A PROPOSAL FOR THE OSA/PARLAY NETWORK INTERFACE AND ASSOCIATED QOS GUARANTEED NETWORK ARCHITECTURE

Prathaban Vissie Moodley

A thesis submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Doctor of Philosophy.

Johannesburg, 2013

Declaration

I declare that this thesis is my own, unaided work, other than where specifically acknowledged. It is being submitted for the degree of Doctor of Philosophy in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

Signed this ____day of ______________ 2013

Prathaban Vissie Moodley

Abstract

Telcos are adapting their business to address the rapid changing technology landscape (Moodley and van Olst 2011). Telcos require a flexible architecture to allow seamless adaptation and to leverage these new technologies to gain a competitive advantage (Moodley and van Olst 2011). This research is focused on the *transport stratum* as an extension to the OSA/Parlay gateway. The proposed OSA/Parlay Network Architecture and Interface has been designed. The OSA/Parlay Network Interface is characterised by *openness*, *simplicity*, *API* based, *QoS* support and *technology independence*. The OSA/Parlay Network Architecture features *simplicity, technology independence, QoS mechanisms; call admission control; intelligent routing* and supporting both *federation* of telcos and interoperation of *legacy* technologies. The OSA/Parlay Network Architecture and Interface has been demonstrated over a Java based Distributed Processing Environment (DPE) using CORBA. These architectural concepts and principles are demonstrated in a simulated environment and illustrate the Next Generation Network architectural characteristics.

The research contribution therefore achieves an open architecture allowing for 3rd party application developers while also ensuring that call and service requests are provisioned end-to-end with guaranteed application level QoS in the transport network. The OSA/Parlay Network Architecture and Network Interface is synthesised from existing architectural standards and provides the following benefits. It is an extension of the OSA/Parlay standard by including the OSA/Parlay Network Architecture and Network Interface realises the OSA/Parlay next generation network. Both the OSA/Parlay Network Interface and the Network Architecture is specified in a technology agnostic manner. This ensures that the architecture remains future proof as it is not reliant on any particular technology. The long sought after Application level QoS is integrated into the architecture. The periodic network state updates inform the central Connection Coordinator object of both topological network changes as well as current performance of the constituent parts of the network. Intelligent routing of connections is achieved by adapting the Dijkstra algorithm to compute the best path based on dynamic network performance and is tested against QoS requirements, while the call admission control decision naturally allows for load balancing of connection paths within the network. This QoS mechanism achieves the goal of guaranteed QoS for a call admitted into the network.

Acknowledgments

The research project was performed at the Centre for Telecommunications Access and Services (CeTAS) at the University of the Witwatersrand, Johannesburg, South Africa. This centre is funded by Telkom SA Limited, Nokia-Siemens Networks, Vodacom, Telsaf Data and the Department of Trade and Industry's THRIP program. I would like to thank Eskom for their financial support, it is much appreciated.

I also thank God for guidance, protection and blessings. I thank my supervisors both past and present, Prof Hu Hanrahan and Prof Rex van Olst for their support, guidance and assistance during this research project. I thank my colleagues at CeTAS for their valuable inputs during this research. I thank my wife and son, Sherone and Purush, for their love, motivation, support and patience.

Contents

List of Figures

List of Tables

Acronyms

3GPP—Third-Generation Partnership Project

API—Application Programming Interface

ATM – Asynchronous Transfer Mode

BM – Bandwidth Manager

CORBA—Common Object Request Broker Architecture

H.323—suite of protocols for multimedia conferencing for packet-switched networks

IN—Intelligent Network/Networking

IP—Internet Protocol

ISDN—Integrated Services Digital Network

ISP – Internet Service Provider

ISUP — ISDN User Part

INAP— Intelligent Network Application Part

ITU—International Telecommunication Union

IVR—Interactive Voice Response

JCAT—Java Coordination and Transactions

JCC—Java Call Control

JTAPI—Java Telephony API

IMS—IP Multimedia System

MG—Media Gateway

MSC—Message Sequence Chart

MSF—Multiservice Switching Forum

NA— Proposed OSA/Parlay Network Architecture

NI— Proposed OSA/Parlay Network Interface

NRA – Network Resource Architecture

OMG—Object Management Group

OMG IDL—OMG Interface Definition Language

ORB— Object Request Broker

OSA/Parlay—Open Service Architecture Parlay—an open API specified by the Parlay Group to enable secure, public access to core capabilities of telecommunication and data networks

PEG—Protocol Expert Group

PSN— Packet Switched Network

PSTN—Public Switched Telephone Network

QoS—Quality of Service

RM-ODP—Revised Model of Open Distributed Processing

SCF—Service Capability Feature

SDL – Specification and Description Language

SIP—Session Initiation Protocol

SLEE—Service Logic Execution Environment

SA—Service Architecture

TINA—Telecommunications Information Networking Architecture

TINA-C—TINA Consortium

X.25 – A suite of standard ITU protocols for packet switched wide area network (WAN) communication

Chapter 1

Introduction

Telecommunication Operators (telcos) are adapting their business to be flexible in meeting the rapidly changing technology and market landscapes (Moodley and van Olst 2011). The convergence trend between telecommunication and information technologies has contributed towards lower revenue from traditional voice traffic while demand for data bearer traffic is increasing (Hameed et al 2009; Bertrand 2007; Samuel et al 2007; Moodley and Hanrahan 2006). Operators are moving towards competing on value added services attracting premium revenue and increasing bearer traffic to the network. (Moodley and Hanrahan, 2006; Jain et al 2000; Magedanz 2012). In addition, value-added services provide a means of differentiating service offerings from those of competitors (Jain et al 2005, Roj 2012, Hameed et al 2009). Opening of the network to third party service providers leverages the innovation and quick deployment of value-added services capabilities of external parties outside of the telco, while attracting both traffic and revenue to the operator (Beddus and Bruce 2000; Bakker et al 2000; Roj 2012; Blum et al 2008, Hanrahan and Mwansa 2003).

Telcos envision a next generation network (NGN) architecture that is flexible to rapid technology changes as well as the ability to leverage new technologies to gain a competitive advantage (Moodley and van Olst 2011, Roj 2012). The future next generation multiservice network is designed to address this need for flexibility and this consists of a 3-layered architecture: (i) a service platform that facilitates the development of advanced services by both telcos and 3rd party developers; (ii) a softswitch type architecture for call and session control; and (iii) a transport network that guarantees Quality of Service (QoS) (Moodley and Hanrahan 2006 ; Wojick 2013; Knightson 2013). The next generation network (NGN) will be packet based (Moodley and Kennedy 2005; Knightson 2013) and will incorporate multiple transport technologies incorporating QoS capabilities wherein the service functions are independent from the transport network (Knightson 2013; Moodley and Hanrahan 2006).

In recognition of the rapid changing technological environment the International Telecommunications Union (ITU) NGN Framework (Wojick 2013; Knightson 2013) is therefore founded on technology agnostic principles. In addition the framework separates the service stratum from the transport stratum, depicted in Figure 1-1 below. This separation enables a flexible NGN architecture in the creation, maintenance and retirement of value-added services whilst not affecting the transport network. Conversely, this separation allows for a variety of transport technologies to be added, maintained or removed without affecting the operation of the value-added services. In particular the transport stratum consists of QoS assurance functionality referred to as the Resource and Access Control Function (RACF) (Knightson 2013; Lu 2006). Additionally, the framework makes reference to interfaces for the Application Layer, End User Terminals, Management Functions, Federated and Legacy Networks. This framework is used as a reference for discussion of existing architectures and the proposed architecture.

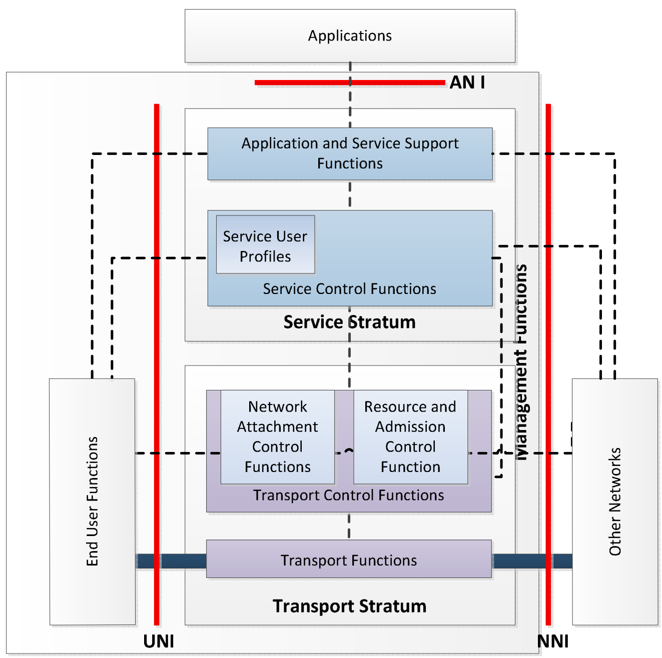

Figure 1-1: ITU NGN Architecture depicting the service and transport strata, adapted from (Knightson 2013)

The Open Services Access / Parlay (OSA/Parlay) (3GPP OSA Part 1 2009; Beddus and Bruce 2000; Bakker et al 2000) is a proposal which addresses the requirements of the service stratum and the interface to the Application layer. A key feature of the OSA/Parlay Gateway is an open, standard, application-programming interface (API). The API provides a level of abstraction in which the application engineer does not need knowledge of the underlying network to develop services (Moodley and Hanrahan 2005). Similarly, the network designer uses general rather than specific service considerations. Another central feature is the migration of intelligence from high-performance centralised hardware such as the traditional Intelligent Network (IN) towards the edges of the network in a Distributed Processing Environment (DPE) (Moodley and Hanrahan 2005).

The OSA/Parlay Gateway is characterised by (Beddus and Bruce 2000; Bakker et al 2000) its open, standard, simplistic and technologically independent design. The simplicity ensures that there are minimal architectural layers in the gateway and thus overhead messages are minimised to support efficiency. In addition, the simple interface that is exposed to the developers ensures that the application developers do not need to have knowledge of the complexity of the network in order to develop advanced services. The open and standardised API mechanism for the interface ensures that the OSA/Parlay Gateway is technologically independent and hence future proof allowing for a multitude of technologies to leverage the API. The interface between service stratum and Application Layer allows for independence and security between the layers. Like most service stratum architectures, the interface to the transport stratum is left as implementation details. The Parlay Group has alluded to the use of protocol adapters for the interface (Moodley and Hanrahan 2005, Hanrahan 2007).

Technology dependent adapters would have a negative effect on the future proof characteristics of the gateway (Moodley and Hanrahan 2005). Due to the rapid technological changes in the telco environment and the need for transport network architectures to interoperate with legacy networks, this interoperation becomes very complex. In particular the transport technologies have proliferated and hence a large number of technology specific protocols must be considered for interoperation, inter alia, INAP, TCP/IP, SIP, H.323, MAP and CAMEL. This ability to interoperate with legacy technologies therefore make network architectural proposals inflexible when considering technology specific protocols and unable to respond to the rapid technological market changes. This inflexibility strongly supports the requirements for a technology independent transport stratum research.

Much of the research efforts have been focused on the service stratum, focusing on aspects such as the Service Orientated Architecture (SOA) and the Service Delivery Platform (SDP), while little attention has been devoted to the transport stratum (Moodley and Hanrahan 2005). “Application provisioning is currently carried out without the cooperation of the lower layers in the transport stratum, which are responsible for the delivery of the data between distributed service entry-points” (Branca 2012). The research efforts in the service stratum have made three important assumptions of the features of the transport network. The service related functionality is clearly separated and independent of the transport related technologies. The QoS functionality is a critical element of the transport network and no specific technologies can characterise the future next generation network (Knightson 2013; Lu 2006; Wojick 2013).

This research is focused on the *transport stratum* as an extension to the OSA/Parlay gateway; we refer to this as the OSA/Parlay Network Architecture. Key features of the OSA/Parlay Network Architecture (Moodley and Hanrahan 2008) are the provisioning of end-to-end transport resources in response to the service request for a particular session. Another key feature is the Call Admission Control (CAC) function which ensures that a new call request admitted into the network is guaranteed a level of quality of service (QoS) for the entire call duration; whilst ensuring that existing calls already admitted into the network do not experience any degradation in QoS. The OSA/Parlay Network Interface located between the call and session control layer (*service stratum*) and the OSA/Parlay Network Architecture (*transport stratum*) should be characterised as an open, standard and technologically independent interface. Similar to the OSA/Parlay Gateway, the OSA/Parlay Network Interface will be API based and defined in the Object Oriented paradigm using Unified Modelling Language (UML). Initiatives such as the Telecommunication Intelligent Network Architecture (TINA), Open Service Access / Parlay (OSA/Parlay) and the Multiservice Switching Forum (MSF) provide the foundation of this research.

A number of next generation network architectures exist and are discussed in Chapter 3. However these network architectures have not been developed for the OSA/Parlay Gateway and none of the architectural proposals address a technology agnostic network architecture apart from the legacy TINA Network Resource Architecture (NRA). Therefore this research makes a proposal for the network architecture together with a network interface suitable for a technology agnostic implementation for the OSA/Parlay architecture.

1.1 Research Objectives

The objective of this research is to develop an OSA/Parlay compliant network architecture together with the network interface. The OSA/Parlay network architecture and interface will be technology agnostic ensuring any legacy or future transport technology to be adopted without affecting both the service and transport strata.

A proof of concept is required to demonstrate the principles and concepts of the network architecture and the network interface characteristics.

The design of the future flexible network architecture must explicitly engineer the network for provisioning QoS guaranteed converged services. "The network architecture is not a detailed definition. Rather it provides the concepts and principles for connections involving the constituent networks (packet and circuit switched) and single elements (media gateways and IVRs)" (Moodley and Hanrahan 2008). The architecture must encompass the following characteristics (Moodley and Hanrahan 2008):

- "be designed as a *simple* architecture;
- demonstrate *technology independence*;
- support *open interfaces*;
- enforce an integrated approach for *end-to-end connections*;
- implement connection admission *QoS* control as part of the overall QoS assurance system;
- provision *intelligent routing* of connections;
- support *federation* between network operators to provide QoS-assured end-to-end connections;
- support *legacy* technologies in the overall architecture."

Scope of research

This research is intended to present the OSA/Parlay based Network Architecture and Network Interface. The following limit the scope of research into this topic:

- The network architecture is not a detailed definition; rather it provides the concepts and principles for connections involving the constituent networks (packet and circuit switched) and single elements (media gateways and IVRs) (Moodley and Hanrahan 2008).
- The functionality designed in the Network Architecture is dependent on the OSA/Parlay Gateway, the Consumer Domain and Transport Network Technologies and these elements are emulated.
- The interface between the OSA/Parlay Gateway and the OSA/Parlay Network Architecture is assumed to be within a trusted domain of the telecommunications operator and retailer.

- Security of the Network Architecture and associated Interface is not considered in the scope of this research.
- Authentication, authorisation and accounting (AAA) of the Network Architecture is not considered.
- Implementation and performance of the OSA/Network Architecture requires access to a telecommunications network and gateway and could not be explored in this research. However the Proof of Concept emulated both the gateway and the telecommunications network to illustrate the principles and concepts.

1.2 Thesis Outline

This thesis is organised as follows:

Chapter 1: The requirements for the research and the objectives are defined.

Chapter 2: In this chapter, the key research questions are reviewed. This includes the implementation of the OSA/Parlay Architecture; Requirements for the Network Architecture and Interface; Key Considerations; NGN Characteristics and guaranteed QoS.

Chapter 3: In this chapter, a review of various architectural standards is considered with respect to the network architecture and interface. The useful features of each transport network and network interface are identified for the inclusion into the OSA/Parlay Network Architecture and Interface.

Chapter 4: This chapter describes the main contribution of this research, the OSA/Parlay Network Architecture together with the detailed interface and methods.

Chapter 5: A use case approach is adopted to illustrate the characteristics of the OSA/Parlay Network Architecture. The use cases are implemented in a proof of concept and the message sequence charts detail the operation of the various components in the architecture and its distributed nature.

Chapter 6: In this chapter, the results of the OSA/Parlay Network Architecture is summarised, recommendations and potential future work is outlined.

Appendix A: Papers that resulted from this research

Appendix B: Basic Use Cases illustrated as Message Sequence Charts

The source code for the proof of concept implementation is provided on a separate removable storage disc.

Chapter 2

Key Research Questions

Convergence between telecommunications and information technology has resulted in a rapid development of the technology landscape. These new technologies will contribute towards the achievement of the future multiservice network which supports data, voice and multimedia. “A key feature of these new technologies is an open, standard, application-programming interface (API). The API provides a level of abstraction in which the application engineer does not need knowledge of the underlying network to develop services. Similarly, the network designer uses general rather than specific service considerations. Another central feature is the migration of intelligence from high-performance centralised hardware such as the traditional Intelligent Network (IN) towards the edges of the network in a Distributed Processing Environment (DPE).” (Moodley and Hanrahan 2005)

The OSA/Parlay Gateway is founded on these new technologies and provides a point of integration between Next Generation Network (NGN) application services and the underlying Call and Session Control infrastructure (*service stratum*). The OSA/Parlay Gateway provides for 3rd Party intelligent application developers the ability to provide services using the telco connectivity by exposing an application interface (Beddus and Bruce 2000; Bakker et al 2000; Moodley and Hanrahan 2006). The OSA/Parlay standard assumes a three-layered architecture, consisting of the transport network layer, the service control layer (the OSA/Parlay Gateway) and the application and services layer (Moodley and Hanrahan 2005; Moodley and Hanrahan 2006) as depicted in Figure 2-1. The Application and Services Layer is independent of the Service Control Layer (OSA/Parlay Gateway), and is accessed through an open, standard and secure API (Moodley and Hanrahan 2005; Moodley and Hanrahan 2006). The gateway allows for a multitude of service control functions such as generic call control; multiparty call control; multimedia call control; user interaction and many others (3GPP OSA Part 4-2 2009; 3GPP OSA Part 4-3 2009; 3GPP OSA Part 4-4 2009; 3GPP OSA Part 5 2009). Current research efforts focus on development of applications and services using the application interface. However, the interface between the OSA/Parlay Gateway and the Transport network, i.e.

the OSA/Parlay Network Interface has not received the same attention (Moodley and Hanrahan 2005).

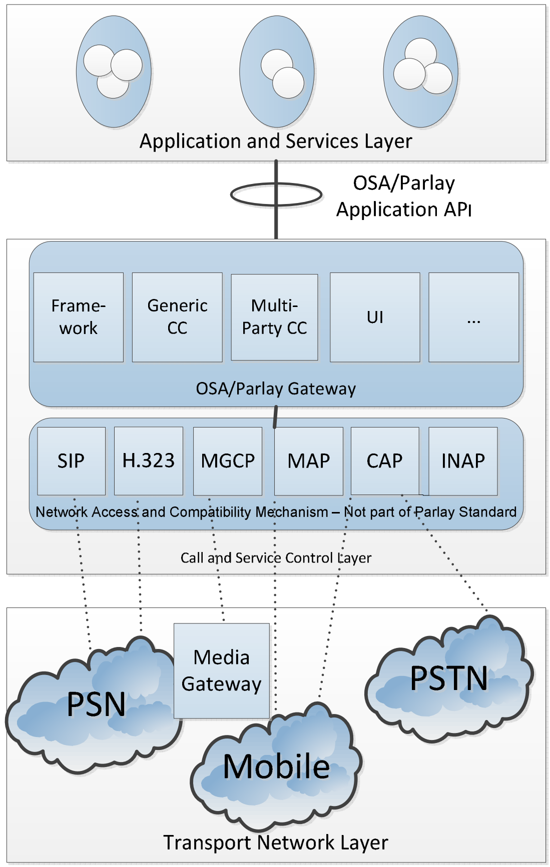

Figure 2-1: OSA/Parlay Architecture, adapted from (3GPP OSA Part 1 2009)

The Parlay standard does not specify the interface between the OSA/Parlay Gateway and the transport network (Moodley and Hanrahan 2005). The OSA/Parlay Network Interface envisaged in this work relies on the single point of communication between the OSA/Parlay Gateway and the network. The underlying OSA/Parlay Network Architecture does not rely on the traditional call/session signalling. A more powerful call admission, connection co-ordination, network federation and end-to-end Quality of Service mechanisms are incorporated. (Moodley and Hanrahan 2006)

The key research questions are:

- Can the OSA/Parlay gateway be extended to the transport stratum using similar concepts as the gateway?

- Can the proposed OSA/Parlay Network Interface be used to separate the Call and Service Control functions from the Transport Network be specified using an open, standard and technology agnostic API supporting QoS?
- Can the proposed OSA/Parlay Network Architecture be specified in a technology agnostic manner, allowing it to be future proof and applicable to rapidly evolving transport technologies?
- Can the Call Admission Control (CAC) function to be designed as an integral part of the Network Architecture to ensure guaranteed QoS for all calls within the network and calls still to be admitted into the network?
- Can the proposed OSA/Parlay Network Architecture be implemented with functionality to integrate to federated telco networks as well as legacy networks, which are protocol specific?
- Can an alternate more powerful routing concept be integrated as a function of the proposed OSA/Parlay Network Architecture?

2.1 Current Implementation of the OSA/Parlay Gateway

The fact that the OSA/Parlay Gateway has not been widely adopted as the standard has left the interface to the network architecture undefined and intended as implementation details. The complexity required to adapt to all transport network technologies limit the full adoption of the OSA/Parlay Gateway. The current incarnations of the OSA/Parlay gateway have been limited to the Third Generation Partnership Project Long Term Evolution (3GPP LTE) and Multiservice Switching Forum (MSF) architectures (3GPP IMS 2012; MSF 2006).

These incarnations adopt a single protocol adaptation in order to simplify the implementation; these details are presented in sections 2.1.1 and 2.1.2. The rich set of service control features are never used as the service details are merely passed onto the implementation protocol through the protocol adaptation; while separate 3GPP or IMS architectural functions provide the service control functionality. Thus the OSA/Parlay Gateway merely exposes the application interface to provision an open network to 3rd party application developers.

2.1.1 OSA/Parlay in the Third Generation Partnership Project (3GPP)

The Long Term Evolution (LTE) architecture (3GPP IMS 2012) has become the dominant vision for mobile networks. The architecture is based on proposals for the Radio Access Network (RAN) and the Evolved Packet Core (EPC). We focus on the EPC, the proposal of LTE is to make use of an all packet core network together with call and session control based on IMS components, see Figure 2-2. A call in the LTE network is orchestrated by the IMS components and centrally by the Serving Call Session Controller (S-CSCF) using the SIP protocol.

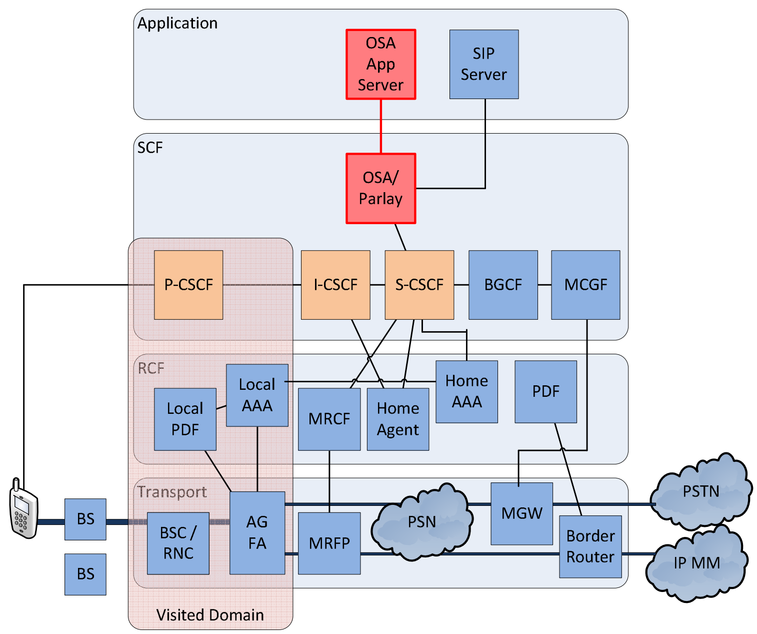

Figure 2-2: 3GPP LTE Network Architecture, adapted from (3GPP IMS 2012)

The architecture incorporates the OSA/Parlay gateway (3GPP IMS 2012), which is interfaced through the S-CSCF function. The gateway presents the open application interface allowing for the 3rd party application developers to leverage the network resources to deliver value-added services. Given the location of the gateway in the architecture, the gateway has to adapt to the Session Initiation Protocol (SIP), to deliver applications using the S-CSCF. This approach simplifies the adaptation of the OSA/Parlay gateway to the network resources, however, it is technology dependent.

The OSA/Parlay gateway interfaces at the ITU service stratum level rather than at the transport stratum. The call and session control features of the gateway are not leveraged as these functions are relied upon by the IMS components. The IMS components of the architecture produce a significant number of overhead messages already leading to inefficiencies (see section 3.3). The incorporation of the OSA/Parlay gateway will add to the overall overheads and further leads to inefficiencies. The architecture is dependent on the underlying transport network technology and the SIP protocol in the Service Control Layer which has a finite lifecycle and thus not future proof.

2.1.2 OSA/Parlay in the Multiservice Switching Forum (MSF)

The Multiservice Switching Forum (MSF) architecture (MSF 2006) is a proposal for the next generation network. The focus of the analysis is on the transport layer; service and control layer and the application layer, see Figure 2-3. Similar to the 3GPP LTE architecture, the core network is based on packet switching technology and incorporates RACF type function for QoS guarantee and admission control. Additionally, the call and session layer incorporates the IMS components to orchestrate calls and connection in the transport network using the SIP protocol.

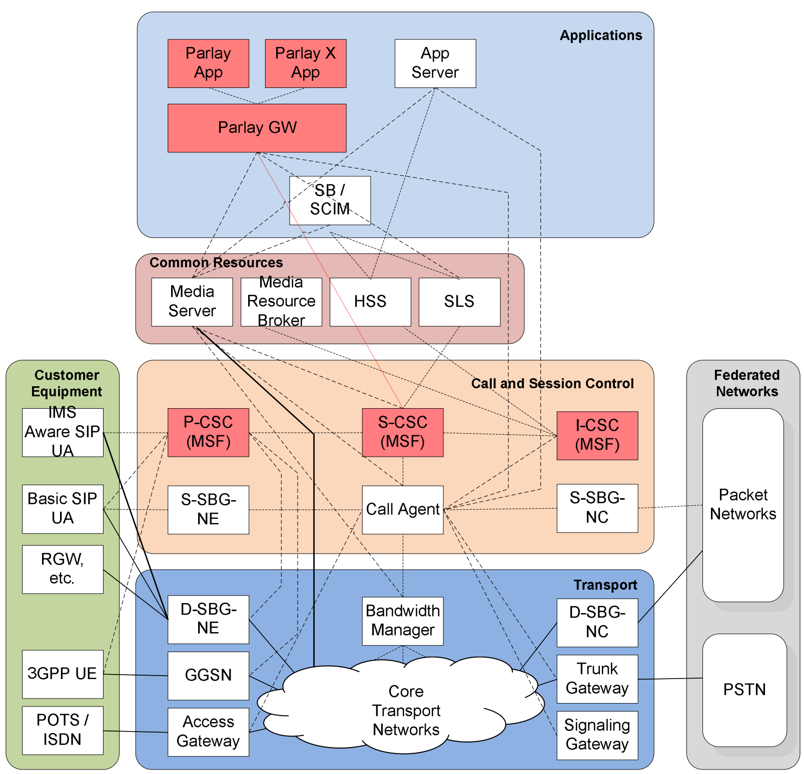

Figure 2-3: MSF Network Architecture, adapted from (MSF 2006)

Similar to the LTE architecture, the MSF architecture (MSF 2006) incorporates the OSA/Parlay gateway, which is interfaced through the S-CSCF function. The gateway presents the open application development interface allowing for the 3rd party application developers to leverage the network resources to deliver value-added services. Similar to the LTE architecture, the gateway has to adapt to the SIP protocol, to deliver applications using the S-CSCF. This approach simplifies the adaptation of the OSA/Parlay gateway to the network resources, however, it is technology dependent.

The call and session control features of the gateway are not leveraged as these functionalities are relied upon by the IMS components. Similarly, the incorporation of the OSA/Parlay gateway adds to the overall overheads and further leads to inefficiencies. The architecture is dependent on the underlying transport network technology and the SIP protocol in the Service Control Layer which has a finite lifecycle and thus not future proof.

2.2 Network Architecture and Interface Requirements

The existing incarnations of the OSA/Parlay gateway have many short comings (see section 2.1) and do not leverage the full features of the gateway as the standard has intended. The proposal of this thesis is to extend the OSA/Parlay standard and to nominate the transport stratum elements which are the OSA/Parlay Network Architecture and the Network Interface (see Figure 2-4). The Network Architecture should encompass the following characteristics:

- *openness*,
- *simplicity*,
- *future-proof*,
- *QoS mechanisms* and
- supporting *federation* between telcos.

"The network architecture does not rely on call/session signalling. More powerful call admission, connection co-ordination, network federation and end-to-end Quality of Service mechanisms are incorporated." (Moodley and Hanrahan 2006). The design requirements for the Network Interface and Network Architecture are synthesised in the remainder of the chapter.

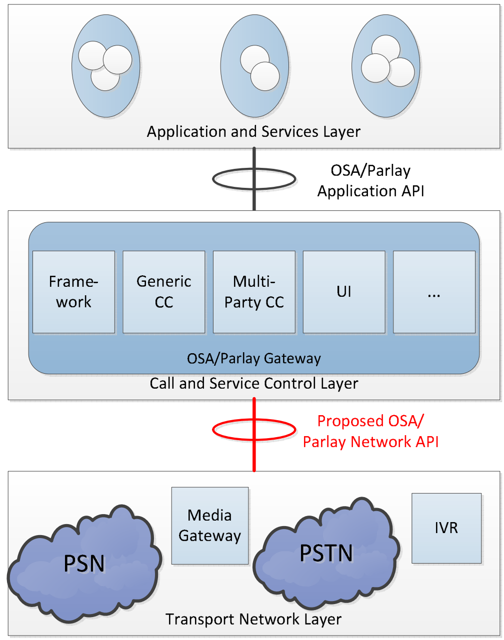

Figure 2-4: Conceptual OSA/Parlay Architecture, adapted from (3GPP OSA Part 1 2009)

2.3 Network Interface Requirements

The OSA/Parlay Network Interface is located between the Services and Control Layer and the Transport Layer (see Figure 2-4). Similar to the OSA/Parlay Application interface, the Network Interface should be specified in an open and technology agnostic manner. The requirements on the Network Interface are as follows and must (Moodley and Hanrahan 2006; Knightson 2013; Lu 2006; Wojick 2013):

- be defined in a *technology-independent* form;
- be suitable for adoption as an *open API*;
- support *end-to-end connections* as well as those to IVRs while hiding the detail of the actual network;
- support *quality of service*;

The network interface is envisaged to be API based rather than using a protocol approach. The API mechanism easily addresses the requirements for *openness, simplicity* and *technology independence* (Moodley and Hanrahan 2006).

2.4 Network Architectural Requirements

"The OSA/Parlay Network Architecture is not a detailed definition, it provides the concepts and principles for connections involving the constituent technologies (packet-switched and circuit-switched) and special resources (media gateways and IVRS). Within the network architecture there are functional entities for connection admission, routing, federation and QoS" (Moodley and Hanrahan 2008). The network architecture must (Moodley and Hanrahan 2008):

- provide *connection admission control* as part of a *QoS assurance system*;
- support connections in *layer* transport networks;
- support *routing* of connections;
- support *federation* between network operators to provide QoS-assured end-to-end connections.

2.4.1 Switched Circuit Network

Switched circuit technology is common in PSTN based telcos. The OSA/Parlay Network architecture must provision interoperation with the PSTN switched circuit technology. Investment decisions will dictate that the return of investment of technology has been achieved before technology migration is considered. Switched circuit technology is inherently characterised by guaranteed QoS due to the nature of the technology. Thus legacy adaptation to the switched circuit technology ensures a guaranteed level of QoS and adaptation can leverage existing technology protocols for legacy support. (Moodley and Hanrahan 2008)

2.4.2 Packet Switched Network

Packet switched technology will be the dominant transport technology of the future multiservice network (Moodley and Kennedy 2005). However, QoS mechanisms must be explicitly incorporated into the Architecture. The QoS mechanism must guarantee a level of service for the entire duration of a connection while ensuring the existing connection already admitted into the network are not adversely affected. The call admission control function is an integral component of the QoS system to make the admit or reject decision. (Moodley and Hanrahan 2008)

2.4.3 Media Gateway

The ability of the OSA/Network Architecture to address interoperability with legacy technologies such as the PSTN to newer technologies such as SIP is a key requirement. Thus the Media Gateway is included as an integral component of the architecture. (Moodley and Hanrahan 2008)

2.4.4 Interactive Voice Response

The interactive voice response (IVR) is a common special resource deployed in telecommunication networks. It is an essential part of services such as prepayment and call center services. Due to the essential nature, the Network Architecture needs to cater for interworking with this resource. (Moodley and Hanrahan 2008)

2.4.5 Quality of Service

"Existing implementations of QoS based systems rely on network-level mechanisms to negotiate and agree on parameters affecting quality of service, for example using the session description carried in SIP messages. Application-related quality of service is however an ultimate goal. The following steps are required. The application requests the required QoS using parameters that are suitably abstracted from the underlying infrastructure. The service provider specifies the maximum QoS that it offers. This level may be specified in policies and may depend on the state of the network. The service provider and application agree on a quality. At the network level, end-to-end interactions may be necessary to determine the QoS available. If the requested quality cannot be provided, the request may be refused, for example a call request is not admitted. The underlying network mechanism must record the QoS achieved for billing and management purposes. The Network Architecture must therefore encapsulate capabilities to query the state of the Transport Network and send connection control requests to the Transport Network. The capabilities of the Network Architecture will include QoS and network-level federation queries and requests." (Moodley and Hanrahan 2008)

2.4.6 Federation

Telcos federate networks in order to provide an effective service to their users. The frequency of this requirement to federate local networks to enable consumers to contract each other is high. Federation feature is also relied upon for international calls which require an interworking of different operators networks. (Moodley and Hanrahan 2008; Knightson 2013; Lu 2006; Wojick 2013)

2.4.7 Simplicity

For the Network Architecture and Interface to be adopted, the level of complexity must be reduced and the simplicity principle should be adopted. This will ensure efficiency of the architecture and reduction of overhead messages. The network interface and architecture should be designed in a manner that will ensure (Moodley and Hanrahan 2008; Knightson 2013; Lu 2006; Wojick 2013):

- Minimal number of architectural layers;
- Decisions to be made at the lowest layer possible (efficiency);
- Tree structured architecture.

2.5 Design Methodology

The Reference Model for Open Distributed Processing (RM-ODP) methodology (RM-ODP 1995) is used for designing complex architectures and interfaces. A framework is developed for the Network Architecture and Network Interface. The framework ensures that the interface will consist of the following characteristics: *openness*, *simplicity*, *future-proof*, *QoS mechanisms* and supporting *federation* between telcos. This methodology ensures that the Network Architecture and Interface to be specified will be technology agnostic. Various tools are used within the RM-ODP methodology to complete the RM-ODP views, these will include SDL, MSC, ICONIX and UML amongst others.

The OSA/Parlay Network Architecture and Interface was developed and coded in the Java NetBeans 7 Integrated Development Environment (IDE). It has been demonstrated over a Java based Distributed Processing Environment (DPE) using CORBA in both the Microsoft Windows® and Ubuntu™ UNIX based operating environments. The call admission control concepts and principles were demonstrated in a simulated environment using ns-2 (UCB et al 2012) by applying the add-on MPLS CR-LDP and RSVP-TE module (Boeringer 2006) and various traffic generation agents. The results from the simulation studies were analysed in MATLAB™.

2.6 Chapter Summary

The OSA/Parlay Network Interface envisaged in this work operates as a single point of communication between the OSA/Parlay gateway and the transport network. The underlying network architecture however does not rely on call/session signalling. More powerful call admission, connection co-ordination, network federation and end-to-end Quality of Service mechanisms are incorporated. The Architectural design principles is presented and include *openness*, *simplicity*, *future-proof*, *QoS mechanisms* and supporting *federation* between telcos.

Chapter 3

NGN Architectures

Telcos have made significant investments in silo orientated architectures to address their respective market and service offerings. To this extent telcos have a large deployment of traditional telecommunication technologies, such as PSTN exchanges, which are deployed to address the traditional voice needs of the market. Similarly, the data providers such as Internet Service Providers (ISP) have traditionally adopted packet switched technologies such as ATM, X.25 and IP to address their traditional data service offerings. In order to differentiate telco service offerings from competitors, telcos have adopted the Intelligent Network (IN) technology to provide customised service offerings to their markets.

However the need for new business opportunities in the telco industry has driven the development of new services and addressing new markets by leveraging on the opportunities provided by technology convergence. A good example of convergence in the traditional PSTN is the adaptation of the technology to transport data traffic as opposed to voice traffic over the network. Convergence is observed in a number of areas in the telecommunication technologies such as terminals, access technologies, core network and services environments. The ultimate trend of convergence is the breakdown of the technology silos to offer new products and services.

This trend of converging technologies of the telecommunications, information technology and broadcasting paradigms have led to the vision of the Next Generation Network (NGN). “A Next Generation Network is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.” (Knightson 2013).

In this chapter the background to existing network architectures and interfaces are presented. These architectures and interfaces are reviewed for the applicability to the research objects and the key research questions as described in section 1.1 and Chapter 2 respectively. The message sequence chart for a simplified 2-party call is used to analyse existing architectural proposals in sections 3.1.4; 3.2.1 and 3.3.1. Useful concepts and principles are incorporated and adapted for the design of the OSA/Parlay Network Architecture and Network Interface in Chapter 4.

3.1 Telecommunications Information Networking Architecture (TINA)

TINA (TINA Overall 1997; Hanrahan 2007) was the one of the first proposals that met the NGN architectural requirements which addressed a multiservice network transporting voice and data traffic and possibly video traffic. TINA was founded on the principles of a scalable telecommunication architecture that would be future proof and provide flexibility for a technology agnostic future. The TINA architecture included characteristics of an open interface, allowing 3rd party service providers to provision advanced services using telco resources. Other characteristics include the principles of creating a multivendor environment for intelligent terminals as well as supporting an environment with multiple transport network technologies.

The TINA standard was a detailed definition and the overall architecture (TINA Overall 1997) was decomposed into four sub architectures:

- The **Service Architecture** defines the architectural components and principles used to design, deploy and manage advanced services. (TINA SA 1997)
- The **Network Architecture** defines concepts and principles to design, deploy and manage both transport and switching technologies in the transport network. (TINA NRA 1997; TINA NRIM 1997)
- The **Management Architecture** specifies the concepts and principles used to design management software used to manage the network technology, resources and the services on the TINA architecture. (TINA MA 1994)
- The **Computing Architecture** specifies concepts and principles for the design of distributed software. (TINA CA 1996)

The principles of technology independence were factored into the architecture. Other key principles for the architecture are the ability to reuse software as well as dealing with processing complexity using the distributed processing methodology. TINA is based on the object orientation paradigm and follows the Revised Model of Open Distributed Processing (RM-ODP) methodology for designing a complex architecture (TINA NRA 1997; TINA SA 1997).

3.1.1 TINA Business Model and Reference Points

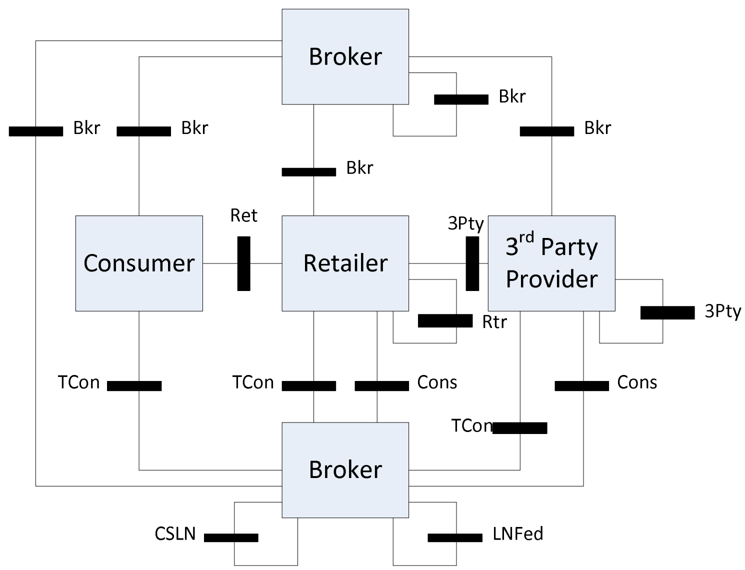

Figure 3-1: TINA Business Model, adapted from (TINA BM 1997)

TINA was the first telecommunication architecture that introduced the concept of a business model (TINA BM 1997; Hanrahan 2007) with a detailed description of the interaction with stakeholders, as shown in Figure 3-1. The consumer is a party that received services from the other stakeholders in TINA and is the source of revenue for other stakeholders. The retailer is the single point of contact for the consumer. The Retailer registers the consumer for services which it can provide itself or through a 3rd party service provider. Since the retailer is the single point of entry into the TINA system, it functions as the entity which controls access into the TINA system. The 3rd Party Provider provides services on request by the Retailer, the retailer requests these services on behalf of the Consumer. The Connectivity Provider is an infrastructure owner and provides the physical connections between the consumer, the retailer and 3rd party providers. The function of the Broker is to provide all the stakeholders with information on what services are available from the different stakeholders.

3.1.2 TINA NRA: Information Viewpoint

The information viewpoint in TINA (TINA NRIM 1997) relies on the concept of sessions as a guiding principle. A session is a temporary allocation of resources that are configured to perform a service task for a finite period of time (TINA NRA 1997). Once the task is complete the resources are made free and reallocated to another unrelated task. The information viewpoint is organised into two broad categories of sessions: i. Communication Session and ii. the Connectivity Session.

The **Communication Session** in essence is a view from the Network Resource Architectures (NRA's) perspective on the logical representation of the service within the Network Architecture. This session serves as an environment to establish, manage and release of a service requested from the TINA Service Architecture. The Logical Connection Graph provides a detailed information model of the communication session and is used to describe the relationship between the communication session and its clients (the Service Architecture). (TINA NRA 1997)

The **Connectivity Session** in its basic form is a view from the NRA's perspective on the physical representation of the technology independent view of the network connections within the Architecture. This session serves as an environment to establish, manage and release network connections requested from the Communication Session. The Physical Connection Graph provides a detailed information model of the connectivity session and is used to describe the relationship between the connectivity session and the Communication Session. (TINA NRA 1997)

3.1.3 TINA Network Resource Architecture

The TINA Network Resource Architecture (NRA) (TINA NRA 1997) is the focus of this analysis; it defines a set of concepts and principles that is used to describe the Transport Network in a technology agnostic manner. The TINA NRA is responsible for setting up; maintaining and releasing transport connections carrying network traffic (see Figure 3-2). The network traffic is assumed to be heterogeneous in nature, thus Quality of Service (QoS) characteristics in terms of end-to-end latency, jitter and data loss will differ.

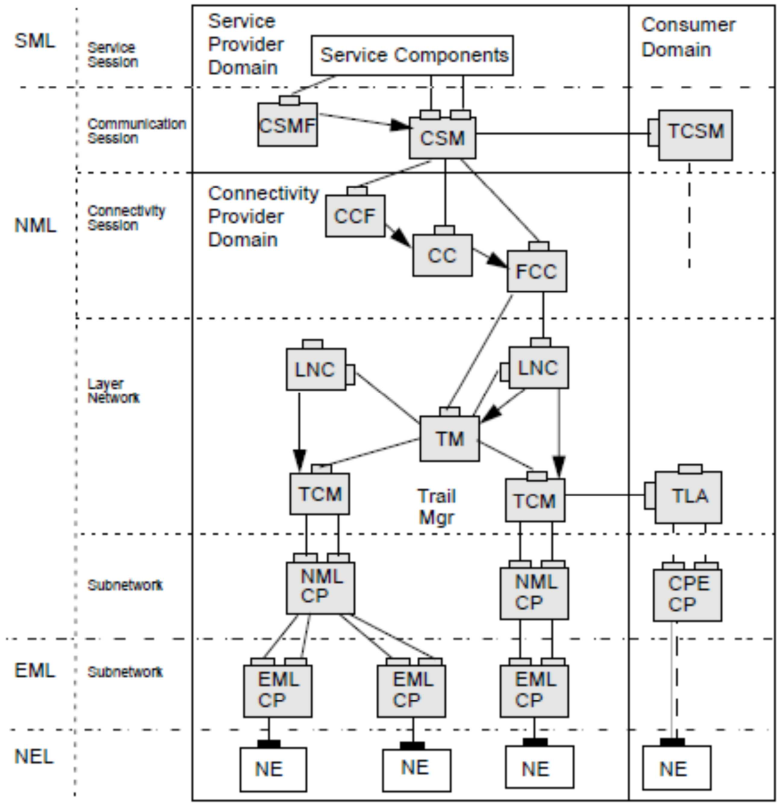

Figure 3-2: TINA Network Resource Architecture, sourced from (TINA NRA 1997)

The Communication Session Manager (CSM) is the single point interface to the TINA Service Architecture. The CSM exposes an interface allowing the service architecture to establish, maintain, modify and release logical connections, termed as stream flow connections. The CSM manages the communication session described in section 3.1.2 and thus maintains a Logical view of the connection in the network. A Communication Session Manager Factory (CSMF) serves as a factory object to coordinate the establishment of CSMs. The Terminal Communication Session Manager (TCSM) is responsible for establishing the nodal part of a connection in the consumer domain. (TINA NRA 1997)

The connection coordinator (CC) is created to support the CSM in realising the physical connections related to the logical connection represented by the CSM, these connections are termed network flow connections. The CC is responsible for the setup, maintenance, modification and release of nodal connections in the network using the Layer Network Coordinator (LNC) described later. The CC manages the connectivity session described in section 3.1.2 and thus maintains a Physical view of the connection in the network. The CC hides the complexity details of the various layer transport networks from the CSM while also determining which layer transport network is to be used to provision the network resources for the service request. Similar to the CSMF object, the Connection Coordinator Factory (CCF) serves as the factory object for creating connectivity sessions. A number of network flow connection may be required to support the service, therefore TINA specified a Flow Connection Controller (FCC) object which controls and manages each network flow instantiated in the network. (TINA NRA 1997)

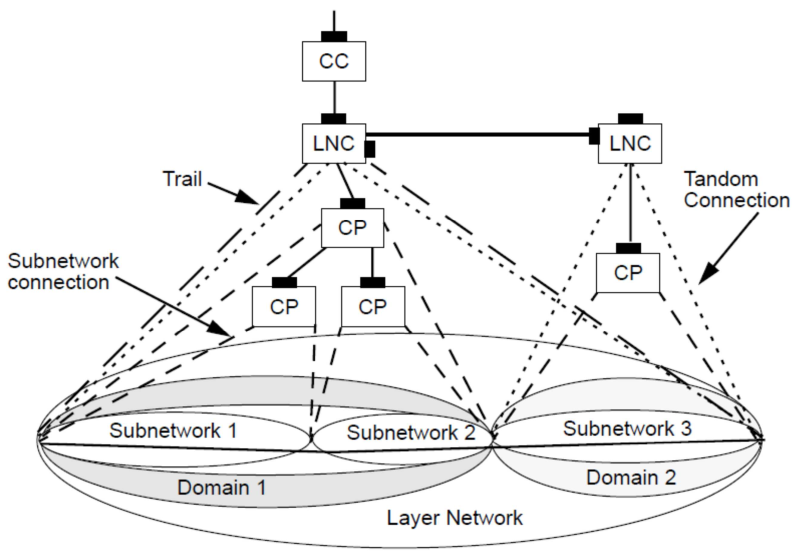

Figure 3-3: TINA organising of Trails and Tandem Connections, sourced from (TINA NRA 1997)

The transport network is constituted by a number of layers, thus the Layer Network Coordinator (LNC) controls and manages each layer in the transport network. The LNC's role is similar to the role of the CC at the connectivity level, except that instead of creating connections with a session, it establishes physical connections in relation to a real network. At this layer in the TINA NRA, the architectural objects are not session orientated, rather these objects exist for the duration of a physical connection. The LNC allows trails and tandem connections to be established or released within layer transport network, see Figure 3-3. As the LNC and its components are not session orientated, there are a large number of trails. A trail is a network connection established across a layer transport network and a tandem is established at the interface of various domains. The LNC supports queries on the status of the layer transport network and its associated trails. A Trail Manager (TM) is created by the Layer Network Coordinator (LNC) of the layer transport network domain in which this trail is initially established. An instance of this object type exists for each trail within the layer transport network. A

Tandem Connection Manager (TCM) is created by the Layer Network Coordinator (LNC) of the layer transport network domain in which this tandem connection exists. One instance of this object type exists for each tandem connection within a layer transport network domain. The Terminal Layer Adapter (TLA) object provides functions for creating, manipulating and deleting network flow endpoints in the consumer domain. The LNC, TM and TLA are technology dependent a must support the technology specific details. (TINA NRA 1997)

The Connection Performer (CP) represents and interfaces to a particular transport network technology node. The CP is used to establish the technology specific network setup, modification and termination of connections. (TINA NRA 1997)

Level	Environment Concept	Information Model	Associated Computational Objects
Communication	Communication session	Logical connection graph	CSM, TCSM, CSMF
Connection	Connectivity session	Physical connection graph	CC, FCC, CCF
Layer Transport Network	Network domain	Trail, tandem connection	LNC, TM, TCM
Subnetwork	Subnetwork element	Subnetwork connection	CP

Table 3-1: Summary of concepts, models and computational objects in TINA, adapted from (TINA NRA 1997)

The summary of the TINA concepts, models and computation objects are presented on Table 3-1.

3.1.4 TINA Use Case: 2-party call

The typical call setup procedure of the TINA NRA is described in this section. The details of the message sequence chart will be used to analyse some of the characteristics of the architecture (Yun and Perros 2010; Moodley and Hanrahan 2008). The terminals are within the same administrative domain, the network entities conform to the architecture defined in TINA NRA.

