to update the SDP. Once this has been successful, the UI SIB plays the announcement to confirm that their vote has been registered and the call completes.

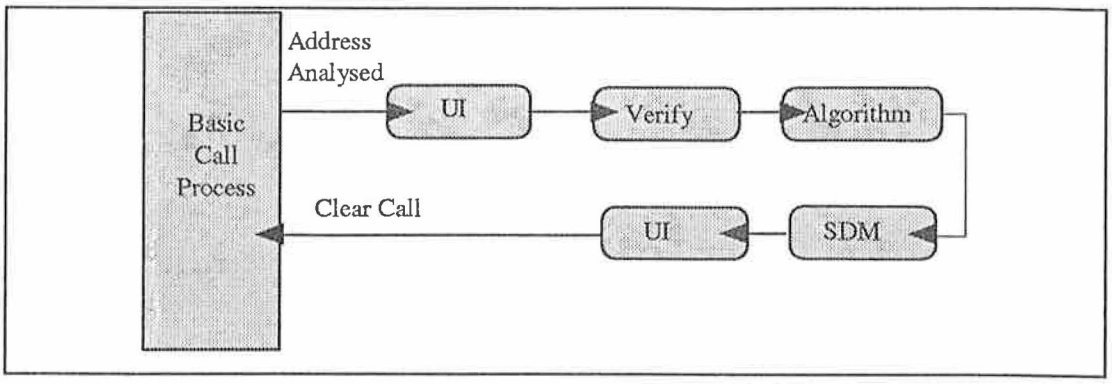

Figure 4-5 - SIB Chain for Televote Service.

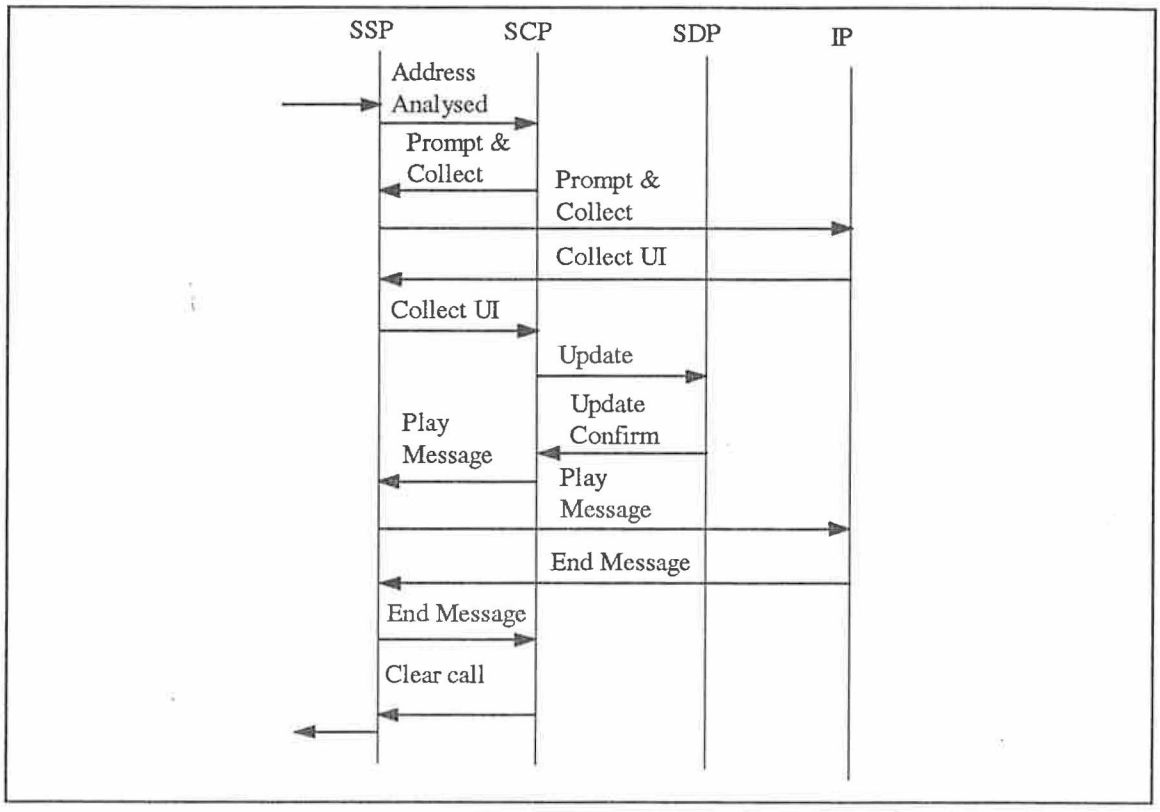

Figure 4-6 - Information Flows for Televote Service.

4.3 Network Resources

In Chapter Two, the IN PEs and the different FEs, which they may contain, were presented. It should be remembered that some of the FEs may be resident in more than one PE and that a subset of the IN PEs can contain all of the FEs necessary for service execution. Here, we define such a subset of four PEs, which contain all of the service execution FEs and are sufficient to support the execution of any CS1 service. The model

of the IN defined here, contains these four PEs (see Figure 4-7) and the following assumptions are made (for a description of the individual FEs the reader is referred back to Chapter Two) :

a) SSP. The SSP contains three FEs:

1) The CCAF;

2) The SSF and

3) The CCF.

For the purposes of a model of IN service execution, the SSP is the entry point for all services to the network. It is assumed that there can be many SSPs distributed throughout the network, providing access to users in the different regions.

b) SCP. The SCP contains one FE - the SCF.

It is assumed that there can be more than one SCP in the network and that each SCP is capable of supporting all the range of services offered in the network. This does not mean that a particular SCP will have requests arriving for every service, but it will be capable of supporting all the services should the need arise. It is also assumed, that the SCP is a computer capable of running multiple SLPIs concurrently and that a certain amount[11] of processing capacity is reserved for each service. This means that one heavily loaded service will not impinge on the other services. This last assumption models a multi-process/thread SCP implementation, where the SLPIs are concurrent processes/threads and is also consistent with what happens on a SCP, with a throttling congestion control mechanism.

c) IP. The IP contains one FE - the SRF.

The IP provides service resource functionality such as speech synthesis and other user interaction functionality. It is assumed that the processing of requests at the IP is uniform (i.e. playing an announcement) and that the time taken for service processing at the IP, is also constant.

---

[11]This amount can be zero if a service is not required at a particular SCP.

d) SDP. The SDP contains one FE - the SDF.

The SDP is the repository for all service information in the IN. The SDP is modelled as one single database in the network. In real implementations, such a database would probably be distributed. However, in many cases, the fact that a database is distributed, can be hidden from the database client, which sees only a single unified database through which it can access all the data it requires. As such, the assumption of a single SDP in the network is a valid one.

The PEs in the network have the following interconnections (see Figure 4-7):

- The SSPs are connected to the SCP over the SS7 signalling network, and to the IP over ISDN links which facilitate the transfer of speech and other service information. The ISDN D channel will also carry the signalling information between the SSP and IP.
- The SSP is also connected to the service users over the access network, and is connected to the other SSPs over the switching network. However, this is not of interest here, as the model is only concerned with the analysis of IN specific service processing.
- The SCP is connected to the SSP and the SDP over the signalling network.

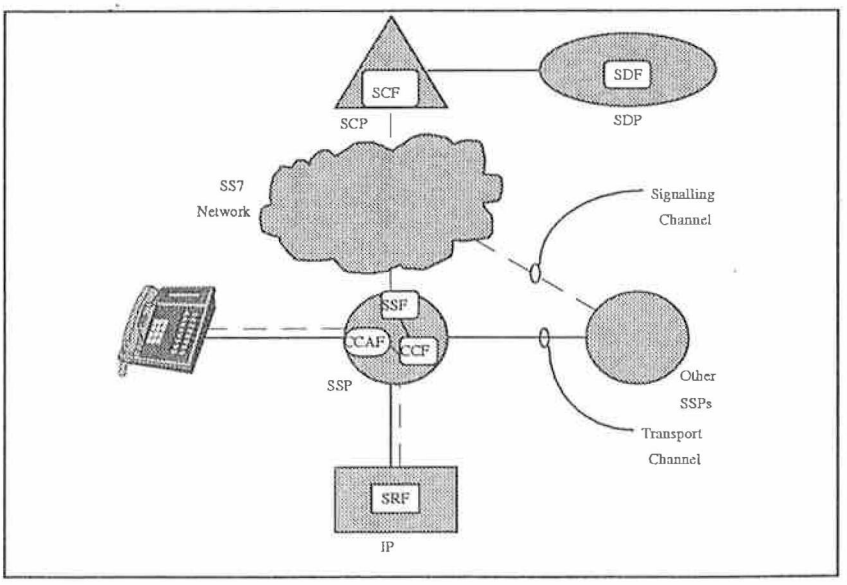

Figure 4-7 - Interconnection of Physical Entities in Intelligent Network.

4.4 Queuing Model of the IN

Now that a model of the IN resources has been defined, a queuing model of the network can be developed. Before doing so, it is important to identify the objective of the model and decide upon the parameters which are of interest in the model. In Chapter Two, it was shown that the main parameters of interest in an IN structured network are the mean PDD and the Impaired Call rate. In order to scope this study, it was decided to concentrate on the analysis of the PDD parameters and the mean response times at each node, which are required to calculate the PDD.

In defining the queuing model, each of the IN PEs are defined as a service resource for which incoming requests from subscribers and from other PEs contend. Each of the PEs can then be modelled as a queue (or set of queues), with the appropriate number of servers and service discipline. Thus, the IN network is modelled as a network of queues, in which the processing of incoming IN calls is modelled through the analysis of their transactions at each of the PEs at which they are processed. Before describing the queues in more detail, the following assumptions should be mentioned:

- It is assumed that there is sufficient capacity in the underlying switching and signalling networks, to be able to handle the load. This means that the model does not need to take into account the underlying PSTN and SS7 signalling networks.
- Each of the services consists of a set of interactions, with the different entities in the network. These interactions may have different transaction times at a given PE, depending on the service involved. However, it is assumed that all of the interactions at a given PE, for a given service, are homogeneous (that is to say, that they all have the same characteristics). This assumption is a simplification of what may happen in a real network, as different messages corresponding to the same service may have different transaction times in the real world. The assumption is made in order to simplify the model, otherwise, the model would require a memory of previous states of the system, thus complicating the model (due to the fact that it is not Markovian).

- The primary parameter of the model, which is of interest, is the mean PDD experienced by the service user. This delay is the time between successive interactions, between the service/network and the user. It is assumed that the delays experienced by the user, is the sum of the times spent in service processing at each of the PEs. It should be noted that this does not include the contribution to PDD, of processing which is not IN specific (e.g. SS7 processing).
- It is assumed that the service time at each of the IN PEs is constant. This means that the service time distribution is deterministic.

The queuing model follows that proposed by Leever et al [18], with the additional refinement of different types of requests. The different requests correspond to the different services, which are implemented on the network. The individual entities in the network are modelled as follows:

- *Service switching point.* The SSP, consists of a number of processors working in parallel. The SSP is modelled as G/D/1 queue, with requests of only one type, a limited joint buffer and the FCFS transaction discipline.
- *Intelligent Peripheral.* For the purposes of the services used in this model (and many of the CS1 services), the function of the IP is to play announcements to the user and collect service data from the service user. The IP is modelled as a G/D/1 queue with requests of only one type and the FIFO transaction discipline.
- *Service Control Point.* The SCP is modelled as a G/G/1 queue, with requests of different types and a processor sharing service discipline. This service discipline is based on the assumption that the SCP is running a multi-tasking operating system and has a process running for each service logic program type required for the service mix, supported by the SCP. The SLP processes are running concurrently with each other and consequently, the effective

processor capacity[12] is shared between them. This means that the SCP can serve requests of different service types simultaneously.

The effective capacity of the processor is denoted by μ and there are K types of service supported by the SCP. This means that there are K SLP processes and the i^{th} SLP process receives a share of the effective processor capacity $\mu_i = m_i \mu$, where $\sum_{i=1}^{K} m_i = 1$. This discipline is consistent with many SCP implementations and, in such cases, it is valid to assume that the operator would be able to choose the values of m_i and, in doing so, be able to allocate more SCP capacity to a particular service which may require it.

- *Service Data Point.* The SDP is modelled as a G/G/1 queue, with requests of different types and a FCFS service discipline, with appropriately chosen parameters.

4.5 Analytic Formulation of Queuing Model

The queuing model of an IN structured network, presented above, is a network queuing model. In order to develop an analytic formulation of the model, it is necessary to consider some related issues:

- The results which are required from the model (or in other words the characteristics of the network which are of interest). In this case, the main parameter of the model, which is of interest, is the mean PDD, as experienced by the service user. This delay is made up of the mean response times at each of the IN PEs, at which the IN call is handled and thus, the delays at each of these elements must be found, in order to find the overall delay.
- In addition, a further constraint on the formulation is that it has to be tractable and readily implemented as a computer program.

---

[12] The effective processor capacity is taken to be the full capacity of the processor, minus the capacity used in context switching between processes and that taken up by administrative programs running on the SCP.

These issues have a bearing on how the analytic formulation of the model is derived. Readers will recall from Chapter Three that the solution for Jackson and BCMP networks describes the steady state joint probability distribution of the station states in product form. This product form requires the calculation of the normalisation constant, which can be quite difficult in certain circumstances. Additionally, such network formulations require that there be only one class of request in the network. This is not the case in the IN queuing model and for these reasons it may be concluded, that the IN model is not suitable for formulation as either a Jackson or BCMP network.

Having identified that the IN queuing model does not fit the characteristics of a Jackson or BCMP network, approximation methods should be considered. In Chapter Three the decomposition method was identified as a useful technique, which provides a very good solution for networks which are almost product form and a good solution, even when many of the product form criteria are violated by the network. In addition, the formulation of the method allows the mean response time, at each station in the network, to be estimated, which in turn allows the response time at the station to be predicted. Each of these response time values is the estimate of the mean response time at the particular PE represented by the queue. Once these values are known, it is quite simple to estimate the delays experienced by users of each of the services, by adding the time spent at each of the PEs for the service processing. Thus, the decomposition method is particularly suitable for formulating the analytic model of the IN queuing network.

4.5.1 Formulation Of Decomposition Method

The Decomposition Method is so called as it partitions the network into a series of sub-networks, each of which can then be solved individually. In the case of the IN model, the partition of the network into sub-networks was performed, such that, each sub-network contains a single IN PE. This decision was made so that the partition would be as general as possible and would not impose or pre-empt any relationships between any particular PEs. This means that each of the PEs constitutes a sub network in its own right and the flows between it and each of the other sub-networks (i.e. the other PEs) are represented and taken into account when determining the waiting time for each station.

The following assumptions were made in formulating the model:

a) The following input parameters are assumed to be known a-priori in a network containing N nodes.

   i) The routing probability matrix P, where element p_{ij} represents the probability that a request at station i will pass on to station j on completion of service.

   ii) The SVC of service time $Ks_{ik}, \forall 1 \le i \le N, 1 \le k \le K$, for each request type k at station i.

   iii) The mean service rate μ_{ik}, for each request type k at each station i.

b) The different request types have the same service characteristics at each of the PEs, except the SCP. However, this complicates the model at the SCP unnecessarily. In order to avoid this, let us first revisit the service discipline at the SCP. The reader will remember that at the SCP, the processor capacity is divided among the SLPIs for each service. The service discipline, is such that the SLPIs run simultaneously taking some proportion of the overall SCP capacity. When a new request arrives at the SCP, it is put in the queue waiting for service by the appropriate SLP. This means that the SCP can be modelled using a queue for each SLP, which has a service rate equal to the service rate of the SLP. This reduces the complexity of the model, by avoiding the consideration of different classes of customers at some of the queues in the network - each queue in the network has only one class of requests. The routing of service requests to the appropriate SLP queue is dealt with in the model by the routing matrix. This implies that $Ks_{i1} = Ks_{i2} = Ks_{i3} = Ks_i, \forall 1 \le i \le N$ and $\mu_{i1} = \mu_{i2} = \mu_{i3} = \mu_i, \forall 1 \le i \le N$.

c) The interarrival times of requests into the network from the exterior are exponentially distributed, which means that the arrival process from the exterior is a Poisson process. This means that the SVC of interarrivals from the exterior equals unity (i.e. $Ka_i^o = 1, \forall 1 \le i \le N$). As these arrivals

correspond to incoming calls, then arrivals from the exterior only occur at the SSP.

The first step in the formulation is to determine the arrivals at each of the stations, using the equations $\lambda_i = \sum_{j=0}^{N} p_{ji}\lambda_j$. The next step is to find the SVC of the inter departures at each station j, which can be found using the following set of equations (see Chapter Three).

$$C_j = -1 + \rho_j^2\left(Ks_j + 1\right) + \left(1-\rho_j\right)\left(2\rho_j + 1 + Ka_j\right), 0 \le j \le N,$$

where:

- $Ka_j = \frac{1}{\lambda}\sum_{i=0}^{N}\left[\left(C_i - 1\right)p_{ij} + 1\right]\lambda_i p_{ij}$ is the SVC of the interarrival times at station j;
- C_j is the SVC of the inter departure times from station j;
- $\rho_j = {}^{\lambda_j}\!/_{\mu_j}$ is the utilisation of the station;
- λ_j is the mean arrival rate at the station;
- μ_j is the mean service rate at the station and
- p_{ij} is the probability that a request at station i will pass onto station j.

This describes a system of equations which can be solved, as follows, using matrix algebra to give the set of values for C_j at each station j.

$$C_j = -1 + \rho_j^2\left(Ks_j + 1\right) + \left(1-\rho_j\right)\left(2\rho_j + 1 + {}^{1}\!/_{\lambda_j}\sum_{i=0}^{N}\left[\left(C_i - 1\right)p_{ij} + 1\right]\lambda_i p_{ij}\right), 0 \le j \le N$$

$$= b_j + \sum_{i=0}^{N} C_i a_{ij}$$

$$\Rightarrow \overline{I}\,\overline{C} = \overline{\mathrm{B}} + \overline{\mathrm{A}}\,\overline{C}$$

$$\Rightarrow \left(\overline{I} - \overline{\mathrm{A}}\right)\overline{C} = \overline{\mathrm{B}} \Rightarrow \overline{C} = \left(\overline{I} - \overline{\mathrm{A}}\right)^{-1}\overline{\mathrm{B}}$$

where $b_j = -1 + \rho_j^2\left(Ks_j + 1\right) + \left(1-\rho_j\right)\left(2\rho_j + 1 + {}^{1}\!/_{\lambda_j}\sum_{i=0}^{N}\left(1 - p_{ij}\right)\lambda_i p_{ij}\right)$

and $\alpha_{ij} = \dfrac{\left(1-\rho_j\right)p_{ji}\lambda_j p_{ji}}{\lambda_i}$

Once these are known, the formulation can be completed using Kingman's formula and Little's Law (see Chapter Three) :

$$\overline{L_i} = \rho_i\left(1+\frac{\rho_i(Ka_i+Ks_i)}{2(1-\rho_i)}\right), \overline{R_i} = \overline{L_i}/\mu_i, \forall 1 \le i \le N$$

One of the assumptions, detailed above, is that the P matrix of transition probabilities between the stations are known a-priori. In fact, the elements of P are dependent on an analysis of the services implemented in the network and the configuration of the network. The general means of calculating the routing probabilities is $p_{ij} = \sum_{k=1}^{K} p_{ij}^k * \hat{p}_i^k, \forall 1 \le k \le K$.

The parameter p_{ij}^k is the probability of a request related to service k passing from station i to station j, given that it is already at station j. The parameter $\hat{p}_i^k$ is the probability of a request of service k being at station i. In order to explain the process of finding the elements of P, consider the following example.

4.5.2 A simple network example.

The network in this example, consists of a single SSP, a single SCP, a single SDP and a single IP (see Figure 4-8). It is assumed that the only service in the network is the Televote service, described earlier. The sequence of requests at the different PEs is User - SSP - SCP - SSP - IP - SSP - User - SSP - SCP - SDP - SCP - SSP - IP - SSP -SCP -SSP- User (see Figure 4-6).

