VPN Provisioning using the OSA Parlay Connection Management Interface

J Yan and H Hanrahan
Centre for Telecommunications Access and Services (CeTAS)[1]
University of the Witwatersrand
Johannesburg
South Africa
{j.yan, h.hanrahan}@ee.wits.ac.za

Abstract: A VPN service plays an important role in the telecommunication service field. This paper describes a method of implementing a VPN service by using the OSA/Parlay standard in a packet network. We first review the VPN service network structure. Second, we focus on the OSA/Parlay connectivity management API to build and manage a VPN service. Finally, we evaluate the implementing OSA/Parlay Connectivity Management API as a VPN service in the packet network.

Keyword: VPN, OSA/Parlay, DiffServ, MPLS, Connectivity Management, QoS.

I. Introduction

Today, companies rely increasingly on information exchange. Companies require telecommunications network service with good quality of service (QoS), multiple services, broadband data link access, and low tariffs. Telecommunication companies want to improve their services and reduce their cost so as to become more competitive. For these reasons, virtual private network (VPN) service was born and is now widely used.

As it is most commonly defined, a VPN allows two or more private networks to be connected over a public access network. In a sense, VPNs are similar to wide area networks (WAN) or a securely encrypted tunnel, but the key feature of VPNs is that they are able to use public networks like the Internet rather than rely on expensive, private leased lines. At the same time, VPNs have the same security and encryption features as a private network, while taking advantage of the economies of scale and remote accessibility of large public networks.

A VPN is an especially effective means of exchanging critical information for employees working remotely in branch offices, at home, or on the road. VPNs deliver information securely between vendors, suppliers and business partners, all separated from each other. Since companies no longer have to invest in the actual infrastructure themselves, they can reduce their operational costs by outsourcing network services to service providers. VPNs can also reduce costs by eliminating the need for long-distance telephone charges to obtain remote access, as client needs only to call into the service provider's nearest access point.

Several VPN service network models have been developed in recent years. VPNs can be set up in a variety of ways, and built over ATM, Frame Relay, and X.25 technologies. However, the most popular current method is to deploy IP/MPLS-based VPNs, designed for that is ease of traffic management. MPLS offers more flexibility and ease of connectivity.

Parlay is an open and technology-independent standard, allowing a wide range of market players to develop and offer advanced telecom services. Parlay supports creating the services and applications in packet network by telecommunication companies or 3rd party independent software companies. The standard supports general telecommunication services, including generic call control, conference call control, mobility management, data session control and VPN. The Connectivity Management Service Capability Feature is an API in the Parlay standard. The API, located between the enterprise operator and the provider network, allows parties to establish QoS parameters for enterprise network packets travelling through the provider network. The API provides the enterprise network operator on-line access to provision QoS measures that control the enterprise's own traffic passing through the provider network. By using APIs, operator can create virtual provisioned pipes (VPrPs) in the provider network to carry the enterprise traffic and support it with pre-specified quality of service attributes.

This paper describes and evaluates using Parlay Connection Management API for setting up virtual private networks service in a MPLS underlying network. Section II reviews the Parlay Architecture. Section III reviews the network layer structure based on DiffServ and MPLS. Section IV describes the VPN application that manages the VPN via the Parlay connection management interface.

II. Overview of OSA/Parlay

Services in Public Switched telecommunications Networks (PSTN) are IN-based value-added services controlled by the network operators. The PSTN is a closed network, that is, one in which service providers outside the network operator cannot control network resources to provide services with telecommunications features. By contrast, an open network permits a service provider other than the network operator to create and offer services which make use of resources within the network. This is achieved by accessing the network service control through an open standard, secure interface with a defined application programming interface (API).

OSA/Parlay is one of the standards supporting open networks. Parlay/OSA integrates telecom network capabilities with IT applications via a secure, measured, and billable interface. Parlay's open application programming

[1] The Centre is support by Telkom SA Limited, Siemens Telecommunications and the THRIP Programme of the Department of Trade and Industry.

interfaces (APIs) hide the detail of specific networks and environments from programmers. Programmers are free to concentrate on innovative new services with telecommunication features. The risks associated with developing new services are reduced. Applications enabled by Parlay's network-independent APIs, are increasingly adopted by network operators, application service providers, and independent software vendors.