Use Case 1: Setup of the Communication and Connectivity Sessions (TINA NRA 1997)

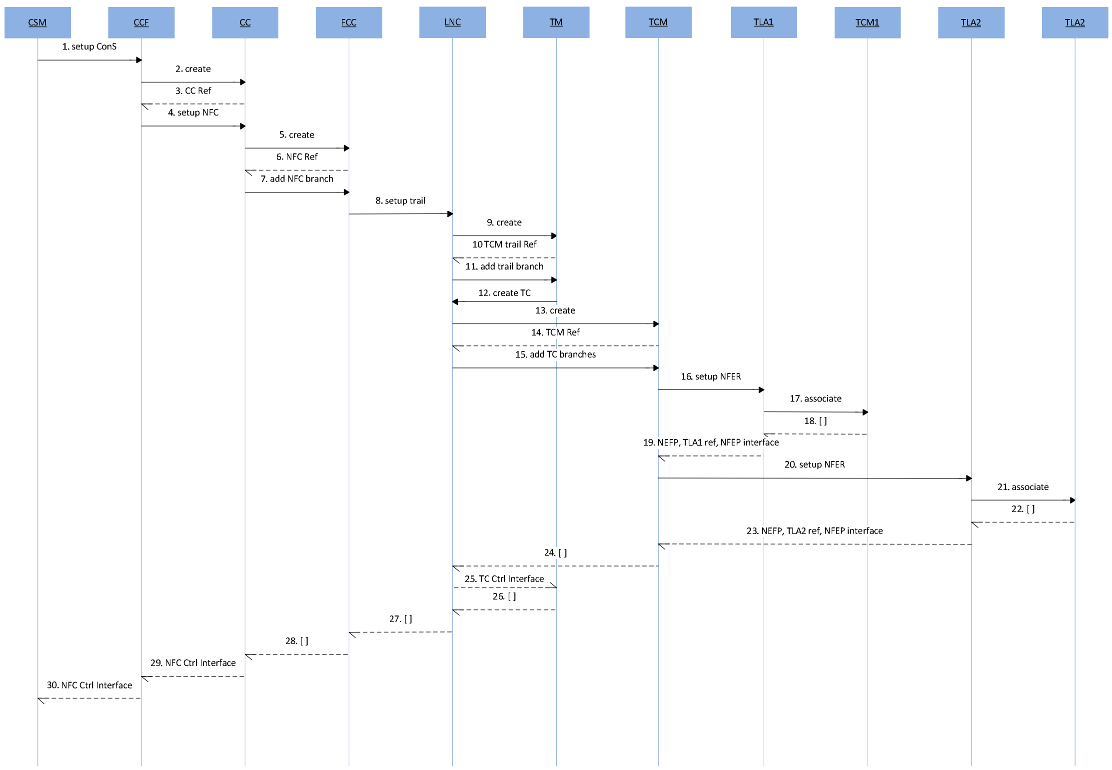

Figure 3-4: TINA 2-party call setup – message sequence chart, adapted from (TINA NRA 1997)

The following message sequence chart is presented as discussed in the TINA NRA standard (TINA NRA 1997)

1. “CSM requests the CCF to setup a new connectivity session.
2. CCF creates a new CC for the new connectivity session; although it is shown in the diagram this is really a DPE operation.
3. CCF is informed of the creation and returned a reference to a connectivity session control interface of CC.
4. For each network flow connection in the connectivity session CCF requests from CC its setup.
5. To fulfil this request CC creates an FCC.
6. A reference to a network flow connection control interface of FCC is returned to CC.
7. CC requests FCC to add branches to the new network flow connection.

8. FCC chooses the necessary layer transport networks for the new network flow connection. This scenario assumes only one layer transport network is necessary. FCC locates the LNC for this layer transport network and request from it a trail setup.
9. LNC creates a TM to manage the new trail. This is again a DPE operation.
10. A reference to a trail control interface of the new TM is returned to LNC upon its instantiation.
11. LNC requests TM to add branches to the new trail.
12. To attend this request TM requests the creation of a tandem connection to LNC.
13. LNC creates a TCM for the new tandem connection (DPE operation).
14. LNC gets a reference to the tandem connection control interface of the new TCM.
15. LNC requests TCM to add branches to the new tandem connection.
16. TCM requests the setup of NFEPs from the TLA in consumer1's domain, including in the request the unique correlation identifier corresponding to the network part of the connection to which the TFC in consumer1's terminal is associated.
17. TLA1 selects or creates (depending on whether they are pre-provisioned or not) the corresponding NFEPs and requests from TCSM1 the association of the TFC, specified by the correlation identifier, with the established NFEPs.
18. TCSM1 performs the association requested, which completes the connection.
19. TLA1 returns to TCM the name of the chosen (and established) NFEP(s) and a reference to its NFEP control interface.
20. Now TCM requests the NFEP setup from TLA2.
21. TLA2 performs the NFEP creation or selection and requests the completion of the connection to TCSM2.
22. TCSM2 performs the association that completes the connection.
23. TLA2 returns to TCM the name of the chosen (and established) NFEP(s) and a reference to its NFEP control interface.
24. Return of Step 15.
25. Return of Step 12, that includes a reference to a tandem connection control interface of TCM.
26. Return of Step 11.
27. Return of Step 8, that includes a reference to a trail control interface of TM.
28. Return of Step 7.
29. Return of Step 4, that includes a reference to a network flow connection control interface of FCC.
30. Return of Step 1, that forwards the previous reference to a network flow connection control interface of FCC."

3.1.5 Analysis against Architectural Requirements

A simple use case for call setup within the TINA NRA based architecture is presented. The architectural characteristics are ascertained based on the characteristics of the control messages used to setup the connections.

A. Simplicity: Any next generation network proposal must provision equal or better levels of service as compared to existing technologies such as the PSTN. The large number of computational objects required to setup basic calls within the same administrative domain translates into large complexities within the architecture. By comparison to ISUP messaging used with the PSTN environment, the TINA NRA based architecture is unnecessarily complex. Much of the TINA NRA was strongly based on concepts of ATM, therefore aspects such as available ports on terminals contributed to the complexity.

B. Technology Independence: At the outset the TINA NRA was based on the technology agnostic principle. The architecture interfaces to specific transport technologies at both the LNC and CP object as illustrated in Figure 3-2. The control signalling depicted in the message sequence chart (Figure 3-4) emphasises the use of technology independent interfaces rather than technology specific protocols for call setup.

C. Open Interfaces: The TINA NRA leverages the use of open interfaces for interfacing to both adjacent domains such as the Service Architecture, Consumer Domain or the specific Transport Technologies (Network Elements) as well as leveraging on open interfaces to interface to internal architectural objects, inter alia, the CSM, CC and LNC.

D. QoS: The Connection Coordinator (CC) object is responsible for making the admit or reject decision based on whether the call request can be served from both the bandwidth and QoS perspective (TINA Overall 1997). The details of route selection done by the Flow Connection Controller (FCC) in which the FCC needs to determine whether there is available branch in a network flow connection (NFC) that satisfies the connection request. Should the network flow not be available on a single layer (Figure 3-5), the FCC would have to co-ordinate the network flow connection across different layers using suitable adaptors (TINA NRA 1997).

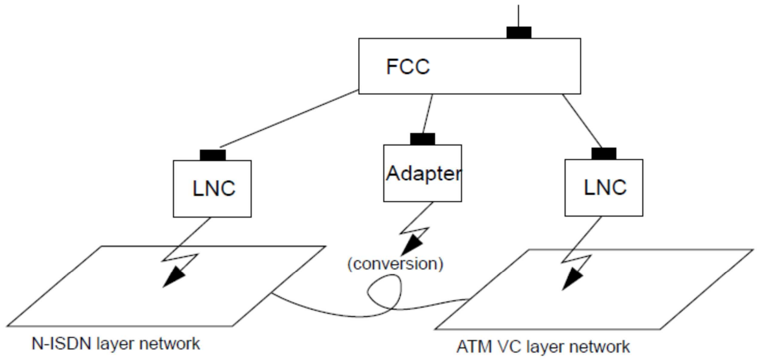

Figure 3-5: TINA use of adapters between layer transport networks, sourced from (TINA NRA 1997).

E. Connection Admission Control: The Connection Coordinator (CC) object is responsible for making the admit or reject decision based on whether the call request can be served from both the bandwidth and QoS perspective (TINA Overall 1997).

F. Routing: The Flow Connection Controller (FCC) is responsible for routing of the connection request in the network. The TINA specification (TINA NRA 1997) does not specify the functionality of how the FCC object determines the best route in the network. TINA NRA assumes that a number of NFC existing within the core network consisting of a number of branches, considered as virtual circuits within the NFC. The FCC leverages available free branches or coordinating the connections between 2 or more braches to facilitate the end-to-end connection request.

G. Federation: The TINA NRA does not address the aspect of federation to another network adequately.

H. Legacy: The TINA NRA does not address interworking with legacy technology.

3.1.6 Conclusion

The TINA standard introduced the concept of a business model for NGN architectures facilitating a number of business scenarios, this concept has been adapted by the OSA/Parlay standard. The TINA Consortium adopted the RM-ODP (TINA Overall 1997) as the methodology used to design and specify complex systems in a technologically agnostic manner which is useful. The architecture clearly separates service control from resource control by specifying the service architecture and network resource architecture respectively. This separation is designed utilising a powerful mechanism of a standard open interface.

The session model is the guiding principle in the TINA NRA Information model. A session is a temporary allocation of resources that are configured to perform a service task for a finite period of time (TINA NRIM 1997; TINA NRA 1997). Once the task is complete the resources are made free and reallocated to another unrelated task. The communication session supports a service request using a logical connection graph. The connectivity session supports the physical connections in the network managed by the physical connection graph. These graphs are leveraged to support call state models used by advanced services as detection points. The session model can be adopted for the OSA/Parlay Network Architecture.

Although the TINA NRA was complex, the concept of some computational objects prove useful. The CSM object is responsible for interacting with the service architecture in developing the logical connection of the service request. The call state information contained in the logical connection graph provides detection points abstracted for the service architecture. The CC object centrally coordinates the admission decision, QoS guarantees and translating the logical connection into physical connections. The LNC object is responsible for the management of non-session related network resources, while the CP object interacts with the technology specific nodes. These computation objects provide insight for the design of the OSA/Parlay Network Architecture.

A number of factors can be attributed to the poor adoption of TINA, the TINA NRA was extensively based on the ATM technology which was too technology specific. The architecture did not address both the interoperability of other network architectures nor interworking with legacy technologies. The complexity of computational object in the TINA NRA was also a factor. These lessons are taken into account and addressed in the design of the OSA/Parlay Network Architecture.

3.2 Multiservice Switching Forum (MSF)

The Multiservice Switching Forum (MSF) (MSF 2006) is a proposal for the NGN architecture. The objective of the MSF Architecture is to facilitate a multivendor environment based on interoperable standards. To speed up the adoption of the MSF, it has defined an architecture that incorporates latest technologies (standards) and addresses outstanding architectural matters such as QoS and Security. A number of existing signalling protocols between various components of the architecture have been specified to ensure quick adoption of the architecture, however this definition is not comprehensive. The architecture incorporates IMS components for call control. The general architecture consists of 6 domains: customer equipment, transport, call & session control, common resources, applications & services and the federation domain.(MSF 2006; Moodley and Hanrahan 2008).

The Customer Equipment Domain consisting of static, nomadic and mobile equipment. Network edge devices are used to interface terminals to the network. The MSF treats all terminal equipment as untrusted and thus need to be registered onto the network. The Basic SIP terminals, residential gateways (RGW) and legacy (POTS/ISDN) register with the Call Agent (CA) and the Home Subscriber Server (HSS) respectively. While the IMS-aware terminals and 3GPP terminals are registered on the Serving Call Session Controller (S-CSC) and the HSS. The IMS-aware terminals in addition are registered on the Subscription Locator Server (SLS) (MSF 2006; Moodley and Hanrahan 2008).

For bearer interfaces, the IMS-aware SIP, Basic SIP and RGW equipment interface to the Data Path Session Border Gateway, Network Edge (D-SBG-NE) for bearer connections. The 3GPP UE interfaces to the Gateway GPRS Supporting Node (GGSN) for bearer connections. The POTS\ISDN terminals interface to the Access Gateway (AGW) for both bearer and signalling. (MSF 2006; Moodley and Hanrahan 2008).

For signalling interfaces, basic SIP and RGW devices signal using SIP and Media Gateway Control Protocol (MGCP) protocols respectively to communicate with the Signalling Path Session Border Gateway, Network Edge (S-SBG-NE). The IMS-aware SIP UA and 3GPP terminals utilise an advanced adaptation of SIP to communicate with the Proxy Call Session Controller (P-CSC). (MSF 2006; Moodley and Hanrahan 2008).

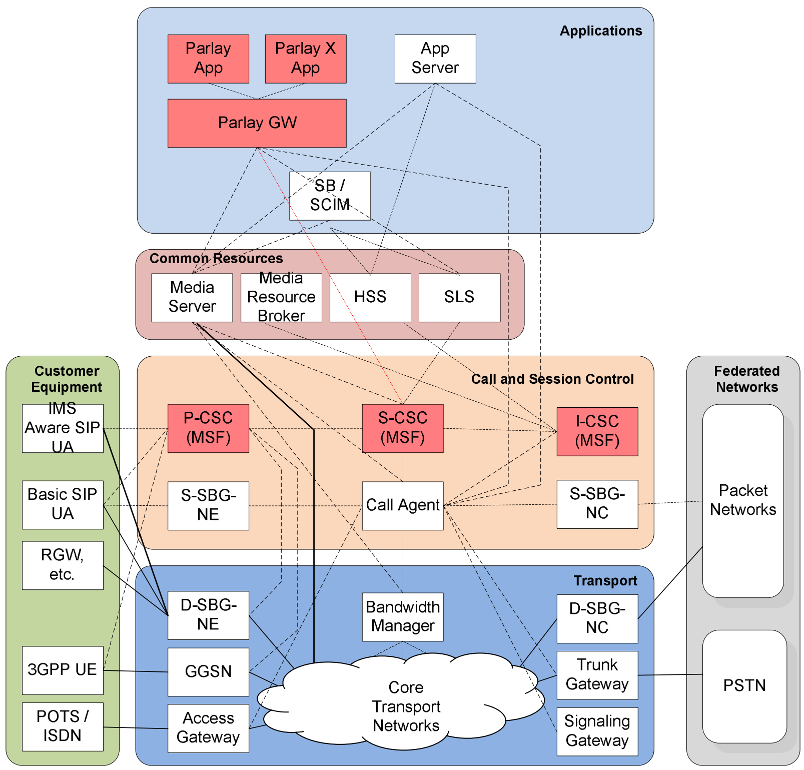

Figure 3-6: MSF Network Architecture, adapted from (MSF 2006)

The Transport domain is primarily responsible for the bearer connections within the architecture. Network Edge elements provide QoS functions of admission control, packet marking and resource allocation. These Network Edge elements in addition are responsible for mediation to other domain elements, Network Address Translation (NAT) and flow measurement. (MSF 2006; Moodley and Hanrahan 2008).

The Data Path Session Border Gateway, Network Core (D-SBG-NC), Trunking Gateway (TGW) and Signalling Gateway (SGW) fulfil edge interface functionality to federated packet networks and the federated PSTN. The Core Transport Network routes traffic between end-points. The Bandwidth manager (BM) provides a QoS guarantees for connections by the implementation of the admission control function. The BM is the cental point of contact for all connection requests to utilise the core transport network. It is aware of network state by liaising for network components, using this information to make the admission control function, however the interface protocols are not defined.

The Call and Session Control domain provisions the initial signalling points for customer terminals, the signalling points for federated networks and elements required to orchestrate calls and sessions within the network. The P-CSC provides initial signalling points for the IMS-aware SIP and 3GPP user terminals, while the S-SBG-NE provide initial signalling points for the Basic SIP and RGW devices. The Signalling Path Session Border Gateway, Network Core (S-SBG-NC) is the signalling interface to federated packet based networks. The Call Agent (CA) provides session setup, control and tear down and has access to routing information for this purpose. It is responsible for the billing and invokes services from the service domain. Similarly the S-CSC provides all the CA functions specifically for the IMS-aware SIP UA and 3GPP UEs. The Interrogating Call and Session Controller (I-CSC) is the responsible proxy element for incoming session from federated IMS type networks uses to interrogate the HSS and SLS for user information. (MSF 2006; Moodley and Hanrahan 2008).

The Common Resources domain contains common elements frequently used for call requests. The HSS is responsible for all the telco's specific subscriber information. Location information of the mobile terminals is stored in the SLS. The Media Server (MS) provides resources such as special tones, IVR type announcements, conference bridging and automatic speech recognitions. The Media Resource Broker (MRB) provides brokering services to locate specific Media Servers. (MSF 2006; Moodley and Hanrahan 2008).

The Application domain consists of advanced services. These services are provisioned in the Application Servers (AS), Parlay Applications and Parlay X Applications. The Parlay Gateway and Parlay X Gateway act as SIP Application servers. The Service Broker (SB/SCIM) provides brokering service to locate specific applications. (MSF 2006; Moodley and Hanrahan 2008).

The Federated domain consists of the legacy PSTN network accessed by the SGW (signalling) and TGW (bearer). The architecture makes provision for the interworking with packet based federated networks utilising the S-SBG-NC (signalling) and D-SBG-NC (bearer). (MSF 2006; Moodley and Hanrahan 2008).

3.2.1 MSF Use Case: 2-Party Call (IMS aware SIP to IMS aware SIP Call)

The typical call setup procedure of the MSF architecture is described in this section. The details of the message sequence chart will be used to analyse some of the characteristics of the architecture (Yun and Perros 2010; Moodley and Hanrahan 2008). The terminals are within the same administrative domain, the network entities conform to the architecture defined in Figure 3-6.

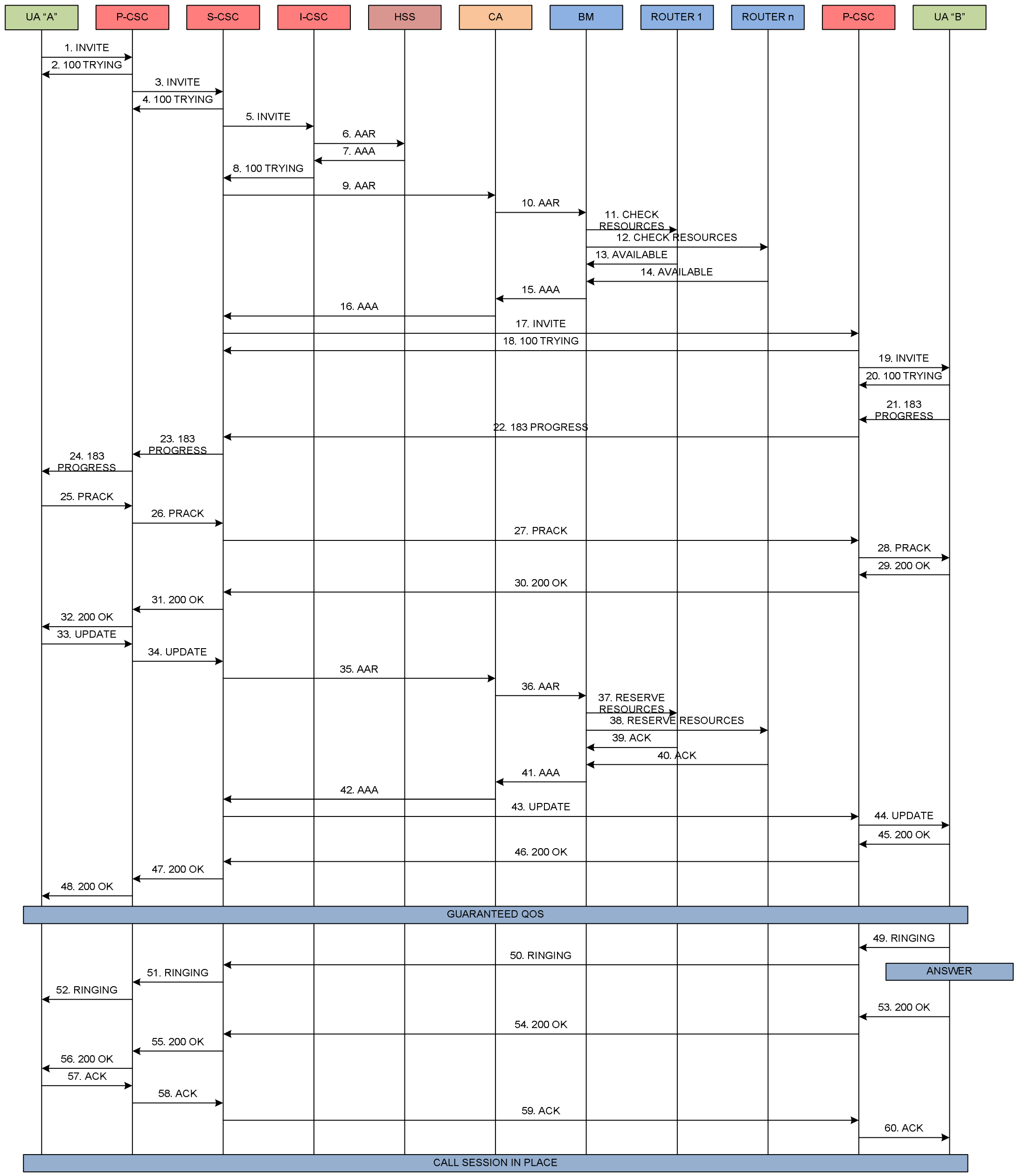

Figure 3-7: MSF 2-party call setup – message sequence chart, adapted from (Moodley and Hanrahan 2008)

The control messages used to setup the call is depicted in Figure 3-7. The call setup sequence is described as follows (Moodley and Hanrahan 2008):

1,2 The originating UA sends a SIP INVITE message to the P-CSC. The proposed connection parameters are carried within the SDP message. The 100 TRYING message assures the UA that the invite is being processed.

3-4 The SIP INVITE message is forwarded to the S-CSC the entity required to orchestrate the call setup. The S-CSC similarly reassures the P-CSC that the invite message is being processed.

5-8 The S-CSC requires the location information of the terminating party. A request is sent to the I-CSC to interrogate the HSS database for the location of the terminating party. The results are forwarded to the S-CSC via the I-CSC.

9-16 The S-CSC must determine whether sufficient resources exist to service the connection request within the core transport network. The enquiry request is sent to the BM via the CA, in turn the BM interrogates network elements (routers) for end-to-end end network status. The results are returned to the S-CSC via the CA from the BM. We deal with the positive result scenario.

17-20 The S-CSC now extends the SIP INVITE message to the terminating P-CSC and party. The associated 100 TRYING messages provides the sending entity assurance that the invite is being processed.

21,22 The terminating party provides an Offer Response message associated with the initial SDP offer using the SIP 183 Session Progress message

23,24 On receipt of the SIP Offer Response message from the terminating party, the S-CSC invokes logic to incorporate network constraints. The modified Offer Response message is sent towards the Originating party via the P-CSC

25-32 The originating UA's positive confirmation to the terminating UA's media offer is sent in a Response Confirmation, implemented as a SIP Provisional Acknowledge or PRACK method. The terminating party acknowledges the confirmation with a Confirm Acknowledge message, implemented using the SIP 200 OK message.

33,34 On receipt of the SIP 200 OK, the originating party sends a forward Resource Confirm message (SIP UPDATE).

35-42 At the S-CSC the SIP UPDATE indicates that the media negotiation has been finalised. The S-CSC invokes the resource reservation: the S-CSC engages the CA to reserve resources in an Authenticate and Authorize Request (AAR). In turn, the CA engages the BM to reserve the appropriate resources. The BM instructs the appropriate network elements to reserve resources.

43,44 Upon receipt of the positive confirmation from the BM via the CA, the S-CSC forwards the resource confirmation message (SIP UPDATE) towards the terminating party.

45-48 The terminating party responds with a reverse direction Resource Confirmation (SIP 200 OK) conveying confirmation of the reservation. The end-to-end path is now reserved and provisioned with the required connection QoS.

45-48 The terminating party responds with a reverse direction Resource Confirmation (SIP 200 OK) conveying confirmation of the reservation. The end-to-end path is now reserved and provisioned with the required QoS for the connection

49-52 The SIP 180 RINGING message indicates the terminating parties alerting state.

53-56 The SIP 200 OK message indicates the terminating party's off-hook status.

57-60 The connection setup is completed by the SIP ACK sent to the terminating party.

3.2.2 Analysis against Architectural Requirements

A simple use case for call setup within the MSF architecture has been presented. The architectural characteristics are ascertained based on the characteristics of the control messages used to setup the connections (Moodley and Hanrahan 2008):

A. Simplicity: Any next generation network proposal must provision equal or better levels of service as compared to existing technologies such as the PSTN. The large number of control messages required to setup basic calls within the same administrative domain translates into a large network overhead within the MSF architecture (Li et al 2007). By comparison to the existing PSTN based ISUP messaging for a 2-party call, the IMS based MSF architecture is unnecessarily complex. The opportunity costs of the architecture are significant, when considering the unavailability of resources and thus loss of revenue due to the significant signalling overheads. (Moodley and Hanrahan 2008).

The architecture depicted in Figure 3-6 has many redundant elements, in particular the S-SBG-NE and CA configuration are SIP based elements. While the P-CSC, S-CSC and I-CSC are components of the IMS, also SIP flavoured elements. The interworking of these elements within the call control functions is a point of difficulty. (Moodley and Hanrahan 2008).

The MSF architecture attempts to cater for all technologies and standards such as VoIP, SIP, IMS, mobility, etc. The result of the all-encompassing architecture is the complexity of many dedicated gateway elements used to deal with each call scenario. (Moodley and Hanrahan 2008).

B. Technology Independence: The architecture is very technology dependant as it has been designed to cater for all common technologies. The MSF has indicated that an unspecified technology dependent protocol will be used for the QoS assurance components of the architecture. As the protocol is unspecified, it poses difficulty of the communication of the CA and BM with network edge elements used for resource reservation as well as admission control. Much of the elements used for Call and Session Control as well as terminals are SIP dependant. In addition many technology specific elements are defined within the architecture making the architecture unscalable and future limiting. (Moodley and Hanrahan 2008).

C. Open Interfaces: Much of the architectural interfaces is SIP dependant and therefore not open. In particular the interface between the Service Broker and Application Server as well as the Service Broker and the Parlay Gateway is SIP based. Protocols are technology dependant and are limited by the definition of the messages contained therein. The SIP protocol requires the application developer to have intricate knowledge of the network to design services. (Moodley and Hanrahan 2008).

D. QoS: MSF architecture adopts an engineering approach to QoS by specifying a central element used for call admission control while the network edge elements ensure that no traffic enters the network without the admission decision. However, as depicted within the call setup use cases, the SIP protocol dictates network-level QoS mechanisms, negotiation between end parties of media streams and QoS parameters. The QoS components are required to query and setup a connection path for every call request. The architecture is inflexible in achieving application-related QoS. (Moodley and Hanrahan 2008).

E. Connection Admission Control: The MSF describes its intention to provide access control on the network edges to the core transport network. However, the interface protocols to achieving this end remains undefined. Thus the CAC mechanism is not achieved as supported in the use case diagrams. (Moodley and Hanrahan 2008).

F. Routing: The architecture is reliant on the transport network technology as MPLS. However, the MSCs show that the architecture is reliant on the routing mechanism of the underlying transport network technology and not by the Call and Session Control elements. (Moodley and Hanrahan 2008).

G. Federation: The architecture defines specific interfaces for federation both with PSTN and Packet based operators. However, from call setup scenarios across administrative domains will result in a proliferation of signalling messages to setup a connection implying large signalling overheads. Specifically, interconnecting into a federated MSF type network results in mirrored control messages hence doubling the number messages. (Moodley and Hanrahan 2008).

H. Legacy: The MSF architecture provides for a number of gateway elements to address the requirement of integrating legacy technologies into the architecture. In particular, the PSTN has been provisioned. (Moodley and Hanrahan 2008).

3.2.3 Conclusion

The MSF architecture proposes the BM function which is the single point of contact for all connection requests in the network. The BM makes use of information on the state of all network components to determine whether the connection request in terms of bandwidth and QoS requirement can be met. In addition, the network edge elements ensure that no traffic from the terminals traverse the network without an admission decision from the BM. A key principle is the communication between the BM element and the Network Edge elements, however, this protocol is undefined in the standard. The QoS principles are useful and can be adopted for the OSA/Parlay Network Architecture.

The IMS elements used for Call and Session Control create a significant overhead and complexity for a basic two party call within the administrative domain of the architecture. Similarly, incorporating technology specific elements to deal with interoperation and legacy networks contribute significantly to the complexity of the architecture. The technology specific protocols and architectural elements limit the future proofing of the architecture.

3.3 3GPP Long Term Evolution (LTE)

The 3rd Generation Partnership Project: Long Term Evolution (3GPP LTE) is a proposal for the next generation mobile network. The 3GPP LTE architecture is an evolution of mobile network architectures. The 3GPP LTE's objective is to provide an all IP based network for a multiservice environment, apart from new proposals for a variety of incorporated radio access networks (3GPP IMS 2012; 3GPP NA 2003; 3GPP Signal Flows IMS 2012; Hanrahan 2007).

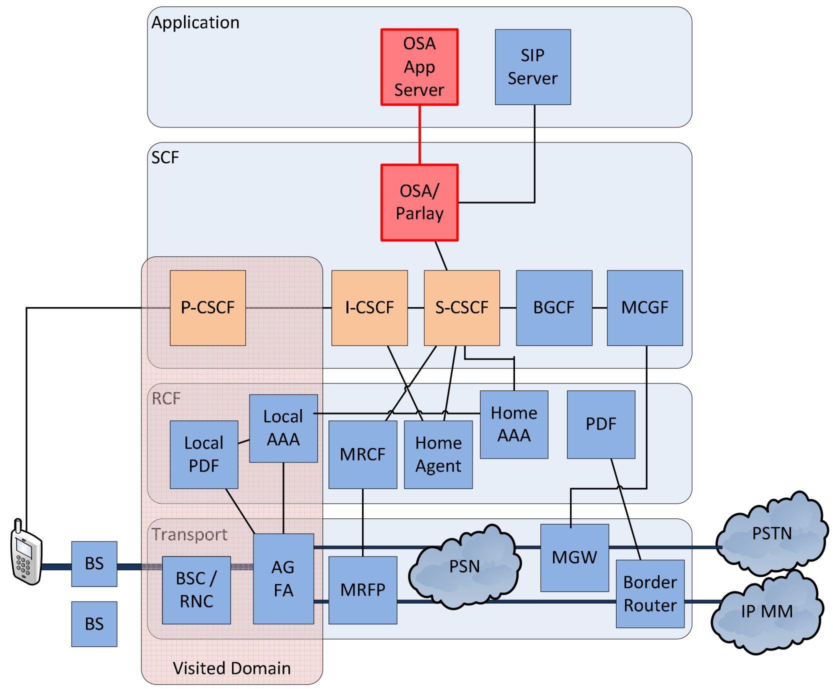

Figure 3-8: 3GPP LTE Network Architecture, adapted from (Hanrahan 2007)

The transport network is based on the Evolved Packet Core (EPC) principle and is an IPv6 network. At the switching layer the Media Gateway (MGW) is used to translate between switched circuit network and media streams from a packet network. The Media Resource Function Processor (MRFP) is responsible for IVR type functions such as announcements and collecting user information. (3GPP NA 2012).

The Service Control Layer is constituted by IMS components which are SIP based. The P-CSCF is the first point of contact for the mobile terminal (UE). The P-CSCF coordinates with the S-CSCF to complete connection requests. In addition the P-CSCF is responsible for QoS by interacting with the Policy Decision Function (PDF). The I-CSCF is used to hide the details of the home network from the visited operator, it enquires from the HSS for the S-CSCF to complete the call request. The I-CSCF also coordinates with the Supporting Location Server (SLS). The S-CSCF central entity for making call requests decisions. The S-CSCF interacts with an the application layer to deliver value-added services. The Breakout Gateway Control Function (BGCF) is responsible for a call from an

IMS network to a switched circuit network of a federated telco. Closely associated with the BGCF is the MGCF which is responsible for signalling to the switched circuit network. (3GPP NA 2012; 3GPP IMS 2012; Magedanz et al 2005; Bertrand 2007)

The application layer provisions for three possible application platforms. The SIP Application Server (SIP-AS) can perform advanced SIP functions and applications. The CAMEL Service Environment is a legacy service platform used for switch circuit services. The Open Service Architecture application server is the OSA/Parlay Gateway. (3GPP NA 2012).

The QoS function is provisioned by the Policy Decision Function (PDF) function (Yavatkar et al 2000; Ericsson 2009), it relies on a policy of QoS rules to determine whether the connection request is allowed to use the network resource. The Policy Decision Point (PDP) applies the rules to the request and produces a decision. The Policy Enforcement Point (PEP) implements the PDP decision which allows or prevents use of the resource. Further details on the PDF are discussed in section 3.3.2.

3.3.1 3GPP LTE Use Case: 2-Party Call

The typical call setup procedure of the 3GPP LTE architecture is described in this section. The details of the message sequence chart will be used to analyse some of the characteristics of the architecture (Yun and Perros 2010; Moodley and Hanrahan 2008). The network entities conform to the architecture defined in Figure 3-8 above.

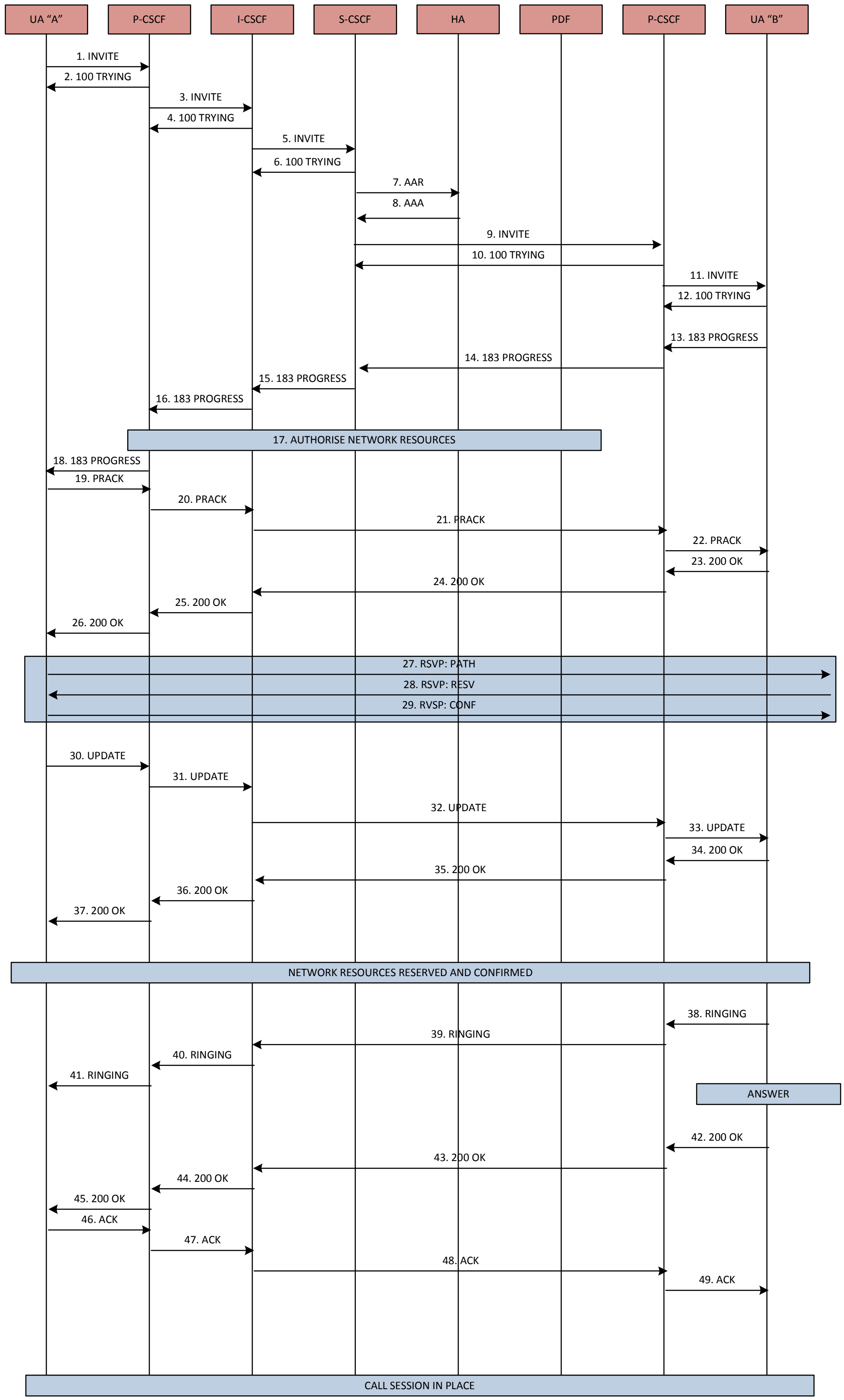

Figure 3-9: 3GPP LTE 2-party call setup - message sequence chart, adapted from (Hanrahan 2007)

The control messages used to setup the call is depicted in Figure 3-9. The message sequence chart is limited to control signalling and excludes bear messaging. It is assumed that the user has been authenticated. The call setup sequence is described as follows (Hanrahan 2007):

1,2 The originating UA sends a SIP INVITE message to the P-CSCF in the visited domain. The proposed connection parameters are carried within the SDP message. The 100 TRYING message assures the UA that the invite is being processed.

3-4 The SIP INVITE message is forwarded to the I-CSCF entity being the element used between the federated networks of the two telcos, it acts as a proxy server for the home domain. In addition, the I-CSCF hides the details of the home domain from the federated visited network. Similar to the P-CSCF, the I-CSCF reassures the P-CSCF that the invite message is being processed.

5-6 The SIP INVITE message is forwarded to the S-CSCF. The S-CSCG is the entity required to orchestrate the call setup. The S-CSCF similarly reassures the I-CSCF that the invite message is being processed.

7-8 The S-CSCF requires the location information of the terminating party. A request is sent to the Home Agent (HA) which maintains the location of the users. The result of the location query is returned to the S-CSCF.

9-12 The S-CSCF now extends the SIP INVITE message to the terminating P-CSCF and party. The associated 100 TRYING messages provides the sending entity assurance that the invite is being processed.

13-16 The terminating party provides an Offer Response message associated with the initial SDP offer using the SIP 183 Session Progress message.

17 On receipt of the SIP Offer Response message from the terminating party, the P-CSCF invokes logic to incorporate network constraints. The P-CSCF is aided by the PDF function authorises the network resources to fulfil the connection request.

18 The 183 Session Progress message is sent to the UA together with the authorisation mechanism. The session path is now known.

19-26 The originating UA's positive confirmation to the terminating UA's media offer is sent in a Response Confirmation, implemented as a SIP Provisional Acknowledge or PRACK method. The terminating party acknowledges the confirmation with a confirm acknowledge message, implemented using the SIP 200 OK message. The use of the network resources have now been authorised.

27-29 The Resource Reservation Protocol (RSVP) is used to reserve the bearer end-to-end resources.

30-37 These message exchanges being the Update and 200 Ok confirm that the resources are to be reserved between the calling parties. These session path already known (see message 18) is used for the transfer of the messages.

38-41 The SIP 180 RINGING message indicates the terminating parties alerting state.

42-45 The SIP 200 OK message indicates the terminating party's off-hook status.

46-49 The connection setup is completed by the SIP ACK sent to the terminating party.

3.3.2 Analysis against Architectural Requirements

A simple use case for call setup within the 3GPP LTE architecture has been presented. The architectural characteristics are ascertained based on the characteristics of the control messages used to setup the connections (Yun and Perros 2010; Moodley and Hanrahan 2008):

A. Simplicity: Any next generation network proposal must provision equal or better levels of service as compared to existing technologies such as the PSTN. The large number of control messages required to setup basic calls within the same administrative domain translates into a large network overhead within the 3GPP LTE architecture (Li et al 2007). By comparison to the existing PSTN based ISUP messaging for a 2-party call, the IMS based 3GPP LTE architecture is unnecessarily complex. The opportunity costs of the architecture are significant, when considering the unavailability of resources and thus loss of revenue due to the significant signalling overheads.

B. Technology Independence: The architecture is technology dependant as it specifies the use of the SIP technology dependant protocol for the IMS components. Protocols are technology dependant and are limited by the definition of the messages contained therein. These technology specific elements makes the architecture unscalable and future limiting.

C. Open Interfaces: Many of the architectural interfaces are SIP dependant and therefore not open interfaces. However, the architecture does expose an open OSA/Parlay interface to Application developers. The architecture does separate the call and session control layer from the transport layer which is specified by the SIP based interfaces. However since the interface is protocol dependent, it is limited and does not satisfy the openness criteria.

D. QoS: The 3GPP TLE has specified a QoS function (Yavatkar et al 2000; Ericsson 2009) and is provisioned by the Policy Decision Function (PDF) . It relies on a policy of QoS rules to determine whether the connection request is allowed to use the network resource. The PDP function computes the decision by comparing the request to the rules, while the PEF function allows or prevents use of the resource based on the PDP decision. The PEF function allocates an IP address which entitles call party access to the area of the network that only administers terminal registration and service requests. Once a connection request is admitted to the network, a new IP address is allocated to the end terminal, provisioning access into the core transport network

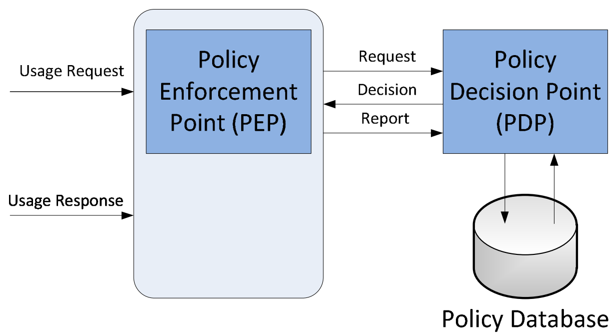

Figure 3-10: 3GPP LTE PDF entities, adapted from (Hanrahan 2007)

Analysing the message sequence chart indicates that network level QoS mechanisms are relied upon. The QoS components are required to query and setup a connection path for every call request. The architecture is inflexible in achieving application-related QoS.

E. Connection Admission Control: The PDF components of the 3GPP LTE Architecture facilitate the call admission control function. In particular, the PDP is the central entity in the architecture which is responsible for coordinating the admission decision of call and connection requests into the transport network.

F. Routing: The architecture is reliant on the RSVP protocol to setup the connection routes in the all IP transport network. The RSVP protocol does not ascertain the best route of a connection path in the network from a holistic view but does ensure together with the QoS assurance that the call request is guaranteed a level of service as requested.

G. Federation: The architecture defines specific interfaces for federation both with PSTN and the Internet. However call setup scenarios across administrative domains will result in a proliferation of signalling messages to setup a connection implying large signalling overheads. Specifically, interconnecting into a federated 3GPP LTE type network results in mirrored control messages hence doubling the number messages.

H. Legacy: The MSF architecture provides for a number of gateway elements to address the requirement of integrating legacy technologies into the architecture. In particular, the PSTN has been provisioned.

3.3.3 Conclusion

The 3GPP LTE architecture proposes the PDP function which is the single point of contact for all connection requests in the network. The PDP makes use of QoS rules to determine whether the connection request in terms QoS requirements can be met. In addition, the P-CSCF on the edge of the network provisions the PEF function to allow or prevent use of the resource based on the PDP decision. The PEF function allocates an IP address to the calling party, which entitles access to the area of the network that only administers terminal registration and service requests. Once a connection request is admitted to the network, a new IP address is allocated to the end terminal, provisioning access into the core transport network. The QoS principles are useful and can influence the design for the OSA/Parlay Network Architecture.

The IMS elements used for Call and Session Control creates a significant overhead and complexity for a basic two party call within the administrative domain of the architecture. The technology specific protocols and SIP based architectural elements limit the future proofing of the architecture.

3.4 OSA/Parlay

The OSA/Parlay architecture leverages new technologies emerging out of the convergence between telecommunication and information technologies (Moodley and Hanrahan 2005) OSA/Parlay assumes a three-layered architecture, consisting of the transport network layer, the service control layer (OSA/Parlay Gateway) and the application & services layer as depicted in Figure 3-11. The OSA/Parlay Gateway is addressed at the ITU service stratum while provisioning an open, secure, technology agnostic API based interface to the Application layer. A key feature of the OSA/Parlay Gateway is the openness of the network as it exposes the open interface to 3rd party application developers. (3GPP OSA Part 1 2009; Moodley and Hanrahan 2005; Beddus and Bruce 2000)

The architecture provisions a rich set of service control features (SCFs) such as generic call control, multiparty call control, multimedia call control, user interaction and many others (3GPP OSA Part 4-2 2009; 3GPP OSA Part 4-3 2009; 3GPP OSA Part 4-4 2009; 3GPP OSA Part 5 2009). The SCFs allow for intelligent call control and service control functions that interact with various underlying network technologies whilst exposing a simple interface for application developers. Applications may be located in a different business domain and can invoke Applications Programming Interface (API) (3GPP OSA Part 1 2009; Beddus and Bruce 2000; Magedanz et al 2007; Blum et al 2008)

Current research efforts have been focused on the development of applications and services and in particular the service stratum. (Moodley and Hanrahan 2005; Qiao et al 2011; Zhang and Guan 2012; Lei-an and Hong-jiang 2010, Atanasov I et al 2012, Atanasov I 2013). The OSA/Parlay standard has left the interface of the gateway to the transport stratum as implementation details. The standard alludes to the use of protocol adapters for interfacing to the various transport technologies. (Moodley and Hanrahan 2005; Hanrahan 2007)

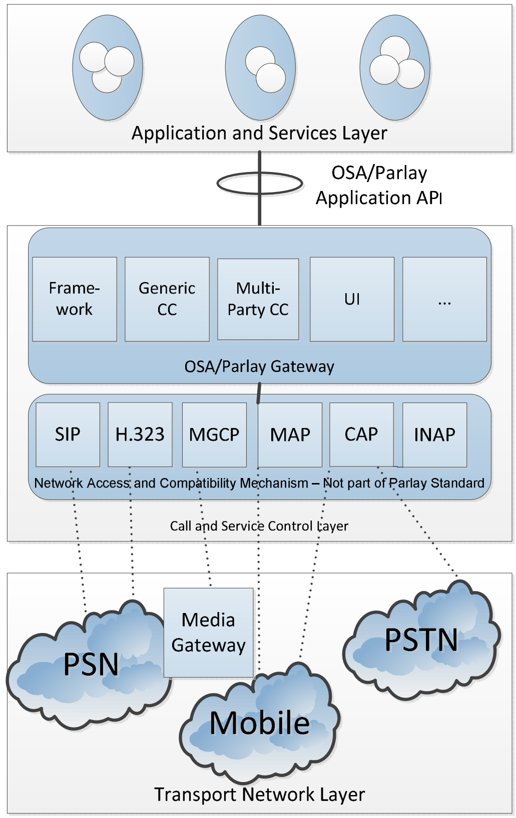

Figure 3-11: OSA/Parlay Service Architecture, adapted from (Moodley and Hanrahan 2005)

A key design principle for the OSA/Parlay architecture is an "open network architecture". Closed networks are characterised by vendor lock-in as the technology platforms have incorporated proprietary implementations. In addition, the scarcity of skills and complexity of protocols have resulted in many of the competing telcos offering similar services together with excessively long lead time to market for new services. Open networks leverage the significant number of application developers available in the IT community. This number of developers are orders of magnitude larger than the service or application developers in the telecommunications community, as illustrated in Figure 3-12. (Beddus and Bruce 2000; Blum et al 2008; Bakker et al 2000)

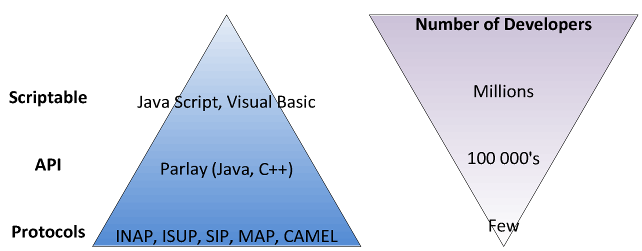

Figure 3-12: OSA/Parlay leverage 3rd Party application developers, adapted from (Magedanz 2012)

Another characteristic is that a multivendor environment is catered for and together with the significant number of developers, telcos have the flexibility to compete on differentiated services and quick time to market. A significant effort in OSA/Parlay was made to create an open interface to the network resources to afford service developers the opportunity to develop advanced services without detailed knowledge of the underlying network technology specifics (Moodley and Hanrahan 2006). These details are hidden using the SCFs whilst exposing generic advanced service features in the Application API based interface.