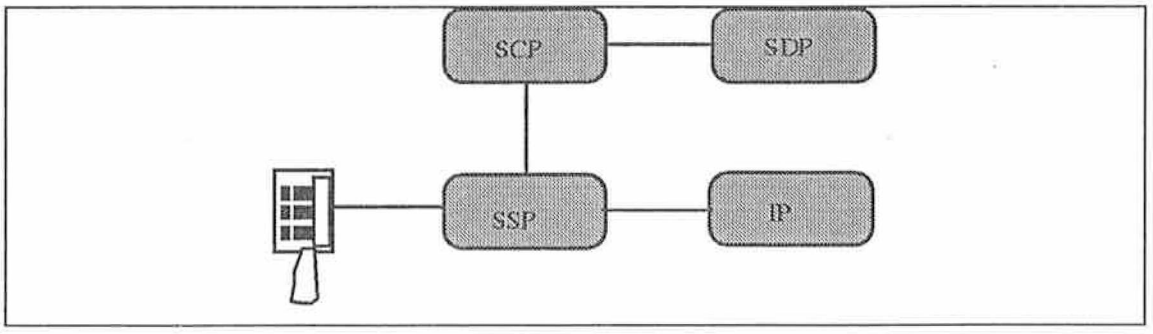

Figure 4-8 - Simple Scenario.

Before continuing, each station in the network is denoted by an index, as follows :

1. SSP;

2. SCP;
3. SDP and
4. IP.

As there is only one service in the network, the values of the parameter $\hat{p}_i^k$ are all unity. This means that it is only the values of p_{ij}^k, which need to be calculated, in order to determine p_{ij}. As the value of p_{ij}^k represents the probability that a request of service k, at station i, will pass onto station j, then it can be seen that the rows of the matrix P, must sum to unity (i.e. $\sum_{i=1}^{N} p_{ij} = 1$). This can be justified simply by the fact that a request departing from station i must go to another station. The one exception to this rule is at the SSP, where requests pass out of the IN network and not onto another station in the network. This rule allows us to develop a system of equations, which are described by $\sum_{i}^{N} p_{ij} = 1$ and represent a system of four equations and sixteen variables, which cannot be solved alone. Luckily, by analysis of the network configuration, the values of many of these variables can be set.

Often, it will be easier to examine the interactions in the service. Taking the departures from the SSP, it can be seen that there are a total of six departures from the SSP to the other PEs (SCP and IP) and to the exterior. Of these departures, three departures are to the SCP, $\Rightarrow p_{12} = 3/6$, two departures are to the IP, $\Rightarrow p_{14} = 2/6$, and a single departure is to the exterior, $\Rightarrow p_{10} = 1/6$.

At the SCP, there are four departures - three departures to the SSP, $\Rightarrow p_{21} = 3/4$, and a single departure to the SDP, $\Rightarrow p_{23} = 1/4$. At the SDP there is only one departure which is back to the SCP $\Rightarrow p_{32} = 1$. Finally, at the IP, there are two departures, both of which are to the SSP $\Rightarrow p_{41} = 1$. As of all of the departures have been considered, all of the other elements of the routing matrix are zero. Thus, in summary:

$$P = \begin{bmatrix} 0 & 3/6 & 0 & 2/6 \\ 3/4 & 0 & 1/4 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The values of all of the model parameters are now known and the mean delays at each of the PEs can be found, as detailed above. Once these are known, then the delays experienced by the service user can be estimated. Continuing the Televote example, there are three user interactions in the service:

1. User-SSP-SCP-SSP-IP-User. This contains two SSP transactions and one transaction at the SSP and SCP. Thus, from the formulation, the mean delay for this interaction can be found as $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \overline{R_{scp}} + \overline{R_{ip}}$.
2. User-IP-SSP-SCP-SDP-SCP-SSP-IP-User. This contains two SSP transactions, two SCP transactions, one SDP transaction and one IP transaction. Thus, $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \left(2 \times \overline{R_{scp}}\right) + \overline{R_{sdp}} + \overline{R_{ip}}$.
3. User-IP-SSP-SCP-SSP-User. This contains two SCP transactions and one SCP transaction, giving $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \overline{R_{scp}}$.

4.6 Simulation of Model

In order to validate the analytic approximation of the IN queuing model, presented above, it is necessary to simulate the queuing model. By comparing the results of the simulation with those obtained from the analytic approximation, it is possible to gauge how accurate the approximation is. In particular, it can be determined if those assumptions made during the approximation are valid. To enable this comparison, a simulation of the queuing model was performed using the OPNET simulation package. The following sections describe OPNET and the simulation model which was developed upon it.

4.6.1 OPNET

OPNET (Optimised Network Engineering Tools) is a graphical computer aided design package, which facilitates the simulation of communication and computer networks.

OPNET is a discrete event simulation environment, which means that the simulation models events occurring in discrete time periods, rather then modelling time continuously. This means that the simulation skips time periods in which no events occur, to the instant in time at which the next event occurs. In doing so, OPNET allows the simulation to run faster than actual time. In practice, this means that the simulation is interrupt driven - events are indicated by interrupts occurring. Such an interrupt could indicate a packet arriving at a node or the end of a service request.

OPNET is particularly suitable to the modelling of communications networks, as it allows the simulated system to be broken into a series of nodes, which communicate with each other through the exchange of packets of information (i.e. it is analogous to a packet switched network). OPNET also provides a library of predefined models, representing resources such as Ethernet networks, radio and satellite antennae and a set of probability distributions, which can be used to model traffic and service characteristics. OPNET simulations are developed by placing and interconnecting a set of graphical icons and setting their parameters using a menu. OPNET simulation models are hierarchical, where different aspects of the model are represented at the appropriate level (or domain) of the hierarchy. The different domains are as follows.

4.6.1.1 The Network Domain

The *Network Domain* is where the different nodes in the network and the links between them are shown. The network domain can also contain sub-networks, which are themselves networks, and can contain further sub-networks, which parallels the situation in reality. Maps can also be loaded into the network model to provide a geographical reference to the topology.

4.6.1.2 The Node Domain

The *Node Domain* defines each of the nodes in the network model in terms of the modules which they contain. These modules are one of a set of predefined types of modules, which include:

1. A processor module, which executes a particular process model.
2. A queue module, which is a buffer for packets, each of which are served according to a particular process model. The queue can also contain distinct sub-queues, each of which are managed by the process model.
3. Packet generators, which generate the associated packet types according to the associated probability distribution (e.g. Poisson) and send them to the module to which they are connected.
4. Point-to-point transmitters, which allow the node to transmit packets to another node to which it is connected in the network model.
5. Point-to-point receivers, which allow the node to receive packets from another node to which it is connected in the network model.

4.6.1.3 Process Domain.

In the Process Domain, the behaviour of node modules are defined in terms of a process model. A process model consists of a *Finite State Machine* (FSM) which is graphically represented by a *State Transition Diagram* (STD) (see Figure 4-9). Each STD consists of a set of states and the transitions which exist between them. Each of the states has a set of 'enter executives' and 'exit executives'. The enter and exit executives are fragments of code, which are written in a language called Proto-C and are executed when the process passes into, or out of, the state respectively. Proto-C is an extension of the C language defined for OPNET and contains a set of libraries, which include functions for the handling and distribution of OPNET packets, manipulation of probability distributions, manipulation of queues and gathering of simulation statistics. The transitions between states in the process model are conditional, which means that there may be several transitions out of the same state. Which of the transitions are taken at a particular time is dependent on which of the transition conditions are met. Naturally, the preconditions for all of the transitions out of a particular state must be mutually exclusive (i.e. only one condition must evaluate to true at any instant), otherwise, the process will not be able to decide which transition to make. A state may also have a transition back into itself.

A state may be either an unforced or a forced state (a forced state is marked black in the diagram, while an unforced state is marked white). When the process enters an unforced state, it executes the enter executives and then blocks, waiting for a further interrupt to occur. When such an interrupt occurs, it executes the exit executives and makes the appropriate transition out of the state. If a state is forced, then the process does not block while in the state, but instead, the enter and then the exit executives are sequentially executed, before the process makes the appropriate transition out of the state. Each process model must contain at least one unforced state, which allows the simulation engine to perform other tasks while the process is blocked. The process can pass through several forced states, before coming to rest at an unforced state. Additionally, a process model can have children process models, each of which has its own STD. The parent process can, at any time, pass control to a child process which assumes control until it passes into an unforced state.

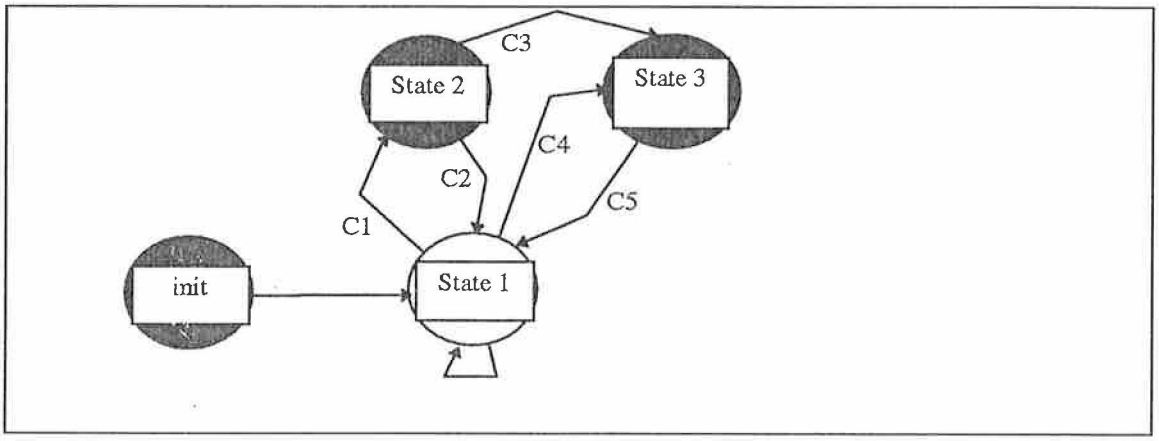

Figure 4-9 - Example process model.

A simulation model is defined by defining the appropriate elements of the model in each of the domains (see Figure 4-10). A particular simulation has a single network model which contains a set of nodes (some of which may be of a different type). Each node has a set of modules, and each module (e.g. a queue or processor module) has a process model associated with it. Each of these process models can in turn have a set of child processes, which are invoked by it.

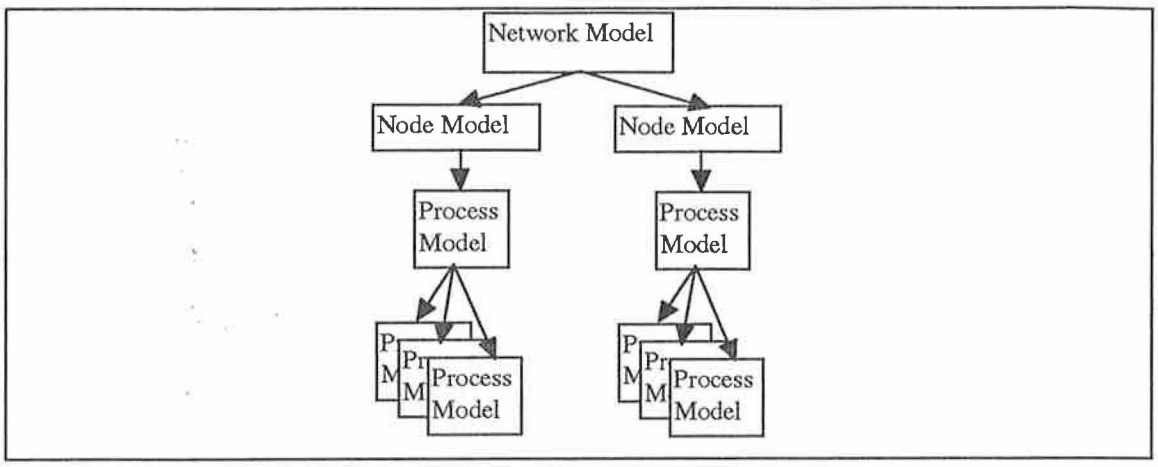

Figure 4-10 - Hierarchy of OPNET simulation model.

OPNET also has a set of support tools, which are:

- The parameter editor, which enables the definition of packet types and formats, as well as their default values.
- The probe tool, which allows the attachment of probes to particular resources in the simulation. The probes gather statistics about the resource during the course of the simulation. For example, a probe can be attached to a queue module and can monitor the queue length during the simulation. These features are in addition to the statistics gathering functions, which are supplied with Proto-C and which can be incorporated into a process model.
- The analysis tool, which allows the analysis of the results of a simulation and enables such results to be displayed graphically.
- The filter tool, which allows the post-processing of simulation results and provides a graphical means of specifying mathematical functions to be performed on the data, through use of block diagrams, where each of the blocks represents a mathematical function to be performed on the data (e.g. an integrator).
- The simulation tool, which allows the graphical definition of several simulation runs which are to be executed consecutively. These consist of the simulation name and the values of the simulation parameters for that run. This provides a

simpler interface for running the simulation rather than specifying the many parameters on the UNIX command line.

4.6.2 Simulation Model

The simulation model is split into a three layer hierarchy, corresponding to the division in OPNET. These layers are :

- The process models. Each IN component contains functionality required for the operation of services. The process models represent this functionality.
- The node models. Each IN PE is represented as a node in the node model. Each node contains the necessary modules and their associated process models.
- The network model, which interconnects the nodes, to form the network.

The remainder of this chapter describes the OPNET packets used in the simulation, the node and process models for each of the IN PEs, a sample network scenario and the statistics, which can be gathered by the simulation.

4.6.3 Message Types in simulation

4.6.3.1 Message Format

OPNET simulates a network by mimicking the passage of messages around the network (see section **Error! Reference source not found.**). The format of these messages is defined in the OPNET parameter editor and the particular format to be used by a traffic generator, is specified as a parameter of that traffic generator.

In the simulation model, a single packet format was defined for use in all of the services. These packets are then exchanged between the different IN physical entities as per the

interaction diagrams in section 4.2. The values of several fields of a message instance are dependent on the service type, to which the message instance is associated and the context of that service instance. Other fields of the message are used to carry simulation results (e.g. the delay incurred) or for general housekeeping and are independent of the service type involved. The fields of the generic message, their meanings are described in Table 4-1, while the values of the service dependent fields are described below, for each service.

Field Name	Service Specific Value	Meaning
Delay	No	The total delay incurred by the request since the last user interaction.
Service	Yes	The type of service to which the request is associated.
Service Mode	Yes	The mode of service for the service instance (e.g. set up Abdial numbers or dial Abdial number).
Message Type	Yes	The type of message indicating the nature of the current request.
Port Number	No	The port address, from which the request entered the physical entity. Used to maintain record of where request came from.
SSP Address	No	The address of the SSP, at which the request entered the network. This is set on leaving the SSP.
SCP Address	No	The address of the SCP, used for processing of the service. This is set on leaving the SCP.

Table 4-1 - Field Types and Values for Simulation Messages.

4.6.3.2 Abbreviated Dialling messages

In the abbreviated dialling service, there are two different service modes :

1. Abdial set-up (where the Abdial numbers are stored) and
2. Abdial usage, where the service user dials an Abdial number to initiate a call.

The values of the message type field are based on the different interactions between the PEs in the interaction diagram for the Abdial service. The possible values of the three service specific fields are shown in Table 4-2.

Field	Values
Service	ABDIAL
Service Mode	SETUP
	USAGE
Message Type	ADDRESS_ANALYSED
	UPDATE_DATA
	UPDATE_CONFIRM
	CLEAR_CALL
	QUERY
	QUERY_RESULT
	PROCEED_NEW_DATA

Table 4-2 - Set of Service Specific Field Values for Abdial Service

4.6.3.3 Call Forwarding messages

In the Call Forwarding service, there are also two different modes of the service:

1. Forward set-up, where the service user invokes the service to set up the diversion of all calls from a directory number A to another number B or alternatively, deactivates the diversion.
2. Forward usage, where calls whose destination is A are routed to B.

The values of the message type field are based on the interactions which occur between the PEs for Call Forwarding (see Figure 4-4). The possible values for the service specific fields are shown in Table 4-3.

Field	Values
Service	CALL_FORWARD
Service Mode	FWD_SETUP
	FWD_USAGE
Message Type	ADDRESS_ANALYSED
	UPDATE_DATA
	UPDATE_CONFIRM
	QUERY
	QUERY_RESULT
	PROCEED_NEW_DATA
	CLEAR_CALL

Table 4-3 - Service Specific Field Values for Call Forwarding Service.

4.6.3.4 Televote messages

In the Televote service, there is only one mode of service and the values of the message type field are based upon the interactions between the physical entities for the service. The possible values of the service specific fields are shown in Table 4-4.

Field	Values
Service	TELEVOTE
Service Mode	USAGE
Message Type	ADDRESS_ANALYSED
	PROMPT_AND_COLLECT
	COLL_UI
	UPDATE_DATA
	UPDATE_CONFIRM
	PLAY_ANNOUNCEMENT
	END_ANNOUNCEMENT
	CLEAR_CALL

Table 4-4 - Service Specific Field Values for Televote Service.

4.6.4 SDP Model

4.6.4.1 Node Model

The SDP node model consists of a processor component, a set of input ports and a set of output ports (see Figure 4-11). The processor performs the functions of the SDP, (i.e. the storage and retrieval of service data) while the ports allow the SDP to communicate with other PEs. As an SDP is only (for the purposes of this discussion) required to be connected to a SCP, then these ports are used for communication with SCPs. This is not the way in which the SDP would be structured in practice. In reality, it is likely that the SDP would exist on a network (such as the SS7 signalling network or a TCP/IP network), which interconnects not only the SCP, but other IN (e.g. the SMS) and probably non-IN resources. However, it is only the interactions between the SCP and the SDP, which are of interest here.

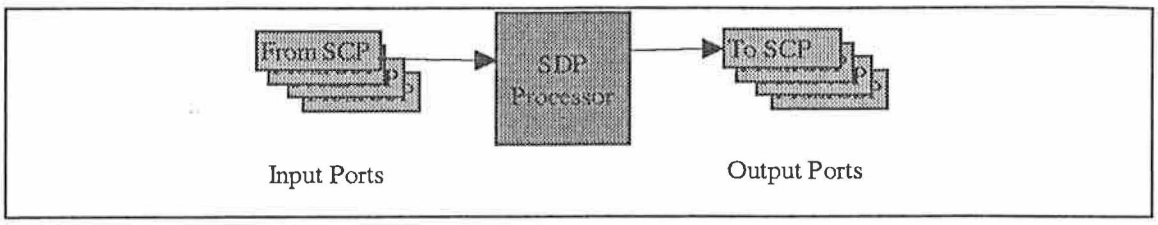

Figure 4-11 - Node model for SDP.

In the node model, it is assumed for convenience, that each port relates to a dedicated SCP and that the same input and output port addresses refer to the same SCP. For example, input port 1 receives traffic from SCP 1, while output port 1 sends traffic from the SDP to SCP 1. In the model implemented, there are four input ports and four output ports, but this number may easily be increased.

4.6.4.2 SDP Process Model

The STDs for the SDP, IP and SSP are the same, containing just two states, but in each case, the logic associated with the state is different. For this reason, the meaning of the states will only be discussed for the SDP.

The SDP process model consists of two states (see Figure 4-12):

1. The '*Init*' state is the state in which the process starts and sets up the initial state of the process, before passing to the idle state. The init state is a forced state, which means that once the process is created, the init executives are executed before the process makes a transition into the serve requests state.
2. The '*Serve Requests*' state is the state in which the process queues and serves arriving requests. Once in the 'serve requests' state, the process loops in the state, constantly serving and queuing requests. Once an event occurs, the process makes a transition out of the 'serve requests' state and back into the same state. The 'serve requests' state is an unforced state, which means that when the process makes a transition into the state, the enter executives are run and the process blocks, while waiting for the next event. Such an event will be one of two types:

i) A new request arriving at the SDP, which will be put in the queue if the SDP processor is busy, or will be served if the processor is not busy.

ii) Completion of service for a request. The request is sent back to the relevant SCP and the next request in the queue (if there is one) is served.

The exit executives of the 'serve requests' state implement the behaviour of the SDP, while the enter executives are empty, allowing the process to block directly on entering the state. The sequence of events on the creation of the process is as follows:

i) The process passes through the init state, initialising the process and then into the serve requests state, where it blocks waiting for an event interrupt. In this case, the event interrupt will signal the arrival of a new request at the SDP. As no requests have yet been served, then the interrupt cannot indicate the termination of a previous request's service.

ii) An interrupt occurs, signalling the arrival of a new request and the process begins the transition out of the 'serve requests' state. Before it makes the transition, it executes the exit executives, which perform the service of the request. The exit executives signal a new interrupt to occur at the time the request finishes service. It marks the server as being busy and notifies OPNET to send the request back to the SCP at the end of the service time. On completion of the exit executives, the process transitions back to the 'serve requests' state and blocks, waiting for a new event.

iii) A new interrupt occurs and this time it may be either of the two interrupts. If it is a new arrival, it will be placed in the queue until the server is no longer busy, while if it indicates the end of the

previous requests service, the server will be marked as no longer busy and if there are any requests in the queue they will be served.