While Parlay API creates the opportunity for new applications both within the telco and third party provider domain, the Parlay Connection Management API allows VPN control functions, normally regarded as an Operations Support System (OSS) type of function, to be implemented as a Parlay application.

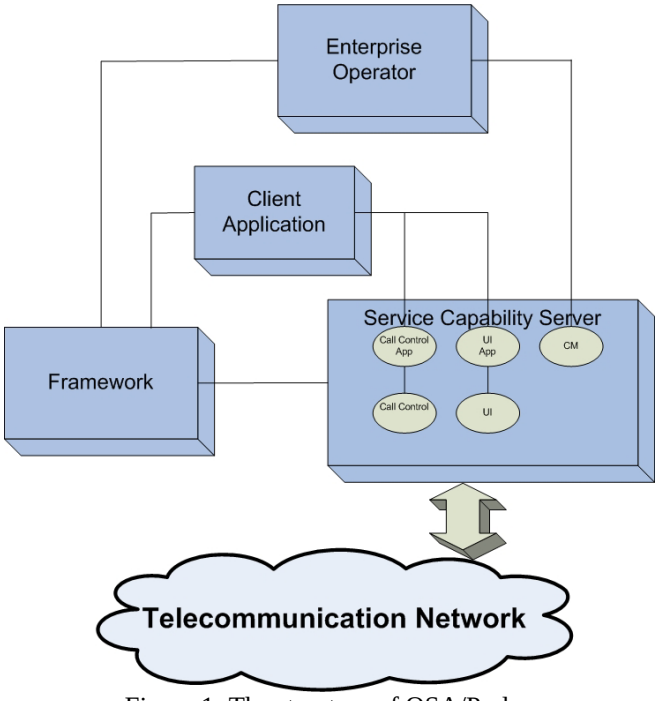

Figure 1: The structure of OSA/Parlay

The Parlay structure is shown in Figure 1. The structure has 4 main blocks. Lines between the blocks show their relationships. Each line locates an interface between two objects. All the common services are implemented inside the Service Capability Server (SCS). The SCS contains 15 kinds of individual Service Capability Features. The Enterprise Operator usually is able to create and modify services in the SCSs, and run and control services via Framework. The Client Application contains logic that uses the services available through the API. When the services are running, the SCSs give commands to the network routers or switches providing data transmission.

The telecommunication network cloud in Figure 1 provides connections to support VPN service. Section III will review the principles for creating VPNs based on DiffServ and network models.

III. VPN SERVICE NETWORK LAYER STRUCTURE

A.Differentiated Service (DiffServ)

DiffServ provides a framework that enables service providers to offer each customer a range of services that are differentiated on the basis of the performance. The features of DiffServ are:

1. Services are differentiated by performance;
2. Services are controlled on a packet by packet basis;
3. A control protocol is not defined;
4. QoS is achieved simply in the core network
5. Methods used are traffic classification and conditioning
6. Relies on IP header to contain the traffic information in the DiffServ code (DSCP)

The DiffServ field in the IP datagram is the IPv4 type of service (TOS) field. The TOS field is called DSCP in Diffserv. The original IPv4 8-bit field and the DSCP are shown in Figure 2.

The DSCP notation is xxxxxx, where the x is 1 or 0. The entire 6-bit field is used by DiffServ nodes as an index into a table to select a specific packets handing mechanism. In particular, xxxxx0 is assigned by standard action; xxxx11 is reserved for experimental or local use; xxxx01 is available for experimental or local use but maybe give to standard action in future.