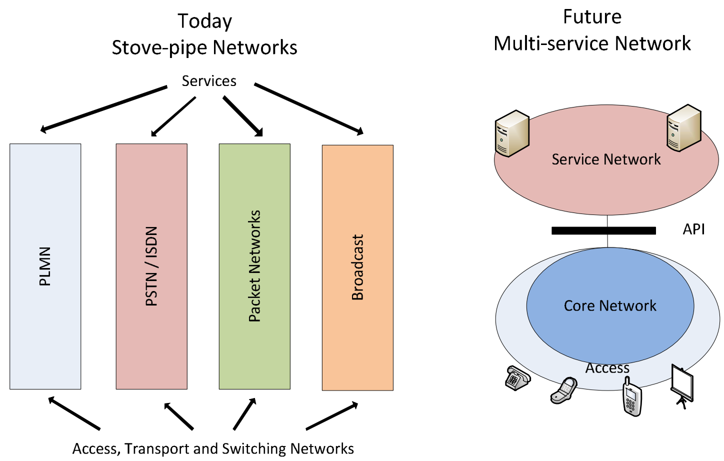

Figure 3-13: Concepts of the Mutliservice Network, adapted from (Magedanz 2012)

An API provides a mechanism to demarcate the domains of the telco and third party service providers. The former existing in a secure environment and the latter in an untrusted environment, the API provides a mechanism dealing with the security issues. The API provides a set of technology agnostic definitions provided as a set of methods and parameters that defines the interface. This technology agnostic specification ensures that the interface is a future proof solution. The API thus allows application developers to develop advanced services without having the need to have detailed knowledge of the network technologies or protocols within the network provider. This ensures that applications have a rapid time to market and hence rapid times to revenue. The telco is advantaged as this interface ensures a multivendor environment and the telco is not locked into a single proprietary vendor. The API ensures that unsecure applications are strictly managed into the telco environment and provides the flexibility of application development at reduced costs and reduced risk to provide differentiated service offerings. The interface allows interoperation to a variety of network technologies as well as signalling systems, while hiding the complexity of the protocols used by these network and signalling technologies. (Magedanz et al 2007; Moodley and Hanrahan 2006; Blum et al 2008; Bakker et al 2000)

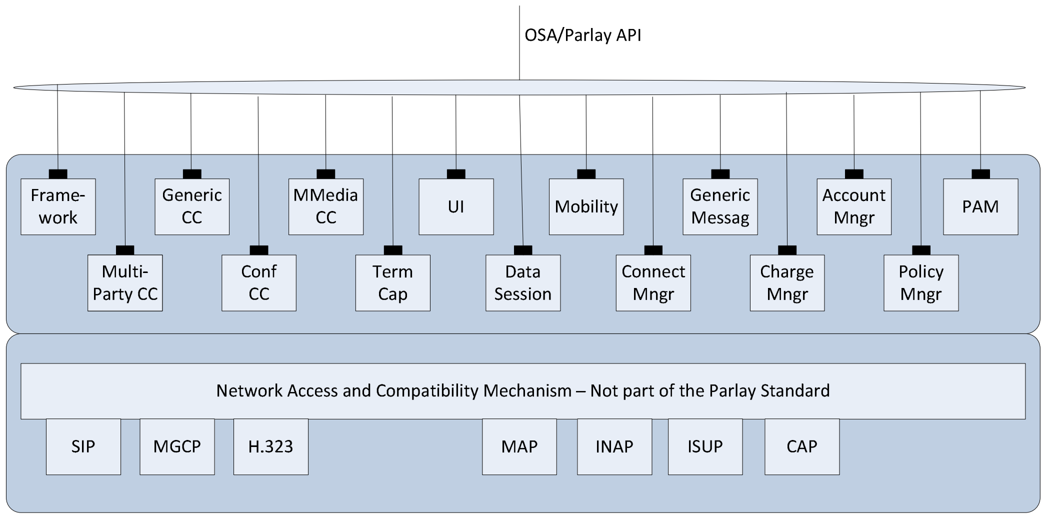

Figure 3-14: Detailed SCF in the OSA/Parlay Service Architecture, adapted from (Hanrahan 2007)

3.4.1 Analysis against Architectural Requirements

Unlike the TINA, MSF and 3GPP LTE architectures, the OSA/Parlay Gateway is not constituted by transport stratum. Therefore presentation of a message sequence chart for a 2-party call is not possible. The architectural characteristics are ascertained based on the characteristics of the concepts and principles applied in the specification of the OSA/Parlay Gateway which can be adapted for use in the OSA/Parlay Network Architecture (Yun and Perros 2010; Moodley and Hanrahan 2008):

A. Simplicity: The OSA/Parlay Gateway specifies the Application Interface in an open and simple manner. This simplicity is needed to hide the complexity of the network technologies and protocols from the application developer who may not have knowledge of the complexities in the network but can still develop advanced applications and services for the telecommunication operator.

B. Technology Independence: The Application Interface together with the OSA/Parlay Gateway is specified in a technology agnostic manner. The interface employs the use of technology agnostic APIs while the OSA/Parlay Gateway components are defined in the Object Orientated paradigm using the technology agnostic Unified Modelling Language (UML) description.

C. Open Interfaces: The OSA/Parlay Application Interface is based on concepts of openness allowing for 3rd party developers to leverage telco resources to provision advanced applications and services.

D. QoS, Connection Admission Control, Routing, Federation and Legacy: Since the OSA/Parlay Gateway provides service stratum functionality, it does not address the issue of QoS, Connection Admission Control, Routing, Federation and interoperability with Legacy technologies. It assumes that the underlying transport stratum provisions this functionality. OSA/Parlay alludes to the use of protocol adapters that may deal with interfacing to legacy technologies.

3.4.2 Conclusion

The OSA/Parlay Gateway is a proposal for the next generation network addressing the service stratum. A powerful mechanism of an open and secure interface is exposed to 3rd party application developers leveraging on the simplicity and technology independence of the interface. This allows IT developers to create advanced applications and services without the need to understand the complexity of technologies and protocols in the transport network. A multivendor environment is created since the Gateway is technologically agnostic and allows for interoperability.

Current research efforts have been focused on the development of applications and services and in particular the service stratum (Moodley and Hanrahan 2005; Qiao et al 2011; Zhang and Guan 2012; Lei-an and Hong-jiang 2010), while little attention has been devoted to the network interface and transport stratum.

The principles and concepts of the open, API-based and technologically independent interface are useful. Similar to the definition of the Application Interface, the OSA/Parlay Network Architecture and Interface will be Object Oriented (OO) and specified using UML.

3.5 Java Advanced Intelligent Network (JAIN)

JAIN is a Java specific implementation of the OSA/Parlay standard. Its goal is to create an open architecture for all stakeholders (Beddus and Bruce 2000). The JAIN architecture applies to both the fixed and mobile networks and has a view of an integrated network architecture. The Java specific implementation ensures that applications are portable in that these applications can be developed and executed on any technology. The Java programming environment allows for rapid development of applications and ease of deployment. JAIN has from the outset incorporated a security feature to ensure secure access to the network from service providers. Java is a well-established IT technology and thus provides a means towards convergence in the Telecommunications and Internet domains. (Beddus and Bruce 2000; de Keijzer et al 2000).

Java as an implementation technology leverages the benefits of reuse of off the shelf standard Java containers; these containers have specified finite state machines to provide state details. The Java approach leverages the existing large community of Java developers and the freely available development tools. (Lim et al 2003).

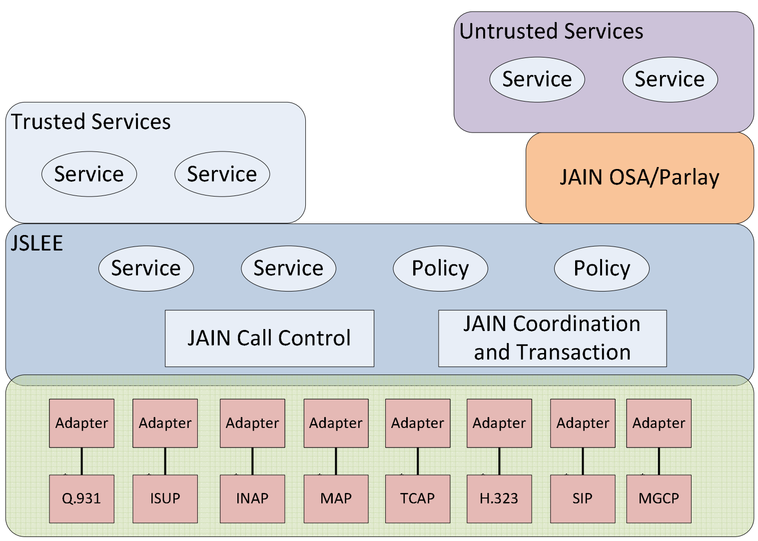

Figure 3-15: JAIN adaptation of the OSA/Parlay Service Architecture with network adapters, adapted from (Hanrahan 2007)

Java has used the API concept and the JAIN architecture leverages of the existing Java APIs. The secure access to network resources is provisioned by placing trusted applications inside the Java Service Logic Execution Environment (JSLEE) while untrusted applications are outside the JSLEE which need to be pre-authorised through the security interface. (de Keijzer et al 2000; Lim et al 2003)

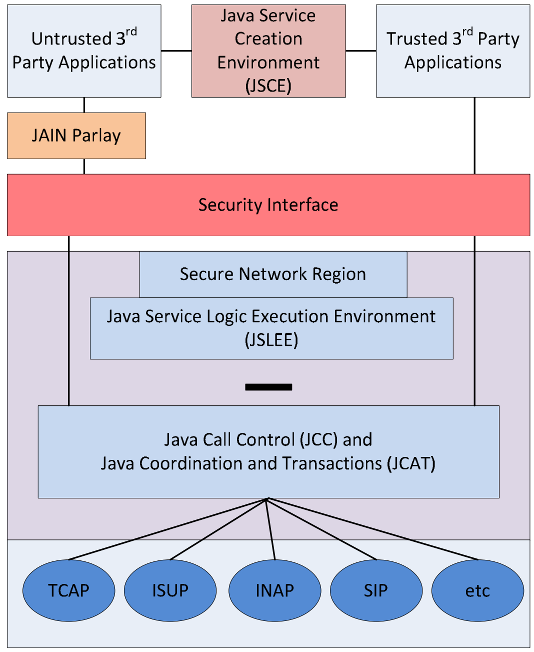

Figure 3-16: JAIN Architecture, adapted from (Jain et al 2000)

JAIN is constituted the following key components:

- JSCE (Java Service Creation Environment)
- JCC (JAIN call control) and JCAT (JAIN co-ordination and transaction) (Jain et al 2000)
- JSLEE (Java Service Logic Execution Environment) (de Keijzer et al 2000; Lim et al 2003)
- Network Protocols Adaptors (Bhat and Gupta 2000)

The JSCE (Java Service Creation Environment) leverages a component service creation approach provided by the Java Beans platform. It also exposes a standard development environment that application developers are familiar with for Telco service creation. JavaBeans is a component model added to the Java, and its primary focus was on reusable components, components that can be visually manipulated and customized in (GUI) development environments to assemble applications using introspection and a runtime event model to provide notification and callbacks.

JSLEE (Java Service Logic Execution Environment) provides an environment for the execution of service logic. JSLEE is designed to ensure a low latency and high throughput environment ensuring that the application environment provides minimal processing delays and is able to process thousands of events per second. The environment is event orientated and designed for the stringent telco conditions of network signalling. The JSLEE is scalable through the use of clustering technologies. (de Keijzer et al 2000; Lim et al 2003)

Network Protocols Adaptors are being developed to interface to existing network technologies (Femminella et al 2012a; Femminella et al 2012b). The JAIN protocol experts group's (PEG) task is to develop Java APIs for protocols using in the traditional "telephony domain, intelligent networks, wireless network and the Internet" (Bhat and Gupta 2000).

3.5.1 Analysis against Architectural Requirements

Similar to the OSA/Parlay Gateway (presented in Chapter 3.4), JAIN is not constituted by transport stratum. Therefore presentation of a message sequence chart for a 2-party call is not possible. The architectural characteristics are ascertained based on the characteristics of the concepts and principles applied in the specification of the OSA/Parlay Gateway which can be adapted for use in the OSA/Parlay Network Architecture (Moodley and Hanrahan 2008):

A. Simplicity: JAIN specifies the Application Interface in an open and simple manner using the Java programming language. This simplicity is needed to hide the complexity of the network technologies and protocols from the application developer whom may not have knowledge of the complexities.

B. Technology Independence: The JAIN Application Interface together with the OSA/Parlay Gateway is specified in a technology agnostic manner. The interface employs the use of technology agnostic APIs. However the use of protocol adapters as the only means for implementing the transport stratum introduces technology dependencies.

C. Open Interfaces: The JAIN Application Interface is based on concepts of openness allowing for 3rd party developers to leverage telco resources to provision advanced applications and services.

D. QoS, Connection Admission Control, Routing, Federation and Legacy: Since JAIN is a proposal for the service stratum functionality, it does not address the issue of QoS, Connection

Admission Control, Routing, Federation and interoperability with Legacy technologies. It assumes that the underlying transport stratum provisions this functionality. JAIN specifies protocol adapters that may deal with interfacing to legacy technologies. (Sun JCC2SIP 2001; Sun JCC2H323 2001, Femminella et al 2012a; Femminella et al 2012b).

3.5.2 Conclusion

The JAIN proposal for the next generation network addressing the service stratum is a Java specific implementation of the OSA/Parlay Gateway. A mechanism of an open and secure interface is exposed to 3rd party application developers leveraging on the simplicity and technology independence of the interface. This allows IT developers to create advanced applications and services without the need to understand the complexity of technologies and protocols in the transport network. A multivendor environment is created since the Gateway is technologically agnostic and allows for interoperability. The use of protocol adapters as the only means for implementing the transport stratum introduces technology dependencies.

The principles and concepts of the open, API-based and technologically independent interface are useful. Java based technology provides a quick and simple means for realising the OSA/Parlay Gateway. The protocol adaptation to federated and legacy network technologies is useful work that can contribute to the OSA/Parlay Network Architecture and Network Interface design.

3.6 Chapter Summary

The OSA/Parlay Network Architecture and Network Interface have a number of key characteristics described in Chapter 1.1 and Chapter 2. The OSA/Parlay Network Architecture and Interface should encompass the following characteristics *openness, simplicity, technology independence, QoS mechanisms; call admission control; intelligent routing* and supporting both *federation* of telcos and interoperation of *legacy* technologies. “The network architecture does not rely on call/session signalling. More powerful call admission, connection co-ordination, network federation and end-to-end Quality of Service mechanisms are incorporated.” (Moodley and Hanrahan 2005).

A number of existing architectural proposals for the transport stratums have been reviewed against the ideal network architectural characteristics. The following architectures have been reviewed:

- The TINA NRA is founded on a technology agnostic principle and specified the interface to the service architecture (service stratum). The session model concept is useful for managing complex connections and service requests. The concepts of some computational objects arranged in a layered architecture provide insight for the design of the OSA/Parlay Network Architecture. Lesson such as addressing interoperability to federated networks and addressing the interoperability with legacy technologies are useful.

- The MSF architecture specifies a useful QoS management system, leveraging on a central entity acting as the single point of contact for all connection requests in the network. The admission decision is based on network state information. The network edge elements ensure that no traffic from the terminals traverse the network without an admission decision from the BM.

- The 3GPP LTE architecture specifies an alternative QoS management system, incorporating a single point of contact for all connection requests in the network. QoS rules determine admission decision. Network Edge element allows or prevents use of the resource based on the admission decision using the temporary IP address mechanism.

- The OSA/Parlay Gateway specifies a powerful mechanism of an *open, standard, simple, secure* and *technology independent* API based interface defined in the object oriented paradigm and specified in UML.

- JAIN contribution to the protocol adaptation to federated and legacy network technologies concepts are useful and can be adopted for the design of the OSA/Parlay Network Architecture. Java based technology provides a quick and simple means for realising the OSA/Parlay Gateway.

The TINA architecture was clearly not a success as it was not widely implemented by industry. Key reasons were that many technologies that TINA relied upon were not available at the time, the architecture was complex and in particular the TINA NRA was too ATM technology focused. While OSA/Parlay Gateway has not been widely adopted due to its lack of specification of the network interface in a similar manner as Parlay has specified the application interface. Integration with all transport protocols is cumbersome and hence the application has been limited to implementations described in section 2.1.

Useful concepts and principles that meet the interface and architectural requirements presented in Table 3-2 are incorporated and adapted for the design of the OSA/Parlay Network Architecture and Network Interface presented in Chapter 4.

	TINA	**MSF**	**3GPP LTE**	**OSA/Parlay**	**JAIN**
Simplicity	Complex	Simple	Complex	Simple	Simple
Technology Independence	Technology agnostic	IMS – SIP dependent	IMS – SIP dependent	Technology agnostic	Java dependent
Openness	Open NI	Multiple protocols for NI	Protocol dependent NI	Protocol adapters for NI	Protocol adapters for NI
QoS	Provisioned by the CC	Provisioned by the BM and Edge	Provisioned by the PDF function (PDP & PEF)	Not specified	Not specified
CAC	Provisioned by the CC	Provisioned by the BM and Edge	Provisioned by the PDF function (PDP, PEF), temporary IP address mechanism	Not specified	Not specified
Routing	Determined by the FCC	Dependent on the RSVP protocol	Dependent on the RSVP protocol	Not specified	Not specified
Federation	Not specified	Federation to packet and switched circuit	Federation to packet and switched circuit	Not specified	Protocol adapters for NI
Legacy	Not specified	Adapts to multiple legacy technologies	Adapts to multiple legacy technologies	Not specified	Protocol adapters for NI

Table 3-2: Summary of useful concepts and principles

Chapter 4

Design of the OSA/Parlay Network Architecture and Network Interface

The contribution of this research is the design of the OSA/Parlay Network Architecture and Network Interface to extend the OSA/Parlay standard towards meeting the requirements of a NGN. The primary objective of the OSA/Parlay Network Architecture is to facilitate the call and service control requests from the OSA/Parlay Gateway and setup the end-to-end connections within the core transport network. The OSA/Parlay Architecture must ensure that new calls entering the transport network are guaranteed the requested level of QoS for the entire duration of the call; while the QoS levels of existing calls within the network are not infringed upon notwithstanding the dynamic nature of calls being admitted and terminated within the transport network (Moodley and Hanrahan 2008).

The OSA/Parlay Network Interface conjoins the OSA/Parlay Gateway to the proposed OSA/Parlay Network Architecture. It should be characterised as an open, standard and technologically independent interface. These characteristics ensure that the OSA/Parlay Network Architecture is future proof from rapid technology developments as it allows for the ease of simple integration of a variety of transport technologies. The OSA/Parlay Network Interface will be API based and defined in the Object Oriented paradigm using UML, a similar approach adopted for the OSA/Parlay Application Interface.

The OSA/Parlay Network Architecture is designed to render an engineered solution towards allocation of network resources that guaranteed QoS requirements rather than the overprovisioning of network resources with best effort concepts. The OSA/Parlay network architecture is not a detailed definition, it specifies concepts and principles for connections involving the packet and switch circuit technologies as well as incorporates special resources such as media gateways and IVRs (Moodley and Hanrahan 2008).

The OSA/Parlay Network Architecture must conform to the following characteristics to ensure a flexible and technology agnostic proposal for the next generation network (Moodley and Hanrahan 2008):

- “be designed as a *simple* architecture;
- demonstrate *technology independence*;
- support *open interfaces*;
- enforce an integrated approach for *end-to-end connections*;
- implement connection admission *QoS* control as part of the overall QoS assurance system;
- provision *intelligent routing* of connections;
- support *federation* between network operators to provide QoS-assured end-to-end connections;
- support *legacy* technologies in the overall architecture.”

Many useful concepts and principles have been analysed from existing architectural proposals presented in Chapter 3. In particular, concepts from proposal such as the Telecommunication Intelligent Network Architecture (TINA), Open Service Access / Parlay (OSA/Parlay) and the Multiservice Switching Forum (MSF) are synthesised toward the development of the OSA/Parlay Network Architecture and Network Interface.

4.1 Design Methodology

The OSA/Parlay Network Architecture is a complex system that aspires to be specified in an open and technology independent manner. In order to manage the design of the complex architecture as well as technology agnostic principles, the revised model for open distributed processing (RM-ODP) methodology has been adopted. The RM-ODP methodology was applied to the development of the TINA NRA (TINA Overall 1997). In order to deal with the complexity, the five viewpoints of RM-ODP are used to focus on the details of a subset characteristics of the system. To obtain the complete view of the system, these five viewpoints are combined.

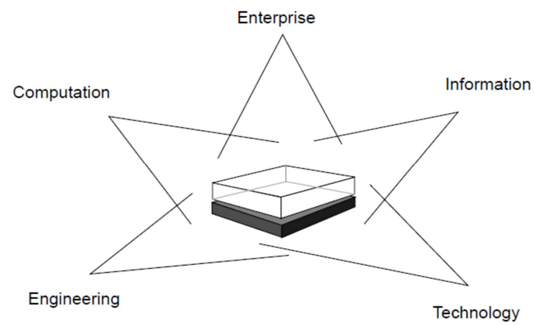

Figure 4-1: RM-ODP Viewpoints, sourced from (TINA Overall 1997)

The five RM-ODP viewpoints (Figure 4-1) are described as follows (TINA Overall 1997):

- **"Enterprise viewpoint** develops the purpose, scope and policies for the system.
- **Information viewpoint** specifies the semantics of information and information processing activities in the system.
- **Computational viewpoint** focuses on the decomposition of the system into a set of interacting objects which are candidates for distribution.
- **Engineering viewpoint** specifies on the infrastructure required to support distribution processing environment.
- **Technology viewpoint** details the choice of technology to support the system."

4.2 Business Model

The OSA/Parlay Network Architecture adheres to the OSA/Parlay business model (Roj 2012; Hanrahan 2007). The business model is based on an Open Network principle, see Figure 4-2. The open network principle allows 3rd party service providers to provide content and advanced services to the consumer. The consumer can access these services between the consumer and 3rd party domains through the application interface. The 3rd party domain provides the services through the retailer domain which provides generic service logic functions. The retailer domain is separated from the network provider by the network interface currently proposed in this thesis.

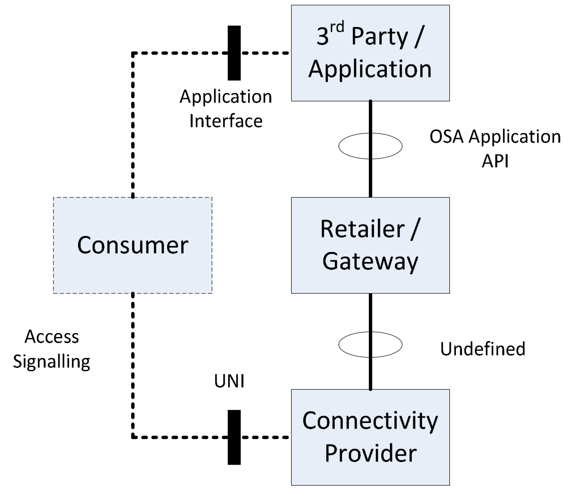

Figure 4-2: OSA/Parlay Business Model, adapted from (Hanrahan 2007)

4.3 Conceptual OSA/Parlay Network Architecture (RM-ODP Enterprise Viewpoint)

The RM-ODP enterprise viewpoint describes the conceptual architecture and details the abstract objects in the network architecture, their purpose, scope and objective. The complete conceptual architecture is depicted in Figure 4-3 below and further describes the environment and domain of the architecture as well as the roles, relationships and interactions between the objects (Moodley and van Olst 2011). Similar to the ITU NGN framework, the architecture consists of the Application Layer; Call Control Layer (ITU service stratum) and the Transport Network Layer (ITU transport stratum).

The Call Control Layer is well defined by the OSA/Parlay Gateway and the interface to the Application Layer is specified by the open, secure, API based Application Interface. The Consumer Domain is illustrated to depict the interface to the OSA/Parlay Network Architecture. Similar to the Application Layer and Call Control Layer, the Consumer Domain remains out of the scope of this research work.

The OSA/Parlay Network Architecture exposes an interface to a Media Gateway element used for federation to other telco networks as well as adaptation to legacy technologies. The common telco

IVR element is depicted with interfaces to the OSA/Parlay Network Architecture. (Moodley and van Olst 2011)

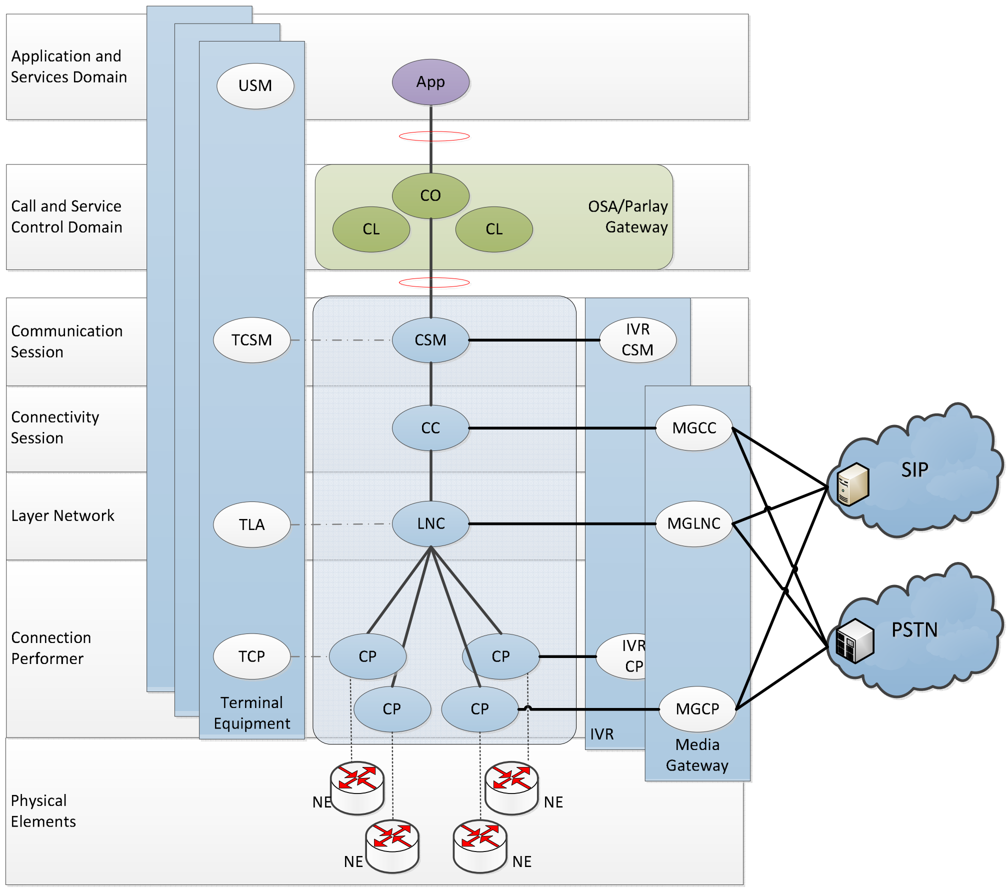

Figure 4-3: OSA/Parlay Conceptual Architecture, adapted from (Moodley and van Olst 2011)

Application Layer Objects:

App: The Application (App) resides in the application and services domain. The App initiates calls and services between the various call parties within the network and possibly IVR's. (Moodley and van Olst 2011)

Call Control Layer Objects:

CO and **CL**'s: The Call Object (CO) and Call Legs (CL's) are the objects within the OSA/Parlay Gateway and resides in the control layer domain. The CO and CL's represent the call and call parties within a call. The call properties including the QoS requested for the call are stored in these objects. (Moodley and van Olst 2011)

Transport Network Layer Objects:

CSM: The Communication Session Manager (CSM) is responsible for the communication between the transport network and the OSA/Parlay Gateway. The CSM is also responsible for the Logical Connection Graph (LCG) for each instantiated call. (Moodley and van Olst 2011)

CC: The Connection Coordinator (CC) is responsible for translating the LCG into physical connections. The CC core functions are to determine which layers are to be used for the call and selects the appropriate Layer Network Coordinator (LNC). It also provides the function of admission control which is responsible for determining whether the requested call properties (QoS) can be supplied by the transport network and on that basis, implements the call admission control function. The CC is also responsible for managing the Administrative Federation on a call by call basis. (Moodley and van Olst 2011)

LNC and **CP**: The Layer Network Coordinator (LNC) is responsible for the connection within the layer transport network. Its function is to set up the connections via the Connection Performer (CP) in sub-layer transport networks. The CP and LNC provide network state information back to the CC. (Moodley and van Olst 2011)

NE: The Network Element (NE) represents the actual physical network element in the network, providing information on its capabilities and the current state of the device. (Moodley and van Olst 2011)

4.3.1 Other Domain specific objects

MGCC, MGLNC and **MGCP**: The Media Gateway-Connection Coordinator (MGCC), Media Gateway-LNC (MGLNC) and Media Gateway-CP (MGCP) are adaptations of the Connection Coordinator (CC), LNC (signalling) and CP (Connection Performers) specific to the Media Gateway special resource. (Moodley and van Olst 2011)

Exch: The Gateway exchange into the PSTN network. (Moodley and van Olst 2011)

IVRCSM and **IVRCP**: The IVR specific objects are IVR-CSM (IVRCSM) and IVR-CP (IVRCP) are adaptations of the Communication Session Manager (control) and CP (Connection Performer) specific to the IVR. (Moodley and van Olst 2011)

4.3.2 Consumer Domain specific objects

UAP: The User Application (UAP) initiates calls and services. (Moodley and van Olst 2011)

TCSM: The Terminal Communication Session Manager (CSM) is responsible for setup of the terminal part of a connection. (Moodley and van Olst 2011)

TLA and **TCP**: The Terminal Layer Adaptor (LNA) is responsible for the connection within the terminal. Its function is to set up the connections via the Terminal Connection Performer (TCP) within the terminal. (Moodley and van Olst 2011)

4.4 Organising Information Principles (RM-ODP Information Viewpoint)

The RM-ODP information viewpoint details the organising information principles. It details the structure and content of the information as well as how the information is managed within the OSA/Parlay Network Architecture. There are three broad categories for information requirements (Moodley and van Olst 2011):

i. The Logical Connection Graph (LCG);
ii. The Physical Connection Graph (PCG); and
iii. The Network State Information (NSI).

The LCG and PCG are associated with the CSM and CC respectively, whilst the NSI is associated with the LNC. The CSM and CC are session orientated objects. A session is a temporary allocation of resources that are configured to service a task for a period of time (TINA NRA 1997). Hence the LSG and PCG are also session orientated and are hence provisioned for the respective task.

4.4.1 Logical Connection Graph

The LCG is associated to the CSM to specify the call connection requirements; it is a logical representation of the connection in the network connectivity domain. The value of the connection graphs is that these is a technology independent description of the service. It is a logical representation of the media flows between call parties in a call. The LCG provides the relationship between the service session (client) and the connectivity session (server). The class diagram is presented in Figure 4-4 below. The Call object is described by the data characteristics: a unique identifier; call bandwidth requirements; QoS requested and call state information is tracked. The Call Legs are described by the data characteristics: a unique identifier, location of the user in the network, user details and call leg state information is tracked. (Moodley and van Olst 2011)

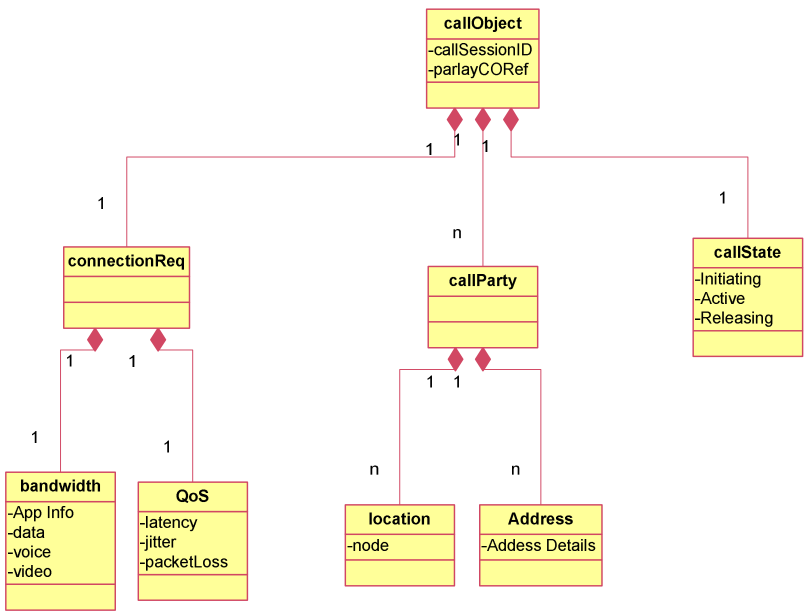

Figure 4-4: Logical Connection Information Objects

4.4.2 Physical Connection Graph

The PCG is associated with the CC and is used to specify end-to-end connectivity within the network. Similarly, the PCG is technology independent. The PCG contains details of the specific access node where the terminal connections access the network in particular the LNC nodes, see Figure 4-5. It describes the network transport links and transit nodal points to effect the connection within the network which are identified by the path through the network consisting of one or more LNCs, links and nodes within each LNC are identified. The PCG also keeps track of the physical connection state information as the LCG is translated into a physical connection. It should be noted that the terminal domain should provide a similar PCG mapping of the physical devices. The terminal information requirements are outside the scope of this research. (Moodley and van Olst 2011)

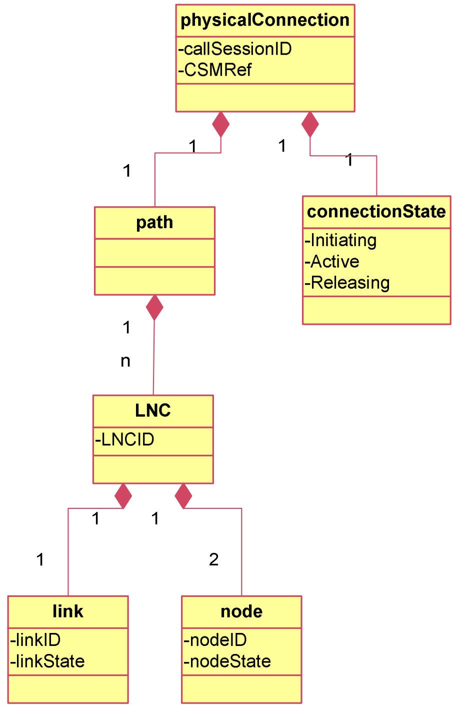

Figure 4-5: Physical Connection Information Objects

4.4.3 Network State Information

The information requirements associated with the LNC, unlike that of the CSM and CC, are not session orientated. The NSI keeps track of the real time state of the network. The OSA/Parlay Network Architecture consists of standard modular design to be used for the core transport network, this is detailed in section 4.6. The state information that each of the 6 transport links and the 4 network nodes are updated in the NSI, see Figure 4-6. Using this design approach ensures that the connections admitted into the network will be provisioned with a guaranteed QoS for the duration of the call (Moodley and Hanrahan 2008). The NSI describes the state of the Core Transport Network by decomposing it into various domains controlled by LNCs. The performance and state information of the links and nodes in each LNC are recorded. Thus both the network topology as well as performance of nodes and links information is available within the NSI information object. (Moodley and van Olst 2011)

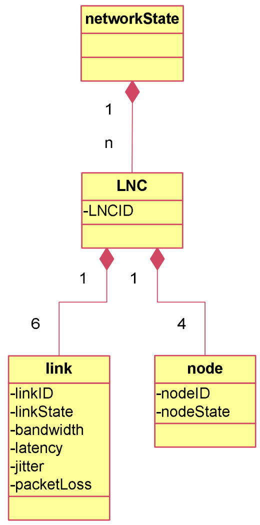

Figure 4-6: Network State Information Objects

4.5 Detailed OSA/Parlay Network Architecture (RM-ODP Computational Viewpoint)

The RM-ODP computational viewpoint enables the distributed properties of the architecture by further decomposition of the enterprise objects. These computational objects will interact with each other through well-defined interfaces. This viewpoint describes the functions of the objects together with the object interfaces.

The computational viewpoint enables the properties of scalability, load control, location and administrative management of the network. Figure 4-7 depicts the computational objects together with conceptual groupings.

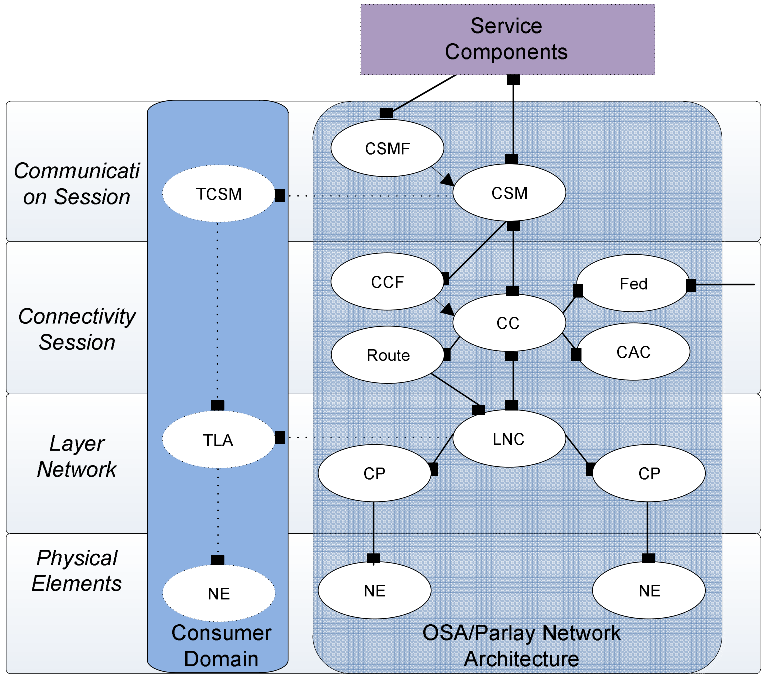

Figure 4-7: OSA/Parlay Computational Viewpoint, adapted from (Moodley and van Olst 2011)

4.5.1 Communication Session Objects

The communication session renders the relationship between the service session (client) and the connectivity session (server) using the LCG. The communication session objects are session orientated and thus include a factory object, communication session manager factory (CSMF) used to instantiate the CSM. (Moodley and van Olst 2011)

The CSM allows service clients to setup, activate, release and send notifications of connections within a communication session. The CSM maps the logical connections into physical connections in the connectivity session.

4.5.2 Connection Session Objects

The connectivity session presents a technology independent view of the core network connections. This session is used to create the network portion of the physical connections. This view is physically realized by mapping it to layers within the core network. Similar to the communication session, the connectivity session includes a factory object, the connection coordinator factory (CCF) and is used to instantiate the CC. The CC is session orientated and therefore used to configure resources to achieve a specific objective or task. The CC allows the communication client to setup, manipulate and release network connections and receive notifications. (Moodley and van Olst 2011)

The CC is supported by the Intelligent Routing (Route) object and determines the terminal locations from a location register. It determines the best route based on the dynamic network state and location of the terminals. The Call Admission Control (CAC) is responsible for determining whether the requested call properties (QoS) can be supplied by the transport network and on that basis, it does a call admission control function. It then identifies specific layers and selects the appropriate Layer Network Coordinator (LNC) for connection setup. Further details of the CAC functionality are given in section 4.6. The Federation (Fed) object is responsible for managing the Administrative Federation on a call by call basis. (Moodley and van Olst 2011)

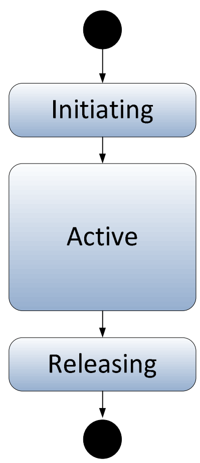

Figure 4-8: Finite State Machine of the Connection within the OSA/Parlay Network Architecture

Finite state machines (FSM) are used to keep track of the state of a connection, which provides the facilities to instantiate advanced services mid-connection. These FSMs are detailed in Figure 4-8 below. The TCSM is responsible for establishing the end terminal part of the logical connection. The scope of which, is not part of this research. (Moodley and van Olst 2011).

4.5.3 Layer Transport Network Objects

The LNC services the connectivity layer by establishing the physical connections in the core network. Unlike the CSM and CC, the LNC is resource orientated and therefore exists regardless of sessions. The LNC services the connection sessions by establishing physical connections within the network. It keeps track of network state information in the various modular layer transport network components. The LNC provides network state information on frequent intervals to the CAC. This ensures that admit or reject decisions are made timeously. (Moodley and van Olst 2011)

The CP represents the physical elements in the network, such as MPLS or IP routers. We assume that the architecture at the layer transport network level interacts with OSI level 3 network elements. The assumption is that there is a client-server relationship for the underlying layers and thus does not need to be managed. (Moodley and van Olst 2011)

4.6 Call Admission Control

Guaranteed QoS has been a long sought after goal for packet switched technology (Song et al 2008). QoS is the assurance that prior to its admission, a new connection awaiting admission into the network is guaranteed a level of performance within the network for the entire connection duration (Moodley and Hanrahan 2007). The CAC entity needs to determine whether the network has sufficient bandwidth and demonstrates acceptable performance parameters in terms of delay, jitter and packet loss to service the connection request (Moodley and Hanrahan 2007; Gylterud and Tognon 2003; Stankiewicz et al 2011).

In addition, CAC provides a secondary function ensuring that existing connections will not deteriorate in quality due to the new connection entering the network (Moodley and Hanrahan 2006). Capacity dimensioning ensures that all parts of the network have sufficient capacity to cater for the maximum Erlang traffic during the normal operations of the network (excluding events such as New Year's Eve).

Three important components are essential for guaranteed QoS. These are: (i) the intelligent routing of connections in the Packet Switched Network (PSN); (ii) dynamic information of the network state; and (iii) making the call admission control decision on whether the network can meet or exceed the QoS requested.

Intelligent routing must go beyond the traditional routing protocols. Popular shortest path based routing protocols such as Open Shortest Path First (OSPF) is a case in point based on the Dijkstra's Algorithm. The algorithm is popular due to its simplicity. The basis of operation for the routing protocol is as follows.

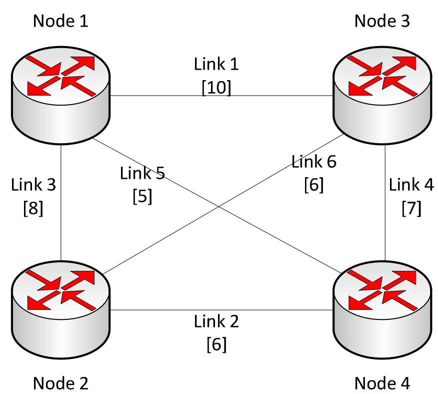

Figure 4-9: Simple 4 node meshed network

A simple meshed four node network is presented in Figure 4-9. On configuring each link, a weight is allocated usually based on distance between nodes. This weight is usually fixed and is hardcoded onto the node. The Dijkstra algorithm computes the shortest path between source and destination. Assume that the source node is *Node1* and destination node is *Node3* together with the following link weights in Table 4-3:

Link	**Weight**
Link1	10
Link2	6
Link3	8
Link4	7
Link5	5
Link6	6

Table 4-1: OSPF Link Weights

There are multiple routes from source to destination node, listed in Table 4-2:

Paths (source to destination)	Path Weight
Link1	10
Link5 – Link 4	12
Link3 – Link 6	14
Link3 – Link 2 – Link 4	21

Table 4-2: Possible Paths from Source to Destination

The shortest path algorithm chooses the path with the lowest weight. However, there are two significant disadvantages. First, any traffic requiring routing form *Node1* to *Node3* will always be routed through Link1 being the shortest path. This leads to significant overloading of the link and thus excessive latencies and packet losses. Second, when either links or nodes fail, the routing protocol needs to re-converge the entire network topology before any routing decisions can be made. Therefore the weakness is that it does not take into account the dynamic state of the network.

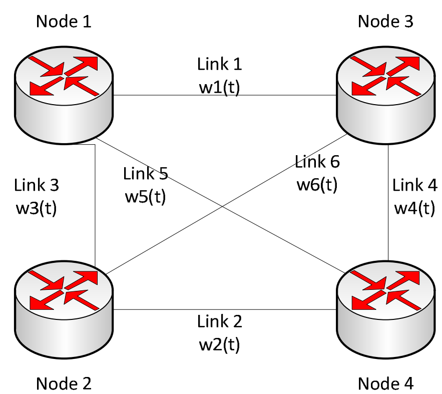

Figure 4-10: Simple 4 node meshed network with dynamic link weights

A unique contribution of this research is principled upon making intelligent routing decisions based on the dynamic state of the network. In order to make intelligent routing decisions the dynamic state of the network links is required, see Figure 4-10. The flowchart depicted in Figure 4-11(ii) describes the logic of obtaining near real time network state information. A timer triggers a probe into the network's state. The network consists of a number of standard modular sub-networks designed to expand the network in a scalable manner, see Figure 4-11(i). Each sub-network is polled for an update on its state by inserting probe traffic into each of the 6 links. The probe traffic is analysed to determine QoS parameters such as latency, jitter and packet loss. This performance information per link is used to update the network state. The next sub-network is similarly probed and its performance recorded. This continues recursively until the entire network has been polled. Therefore the network state with details about each network link is available in a regular manner.

In order to guarantee QoS for a call request the following is required, see flowchart in Figure 4-11(iii). A new call request triggers this algorithm. The topology and performance of the network is retrieved for that point in time. Consider at that point in time, either a link performing poorly under rain fade or node failure conditions which are taken into account through the network topology and performance information. These parameters together with the location of the source and destination nodes are used as input into the Dijkstra algorithm. The adapted Dijkstra algorithm is used to compute the best path based on dynamic network performance. The performance of the potential path is then compared to the call request parameters. Should the performance of a network path meet or exceed the requested performance, the path is selected and setup in the network. Should the potential path not meet the call requested criteria, the call may be rejected and the maximum performance of the best path is forwarded to the service stratum to negotiate Application level performance decisions.

The QoS mechanism described is a unique contribution of this research which is applicable to any packet switched network. It can be applied to existing transport technologies such as IP or MPLS routers as well as any future packet based transport technology. Unlike existing routing algorithms, load balancing is inherently provisioned. Consider two identical possible paths from source to destination. One path may be loaded slightly higher than the alternative. However, when a new call request is studied, the alternative will have better performance. Thus the call will be routed to the alternative. This novel CAC mechanism takes into account changes in network topology such as link or node failure without the need to re-converge routing tables. Similarly, loading of the links and dynamic state of links in the network are considered when making the routing decision.

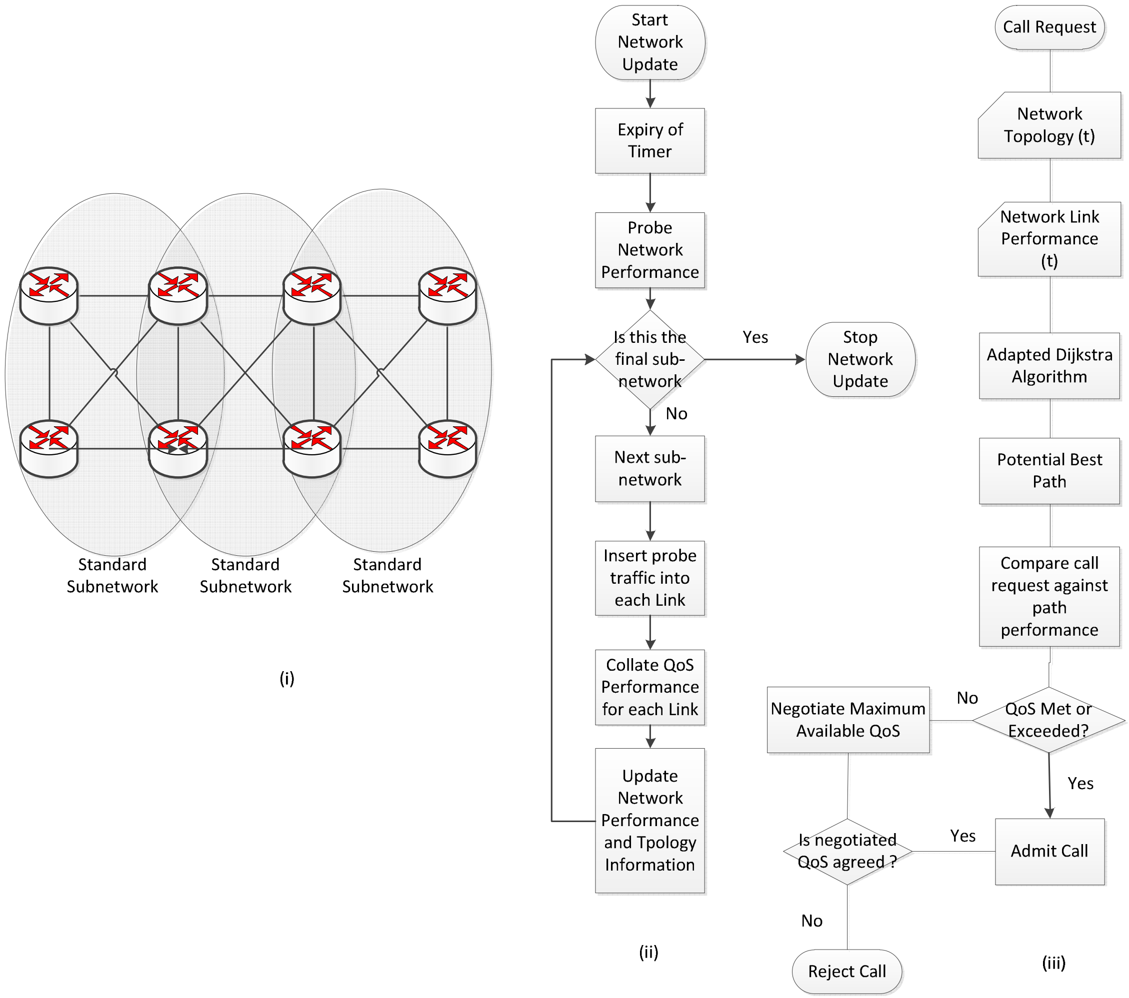

Figure 4-11: (i) standard modular subnetwork configuration (ii) Flowchart of the Network State Update (iii) Flowchart for Call Admission Control

The remainder of section 4.6 describes the detailed functions of the CAC mechanism:

- Standard modular layer transport network design;
- Periodic probe of the Network QoS Performance;
- Route Selection for a new Call Request;
- Call Admission Control; and
- Edge Enforcement

4.6.1 Standard modular layer transport network design

A standardised modular design philosophy is proposed to be adopted for the core transport network. In order to avoid creating large complexities in the information requirements of the architecture, it is proposed that a simple model of configuration of the physical equipment from the sub-layered network be maintained throughout the transport network layers (see Figure 4-12). The model would be configured from the CPs up to the CAC object. It ensures that a single information structure be used throughout the architecture. (Moodley and van Olst 2011)

The proposed standard modular design provides an organised manner to deal with the complexity of a large network. The topology of the network is easily ascertained and each link and node is uniquely identified while shared nodes and links are given dual identifiers, see Figure 4-12. Therefore Link1 in LNC A is uniquely identified by *LNC A, Link 1*. While the shared node between LNC A and LNC B is identified by either *LNC A, node 3* or *LNC B, node 1*. The periodic probing of the performance of the network (see section 4.6.2) is done in a simple organised manner.