At the SDP, there are two types of service request which may occur. These are (see the transition diagrams in section 4.2):

i) An *update* request, which requests the SDP to update some service information. This is indicated in the request by the packets Message Type field having the value UPDATE_DATA. In this case, the SDP serves the request according to the Update distribution[13] before it changes the Message Type field to have the value UPDATE_CONFIRM. This confirms that the request was performed.

ii) A *query* request, which requests the SDP to provide it with some service data. This is indicated in the request by the packet having its Message Type field having the value QUERY. This request is served according to the query distribution and its Message Type is set to QUERY_RESULT, before it is returned to the SCP.

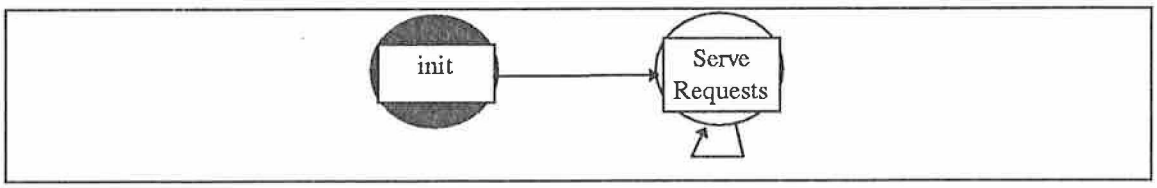

Figure 4-12 - Process Model for SDP.

On consideration of the SDP process model, it can be seen that it models a G/G/1 queue, providing, as it does, a general service time distribution and a single server operating a

[13] It is assumed that the service time distribution for an Update may be different to that for a Query, although for a particular simulation they can be set to be the same. This allows the simulation to model scenarios in which these distributions are similar or are different.

FIFO service discipline. When the service time distribution is set to constant[14], it can be seen that the process models a G/D/1 queue.

4.6.4.3 SDP Parameters.

The following parameters are specified in the SDP node model and are utilised by the process model:

- The mean time taken for an update request;
- The mean time taken for a query;
- The type of service time distribution for an update request. This distribution is the constant distribution by default; and
- The type of service time distribution for a query request. This distribution is also the constant distribution by default.

4.6.5 SSP Model

4.6.5.1 SSP Node Model

The node model for the SSP consists of a processor component, a set of output and input ports and a set of traffic generators (see Figure 4-13). The processor performs all of the functions associated with the SSP, which are implemented in the SSP process model. The traffic generators generate the incoming subscriber traffic to the SSP. They are intended to model the interaction of a service user with the relevant service. There is at least one generator for each service type and where a service may involve different interactions with the service user, there may be a generator for each type of interaction (see Figure 4-2) (e.g. call forwarding set-up and the operation of call forwarding).

The input and output ports allow the SSP to communicate with other entities in the network. In the case of the SSP, these are the IP and the SCP (as the subscribers are

---

[14] The constant distribution is the same as the deterministic distribution - the variance of the distribution is zero.

modelled in the SSP itself). In reality, the SSP is connected to a variety of different resources via the signalling network, but, as it is only the IN related interactions which are of interest, these other resources may be ignored. As in the SDP, it is assumed that the same input port number and output port number correspond to the same external resource. In the initial model, there are only four sets of ports, allowing two SCPs and IPs to be associated with any SSP. This is easily extended by increasing the number of port pairs and the distinction is made between SCPs and IPs by letting all even port numbers indicate an SCP, while an odd port number identifies an IP. It will always be assumed that port set 0 indicates the default SCP, while port set 1 indicates the default IP.

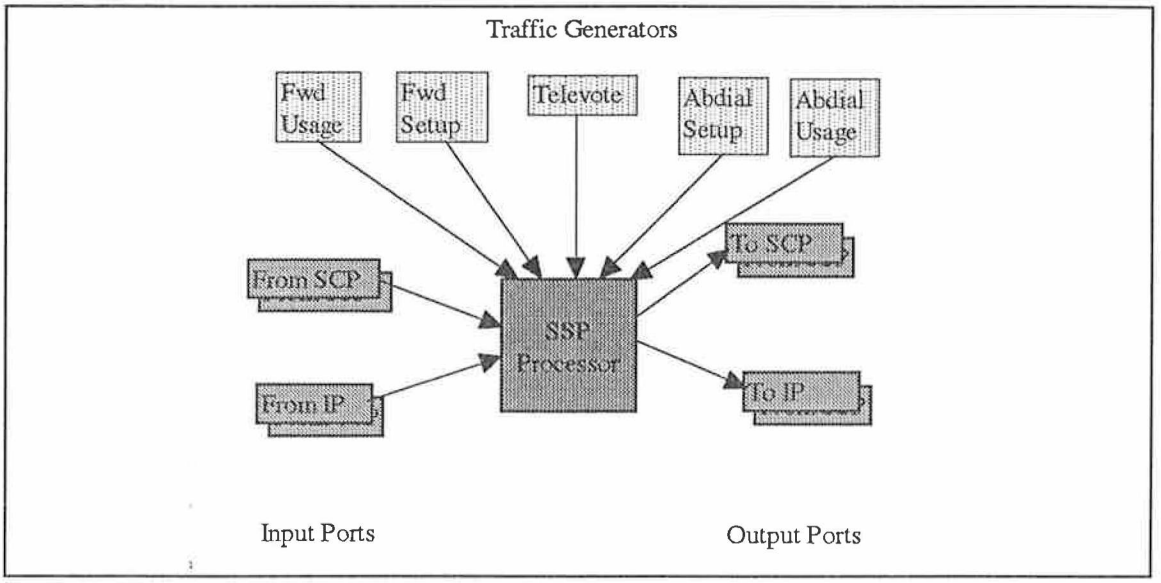

Figure 4-13 - SSP Node Model

4.6.5.2 SSP Process Model

The SSP process model is a finite state machine consisting of two states (see Figure 4-12). The state transition diagram is the same as that of the SDP, but the actions taken in the state executives are different. In the simulation, the SSP receives several different types of messages, each relating to an interaction in the service interaction diagrams shown in section 4.2 and, depending on the message, the SSP will take a specified action. The list of messages and actions for the SSP is shown in Table 4-5.

As in the SDP, the process model behaves as a G/G/1 queue, where the arrival characteristics are decided by the generating process (in the case of the SSP these are Poisson (user call initiation) and general (IP and SCP originating traffic).

Message	Context	Action Taken by SSP
ADDRESS_ANALYSED	This message arriving at an SSP indicates an arrival of a new call to the SSP.	The SSP forwards the message onto the SCP for analysis.
PROCEED_NEW_DATA	This message is received from the SCP during the Forward or Abdial usage services and indicates to the SSP that it should continue call processing with the new data accompanying the message.	This message indicates the end of the IN specific aspects of both the Forward and Abdial usage services. Thus, the SSP calculates the delays experienced by the request and destroys the OPNET packet.
CLEAR_CALL	This message is received from the SCP during any service and indicates that a call should be cleared	The SSP calculates the SSP related statistics, before deleting the OPNET packet.
TERMINATION_AUTHORISED	This message is encountered as a trigger is fired in the call process, during the Call Forward service. It indicates that the network is attempting to connect a caller to the DN.	The SSP forwards the message to the SCP.
PLAY_ANNOUNCEMENT	Received from the SCP during the Televote service and indicates that the IP should play an announcement to the user.	The SSP forwards the message to the IP.
END_ANNOUNCE	Received from the IP during the Televote service, and indicates that the IP has ended an announcement.	The SSP forwards the message to the SCP.
COLL_UI	Received from the IP during the Televote service, after input has been collected from the user.	The SSP forwards the data to the SCP.
PROMPT_AND_COLLECT	Received from the SCP during the Televote service, when the SCP requires the user input.	After processing of the request, the SSP forwards it to the IP.

Table 4-5 - Message Types and Actions in the SSP

4.6.5.3 SSP Parameters.

The following parameters are specified in the SSP node model and are used by the SSP process model:

- The mean transaction time for requests;

- The distribution of the transaction time, which is constant by default;
- The interarrival rate of calls for each service type;
- The interarrival distribution for calls for each service type; and
- The packet format for calls for each service type;

4.6.6 IP Model

4.6.6.1 IP Node Model

The node model of the IP contains a module component, input ports and output ports (see Figure 4-14). The processor component performs the functions of the IP, as defined in the IP process model. The input and output ports allow the IP to communicate with the outside world. These ports are used to communicate with the SSP, and there is a pair of input and output ports assigned to each of the SSPs with which the IP communicates. In reality, there is also communication between the SCP and the IP. This communication takes place once the SSP has set up a connection between the subscriber and the IPs service resource. The message from the SCP informs the IP of the nature of the interaction which is to take place. However, as the SSP will already have initiated the link between the subscriber and the service resource, it does not affect the flow of the simulation or the sequence of events if the SCP-IP interactions are ignored. In the initial model, there is a single pair of input and output ports, which are used to communicate with the SSP to which the IP is associated. Once more, this can be extended to take into account communication with several SSPs, by merely increasing the number of port pairs.

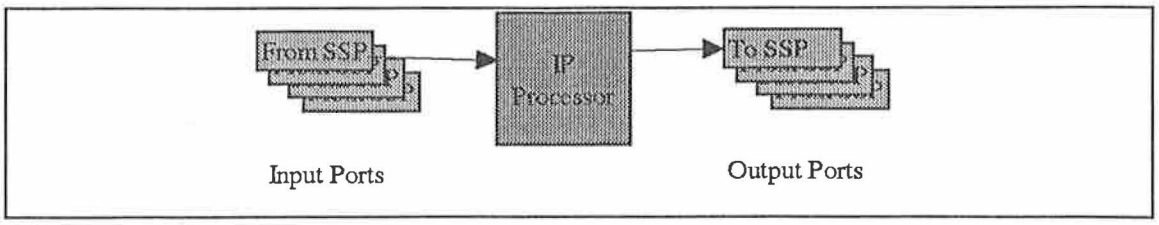

Figure 4-14 - IP Node Model

4.6.6.2 IP Process Model.

The IP process model also has the same STD as the SDP, but the state executives are different. All requests received by the IP are returned to the SSP, after being served. There are two types of messages which are received by the IP:

1. 'Prompt_and_Collect'. Indicates an announcement should be played to the user and input should be collected from the user. The IP then sends the 'Coll_UI' message back to the SSP.
2. 'Play_Announcement'. Indicates that an announcement should be played to the service user. Once it is finished, the IP sends an 'End_announce' message to the SSP.

4.6.6.3 IP Parameters

The parameters of the IP node model are:

- The mean holding time of a service resource;
- The distribution of the mean holding time of the service resource; and
- The number of service resources which are in the IP.

4.6.7 SCP Model

4.6.7.1 SCP Node Model

The SCP node model contains a central queue module and a set of input and output ports (see Figure 4-15). The queue module implements all of the service logic and performs the functions associated with a SCP. The input and output ports connect the SCP to the SSPs and SDPs, with which it is associated. It is assumed that the even port addresses specify an SDP port, while odd port addresses indicate SSP ports.

In reality, the SCP would also communicate with the IP, but as explained earlier (see section 4.6.6.1), for the purposes of the simulation, the SCP-IP interactions can be done via the SSP, without any loss of generality[15]. There is also the possibility of inter-SCP communication for load balancing purposes, but inter-SCP communication is not considered in this simulation model.

The SCP in the IN queuing model, is assumed to have a certain capacity allocated to each service, which it implements. This is to model the fact that the service logic programs may be running, even when they are not actually serving requests. Thus, in order to simulate this, the SCP processor contains a separate queue for each service (e.g. in the simulation there are three services considered and thus the processor has three queues). The method in which the processor serves the queues (i.e. the service discipline) is implemented in the SCP process model and will be explained there (see section 4.6.7.2).

4.6.7.2 SCP Process Model

One of the main differences between the SCP process model and the other process models is that the SCP process model creates child processes. In this case, we call the main SCP process model the root (or parent) process. The child processes of the root process correspond to the SLPIs in the SCP. Each of the SLPI processes has its own

---

[15] This assumption is based on the fact that the performance of the signalling network is satisfactory. If the signalling network is overloaded, then the SCP-IP communication would have an effect on the delay. However, SS7 issues are not considered in this model.

process model, which implements the service logic. Before going on to discuss these SLPI processes in detail, the root process model must first be described.

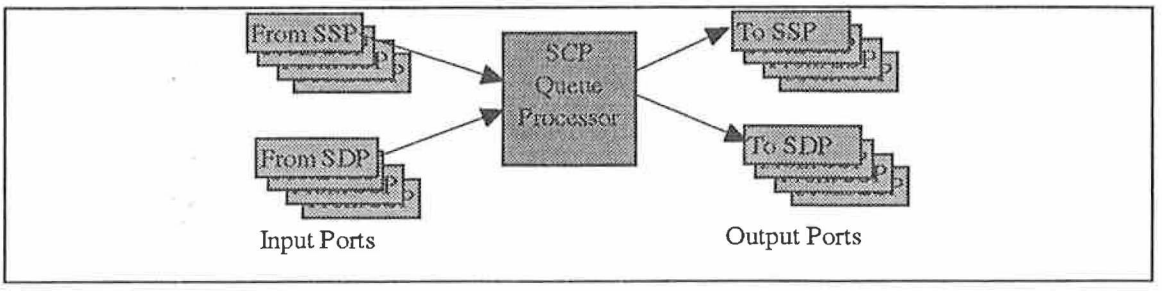

Figure 4-15 - Node Model for SCP.

4.6.7.2.1 SCP Root Process

The SCP root process STD is (as mentioned above) the same as that for the SDP and SSP (see Figure 4-16). However, the functionality, which is implemented in the process, is different. The root process acts as a dispatcher for requests. The splitting of the service logic and the SCP infrastructure into separate processes simplifies the process models involved. This is because building all of the service and SCP logic into a single process model would result in a large, complex, monolithic process model, making it more difficult to add new services. If each SLP is a separate process, then it is easier to add a new service - the new SLP process model is defined, and some small code is added to the SCP in order to allow it to recognise the new service and invoke the SLPI. Also, the number of sub-queues in the SCP node model is increased. When each new request arrives at the SCP, it is either served by the appropriate SLPI or is placed in the relevant queue to await service by the SLPI.

In real systems, there would often be one single buffer for the incoming requests and the appropriate requests would be removed from the buffer. However, the use in the simulation of three separate sub-queues for the request types does not introduce any loss of generality, as when taken together, they constitute one single buffer and the fact that they are separated into sub-queues, according to their service types, means that the service discipline is easier to implement. The STD for the root process is shown in Figure 4-16. The two states are:

- The Init state, is the state in which the process model is started and while in the init state, the state variables of the process model are initialised. Then a transition is made into the 'Dispatch Requests' state. In the init state, the processes for the SLPIs are created and invoked so that they may initialise. During this process, each of the SLPIs are marked as being busy, so that incoming requests are queued until they are ready for them.
- The Dispatch Requests state, takes incoming requests and if the required SLPI process is not busy, it passes the request onto that process and marks the SLPI as being busy. If the SLPI process is busy, then the request is placed in the appropriate service queue awaiting service. The Dispatch Requests state also takes notifications of the completion of a request's service. If a request of the type associated with the SLPI is pending service, it is sent to the SLPI process. Otherwise, the SLPI process is marked as being idle and waiting for a new request. The enter executives for the dispatch state are empty and the dispatching functions are written in the exit executives. It should be noted that when a SLPI is invoked by the root process to serve a request, the root process is blocked until the SLPI process makes a transition into a unforced state and becomes blocked. However, as will be seen below, the SLPI process moves into an unforced state very quickly and the dispatcher process is only blocked for a negligible period of time.

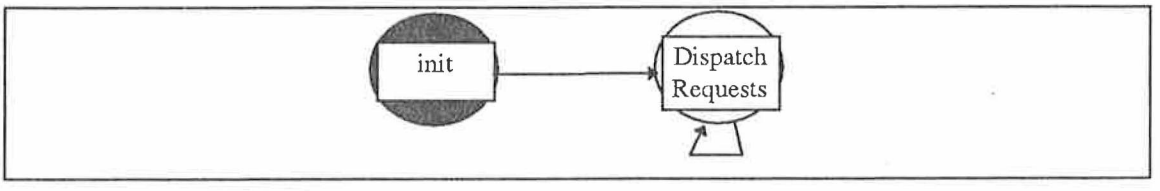

Figure 4-16 - Process Model for SCP root process.

4.6.7.2.2 Service Discipline in the SCP.

In the simulation model, the service discipline is implemented in a very simple manner. The SCP root process keeps track of which SLPIs are busy serving requests, at each instant in time. As new requests arrive or existing requests complete service, the next pending request (if there is one) is sent on to the SLPI. Within the SLPI, the percentage of

overall processor capacity, which the SLPI is receiving is known (it is a parameter of the SCP node model). The SLPI finds the service time for a particular request, by generating a number from the service time distribution. This service time is the length of time the service would take if the full processor capacity were allocated to the service. Thus, in order to find the actual service time which will occur, it is necessary to scale this value by the capacity allocated to the service as follows: $AST = NST * {}^{100}/_{capacity}$, where AST is the actual service time, NST the normalised service time (i.e. the service time for the service request if 100% of the capacity was available) and capacity is the percentage of processor capacity which is allocated to the service.

4.6.7.2.3 Call Forward Process

The STD for the call forward process contains two states (see Figure 4-17) and it is the same as that of the Televote and Abdial SLPs. The process models for these service are not discussed in any detail, as their states are exactly the same as the Call Forward model. The service logic, however, is different and this is discussed under each service. The states are:

- The *Init* state, is the state in which the process model is started and while in the init state, the state variables of the process model are initialised. Then a transition is made into the serve requests state. Before it makes a transition to the serve requests state, the process also sends an interrupt to the SCP root process to inform it that the SLPI is now ready to serve requests.
- The '*Serve Requests*' state takes the incoming requests sent by the root process and serves them. It determines what the service time should be, according to the service time distribution of the service at the SLPI and schedules an interrupt, for that time, to the root process indicating the end of service. It also modifies the Message Type field for the request, before sending it onto the SDP or SSP. The different value changes for the Message Type field and the destination of the packet after the SCP is shown in Table 4-6, while the reader is referred the transition diagram for call forwarding service in section 4.2.

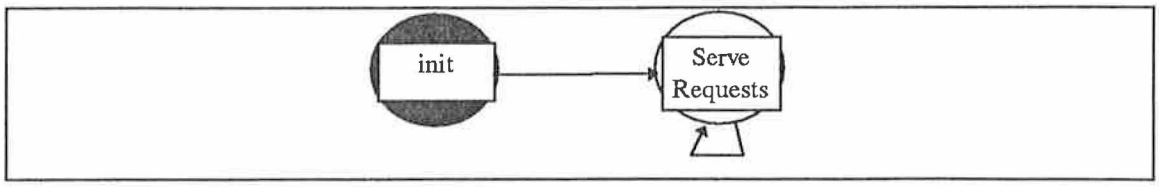

Figure 4-17 - Process Model for Call forward SLP.

Value for incoming request	Value for outgoing request.	Destination
ADDRESS_ANALYSED	UPDATE_DATA	SDP
TERMINATION_AUTHORISED	QUERY	SDP
UPDATE_CONFIRM	CLEAR_CALL	SSP
QUERY_RESULT	PROCEED_NEW_DATA	SSP

Table 4-6 - Value Changes for Message Type Field of Call Forwarding Request at SCP.

4.6.7.2.4 Televote Process

The STD for the Televote SLP process is the same as the STD for the Call Forwarding SLP process, containing the same two states. (see Figure 4-17). The message changes and destinations of outgoing requests are shown in Table 4-7.

Value for incoming request	Value for outgoing request.	Destination
ADDRESS_ANALYSED	PROMPT_AND_COLLECT	SSP
COLL_UI	UPDATE_DATA	SDP
UPDATE_CONFIRM	PLAY_ANNOUNCEMENT	SSP
END_ANNOUNCEMENT	CLEAR_CALL	SSP

Table 4-7 - Value Changes for Message Type Field for Televote Request at SCP.