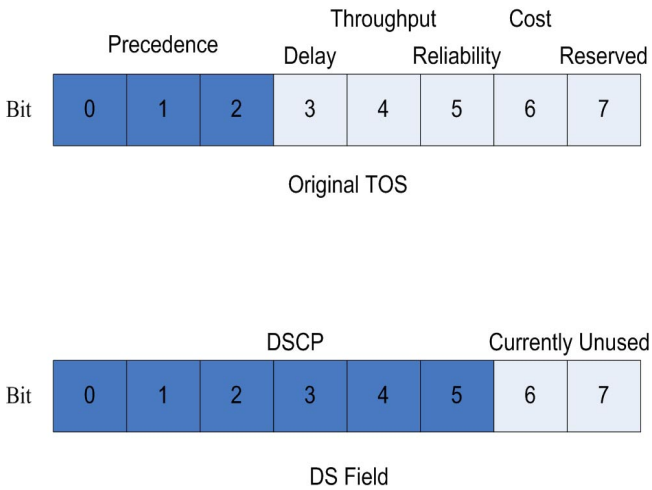

Figure 2: IPv4 TOS Field and DS Field

B. Overview of the VPN service network technologies

There are several existing VPN network solutions at the moment. Basically, they can be divided into two parts. The first type, called layer-2 VPN, is based on the data link layer in the ISO Open Systems Interconnection Reference Model (OSI-RM). The provider extends layer-2 services to the customer’s sites. A key issue of layer-2 VPN is that the provider is unaware of layer-3 specific VPN information. The customers and the providers do not exchange any routing information with each other. Forwarding decisions in the provider network are based only on layer-2 information such as MAC address, ATM VC identifier, MPLS label and port number. There are two directions in layer-2 VPN: Virtual Private Wire Service (VPWS) shown in Figure 3 and Virtual Private LAN Service (VPLS) shown in Figure 4. The major difference between the two types is that the VPWS provides a provider edge (PE) to PE link service while the VPLS provides a PE to provider core node (P node) to PE link service. The VPWS approach can be thought as deriving from traditional leased line service. The PEs are connected in a partial or full mesh. The VPLS approach emulates a local area network (LAN) environment where a site automatically gains connectivity to all the other sites attached to the same emulated LAN.

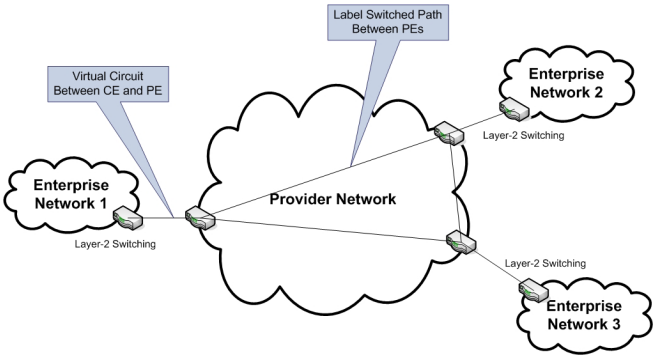

Figure 3: Virtual Private Wire Service

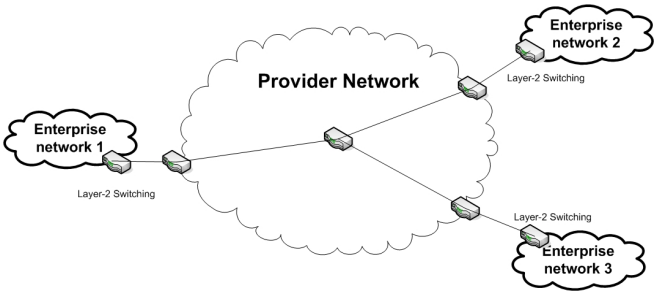

Figure 4: Virtual Private LAN Service

The second broad type of VPN is the layer-3 VPN. The provider offers network layer connectivity, which is at layer-3 in OSI-RM, between customer networks and provider network. The customers can optionally specify more advanced layer-3 topologies than simple partial or full mesh, typically intranet and extranet integration or hub-and-spoke. Forwarding in a layer-3 VPN is based on IP addresses, hence the provider edges and customer edges exchange routing information.

There are also two main approaches in layer-3 VPN. One is BGP/MPLS VPN, which is known as RFC 2547bis. Virtual Router (VR) is another solution that is shown in Figure 5. Both approaches concentrate on the VPN functionality at the PE and hide VPN-specific information from the P nodes, to improve scalability. In the BGP/MPLS VPN approach, a routing context is represented as a separate routing and forwarding table (VRF) in the PE. Each PE node runs a single instance of a BGP variant called Multiprotocol BGP (MPBGP) for VPN route distribution across the core network. PE nodes use MPLS labels to keep VPN traffic getting priority and transmitting packets across the core network in tunnels. The tunnels are not necessarily MPLS types. Tunnels can be of any types, such as IPSec or GRE tunnels. If a tunnel type other than MPLS is used, the only nodes that need to know about MPLS are the PEs. Any routing protocol can be run between Customer Edges (CEs) and PEs.

In the VR approach, PE nodes have one VR instance running for each VPN context. A VR emulates a physical router and functions. VRs belonging to the same VPN are connected to each other via tunnels across the core network. The tunnels can be of any types the same as RFC 2547bis.