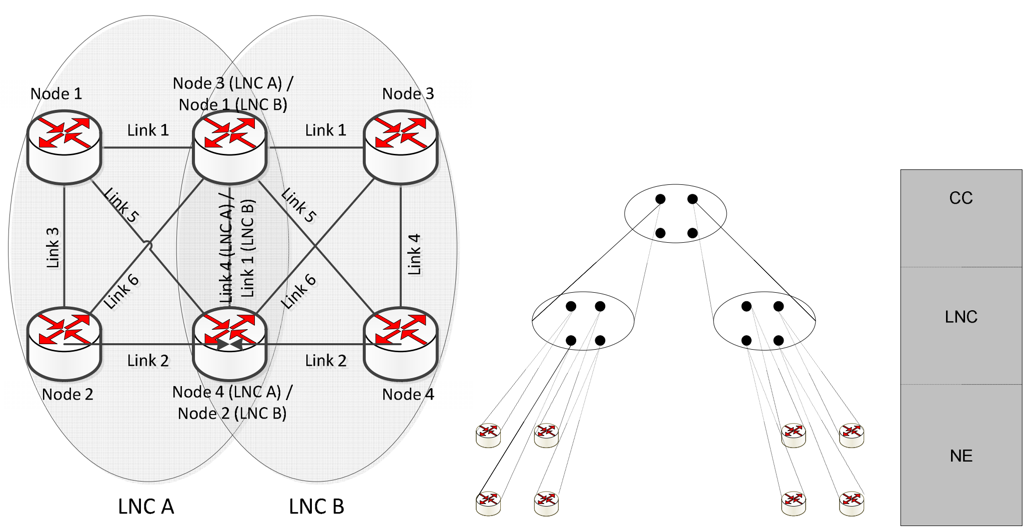

Figure 4-12: (i). example of a standard modular design (ii). consolidated information model

4.6.2 Periodic probe of the Network QoS Performance

QoS guarantees that the quality parameters such as latency, jitter and packet loss are within tolerance limits for the specified classes of traffic traversing the network such as time critical and non-time critical class traffic (Stankiewicz et al 2011, Yun and Perros 2010). The QoS performance metrics are defined below (Moodley and Hanrahan 2007):

- **Latency** is the time taken for a packet to traverse the network from the source node to the sink node.
- **Jitter** is defined as the average standard deviation from the mean (latency) of the observed data.
- **Packet Loss** is the number of packets dropped when transmitted from source to sink. Expressed as a percentage.

Periodically each node will inject probe traffic of a particular type of application for example voice traffic. The receiving node will record the received probe traffic and will analyse the QoS parameters of latency, jitter and packet loss. This information will update the NSI Information Object with the performance parameters (see section 4.4.3) for the entire network. The results of which will be used by the CAC object to determine the admission or rejection decision.

4.6.3 Route Selection for the Call Request

In order to make a routing decision, the new connection request will specify the location of terminals of the connection. The CAC will make use of an adaptation of the Dijkstra Algorithm to determine the best path based on dynamic network performance. Dijkstra's Algorithm is commonly implemented in the ICT industry to determine the popular shortest path in a network between a specified source and destination. In particular routing protocols such as IS-IS and OSPF have adopted the Dijkstra Algorithm. These protocols implement the algorithm using static values for weighting the links in the algorithm, typical static distance values are commonly used. There are two significant disadvantages for this implementation. The first disadvantage is that as traffic enters this network, the shortest link will always be chosen, hence this link can become congested whilst the longer link does not have traffic routed to it. Thus the implementation does not take into account the state of the network such as loading of links to make routing decisions (Wang et al 2011). The second disadvantage is whenever there are network state changes such as a link (or element) or router (node or vertix) failure; the routing protocol will need to re-converge on the new network topology before routing decisions can be made. Thus the key disadvantages are that the routing protocols based on the Dijkstra Algorithm are suited to the steady state conditions of the network rather than the dynamic nature of the network (Wang et al 2011).

The adaptation of Dijkstra Algorithm is as follows: the weights of the links in the network (elements) will use line state information of either: latency, jitter or packet loss; these parameters are dynamically updated periodically (see section 4.4.3) thus the algorithm parameters are constantly updated with the dynamic network state information and network topology. For a time dependant application such as voice, the latency parameter will be used in the Dijkstra algorithm for the weighting of links. Thus the algorithm will calculate the shortest path in terms of dynamic latency from the source to destination node and not based on static distance values.

Thus the proposed path is the best performing route in the network at that particular point in time for the connection request. The standard network module philosophy allows QoS metrics from each module to be aggregated for the particular session connection, such that the latency, jitter and packet loss metrics are computed for the proposed path. Thus the QoS for any particular session connection can be rapidly ascertained.

4.6.4 Call Admission Control

The CAC receives input of the routing decision which is the proposed path together with the specific performance detailing latency, jitter and packet loss. This is to ensure that the connection route that will be chosen for the connection request will meet or exceed the associated QoS request of the new call entering the network notwithstanding the existing loading and traffic traversing the connection path.

The CAC logic will determine from the requested QoS, whether the proposed path in the network can meet or exceed the minimum requested requirements for all QoS parameters. The performance of the potential path is then compared to the call request parameters. Should the performance of a network path meet or exceed the requested performance, the path is selected and setup in the network. Should the potential path not meet the call requested criteria, the call may be rejected and the maximum performance of the best path is forwarded to the service stratum to negotiate Application level performance decisions. The CAC decision is communicated to the OSA/Parlay Gateway via the CSM.

4.6.5 Edge Enforcement

In order to assure guaranteed QoS within the transport network, unauthorised and rejected connections are barred from entering the network. Similar to the 3GPP LTE concept, the proposal is that connection requests are allocated an IP address, which entitles access to the area of the network that only administers terminal registration and service requests. Once a connection request is admitted to the network, a new IP address is allocated to the end terminal, provisioning access into the core transport network.

4.7 OSA/Parlay Network Interface

The OSA/Parlay Network Interface is the mechanism which separates the Call and Service Control (OSA/Parlay Gateway) from the transport network (OSA/Parlay Network Architecture) see Figure 4-13. The interface provides a mechanism in which the service and control layer can leverage the features and functionality of the OSA/Parlay Network Architecture. The interface is specified using an API further specified into a number of methods. The API is technology independent and hence future proof. It supports end-to-end connections as well as connections to special resources such as a media gateway or interactive voice response unit. The methods and parameters are described below.

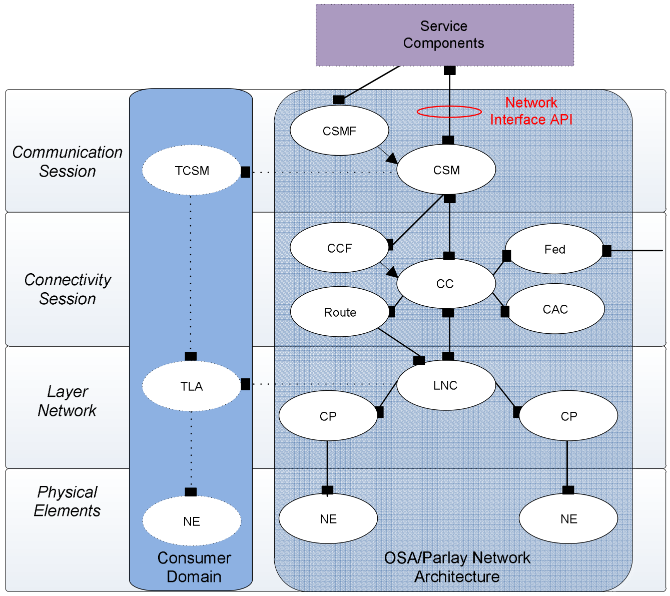

Figure 4-13: OSA/Parlay Network Interface

- **createCall()**

The OSA/Parlay Gateway prepares the OSA/Parlay Network Architecture for a new call by invoking the createCall() method. The architecture in turn creates a Connection Session Manager (CSM) object to deal with the new call session. The architecture allocates a unique identifier for the new call which is used by the gateway for all advanced call and service control actions. The reference to the newly created CSM Object interface is to be placed in the Broker service. The object reference together with the unique identifier is used for all further advanced call and service control actions.

Parameters	*Element Type*	*Optional*	*Element Description*
appInterface	Object	No	Specifies the Gateway Object interface for callbacks from the call created.
callSessionID	TpSessionID	No	**Returns** the CSM sessionID of the call created in the OSA/Parlay Network Architecture.

Table 4-3: createCall method

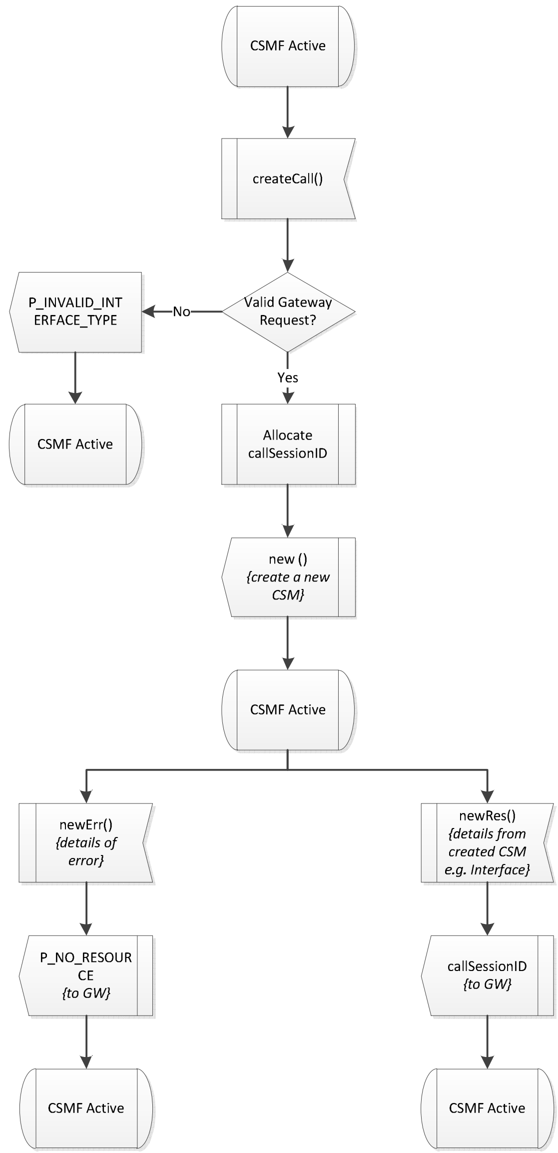

Figure 4-14: createCall SDL

- **createUICall()**

The OSA/Parlay Gateway prepares the OSA/Parlay Network Architecture for a new IVR call by invoking the createUICall() method. The architecture in turn creates a Connection Session Manager (CSM) object to deal with the new call session. The architecture allocates a unique identifier for the new call which is used by the gateway for all advanced call and service control actions. The reference to the newly created CSM Object interface is to be placed in the Broker service. The object reference together with the unique identifier is used for all further advanced call and service control actions. Thereafter the CSMF Object routes the call to the IVR using the routeReq () method (see below). The target address of the IVR is used.

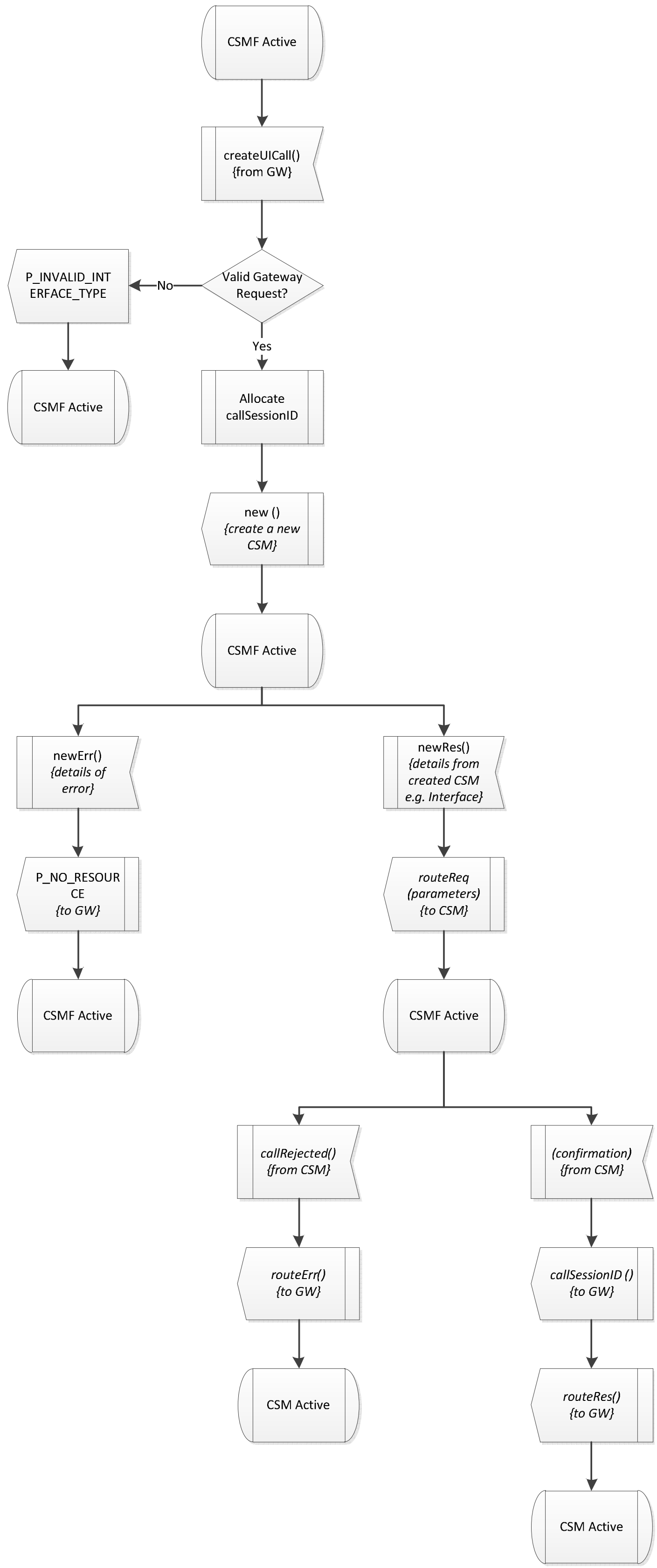

Figure 4-15: createUICall SDL

Parameters	*Element Type*	*Optional*	*Element Description*
appInterface	Object	No	Specifies the Gateway Object interface for callbacks from the call created.
originatingAddress	TpAddress	No	Specifies the call party address
appInfo	TpCallAppInfoSet	Yes	Details the Application such as voice or video
connectionPropertie s	TpCallLegConnectionP roperties	No	Specifies the connection properties including the QoS request
callSessionID	TpSessionID	No	**Returns** the CSM sessionID of the call created in the OSA/Parlay Network Architecture.

Table 4-4: createUICall method

- **routeReq()**

The method is invoked to setup a 2-party call within the network. The request contains the QoS requirements, originating and target addresses are passed to the OSA/Parlay Network Architecture. The request is essentially the CSM LCG details towards the CC. Additionally, the unique identifier for the call allocated by the createCall () method (or createUICall() method) provides confidence that the request is intended for this unique call. The arming of criteria will be facilitated separately by the createNotification() or changeNotification() methods. If the call is successfully admitted into the network, the RouteRes () is returned to the gateway. While if the call is rejected, the RouteErr () method is returned to the gateway.

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.
targetAddress	TpAddress	No	Specifies the destination party to which the call should be routed
originatingAddress	TpAddress	No	Specifies the address of the originating party.
connectionPropertie s	TpCallLegConnectionP roperties t	No	Specifies connection-related information (such as bandwidth and QoS level)
appInfo	TpCallAppInfoSet	Yes	Details the Application such as data, voice or video

Table 4-5: RouteReq method

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.

Table 4-6: RouteRes method

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.
errorIndication	TpCallError	No	Specifies the error which led to the original request failing
path	TpString	Yes	Application related QoS – best offered path
latency	TpInt32	Yes	Application related QoS – best offered latency
jitter	TpInt32	Yes	Application related QoS – best offered jitter
loss	TpInt32	Yes	Application related QoS – best offered loss

Table 4-7: RouteErr method

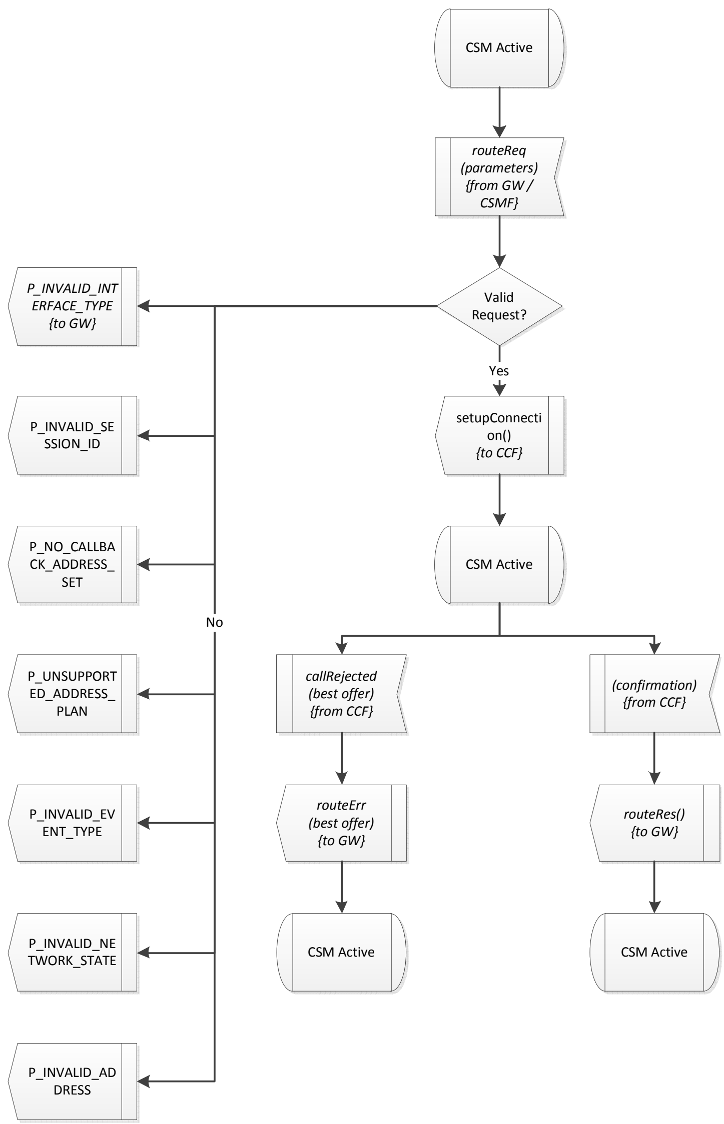

Figure 4-16: routeReq SDL (together with routeRes and routeErr SDL)

- **releaseCall()**

The gateway invokes this method to release the call and all associated network resources. This confirmation of the release condition is indicated by the callEnded () method.

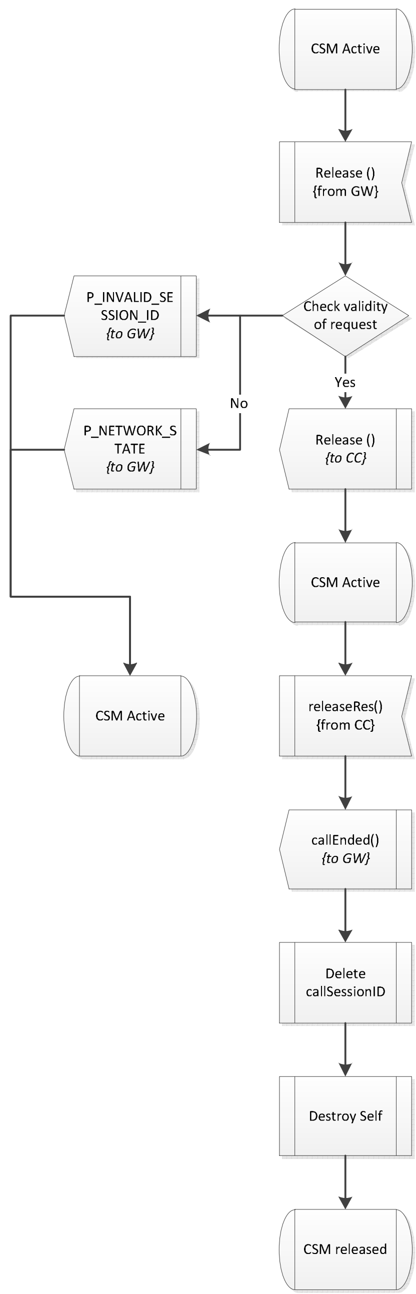

Figure 4-17: release SDL

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.
Cause	TpReleaseCause	No	Specifies the cause of the release

Table 4-8: releaseCall method

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.
Report	TpCallEndedReport	No	Specifies the reason the call is terminated.

Table 4-9: callEnded method

- **releaseCallLeg()**

The method is invoked to release the party from a multiparty call, although outside the scope of this research, it is included as the interface must cater for multiparty calls. The associated network resources to the specific party are released. If the call leg is only one of two parties participating in the call, the entire call is released and callEnded () method is forwarded to the gateway. If the call leg is in a multiparty call, the callLegEnded () method is sent to the gateway.

Parameters	*Element Type*	*Optional*	*Element Description*
callLegSessionID	TpSessionID	No	Specifies the unique call leg session ID of the call.
Cause	TpReleaseCause	No	Specifies the cause of the release

Table 4-10: releaseCallLeg method

Parameters	*Element Type*	*Optional*	*Element Description*
callLegSessionID	TpSessionID	No	Specifies the unique call leg session ID of the call.
Report	TpCallEndedReport	No	Specifies the reason the call leg is terminated.

Table 4-11: callLegEnded method

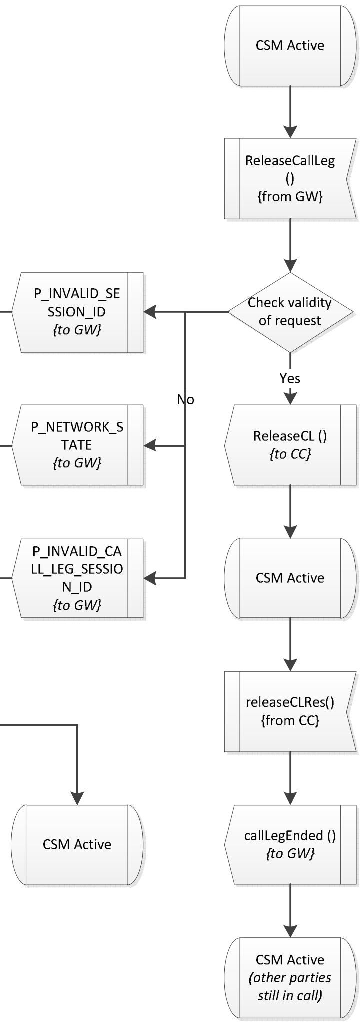

Figure 4-18: releaseCallLeg SDL

- **createNotification()**

This method enables the OSA/Parlay Gateway to set or arm triggers based on the call state of a connection. The architecture reports to the gateway when a specific call conditions occur. A typical advanced service is the popular freephone application (or 0800 service). Once the call address is analysed, the condition is reported to the advanced service control. The advanced service control redirects the call to a new calling destination. This arming of detection events is requested using the createNotification () method and stored on the PCG information object. A unique notificationID for the createNotification() method is allocated and returned to the OSA/Parlay Gateway. When the armed event takes place the OSA/Parlay Network Architecture informs the OSA/Parlay Gateway call back object using the reportNotification () method together with the unique notificationID.

Parameters	*Element Type*	*Optional*	*Element Description*
appInterface	Object	No	Specifies the Gateway Object interface for callbacks from the call created.
notificationRequest	TpCallNotificationRequest	No	Specifies the event specific criteria used to arm trigger detection points.
callSessionID	TpSessionID	No	**Returns** the unique identifier for the notification request from the OSA/Parlay Gateway

Table 4-12: createNotification method

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	**Returns** the CSM sessionID of the call created in the OSA/Parlay Network Architecture.
callLegReferenceSet	TpCallLegIdentifierSet	No	Specifies the set of all call leg references in the call
notificationInfo	TpCallNotificationInfo	No	Specifies the details of the event that was triggered
assignmentID	TpAssignmentID	No	The unique identifier for the notification request from the OSA/Parlay Gateway

Table 4-13: reportNotification method

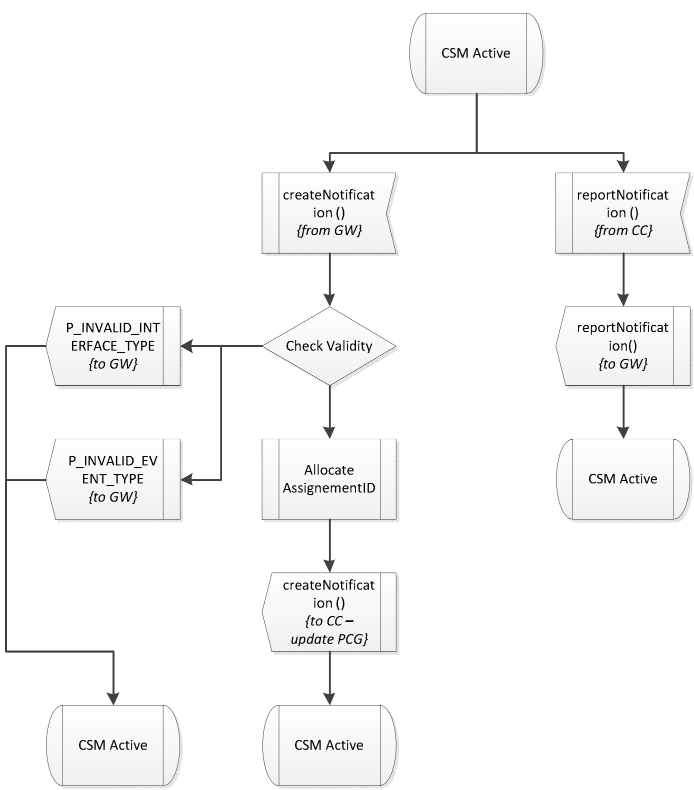

Figure 4-19: createNotification SDL

- **changeNotification()**

The changeNotification() method requested by the gateway to indicate the set of arming criteria has changed. The changes are indicated within the method. All stored criteria in the PCG information object will be replaced with the criteria specified in the changeNotification() method.

Parameters	*Element Type*	*Optional*	*Element Description*
assignmentID	TpAssignmentID	No	Specifies the unique identifier for the initial notification request
callSessionID	TpSessionID	No	The unique call identifier
notificationRequest	TpCallNotificationRequest	No	Specifies the new event specific criteria used to arm trigger detection points.

Table 4-14: changeNotification method

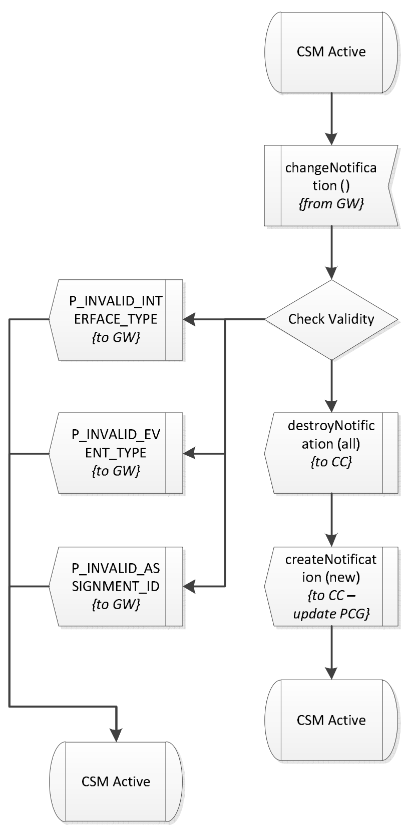

Figure 4-20: changeNotification SDL

- **destroyNotification()**

The destroyNotification() method is applied by the gateway to remove all armed criteria for the connection. All notification criteria stored in the PCG information object is deleted.

Parameters	*Element Type*	*Optional*	*Element Description*
assignmentID	TpAssignmentID	No	Specifies the unique identifier for the initial notification request
callSessionID	TpSessionID	No	The unique call identifier

Table 4-15: destroyNotification method

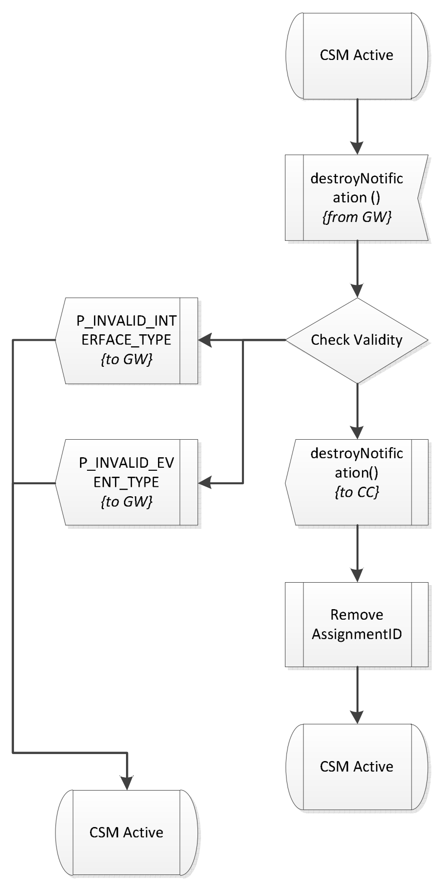

Figure 4-21: destroyNotification SDL

- **continueProcessing()**

The method continues normal processing of the call. The gateway invokes this method after being notified of an armed event and advanced call processing is complete or not required.

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.

Table 4-16: continueProcessing method

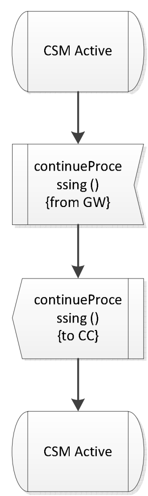

Figure 4-22: continueProcessing SDL

- **callAborted()**

The method notifies the gateway that the call has been abnormally terminated in the network architecture, no further interaction is possible.

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionId	TpSessionID	No	Specifies the unique call session ID of the call.
error	TpCallError	No	Details the reason for abnormal termination

Table 4-17: callAborted method

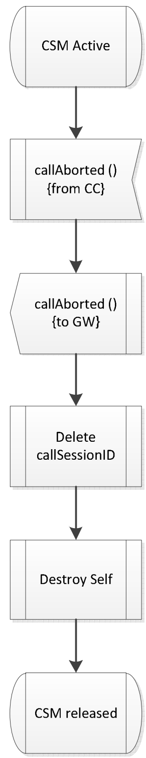

Figure 4-23: callAborted SDL

- **sendInfoAndCollectReq ()**

The method is used to pass controls and collect information to and from the Interactive Voice Response (IVR) Unit. The results of the interaction with the IVR is returned using the sendInfoAndCollectRes() method whilst an error in the operation returns the sendInfoAndCollectErr() method.

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.
userInteractionSessionID	TpSessionID	No	Specifies the unique IVR ID
info	TpUIInfo	Yes	Specifies additional IVR related information
language	TpLanguage	No	Specifies the language which announcement should be played
variableInfo	TpUIVariableInfoSet	Yes	Additional UI information
criteria	TpUICollectCriteria	No	Specifies if any digits needs to be collected
responseRequested	TpUIResponseRequest	Yes	Specifies if the message is an announcement or whether a response is to be collected from the called party

Table 4-18: sendInfoAndCollectReq method

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.
userInteractionSessionID	TpSessionID	No	Specifies the unique IVR ID
response	TpUIReport	Yes	A report provided by the IVR unit
collectedInfo	TpString	Yes	Any digits or voice recording collected from the session

Table 4-19: sendInfoAndCollectRes method

Parameters	*Element Type*	*Optional*	*Element Description*
callSessionID	TpSessionID	No	Specifies the unique call session ID of the call.
userInteractionSessionID	TpSessionID	No	Specifies the unique IVR ID
error	TpUIError	Yes	Error details

Table 4-20: sendInfoAndCollectErr method

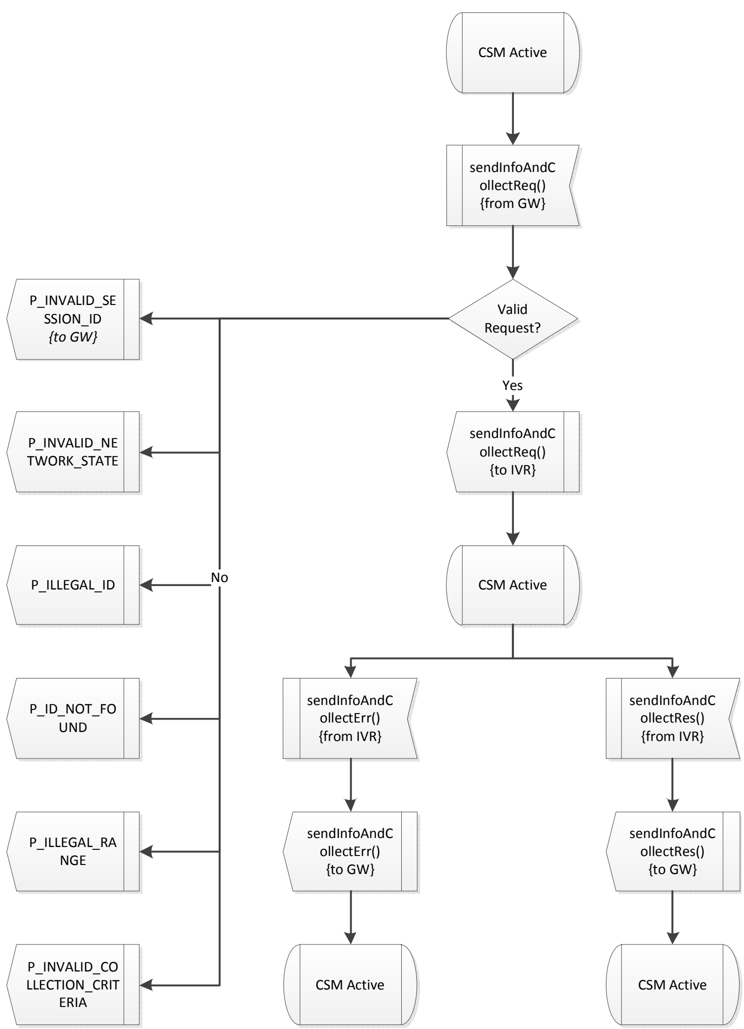

Figure 4-24: sendInfoAndCollectReq SDL

4.8 Detailed functionality of Connectivity Session Objects

The Connectivity Session objects are responsible for the coordination of the physical connections within the layer transport network to realise the logical connection graph (LNC) from the CSM. Similar to the exposed open network interface, the internal interfaces between the architectural objects are API based. The key functions of the connectivity session objects are described below together with SDL diagrams respectively.

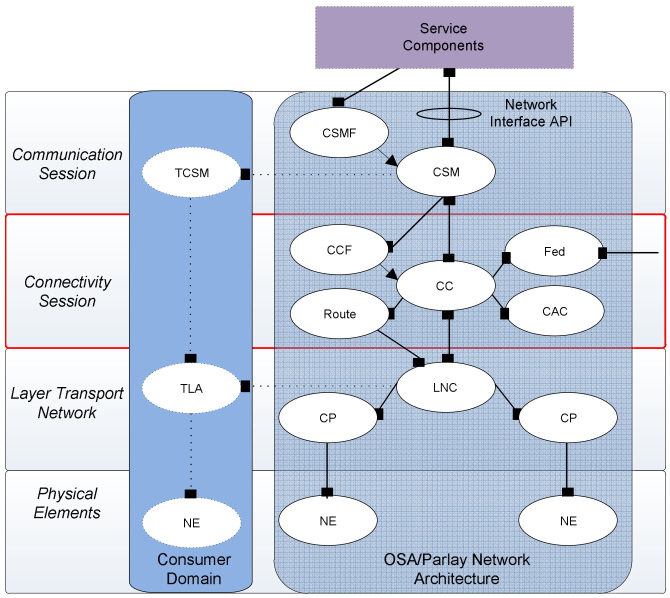

Figure 4-25: OSA/Parlay Network Architecture

- **Connection Coordinator Factory**

The prime function of the CCF object is to create unique Connection Coordinator (CC) objects for a specific call or connection request, it is essentially a factory object. The CCF receives a request from either the CSM or Fed object requesting a connection to be setup in the network, the request is a translation of the LCG containing the connection QoS performance request. The CCF object proceeds to create a unique CC object dedicated to the call or connection. Upon receiving confirmation of the CC instantiation, the CCF passes the connection setup request to the CC. The CC object orchestrates the connection in the physical network.

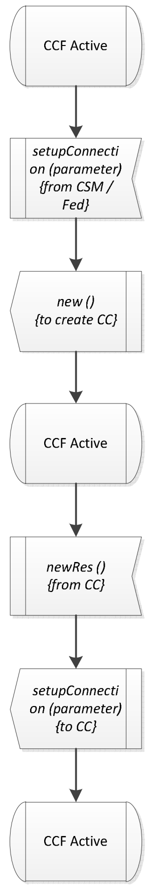

Figure 4-26: CCF setupConnection ()

- **Connection Coordinator**

The CC is responsible for the overall coordination of connections in the OSA/Parlay network to realise the physical connection. The CC co-ordinates with connectivity session objects to realise the admission of a new connection request into the transport network. Similarly, the CC co-ordinates the release of transport network resources on the connection termination. Additionally, the CC provides key functions of reporting on the state of the connection allowing for advanced call handling. The abnormal termination of the call in the transport network is managed by the CC.

The CC receives a request from the CCF to setup a connection in the transport network together with the details of the source, destination, bandwidth and QoS requirements. These details contain the LCG information from the CSM and the request is to physically realise the LCG within the transport network. The CC co-ordinates interactions between the connectivity session objects to make the connection admission decision. The CC requests the Fed object to first check whether a party is in a federated network. For a federated party, the routing destination is set to the Media Gateway. For a local party, the destination routing remains unchanged. Thereafter the route object is requested to provide the shortest route in the network based on dynamic performance of the links. These details are forwarded to the CAC to ensure that the best route can meet or exceed the connection requested performance criteria. On the call being admitted by the CAC, the route is setup in the LNC using the setupConnection method. Details of the physical connections are stored in a unique PCG associated with the connection. Should the call be rejected, the best offered performance is forwarded in the callRejected method back to the Gateway via the CSM for application level negotiation.

The connection is released by request from the CSM or Fed objects respectively. The CC retrieves the physical connection graph (PCG) and notes the LNCs and unique identifiers for the connection. The identified LNCs are requested to remove the connection (or aggregation) from the network. When the connections in each LNC have been released, confirmation is sent to the requesting object. The unique sessionID for the CC is deleted and the CC destroys itself.

The CC may receive information of an abnormal termination of the connection from the LNC object affected by the equipment or link failure. The requesting object (CSM or Fed) is informed of the abnormal condition and the aborted connection using the callAborted method. The unique sessionID for the CC is deleted and the CC destroys itself.

The CC updates the PCG on state changes in the connection. The PCG is used to arm and disarm triggers based on the connection state. The createNotifcation provides details of the triggers to arm, these triggers are updated on the PCG. In contrast, the destoryNotification request, removes all armed triggers on the PCG. Should any armed event be triggered, the reportNotification is sent to the CSM. All further call or connection processing is halted and directed from the Gateway until a continueProcessing message is received. The continueProcessing request removes the affected trigger and continues normal call operations.

The CAC object polls the network for topology changes such as link or node failure. This information is updated to the CC. The CC updates the PCG with the dynamic network state information. Should these network state changes trigger an armed event in the PCG, the CSM is informed using the reportNotification message. If no triggers are armed relating to the connection state change, the CC will proceed as normal.

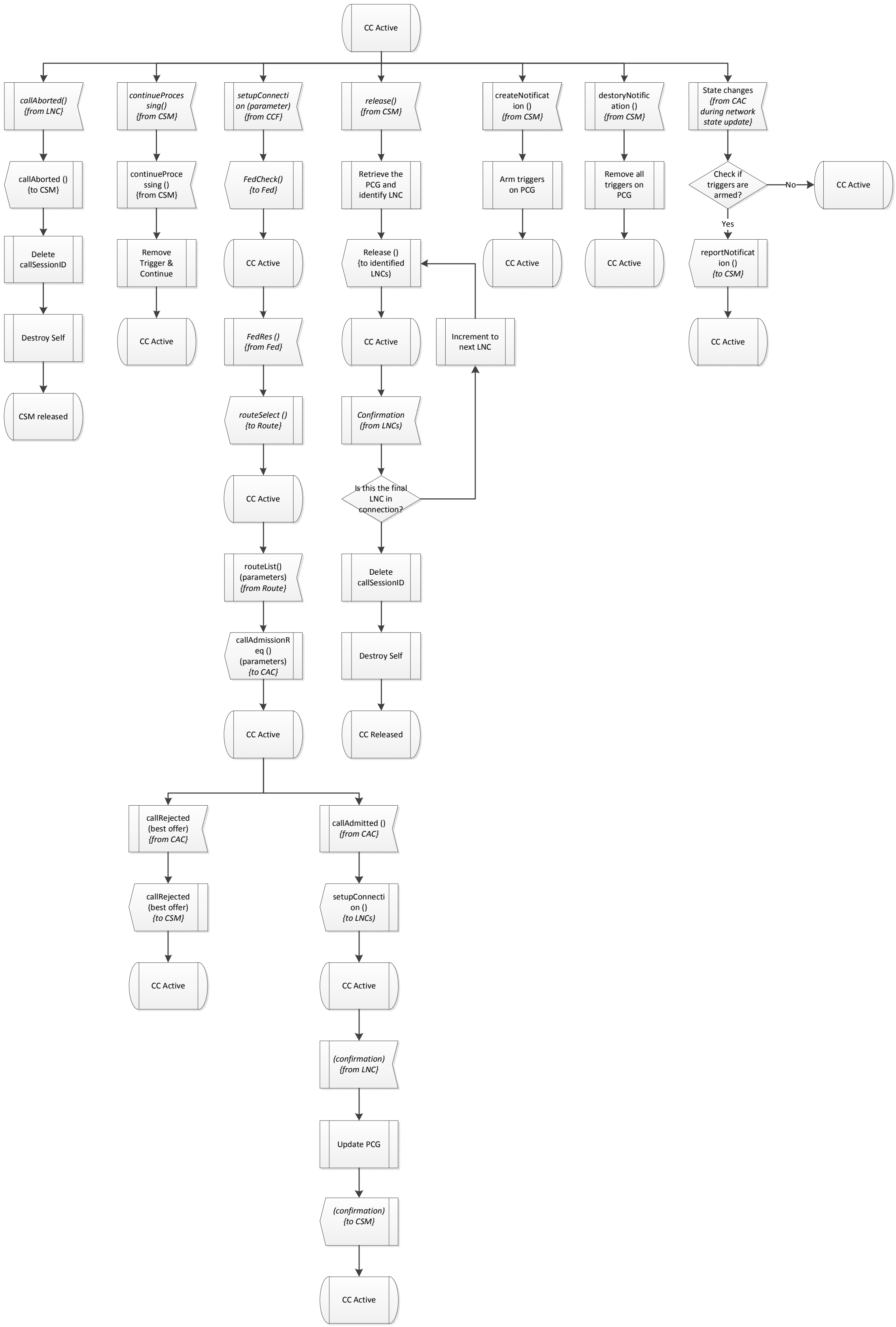

Figure 4-27: Connection Coordinator functionality

- **Fed Object**

The Fed object orchestrates the federation between the OSA/Parlay network and other teleco networks and is responsible for a federated network initiated calls as well as federated initiated releases. The initiation of a call request from a federated network (described in section 4.9), is initiated by the receipt of a createCall request from the Media Gateway. It updates the originating party's routing to the Media Gateway. The connection is co-ordinated through the CCF and by the CC using the setupConnection request. The confirmation of the unique identifier from the CCF is stored by the Fed object for future use. A federated telco can initiate a release by sending a release() to the Fed object. The unique identifier for the call identifies the CC to orchestrate the release of the connection in the OSA/Parlay network.

The OSA/Parlay initiated call request to the Fed object is to determine if one of the parties are in a federated network. If a party is in a federated network, the destination route is set to the Media Gateway, as the OSA/Parlay network will setup the connection from the originating party to the media gateway. Should there not be a federated party, the destination route is not changed and the result is passed to the CC object.

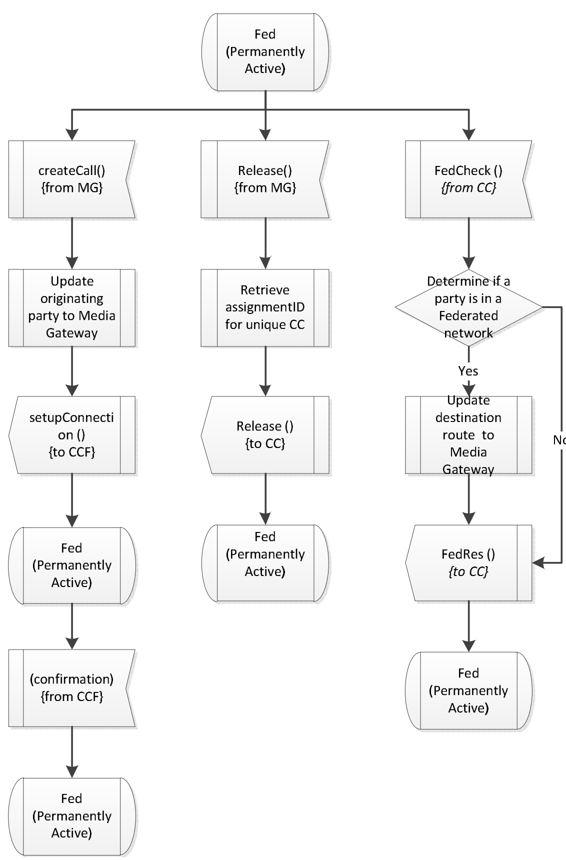

Figure 4-28: Fed Object functionality

- **Route Object**

The Route object is responsible to determine the best performing route from source to destination contained in the routeSelect request. The Route object uses the results from the CAC network state update to develop the dynamic network topology at the time of the connection request. The dynamic link latencies are updated in the network topology which is used as input into the adapted Dijkstra algorithm. The output of the Dijkstra algorithm is the shortest route from source to destination based on the dynamic state of the network (topology and link latency). The result is returned to the CC object.

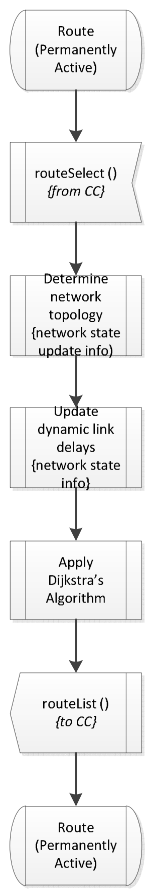

Figure 4-29: Route Object functionality

- **CAC Object**

The CAC object has two functions, the first being the admission decision for the call request waiting to be admitted into the network. The CC requests the CAC to make the call admission decision using the callAdmission () method. The CAC proceeds to determine from the selected route the details of end-to-end latency and bandwidth available. Should either the latency or bandwidth offered not meet the requested performance, the call is rejected and the best network offered performance is forwarded for application level negotiation.

The second function of the CAC is to periodically probe the entire network to determine the QoS link performance as well as any topology changes in the network. This is done by determining the number of sub-network transport layers (or LNCs). Each LNC is polled and probe traffic inserted to link to determine QoS parameters for each link. The network topology and link states are updated.

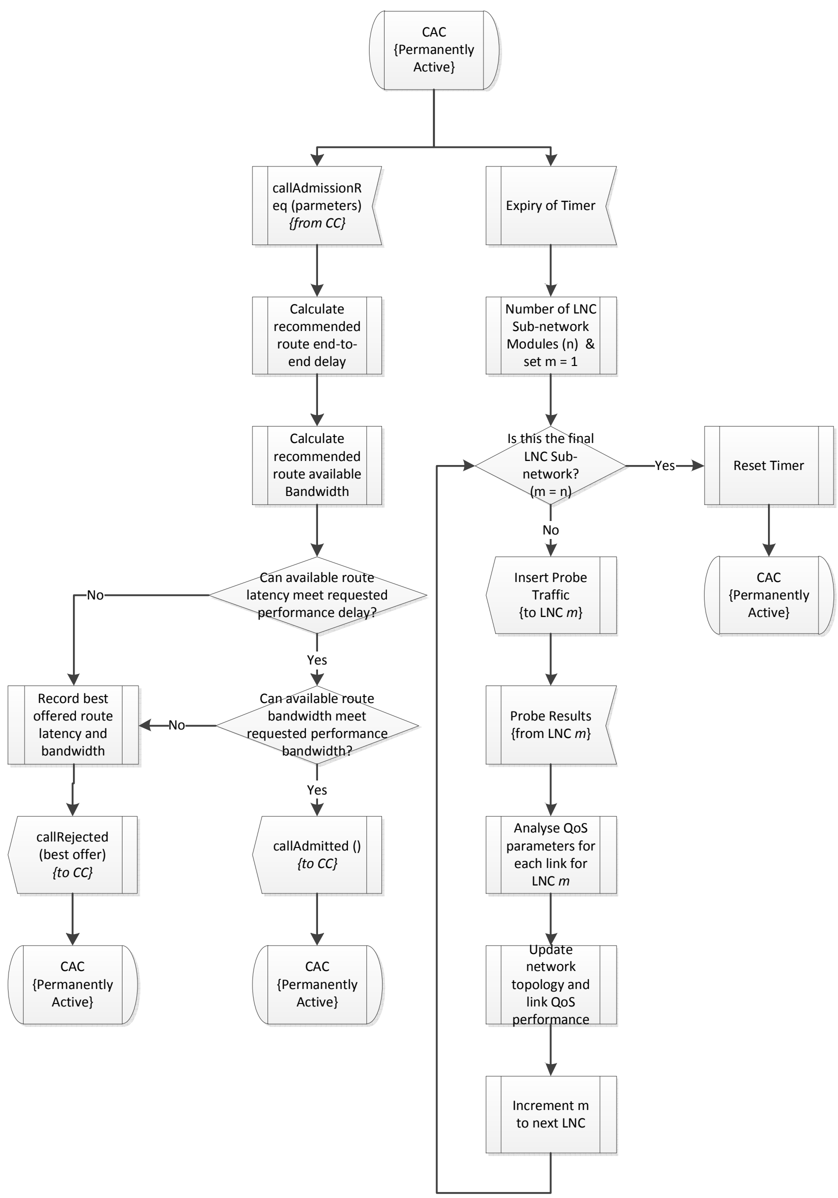

Figure 4-30: CAC Object functionality

4.9 Federation and Legacy Protocol Adaptation

Telco networks consist of a large number of assets associated with large financial investments. Due to the nature and size of the network, telcos cannot adopt a fork lift approach to replacement of technologies already deployed within the network. In addition, many technologies have a significant investment value attached and financial management requires the technology to be fully repaid before migration to newer technologies. Thus it is often the case that many telcos have to deal with the incorporation of legacy technology into the new OSA/Parlay Network Architecture. (Moodley and Hanrahan 2008).