4.6.7.2.5 Abdial Process

The STD for the Abdial SLP process is the same as the STD for the Call Forwarding SLP process, containing the same two states (see Figure 4-17). The message changes and destinations of outgoing requests are shown in Table 4-8.

Value for incoming request	Value for outgoing request.	Destination
ADDRESS_ANALYSED	UPDATE_CONFIRM (if the Service Mode is ABDIAL_SETUP)	SDP
ADDRESS_ANALYSED	QUERY (if the Service Mode is ABDIAL_USAGE)	SDP
UPDATE_CONFIRM	CLEAR_CALL	SSP
QUERY_RESULT	PROCEED_NEW_DATA	SSP

Table 4-8 - Value Changes for Message Type Field of Abdial Request at SCP.

4.6.7.3 SCP Parameters

The parameters of the SCP model are:

- The percentage of the SCP processor capacity which is allocated to each service;
- The mean transaction times for each type of service at the SCP; and
- The distribution type of the transaction time for each service at the SCP.

4.6.8 Statistics Measurement

The primary purpose of this simulation is to determine the mean delay experienced by an IN service user under different conditions and, in doing so, determine the accuracy of the analytic approximation, developed earlier. The simulation implementation includes code which calculates the mean delay accrued by a service instance at each PE and the overall delay experienced by the service user. Additionally, the simulation gathers the interarrival times of requests at each PE. OPNET provides support for the gathering of other statistics from the simulation, but in order to scope the size of this report, these are not discussed here.

Chapter 5

5. Analysis of Results

In the previous chapter, both the simulation and the approximate analytic formulation of the queuing model were presented. In order to determine whether the analytic approximation provides good results, the results must be compared with those from a corresponding simulation.

To simplify the description, several different scenarios are presented, starting with the least complex solution and progressing towards the most complex. The least complex solution is one in which there is only a minimum of Network Elements (NEs) - one of each type of NE (i.e. only one SSP, SCP, SDP, IP) - and only one service in the network. This can then be made more complex, by introducing more services into the network and by increasing the number of NEs in the network.

As a means of scoping this chapter, the networks presented here are all networks in which there are a minimum number of PEs (i.e. one of each). However, results from the analytic solution and from simulations were gathered for larger networks and the results which are presented here are also representative of the results gathered for larger networks.

5.1 Results

5.1.1 Single Service Network

In this scenario, there is one of each type of IN PE in the network (see Figure 5-1) and there is only one service in the network - Call Forwarding.

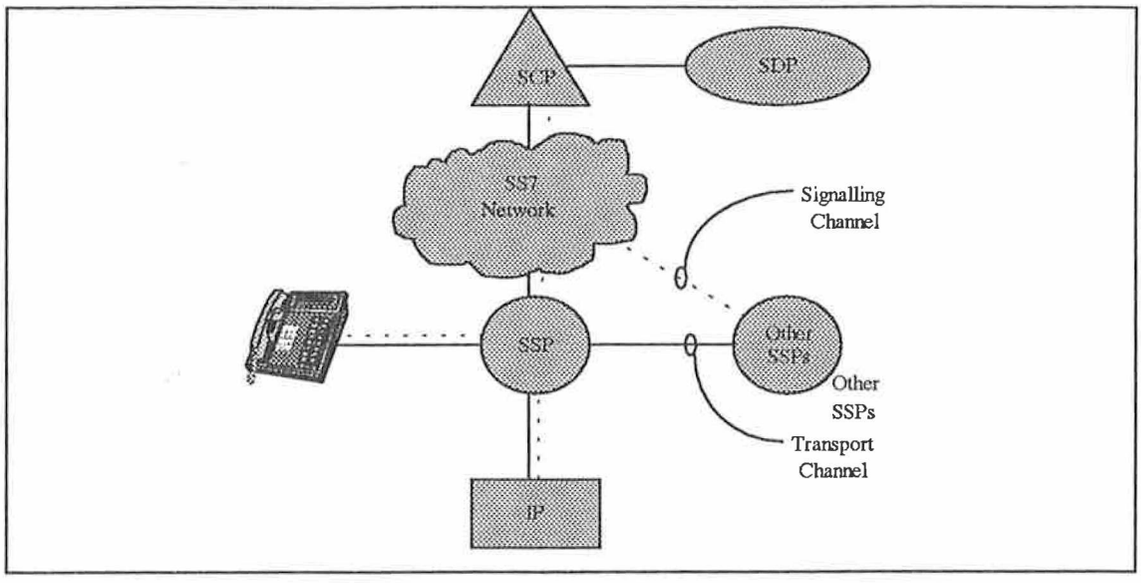

Figure 5-1 - Physical Entities in the network scenario

5.1.1.1 Simulation

In this scenario (in common with those discussed later), a body of simulations were performed. These simulations differed in the intensity of arrivals of the Call Forward requests into the network, which are shown in Table 5-2. The range of values for the arrival rate are $\lambda = \{5..45\}$, which corresponds to a Busy Hour Call Attempts (BHCA) range of {18000, 162000} BHCA.

The capacities of each of the network elements were chosen so that they would have the same utilisation factor, ρ, for the same simulation. This means that the whole of the network has the same utilisation and this was done to ease the comparison of the results.

Thus, the capacities of each element were chosen, based on the proportion of the service load which they would carry. Before presenting these loads, it is useful to revisit the interactions which occur in the Call Forward service. These interactions are shown in Figure 5-2.

The first point which should be made, is that there are no interactions with the IP. This means that the IP is not considered in the simulation, or in the analytic model for this network scenario. The capacities at each of the network elements are shown in Table 5-1.

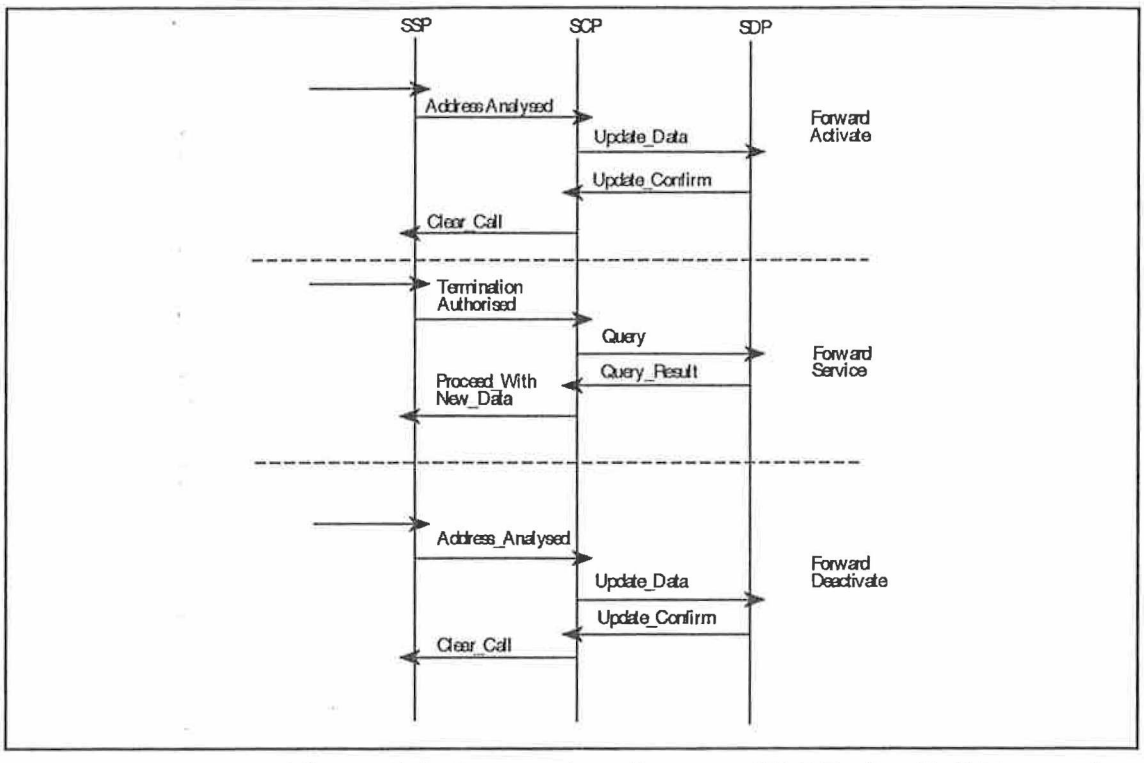

Figure 5-2- Interactions between IN PEs for Call Forward Service

Physical Entity	Capacity
SSP	90
SCP	90
SDP	45

Table 5-1 - Capacities of IN PEs

External Load	Network Utilisation
40	89%
35	78%
30	67%
25	56%
20	44%
15	33%
10	22%
5	11%

Table 5-2 - External Arrival and Network Utilisation Factors for Single Service Network

The inter arrival times and the response times of requests, at each of the PEs, are measured in the simulation and written out to an OPNET statistic wire. These statistics are written out to file using an OPNET probe and can be loaded into the OPNET analysis tool for further processing. At the end of each simulation, the output file can be loaded into the OPNET analysis package and each of the statistics gathered, can be graphed and analysed. In particular, the means, expected values and variances are calculated for each set of statistics. In the case of the simulations performed here, the interarrival times and the response times at each of the elements are of interest. Additionally, the simulation measures the post dialling delay experienced by each of the service requests, while undergoing service processing in the IN.

From the interarrival times, the mean arrival rate of requests at the network element can be found using the formula $\lambda = 1/\overline{X}$, where $\overline{X}$ is the mean interarrival time for incoming requests to the station. Once the arrival rate and the response time are known, the number of requests at the station can be found using a combination of Little's Law and Kingman's formula (as described in Chapter Four).

5.1.1.2 Analytic Formulation for Single Service Network

The parameters required for the analytic formulation, are as follows :

- The arrival rate of calls into the network λ_i, which are the values detailed above;
- The SVC of the interarrival time of calls to station i, Ka_i, which is one for Poisson arrivals;
- The capacities of each of the PEs μ_i, which are as presented in Table 5-1;
- The SVC of service time, at each element in the network Ks_i, which is zero at each PE, due to the deterministic nature of the service time; and
- The probability (or routing matrix) P, which specifies the routing of requests in the network. This matrix must be determined based on the characteristics of the services on the network and the configuration of the network. In this case, the routing matrix can be determined based on the characteristics of the Call Forward service and the fact that there is only one of each element type in the network. The specification of the routing matrix is discussed in more detail below.

The specification of the routing matrix is performed by analysing the interactions which occur as part of the Call Forwarding service and mapping these onto the network configuration. The interactions in the Call Forward service are shown in Figure 5-2. Two requests arrive at the SSP, one passes from the SSP to the SCP and the other signifies the end of the IN specific processing and thus ,for the purposes of our model, passes from the network. Two requests arrive at, and depart from, the SCP. One is sent to the SDP and when a reply is received from the SDP, the request is then sent onto the SSP. There is one arrival at the SDP and once SDP processing has completed, it is sent back to the SCP. From these interactions, the routing matrix can be defined.

First as a notation, each of the IN nodes is given an index to identify it (see Figure 5-3):

1. SSP;
2. SCP; and
3. SDP.

The IP is not considered, as it is not required by the Call Forward service. The elements of the routing matrix p_{ij}, define the probability that a request at node i will pass to node j. It should also be remembered that as all requests at a particular station must pass to another station or out of the network, the probabilities of a request at station i moving to another station is 1. This fact is useful in determining the elements of the routing matrix. The general means of calculating the routing probabilities is $p_{ij} = \sum_k p_{ij}^k * \hat{p}^k{}_i, \forall k$. The parameter p_{ij}^k is the probability of a request related to service k passing from station i to station j, given that it is already at station j. The parameter $\hat{p}^k{}_i$ is the probability of a request of service k being at station i.

The latter set of probabilities $\hat{p}^k{}_i$ are all unity as there is only one service in the network. The former set of probabilities p_{ij}^k can be found by examining the interaction diagram for the Call Forward service. The SSP has two departures from it, one to the SCP and one out of the network. This means that $p_{12} = \frac{1}{2}$, $p_{11} = 0$, $p_{13} = 0$, as there are no departures from the SSP to the SDP, or back to itself.

The SCP has two departures, one to the SSP and one to the SDP. This means that $p_{21} = \frac{1}{2}$, $p_{22} = 0$, $p_{23} = \frac{1}{2}$, as there are no departures from the SCP to itself.

The SDP has one departure to the SCP. This means that $p_{31} = 0$, $p_{32} = 1$, $p_{23} = 0$, as there are no departures to either the SSP or back to itself. This means that the routing matrix is now defined as:

$$P = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}.$$

Now that the routing matrix is complete, the parameters needed for the model have been found. Figure 5-3 shows the queuing network representation of the single service network, as it would be viewed when using the decomposition method.

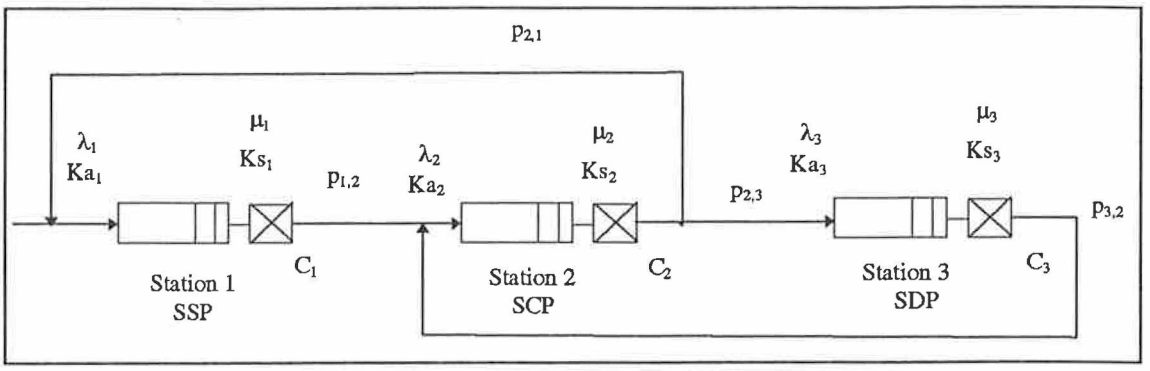

Figure 5-3 - Queuing Network Representation of Single Service Network.

5.1.1.3 Comparison of Results

Both the simulation and the implementation of the analytic formulation, were run for the configuration described above and each of the nine external loads in Table 5-2. The results for both are shown in the graphs below. The graphs in Figure 5-4 through Figure 5-6 show the results for the predicted (analytic solution) and the measured (simulated) mean number of requests at each of the PEs concerned. In each case, both the predicted and measured values rise steadily with the utilisation of the PE, so that the two sets of results follow the same trend. However, in order for the analytic formulation to be a good model, it is important not only that the results have the same characteristics, but also that the values are very close to the measured results. As can be seen from the graphs of the mean number of requests at each node, the predicted results are close to the measured results. This is particularly the case for lower utilisation factors.

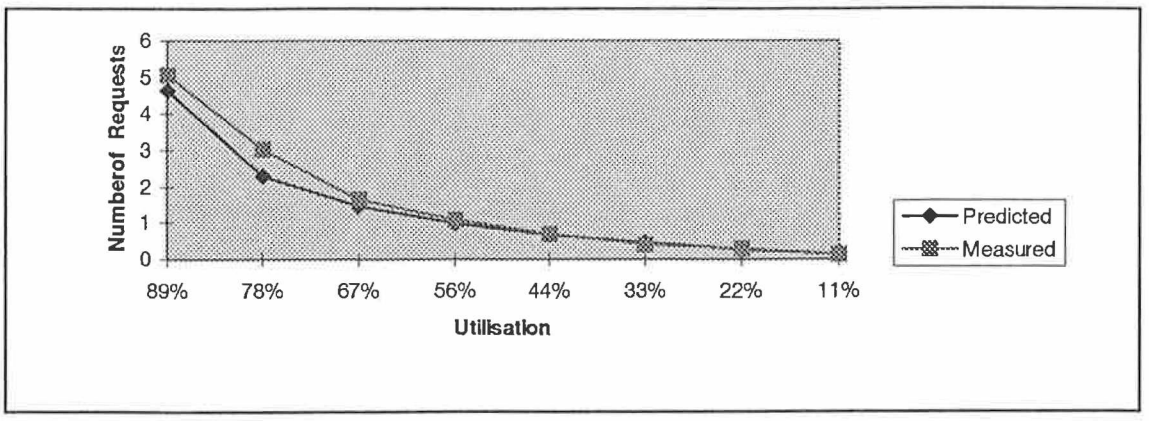

Figure 5-4 - Comparison for Number of Requests at SSP

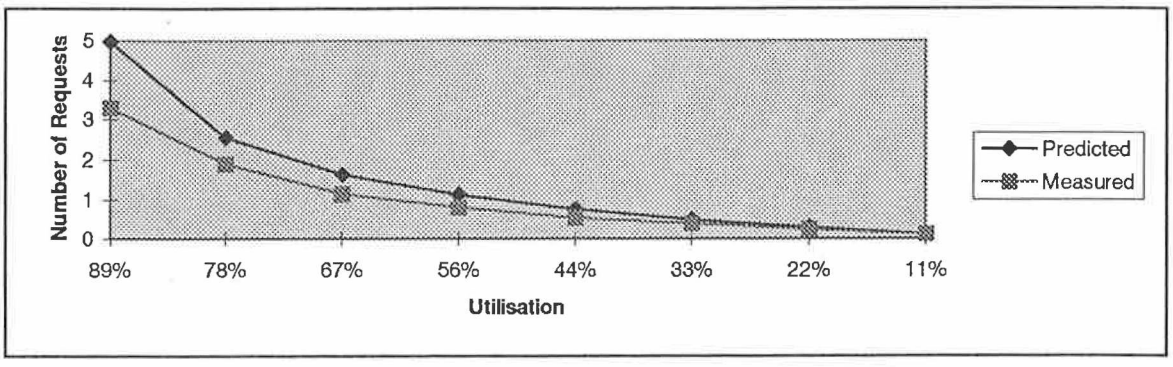

Figure 5-5 - Comparison for Number of Requests at Call Forward SLP

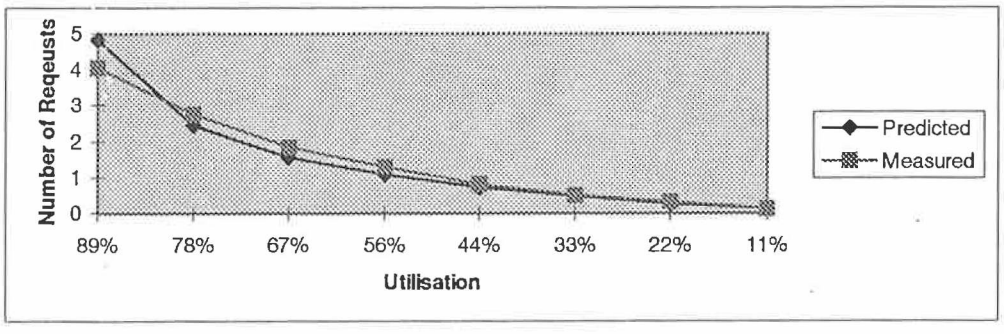

Figure 5-6 - Comparison for Number of Requests at SDP

The results for the mean response times are shown in the graphs in Figure 5-7 through Figure 5-9. As would be expected, the results for the measured and predicted, mean response time increase in direct proportion to the load upon the PE. Once more, the predicted results are very close to those measured from the simulation, particularly at lower utilisation rates. However, the results are less accurate as the utilisation factor increases.