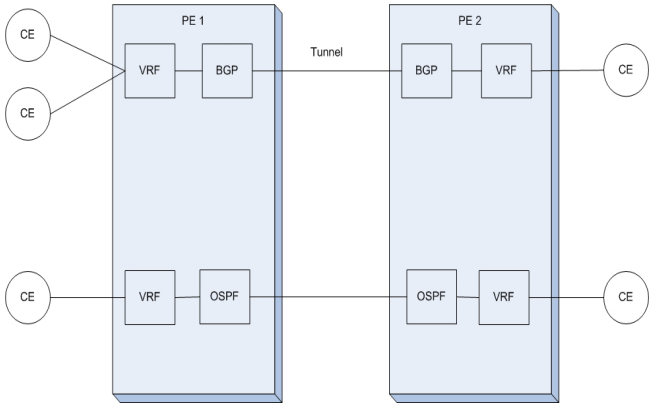

Figure 5: Virtual Router

Table 1 gives a comparison of 4 kinds of the VPN solutions. The layer-2 VPN is in some ways more flexible, particularly in terms of the higher layer protocols used in the VPN. The layer-2 VPN is also more scalable than layer-3 VPN from the provider perspective. The main reason is that when the provider offers layer-3 connectivity, routing information must be taken care of explicitly in the provider network. On the downside, some layer-2 solutions require that all the VPN sites run the same layer-2 protocol, which is not always possible. Layer-3 VPN can have advantages in terms of management. For example, in a managed layer-2 VPN, the customer is still responsible for all IP routing between the customer sites, whereas in a managed layer-3 VPN, the service provider can take over this management burden.

	VPWS	VPLS	RFC 2547bis	VR
Network type	ATM, FR	Ethernet	IPv4, IPv6	IPv4, IPv6
Traffic control	Yes, with MPLS	Yes, with MPLS	Yes, with MPLS	Yes, with MPLS
Security	Using MPLS, almost the same level as ATM or FR			
VPN support in CE	No	No	No	No
VPN support in PE	Yes	Yes	Yes	Yes
Solution scalability for PE devices	Well	Not well	Well	Not well
Solution scalability for sites	Poorly	10s	Well	100s
Maturity of technology	Mature	Immature	Mature	Immature
Migration and ongoing management cost	Low Migration, high ongoing management	Low	High migration, low ongoing management	High migration, low ongoing management
For Value-add Service	Difficult	Difficult	Easy	Easy

Table 1: The comparison of 4 kinds of VPN solutions

C. Experimental VPN structure

We chose the IP/MPLS based layer-3 VPN in the project. MPLS is the focus of research because it has gained increasing interest from service providers in recent years.

MPLS was originally used for traffic engineering purposes but now finds application in implementing provider provisioned VPNs. MPLS makes the service very simple for routing and controlling traffic. VPN service is very scalable and flexible to facilitate large scale deployment and management. MPLS allows the policies that are used to create a VPN to be implemented by the service provider alone, or by the service provider working together with the customer. The network structure is shown in Figure 6.