Integration to federated networks and legacy technologies requires adaptation. These existing and legacy technologies use specific protocols such as ISUP, H.323 and SIP. We propose protocol adaptation for integration of these legacy technologies into the network architecture. Adaptation of the JAIN approach (SUN JCC2SIP 2001) which describes the SIP protocol to the Network Interface (API) mapping is extended to the ISUP and H.323 protocols. The mapping between the API and protocols are described for an Application initiated call to a federated network in Table 4-21 below. This protocol adaptation is contained in the Media Gateway allowing network federation and interoperability to legacy technologies through the Gateway Exchange. Similarly, the mapping for a federated network initiated call and release to the OSA/Parlay network is described in Table 4-22 below.

Federation can be initiated by the OSA/Parlay network or by a federated network towards the OSA/Parlay network. The OSA/Parlay initiated case for a new call request, a release of an existing call and an abnormal termination of call in the OSA/Parlay network is depicted in Figure 4-31 below. Similarly, the case for a federated initiated call request, connection of the federated call for bearer media transfer and the termination of the call initiated from the federated network is depicted in Figure 4-32 below. Illustration of the federated network interactions within a service is detailed in message sequence chats described in sections 5.4.2 and Appendices 9.1.3 - 9.1.6.

Network Interface (API)	ISUP	H.323	SIP
routeReq (setupConnection)	ISUP IAM ISUP ACM ISUP ACK	H.225 SETUP (with Bearer) H.245 Capability Exchange H.245 Logical Channel Setup H.225 CALL PROC H.225 CONN	SIP INVITE SIP 100 TRYING SIP 183 SESSION PROGRESSING
releaseCall	ISUP REL ISUP RLC	H.245 End Session Command H.245 End Session Command H.225 Release Complete	SIP BYE SIP 200 OK
callAborted	ISUP REL ISUP RLC	H.245 End Session Command H.245 End Session Command H.225 Release Complete	SIP BYE SIP 200 OK
createCall createUICall releaseCallLeg createNotification changeNotification destroyNotification continueProcessing sendInfoAndCollectReq	No Mapping	No Mapping	No Mapping

Table 4-21: Network Interface API mapping to Protocols, adapted from (Sun JCC2SIP 2001)

ISUP	H.323	SIP	OSA/Parlay Fed
ISUP IAM (ISUP ACM)	H.225 SETUP (with Bearer) H.245 Capability Exchange H.245 Logical Channel Setup H.225 CALL PROC	SIP INVITE (SIP 100 TRYING)	createCall
ISUP ANM	H.225 CONN	SIP 200 OK	(*Media Bearer Connected*)
ISUP REL	H.245 End Session Command (H.245 End Session Command)	SIP BYE	release

Table 4-22: ISUP, H.323 and SIP Protocol mapping to OSA/Parlay Media Gateway and Fed Object, adapted from (Sun JCC2SIP 2001)

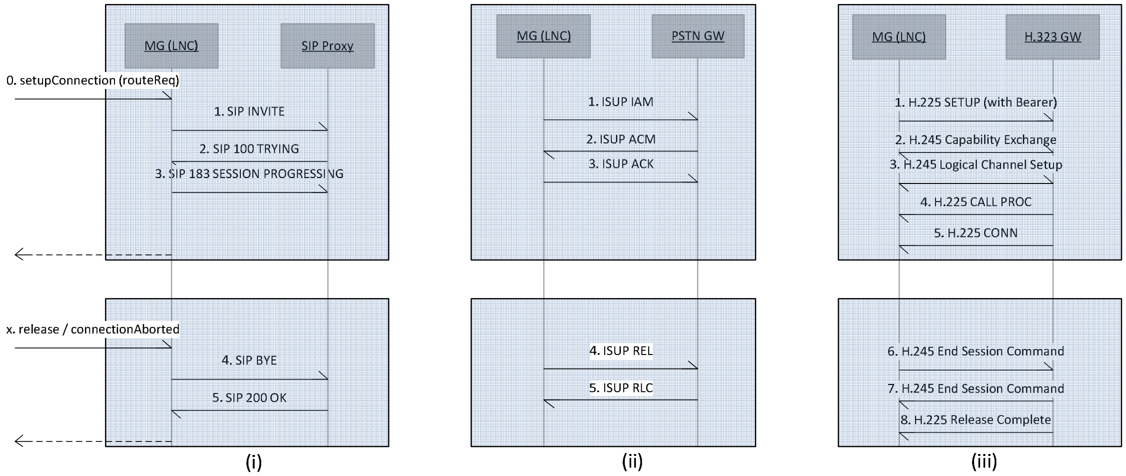

Figure 4-31: OSA/Parlay initiated Federation MSC to: (i) SIP; (ii) PSTN ISUP, (iii) H.323

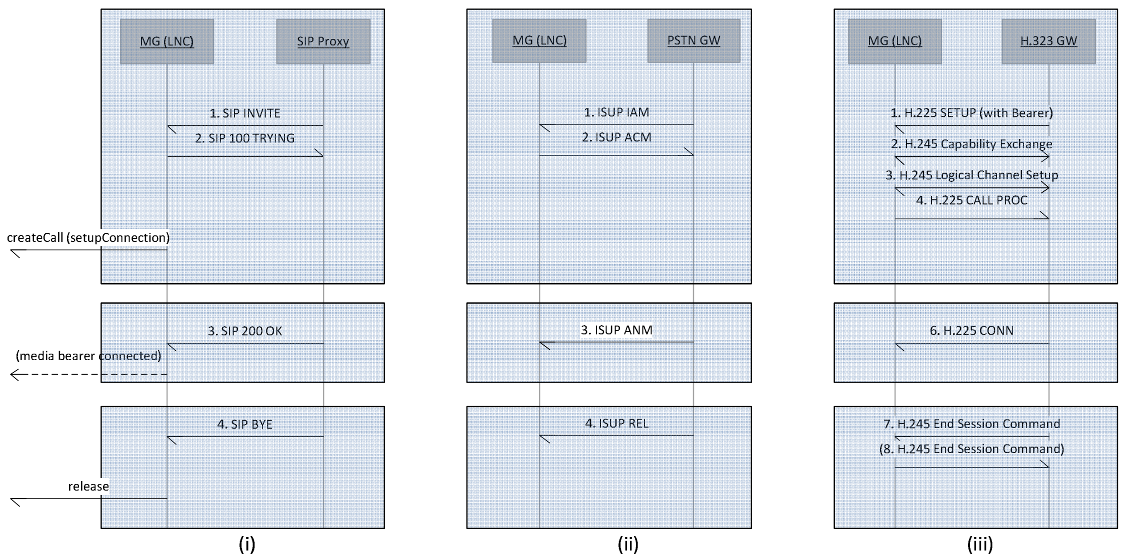

Figure 4-32: (i) SIP initiated federation to OSA/Parlay; (ii) PSTN ISUP initiated federation to OSA/Parlay; (iii) H.323 initiated federation to OSA/Parlay

4.10 Chapter Summary

The design of the OSA/Parlay Network Interface and Architecture has been presented. The OSA/Parlay Network Interface is characterised as an open, standard and technologically independent interface. The OSA/Parlay Network Architecture should be future proof from rapid technology developments and facilitate ease of simple integration of a variety of transport technologies. The OSA/Parlay network architecture is not a detailed definition; it specifies concepts and principles for connections involving the packet and switched circuit technologies as well as incorporate special resources such as media gateways and IVRs (Moodley and Hanrahan 2008). It conforms to the characteristics of being a simple layered architecture specified in a technology agnostic manner. The architecture should support an integrated approach to intelligent routing, QoS and call admission control for end-to-end connections. The architecture must explicitly support both federation of telcos and interoperability with legacy technologies. Useful concepts and principles from existing architectural proposals have been synthesised in the OSA/Parlay Network Architecture and Network Interface.

An information organising principle relies on the use of sessions. A session is a temporary allocation of resources that are configured to service a task for a period of time (TINA NRA 1997). The detailed architecture describes key computational objects and their purpose. The Communication Session Manager (CSM) is responsible for the communication between the transport network and the Parlay Gateway. The CSM contains the service envisioned view of the call within the Logical Connection Graph (LCG) for each instantiated call. The Connection Coordinator (CC) is responsible for the translating of the LCG into physical connections. A core function is to intelligently route the connection based on the terminal locations and using the dynamic network state and performance to determine the best route between the call parties. It also provides the function of admission control which determines whether the requested call properties (QoS) can be supplied by the transport network and on that basis, does a call admission control function. In so, the CC determines the layers are to be used for the connection and selects the appropriate Layer Network Coordinators (LNCs) to setup the connection. The CC is also responsible for managing the Administrative Federation on a call by call basis. The Layer Network Coordinator (LNC) is responsible for the connection within the layer transport network. In addition, the LNC provides network state information on QoS performance of links and nodes in the network.

The design details the open and technology independent API based interface. The OSA/Parlay Network Interface is specified using the API mechanism together with supporting methods. Legacy protocol adaptation is presented based on the JAIN approach, this is incorporated into the media gateway function to interoperate with legacy technologies. This proposal of the OSA/Parlay Network Architecture, Network Interface and Call Admission Control function represents a unique contribution of this research.

Chapter 5

Demonstration of the OSA/Parlay Network Architecture and Interface

The OSA/Parlay Network Architecture must perform the functions of call setup, maintenance and tear down of connection requests in the transport network. In order to demonstrate the OSA/Parlay Network Architecture and Network Interface, two services are exhibited. The first service is a simple 3rd party initiated 2-party call within the same administrative domain while the second service is a calling card service. Each use case is designed to illustrate key characteristics of the OSA/Parlay Network Architecture and Network Interface. These characteristics include *openness, simplicity, future-proof, QoS mechanisms* and supporting *federation* between telcos.

The demonstration is done over a Distributed Processing Environment (DPE) using CORBA. It provides a level of abstraction such that an application developer does not need to cater for the details of the operating system and network protocols by using an open standard interface. Key features of CORBA is that it is technology agnostic and objects have location transparency. The interface and architecture is implemented in the Java programming language using the NetBeans integrated development environment (IDE). The network architecture is not a detailed definition; rather it provides the concepts and principles for connections involving the constituent networks (packet and circuit switched) and single elements (media gateways and IVRs) (Moodley and Hanrahan 2008). These concepts and principles are demonstrated in a simulated environment and illustrate the NGN architectural characteristics. The operation of the CAC function is demonstrated in the ns-2 simulation environment.

The remainder of the chapter is organised as follows. Section 5.1 presents the services used to demonstrate the key concepts and principles of the OSA/Parlay Network Architecture and Network

Interface. Section 5.2 presents the Java DPE based technology and the implementation details for the demonstration. Section 5.3 describes the testing environment based on the Java programming language using the Netbeans integrated development environment (IDE). Section 5.4 details the results of the demonstration in the form of message sequence charts. The message sequence charts are analysed against the OSA/Parlay Network Architectural characteristic in section 5.5. The operation of the CAC function is demonstrated in the ns-2 simulation environment and presented in section 5.6.

5.1 Description of Services

The primary objective of the OSA/Parlay Network Architecture is to facilitate the call and service control requests from the OSA/Parlay Gateway and setup physical end-to-end connections within the core transport network. The OSA/Parlay Architecture must ensure that new calls entering the transport network are guaranteed the requested level of QoS for the entire duration of the call; while the QoS levels of existing calls within the network are not infringed upon notwithstanding the dynamic nature of calls being admitted and terminated within the transport network (Moodley and Hanrahan 2006). The OSA/Parlay Network Architecture specifies concepts and principles for connections involving the packet and switch circuit technologies as well as incorporate special resources such as media gateways and IVRs. The following use cases are considered to illustrate the characteristics described above. These use cases had been designed, coded and tested as part of the experimental work that was conducted. The source code for the proof of concept implementation is provided on a separate removable storage disc.

- **Simple 3rd party initiated 2-Party Call:** The OSA/Parlay Gateway initiates a 2-party call within the OSA/Parlay Network Architecture using the API based OSA/Parlay Network Interface. The two calling parties are in the same administrative domain. The calling performance requirements are submitted and the positive case of the network admitting the request is presented. The typical call setup procedure is described. The details of the message sequence chart will be used to analyse some of the characteristics of the architecture (Yun and Perros 2010; Moodley and Hanrahan 2008).

- **Calling Card Service:** The calling card service is a much more complex service and is popular for international calling. A typical calling card service operates as follows: The user purchases a calling card which has a unique pin and preloaded credit. The user (calling party) makes a call to the service. The IVR requests the Pin number and verifies the credit available. The IVR then request the destination party's number. The service routes the call to the destination in a federated telco network. The details of the message sequence chart will be used to analyse some of the characteristics of the architecture (Yun and Perros 2010; Moodley and Hanrahan 2008).

5.2 Distributed Processing Environment

The RM-ODP Engineering viewpoint describes the infrastructure required to support a distribution processing environment. This demonstration is implemented using the Java Orb which is an implementation of Common Object Request Broker Architecture (CORBA). CORBA (RM-ODP 1995) allows objects to communicate with each other although these objects reside in a heterogeneous distributed computing environment. It provides a level of abstraction such that an application developer does not need to cater for the details of the operating system and network protocols by using an open standard interface. This interface is commonly referred to as the Interface Description Language (IDL) in CORBA and is language and operating system independent. Applications become portable due to this independent feature of the IDL. The CORBA standard has also defined the Internet Inter-ORB Protocol (IIOP) messages, which allow different implementations of CORBA to interoperate. The detailed architecture of CORBA is depicted Figure 5-1 below. The bold black lines depict the physical operation of the Object Request Broker (ORB) while the dotted blue line represents the logical communication. The OSA/Parlay Network Interface as well as interfaces between the computational objects are implemented using the IDL description.

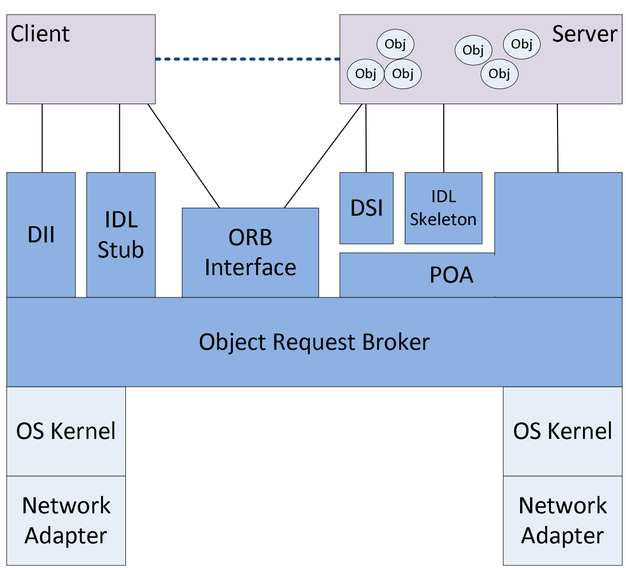

Figure 5-1: The CORBA Architecture, adapted from (RM-ODP 1995)

5.3 Demonstration Environment

The RM-ODP Technology viewpoint specifies the implementation of the design using technology specifics to support the OSA/Parlay Network Interface and Architecture. The demonstration environment is based on the Java programming language using the Netbeans integrated development environment (IDE). The Java environment incorporates the Java implementation of CORBA. The OSA/Parlay Network Architectural Computational Objects detailed in section 4.5 are implemented in Java programming language whilst the interfaces between the computational object are defined in the IDL description. Similarly, the OSA/Parlay Network is defined in the IDL description.

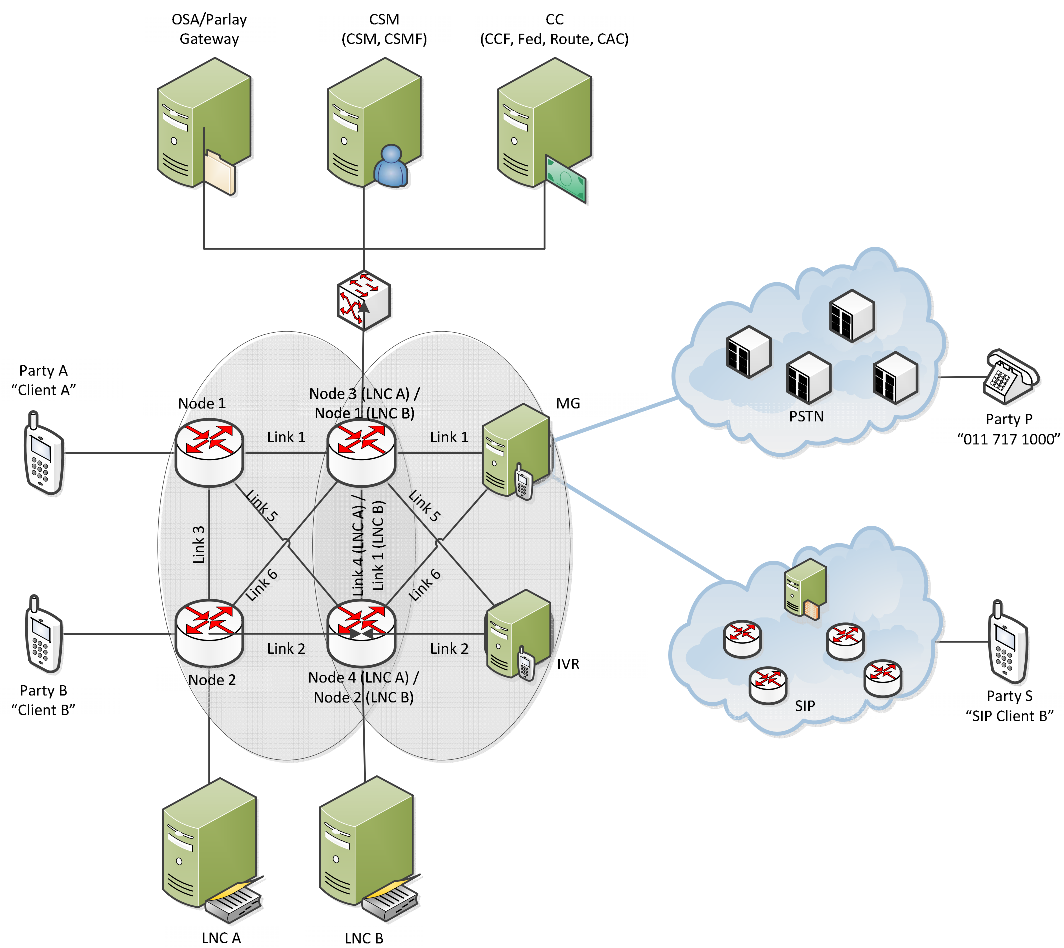

Figure 5-2: The OSA/Parlay Demonstration Environment

The OSA/Parlay Network Architecture is implemented over a converged network infrastructure, see Figure 5-2. The computational objects are distributed over the Distributed Processing Environment. In order to demonstrate the principles and concepts, specific elements emulate the user equipment in the consumer domain, the OSA/Parlay Gateway, the transport network elements, the media gateway and the IVR.

5.4 Results

The results of the demonstration of both the 3rd party initiated 2-party call service and the calling card service are presented in the form of message sequence charts. The analysis of which are discussed in section 5.6.

5.4.1 3rd Party initiated 2-Party Call Service

A simple two party call within the OSA/Parlay Network Architecture is described using the message sequence chart depicted in Figure 5-3. The OSA/Parlay Gateway initiates the 3rd party call attempt by informing the OSA/Parlay Network Architecture of an imminent call. It uses the createCall () method on the Network API based interface. This method is invoked on the CSMF factory object which is permanent and the initial point of contact. The CSMF creates to new CSM Object dedicated to the imminent call and unique identifier callSessionID is allocated to the call. The unique identifier to the newly created CSM object is returned to the gateway depicted in messages 1 – 2.

The OSA/Parlay Gateway thereafter requests that the connection be admitted into the network. The details of the connection such as source and destination address, connection requirements based on QoS parameters are contained in the routeReq () method and invoked on the CSM object using the unique object reference and unique identifier as shown in message 3.

Messages 4 – 6 illustrate the following operation; the CSM requests the Connectivity Session to setup the connection with the connection parameters. As the CC object does not exist, the method is invoked on the CCF object. The CCF creates a new CC Object dedicated to the connection using the same unique identifier as the CSM. The reference to the new CC Object together with the unique identifier is returned to the CSM Object. This ensures that the CSM has the details to contact the CC Object to invoke further operations. The CCF informs the CC to setup the connection in the network.

The CC requests the Fed Object to determine with the call parties are in a federated network, the response is that the parties are in the local network. The CC thereafter requests the Route Object to determine the best path from the calling party to the called party using the adapted Dijkstra's Algorithm. The CC invokes the routeSelect () method on the Route Object. The proposed path is determined by both the connection request as well as the dynamic state of the network in terms of performance of links and topology described in sections 4.6.2 and 4.6.3. The best route together with its performance details are returned using the routeList () method as indicated by messages 9 – 10.

The decision whether to admit or reject the call request is made by the CAC object, messages 11 – 12. The request is forwarded to the CAC object together with the proposed connection path and the performance details using the callAdmissionReq () method. The CAC determines whether the network can meet or exceed the bandwidth and performance request within the network. We deal with the positive response. The CAC confirms the admission decision to the CC by sending the callAdmitted () method.

The connection path (Node 1 – Node 3 – Node 2) is selected based on the dynamic state of the network at the specific time. The CC request that the connection path be setup in the LNC network using the setupConnection () method to each node in the path, indicated by messages 13 – 16.

The confirmations indicate that the connection path has been setup and the OSA/Parlay Gateway is informed using the routeRes () method. At this point the call session is in place within the core transport network.

The analysis of the simple 3rd party initiated call indicates that a minimal number of 17 messages are required for a simple 2-party call. This is at least an order of magnitude more efficient than the existing architectural options: TINA – 30 messages; MSF – 60 messages; 3GPP LTE – 49 messages (see Chapter 3). The simple architecture allows for decisions to be made at the lowest layer possible and the proliferation of messages between objects are minimised as compared to existing architectures. The interfaces to the internal objects are based on the API concept ensuring technology agnostic and future proofing characteristics. The architecture incorporates intrinsic QoS mechanism which provides application related functions. The routing decision is based on the dynamic state of the network and the CAC makes the admission or rejection decision based on the QoS performance requested from the connection.

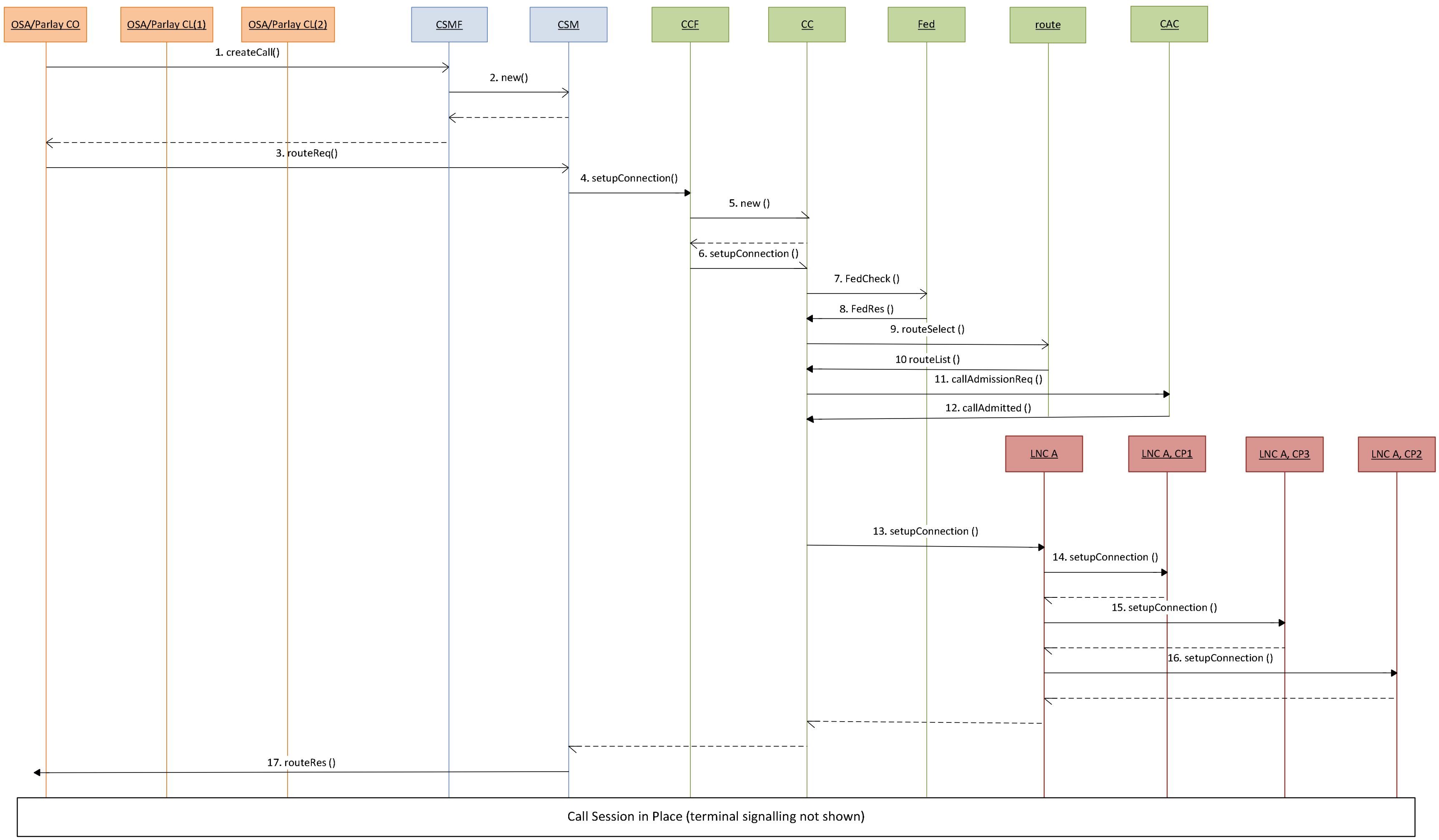

Figure 5-3: 3rd Party initiated 2-Party Call Service

5.4.2 Calling Card Service

The Calling Card service within the OSA/Parlay Network Architecture is described using the message sequence chart depicted in Figure 5-4 to Figure 5-8. The OSA/Parlay Gateway initiates a calling card service on request from the user by informing the OSA/Parlay Network Architecture of an imminent IVR call using the createUICall () method. This method is invoked on the CSMF factory object which is permanent and the initial point of contact. The CSMF creates a new CSM Object dedicated to the imminent call and a unique identifier callSessionID is allocated to the call. The unique identifier to the newly created CSM object is returned. Since the call is to the IVR unit, the CSMF replaces the destination party address to the IVR address within the network before forwarding the routeReq () method to the CSM, indicated in messages 1 – 3.

Messages 4 – 6 illustrate the following operation; the CSM requests the Connectivity Session to setup the connection with the connection parameters. The CCF creates a new CC Object dedicated to the connection using the same unique identifier as the CSM. The new CC Object's unique identifier is returned to the CSM Object. The CCF informs the CC to setup the connection in the network.

The CC requests the Route Object to determine the best path from the calling party to the called party using the adapted Dijkstra's Algorithm. The CC invokes the routeSelect () method on the Route Object. The proposed path is determined by both the connection request as well as the dynamic state of the network in terms of performance of links and topology described in sections 4.6.2 and 4.6.3. The proposed route together with the route performance details are returned using the routeList () method as indicated by messages 7 – 8.

The decision whether to admit or reject the call request is made by the CAC object, messages 9 – 10. The request is forwarded to the CAC object together with the proposed connection path and the performance details using the callAdmissionReq () method. The CAC determines whether the network can meet or exceed the bandwidth and performance request within the network. We deal with the positive response. The CAC confirms the admission decision to the CC by sending the callAdmitted () method.

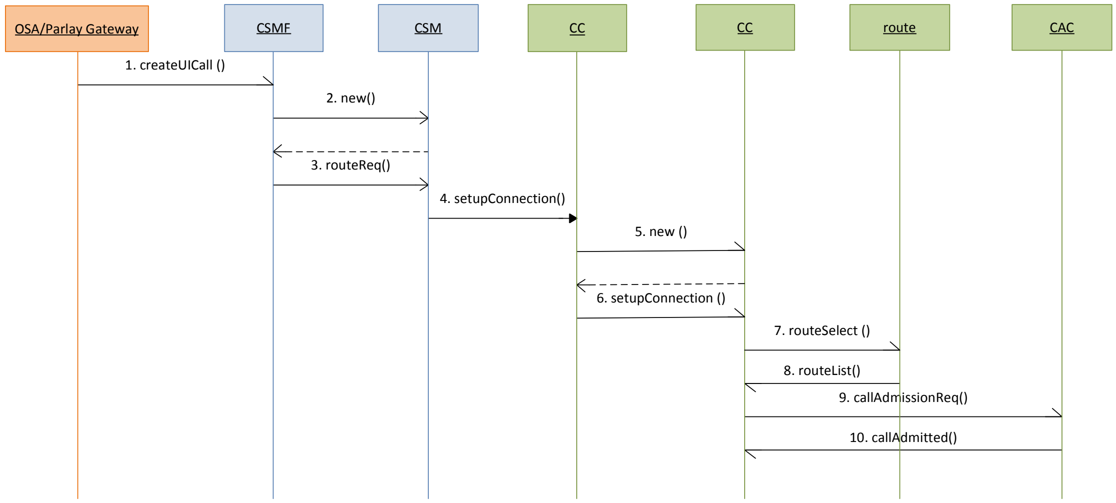

Figure 5-4: Calling Card Service – call admission to the IVR unit

The connection path (Node A.1 – Node A.3 – Node B.1 to IVR) is selected based on the dynamic state of the network at the specific time. The CC request that the connection path be setup in the LNC network using the setupConnection () method to each node in the path, indicated by messages 11 – 16.

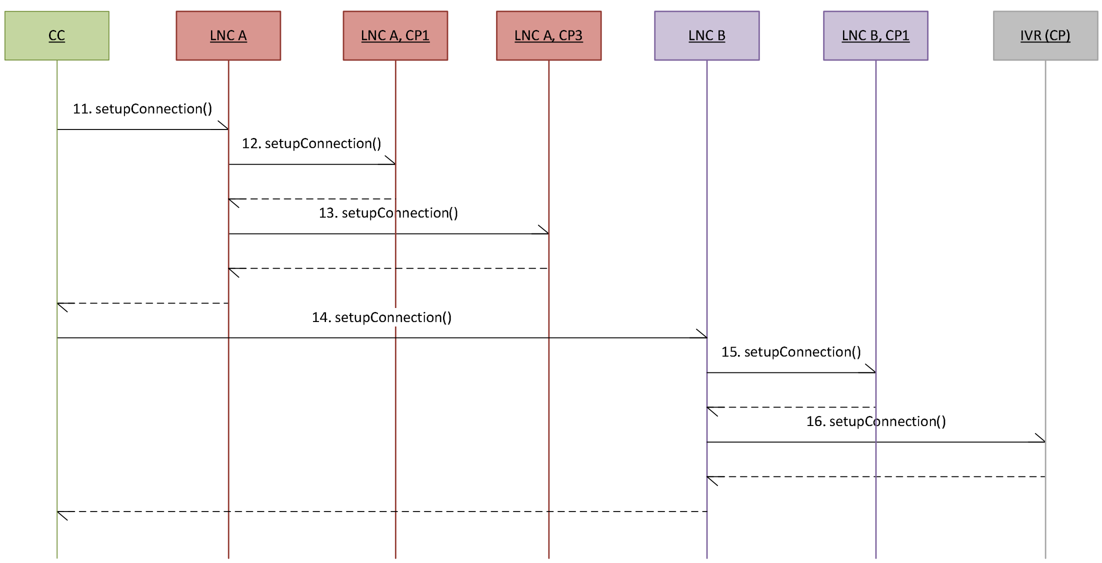

Figure 5-5: Calling Card Service – setup of the connection between the user and IVR unit

The calling card service first requests the pin and then announces to the user the credit balance available thereafter the IVR requests that the user inserts the destination party's address details. This information is collected and forwarded to the service in messages 17 – 24. The calling card service requested that the terminating state be armed using the createNotification () method, which is used to determine when the IVR is released from the call in messages 25 – 26. At that point the call will be rerouted to the destination address.

The IVR and network resources are released from the call using the releaseUI () and release () method, depicted in messages 27 – 36. The armed event is triggered and the service is informed via the CSM using the reportNotification () method in messages 37 – 38.

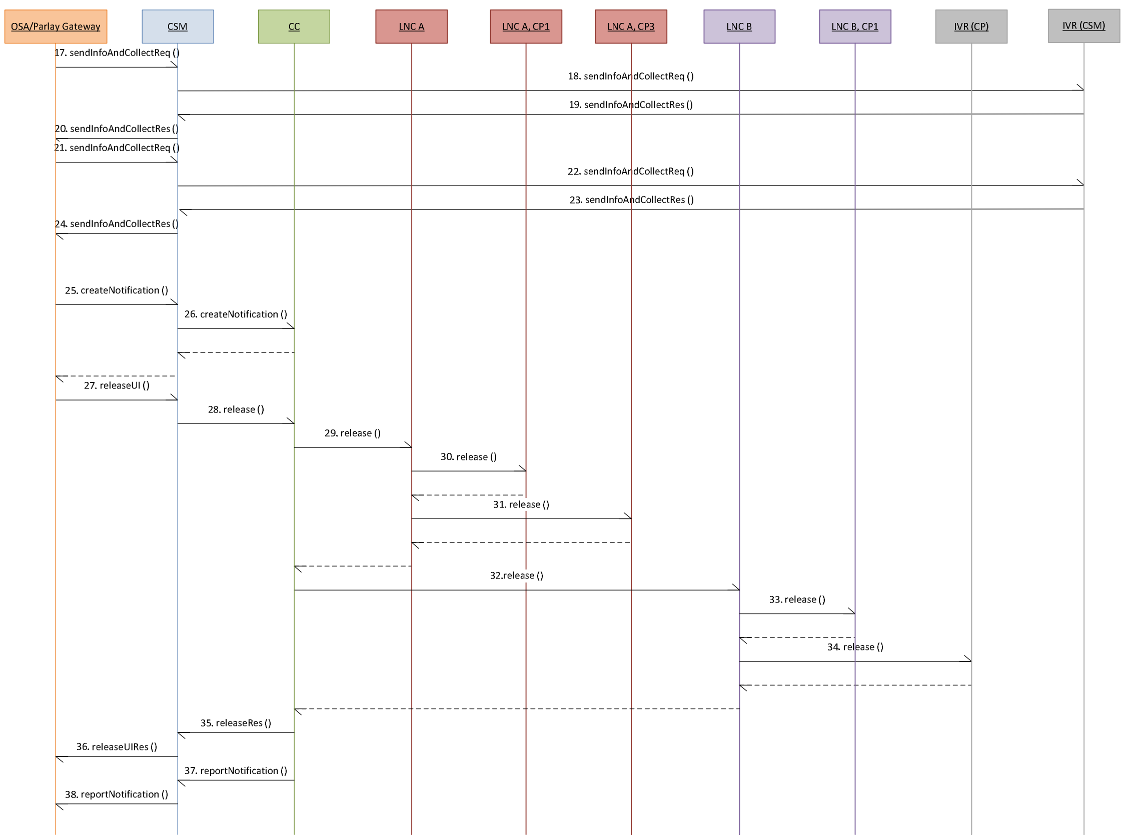

Figure 5-6: Calling Card Service – playing the announcements and collecting the calling destination details.

On receiving the reportNotification of the armed event at the gateway (messages 37 – 38), the calling card service redirects the call to the destination party. The routeReq () provides the details of the destination party together with performance parameters. In this case the CC does not recognise the destination address and requests the Fed Object to determine whether the destination party is in a federated network. The result is returned to the CC in messages 41 – 42.

The CC requests the Route Object to determine the best path from the calling party to the destination party using the adapted Dijkstra's Algorithm. The identified path and its performance details are returned using the routeList () method. The decision whether to admit or reject the call request is made by the CAC object, messages 45 – 46. The request is forwarded to the CAC object together with the proposed connection path and the performance details of the proposed path using the callAdmissionReq () method. The CAC determines whether the network can meet or exceed the bandwidth and performance request within the network. We deal with the positive response. The CAC confirms the admission decision to the CC by sending the callAdmitted () method.

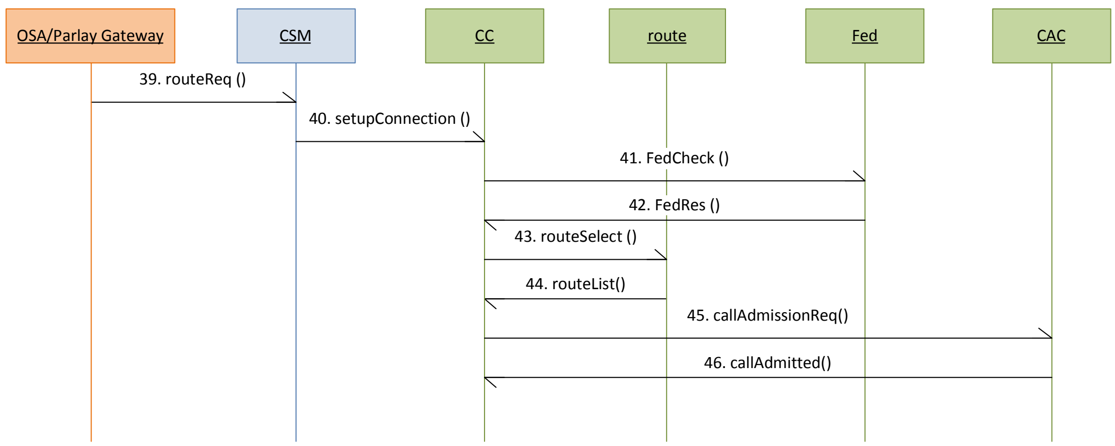

Figure 5-7: Calling Card Service – setup of the connection between the user and destination party

The connection path (Node A.1 – Node A.3 – Node B.1 to MG) is selected based on the dynamic state of the network at the specific time. The CC request that the connection path be setup in the LNC network using the setupConnection () method to each node in the path depicted in messages 47 – 53. We deal with two possible federated networks being PSTN based or SIP based. The messages indicate the translation from API methods to protocol specific messages being either ISUP or SIP respectively in messages 54 – 57.

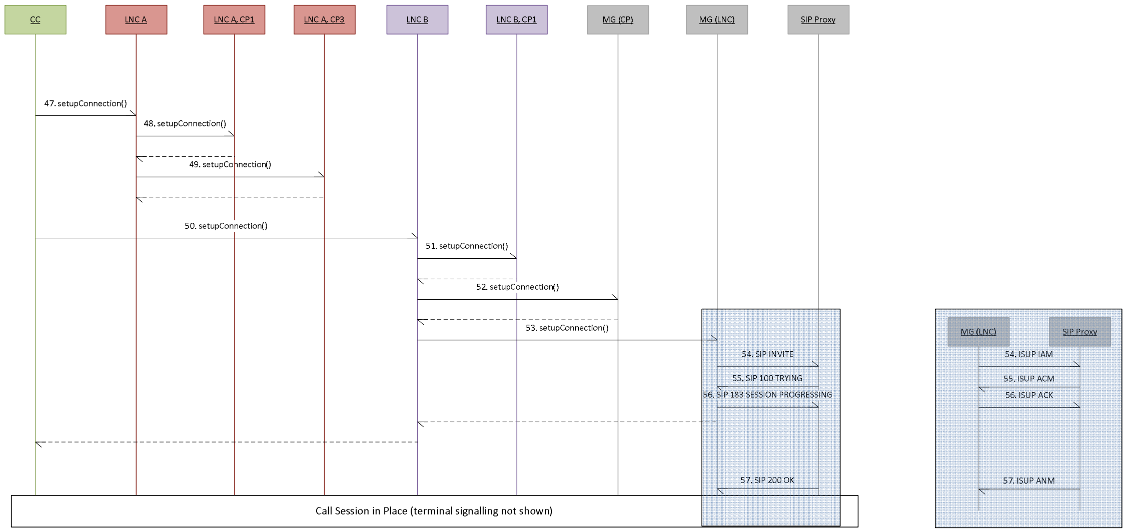

Figure 5-8: Calling Card Service – (i) setup of the connection between the user and destination party in a federated SIP network; and (ii) a destination party in the PSTN federated network

The analysis of the complex calling card service indicates the efficiency of the architecture by the number of messages. The interfaces to the internal objects are based on the API concept ensuring technology agnostic and future proofing characteristics. Similarly the interfaces to the special resources such as the IVR and Media Gateway are based on the API concept ensuring the interface is technology agnostic and future proof. The interaction with the IVR special resource indicates the integration of the special resource within the architecture and messages are not proliferated between objects. The interaction with the Media Gateway special resource provides insight into the seamless manner in which the OSA/Parlay architecture can federate to both the packet based SIP network or the legacy PSTN network. The QoS for the connection between the OSA/Parlay network and the SIP network is only guaranteed within the OSA/Parlay network part of the connection. The QoS for the connection between the OSA/Parlay network and the PSTN will be guaranteed as the PSTN intrinsically provides QoS guarantees.

5.5 Demonstration of the CAC

The CAC functionality was demonstrated separately in the ns-2 (UCB et al 2012) network simulation environment. ns-2 features simulating discrete events at packet level. Thus, state changes such as traffic being introduced or terminated within the network are simulated as discrete events. The original intention for the simulator targeted research into packetised networks. The simulator supports a wide variety of technologies (Huang 2012) such as:- point-to-point links, LANs, unicast and multicast networks, transport protocols such as TCP and UDP, application level protocols (FTP, etc.), mobile networks, etc. Two particular advantages of ns-2 are the detailed trace logs, which have been utilised to derive network state information; and the visualization tool, which is useful in providing information on topology, packet animation and inspection tools." (Moodley and Hanrahan 2007)

The three important components of guaranteed QoS are demonstrated. These are: (i) the intelligent routing of connections in the PSN; (ii) dynamic information of the network state; and (iii) making the call admission control decision on whether the network can meet or exceed QoS requested. Performance metrics of latency, jitter and packet loss are used. These metrics measure the time taken for a packet to traverse the network, the variance of the delay and the loss of packet due to routing constraints. (Moodley and Hanrahan 2007).

The following network topology was setup using MPLS routers. "MPLS makes extensive use of the Label Distribution Protocol (LDP), which is used to establish Label Switched Paths (LSPs). The LDP protocol ensures that LSPs are setup for specified FECs from source to destination, it also ensures that labels are correctly associated in the routing tables of the MPLS Label Switched Routers (LSRs). Explicit Routing allows the MPLS network to setup a permanent LSP in the MPLS network from source to sink with the required class based QoS parameters." (Moodley and Hanrahan 2007). The proposal of the dynamic Dijkstra Algorithm ensures that the best path in the MPLS network is selected and this route is appropriately reserved in the MPLS network utilising the constraint routed LDP protocol for bandwidth selection and path reservation.

5.5.1 Application of Dynamic Dijkstra Algorithm for Call Admission

The adaptation of the Dijkstra's algorithm was coded for the CAC functionality within the simulator. The dynamic latencies of links need to be measured at runtime and used in making call admission and routing decisions during simulation. Latency is the time difference between packet received and packet sent. This is accomplished by modifications to the ns-2 simulation code. An approach by Ke (Ke 2011) used to calculate the latency post simulation has been adapted to provide latency measures during simulation. The packet sent time is inserted into each packet, this information together with the packet sequence numbers are used to correlate packets at the receiving node to determine latency and jitter metrics. The ~/common/packet.h code in ns-2 is modified by lines depicted in Figure 5-10 and the ~/apps/udp.cc code is modified by lines depicted in Figure 5-10. Thus a new field, sendtime_ is added and when a udp packet is created, the local time of the simulator is recorded into the sendtime_ field.

```
struct hdr_cmn {
    enum dir_t { DOWN= -1, NONE= 0, UP= 1 };
    packet_t ptype_;    // packet type (see above)
    int   size_;        // simulated packet size
    int   uid_;      // unique id
    int   error_;       // error flag
    int   errbitcnt_;   // # of corrupted bits jahn
    int   fecsize_;
    double    ts_;      // timestamp: for q-delay measurement
    int   iface_;       // receiving interface (label)
    dir_t direction_;    // direction: 0=none, 1=up, -1=down
    double  sendtime_; // // new field to capture the send time of the pkt
    ...
    ...
    ...
    inline int& addr_type() { return (addr_type_); }
    inline int& num_forwards() { return (num_forwards_); }
    inline int& opt_num_forwards() { return (opt_num_forwards_); }
    //monarch_end
    inline double& sendtime() { return (sendtime_); }  // new field added

  }
```

Figure 5-9: Modified code for ~/common/packet.h, adapted from (Ke 2011)

```
void UdpAgent::sendmsg(int nbytes, AppData* data, const char* flags)
 {
  ...
  double local_time = Scheduler::instance().clock();
        while (n-- > 0) {
                p = allocpkt();
                hdr_cmn::access(p)->size() = size_;
                hdr_rtp* rh = hdr_rtp::access(p);
                rh->flags() = 0;
                rh->seqno() = ++seqno_;
                hdr_cmn::access(p)->timestamp() =
                   (u_int32_t)(SAMPLERATE*local_time);
                hdr_cmn::access(p)->sendtime_ = local_time;    // The send time added to this field
                // add "beginning of talkspurt" labels (tcl/ex/test-rcvr.tcl)
                if (flags && (0 ==strcmp(flags, "NEW_BURST")))
                        rh->flags() |= RTP_M;
                p->setdata(data);
                target_->recv(p);
        }
        n = nbytes % size_;
        if (n > 0) {
                p = allocpkt();
                hdr_cmn::access(p)->size() = n;
                hdr_rtp* rh = hdr_rtp::access(p);
                rh->flags() = 0;
                rh->seqno() = ++seqno_;
                hdr_cmn::access(p)->timestamp() =
                   (u_int32_t)(SAMPLERATE*local_time);
                hdr_cmn::access(p)->sendtime_ = local_time;    // The send time added to this field
        // add "beginning of talkspurt" labels (tcl/ex/test-rcvr.tcl)
                if (flags && (0 == strcmp(flags, "NEW_BURST")))
                        rh->flags() |= RTP_M;
                p->setdata(data);
                target_->recv(p);
        }
        idle();
}
```

Figure 5-10: Modified code for ~/apps/udp.cc, adapted from (Ke 2011)

The existing LossMonitor agent is adapted to report on packet latencies in addition to the existing packet loss function, unlike the approach taken by Ke (Ke 2011) to create a new sink agent. Based on the work by Ke (Ke 2011), the modified code for ~/tools/LossMonitor.h and ~/tools/LossMonitor.cc are illustrated in Figure 5-11 and Figure 5-12 respectively.

The modified LossMonitor Agent is able to determine instantaneous latency on the link (*current_owd*). In addition there is functionality to determine the instantaneous jitter performance of the link (*current_ipdv*) and the cumulative totals for both latency and jitter used to determine average latency and jitter respectively. On identifying the packet loss condition, which indicates that either the bandwidth constraints have been reached or there is link failure, the instantaneous latency is set to a large value. This large value ensures that this link is not considered as viable in the Dijkstra algorithm. The instantaneous latency per packet (*current_owd*) is used as the latency measure in the simulation studies. The *current_owd* field is in applied in the c++ code, this field is referred to as *owd_* in tcl code, using *bind("owd_", &current_owd_); // binding to tcl* in Figure 5-12

```
class LossMonitor : public Agent {
public:
        LossMonitor();
        virtual int command(int argc, const char*const* argv);
        virtual void recv(Packet* pkt, Handler*);
protected:
        int nlost_;
        int npkts_;
        int expected_;
        int bytes_;
        int seqno_;
        double last_packet_time_;

        // new features for loss monitor to calculate runtime delay, jitter

        double now;
        double sum_ipdv_;          // jitter sum can be used to calculate average jitter
        double sum_owd_;           // one way delay the sum can be used to calculate average owd
        double current_owd_;       // current measured in one way delay
        double current_ipdv_;      // the current measured jitter
        double min_owd_; // minimum one way delay value
        double previous_owd_;      // the last measure to one way delay

};

#endif // ns_loss_monitor_h
```

Figure 5-11: Modified code for ~/tools/LossMonitor.h, adapted from (Ke 2011)

```
LossMonitor::LossMonitor() : Agent(PT_NTYPE)
{
	bytes_ = 0;
	nlost_ = 0;
	npkts_ = 0;
	expected_ = -1;
	last_packet_time_ = 0.;
	seqno_ = 0;

	// new lines for delay and jitter
	sum_ipdv_ = 0;
	sum_owd_ = 0;
	current_owd_ = 0;
	current_ipdv_ = 0;
	previous_owd_ = 0;
	min_owd_ = 100000000;

	bind("nlost_", &nlost_);
	bind("npkts_", &npkts_);
	bind("bytes_", &bytes_);
	bind("lastPktTime_", &last_packet_time_);
	bind("owd_", &current_owd_);  // binding to tcl
	bind("expected_", &expected_);
}

void LossMonitor::recv(Packet* pkt, Handler*)
```

Figure 5-12: Modified code for ~/tools/LossMonitor.cc, adapted from (Ke 2011)

In order to apply the dynamic latency measure, a source and LossMonitor sink agents are setup over each link in the topology. The traffic agent inserts the appropriate probe traffic into the network and the dynamic latencies are recorded periodically as illustrated in Figure 5-13 below.

Figure 5-13: Tcl code for periodic recording of the dynamic latencies for each link

During simulation, when a new connection is setup from source node toward a destination node, the Dijkstra algorithm is invoked as described in Figure 5-14 below. The algorithm uses the dynamic measured latencies as input into the algorithm and produces the shortest path in the topology at that particular instant in time for the requesting connection. The tcl code for the algorithm has been adapted from (Rosettacode 2011) which is a static implementation and modified to become dynamic. For the purpose of implementation, the topology applied is undirected assuming the duplex links are symmetric and the destination node will always be an edge node in the MPLS topology.