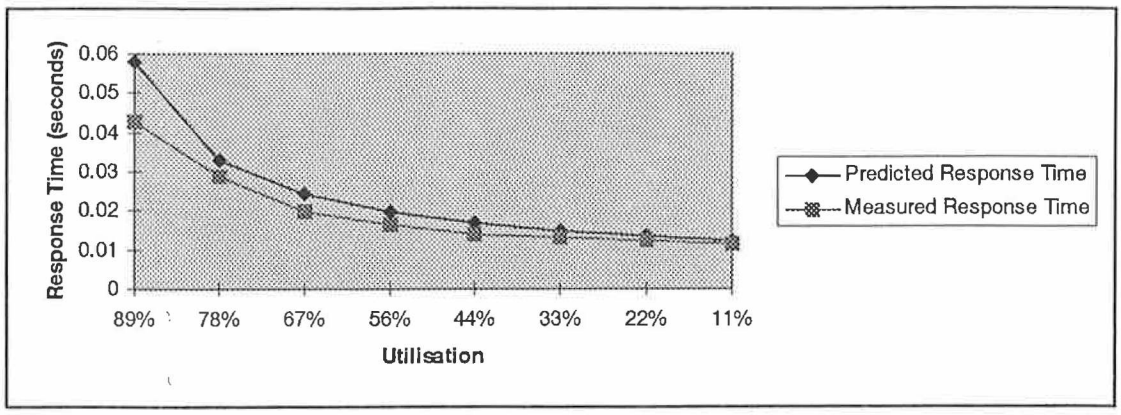

Figure 5-7 - Comparison of Response Times at SSP

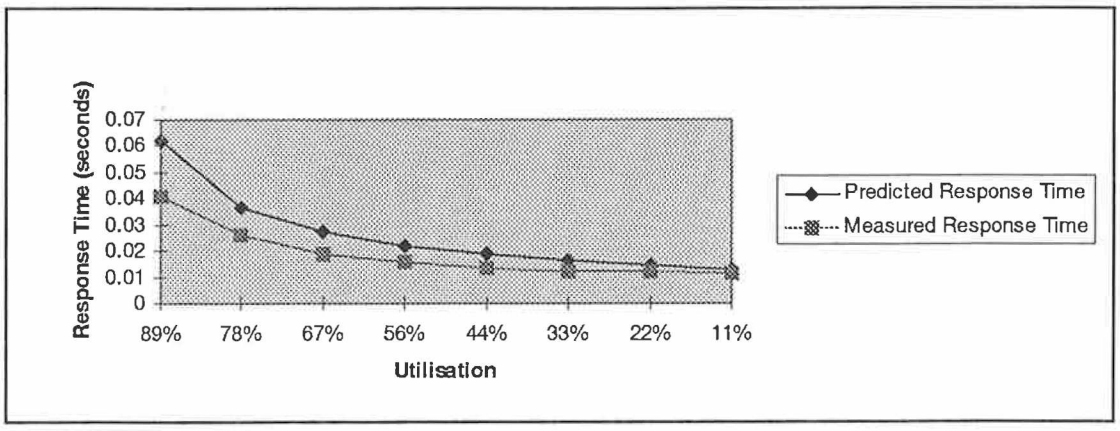

Figure 5-8 - Comparison of Response Times at Call Forward SLP

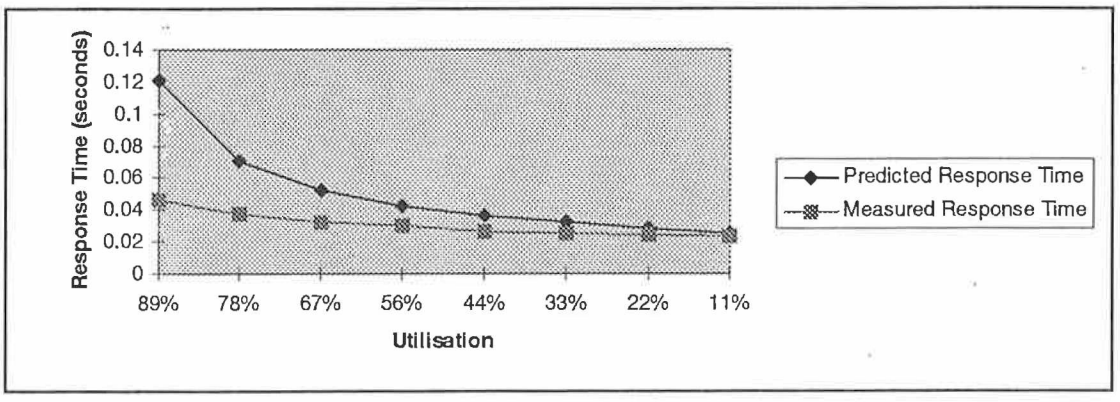

Figure 5-9 - Comparison of results for Response Times at SDP

The analytic model allows the prediction of the response times at each of the IN PEs. However, in the context of this report, the most important result is the PDD between user interactions related to IN specific processing. In other words, what is the contribution to PDD, of IN specific processing? If the mean response times for requests at each of the routes are known, then we can determine the PDDs by summing the response times accrued by the requests as they are processed at each IN PE. In the case of the Call Forward service, this post dialling delay is due to two

transactions at the SSP, two at the SCP and one at the SDP. This means that the mean PDD experienced by the Call Forwarding service can be written as follows $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \left(2 \times \overline{R_{scp}}\right) + \overline{R_{sdp}}$. This PDD was calculated for each of the nine loads on the network and the results are shown in Figure 5-10. Figure 5-10 shows a comparison of the predicted and measured mean post dialling delays for the Call Forwarding service. In common with the results presented earlier, the results are very close for lower utilisation of the network, but as the network utilisation increases the difference between the measured and predicted values also increases.

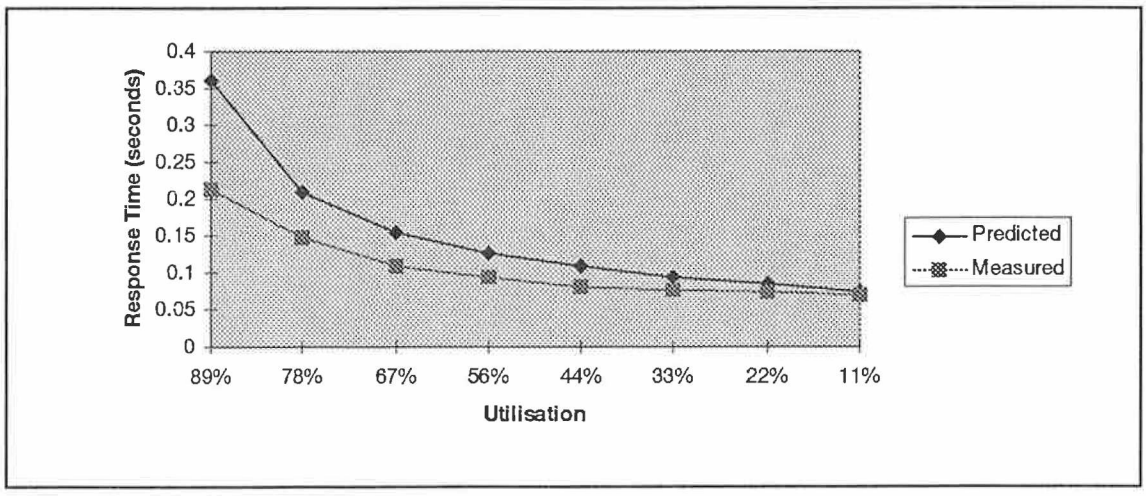

Figure 5-10 Comparison of Results for Predicted and Measured PDD for Call Forward Service

5.1.2 Two service Network

The two service network is a more complex scenario than the single service network as the characteristics of both services must be determined. This task is made simpler by the assumptions made earlier about response times at the different PEs and it is worthwhile to repeat them here. It is assumed that the response times for requests at the SSP, SDP and IP[16] are independent of the service, while the response times at the SCP are dependent on the service concerned. This means the response times calculated at the SSP, SDP and IP are common to all services, while those at the SCP are specific to each service.

[16] This assumption is academic in the case of the IP, as only the Televote service of the three considered, uses the IP.

The second service is added to the network above and in this case the Abbreviated Dialling (Abdial) service was chosen. An analysis of the interactions in the Abdial service (see Figure 5-11), shows that they are the same as the interactions which occur in the Call Forward service (see Figure 5-2). This means that there is no requirement to consider the IP, as neither service requires it.

In order to simplify the presentation and analysis of the results for this scenario, the parameters of the model have been chosen so that there is an equal arrival rate of requests for each service and the capacity of the SCP is equally split between the Call Forward and Abdial services. In common with the previous scenario, the capacities of the IN PEs in the network are chosen so that their utilisation is the same for a given load upon the network. The capacities are shown in Table 5-3. The same nine values are chosen for the external loads and the corresponding network utilisation are shown in Table 5-4.

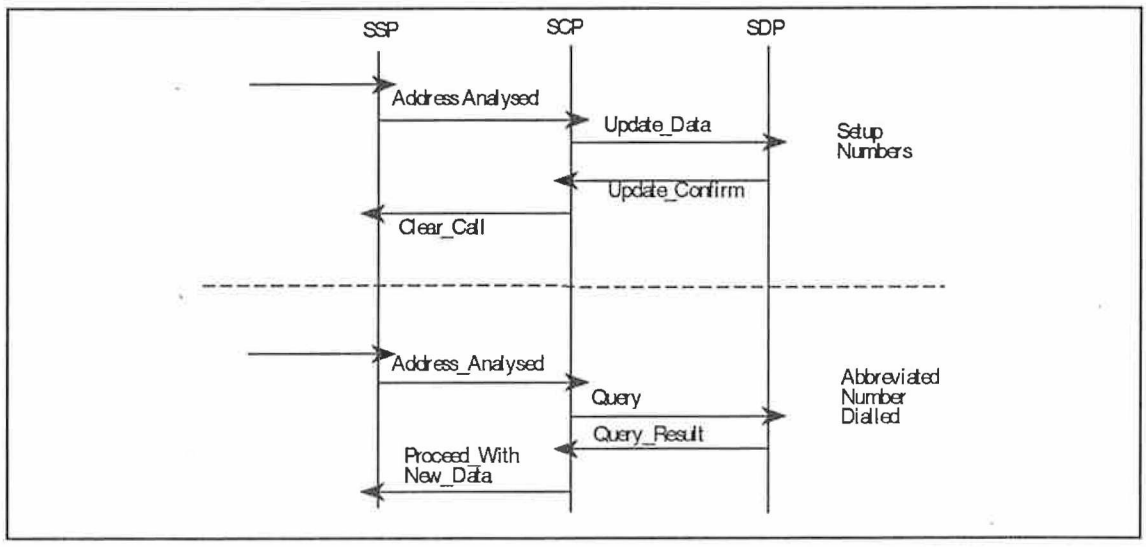

Figure 5-11 - Interactions for Abdial Service

Physical Entity	Capacity
SSP	90
SCP	90
SDP	45

Table 5-3 - Capacities of IN Resources in Two Service Network.

External Load	Network Utilisation
40	89%
35	78%
30	67%
25	56%
20	44%
15	33%
10	22%
5	11%

Table 5-4 - External Loads and Network Utilisation for Two Service Network.

5.1.2.1 Parameters of the Simulation

The parameters of the simulations reflect the network configurations and loads, as presented above. In particular, the parameters of the SCP must be set so that the capacities allocated to each of the services are correct. This means that the Call Forward and Abdial capacities are both 50%, while the Televote capacity is zero.

5.1.2.2 Parameters of the Analytic Solution

The parameters of the analytic solution reflect those presented above, but the manner in which the view of the network is developed is different. This is because of the fact that there are two services on the network with potentially different service times at each IN PE. The simplifying assumption made was that the SCP was the only PE at which the service times are service dependent. This means that the SSP, SDP can be modelled as a single queue with a single server. However, the solution for the SCP is more difficult as there are different service times at the queue for each class of customer. A simple solution to this problem is to model the SCP with two queues - one for each service. Each queue is served by a separate server. This does not infringe the assumptions in the model and is consistent with the service discipline implemented in the simulation. The reader will remember that in the SCP, all requests

are buffered in a common queue, but that each SLP only removes its' own requests from the queue. This means that the service discipline is modelled using a separate queue for each service. Thus, each of the queues has a capacity which is equal to the proportion of the SCP capacity allocated to the associated service type. In the current scenario, these capacities are 50% each so that the service rate of each queue is 45 transactions/second.

It now remains to determine the routing matrix for the two service scenario. The first task is to identify the notation used. In this scenario, the PEs are allocated the following numbers (see Figure 5-12):

1. SSP;
2. SCP (Call Forward); and
3. SCP (Abdial);
4. SDP.

The general means of calculating the routing probabilities is, $p_{ij} = \sum_k p_{ij}^k * \hat{p}^k{}_i, \forall k$.

The values of p_{ij}^k are found using the method described in section 5.1.1.2. These are:

$$P^{Abdial} = \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P^{CallForward} = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The reader will recall that the parameter $\hat{p}^k{}_i$ is the probability of a request of service k being at station i. This can be determined by examining the proportion of requests at the PE, belonging to service k. The values of the second parameter are:

$$\hat{P}^{Abdial} = \begin{bmatrix} 1/2 \\ 0 \\ 1 \\ 1/2 \end{bmatrix}, \hat{P}^{CallForward} = \begin{bmatrix} 1/2 \\ 1 \\ 0 \\ 1/2 \end{bmatrix}.$$

This means that the routing probabilities are:

$$P = \begin{bmatrix} 0 & 1/4 & 1/4 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}.$$

The parameters of the analytic model are now complete. Figure 5-12 shows the queuing network representation of the single service network, as it would be viewed when using the decomposition method.

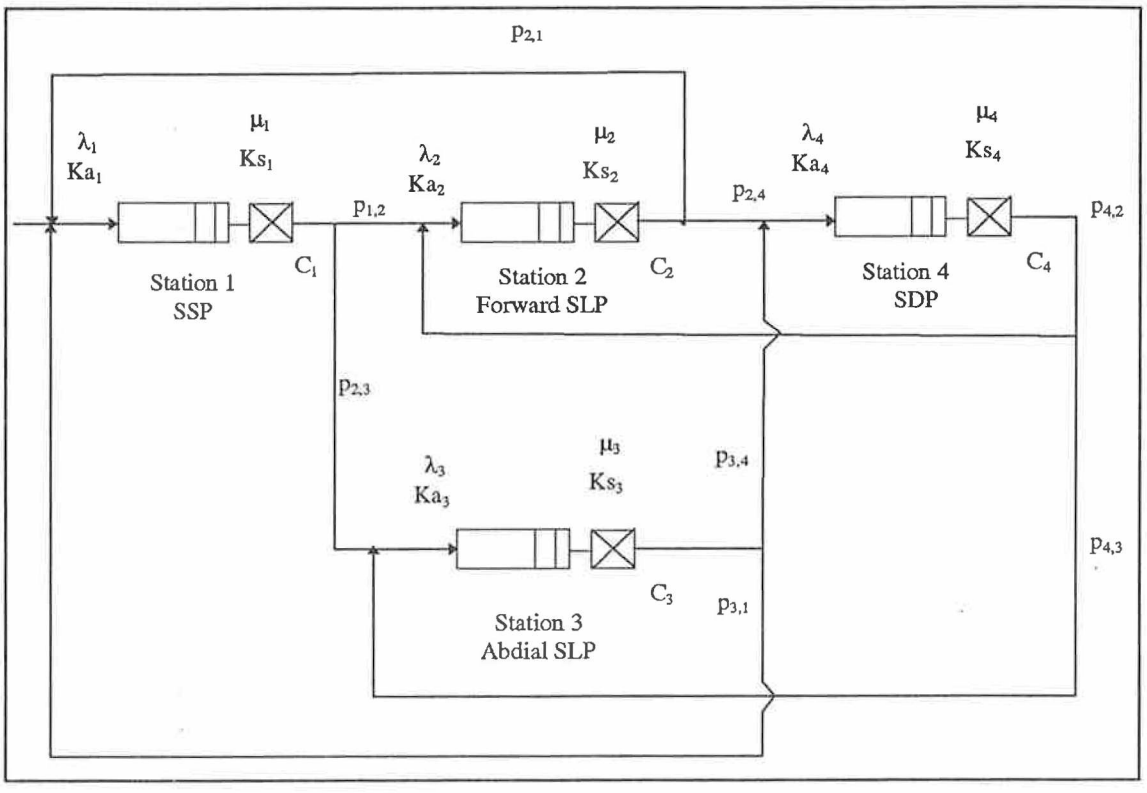

Figure 5-12 - Queuing Network Representation of the Two Service Network.

5.1.2.3 Comparison of results

The graphs shown in Figure 5-13 through Figure 5-16, show the results for the predicted and measured mean number of requests, at each of the PEs concerned. In each case, both the predicted and measured values rise steadily with the utilisation of the PE, so that the two sets of results follow the same trend. In common with the previous scenario, the predicted results are close to the measured results. This is particularly the case for lower and medium utilisation factors.

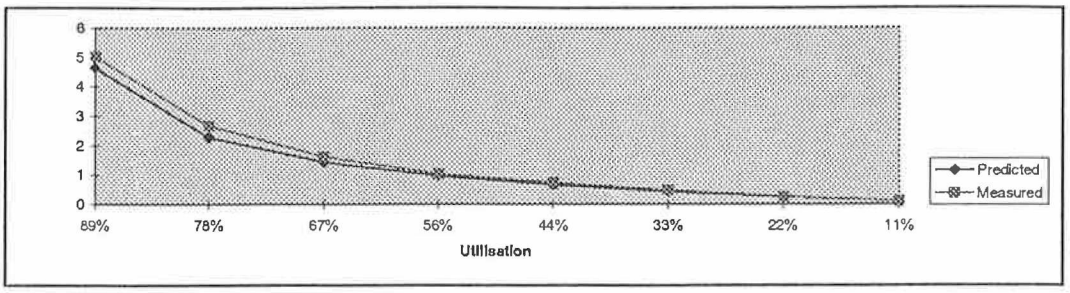

Figure 5-13 - Comparison of Results for Number of Requests at SSP.

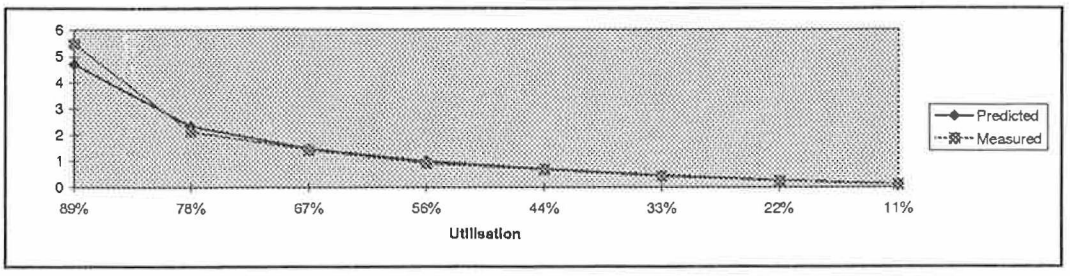

Figure 5-14 - Comparison of Results for Number of Requests at Call Forward SLP.

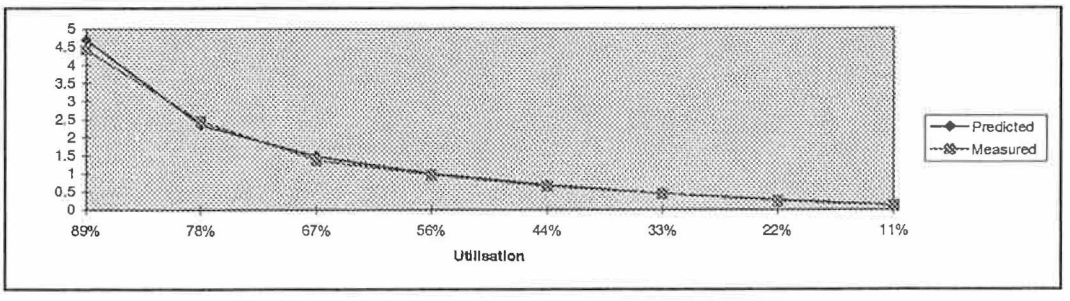

Figure 5-15 - Comparison of results for number of requests at Abdial SLP

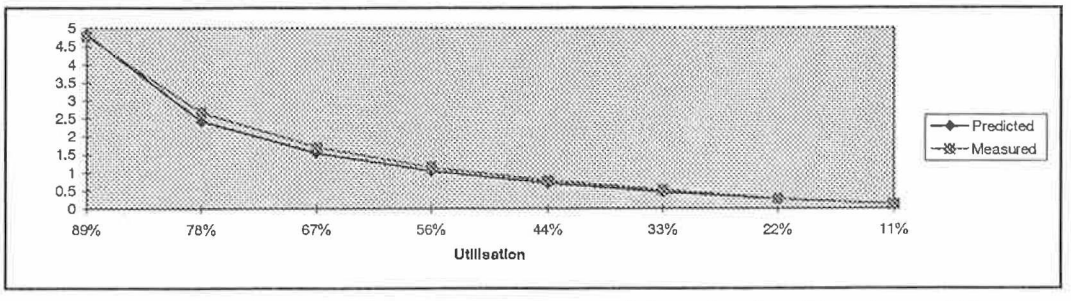

Figure 5-16 - Comparison of Results for Number of Requests at SDP.