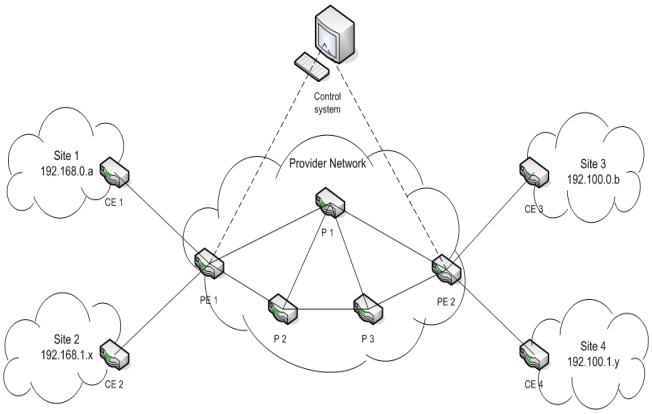

Figure 6: IP/MPLS Virtual Private Network Structure

The CE is the host for the site, and is typically a router. The CE is used for communicating with the PE router and distributing the information from the PE router to the local users. It can use the static route, EBGP to exchange the routing information with PE. Each PE contains a different VRF for linking with another PE. Each PE can use several different ports for one VRF. Therefore, the PE router can maintain multiple forwarding tables that support the per-VPN segregation of routing information. The data transport among PEs uses IBGP. PE routers can maintain IBGP sessions to route reflectors as an alternative to a full mesh of IBGP sessions. VPN data traffic is using MPLS across the provider's backbone, the ingress PE router functions as the ingress LSR and the egress PE router functions as the egress LSR. Any provider routers (P) cannot contact a CE router. They are confined to the PEs for forwarding data. P-routers are only required to maintain routes to the provider's PE routers; they are not required to maintain specific VPN routing information for each customer site. The control system shown on the top of Figure 6 is used for testing, setting up, managing, monitoring and maintain the VPN service. PE routers in the figure are all softswitches, thus the control system will be a computer with a host and interface software of VPN service while PE routers contain clients.

Assuming Site 1 and Site 3 are in company A, they can communicate with each other. Site 2 and Site 4 are in company B and can also link with each other. Site 1 and Site 2 stay in the same area sharing one PE. Site 3 and Site 4 are the same situation as Site 1 and Site 2. The data flow from one CE to another in VPN which is in figure 4 will be:

1. CE forwards packets to PE;
2. PE looks up the route in the VPN VRF and adds a MPLS label;
3. Packets are forwarded over the tunnel to the remote PE device;
4. The remote P device receives the packets, looks at the MPLS label, changes the MPLS label and transmits to the next P device or PE device;
5. Terminal PE uses the MPLS label to forward the packets to the terminal CE.

IV. VPN SERVICE IN APPLICATION LAYER

A. VPN service and Connectivity Management SCF

In OSA/Parlay, the standard is divided 14 parts and 15 Service Capability Features (SCF). Connectivity Management SCF is one of the SCFs that allow two parties to establish QoS parameters for private network packets traveling through the provider network. The QoS measures used in the private network are outside the scope of the service. The API does not require any specific QoS method to be used in the private network, nor in the provider network. However, in order for Provisioned QoS service to be applied to packets arriving from the private network into the provider network, the packets have to be marked using DSCP marking. Any packet without DSCP marking cannot have ensured QoS. The API is located between enterprise operator and SCS. The API support functions such as VPN installation, creation and modification of Virtual provisioned Pipes (VPrP). The API also supports getting and setting QoS parameters, service access points and sites in the provider network. The service uses and Object Management Group (OMG) Interactive Data Language (IDL) definition and was implemented in C++ in the project.

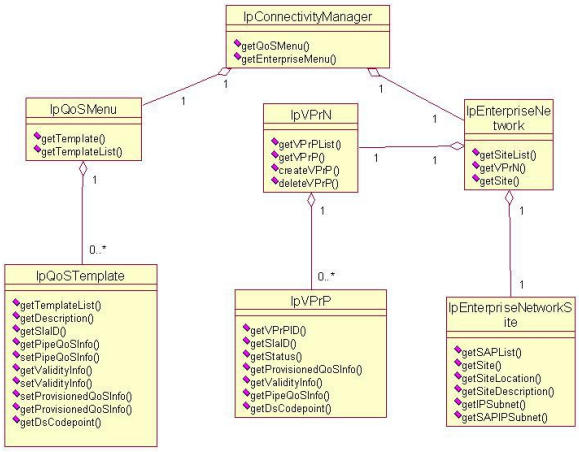

Figure 7: The Class diagram of Connectivity Management API

The IDL class diagram is shown in Figure 7. The service has seven interfaces. IpConnectivityManager is the entry point to the whole service. After the enterprise operator uses the framework authentication and authorization interface to log in, the operator's connectivity management (CM) application can enter and explore the network resource. CM has two branches. One direction is getting QoS menu interface. The QoS menu includes several QoS templates. Each template has different parameters, including the maximum and minimum delay and jitter in milliseconds, the acceptable packets loss, etc. Thus, by issuing different requests, the service provider can set various values and establish desired levels of QoS. This information can then be kept using the IpQoSTemplate interface. When the service needs the relevant QoS level, it is readily invoked from the template. Other interfaces inheriting from the IpConnectivityManager are concerned with enterprise network parameters. The IpEnterpriseNetworkSite can set

and retrieve IP addresses and subnet addresses for all the registered site access points and sites in the network. IpVPrP interface is used for setting and getting the states and information for the virtual provisioned pipe in the network. The information includes service level agreement (SLA), QoS and DSCP values. Usually, these parameters are set by the service provider. The service clients can give the information of local network to the service provider according the demand.