The result of the dynamic Dijkstra algorithm is that the best explicit route is selected for the awaiting call request based on the dynamic state of the network. The path performance in terms of latency is tested to ensure that the path performance exceeds the requested performance criteria. The explicit route is used as an input to the constrained routed LDP protocol to ensure that the path selected has sufficient available bandwidth to service the request. Thereafter the CR-LDP protocol reserves the selected path in the transport MPLS network.

```
proc dijkstra {graph origin} {
  # Initialize
  dict for {vertex distmap} $graph {
          dict set dist $vertex Inf
          dict set path $vertex {}
  }
  dict set dist $origin 0
  dict set path $origin [list $origin]

  while {[dict size $graph]} {
          # Find unhandled node with least weight
          set d Inf
          dict for {uu -} $graph {
             if {$d > [set dd [dict get $dist $uu]]} {
                    set u $uu
                    set d $dd
             }
          }

          # No such node; graph must be disconnected
          if {$d == Inf} break

          # Update the weights for nodes lead to by the node we've picked
          dict for {v dd} [dict get $graph $u] {
             if {[dict exists $graph $v]} {
                    set alt [expr {$d + int($dd)}]
                    if {$alt < [dict get $dist $v]} {
                       dict set dist $v $alt
                       dict set path $v [list {*}[dict get $path $u] $v]
                    }
             }
          }

          # Remove chosen node from graph still to be handled
          dict unset graph $u
  }
  return [list $dist $path]
}

proc makeUndirectedGraph arcs {
  # See figure 5.2 for standard references for links
  # links in reverse to ensure undirected graph
  global l1 l2 l3 l4 l5 l6
  dict set graph 1 3 $l1
  dict set graph 2 4 $l2
  dict set graph 1 2 $l3
  dict set graph 3 4 $l4
  dict set graph 1 4 $l5
  dict set graph 2 3 $l6
  dict set graph 3 1 $l1
  dict set graph 4 2 $l2
  dict set graph 2 1 $l3
  dict set graph 4 3 $l4
  dict set graph 4 1 $l5
  dict set graph 3 2 $l6
  return $graph
}

set arcs {
}


lassign [dijkstra [makeUndirectedGraph $arcs] $sourcedijkstra] costs path
puts "path from source to destination costs [dict get $costs $destdijkstra]"
puts "route from source to destination is: [join [dict get $path $destdijkstra] {_}]"
```

Figure 5-14: Tcl code dynamic Dijkstra Algorithm, adapted from (Rosettacode 2011)

5.5.2 Experimental Environment

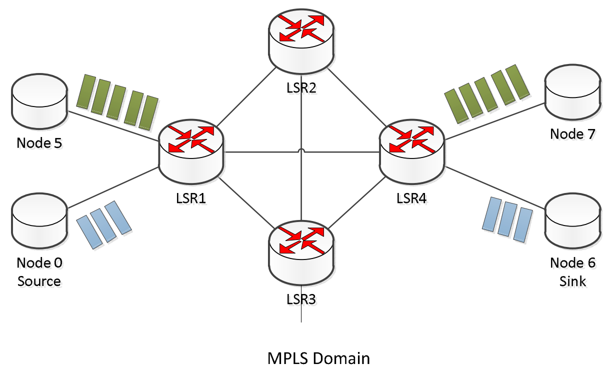

Figure 5-15: Topology of the ns-2 simulated network, adapted from (Moodley and Hanrahan 2007)

The simulated environment provides a means to simulate various dynamic traffic conditions. Further a number of traffic applications can be simulated including data, voice and video traffic. (Moodley and Hanrahan 2007). The generic topology illustrated in Figure 5-15 consists of 4 MPLS routers together with 2 source nodes and 2 sink nodes. This topology has been applied for the simulation studies described in section 5.5.3 and section 5.5.4. The following are key characteristics of the simulation setup:

- Node 0 hosts the source probe traffic which flows through the network towards sink on Node 6.
- Node 5 hosts the load traffic for the network. It hosts a number of sources of traffic. The traffic flows through the network towards the sinks on Node
- 1 Mb/s bandwidth on all links within the core MPLS network (LSR1 – LSR2 – LSR3 – LSR4).
- 2 Mb/s bandwidth in the Access links into the network (Nodes 0 & 5 to LSR1; LSR4 to Nodes 6 & 7).
- Class Based Queuing was implemented with a queue length of 12. All traffic was placed in the same class, since the simulation only investigates a single class of traffic.
- The test traffic was applied from source to sink and was allowed to reach a steady state condition.
- MPLS CR-LDP is applied to test the available path bandwidth against requirements and to make path reservations.

5.5.3 Probe traffic – Determining the state of a link

Probe traffic is used to determine whether a call awaiting connection into the transport network can be serviced for the entire call duration with the requested performance requirements while not adversely affecting the performance of existing calls within the network. The probe traffic presented is a video source with bit rate of 512kb/s. The data is bursty in nature as demonstrated in Figure 5-16(i). The exiting call traffic within the network link is illustrated between at t ≈ 0 seconds and t = 40 seconds. A step load change of the 512kb/s probe traffic is inserted into the network between t = 40s and t = 300s and the effects on the performance metrics are monitored and analysed.

The performance of the link is illustrated in Figure 5-16. The first graph depicts the measured data plotted against time. The second graph calculates the latency of the probe traffic with a moving average window applied. Similarly, the third and fourth graphs present the moving average of the jitter and packet loss respectively.

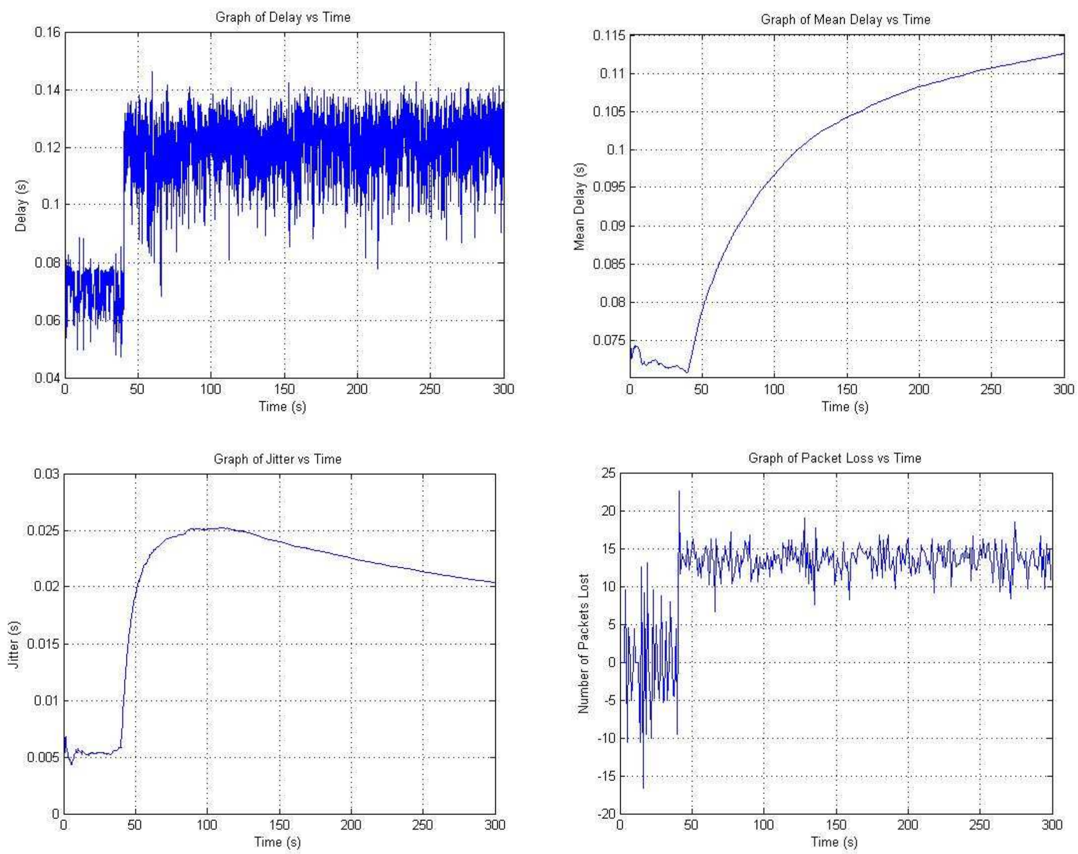

Figure 5-16: QoS Metric Performance traces from Probe Video Traffic

(i) Actual delay measured; (ii) latency; (iii) jitter; and (iv) packet loss

An indication of the link latency is observed as the mid-point of the burstiness in Figure 5-16(i). The jitter metric (Figure 5-16(iii)) is a function of the burstiness of the probe video traffic. The steady state latency prior to the admission of the probe traffic is approximately 73ms while after the probe traffic was applied to the link, the latency of the link increases to approximately 115ms. The observed latency is a function of the fixed propagation delay of the link as well as the queue delay at each MPLS LSR. The jitter performance of the link increases from approximately 5ms to a steady state of 20ms. After t = 60 seconds of applying the probe traffic, there is a significant increase in latency and jitter. This can be attributed to the queue buffers of the LSR being increasingly filled. Thereafter the queue buffer reaches a steady state and the latency and jitter performance begin to settle. Due to the burstiness of the probe video traffic, some packet loss is experienced (see Figure 5-16(iiii)). The QoS performance metrics provide insight into the dynamic performance of the state of the link at a particular point in time and under different loading conditions.

The probe traffic represents a video call, the maximum link performance that can be tolerated for a video application is a latency of 150ms and jitter of 30ms. The CAC function will only consider the latency metric. As the link latency performance is approximately 115ms, well below the 150ms threshold the latency test is passed. The CAC then applies the constraint routing protocol to confirm that sufficient bandwidth in the link is available and the video traffic is admitted into the transport network.

5.5.4 Illustration of the CAC function

The CAC must ensure that the call request is guaranteed a level of QoS for the entire duration while existing connections do not deteriorate in QoS beyond its contracted levels. The probe source is used to analyse the QoS performance effects with the CAC functionality implemented. The probe video source at 64kb/s is initially simulated in the network and is used to illustrate that the performance parameters are not violated during the admission of additional calls into the transport network. In order to study the effectiveness of the CAC, only one link (LSR1 – LSR4) was active in the topology while the other 5 links were set to fail during the simulation.

The CAC function admits connection requests from an idle link state until the link reaches its capacity at approximately t = 45 seconds, as illustrated in Figure 5-17(i). Calls are simulated to have a random duration with an average of 48 seconds before terminating. This call pattern simulates the busy call hour period for a telco. During this period calls are being rejected based on capacity and the network unable to meet the performance requirements. The call rejection rate is illustrated in Figure 5-17(i). After t = 495 seconds the existing calls begin to terminate.

The performance of the single video probe traffic source provides insight into the QoS performance of existing traffic already admitted into the network. The observed latency of the existing video call is illustrated in Figure 5-17(ii) which depicts a rapid increase in latency with the admission of calls into the transport network. During the critical period between t = 45 seconds and t = 495 seconds, the latency of the video source is bursty and reaches maximums of approximately 148ms. As the call load decreases, the latency observed decreased rapidly. The CAC function makes use of this latency information to determine whether to admit further calls into the network. This critical threshold of 150ms is used as a determining factor by the CAC to make the admission or reject decision as observed in Figure 5-17(ii). The CAC function ensures that that the observed latency does not exceed the contracted performance limit of 150ms for the video application. The jitter and packet loss is negligible and within the contracted performance parameters for below 30ms and 5% respectively.

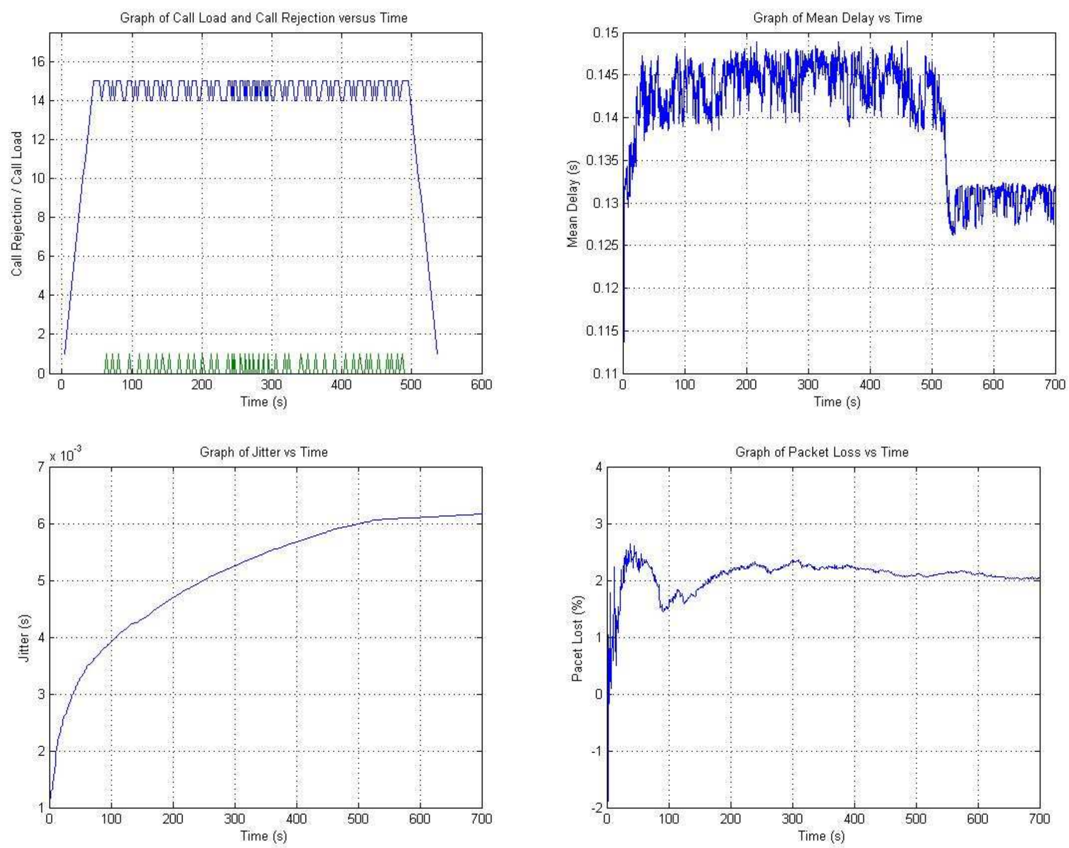

Figure 5-17: Illustration of the CAC function

(i) Call Load and Call Rejection rate; (ii) latency; (iii) jitter; and (iv) packet loss

The demonstration of the CAC function provides confidence that guaranteed QoS is achievable and that the CAC function allows the network links to be used optimally without compromising the QoS performance.

5.5.5 Simulation of the 3rd Party initiated 2-Party Call Service

The simulation of the 2 party call service consisted of voice traffic at 64kb/s. Voice traffic exhibit the characteristics of constant bit rate traffic. The calling rate was one call every 20 seconds. The latency of the link starts at 46.5ms in a lightly loaded condition until t = 80 seconds when a fourth traffic source is added into the link, as illustrated in Figure 5-18(ii). The latency represents the propagation delays of the path, while the queue delay has a negligible influence on the latency. After the fourth traffic source is added to the link at t = 80 seconds, the packets begin to experience a partially filled queue buffer at the router. This results in both increased latency and jitter (see Figure 5-18(ii) and (iii)). At t = 300 seconds, the queue buffers are increasingly filled resulting in traffic latency of approximately 49ms and jitter of 2.5ms which are well below the latency threshold of 150ms and jitter threshold of 30ms. The traffic does not experience any packet loss during the simulation (see Figure 5-18(iv)).

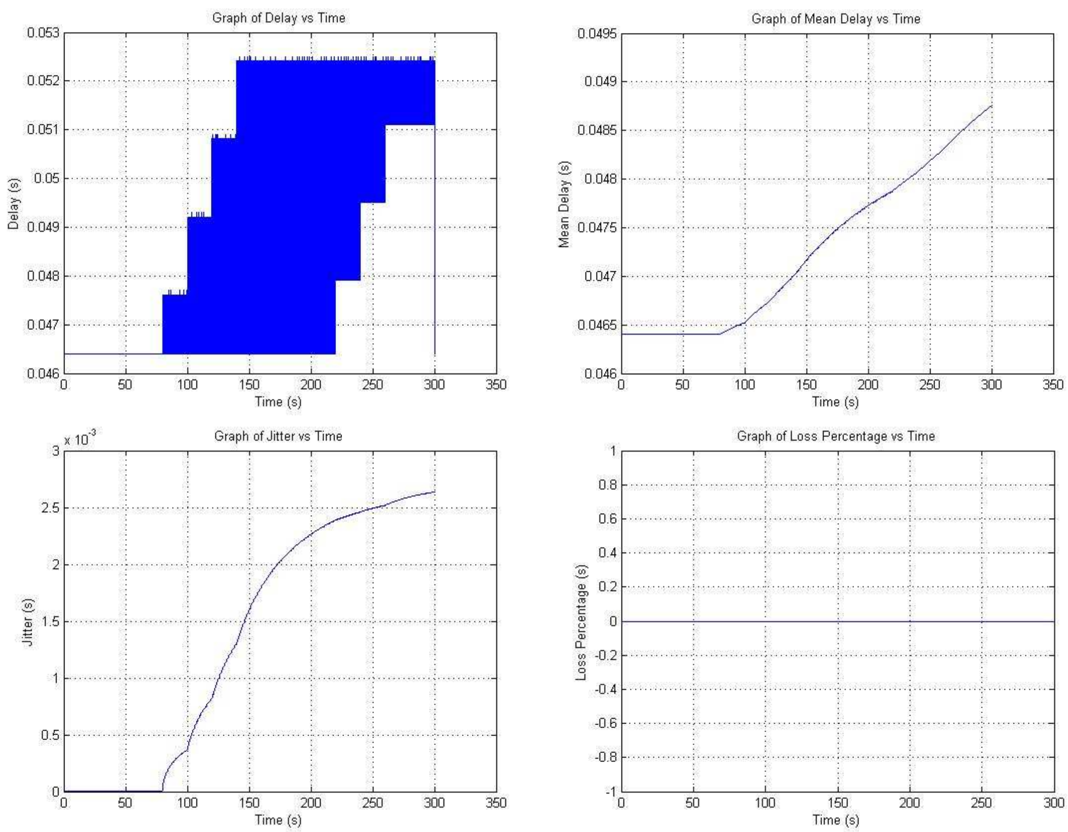

Figure 5-18: QoS Metric Performance traces from 2 Party Call

(i) Actual delay measured; (ii) latency; (iii) jitter; and (iv) packet loss

5.5.6 Simulation of the Calling Card Service

The simulation of the calling card service consisted of a single voice traffic source at 64kb/s. The simulation study consisted of the single voice call to both the IVR and the Media Gateway. The calling card service is initially connected to the IVR to authenticate the user and collect some information. This is illustrated in Figure 5-19 (i) and (ii) the period of $t \approx 0$ seconds to $t = 30$ seconds. Thereafter the call is disconnected form the IVR and rerouted to the Media Gateway as the called party is in a federated network. This is illustrated in Figure 5-19 (i) and (ii) the period of $t = 31$ seconds to $t = 150$ seconds. The latency experienced by the voice call is approximately 47ms while the jitter has a minimal effect increasing from 1,4ms to 2,4ms (see Figure 5-19 (iii)). The service does not experience any packet loss.

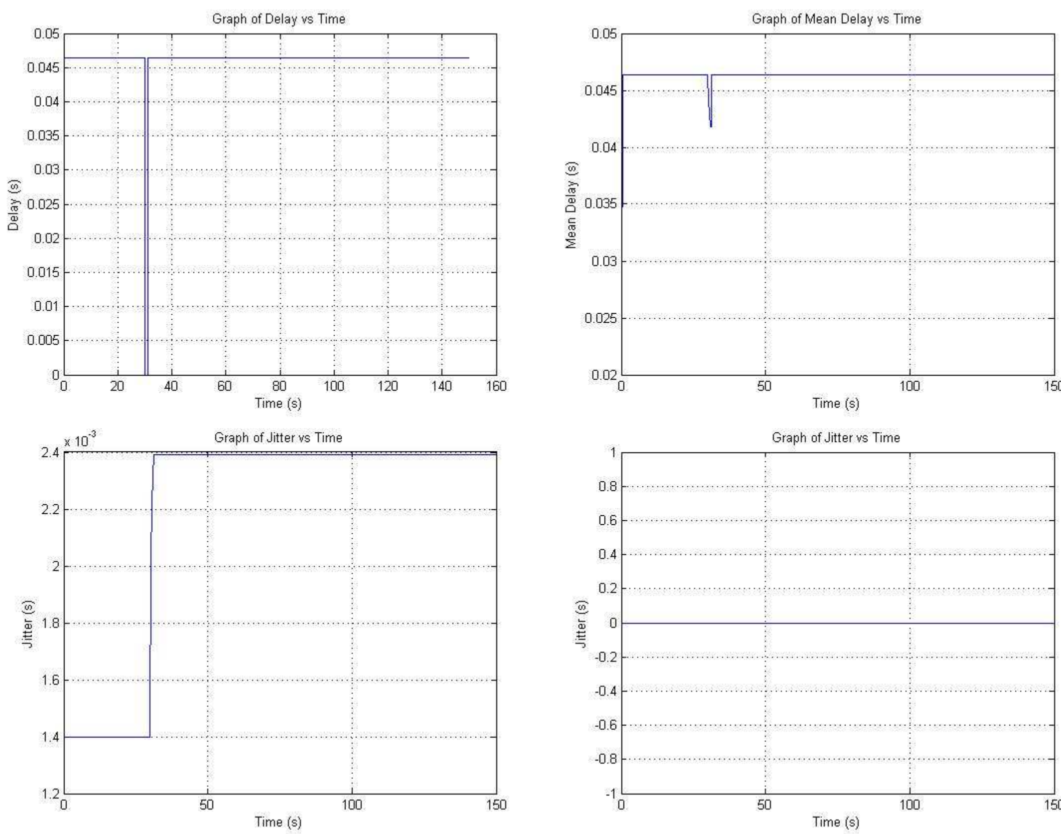

Figure 5-19: QoS Metric Performance traces from Calling Card Service

(i) Actual delay measured; (ii) latency; (iii) jitter; and (iv) packet loss

5.5.7 Effect on Route Selection based on dynamic Dijkstra Algorithm

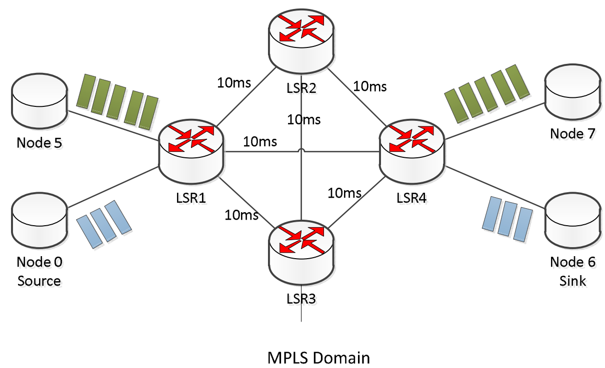

Figure 5-20: Topology simulated for effects on route selection

Consider a simplistic topology as depicted in Figure 5-20 above, with a symmetrical network consisting of 1Mb/s bandwidth and 10ms propagation delay between the links in the core. The traffic is sourced at Node 0 and terminates at Node 6. By application of the dynamic Dijkstra Algorithm, the initial best route is LSR1 – LSR4 with an initial propagation delay of 10ms (see Figure 5-21). When the first traffic flow begins entering the link, the network dynamics change. The admission of the second flow of traffic into the transport network takes this dynamic changes in the network into account and the best route selected by the algorithm is LSR1 – LSR2 – LSR4 with a propagation delay of 20ms. When the traffic begins to flow, the network dynamics change. On the request for admission of the third traffic flow, the best offered route is LSR1 – LSR3 – LSR4 with a propagation delay of 20ms (see Figure 5-21). The result of which is natural balancing effect on the network ensuring that all best performing links dynamically are utilised should it meet or exceed the requested performance criteria. Future work should be undertaken to research the effects on a complex network topology and study the performance of the priority in which paths are selected.

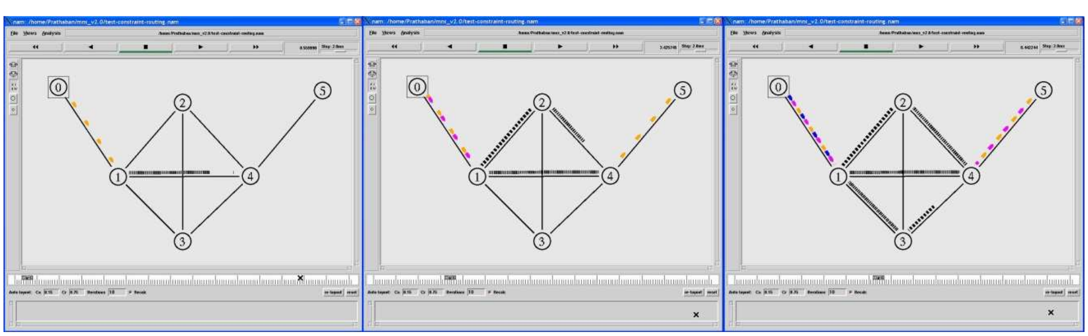

Figure 5-21: NAM illustration of route selection at increasing time intervals

5.5.8 Comparison of the Dynamic Dijkstra Algorithm against the Constraint Routed Algorithm

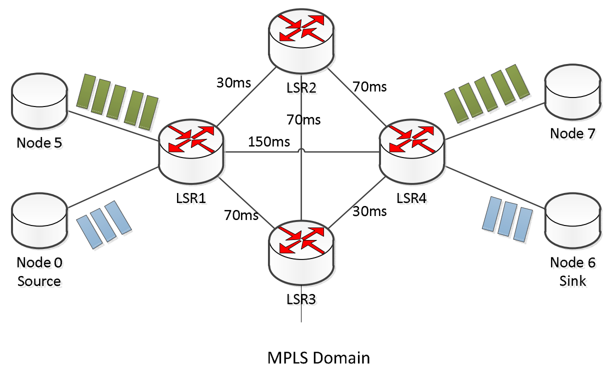

Figure 5-22: Topology simulated for effects on route selection

Consider a simplistic topology as depicted in Figure 5-22 above, with network consisting of 1Mb/s bandwidth between the links in the core and various propagation delays for each link. The traffic is sourced at Node 0 and terminates at Node 6. The following link propagation delays are set:

- LSR1 – LSR2 : 30ms
- LSR1 – LSR3 : 70ms
- LSR1 – LSR4 : 150ms
- LSR3 – LSR4 : 30ms
- LSR2 – LSR4 : 70ms
- LSR2 – LSR3 : 70ms

Application of the constraint routed algorithm to determine route selection. Each route request was for a bandwidth of 700kb/s in the network with voice traffic at the same rate. Due to the nature of the algorithm is based on selecting the path with the largest available bandwidth thereafter the shortest route in terms of number of router hops. As the bandwidth is identical for all links, the first route selected is from LSR1 – LSR4 which the shortest route in terms of number of router hops. The latency is observed for the first traffic flow in Figure 5-23(i). The initial latency for the link is 173ms which is well above the 150ms threshold for voice traffic. The latency increases on the admission of traffic flow 2 and traffic flow 3 at t ≈ 5.8 seconds and t ≈ 10.1 seconds respectively. Traffic flow 2 takes the path LSR1 – LSR2 – LSR4 and an initial latency of approximately 125ms is observed at t ≈ 5.8 seconds, after the effects of traffic flow 3 is applied the final latency results in approximately 125ms which is within the latency tolerance for voice traffic. Traffic flow 3 is admitted into the network along the path LSR1 – LSR3 – LSR4 at t ≈ 10.1 seconds and has a latency of approximately 126ms. This latency is also within the 150ms tolerance for voice traffic. The constraint routed algorithm makes an incorrect initial choice for the first traffic flow for an application such as voice traffic which has latency tolerance limitations.

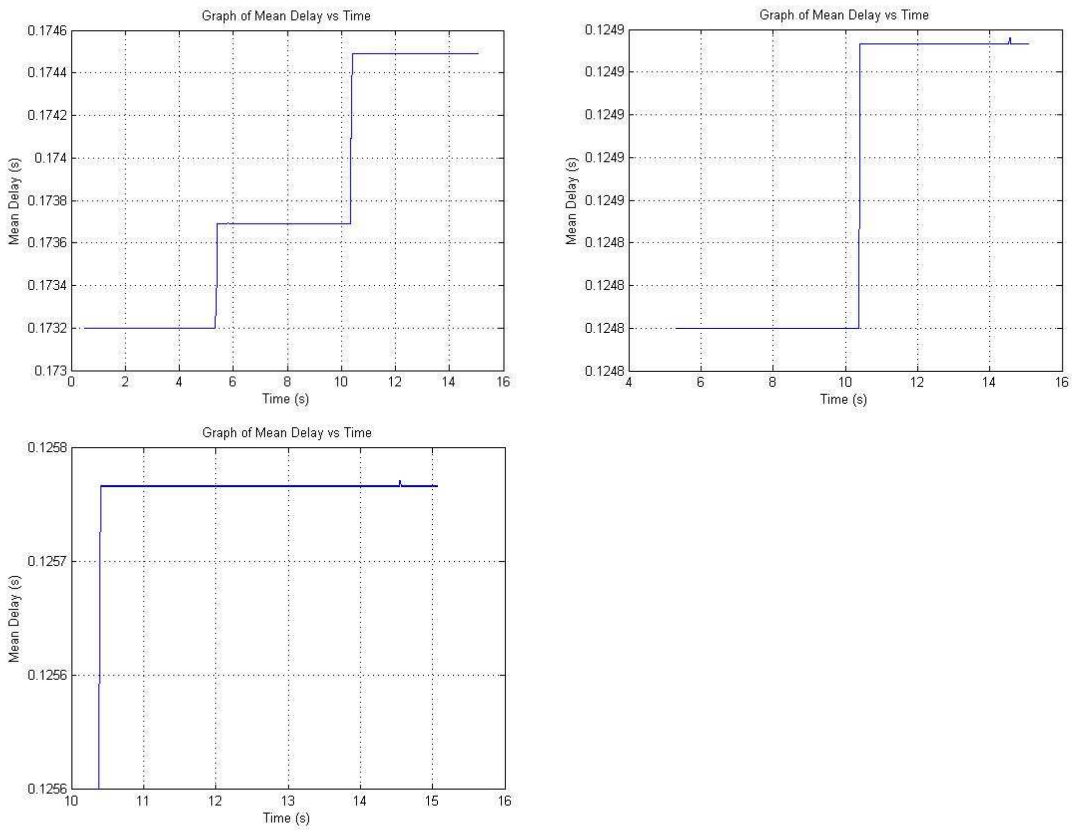

Figure 5-23: Latency curves for Constraint Routed Algorithm (i) Traffic flow 1; (ii) Traffic flow 2; (iii) Traffic flow 3

Application of the dynamic Dijkstra algorithm to determine route selection. Each route request was for a bandwidth of 700kb/s in the network with voice traffic at the same rate. Due to the nature of the algorithm the initial static shortest path is selected being the LSR1 – LSR2 – LSR4. The traffic source is admitted into the network at t ≈ 0.1 and the latency experienced by the traffic source is approximately 125ms and is within the voice latency tolerance of 150ms (see Figure 5-24(i)). Traffic flow 2 is admitted into the network at t ≈ 5.6 seconds experiencing a latency of approximately 126ms, also within the tolerance for voice traffic of 150ms (see Figure 5-24(ii)). The third request for traffic admission into the network is rejected as the latency for the path LSR1 – LSR4 is equal or greater than the threshold of 150ms. Future work should be undertaken to study the performance of the dynamic Dijkstra Algorithm against other routing algorithms. This future work should include studies into sensitivity of the input parameters and the time taken to resolve a shortest path in varying sizes of network topologies.

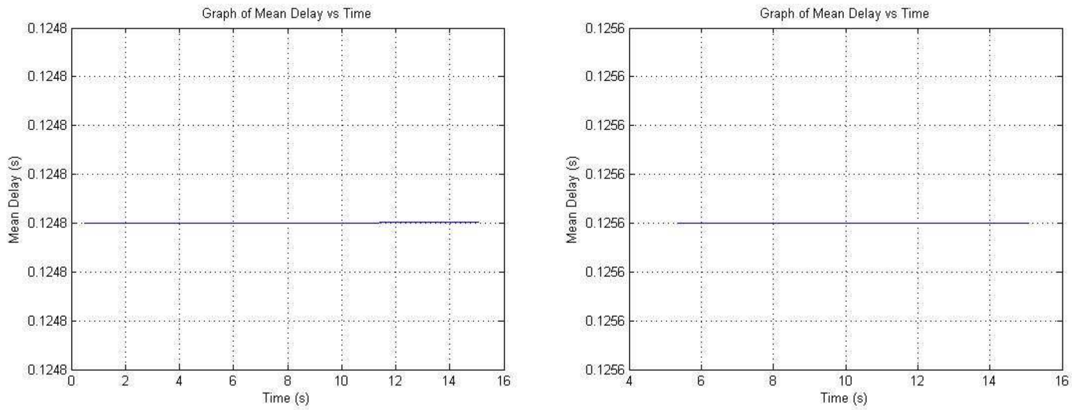

Figure 5-24: Latency curves for dynamic Dijkstra Algorithm (i) Traffic flow 1; (ii) Traffic flow 2

5.6 Discussion

The architectural characteristics of the OSA/Parlay Network Architecture and Interface are ascertained from an analysis of the detailed message sequence charts presented in section 5.4. The architectural characteristics are ascertained based on the characteristics of the control messages used to setup the connections (Moodley and Hanrahan 2008).

Further, basic use cases are presented in Appendix B illustrating the principles and concepts incorporated into the proposed OSA/Parlay Network Architecture Interface.

5.6.1 Analysis against Architectural Requirements

A. Simplicity: Any next generation network proposal must provision equal or better levels of service as compared to existing technologies such as the PSTN. The relatively small number of control messages required to setup basic calls within the same administrative domain is limited to 15 messages. By comparison to the TINA, MSF and 3GPP LTE cases, the message in the OSA/Pralay network architecture is an order of magnitude less. The minimal number of messages illustrate the simplicity of the layered OSA/Parlay network architecture

B. Technology Independence: The OSA/Parlay network architecture is technology agnostic as it has been designed using the API and associated methods rather than specific technologies and protocols. Apart from the Media Gateway function which implements a translation of the API based methods to specific protocols. The Media Gateway function addresses the requirement to interface to legacy based network which are technology specific. In addition, both the OSA/Parlay Network Architecture as well as the CAC functionality is technology independent and can be implemented on any technology such as the popular IP or MPLS transport technologies.

C. Open Interfaces: The OSA/Parlay Network Interface is specified using an open and standard API. The interface is specified in the CORBA IDL format which makes the interface open and implementable by a number of technologies. In addition the internal interfaces adopt a similar approach and therefore are also specified in an open and standard API.

D. QoS: The OSA/Parlay Network Interface provides a mechanism for the Application related functions to request connections together with QoS performance requirements. In addition the OSA/Parlay Network Architecture provisions a sophisticated QoS guaranteed mechanism using periodic probe traffic to update the state of the links in the entire network. This dynamic information is used when determining the connection routes from source to destination and the CAC function admits the call if the network path can meet or exceed the requested QoS.

E. Connection Admission Control: The OSA/Parlay Network Architecture incorporates a powerful mechanism for CAC, which uses the dynamic state of the network performance together with the requested connection and QoS performance to make the admit or reject decision.

F. Routing: Intelligent routing is achieved in the OSA/Parlay Network Architecture as the routing decision is not based on static network information such as distance between nodes (typically used by OSPF protocol). Rather the dynamic state of the network together with performance information based on loading conditions are incorporated into a simple routing mechanism based on the adapted Dijkstra's Algorithm to select the best route based on dynamic latencies rather than static distances.

G. Federation: The OSA/Parlay Architecture ensures that it has the ability to interoperate with different federated networks. The demonstration in section 5.4.2 illustrates the federation of the OSA/Parlay Network with both the federated PSTN and SIP networks. The federation can be extended to other technologies by updating the Media Gateway supported technology specific protocols.

H. Legacy: The Media Gateway element is used to deal with the case of legacy technologies. In particular the PSTN is fast becoming a legacy technology. The demonstration in section 5.4.2 illustrates the interoperation with the PSTN.

5.6.2 OSA/Parlay Network Interface

The OSA/Parlay Network Interface leverages the API mechanism. The API approach intrinsically ensures that the interface is open and technology-independent. The API is specified using generic methods for employing the OSA/Network Architecture. These methods are specified in a technology neutral description using IDL. The interface is able to realise end-to-end connections as well as special resources such as the media gateway as well as IVR. The interface supports application level QoS by incorporating a QoS negotiation method allowing for application level negotiation.

The methods that constitute the interface are simple and generic whilst not limiting basic or advanced functionality from the call and service control layer. Services Orientated Architectures are leveraged by the fast development of services in the information technology world. Services architectures such as OSA/Parlay define open interfaces wherein 3rd party application developers can develop advanced services without detailed knowledge of the underlying network architecture, using a level of abstraction provided by the Application Programming Interface (API) mechanism. This mechanism is replicated within the architecture as a technology independent interface between various network components hence contributing to the future proofing of the architecture while allowing integration of any of the rapidly developing transport technologies. The design of the interface achieves the goals set out in section 2.3 of *openness*, *simplicity*, *API* based, *QoS* and *technology independence*.

5.6.3 OSA/Parlay Network Architecture

The OSA/Parlay Network Architecture is not a detailed description but demonstrates concepts and principles. The OSA/Parlay Network Architecture is designed in a simple manner. Detailed examination of the message sequence chart indicate that minimal control messages are required to setup common calling scenarios. In particular the two party call within the OSA/Parlay Network Architecture requires 15 messages for call setup. When comparing the number of overhead messages to existing proposals such as 3GPP LTE or MSF, the number of message proliferate to 60 and 49 messages each respectively. This illustrates significant efficiency gains compared to current architectural proposals.

The OSA/Parlay Network Architecture has minimal architectural layers as well as computational objects. The architectural layers are clearly demarcated by functionality and consist of 4 layers which include the transport technology network elements. The message sequence charts illustrate API method requests on computational objects require that the object provides a decision, while not proliferating the messages to neighbouring objects as is the case with IMS type architectures and SIP messages traversing the network.

The tree structures architecture provisions the ability to consolidate network state information from a complex network towards the CC entity in a simplified and efficient manner. The use of RM-ODP in defining a technology agnostic architecture is depicted in the messages throughout the layers of the architecture which are technology independent and do not rely upon technology specific protocols, the messages use the technology agnostic API method technique.

QoS, intelligent routing and a call admission control mechanism are integrated into the architecture. The periodic network state updates inform the central CC object of both topological network changes as well as current performance of the constituent parts of the network. Intelligent routing of connections are applied using the adaptation of the Dijkstra's algorithm used to compute the best dynamic based path which is tested against QoS performance while the CAC decision naturally allows for load balancing of connection paths within the network.

The use cases which describe the interoperation with both SIP type networks as well as PSTN networks clearly demonstrate the architectural characteristics of federation to other telco networks as well as interoperation with legacy technologies. The use of the media gateway special resource is demonstrated in these use cases. The use case involving the IVR special resource illustrates the integration of the IVR special resource.

5.7 Chapter Summary

In order to demonstrate the OSA/Parlay Network Architecture and Network Interface, two services are exhibited. The first service is a simple 3rd party initiated 2-party call within the same administrative domain while the second service is a calling card service. Each use case is designed to illustrate key characteristics of the OSA/Parlay Network Architecture and Network Interface. These characteristics include *openness, simplicity, future-proof, QoS mechanisms* and supporting *federation* between telcos.

The demonstration is conducted over a Distributed Processing Environment (DPE) using CORBA. These architectural concepts and principles are demonstrated in a simulated environment and illustrate the NGN architectural characteristics. The demonstration environment is based on the Java programming language using the Netbeans integrated development environment (IDE). The Java environment incorporates the Java implementation of CORBA.

The results from the use cases demonstrate that the OSA/Parlay Network Interface is specified in the open, simple, technology independent API based manner. The interface supports QoS for end-to-end connections. The interface incorporates methods to leverage special resources such as the media gateway and IVR. The OSA/Parlay Network Architecture integrates intelligent routing, connection admission control as part of the overall QoS system. The architecture supports a simple layered architecture together with supporting federation of telcos and interoperability of legacy technologies. The OSA/Parlay Network Architecture is technology agnostic and hence future proof.

The CAC functionality was demonstrated separately in the ns-2 (UCB et al 2012) network simulation environment. ns-2 features simulating discrete events at packet level, providing insight into the effects on traffic at the packet level under various loading conditions and application of new routing algorithms. The three important components of guaranteed QoS are demonstrated. These are: (i) the intelligent routing of connections in the PSN; (ii) dynamic information of the network state; and (iii) making the call admission control decision on whether the network can meet or exceed QoS requested.

The adaptation of the Dijkstra's algorithm was coded for the CAC functionality within the simulator. The dynamic latencies of links need to be measured at runtime and used in making call admission and routing decisions during simulation. The ns-2 simulator traffic source and sink agents are modified to provide measurements of the link latencies at runtime. The measured latency is periodically polled during the simulation to update the dynamic nature of the links. When making path selection decisions, the dynamic link latencies are used as input parameters into the Dijkstra algorithm allowing for path selection based on the instantaneous state of the network.

The simulation of the probe video traffic provides insight into the dynamic performance metrics of the link. The CAC function makes use of the performance metrics, in particular the link latency used to make a call admission decision. The CAC applies latency and bandwidth thresholds for various applications; an example of video application requires a maximum threshold of 150ms of latency and 30ms of jitter. The CAC relies upon the constraint routed algorithm to ensure that the requested bandwidth is available in the selected path and provides the reservation mechanism. The CAC applies this logic to make the call admission decision.

The simulation of the busy call hour pattern demonstrates the effects of the CAC decisions to admit a new connection and its effects on existing traffic already admitted into the transport network. The CAC applies the call admission logic allowing for the network to be optimally utilised whilst ensuring that existing traffic does not breach the performance criteria in terms of latency. The demonstration of the CAC function provides confidence that guaranteed QoS is achievable and that the CAC function allows the network links to be used optimally without compromising the QoS performance.

A simplistic study of the dynamic Dijkstra algorithm approach against the constrained routed approach was undertaken. The results provide evidence for the cases of time critical applications such as voice and video applications are better served by path selection from the dynamic Dijkstra Algorithm which ensures that the path provided can accommodate the time critical thresholds as well as bandwidth requirements. The constraint routed approach is biased toward bandwidth availability and therefore does not consider aspects such as link latency. Future work should be undertaken to study the performance of the dynamic Dijkstra algorithm against other routing algorithms. This future work should include studies into sensitivity of the input parameters and the time taken to resolve a shortest path in varying sizes of network topologies.

Chapter 6

Conclusion

This chapter presents a summary of the research into the design of the OSA/Parlay Network Architecture and Network Interface. The novelty of this work, its limitations, and further work is presented.

6.1 Summary

This section presents the need for the OSA/Parlay Network Architecture and Network Interface as well as the key interface and architectural characteristics. The background to existing network architectures and interfaces are presented. Useful concepts and principles are highlighted and synthesised for the design of the OSA/Parlay Network Architecture and Network Interface. A brief summary of the result from the demonstration of the OSA/Parlay Network Architecture and Network Interface is described.

6.1.1 Requirement for the OSA/Parlay Network Architecture and Network Interface

Telcos are adapting their business to address the rapidly changing technology landscape (Moodley and van Olst 2011). The convergence trend between telecommunication and information technologies has contributed toward downward pressures on traditional sources of revenue for the Telco. Therefore operators view value-added services as a means to secure new sources of revenue at premium tariffs and direct additional traffic to the network (Moodley and Hanrahan 2006). Telcos require a flexible architecture to allow seamless adaptation and to leverage these new technologies to gain a competitive advantage (Moodley and van Olst 2011).

The OSA/Parlay Gateway is a proposal which addresses the requirements of the service stratum and the interface to the Application layer. A key feature of the OSA/Parlay Gateway is an open, standard, application-programming interface (API). The API provides a level of abstraction in which the application engineer does not need knowledge of the underlying network to develop services. Similarly, the network designer uses general rather than specific service considerations. Another central feature is the migration of intelligence from high-performance centralised hardware such as the traditional Intelligent Network (IN) towards the edges of the network in a Distributed Processing Environment (DPE). However, the Parlay Group has not specified the Network Interface, it has alluded to the use of protocol adapters for the interface. (Moodley and Hanrahan 2005)

Technology dependent adapters would have a negative effect on the future proof characteristics of the gateway. Due to the rapid technological changes in the telco environment and the need for transport network architectures to interoperate with legacy networks, this interoperation becomes very complex. In particular the transport technologies have proliferated and hence a large number of technology specific protocols must be considered for interoperation, inter alia, INAP, TCP/IP, SIP, H.323, MAP and CAMEL. This ability to interoperate with legacy technologies therefore make network architectural proposals inflexible when considering technology specific protocols and unable to respond to the rapid technological, regulatory and market changes. This inflexibility strongly supported the requirements for a technology independent transport stratum research.

This research is focused on the *transport stratum* as an extension to the OSA/Parlay architecture; we refer to this as the OSA/Parlay Network Architecture. Key features of the OSA/Parlay NA are the provisioning of end-to-end transport resources in response to the service request for a particular session. Another key feature is the Call Admission Control (CAC) function which ensures that a new call request admitted into the network is guaranteed a level of quality of service (QoS) for the entire call duration; whilst ensuring that existing calls already admitted into the network do not experience any degradation in QoS. The OSA/Parlay Network Architecture and Network Interface should be characterised as open, standard and technologically independent. (Moodley and Hanrahan 2006)

In order to design the future flexible network architecture that explicitly engineers the network for provisioning converged services, generic network architectural characteristics need to be described. The network architecture is not a detailed definition. Rather it provides the concepts and principles for connections involving the constituent networks (packet and circuit switched) and single elements (media gateways and IVRs). The architecture encompasses the following characteristics (Moodley and Hanrahan 2008):

- “be designed as a *simple* architecture;
- demonstrate *technology independence*;
- support *open interfaces*;
- enforce an integrated approach for *end-to-end connections*;
- implement connection admission *QoS* control as part of the overall QoS assurance system;
- provision *intelligent routing* of connections;
- support *federation* between network operators to provide QoS-assured end-to-end connections;
- support *legacy* technologies in the overall architecture.”

6.1.2 Existing Network Architectures and Network Interfaces

A number of existing architectural proposals for the transport stratums have been reviewed against the ideal network architectural characteristics described above. The TINA NRA is founded on a technology agnostic principle and specified the interface to the service architecture (service stratum). The session model concept is useful for managing complex connections and service requests. The concepts of some computational objects arranged in a layered architecture provide insight for the design of the OSA/Parlay Network Architecture. Lessons such as addressing interoperability to federated networks and addressing the interoperability with legacy technologies are useful.

The MSF architecture specifies a useful QoS management system, leveraging on a central entity acting as the single point of contact for all connection requests in the network. The admission decision is based on network state information. The network edge elements ensure that no traffic from the terminals traverse the network without an admission decision from the BM. The 3GPP LTE architecture specifies an alternative QoS management system, incorporating a single point of contact for all connection requests in the network. QoS rules determine admission decision. Network Edge element allows or prevents use of the resource based on the admission decision using the temporary IP address mechanism.

The OSA/Parlay Gateway specifies a powerful mechanism of an *open, standard, simple, secure* and *technology independent* API based Application interface defined in the object oriented paradigm and specified in UML. JAIN addresses an approach to protocol adaptation to federated and legacy network technologies concepts are useful and can be adapted for the design of the OSA/Parlay Network Architecture Media Gateway. Java based technology provides a quick and simple means for realising the OSA/Parlay Gateway.

6.1.3 Design of the OSA/Parlay Network Architecture and Network Interface

The API based OSA/Parlay Network Interface is characterised as an open, standard and technologically independent interface. The OSA/Parlay Network Architecture should be future proof from rapid technology developments and facilitate ease of simple integration of a variety of transport technologies. It conforms to the characteristics of being a simple layered architecture specified in a technology agnostic manner using the RM-ODP methodology. The architecture supports an integrated approach to intelligent routing, QoS and call admission control for end-to-end connections. The architecture must explicitly support both federation of telcos and interoperability with legacy technologies.

An information organising principle relies on the use of sessions. A session is a temporary allocation of resources that are configured to service a task for a period of time (TINA NRA 1997). The detailed architecture describes key RM-ODP computational objects and their purpose. The Communication Session Manager (CSM) is responsible for the communication between the transport network and the Parlay Gateway. The CSM contains the service envisioned view of the call within the Logical Connection Graph (LCG) for each instantiated call. The Connection Coordinator (CC) is responsible for translating the LCG into physical connections. A core function is to intelligently route the connection and in so doing determine the layers that are to be used for the connection and selects the appropriate Layer Network Coordinators (LNCs). It also provides the function of admission control which determines whether the requested call properties (QoS) can be supplied by the transport network and on that basis, does a call admission control function. The CC is also responsible for managing the Administrative Federation on a call by call basis.

The Layer Network Coordinator (LNC) is responsible for the connection within the layer transport network. In addition, the LNC provides network state information on QoS performance of links and nodes in the network. Legacy protocol adaptation is presented based on the JAIN approach, this is incorporated into the media gateway function to interoperate with legacy technologies.

6.1.4 Demonstration of the OSA/Parlay Network Architecture and Network Interface

In order to demonstrate the OSA/Parlay Network Architecture and Network Interface, two services are exhibited. The first service is a simple 3rd party initiated 2-party call within the same administrative domain while the second service is a calling card service. Each use case is designed to illustrate key characteristics of the OSA/Parlay Network Architecture and Network Interface. The demonstration is conducted over a Distributed Processing Environment (DPE) using CORBA. These architectural concepts and principles are demonstrated in a simulated environment and illustrate the NGN architectural characteristics. The demonstration environment is based on the Java programming language using the Netbeans integrated development environment (IDE). The Java environment incorporates the Java implementation of CORBA. The results are presented in Chapter 5 of this thesis.