The results for the mean response times are shown in the graphs in Figure 5-17 through Figure 5-20. As would be expected the results for the measured and predicted mean response time increase, in direct proportion to the load upon the PE. Once more the predicted results are very close to those measured from the simulation, particularly at lower utilisation rates. The results for the two SLPs are particularly good and show

that the modelling of the service discipline with two separate queues does not affect the accuracy of the model.

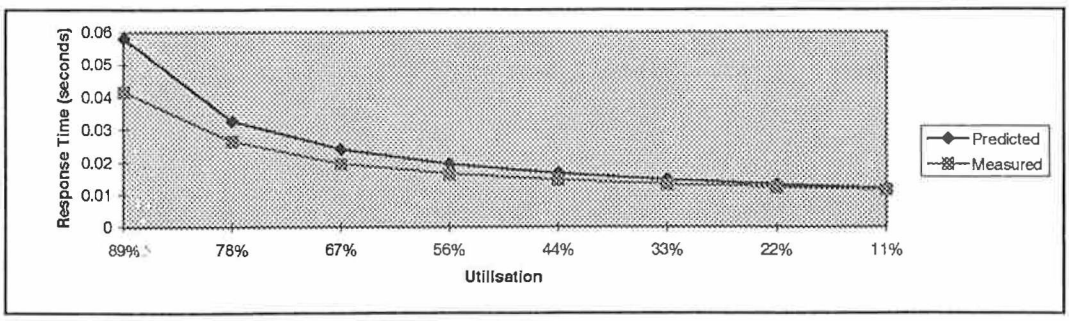

Figure 5-17 - Comparison of Results for Response Time at SSP.

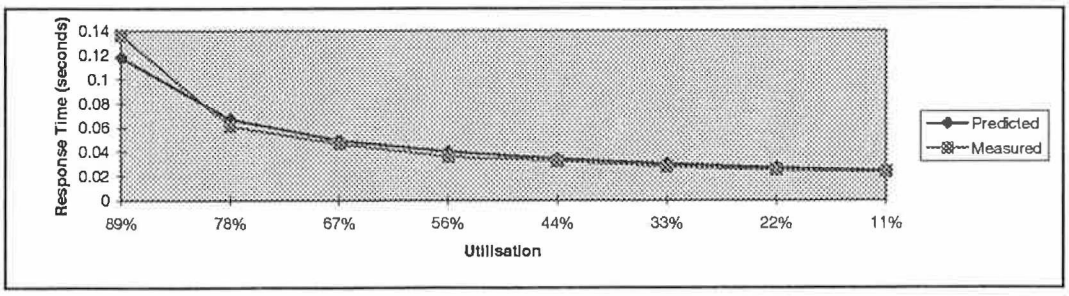

Figure 5-18 - Comparison of Results for Response Time at Call Forward SLP.

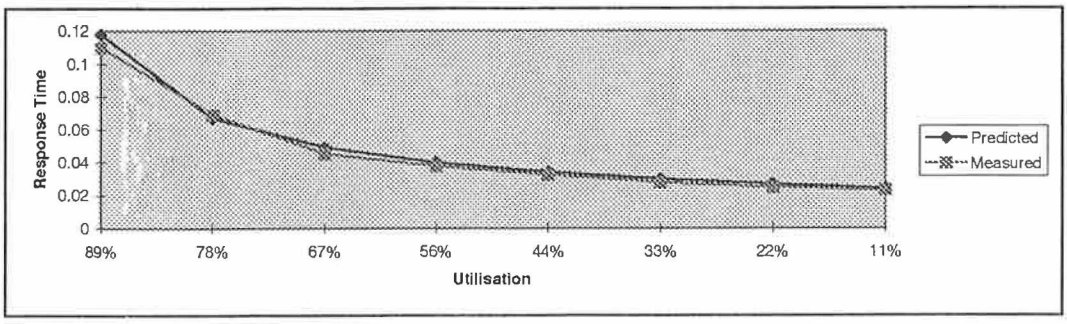

Figure 5-19 - Comparison of Results for Response Time at Abdial SLP.

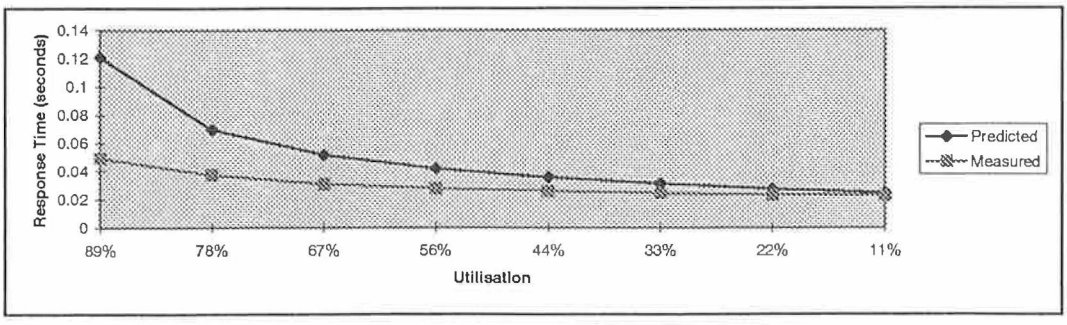

Figure 5-20 - Comparison of Results for Response Time at SDP.

The results for the mean post dialling delay for the Call Forward and Abdial services are shown in Figure 5-21 and Figure 5-22 respectively. As the interactions between the PEs are the same for both services, then the PDD is due to two transactions at the SSP, two at the SCP and one at the SDP. This means that the mean post dialling delay experienced by either service can be written as follows: $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \left(2 \times \overline{R_{scp}}\right) + \overline{R_{sdp}}$. Once more the results are very close, particularly at the lower utilisation rates, but the difference increases as the utilisation increases. Note that the curves on both graphs are very similar. This is because the capacities and service times for both services at the SCP are the same. If these parameters were different then the curves would have the same characteristics but possibly different values.

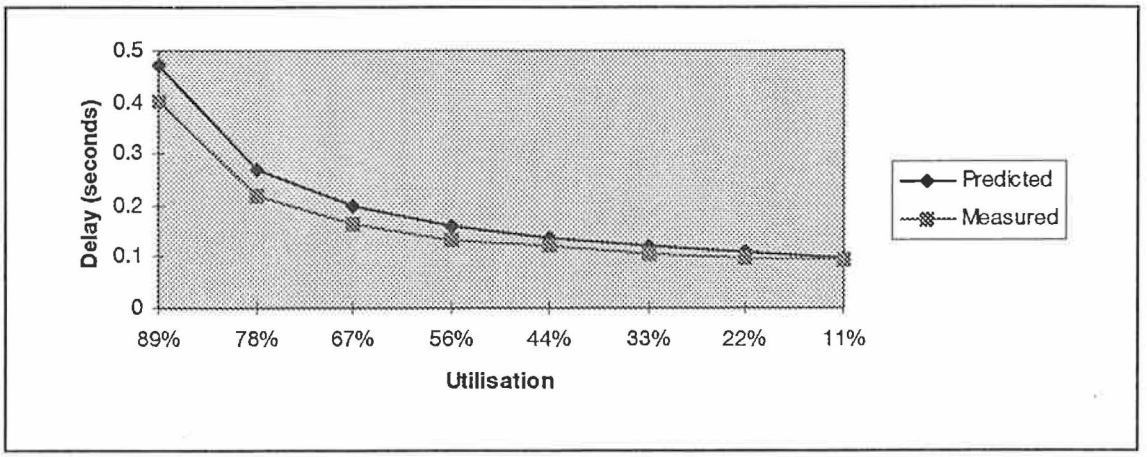

Figure 5-21 - Comparison of Results for PDD for Call Forward Service.

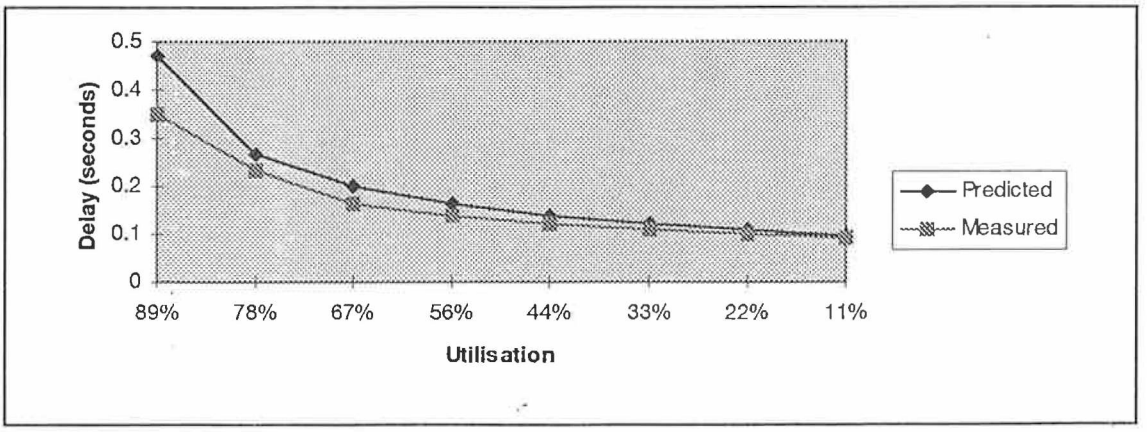

Figure 5-22 - Comparison of Results for PDD for Abdial Service.

5.1.3 Three Service Network

The final scenario which will be presented here, is the three service network. This scenario builds upon the two service network, with the addition of a third service - Televote. The Televote service makes extensive use of the IP, and thus the IP is considered in the results here.

Once again, to simplify the presentation and analysis of the results for this scenario, the parameters of the model have been chosen so that there is an equal arrival rate of requests for each service and the capacity of the SCP is equally split between the three services. Similarly to the previous scenario, the capacities of the IN PEs in the network are chosen so that their utilisation factors are the same for a given load upon the network. The capacities are shown in Table 5-5. The same nine values are chosen for the external loads and the corresponding network utilisation values are shown in Table 5-6.

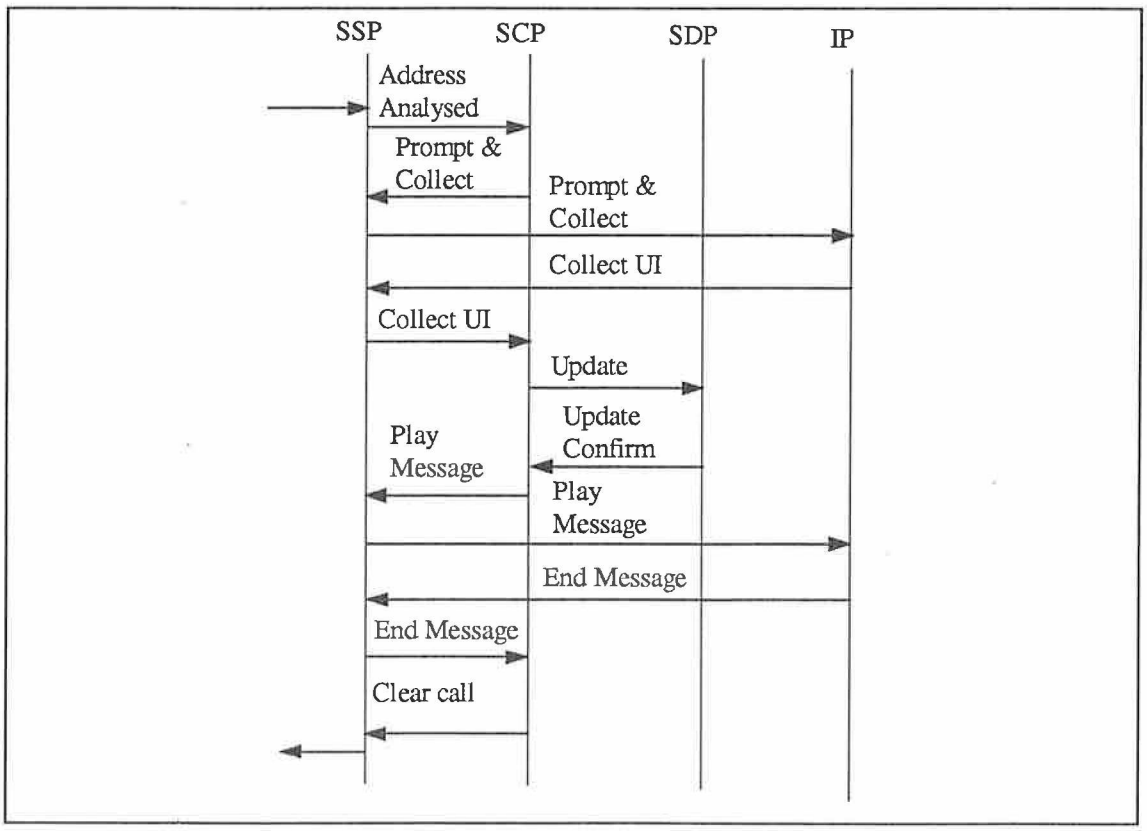

Figure 5-23 - Interactions for the Televote Service.

Physical Entity	Capacity
SSP	150
SCP	120
IP	30
SDP	45

Table 5-5 - Capacities of IN resources in Three Service Network.

External Load	Network Utilisation
40	89%
35	78%
30	67%
25	56%
20	44%
15	33%
10	22%
5	11%

Table 5-6 - External loads and Network Utilisation Factors for Three Service Network.

5.1.3.1 Simulation Parameters

The parameters of the simulations reflect the network configurations and loads, as presented above. In particular, the parameters of the SCP must be set so that the capacities allocated to each of the services are representative of their respective loads upon the SCP. This means that the Call Forward and Abdial capacities are both 25%, while the Televote capacity is 50%. The reason that the Televote capacity is twice that of the other services, is that Televote places twice the load upon the SCP as the other services.

5.1.3.2 Analytic solution parameters

The parameters of the analytic solution reflect those presented above, but the previous approach to modelling the different service times at the SCP, is continued. This means that the SCP is modelled by three queues - one for each of the services. Each of the queues has a capacity, which is equal to the proportion of the capacity of the SCP, which is allocated to the service. In the current scenario, these capacities are 25% to each of Call Forward and Abdial and 50% to Televote. This means that the service rate of the Televote queue is 60 requests/second (216000 BHCA), while the other two are 30 requests/second (108000 BHCA).

Finally, the routing matrix must be found for the three service scenario. The first task is to identify the notation used. In this scenario, the PEs are allocated the following numbers (see Figure 5-24):

1. SSP;
2. SCP (Televote);
3. SCP (Call Forward);
4. SCP (Abdial);
5. SDP; and
6. IP.

The general means of calculating the routing probabilities is, $p_{ij} = \sum_k p_{ij}^k * \hat{p}^k{}_i, \forall k$.

From inspection of the interaction diagram for the Televote services, the values of the first parameter can be determined and the values for the other two services are already known. These are:

$$P^{Televote} = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 1/3 \\ 3/4 & 0 & 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$P^{Abdial} = \begin{bmatrix} 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$P^{CallForward} = \begin{bmatrix} 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The values of the second parameter are :

$$\hat{P}^{Televote} = \begin{bmatrix} 6/10 \\ 1 \\ 0 \\ 0 \\ 1/3 \\ 1 \end{bmatrix}, \hat{P}^{CallForward} = \begin{bmatrix} 2/10 \\ 0 \\ 1 \\ 0 \\ 1/3 \\ 0 \end{bmatrix}, \hat{P}^{Abdial} = \begin{bmatrix} 2/10 \\ 0 \\ 0 \\ 1 \\ 1/3 \\ 0 \end{bmatrix}.$$

These values are explained by determining the overall number of requests at the PE and then determining what proportion of each, belong to which service. This means that the routing probabilities are:

$$P = \begin{bmatrix} 0 & 3/10 & 1/10 & 1/10 & 0 & 2/10 \\ 3/4 & 0 & 0 & 0 & 1/4 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The parameters of the analytic model are now complete. Figure 5-24 shows the queuing network representation of the three service network. as it would be viewed when using the decomposition method.

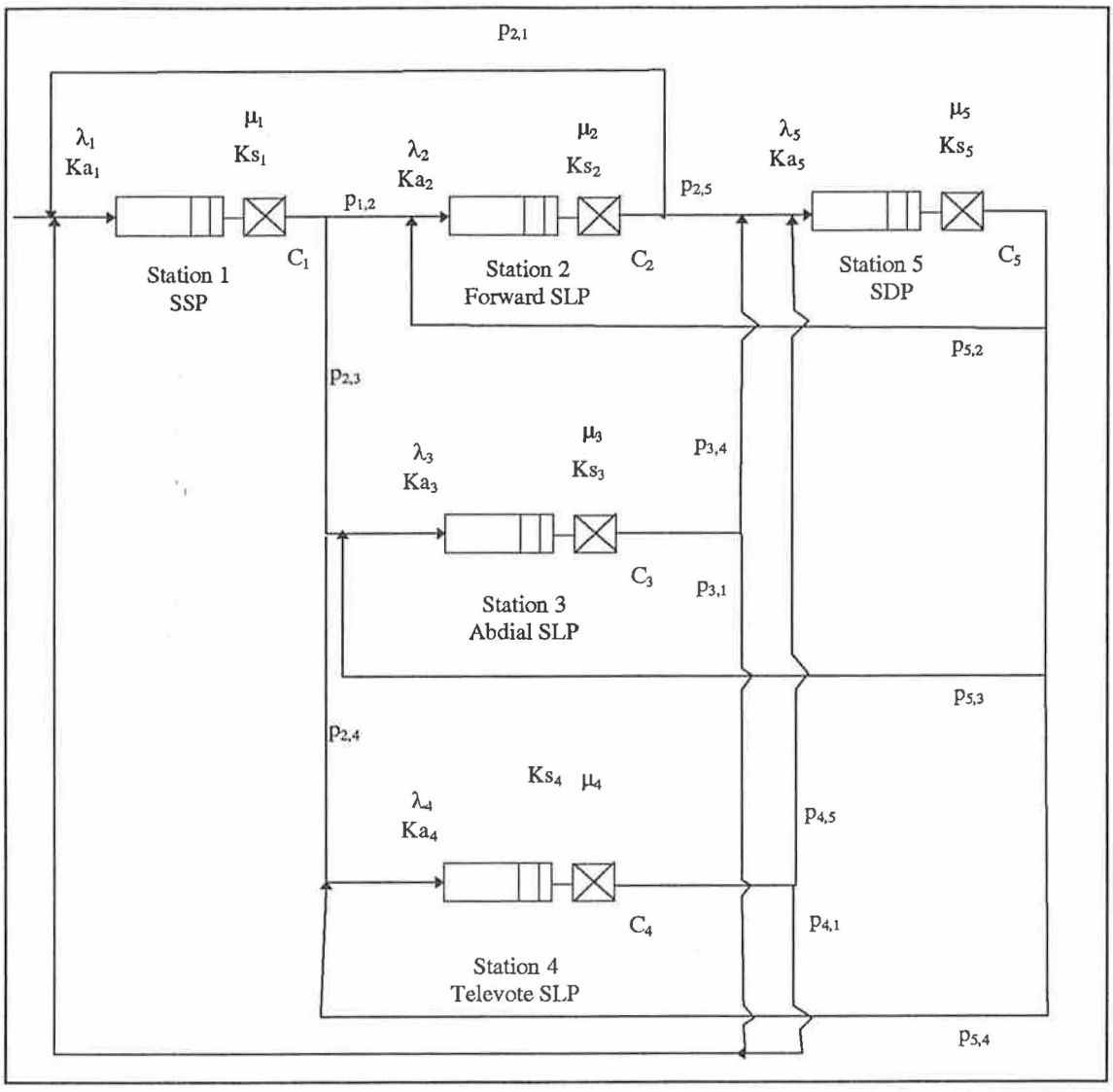

Figure 5-24 - Queuing Network Representation of Three Service Network.

5.1.3.3 Comparison of results

Once more, both the simulation and analytic solution were run for each of the external loads. presented above and the results gathered. The graphs shown in Figure 5-25 through Figure 5-30, show the results for the predicted (analytic solution) and the measured (simulated) mean number of requests, at each of the PEs concerned. In each case, both the predicted and measured values rise steadily with the utilisation of the PE, so that the two sets of results follow the same trend. In this case also, the predicted results are close to the measured results and the predictions are closer to the measured values for lower and medium utilisation factors.