While the class diagram shows the static relationship among the classes and objects, the sequence diagram describes the dynamic behavior. Figure 8 shows the process of creating a virtual provisioned pipe by Connectivity Management API.

1. An operator client browses the IpConnectivityManager and asks for the network information.
2. It connects the IpEnterpriseNetwork and IpEnterpriseNetworkSite interfaces to get information of registered SAPs and sites.
3. The operator client also needs to know the status of the QoS. It links to IpQoSMenu and IpQoSMenuTemplate. From there, the operator can get and set the Pipe information, Provisioned QoS information and Validity information.
4. After getting the IP address of SAPs and sites with VPN service and the status of QoS in the network, a new virtual provisioned pipe can be created.

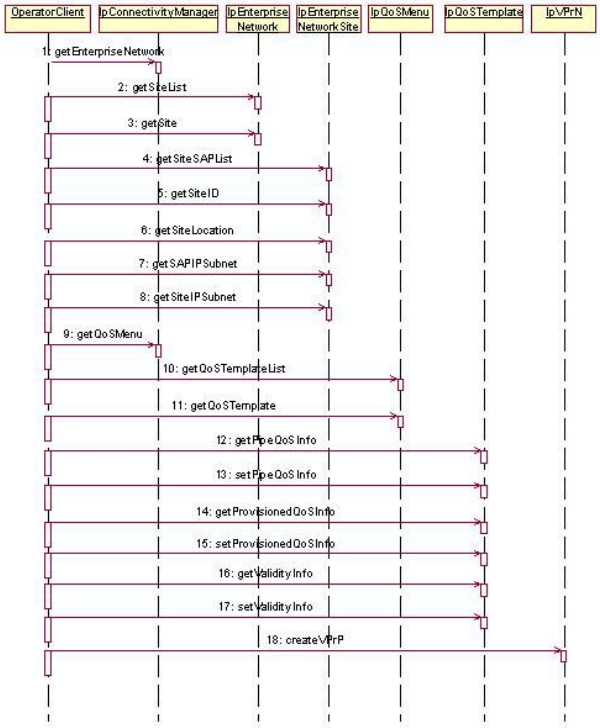

Figure 8: Sequence diagram of creating virtual provisioned pipe

Figure 8 illustrates the relationship between the classes. Here are Passive interface functions, which are grouped in Part 1, 2 and 3. Passive interface functions are used to retrieve information (get) relative to VPrP, VPrN, and QoS templates provided by the service provider. Active functions in Part 4 are used to provision and delete services (get and set). Some functions are numbered according to their appearance in Figure 8.

Part1: Retrieve information on a Virtual Private Network, including sites and their service access points:

1) getEnterpriseNetwork(IpConnectivityManager)
2) getSiteList(IpEterpriseNetwork)
3) getSite(IpEterpriseNetwork)
4) getSAPList(IpEnterpriseNetworkSite)
5) getSiteID(IpEnterpriseNetworkSite)
6) getSiteLocation(IpEnterpriseNetworkSite)
   getSiteDescription(IpEnterpriseNetworkSite)
7) getSAPIPSubnet(IpEnterpriseNetworkSite)
8) getSiteIPSubnet(IpEnterpriseNetworkSite)

Part2: Retrieve QoS services offered by provider, stored in QoS templates:

9) getQoSMenu(IpConnectivityManager)
10) getTemplateList(IpQoSMenu)
11) getTemplate(IpQoSMenu)
    getTemplateType(IpQoSTemplate)
    getDescription(IpQoSTemplate)
12) getPipeQoSInfo(IpQoSTemplate)
16) getValidityInfo(IpQoSTemplate)
14) getProvisionedQoSInfo(IpQoSTemplate)
    getDsCodepoint(IpQoSTemplate)

Part3: Retrieve information on a Virtual Provisioned Network and Virtual Provisioned Pipes:

getVPrN(IpEterpriseNetwork)
getVPrPList(IpVPrN)
getVPrP(IpVPrN)
getVPrPID(IpVPrP)
getSlaID(IpVPrP)
getStatus(IpVPrP)
getProvisionedQoSInfo(IpVPrP)
getPipeQoSInfo(IpVPrP)
getDsCodepoint(IpVPrP)

Part4: Set up a new Virtual Provisioned Pipe:

18) createVPrP(IpVPrN)
    deleteVPrP(IpVPrN)
    setSlaID(IpQoSTemplate)
13) setPipeQoSInfo(IpQoSTemplate)
17) setValidityInfo(IpQoSTemplate)
15) setProvisionedQoSInfo(IpQoSTemplate)

B. ACE and TAO

Implementation of an application using Parlay APIs requires a distributed processing environment. In this project we use the Adaptive Communication Environment (ACE). ACE is an object-oriented (OO) framework that implements many core design patterns for concurrent communication software. ACE provides a rich set of reusable C++ wrappers and framework components that perform common communication tasks across a range of Operating System (OS) platforms. Such communication tasks provided by ACE include event demultiplexing, and event handler dispatching, signal handling, service initialization, interprocess communication, shared memory management, message routing, dynamic reconfiguration of distributed services, concurrent execution and synchronization.

ACE is targeted for developers of high-performance and real-time communication services and applications. It simplifies the development of OO network applications and services that utilize interprocess communication, event demultiplexing, explicit dynamic linking, and concurrency. In addition, ACE automates system configuration and

reconfiguration by dynamically linking services into applications at run-time and executing these services in one or more processes or threads.

The ACE ORB (TAO) is a CORBA version 2.6 compliant, C++, object request broker (ORB). It is a second generation ORB developed with a highly extensible architecture as a result of its use of what might be termed a pattern framework, more commonly known as the ACE library. Although designed to meet the demanding requirements of hard real time systems, TAO can be easily used "out of the box" for general purpose CORBA middleware applications. Its design to meet exacting real time needs, which can be described as QoS requirements, has resulted in superior predictability, complete end to end determinism, high performance, and a scaleable implementation.

V. Conclusion

This paper has reviewed the OSA/Parlay standard, and the Connection Management API in particular, and the basic structure of a VPN. The advantages of each kind of VPN have been analyzed. The RFC 2547bis solution has been adopted. Our VPN service application is based on the ETSI OSA/Parlay Connectivity Management SCF. The application has been described using a class diagram and dynamic sequence diagram. The diagrams illustrate the OSA/Parlay VPN service being easily manageable and available for the application provider.

The VPN service simulation is based on a CORBA platform, using the OSA/Parlay standard. We have used TAO to compile the IDL file and generated the C++ implementation files. The implemented files will represent three programs which are interface, server and client. The simulation system will consist of a control computer and two softswitches. The control computer contains the interface and server programs. The softswitch contains the client program. The control computer uses the server program for creating, monitoring and modifying the VPN service.

VI. References

[1] E. Rosen, Y. Rekhter. RFC 2547bis: BGP/MPLS VPNs. http://www.ietf.org/rfc/rfc2547.txt, March 1999.

[2] P. Brittain, A. Farrel. MPLS Virtual Private Networks. Data Connection, November 2000.

[3] G. Rosenbaum, W Lau, S.Jha. An Analysis of Virtual Private Network Solutions. University of New South Wales, Australia.

[4] V. Fineburg. QoS Support in MPLS Networks. MPLS forum, Frame Relay forum, May 2003.

[5] MPLS: Resilient and scalable. MPLS World Congress 2003.

[6] A. Bagasrawala. Next Generation VPNs: Network-Based services based on Virtual Routing and MPLS delivering truly scaleable, customized VPNs. Lucent Technologies, 2001.

[7] C. Horney. Quality of Service and Multi-Protocol Label Switching. Nuntius.

[8] Z. Wang. Internet QoS: Architectures and Mechanisms for Quality of Service. Lucent Technologies, 2001.

[9] U. Black. QoS in wide area networks. Prentice Hall PTR, 2000.

[10] K. Nichols, S. Blake, F. Baker, D. Black. RFC2474: Definition of the Differentiated Service Field (DS Field) in the IPv4 and IPv6 Headers. Internet Society, December 1998.

[11] ETSI ES 202 915-10 v1.2.1: Open Service Access (OSA); Application Programming Interface (API); Part 10: Connectivity Manager SCF. ETSI Standard, August 2003.

[12] ETSI ES 202 915-2 v1.2.1: Open Service Access (OSA); Application Programming Interface (API); Part 2: Common Data Definitions. ETSI Standard, August 2003.

[13] TAO Developer’s Guide, Building a standard in performance Version 1.1a. OCI, 2000.