In summary, the analysis of the simple 3rd party initiated call indicates that a minimal number of 17 messages are required for a simple 2-party call. This is at least an order of magnitude more efficient than the existing architectural options. The simple architecture allows for decisions to be made at the lowest layer possible and the proliferation of messages between objects are minimised. The interfaces to the internal objects are based on the API concept ensuring technology agnostic and future proofing characteristics. Similarly the interfaces to the special resources such as the IVR and Media Gateway are based on the API concept ensuring the interface is technology agnostic and future proof. The architecture incorporates intrinsic QoS mechanism which provides application related performance functions. The routing decision is based on the dynamic state of the network and the CAC makes the admission or rejection decision based on the QoS performance requested from the connection.

6.2 Contribution of this work

The OSA/Parlay standard specifies the call and service control functions together with the open interface to the application domain. The Parlay standard has not specified the interface to the transport stratum and has alluded to the use of technology dependent protocol adapters. In this research the OSA/Parlay compliant Network Architecture and API based Network Interface are proposed. The proposed OSA/Parlay Network Interface is characterised by *openness*, *simplicity, API* based approach, integrated *QoS* support and *technology independence*. The OSA/Parlay Network Architecture features *simplicity, technology independence, QoS mechanisms; call admission control; intelligent routing* and supporting both *federation* of telcos and interoperation of *legacy* technologies.

Thus the OSA/Parlay Network Architecture and its Interface are an extension to the existing OSA/Parlay Gateway and realise the OSA/Parlay next generation network. Therefore the open architecture allows for 3rd party application developers while also ensuring that call and service requests are provisioned end-to-end with guaranteed, application level QoS in the transport network. The OSA/Parlay Network Architecture and Network Interface are synthesised from existing architectural standards and provides the following benefits:

- It is a proposal for the OSA/Parlay NGN
- It is technology independent facilitating multiple transport technologies
- It facilitates Application related QoS
- It achieves the goal of guaranteed QoS

The extension of the OSA/Parlay standard by including the OSA/Parlay Network Architecture and Network Interface realise the OSA/Parlay next generation network. The proposal is a packet-based network which facilitates advanced services. The architecture is technology agnostic and therefore supports multiple transport technologies. The integrated intelligent routing, call admission control and QoS mechanisms enable the application level QoS features. The OSA/Parlay API based Network Interface supports decoupling of the call and service control from the transport network.

Both the OSA/Parlay Network Interface and the Network Architecture are specified in a technology agnostic manner. This ensures that the architecture remains future proof as it is not reliant on any particular technology. The future proof characteristic allows the architecture to be flexible in that transport technologies can migrate without affecting both the service and control layer as well as the network architecture.

The long sought after Application level QoS is integrated into the architecture. The intelligent routing and a call admission control mechanism are integrated into the architecture. The periodic network state updates inform the central CC object of both topological network changes as well as current performance of the constituent parts of the network. Intelligent routing of connections are applied using the adaptation of the Dijkstra algorithm, this is used to compute the best dynamic path which is tested against QoS performance while the CAC decision naturally allows for load balancing of connection paths within the network. The routing of the connection is done holistically based on the network's topology and state unlike current proposals such as RSVP which relies on local node information to setup connections in the network. This QoS mechanism achieves the goal of guaranteed QoS for a call admitted into the network relying on the application level QoS principle should the QoS need to be negotiated.

6.3 Answering the Key Research Questions

This research set out to answer key research questions for this research project. The design of the proposed OSA/Parlay Network Architecture and Interface together with the proposal for the call admission control function has been presented and demonstrated. We review these key research questions to determine if these questions have been satisfied.

- *Can the OSA/Parlay gateway be extended to the transport stratum using similar concepts as the gateway?*
  Yes. The OSA/Parlay Network Interface and Network Architecture is a proposal for the transport stratum. It leverages the openness, simplicity and technology agnostic characteristics which is the basis of the OSA/Parlay gateway principles.

- *Can the proposed OSA/Parlay Network Interface used to separate the Call and Service Control functions from the Transport Network be specified using an open, standard and technology agnostic API supporting QoS?*
  Yes. The OSA/Parlay Network Interface relies on the open, standard and technology agnostic API mechanism to future proof the network interface. The interface specifies methods for call setup, connection routing and establishment of notifications. The setting up of connections in the network architecture provides the mechanism for the application to a make QoS based requests. Additionally, the interface supports Application related QoS negotiation, providing details of the best offered QoS when the application request cannot be realised by the dynamic state of the network resources.

- *Can the proposed OSA/Parlay Network Architecture be specified in a technology agnostic manner, allowing it to be future proof and applicable to rapidly evolving transport technologies?*

  Yes. The design of the OSA/Parlay Network architecture conforms to the RM-ODP viewpoints which specify the system in an open, technology independent and distributed philosophy. In particular the information and computational objects are technology agnostic and rely on internal open interfaces for communication and distribution. The demonstration in the CORBA environment allows the network architecture to be implemented in any programming language (Java, C++, etc.). Additionally, the OSA/Parlay Network Architecture can be implemented on any packet based transport technology such as IP or MPLS.

- *Can the Call Admission Control (CAC) function to be designed as an integral part of the Network Architecture to ensure guaranteed QoS for all calls within the network and calls still to be admitted into the network?*

  Yes. Guaranteed QoS has been a long sought after goal for packet switched technology (Song et al 2008). QoS is the assurance that prior to its admission, a new connection awaiting admission into the network is guaranteed a level of performance within the network for the entire connection duration (Moodley and Hanrahan 2007), while the QoS levels of existing calls already admitted into the network does not deteriorate. A unique contribution of this research is principled upon making intelligent routing decisions based on the dynamic state of the network. The intelligent routing and a call admission control mechanism are integrated into the architecture. The periodic network state updates inform the central Route object of both topological network changes as well as current performance of the constituent parts of the network. Intelligent routing of connections are applied using the adaptation of the Dijkstra algorithm used to compute the best dynamic path which is tested against QoS performance while the CAC decision naturally allows for load balancing of connection paths within the network. This QoS mechanism achieves the goal of guaranteed QoS for a call admitted into the network.

- *Can the proposed OSA/Parlay Network Architecture be implemented with functionality to integrate to federated telco networks as well as legacy networks, which are protocol specific?*

  Yes. The OSA/Parlay Network Architecture has been designed to support both federated networks and legacy technologies. This functionality is relied upon by the Media Gateway element. An interface to the Media Gateway has been designed which facilitates the translation from the API methods to specific protocols such as SIP or ISUP. These cases have been demonstrated in section 5.4.2 and in Appendix B.

- *Can an alternate more powerful routing concept be integrated as a function of the proposed OSA/Parlay Network Architecture?*
  Yes. A unique contribution of this research is the design and demonstration of the intelligent routing concept. This concept makes use of dynamic network state information which is used as an input in the adaptation of the Dijkstra Algorithm. This algorithm has a dynamic holistic view of the network nodes and links. The best route from a particular source to destination node is determined and can be specified to be calculated on the basis of the shortest latency, best jitter or packet loss performance. The selected route is considered against all call requested parameters to confirm or reject the call request. The connection setup is done holistically unlike current proposals such as RSVP which relies on local node information to setup connections in the network.

6.4 Research Limitations

The OSA/Parlay Network Architecture is not a detailed definition; rather it provides the concepts and principles for connections involving the constituent networks (packet and circuit switched) and single elements (media gateways and IVRs) (Moodley and Hanrahan 2008; Moodley and van Olst 2011). Therefore the demonstration of the design was limited to illustration of the principles and concepts in a simulated environment. In particular functionality designed in the Network Architecture is dependent on the OSA/Parlay Gateway, Consumer Domain and Transport Technology which was emulated.

The interface between the OSA/Parlay Gateway and the OSA/Parlay Network Architecture is assumed to be within a trusted domain of the telecommunications operator and retailer. The use of the Network Resources requires prior service agreements which are not considered in this research.

Security of the Network Architecture and associated Interface requirements for cross-domain invocation are not considered. Similarly, authentication, authorisation and accounting of the network operator is not considered.

6.5 Future Work

The scope of this work was limited to two party calls. However, an important functionality for telcos is the ability to provide multiparty call services. We propose that future work investigate the possibility of including a conference bridge network element into the architecture. The routing of the connection paths must be demonstrated from the calling parties toward the conference bridge network element. Alternatively, the technologies such as multicasting should be considered for application of multiparty calls.

The provisioning of the edge enforcement function is the proposal of using a temporary IP address is allocated to each registered terminal on the network. This temporary IP address is allocated to a sub-masked network in which only signalling messages are accommodated and this subnetwork is logically separated from the transport network resources. When the connection requests are admitted to the network a session based IP address and subnet mask is allocated to provide access to the network resources. However, this mechanism seems to be very mechanical in approach. Future work should address a more elegant solution.

The interoperation of the OSA/Parlay Network Architecture with physical transport technologies was outside the scope of this research. However, it is important that the future work looks into implementing the architecture over a transport technology. Since the architecture is technology agnostic the work should incorporate both the popular IP routing and MPLS routing functions.

Due to the inherent function of explicit routing of connections in MPLS networks, the CAC decision is easily translated into an explicit routing connection setup instruction in an MPLS network. The mechanism to be demonstrated is to compare the performance in processing as well as guaranteed QoS of the MPLS routing protocols against the routing function developed in the OSA/Parlay Network Architecture.

The time period used to probe the state of the network is critical to ensuring that the network state information is reflective of the current network performance. If this period is too long, important performance information may be excluded. While if the time period is too short this will lead to significant overhead in the OSA/Parlay Network Architecture.

Selection of the time period should not be a static value, as the loading of the network is time dependent and the load curve is based on time of use tariffs implemented by the telco. Further research should correlate the time period to the load curve of a network to have much more reflective state information.

The QoS guaranteed function can be independently applied to any packet based technology without the need to implement the OSA/Parlay Network Architecture. The QoS function is technology agnostic and can be applied to any packet switched network. Thus further work should focus on application of the QoS guaranteed functionality on a packet switched network such as IP or MPLS.

The QoS function could be further enhanced by instead of identifying a single best route within the network using the adaptation of the Dijkstra Algorithm, a prioritised list of the three best paths from source to destination can be determined using the K^{th} shortest path algorithms as described by Yen's Algorithm (Yen 1970) or Eppstein's Algorithm (Eppstein 1999).

The use of probe traffic to determine the performance of all links with the network does consume resources. Further work should be undertaken to determine whether performance details of the links within the network can be ascertained from live traffic traversing the link instead of probe traffic. This would certainly reduce the overhead of this QoS mechanism.

The interface and methods specification of the Network Architecture to Terminal signalling was outside the scope of this thesis. However, this is an important element to implement the OSA/Parlay Network Architecture. It is proposed the future work should be undertaken in this area.

Security aspects of the architecture were outside the scope of this research. For practical application of the OSA/Parlay Network Architecture and Interface, these aspects will need to be addressed. Similar research should be undertaken for OSS/BSS aspects such as billing.

Future work should be undertaken to study the performance of the dynamic Dijkstra algorithm against other routing algorithms. This future work should include studies into sensitivity of the input parameters and the time taken to resolve a shortest path in varying sizes of network topologies. Additionally, studies to determine the effects on a complex network topology and the performance of costs in terms of processing delays should be investigated.

6.6 Conclusion

The research contribution achieves an open architecture allowing for 3rd party application developers while also ensuring that call and service requests are provisioned end-to-end with guaranteed application level QoS in the transport network. The OSA/Parlay Network Architecture and Network Interface is synthesised from existing architectural standards and provides the following benefits. It is an extension of the OSA/Parlay standard by including the OSA/Parlay Network Architecture and Network Interface realises the OSA/Parlay next generation network. Both the OSA/Parlay Network Interface and the Network Architecture is specified in a technology agnostic manner. This ensures that the architecture remains future proof as it is not reliant on any particular technology. The long sought after Application level QoS is integrated into the architecture. The periodic network state updates inform the central Connection Coordinator object of both topological network changes as well as current performance of the constituent parts of the network. Intelligent routing of connections is achieved by adapting the Dijkstra algorithm to compute the best path based on dynamic network performance and is tested against QoS requirements, while the call admission control decision naturally allows for load balancing of connection paths within the network. This QoS mechanism achieves the goal of guaranteed QoS for a call admitted into the network.

References

[3GPP OSA Part 1 2009] 3GPP, "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Release 9)", 3rd Generation Partnership Project, 2009

[3GPP OSA Part 4-2 2009] 3GPP, "Open Service Access (OSA); Application Programming Interface (API); Part 4: Call control; Sub-part 2: Generic call control Service Capability Feature (SCF) (Release 9)", 3rd Generation Partnership Project, 2009

[3GPP OSA Part 4-3 2009] 3GPP, "Open Service Access (OSA); Application Programming Interface (API); Part 4: Call control; Sub-part 3: Multi-Party Call Control SCF (Release 9)", 3rd Generation Partnership Project, 2009

[3GPP OSA Part 4-4 2009] 3GPP, "Open Service Access (OSA); Application Programming Interface (API); Part 4: Call control; Sub-part 4: Multi-Media Call Control SCF (Release 9), 3rd Generation Partnership Project, 2009

[3GPP OSA Part 5 2009] 3GPP, "Open Service Access (OSA); Application Programming Interface (API); Part 5: User Interaction SCF (Release 9)", 3rd Generation Partnership Project, 2009

[3GPP IMS Session Handling 2003] 3GPP. TS 23.218: IP Multimedia (IM) Session Handling; IP Multimedia (IM) call model, Stage 2 (release 12). 3rd Generation Partnership Project, 2012

[3GPP IMS 2012] 3GPP. TS 23.228: IP Multimedia Subsystem (IMS); Stage 2 (release 11). 3rd Generation Partnership Project, 2012

[3GPP NA 2003] 3GPP. TS 23.002: Network Architecture (Release 6). 3rd Generation Partnership Project, 2003.

[3GPP Signal Flows IMS 2012] 3GPP. TS 24.228: Signalling Flows for the IP Multimedia Call Control Based on SIP and SDP. 3rd Generation Partnership Project, 2003.

[Atanasov I 2013] Atanasov I, “Study on Deployment of Web Services for User Interaction in Multimedia Networks”, Cybernetics and Information Technologies. Volume 13, Issue 2, Pages 63–74, ISSN (Online) 1314-4081, ISSN (Print) 1311-9702, May 2013

[Atanasov I et al 2012] Atanasov I, Pencheva E, Marinska D. “Parlay X web services for policy and charging control in multimedia networks”, Adv. MultiMedia 2012, Article 2 (January 2012)

[Arjun et al 1999] Anjum F, Caruso F, Jain R, et al, “ChaiTime1: A System for Rapid Creation of Portable Next-Generation Telephony Services Using Third-Party Software Components”, IEEE Conf. on Open Architectures and Network, 1999

[Bakker et al 2000] Bakker J, McGoogan JR, Opdyke WF, Panken F, "Rapid development and delivery of converged services using APIs", Bell Labs Technical Journal, Volume 5, Issue 3, pages 12–29, 2000

[Beddus and Bruce 2000] Beddus S and Bruce G, “Opening Up Networks with JAIN Parlay”, IEEE Communications Magazine, April 2000

[Bertrand 2007] Bertrand G, "The IP Multimedia Subsystem in Next Generation Networks", May 2007, www.tele.pw.edu.pl/~mareks/auims/IMS_an_overview-1.pdf, last accessed: Jan 2013

[Bhat and Gupta 2000] Bhat RR and Gupta R, “JAIN Protocol APIs”, IEEE Communications Magazine, January 2000

[Blum et al 2008] Blum N, Magedanz T, Schreiner F, "Services, Enablers and Architectures: Definition of a connected Web 2.0 / Telco Service Broker to enable new flexible Service Exposure Models", 12th International Conference on Intelligence in Information and Communication Networks, ICIN 2008. Proceedings : Bordeaux, 20-23 October 2008

[Boeringer 2006] Boeringer, R. RSVP-TE patch for MNS/ns-2. http://www.ideo-labs.com/index.php?structureID=44. Last accessed on August 2006

[Branca and Atzori 2012] Branca G and Atzori L, "A Survey of SOA Technologies in NGN Network Architectures", IEEE Communications Surveys & Tutorials, Vol. 14, No. 3, Third Quarter 2012

[Branca 2012] Branca G, "Architectures and Technologies for Quality of Service Provisioning in Next Generation Networks", University of Cagliari, Italy, March 2012

[de Keijzer et al 2000] de Keijzer J, Tait D, and Goedman R, “JAIN: A New Approach to Services in Communication Networks”, IEEE Communications Magazine, January 2000

[Eppstein 1999] Eppstein D, “Finding the k Shortest Paths” SIAM Journal on Computing, vol.28, no.2, pp.652–673, 1999

[Ericsson 2009] Ericsson HE, "QoS Control in the 3GPP Evolved Packet System", Communications Magazine, IEEE , vol.47, no.2, pp.76-83, February 2009

[Femminella et al 2012a] Femminella, M; Giacinti, F; Reali, G, “An Extended Java Call Control for the Session Initiation Protocol”, MultiMedia, IEEE, Volume: 19, Issue: 4 , pp.60-71, 2012

[Femminella et al 2012b] Femminella, M; Giacinti, F; Reali, G, “Introduction of Media Gateway Control functions in Java Call Control” , MultiMedia, IEEE, Volume: 19, Issue: 4, pp.1-6, 2012

[Gallon MSF 2003] Gallon C, “Quality of Service for Next Generation Voice Over IP Networks”, MultiService Switching Forum, February 2003, www.msforum.org

[Gylterud and Tognon 2003] Gylterud G and Tognon S, "Non-functional aspects of OSA", Proceedings of the Eighth International Conference on Intelligent Networks (ICIN), Bordeaux, France, April 2003

[Hameed et al 2009] Hameed S, Raza A, Badii A, Lee S, "Converged Next Generation Network Architecture & its Reliability", European Conference on Modelling And Simulation, June 2009

[Hanrahan and Mwansa 2003] Hanrahan HE and Mwansa D, "A Vision for the Target Next Generation Network", Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC), 2003

[Hanrahan 2007] Hanrahan HE, 'Network Convergence: Services, Applications, Transport and Operations Support", first edition, Wiley, New Jersey, USA, 2007

[Huang 2012] Huang, P., “ns-2 Tutorial”, http://www-scf.usc.edu/~bhuang/ nsberkeley.ppt, last accessed July 2012.

[Jain et al 2000] Jain R, Anjum FM, Missier P, and Shastry S, “Java Call Control, Coordination, and Transactions”, IEEE Communications Magazine, January 2000

[Ke 2011] Ke CH, “describes how to measure UDP packet of One Way Delay (OWD), IP Delay Variance (IPDV), and Packet Loss count” translated from Chinese, http://140.116.164.80/~smallko/ns2/measure.htm, last accessed January 2011

[Kim 2004] Kim E, "ITU Activities - NGN", http://www.itu.int/ITU-D/treg/Events/Seminars/2004/China/04-S2-ITU.pdf, last accessed: Jan 2013

[Knightson 2013] Knightson K, "What is NGN: Basic Architecture", www.itu.int/ITU-T/worksem/grid/presentations/s2p1-knightson.pdf, last accessed: Jan 2013

[Li et al 2007] Li B, Wang D, Zhang S, "Policy Based SIP Signaling Management in IMS", Proceedings of 11th International Conference on Intelligence in Service Delivery Networks, ICIN, Bordeaux, France, 2007

[Lim et al 2003] Lim S, O’Doherty P, Ferry P, Page D, “JAIN™ SLEE Tutorial”, 2003, Sun Microsystems, Inc.

[Lei-an and Hong-jiang 2010] Lei-an L, Wang Hong-jiang, "Research and realization of mobile value-added service based on parlay/OSA", Computer Engineering and Design, Vol.31, No.13, pp 2980-2983., 2010

[Lu 2006] Lu H, "Quality of Service for Next Generation Networks Generation Networks", https://www.itu.int/ITU-T/worksem/ngn/200603/presentations/s2_lu.pdf, last accessed: Jan 2013

[Magedanz 2012] Magedanz T, "Overview of the IP Multimedia System (IMS) - Principles, Architecture and Applications", http://www.fokus.gmd.de/testbeds/ims_playground/Paper/FOKUS-IMS-Tutorial-public.pdf, last accessed: Dec 2012

[Magedanz et al 2007] Magedanz T, Knüttel K, KathO, "MDA-based Service Creation for OSA/Parlay within 3Gbeyond Environments", Computer, Volume 40 Issue 11, pp 46-50, Nov 2007

[Magedanz et al 2005] Magedanz T, Witaszek D, Knuettel K, Weik P, "The IMS Playground @ Fokus – An Open Testbed for Next Generation Network Multimedia Services", Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on , vol., no., pp. 2- 11, 23-25 Feb. 2005

[Moerdijk and Klostermann 2003] Moerdijk AJ and Klostermann L, "Opening the Networks with Parlay/OSA: Standards and Aspects behind the APIs", IEEE Network, vol. 17, no. 3, pp. 58–64, May-June 2003

[Moodley and Kennedy 2005] Moodley, P., and Kennedy, I.G., "A position paper: Modelling self-similar traffic", ICT 2005 - 12th International Conference on Telecommunications 3-6 May 2005, Cape Town. © 2005 IEEE. ISBN: 0-9584901-3-9 Reprinted in Trans SAIEE.

[Moodley and Hanrahan 2005] Moodley, P.V., and Hanrahan, H.E., "A Network Application Programming Interface for OSA/Parlay", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2005 Proceedings, September 2005

[Moodley and Hanrahan 2006] Moodley, P.V., and Hanrahan, H.E., "Design of a Parlay Network Application Programming Interface and Associated Architecture", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2006 Proceedings, September 2006.

[Moodley and Hanrahan 2007] Moodley, P.V., and Hanrahan, H.E., "Investigation into Performance Metrics for Connection Admission Control in an MPLS Simulated Network", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2007 Proceedings, September 2007

[Moodley and Hanrahan 2008] Moodley, P.V., and Hanrahan, H.E., "Evaluation of the IMS-based MSF architecture against network architectural requirements", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2008 Proceedings, September 2008.

[Moodley and van Olst 2011]	Moodley, P.V., and van Olst, R., “RM-ODP design of the OSA/Parlay Network Interface and Associated Architecture”, IEEE AFRICON 2011, 13-15 September 2011, Livingstone, Zambia. © 2011 IEEE. ISBN: ISBN 978-1-61284-991-1
[MSF 2006]	MultiService Switching Forum, "MSF Release 3 Architecture", June 2006, www.msforum.org
[Pirhadi et al 2009]	Pirhadi M, Hemami SMS, Khademzadeh A, "Resource and Admission Control Architecture and QoS Signaling Scenarios in Next Generation Networks", World Applied Sciences Journal, Volume 7 (Special Issue for Computer & IT), pp 87-97, 2009
[Qiao et al 2011]	Qiao X, Li X, Fensel A, Su F, “Applying semantics to Parlay-based services for telecommunication and Internet networks”, Central European Journal of Computer Science, Volume 1, Issue 4, pp 406-429, December 2011
[RM-ODP 1995]	ISO/IEC 10746-1, “Open Distributed Processing Reference Model Part 1: Overview,” ITU-T Recommendation, May 1995.
[Rosettacode 2011]	Rosettacode.org, “Dijkstra's algorithm”, http://rosettacode.org/wiki/Dijkstra's_algorithm, last accessed May 2011
[Roj 2012]	Rój MK, "An Introduction to Parlay/OSA APIs", Warsaw University of Technology, meag.tele.pw.edu.pl/pubs/pdf/thesis-main.pdf, last accessed: December 2012
[Samuel et al 2007]	Samuel LG, Marx G, Finizola L, "The Meaning and Impact of Convergence", Proceedings of 11th International Conference on Intelligence in Service Delivery Networks, ICIN, Bordeaux, France, 2007
[Sedgewick and Wayne 2012]	Sedgewick R and Wayne K, "Shortest Paths", Princeton University, USA, http://www.cs.princeton.edu/~rs/AlgsDS07/15ShortestPaths.pdf,last accessed: November 2012
[Song et al 2008]	Song J, Lee SS, Kang K, et al, "Scalable Network Architecture for Flow-Based Traffic Control", ETRI Journal, vol.30, no.2, pp.205-215, April 2008
[Stankiewicz et al 2011]	Stankiewicz R, Cholda P, Jajszczyk A, "QoX: What is it really?" IEEE Communications Magazine, Volume 49, Number 4, 148-158, 2011
[Sun JCC2H323 2001]	Sun Microsystems, "Java™ Call Control v1.0 to H.323 API Mapping", http://192.9.162.55/products/jain/JCC2H323.pdf, ast accessed: 2001
[Sun JCC2SIP 2001]	Sun Microsystems, "Java™ Call Control v1.0 to Session Initiation Protocol Mapping", http://192.9.162.55/products/jain/JCC2SIP.pdf, last accessed: 2001

[TINA BM 1997]	TINA: "TINA Business Model and Reference Points", Version 4, May 1997, www.tinac.org
[TINA CA 1996]	TINA: "Computational Modelling Concepts", Version 3.2, May 1996
[TINA MA 1994]	Management Architecture, Document No. TB_GN.010_2.0_94, TINA-C, December 1994
[TINA NRA 1997]	TINA: "Network Resource Architecture", Version 3, February 1997, www.tinac.org
[TINA NRIM 1997]	TINA: "Network Resource Information Model Specification", Version 3, February 1997, www.tinac.org
[TINA Overall 1997]	TINA: "Overall Concepts and Principles of TINA", Version 3, February 1997, www.tinac.org
[TINA SA 1997]	TINA: "Service Architecture", Version 3, February 1997, www.tinac.org
[UCB et al 2012]	UCB/LBNL/VINT Network Simulator, ns, URL: http://www-mash. cs.berkeley.edu/ns, last accessed July 2012
[Van Dooren 2012]	Van Dooren P, "Graph Theory and Applications", Université catholique de Louvain, Louvain-la-Neuve, Belgium, http://perso.uclouvain.be/paul.vandooren/DublinCourse.pdf, last accessed: August 2012
[Wang et al 2011]	Wang N , Ho K, Pavlou G, "AMPLE: an adaptive traffic engineering system based on virtual routing topologies", IEEE Communications Magazine, Volume 49, Number 3, 185-191, 2011
[Wojick 2013]	Wojcik R, "Next Generation Networks architecture by ITU-T", http://www.kt.agh.edu.pl/~pacyna/lectures/advances_in_mobility_and_security/slides/ngn.pdf, last accessed: Jan 2013
[Yavatkar et al 2000]	Yavatkar R, Pendarakis D, and Guerin R., "A Framework for Policy-based Admission Control", Internet Engineering Task Force, RFC2753, 2000.
[Yen 1970]	Yen JY, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks". Quarterly of Applied Mathematics, Volume 27, Number 4, January 1970
[Yun and Perros 2010]	Yun C and Perros H, "QoS Control for NGN: A Survey of Techniques", Journal of Network and Systems Management, Volume 18 Issue 4, pp 447-461, December 2010
[Zhang and Guan 2012]	Zhang C and Guan Y, "Design and Implementation of Value-Added Services Based on Parlay/OSA API", Recent Progress in Data Engineering and Internet Technology, Springer, Volume 157, pp 507-514, 2012

Appendix A: Papers

The following papers resulted from this research:

Moodley, P.V., and Hanrahan, H.E., "A Network Application Programming Interface for OSA/Parlay", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2005 Proceedings, September 2005

Moodley, P.V., and Hanrahan, H.E., "Design of a Parlay Network Application Programming Interface and Associated Architecture", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2006 Proceedings, September 2006.

Moodley, P.V., and Hanrahan, H.E., "Investigation into Performance Metrics for Connection Admission Control in an MPLS Simulated Network", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2007 Proceedings, September 2007

Moodley, P.V., and Hanrahan, H.E., "Evaluation of the IMS-based MSF architecture against network architectural requirements", Southern African Telecommunications Networks and Applications Conference (SATNAC) 2008 Proceedings, September 2008.

Moodley, P.V., and van Olst, R., "RM-ODP design of the OSA/Parlay Network Interface and Associated Architecture", IEEE AFRICON 2011, 13-15 September 2011, Livingstone, Zambia. © 2011 IEEE. ISBN: ISBN 978-1-61284-991-1

The following paper has a bearing on this research:

Moodley, P., and Kennedy, I.G., “A position paper: Modelling self-similar traffic”, ICT 2005 - 12th International Conference on Telecommunications 3-6 May 2005, Cape Town. © 2005 IEEE. ISBN: 0-9584901-3-9 Reprinted in Trans SAIEE.

Appendix B: Simple Use Cases

The primary objective of the OSA/Parlay Network Architecture is to facilitate the call and service control requests from the OSA/Parlay Gateway and setup end-to-end connections within the core transport network. The OSA/Parlay Architecture must ensure that new calls entering the transport network are guaranteed the requested level of QoS for the entire duration of the call; while the QoS levels of exiting calls within the network are not infringed upon notwithstanding the dynamic nature of calls being admitted and terminated within the transport network (Moodley and Hanrahan 2006). The OSA/Parlay Network Architecture specifies concepts and principles for connections involving the packet and switch circuit technologies as well as incorporate special resources such as media gateways and IVRs. The following use cases are considered for illustrative purposes to demonstrate these architectural concepts and princples.

- **Periodic Network State Update.** The OSA/Parlay Network Architecture includes a mechanism for periodically updating the state of the network. This state update informs both the network topology as well as the performance metrics of the nodes and links within the network. The route object initiates the update on expiry of a timer, all computational objects are within the same administrative domain.
- **Connection Performer Initiated Network State Update.** When a connection performer detects an abnormal condition, it initiates a network state update for its LNC administered subnetwork.
- **OSA/Parlay Application Initiated Call to the PSTN.** The OSA/Parlay 3rd party application initiated call to the "legacy" PSTN federated network.
- **PSTN Initiated Call to the OSA/Parlay Network** illustrates a call initiated from a federated PSTN network to the OSA/Parlay Network Architecture.
- **OSA/Parlay Application Initiated Call to a SIP Network.** The OSA/Parlay 3rd party application initiated call to the SIP federated network.
- **SIP Initiated Call to the OSA/Parlay Network** illustrates the case of a federated call initiated from the SIP network and terminated on the OSA/Parlay Network Architecture.
- **IVR Call to 1-Party:** The OSA/Parlay Network Architecture integrates the IVR special resource, in this case a call initiated from the 3rd party application to the IVR and a single called party in the OSA/Parlay Network Architecture.

9.1.1 Periodic Network State Update

The NA includes a mechanism for periodically updating the state of the network. This state update informs both the network topology as well as the performance metrics of the nodes and links within the network. The control messages used for the network state update are depicted in Figure 9-2Figure 9-2. The route object initiates the update on expiry of a timer, all computational objects are within the same administrative domain, and the objects conform to the computational architecture defined in Figure 9-1.

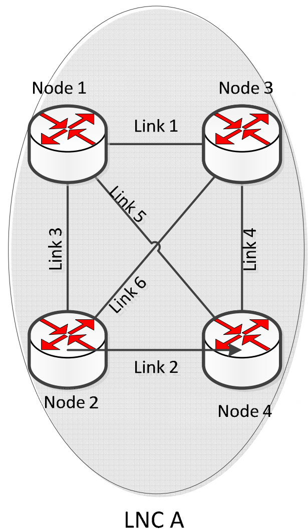

Figure 9-1: Topology of the LNC network and CP's

The update sequence is described as follows:

1-2 The periodic update is triggered by a timer which is set to a defined period on the route object. The triggering of the timer initiates and updateReq () message to each LNC within the administrative domain. This illustration is done for the message sequence for a single LNC and its constituents.

3-4 The LNC, on the receipt of the updateReq () message, requests its constituents which are CPs to update their status as well as performance. Thus the updateReq() is sent to CP1. The CP1 logs the status of the node as well as creates probe traffic and inserts the probe traffic through each link. The probe traffic includes a timestamp. Upon receipt of the probe traffic at the destinations (each adjacent CP), the destination nodes (CPs) analyse the performance of the link by calculating the latency, jitter and packet loss metrics.

This information is returned to CP1 where all the performance information of the node and links are packaged and returned to the associated LNC Object.

5-10 Similarly, each CP (CP2, CP3, CP4) in the LNC administrative domain is polled for performance information on the node and links respectively. Each CP will capture the performance information of the respective node. The CP will create probe traffic and test the performance of each link with respect to its node in terms of latency, jitter and packet loss. These performance details are forwarded to the LNC Object.

11 The LNC consolidates all the performance information from its constituents and forwards this information to the route object. The route object will poll each LNC in the network in sequence until the entire network completed (not shown inFigure 9-2). Therefore the route object will have the network state information on the performance of each node and link within the network. Should particular node(s) or link(s) be non-operational, the topology of the network will be updated with these changes.

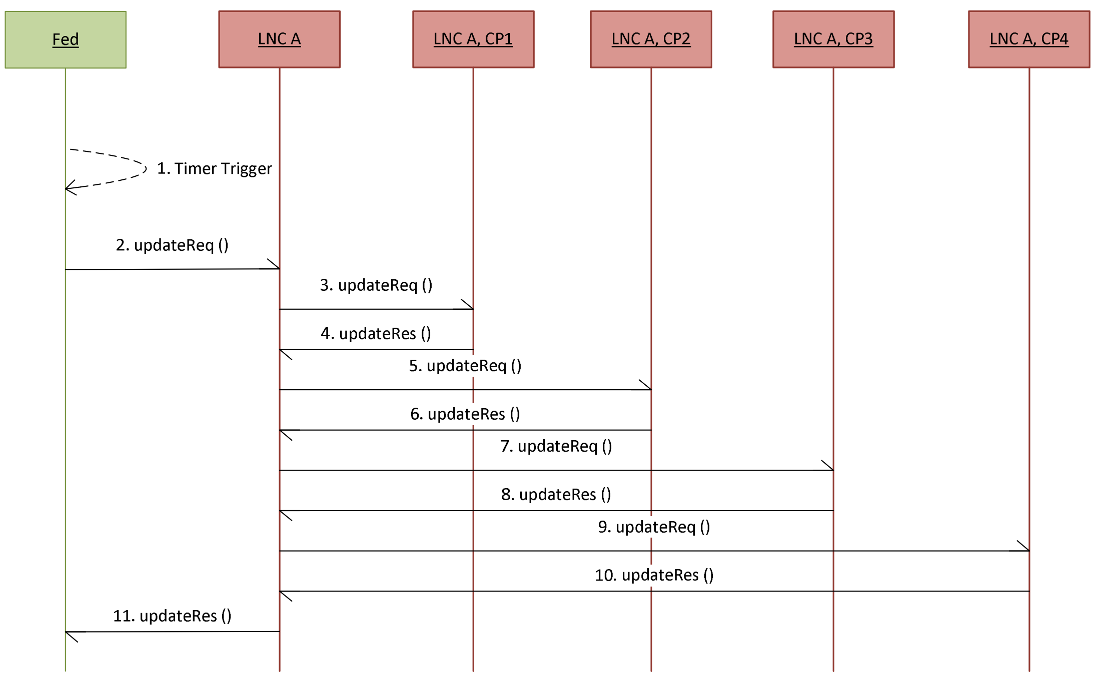

Figure 9-2: Periodic Network State Update MSC

9.1.2 CP Initiated Network State Update

The NA includes a mechanism for which a node (or CP) when detecting an abnormal condition can initiate a network state update for its LNC administered subnetwork. This update informs both the network topology as well as the performance metrics of the nodes and links within the network. The control messages used for the CP initiated network state update are depicted in Figure 9-3. The CP object initiates the update on detection of an abnormal condition, all computational objects are within the same administrative domain, and the objects conform to the computational architecture defined in Figure 9-1.

The update sequence is described as follows:

1 The affected CP detects an abnormal node or link condition. This triggers a report of the condition to the LNC using the networkStateChange () message.

2-3 The LNC, on the receipt of the networkStateChange () message, requests its constituents which are CPs, to update their status as well as performance. Thus the updateReq() is sent to CP1. The CP1 logs the status of the node as well as creates probe traffic and inserts the probe traffic through each link. The probe traffic includes a timestamp. Upon receipt of the probe traffic at the destinations (each adjacent CP), the end nodes (CPs) analyse the performance of the link by calculating the latency, jitter and packet loss metrics.

This information is returned to CP1 where all the performance information of the node and links are packaged and returned to the associated LNC Object.

4-9 Similarly, each CP (CP2, CP3, CP4) in the LNC administrative domain is polled for performance information on the node and links respectively. Each CP will capture the performance information of the respective node. The CP will create probe traffic and test the performance of each link with respect to its node in terms of latency, jitter and packet loss. These performance details are forwarded to the LNC Object.

10 The LNC consolidates all the performance information from its constituents and forwards this information to the route object.

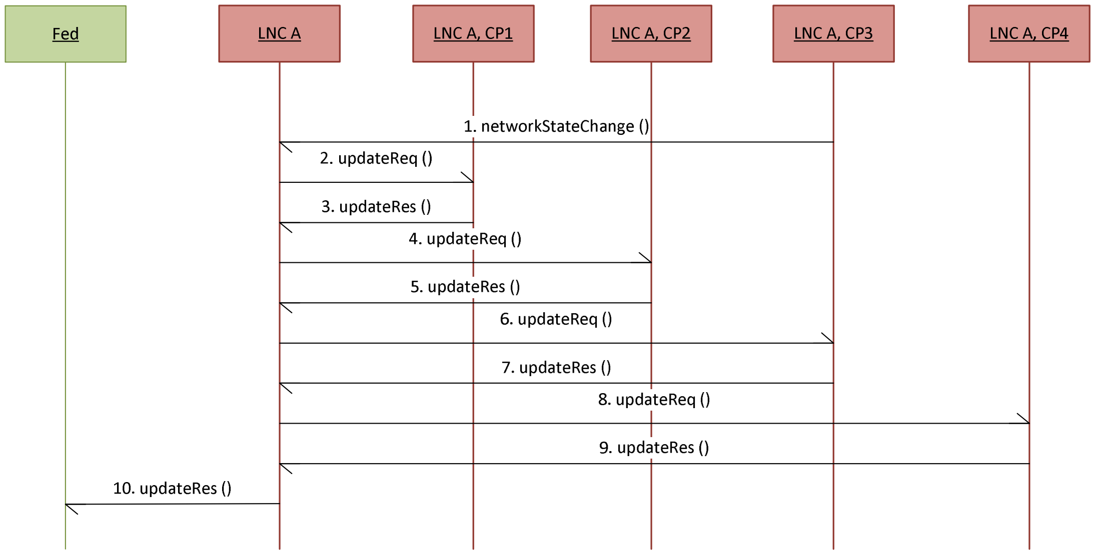

Figure 9-3: CP Initiated Network State Update MSC

9.1.3 Application Initiated Call to the PSTN

The NA makes provision for an application initiation call to a PSTN federated network. The control messages used for setting up the federated party call between two different administration domains is depicted in Figure 9-4 and Figure 9-5. The objects conform to the computational architecture defined in Figure 5-2.

The call setup sequence is described as follows:

1 The OSA/Parlay Gateway Call Object initiates the call setup by sending the createCall () method to the CSMF. This message indicates that the OSA/Parlay Gateway is preparing for an imminent call.

2 The CSMF which is a factory object used to create CSM Objects on demand, receives the createCall() method and creates the CSM object dedicated to this imminent call using the new () method. The CSM Object is created and the object unique identifier is returned to the CSMF and the OSA/Parlay Call Object.

3 The routeReq () method from the OSA/Parlay Call Object indicates that the Application requires the call to be setup within the network. The routeReq () method

is sent to the CSM Object which is dedicated to this particular call or connection. The CSM records the Calling Parties details and the connection details (bandwidth and QoS requirements). It uses these and other details which are sent in the routeReq () method to construct the LCG.

4-6 The CSM uses the setupConnection () method to create the CC. Since the CC does not exist, the CSM applies the method to the CCF which in turn creates a new CC Object for the call or connection using the new () method. The CC Object is created and the object unique identifier is returned to the CCF and the CSM Objects.

7-8 In this case the CC does not recognise the destination address within the OSA/Parlay Administrative domain and requests the Fed Object to determine whether the destination party is in a federated network. The result is returned to the CC Object.

9-10 The CC requests the Route object to select the best route using the parties location in the network and the dynamic network state information using the routeSel() method. The route object utilises its knowledge of the state of the network (see section 4.6) and applies the Dijkstra Algorithm to determine the best route together with performance details of the proposed route from the source to the Media Gateway in the transport network. The availability of the Media Gateway in terms of capacity for additional calls is taken into account. This route together with performance details are returned to the CC using the routeList () method.

Should a route not exist between the source node and the Media Gateway, or if the Media Gateway has reached its capacity or is unavailable, then an empty list will be returned to the CC (not shown). Details of the unavailability of the network resources will be indicated in the method.

11-12 The CC thereafter requests the CAC to determine whether the route meets the QoS requirements of the call request. The CAC determines whether the requested service parameters of latency, jitter and packet loss are met or exceed by the proposed route. The case of a admit decision is depicted in the message sequence chart wherein the route meets the requested criteria is selected. This route selection and admission decision is communicated using the callAdmitted () method.

At this stage in the call setup sequence, the QoS is guaranteed for the call or connection request within the network.

Should no route be found suitable, the call request is rejected with a callRejected () method (not shown). Details of the best offered network performance are returned allowing for Application related QoS negotiation.

13 The CC uses the details of the selected route to begin setting up the path in the network through the identified LNC and CP Objects. In this case the path traverses LNC A and LNC B. The CC requests that the path be setup in LNC A the path details and request are contained in the setupConnection () method.

14-15 Upon receipt of the setupConnection () request from the CC, LNC A proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and CP 3. The confirmations of the connections being setup are returned to LNC A and CC respectively.

16-18 Since the call destination is on a federated PSTN network, the connection must be routed to the LNC B subnetwork. Upon receipt of the setupConnection () request from the CC, LNC B proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and MG.

19 LNC B recognises that the connection is to the Media Gateway and thus forwards the setupMGConnection() method used for control signalling to the Media Gateway.

20-23 The PSTN uses the ISUP protocol. Thus the ISUP IAM message is sent to the Gateway Exchange. The messages to the terminal in the PSTN are not shown. Once the PSTN terminal receives the IAM message it returns the ISUP ACM message. This ACM message is returned from the Gateway Exchange to the MG. In turn the MG acknowledges the ISUP ACM message with an ISUP ACK message.

The confirmations of the connections being setup are returned to LNC B and CC respectively in message 23. At this point in the call setup the transport network path has been setup for the connection between Party A and the Media Gateway with guaranteed QoS. The messages to the terminals are not depicted as this is outside the scope of this work. The messages on the originating side traverse from CSM to the TCSM.

It is assumed that the terminal signalling has taken place and the call is in progress, see message 24.

25 The party in the NA decides to terminate the call session, the terminal signalling is received by the Application. Thus the OSA/Parlay Call Object sends the release () method to the CSM, indicating the request to terminate the call or connection session. The CSM modifies the LCG.

26 The CSM passes the release () method onto the CC to implement the physical disconnection of the connection in the transport network. The CC makes use of the selected route information to determine the LNCs to be contacted.

27-29 The CC passes the release () method onto the LNC A being the source subnetwork and the release () method is sent to the specific CP's (CP 1 and CP 3) to release the connections from the nodal perspective. The confirmation that the release has been affected is sent to LNC A and the CC respectively.

30-32 Similarly, the CC passes the release () method onto the LNC B being the subnetwork with the MG and the release () method is sent to the specific CP and MG.

33 LNC B recognises that the connection is to the Media Gateway and thus forwards the releaseMGConnection() method used for control signalling to the Media Gateway

34-36 The MG informs the PSTN Gateway Exchange of the release using the ISUP REL message. The messages to the terminal in the PSTN are not shown. This ISUP REL message traverses the PSTN network to the terminal, on receipt at the terminal the ISUP RLC message is returned. The ISUP REL message is returned from the Gateway Exchange to the MG.

At this point all the connections in the transport network relating to the call request have been successfully removed. There is no longer a transport network connection.

The confirmations are sent to the LNC B. The LNC B in turn sends the confirmatory message to the CC which in turn sends the confirmation to the CSM before the CC destroys itself. Similarly upon receipt at the CSM the confirmatory message is sent to the OSA/Parlay Call Object in message 36 and thereafter the CSM destroys itself.

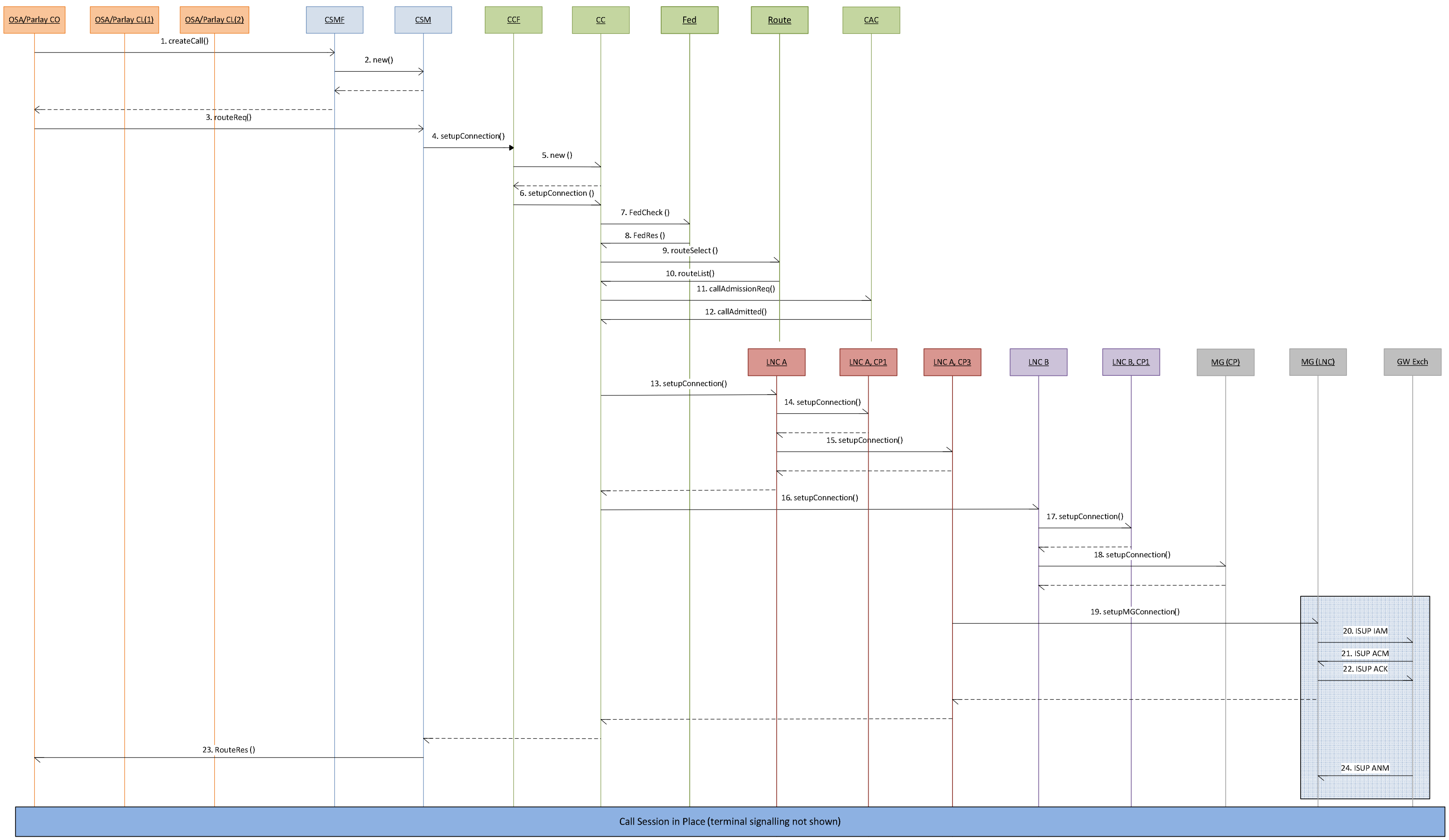

Figure 9-4: Application Initiated call to the PSTN

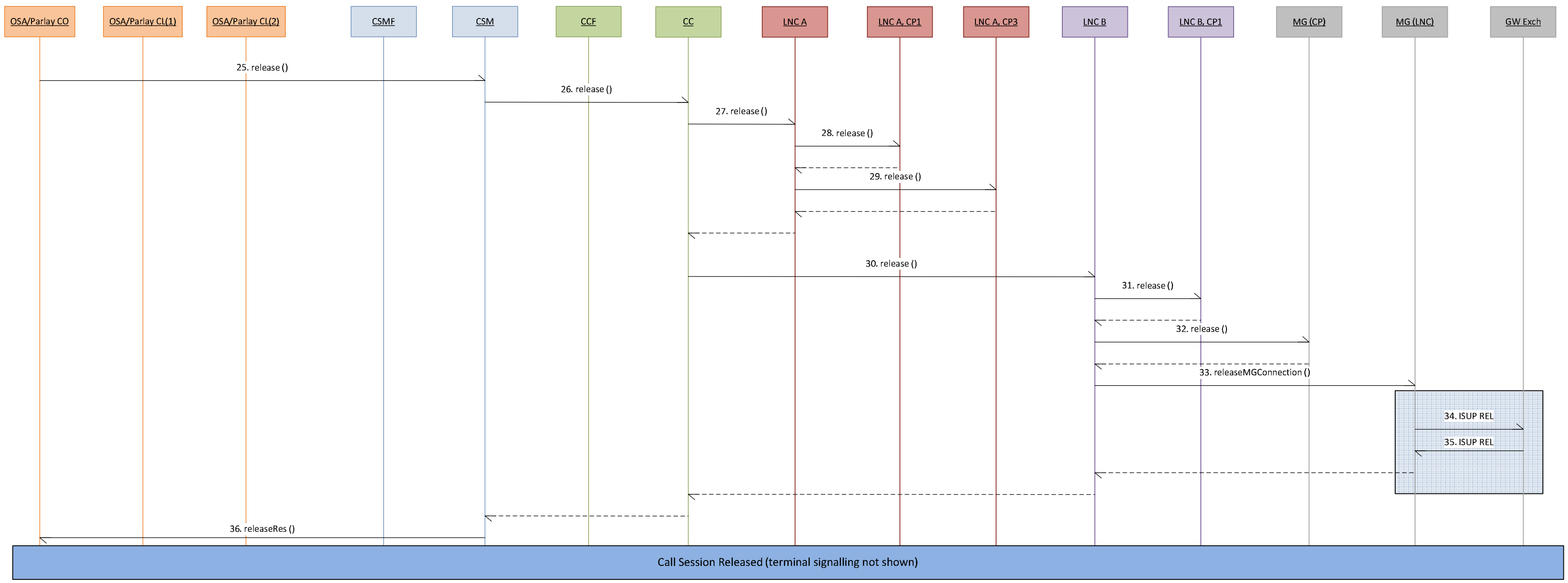

Figure 9-5: Application Initiated call to the PSTN continued

9.1.4 PSTN Initiated Call

The NA allows for federation to other types of networks, in this case a federated call in initiated from the PSTN and terminated on the NA. The control messages used for setting up the PSTN to NA call between two different administration domains and is depicted in Figure 9-6 and Figure 9-7. The objects conform to the computational architecture defined in Figure 5-2.