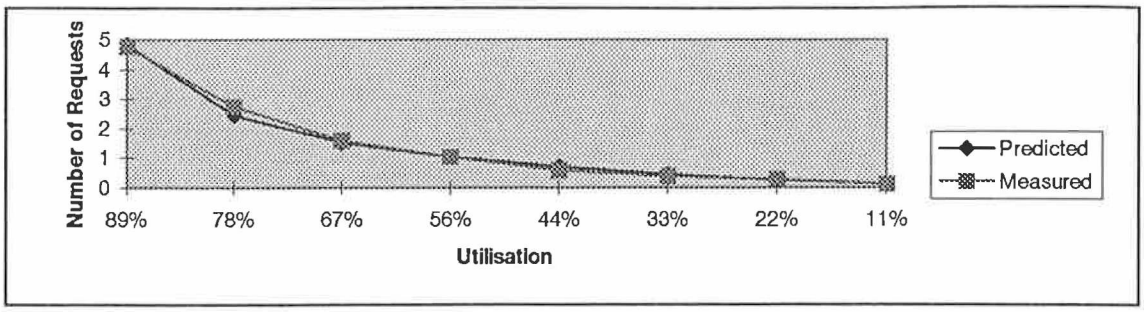

Figure 5-25 - Comparison of Results for Mean Number of Requests at SSP.

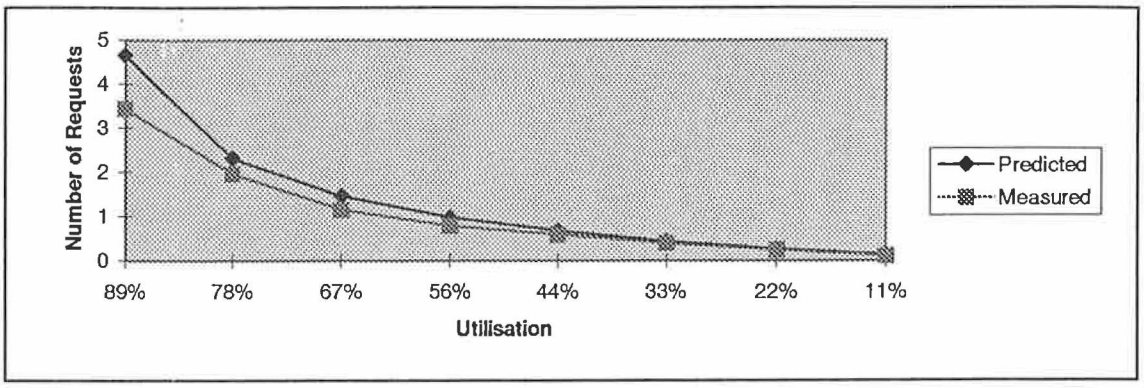

Figure 5-26 - Comparison of Results for Mean Number of Requests at Televote SLP.

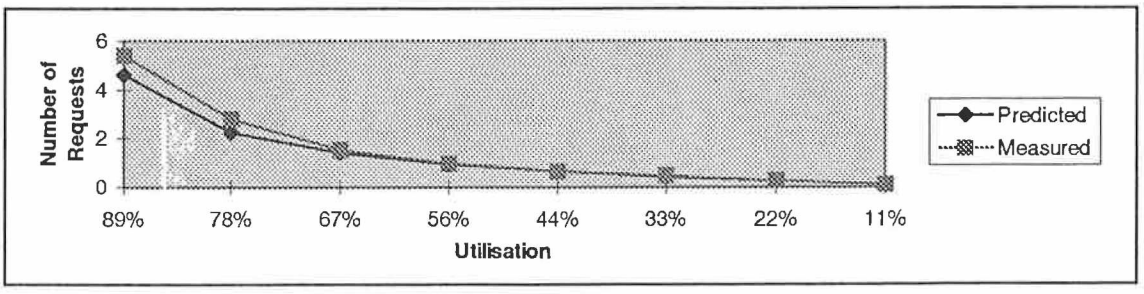

Figure 5-27 - Comparison of Results for Mean Number of Requests at Call Forward SLP.

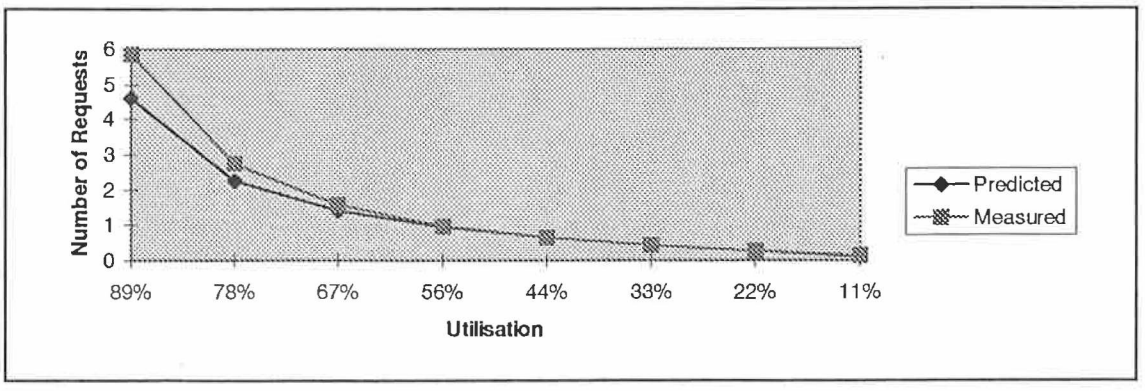

Figure 5-28 - Comparison of Results for Mean Number of Requests at Abdial SLP.

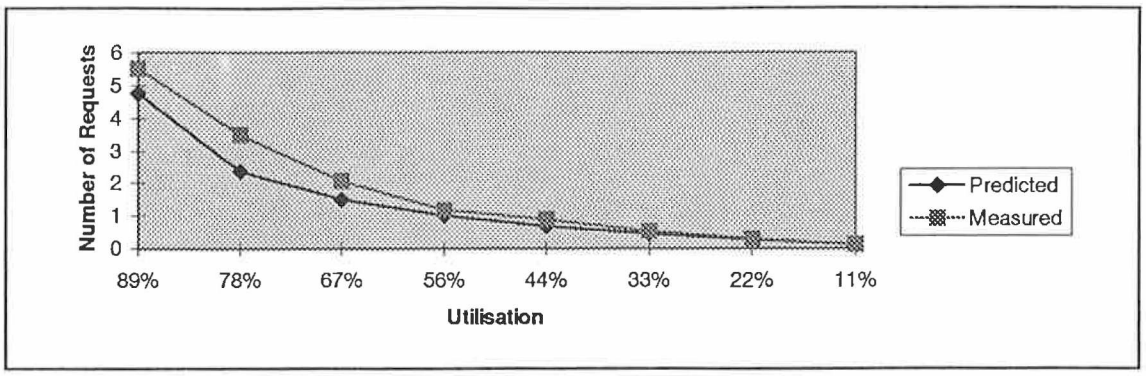

Figure 5-29 - Comparison of Results for Mean Number of Requests at SDP.

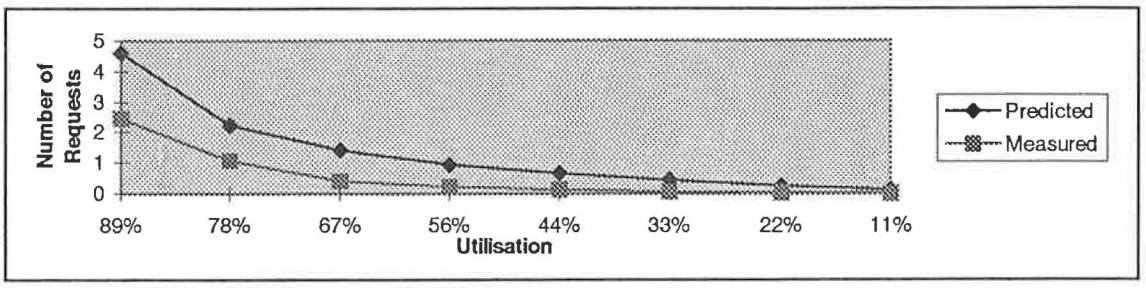

Figure 5-30 - Comparison of Results for Mean Number of Requests at IP.

The results for the mean response times are shown in the graphs in Figure 5-31 through Figure 5-36. The predicted results are very close to those measured from the simulation, with the predictions closer to the measured values at lower utilisation rates.

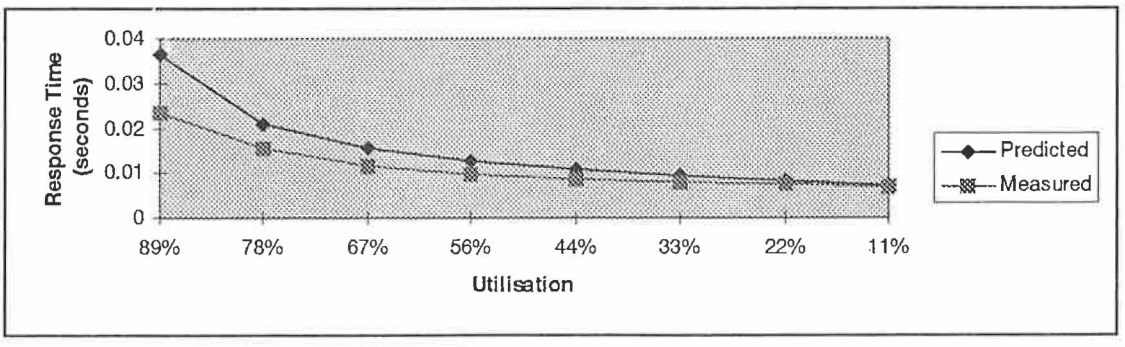

Figure 5-31 - Comparison of Results for Mean Response Time at SSP.

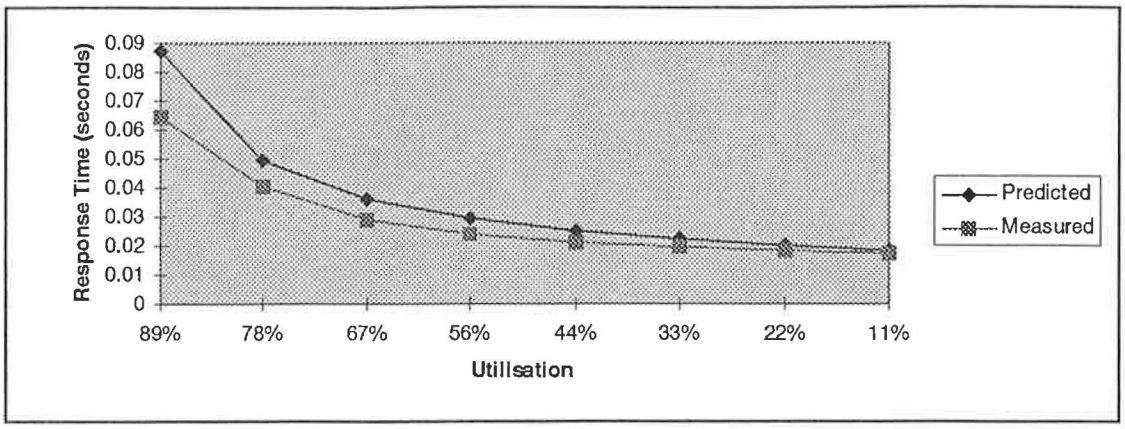

Figure 5-32 - Comparison of Results for Mean Response Time at Televote SLP.

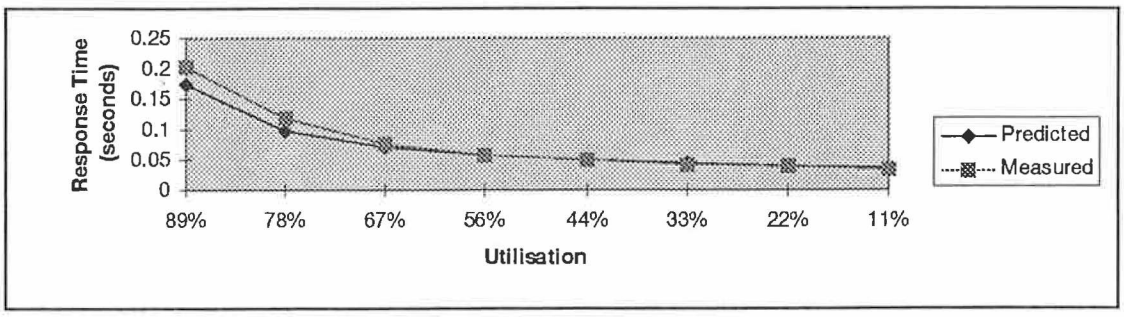

Figure 5-33 - Comparison of Results for Mean Response Time at Call Forward SLP.

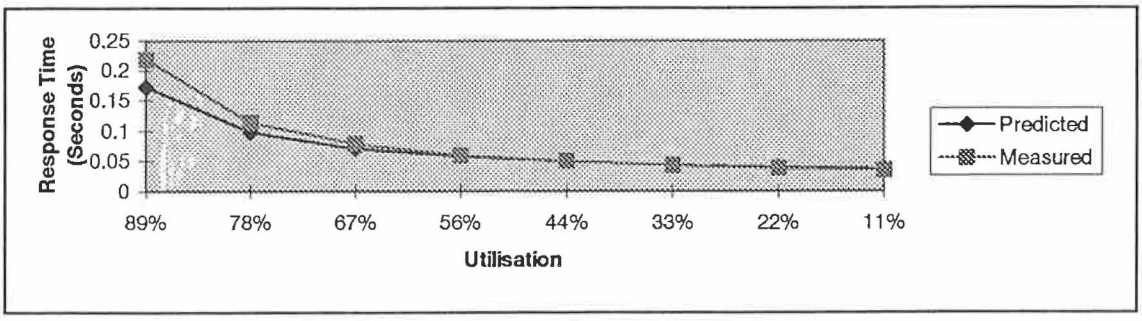

Figure 5-34 - Comparison of Results for Mean Response Time at Abdial SLP.

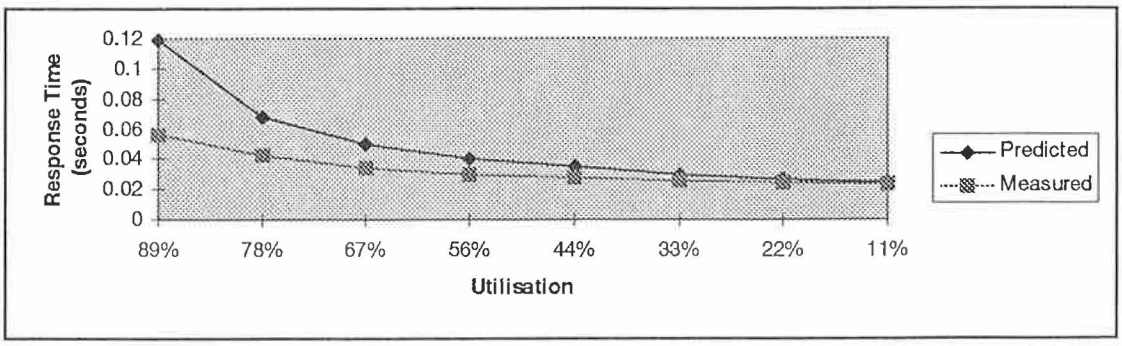

Figure 5-35 - Comparison of Results for Mean Response Time at SDP

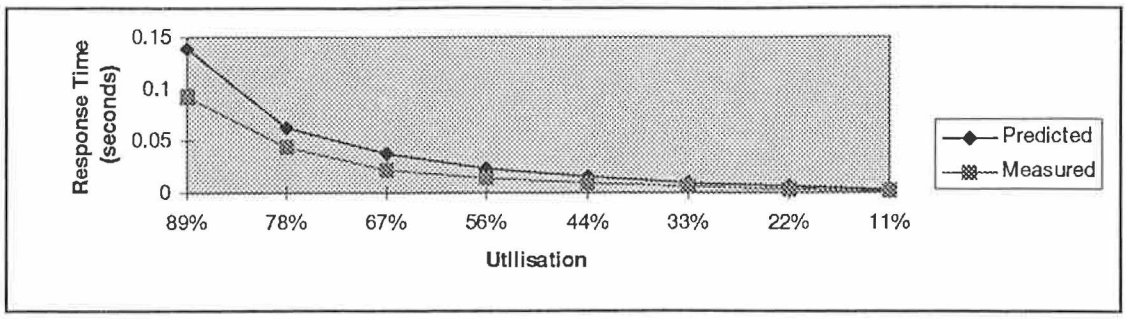

Figure 5-36 - Comparison of Results for Mean Response Time at IP.

The results for the mean PDD for the three services, are shown in Figure 5-37 through Figure 5-41. The interactions for the Call Forward and Abdial services have already been presented earlier, with the related formula for the mean PDD. However, the Televote service is more complex, and there are three response times which should be calculated:

1. The time for IN specific processing between the call start and the first user interaction with the IP. This is denoted as Televote 'part 1' and the mean response time for part 1 is $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \overline{R_{scp}} + \overline{R_{ip}}$.

2. The time for IN specific processing between the first IP user interaction and the second IP user interaction. This is denoted as Televote 'part 2' and the mean response time for this part is $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \left(2 \times \overline{R_{scp}}\right) + \overline{R_{sdp}} + \overline{R_{ip}}$.

3. The time for IN specific processing between the second IP user interaction and the termination of the call. This is denoted as Televote 'part 3' and the mean response time for part 3 is $\overline{R} = \left(2 \times \overline{R_{ssp}}\right) + \overline{R_{scp}}$.

Once more, the results for the Call Forward and Televote services are very close to each other, due to the fact that the capacities and service times for both services at the SCP are the same. A more important feature of these two curves, is the difference between the predicted and measured values - the difference between them is very small. This means that the prediction model was very accurate for these two services in this scenario.

The results for the three Televote parts are not as accurate - with the results for Televote part two in particular showing a divergence between the measured and predicted values for high utilisation factors. There are two reasons for this:

1. The nature of the Televote service. Televote is a more complex service then either Abdial or Call Forward and requires mode interactions with the PEs. In particular, it is the only one of the services considered which requires an interaction at the IP. The IP is directly connected to the SSP and the reader will recall from Chapter Four, that the assumption was made that all requests are homogeneous at the SSP.

   However, in the simulation, the requests coming into the SSP from the IP will be routed to the Televote SCP and will have the associated inter arrival characteristics. In the analytic model, this stream of requests from the IP will be split across the three SCPs based on the values in the routing matrix[17]. This effectively introduces an error in the estimation of the streams coming from the SSP. As the routing matrix is weighted in favour of the Televote SCP, in this case the error manifests itself there (see Figure 5-32) more than anywhere else.

   This analysis is backed up by the fact that when a single service network supporting only the Televote service is analysed, this error disappears (as there is only one type of request at the SSP). Thus, the problem here is with the simplifying assumption that all requests are homogeneous at the SCP.

2. The second reason is related to the fact that Televote part 2 has the greatest number of PE interactions of all of the services/service parts considered. Thus, the contribution of the estimation error at each of the PEs, is magnified a little more.

[17] The routing matrix makes a routing decision based purely on the current location of the request and does not take into account where the request has been before the current station.

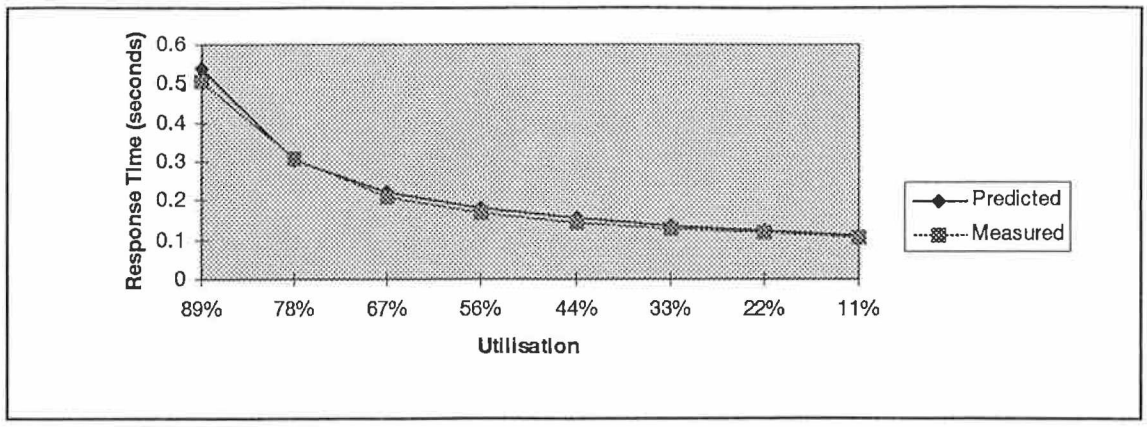

Figure 5-37 - Comparison of Results for PDD for Call Forward.

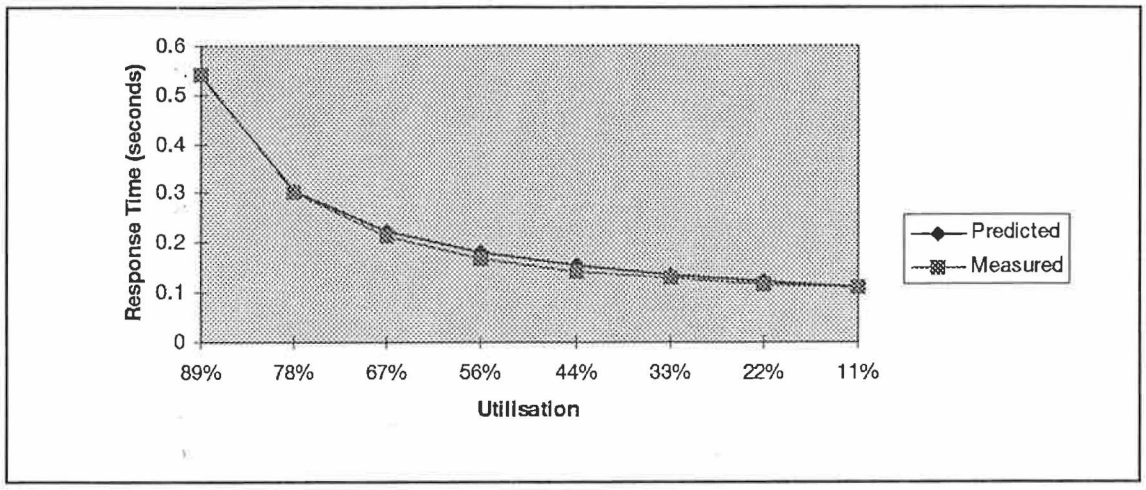

Figure 5-38 - Comparison of Results for PDD for Abdial.

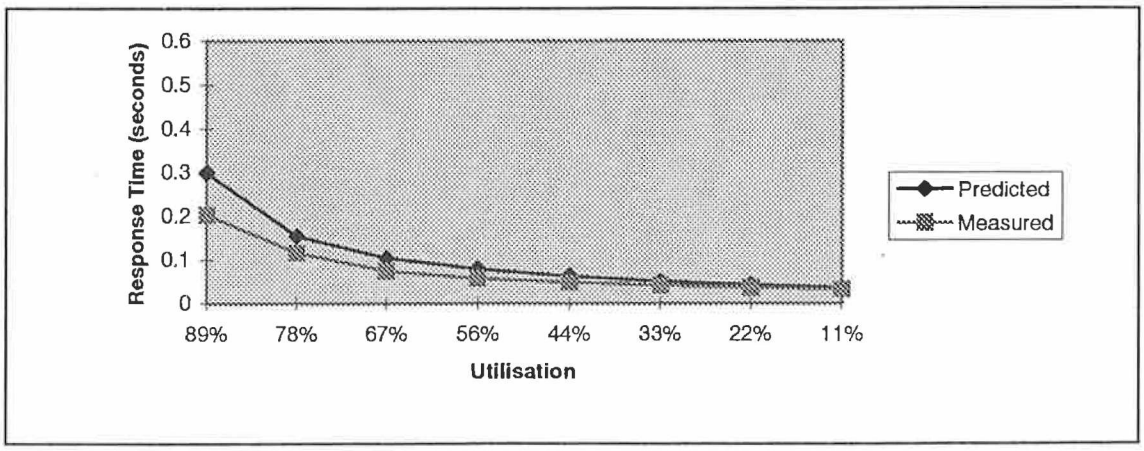

Figure 5-39 - Comparison of Results for PDD for Televote part 1.

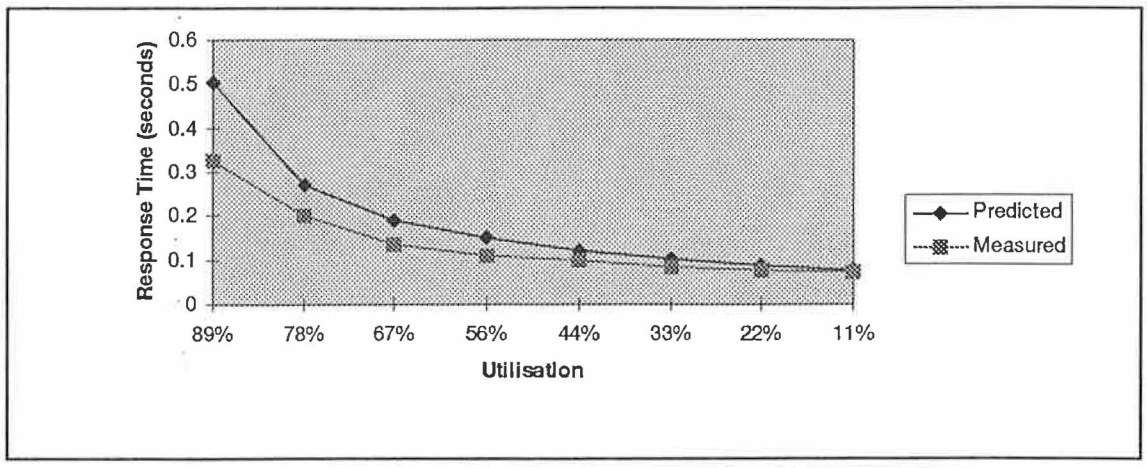

Figure 5-40 - Comparison of Results for PDD for Televote part 2.

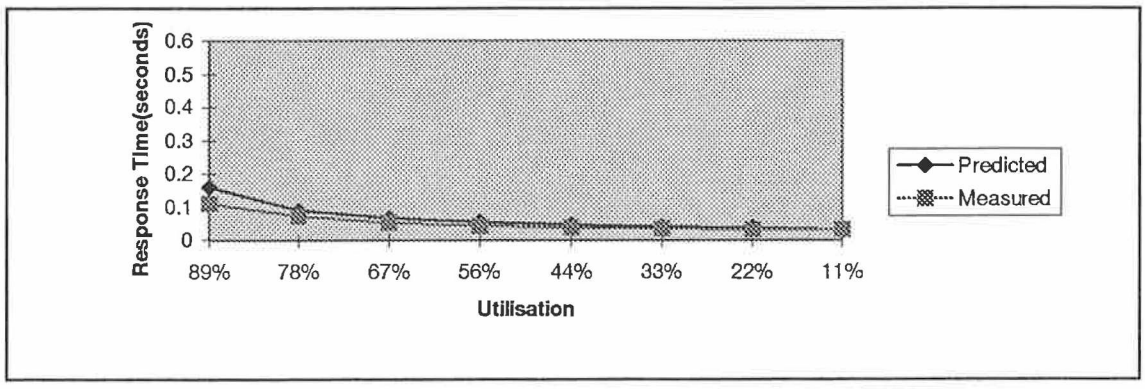

Figure 5-41 - Comparison of Results for PDD for Televote part 3.

To summarise, a simple model of the performance of an IN structured network has been proposed. This model has been analytically formulated using the decomposition method and allows the estimation of the mean load upon each IN resource, the mean response time at each IN resource and the mean service response time for an IN service. This service response time is equal to the component of the PDD of the IN call due to IN specific processing.

A comparison of the results of the simulation and the analytic model shows that the approximation is a good one. In particular, the results show that the analytic model allows the accurate prediction of the behaviour of the network for low and medium loads. However, as the utilisation of the resource increases the accuracy of the model is not as good. This behaviour was found to occur in all of the network scenarios, which were addressed. This characteristic has not been described in publications by

other authors using the method for analysis of IN, but it is implicit in the results published by Lodge et al [23],[24].

An analysis of the results of the three service network, shows that the assumption that all requests are homogeneous at the SSP can introduce an estimation error. This estimation error is really only significant at high utilisation factors.

Chapter 6

6. Conclusions

6.1 Conclusions/Summary

The objective of the work described in this report was to define a simple and tractable model of the performance of an IN structured telecoms network. In particular, the focus of the work was to define the model, such that it would estimate the performance of the service specific processing of the IN, under normal (or steady state) conditions. The work did not aim to address the performance of the wider switching or signalling networks, as this would have meant that the scope of the work would have been too large. The model defined here is both general and extendible and could form the basis of such a unified model of a telecoms network.

As a first step to achieve this goal, a survey of the current state of the art in the modelling of the performance of an Intelligent Network was performed. The results of this survey show that while the performance issues and challenges related to the performance of the network are well documented, there is a lack of solutions to these problems. The performance parameters of primary interest for an IN, as identified from this survey are:

- The PDD, which is the delay which the user perceived between successive interactions with a service;
- The Impaired Call Rate, which is the probability that an IN call will not gain access to full service features;
- The Blocking Probability, which is the probability that a call will be blocked.

However, most of the work which has been conducted into the performance modelling of an IN structured network, has concentrated on congestion control and the modelling of congestion strategies for the IN. As the aim of the work described here, is to model the steady state behaviour, much of this work is not readily applicable. The main work of relevance at the time of the development of the model was that of Leever et al [18] who model the network resources as a series of M/G/1 queues. Subsequent work in parallel with the model development, has validated many of the assumptions made

here. However, the key difference between the model presented here and other models, is that the nature of the flows between the IN entities is considered.

In order to develop the model of the network, a set of three candidate IN services were chosen, Televote, Abbreviated Dialling and Call Forward. Implementations of these services have been described, with a mapping from the SIB description to the information flows between the IN functional entities

The queuing model which was developed for the IN considered these three services, and a subset of the overall set of IN physical entities required to implement all of the IN functional entities. This subset consisted of the SSP, IP, SCP and SDP. As a simplifying assumption, it was decided that the main performance parameter of interest in the IN was the mean PDD for each service type. The estimation of this value also required the prediction of the mean response time experienced by the service requests, at each of the IN PEs.

In the queuing model defined, the network resources are modelled as a network of G/G/1 queues. The analytic formulation of this solution, uses the decomposition method as a means of estimating the mean response time experienced by each service request type, at each of the IN PEs. By adding these values together for the overall service processing of a IN call, the mean post dialling delay can also be estimated.

In order to validate this analytic solution, a simulation of the same IN structured network, with the same set of services was developed. This simulation was used to determine the accuracy of the results produced by the analytic formulation. To this end, a large number of simulations of different scenarios were performed, and the results were compared with those from the corresponding analytic solution. A comparison for a representative subset of these results, has been presented in this report.

A comparison of the simulated and estimated results, shows that the characteristics are similar for both sets of results as the utilisation of the network resources changes. However, it can be seen that the accuracy of the estimates from the analytic formulation suffers as the utilisation factor of each resource increases. This phenomenon can be seen to occur for all of the IN resources.

Overall, it can be seen that the analytic formulation, which has been proposed, is reasonably accurate and is quite easily solved on a computer. An initial version of the model was used as part of the prototype IN provisioning/performance tool described

in [25]. While the model may not be accurate enough to use as a standalone industrial strength tool for performance analysis/prediction, it could be very useful when used in conjunction with a simulation.

In such a scenario, the analytic formulation could be solved to generate a set of proposed parameter values, which are used to start a simulation. In this way, the search space of the simulation is much reduced, allowing the simulation to validate and 'fine tune' the results of the analytic solution.

6.2 Recommendations for Future Work

A very important area of work to be done, is to examine in more detail the complex interactions which occur between the IN services and between the IN network resources. In doing so, a greater understanding of these issues could be gained. Additionally, the model could be enhanced to remove some of the simplifying assumptions made during the course of this work.

Another improvement would be to gather more information about the characteristics of the chosen parameters. In particular, the distributions of the response times and PDDs would be useful information. This means that not only would the user have knowledge of the mean response times and PDDs, but would also know of the variations of the parameters about these mean values. This would be an asset to a network planner.

Chapter 7

7. References

[1] Ahlfors U., Overload Control Issues in evolving Distributed Network Architectures, Technical Report, Dept. of Communications Systems, Lund University, CODEN : LUTEDX (TETS-7149)/1-21/(1995) & local23. Lund, Sweden, 1994.

[2] ITU-T, Recommendation Q.1201, Principles of Intelligent Network Architecture, 1993.

[3] ITU-T, Recommendation Q.1202, Intelligent Network - Service Plane Architecture, 1993.

[4] ITU-T, Recommendation Q.1211, Introduction to intelligent network capability set 1, 1993.

[5] ITU-T, Recommendation Q.1213, Global functional plane for intelligent network CS-1, 1995.

[6] ITU-T, Recommendation Q.1214, Distributed functional plane for intelligent network CS-1, 1995.

[7] ITU-T, Recommendation Q.1215, Physical plane for intelligent network CS-1, 1995.

[8] Ambrosch W, Maher A., Sasscer B., The Intelligent Network, Springer Verlag, Heidelberg, Germany, 1989.

[9] Yan J., MacDonald D., Teletraffic Performance in Intelligent Network Services, The Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks- Proceedings of the 14th International Teletraffic Congress - ITC14, pp 357-366, Antibes Juan-les-pins, France, 1994.

[10] Jensen T., Dimensioning of Intelligent Networks, Proceedings of Telecom 95, pp 561-565, Geneva, Switzerland, 1995.

[11] Petersson S., Arvidsson Å, Network Oriented load Control in Intelligent Networks, Proceedings of APSITT97, Hanoi, Vietnam, March 1997.

[12] Gulyani M., Simulation and Performance Analysis of a Telecommunications System based on Advanced Intelligent Network Architecture, Masters of Engineering Thesis, Dublin City University, July 1993.

[13] Pierce M, Fromm F, Fink F., Impact of the Intelligent Network on the Capacity of Network Elements, IEEE Communications Magazine, pp25-30, December 1988.

[14] MacDonald D., Archambault A., Using Customer Expectation in Planning the Intelligent Network, Proceedings of 14th International Teletraffic Conference, pp 95-104, Antibes Juan-les-pins, France, 1994.

[15] ITU-T, Recommendation E.721, Network grade of service parameters and target values for circuit-switched services in the evolving ISDN, 1991.

[16] Rumsewicz M., Node Control Issues in Evolving Telecommunications Networks, Proceedings of ITC Specialists Seminar on Control in Communications, pp 11-22, Lund, Sweden, September 1996.

[17] Kihl M., Rumsewicz M., Flow Model Analysis of an Intelligent Peripheral Overload Control Strategy, Proceedings of ITC Specialists Seminar on Control in Communications, pp 103-114, Lund, Sweden, September 1996.

[18] Leever P., Vermeer G., Reijmerink R., Franken L., Haverkort B., Performance Evaluation of Intelligent Network Services, Tenth UK Teletraffic Symposium. Performance Engineering in Telecommunication Networks, April 1993.

[19] Kwiatkowski M, Northcote B, Calculating Mean Delays in Intelligent Network Under Overload, Proceedings of Australian Telecommunications Networks and Applications Conference 1994, pp 743-748, Melbourne, Australia, December 1994.

[20] Kühn P., Schopp M., Signalling Networks of ISDN, IN and Mobile Networks - Modelling, Analysis and Overload Control, Proceedings of

ITC Specialists Seminar on Control in Communications, pp 35-48, Lund, Sweden, September 1996.

[21] Bafutto M., Schopp M., Planning the Capacity and Performance of Intelligent Networks based on the ITU-TS IN Capability Set 1, Proceedings of Regional ITC Seminar 95, Pretoria, South Africa, 1995.

[22] Lodge F., Botvich D., Curran T., A Fair Algorithm for Throttling Combined IN and non-IN Traffic at the SSP of the Intelligent Network, IEE Teletraffic Symposium, Manchester, UK, March 1997.

[23] Lodge F., Curran T., A Congestion Control Strategy for Combined IN and non IN Traffic Load at the Service Switching Point of an Intelligent Network, Proceeding of Networks 96, Sydney, Australia, November 1996.

[24] Lodge F., Personal Interview, June 10, 1997.

[25] Newcombe A, Botvich D., Lodge F, Curran T, A Decision Support System for Assurance of Quality of Service in Intelligent Network Service Provisioning, Towards a Pan-European Telecommunication Service Infrastructure, Proceedings of the second International Conference on Intelligent in Broadband Service and Networks, pp 419-431, Aachen, Germany, September 1994.

[26] Arvidsson Å, Petersson S., Angelin L., Congestion Control in Intelligent Networks for Real Time Performance and Profit Optimisation, Proceedings of ITC Specialists Seminar on Control in Communications, pp 347-358, Lund, Sweden, September 1996.

[27] Wohlin C., Nyberg C., Reusable Simulation Models for Performance Analysis of Intelligent Networks, Technical Report, University of Lund, Sweden, 1995.

[28] Becker H., Deuter M., Jell T., Weber W., ETSIN An Evaluation Tool for Value Added Services in Intelligent Networks, Proceedings of 1992 International Zurich Conference on Digital Communications, Intelligent

Networks and their Applications, pp 135-146, Zurich, Switzerland, March 1992.

[29] Galletti M, Grossini F., Performance Simulation of Congestion Control Mechanisms for Intelligent Networks, Proceedings of 1992 International Zurich Conference on Digital Communications, Intelligent Networks and their Applications, pp 391-406, Zurich, Switzerland, March 1992.

[30] Folkestad A., Emstad P., A Token based SDP Load Control Scheme, Proceedings of ITC Specialists Seminar on Control in Communications, pp 151-162, Lund, Sweden, September 1996.

[31] Minetti R. (Ed), Service Architecture, Telecommunications Information Networking Architecture Consortium, 1996.

[32] Gross D., Harris C., Fundamentals of queuing theory, Wiley & sons, New York, 1985.

[33] Kleinrock L., Queuing Systems Volume 1: Theory, Wiley & sons, New York, 1975.

[34] Harrison P., Patel N., Performance Evaluation of Computer Architectures and Networks, Addison-Wesley, Wokingham, UK, c1993.

[35] Gelenbe E., Pujolle G., Introduction to Queuing Networks, Wiley & sons, Chichester, UK, 1987.

[36] Lodge F., Curran T., Gulyani M., Newcombe A., Intelligent Network Congestion Control Strategies and their Impact on User Level quality of Service, Australian Telecommunications Networks and Applications Conference 1994, Melbourne, Australia, December 1994.

8. Glossary of Terms

AIN	Advanced Intelligent Network.
BCMP	Baskett, Chandy, Muntz and Palacios.
BCP	Basic Call Process.
BCSM	Basic Call State Model.
BHCA	Busy Hour Call Attempts.
CCAF	Call Control Agent Function.
CCF	Call Control Function.
CID	Call Instance Data.
CS1	Capability Set One.
DFP	Distributed Functional Plane.
DN	Destination Number.
DP	Detection Point.
FCFS	First Come First Served.
FE	Functional Entity.
FIFO	First In First Out.
FSM	Finite State Machine.
GFP	Global Functional Plane.
GoS	Grade of Service.
GSL	Global Service Logic.
ICR	Impaired Call Rate.
IN	Intelligent Network.
INCM	Intelligent Network Conceptual Model
IP	Intelligent Peripheral.
IS	Infinite Servers.
LCFS	Last come first served.
MSU	Message Signalling Unit.
NE	Network Element.
OBCSM	Originating Basic Call State Model.

OPNET	Optimised Network Engineering Tools.
PCN	Personal Communications Network.
PCS	Personal Communications Services.
PDD	Post Dialling Delay.
PE	Physical Entity.
PIC	Point in Call.
POI	Point of Initiation.
POR	Point of Return.
PP	Physical Plane.
PS	Processor Sharing.
PSTN	Public Switched Telephone Network.
QoS	Quality of Service.
SCEF	Service Creation Environment Function.
SCF	Service Control Function.
SCP	Service Control Point.
SDF	Service Data Function.
SDM	Service Data Management.
SDP	Service Data Point.
SIB	Service Independent Building Block.
SLP	Service Logic Program.
SLPI	Service Logic Program Instance.
SMAF	Service Management Agent Function.
SMF	Service Management Function.
SMS	Service Management System.
SP	Service Plane.
SRF	Service Resource Function.
SSD	Service Support Data.
SSF	Service Switching Function.
SSP	Service Switching Point.
STD	State Transition Diagram.

SVC	Square of the Variation Coefficient.
TINA	Telecommunications Information Networking Architecture.
TBCSM	Terminating Basic Call State Model.
UPT	Universal Personal Telephony.
VPN	Virtual Private Network.
WWW	World-wide Web.