The call setup sequence is described as follows:

1 The Media Gateway initiates the call setup by sending the ISUP IAM message to the MGCC Object. This message contains the details of the destination call party.

2 On receipt of the ISUP IAM message, the MGCC acknowledges receipt of the call request and returns the ISUP ACM.

3-4 The MGCC Object initiates the call within the OSA/Parlay Network Architecture by invoking the createCall () method on the Fed Object. The Fed object being the component responsible for all federated calls to and from the OSA/Parlay Network Architecture. In turn the Fed Object invokes the setupConnection () on the CCF used to create the CC Object dedicated to the call.

4-6 The CCF uses the setupConnection () method to create the CC. Since the CC does not exist, the new CC Object is created for the call or connection using the new () method. The CC Object is created and the object unique identifier is returned to the CCF Object.

7-8 In this case the CC does not recognise the destination address within the OSA/Parlay Administrative domain and requests the Fed Object to determine whether the destination party is in a federated network. The result is returned to the CC Object.

9-10 The CC requests the Route object to select the best route using the parties location in the network and the dynamic network state information using the routeSel() method. The route object utilises its knowledge of the state of the network (see section 4.6) and applies the Dijkstra Algorithm to determine the best route together with performance details of the proposed route from the source to the Media Gateway in the transport network. The availability of the Media Gateway in terms of capacity for additional calls is taken into account. This route together with performance details are returned to the CC using the routeList () method.

Should a route not exist between the source node and the Media Gateway, or if the Media Gateway has reached its capacity or is unavailable, then an empty list will be returned to the CC (not shown). Details of the unavailability of the network resources will be indicated in the method.

11-12 The CC thereafter requests the CAC to determine whether the route meets the QoS requirements of the call or connection request. The CAC determines whether the requested service parameters of latency, jitter and packet loss are met or exceed by the proposed route. The case of a admit decision is depicted in the message sequence chart wherein the route meets the requested criteria is selected. This route selection and admission decision is communicated using the callAdmitted () method.

At this stage in the call setup sequence, the QoS is guaranteed for the call or connection request within the network.

Should no route be found suitable, the call request is rejected with a callRejected () method (not shown). Details of the best offered network performance are returned allowing for Application related QoS negotiation.

13 The CC uses the details of the selected route to begin setting up the path in the network through the identified LNC and CP Objects. In this case the path traverses LNC A and LNC B. The CC requests that the path be setup in LNC A the path details and request are contained in the setupConnection () method.

14-15 Upon receipt of the setupConnection () request from the CC, LNC A proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and CP 3. The confirmations of the connections being setup are returned to LNC A and CC respectively.

16-18 Since the call destination is on a federated PSTN network, the connection must be routed to the LNC B subnetwork. Upon receipt of the setupConnection () request from the CC, LNC B proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and MG.

19 LNC B recognises that the connection is to the Media Gateway and thus forwards the setupMGConnection() method used for control signalling to the Media Gateway.

20 The MG sends the ISUP ACK message indicating that the network resources within the OSA/Parlay Network Architecture have been setup and awaiting the call. At this point in the call setup the transport network path has been setup for the connection between Party A and the Media Gateway with guaranteed QoS. The messages to the terminals are not depicted as this is outside the scope of this work. The messages on the originating side traverse from CSM to the TCSM.

It is assumed that the terminal signalling has taken place and the call is in progress, see message 21.

22 The party in the federated network terminates the call session by sending the ISUP REL message to the MGCC.

23-24 The MGCC translates the ISUP REL message into the release () method invoked on the Fed object to implement the physical disconnection of the connection in the transport network. The CC makes use of the selected route information to determine the LNCs to be contacted to release the connection.

25-27 The CC passes the release () method onto the LNC A being the source subnetwork and the release () method is sent to the specific CP's (CP 1 and CP 3) to release the connections from the nodal perspective. The confirmation that the release has been affected is sent to LNC A and the CC respectively.

28-30 Similarly, the CC passes the release () method onto the LNC B being the subnetwork with the MG and the release () method is sent to the specific CP and MG.

31 LNC B recognises that the connection is to the Media Gateway and thus forwards

the releaseMGConnection() method used for control signalling to the Media Gateway

32 The MG informs the PSTN Gateway Exchange of the release using the ISUP RLC message. The messages confirm that the network resources in the OSA/Parlay Network have been released.

At this point all the connections in the transport network relating to the call request have been successfully removed. There is no longer a transport network connection. The confirmations are sent to the LNC B. The LNC B in turn sends the confirmatory message to the CC object.

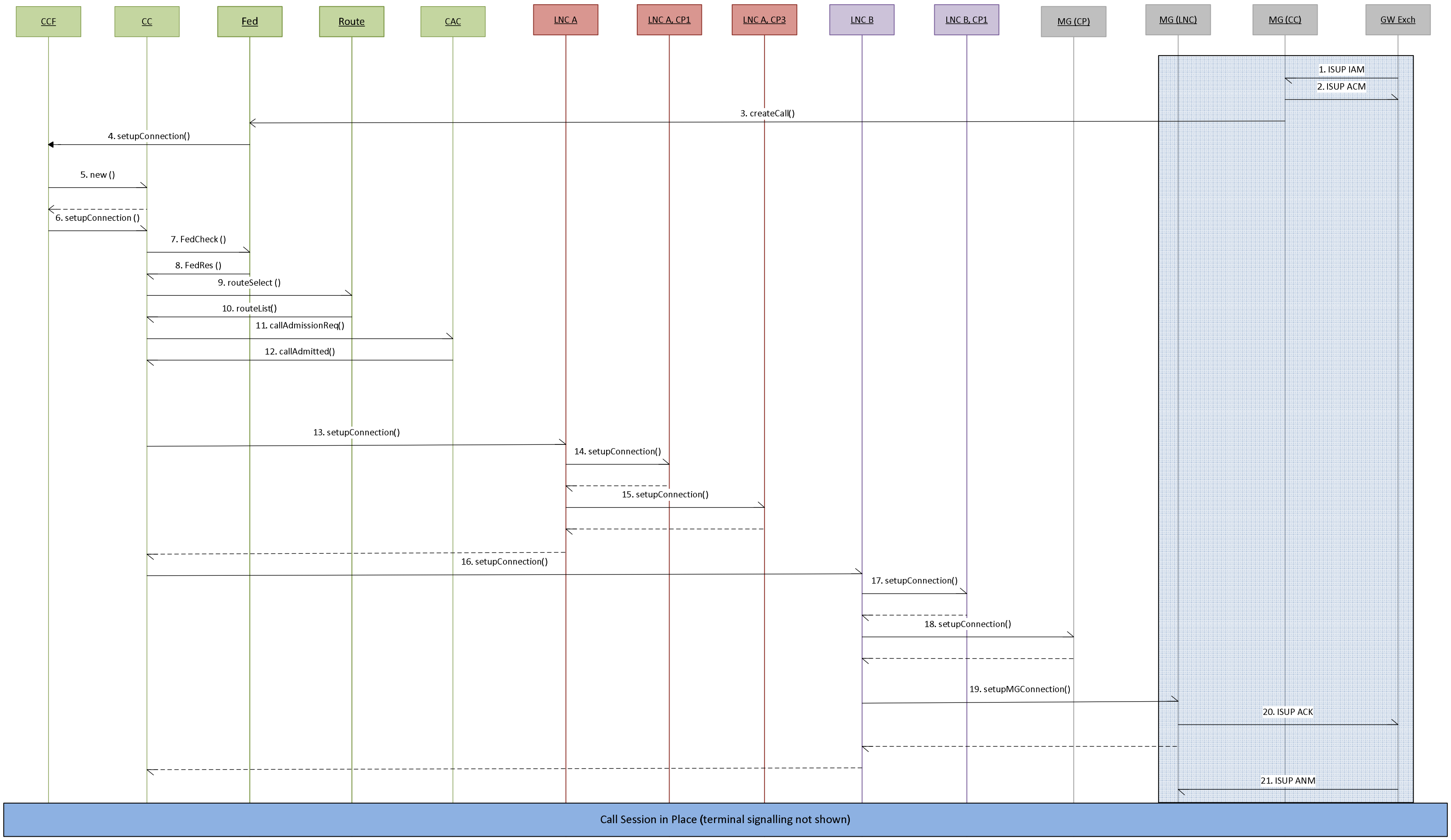

Figure 9-6: PSTN Initiated Call

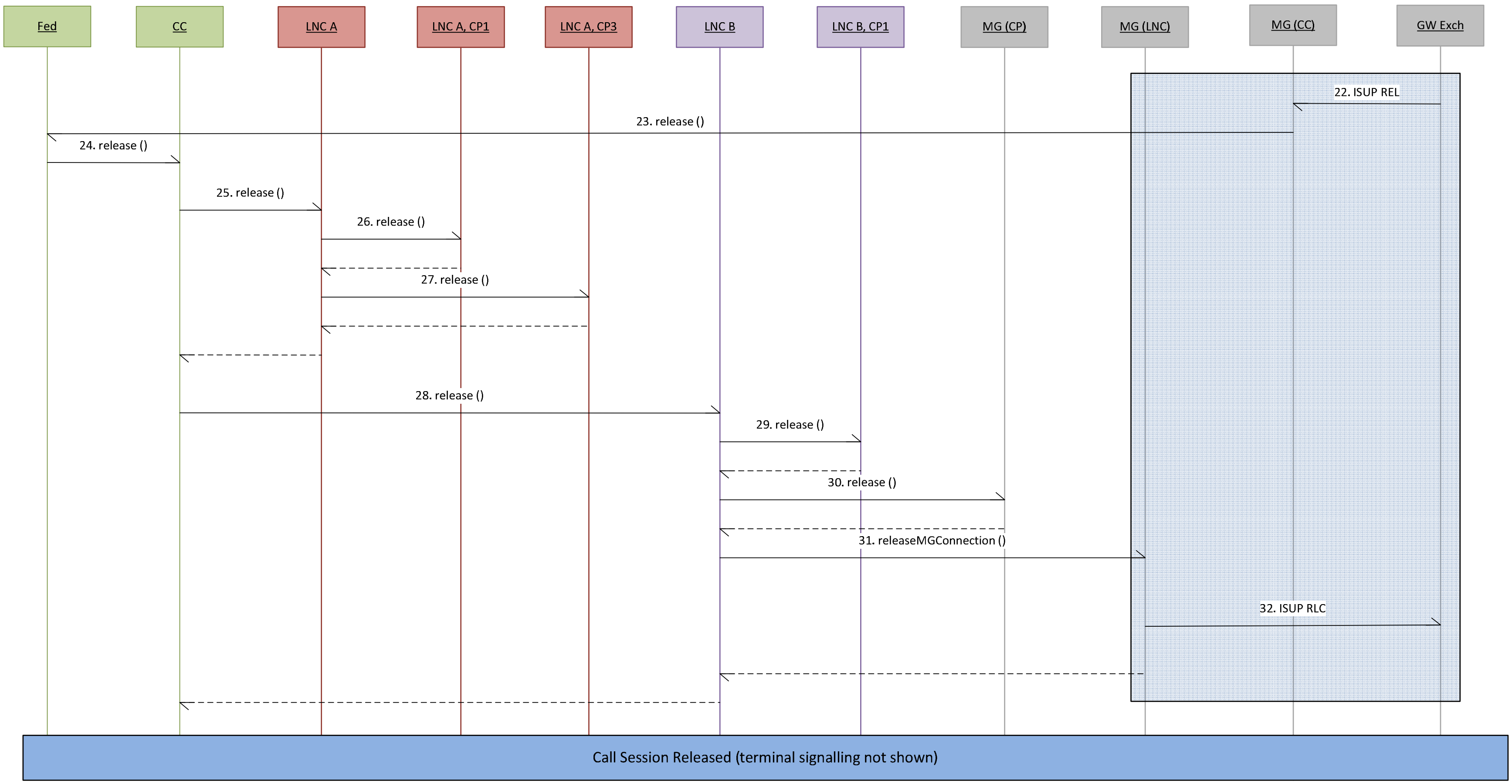

Figure 9-7: PSTN Initiated Call continued

9.1.5 Application Initiated Call to a SIP Network

The NA makes provision for an application initiation call to a SIP network. The control messages used for setting up the two party call between two different administration domains and is depicted in Figure 9-8 and Figure 9-9. The objects conform to the computational architecture defined in Figure 5-2.

The call setup sequence is described as follows:

1 The OSA/Parlay Gateway Call Object initiates the call setup by sending the createCall () method to the CSMF. This message indicates that the OSA/Parlay Gateway is preparing for an imminent call.

2 The CSMF which is a factory object used to create CSM Objects on demand, receives the createCall() method and creates the CSM object dedicated to this imminent call using the new () method. The CSM Object is created and the object unique identifier is returned to the CSMF and the OSA/Parlay Call Object.

3 The routeReq () method from the OSA/Parlay Call Object indicates that the Application requires the call to be setup within the network. The routeReq () method is sent to the CSM Object which is dedicated to this particular call or connection. The CSM records the Calling Parties details and the connection details (bandwidth and QoS requirements). It uses these and other details which are sent in the routeReq () method to construct the LCG.

4-6 The CSM uses the setupConnection () method to create the CC. Since the CC does not exist, the CSM applies the method to the CCF which in turn creates a new CC Object for the call or connection using the new () method. The CC Object is created and the object unique identifier is returned to the CCF and the CSM Objects.

7-8 In this case the CC does not recognise the destination address within the OSA/Parlay Administrative domain and requests the Fed Object to determine whether the destination party is in a federated network. The result is returned to the CC Object.

9-10 The CC requests the Route object to select the best route using the parties location in the network and the dynamic network state information using the routeSel() method. The route object utilises its knowledge of the state of the network (see section 4.6) and applies the Dijkstra Algorithm to determine the best route together with performance details of the proposed route from the source to the Media Gateway in the transport network. The availability of the Media Gateway in terms of capacity for additional calls is taken into account. This route together with performance details are returned to the CC using the routeList () method.

Should a route not exist between the source node and the Media Gateway, or if the Media Gateway has reached its capacity or is unavailable, then an empty list will be returned to the CC (not shown). Details of the unavailability of the network resources will be indicated in the method.

11-12 The CC thereafter requests the CAC to determine whether the route meets the QoS requirements of the call or connection request. The CAC determines whether the requested service parameters of latency, jitter and packet loss are met or exceed by the proposed route. The case of a admit decision is depicted in the message sequence chart wherein the route meets the requested criteria is selected. This route selection and admission decision is communicated using the callAdmitted () method.

At this stage in the call setup sequence, the QoS is guaranteed for the call or connection request within the network.

Should no route be found suitable, the call request is rejected with a callRejected () method (not shown). Details of the best offered network performance are returned allowing for Application related QoS negotiation.

13 The CC uses the details of the selected route to begin setting up the path in the network through the identified LNC and CP Objects. In this case the path traverses LNC A and LNC B. The CC requests that the path be setup in LNC A the path details and request are contained in the setupConnection () method.

14-15 Upon receipt of the setupConnection () request from the CC, LNC A proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and CP 3. The confirmations of the connections being setup are returned to LNC A and CC respectively.

16-18 Since the call destination is on a federated SIP network, the connection must be routed to the LNC B subnetwork. Upon receipt of the setupConnection () request from the CC, LNC B proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and MG.

19 LNC B recognises that the connection is to the Media Gateway and thus forwards the setupMGConnection() method used for control signalling to the Media Gateway.

20-23 As the connection is to a SIP network, the SIP INVITE message is sent to the SIP Proxy. The messages to the terminal in the SIP network are not shown. Once the SIP terminal receives the INVITE message it returns the SIP 100 TRYING message. The SIP 183 SESSION PROGRESS message is returned to the SIP Proxy.

The confirmations of the connections being setup are returned to LNC B and CC respectively in message 23.

At this point in the call setup the transport network path has been setup for the connection between Party A and the Media Gateway with guaranteed QoS. The messages to the terminals are not depicted as this is outside the scope of this work. The messages on the originating side traverse from CSM to the TCSM.

It is assumed that the terminal signalling has taken place and the call is in progress, see message 24.

25 The party in the NA decides to terminate the call session, the terminal signalling is received by the Application. Thus the OSA/Parlay Call Object sends the release () method to the CSM, indicating the request to terminate the call or connection session. The CSM modifies the LCG.

26 The CSM passes the release () method onto the CC to implement the physical disconnection of the connection in the transport network. The CC makes use of the selected route information to determine the LNCs to be contacted.

27-29 The CC passes the release () method onto the LNC A being the source subnetwork and the release () method is sent to the specific CP's (CP 1 and CP 3) to release the connections from the nodal perspective. The confirmation that the release has been affected is sent to LNC A and the CC respectively.

30-32 Similarly, the CC passes the release () method onto the LNC B being the subnetwork with the MG and the release () method is sent to the specific CP and MG.

33 LNC B recognises that the connection is to the Media Gateway and thus forwards the releaseMGConnection() method used for control signalling to the Media Gateway

34-36 The MG informs the SIP Proxy of the release using the SIP BYE message. The messages to the terminal in the PSTN are not shown. This SIP BYE message traverses the SIP network to the terminal, on receipt at the terminal the SIP 200 OK message is returned. This message is returned from the SIP Proxy to the MG.

At this point all the connections in the transport network relating to the call request have been successfully removed. There is no longer a transport network connection. The confirmations are sent to the LNC B. The LNC B in turn sends the confirmatory message to the CC which in turn sends the confirmation to the CSM before the CC destroys itself. Similarly upon receipt at the CSM the confirmatory message is sent to the OSA/Parlay Call Object in message 36 and thereafter the CSM destroys itself.

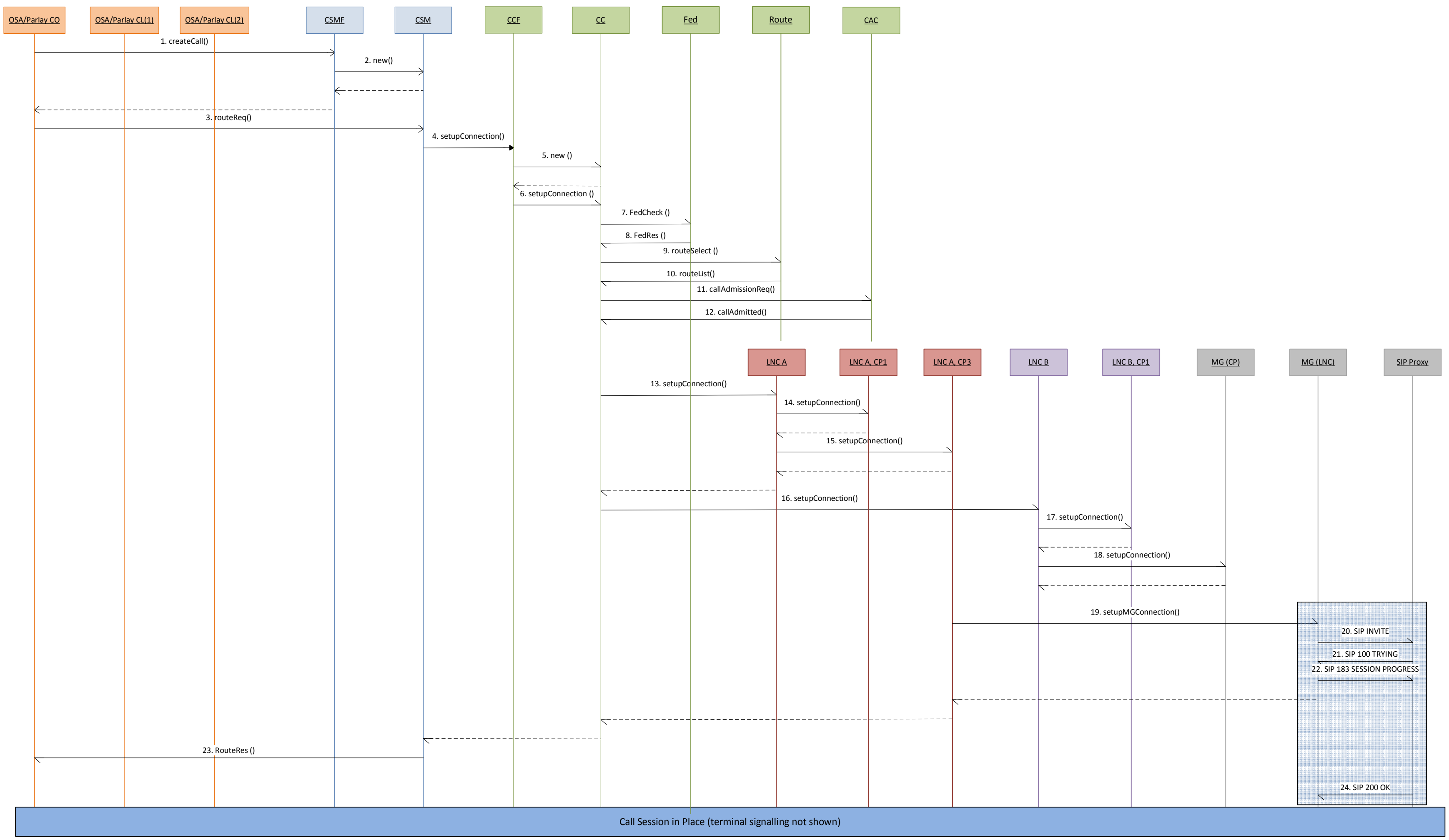

Figure 9-8: Application Initiated Call to the SIP Network

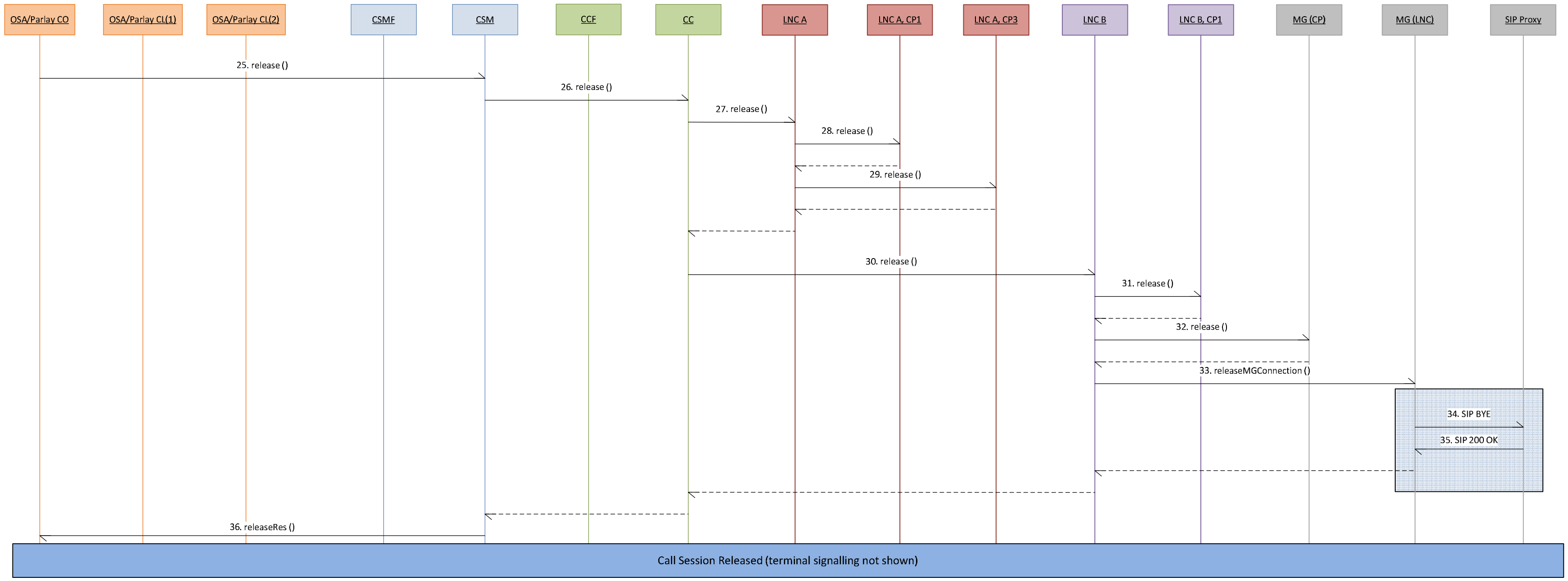

Figure 9-9: Application Initiated Call to the SIP Network continued

9.1.6 SIP Initiated Call

The NA allows for federation to other types of networks, in this case a federated call is initiated from the SIP and terminated on the NA. The control messages used for setting up the PSTN to NA call between two different administration domains and is depicted in Figure 9-10 and Figure 9-11. The objects conform to the computational architecture defined in Figure 5-2.

The call setup sequence is described as follows:

1 The Media Gateway initiates the call setup by sending the SIP INVITE message to the MGCC Object. This message contains the details of the destination call party.

2 On receipt of the SIP INVITE message, the MGCC acknowledges receipt of the call request and returns the SIP 100 TRYING message.

3-4 The MGCC Object initiates the call within the OSA/Parlay Network Architecture by invoking the createCall () method on the Fed Object. The Fed object being the component responsible for all federated calls to and from the OSA/Parlay Network Architecture. In turn the Fed Object invokes the setupConnection () on the CCF used to create the CC Object dedicated to the call.

4-6 The CCF uses the setupConnection () method to create the CC. Since the CC does not exist, the new CC Object is created for the call or connection using the new () method. The CC Object is created and the object unique identifier is returned to the CCF Object.

7-8 In this case the CC does not recognise the destination address within the OSA/Parlay Administrative domain and requests the Fed Object to determine whether the destination party is in a federated network. The result is returned to the CC Object.

9-10 The CC requests the Route object to select the best route using the parties location in the network and the dynamic network state information using the routeSel() method. The route object utilises its knowledge of the state of the network (see section 4.6) and applies the Dijkstra Algorithm to determine the best route together with performance details of the proposed route from the source to the Media Gateway in the transport network. The availability of the Media Gateway in terms of capacity for additional calls is taken into account. This route together with performance details are returned to the CC using the routeList () method.

Should a route not exist between the source node and the Media Gateway, or if the Media Gateway has reached its capacity or is unavailable, then an empty list will be returned to the CC (not shown). Details of the unavailability of the network resources will be indicated in the method.

11-12 The CC thereafter requests the CAC to determine whether the route meets the QoS requirements of the call or connection request. The CAC determines whether the requested service parameters of latency, jitter and packet loss are met or exceed by the proposed route. The case of a admit decision is depicted in the message sequence chart wherein the route meets the requested criteria is selected. This route selection and admission decision is communicated using the callAdmitted () method.

At this stage in the call setup sequence, the QoS is guaranteed for the call or connection request within the network.

Should no route be found suitable, the call request is rejected with a callRejected () method (not shown). Details of the best offered network performance are returned allowing for Application related QoS negotiation.

13 The CC uses the details of the selected route to begin setting up the path in the network through the identified LNC and CP Objects. In this case the path traverses LNC A and LNC B. The CC requests that the path be setup in LNC A the path details and request are contained in the setupConnection () method.

14-15 Upon receipt of the setupConnection () request from the CC, LNC A proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and CP 3. The confirmations of the connections being setup are returned to LNC A and CC respectively.

16-18 Since the call destination is on a Federated PSTN network, the connection must be routed to the LNC B subnetwork. Upon receipt of the setupConnection () request from the CC, LNC B proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and MG.

19 LNC B recognises that the connection is to the Media Gateway and thus forwards the setupMGConnection() method used for control signalling to the Media Gateway.

20 The MG sends the SIP 183 SESSION PROGRESS message, indicating that the network resources within the OSA/Parlay Network Architecture have been setup and awaiting the call. At this point in the call setup the transport network path has been setup for the connection between Party A and the Media Gateway with guaranteed QoS. The messages to the terminals are not depicted as this is outside the scope of this work. The messages on the originating side traverse from CSM to the TCSM.

It is assumed that the terminal signalling has taken place and the call is in progress, see message 21.

22 The party in the federated network terminates the call session by sending the SIP BYE message to the MGCC.

23-24 The MGCC translates the SIP BYE message into the release () method invoked on the Fed object to implement the physical disconnection of the connection in the transport network. The CC makes use of the selected route information to determine the LNCs to be contacted to release the connection.

25-27 The CC passes the release () method onto the LNC A being the source subnetwork and the release () method is sent to the specific CP's (CP 1 and CP 3) to release the connections from the nodal perspective. The confirmation that the release has been affected is sent to LNC A and the CC respectively.

28-30 Similarly, the CC passes the release () method onto the LNC B being the subnetwork with the MG and the release () method is sent to the specific CP and MG.

31 LNC B recognises that the connection is to the Media Gateway and thus forwards the releaseMGConnection() method used for control signalling to the Media Gateway

32 The MG informs the SIP Proxy of the release using the SIP 200 OK message. The messages confirms that the network resources in the OSA/Parlay Network have been released.

At this point all the connections in the transport network relating to the call request have been successfully removed. There is no longer a transport network connection. The confirmations are sent to the LNC B. The LNC B in turn sends the confirmatory message to the CC object.

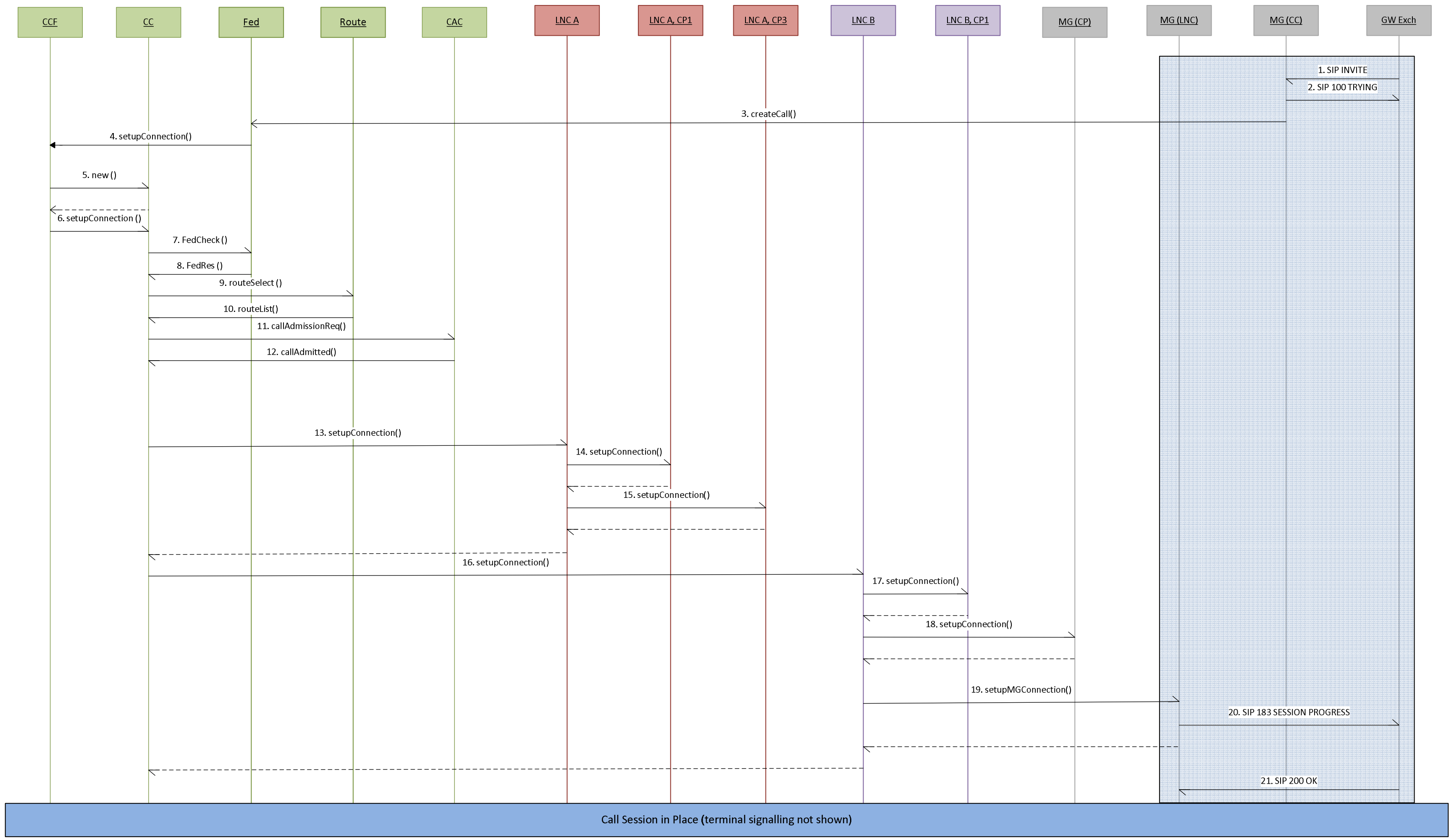

Figure 9-10: SIP Initiated Call to the OSA/Parlay Network Architecture

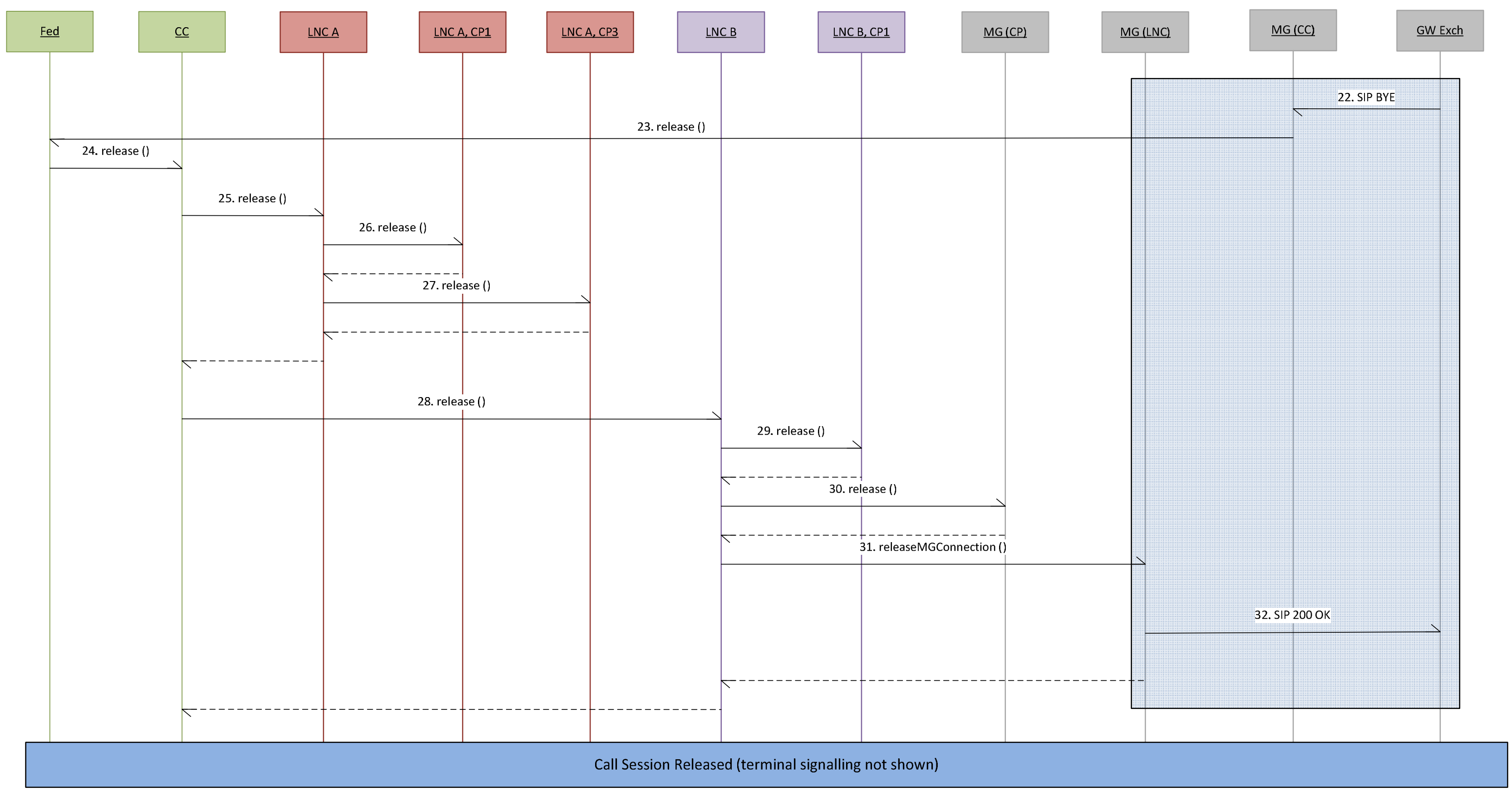

Figure 9-11: SIP Initiated Call to the OSA/Parlay Network Architecture continued

9.1.7 IVR Call to 1-Party

The NA makes provision for an application initiation call between IVR and a single called party. The control messages used for setting up the two party call between two different administration domains and is depicted in Figure 9-12 and Figure 9-13. The objects conform to the computational architecture defined in Figure 5-2.

The call setup sequence is described as follows:

1 The OSA/Parlay Gateway UI Call Object initiates the call setup by sending the createUICall () method to the CSMF. This message indicates that the OSA/Parlay Gateway is preparing for an imminent call.

2 The CSMF which is a factory object used to create CSM Objects on demand, receives the createUICall() method and creates the CSM object dedicated to this imminent call using the new () method. The CSM Object is created and the unique identifier is returned to the CSMF and the OSA/Parlay Call Object.

3 The CSMF recognises the UI call and therefore replaces the destination address with the details of the IVR unit. The CSMF initiates the call setup by applying the routeReq() method on the CCF Object as the CC for the call has not been created yet.

4-6 The CSM uses the setupConnection () method to create the CC. Since the CC does not exist, the CSM applies the method to the CCF which in turn creates a new CC Object for the call or connection.

7-8 The CC requests from the route object a short prioritised list of the best (shortest) routes in the current state of the network based on latency using the routeSel() method. The route object utilities its knowledge of the state of the network (see section 4.6) and applies adapted Dijkstra Algorithm to determine the best route from the source to the IVR in the transport network. The availability of the IVR in terms of capacity for additional calls is taken into account. This route and performance details are returned to the CC using the routeList () method.

Should a route not exist between the source node and the IVR, or if the IVR has reached its capacity or is unavailable, then an empty list will be returned to the CC (not shown).

9-10 The CC thereafter requests the CAC to determine whether the route meets the QoS requirements of the call or connection request. The CAC determines whether the requested service parameters of latency, jitter and packet loss are met or exceed by the proposed route. The case of a admit decision is depicted in the message sequence chart wherein the route meets the requested criteria is selected. This route selection and admission decision is communicated using the callAdmitted () method.

At this stage in the call setup sequence, the QoS is guaranteed for the call or connection request within the network.

Should no route be found suitable, or an empty list submitted, the call request is rejected with a callRejected () method (not shown).

11 The CC uses the details of the selected route to begin setting up the path in the network through the identified LNC and CP Objects. In this case the path traverses the LNC A and LNC B. The CC requests that the path be setup in LNC A the path details and request are contained in the setupConnection () method.

12-13 Upon receipt of the setupConnection () request from the CC, LNC A proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and CP 3. The confirmations of the connections being setup are returned to LNC A and CC respectively.

14-16 Since the call destination is the IVR unit in LNC B, the connection must be routed to the LNC B subnetwork. Upon receipt of the setupConnection () request from the CC, LNC B proceeds to setup the connections as detailed in the path information received from the CC by sending setupConnection () method requests to CP 1 and IVR(CP).

The confirmations of the connections being setup are returned to LNC B and CC respectively.

At this point in the call setup the transport network path has been setup for the connection between Party A and IVR with guaranteed QoS. The messages to the terminal are not depicted as this is outside the scope of this work. The messages on the originating side traverse from CSM to the TCSM.

It is assumed that the terminal signalling has taken place and the call is in progress

17-18 The sendInfoAndCollectReq () message is now forwarded to the CC, the CC in turn forwards the sendInfoAndCollectReq () message to the IVR (CSM) which requests the IVR unit to play a message to Party A and record input from Party A.

19-20 The results of the input obtained from Party A are contained in the sendInfoAndCollectRes () method which is forwarded from the IVR (CSM) to the CC, and in turn from the CC to the CSM; and finally forwarded to the OSA/Parlay UI Call Object for further analysis.

21 The service logic decides to terminate the call session. Thus the OSA/Parlay UI Call Object send the release () method to the CSM, indicating the request to terminate the call or connection session. The CSM modifies the LCG.

22 The CSM passes the release () method onto the CC to implement the physical disconnection of the connection in the transport network. The CC makes use of the selected route information to determine the LNCs to be contacted.

23-25 The CC passes the release () method onto the LNC A being the source subnetwork and the release () method is sent to the specific CP’s to release the connections from the nodal perspective. The confirmation that the release has been affected is sent to LNC A and the CC respectively.

26-28 Similarly, the CC passes the release () method onto the LNC B being the special resource subnetwork and the release () method is sent to the specific CP and IVR.

29-30 At this point all the connections in the transport network relating to the call request have been successfully removed. There is no longer a transport network connection. The confirmations are sent to the LNC B. The LNC B in turn sends the confirmatory message to the CC which in turn sends the confirmation to the CSM before the CC destroys itself. Similarly upon receipt at the CSM the confirmatory message is sent to the OSA/Parlay Call Object and thereafter the CSM destroys itself

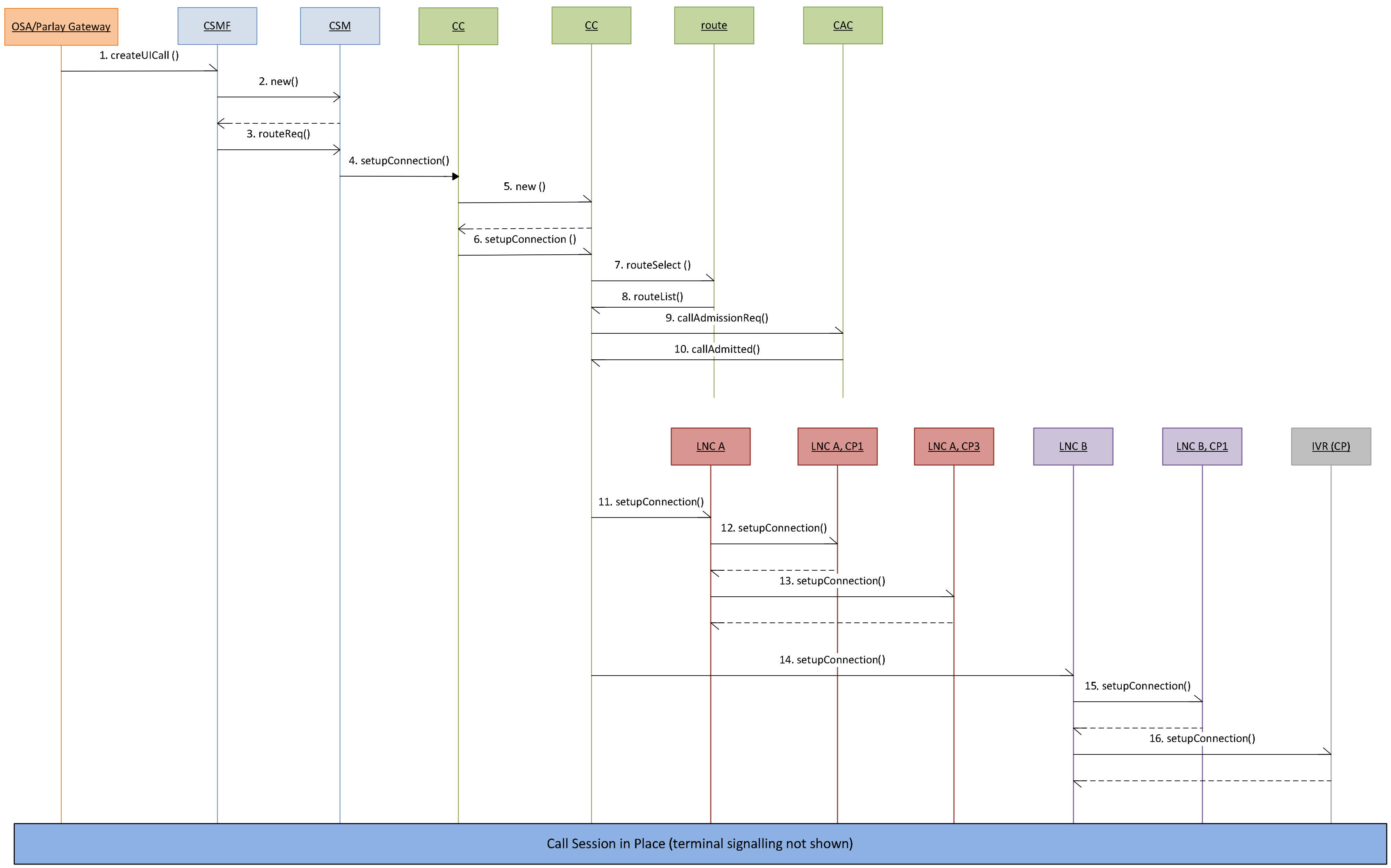

Figure 9-12: Application initiated call to the IVR

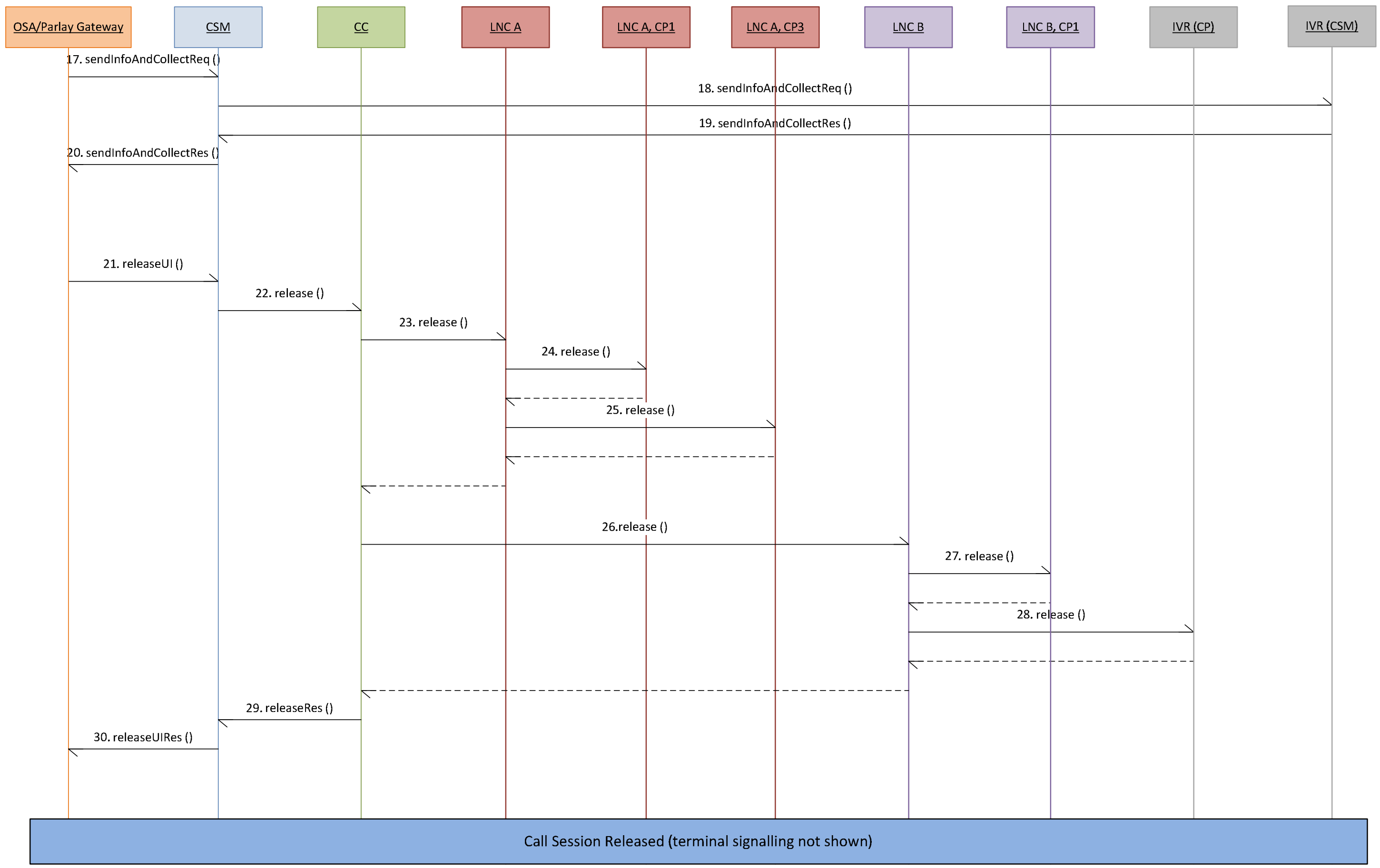

Figure 9-13: Application initiated call to the IVR continued

9.1.8 Conclusion

The OSA/Parlay Network Architecture specifies concepts and principles for connections involving the packet and switch circuit technologies as well as incorporate special resources such as media gateways and IVRs. The use cases presented illustrate the functions of the Network Interface and Network Architecture accommodating both switch circuit technologies and packet switch technologies. In addition, the proposed Architecture interfaces to common special resources being the Media Gateway and the Interactive Voice Response unit. The use cases illustrated from section 9.1.3 to section 9.1.7 were demonstrated in the proof of concept, the source code for the implementation is provided on a separate removable storage disc.