AFRL-IF-RS-TR-2003-233
Final Technical Report
October 2003

NEXT GENERATION VIRTUAL PRIVATE NETWORKS

Science Applications International Corporation

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J180

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-233 has been reviewed and is approved for publication.

APPROVED:

/s/

DANIEL J. HAGUE
Project Engineer

FOR THE DIRECTOR:

/s/

WARREN H. DEBANY, Jr.
Technical Advisor, Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Oct 03	Final Dec 99 – Jan 03

4. TITLE AND SUBTITLE

NEXT GENERATION VIRTUAL PRIVATE NETWORKS

5. FUNDING NUMBERS

C - F30602-00-C-0009
PE - 62301E
PR - J180
TA - 23
WU - 01

6. AUTHOR(S)

Thomas B. Harris and Andrei Ghettie

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Science Applications International Corporation
1100 First Avenue, Suite 300
King of Prussia, PA 19406

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/IFGA
525 Brooks Road
Rome, NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2003-233

11. SUPPLEMENTARY NOTES

DARPA Program Manager: Gary Koob, ITO, 703-696-7463
AFRL Project Engineer: Daniel J. Hague, IFGA, 315-330-1885, hagued@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

The objective of this study was to develop advanced technologies to exploit, enhance and demonstrate the capabilities of the Next Generation Internet (NGI). Our scope included both development of novel network technology and demanding NGI applications that were integrated to achieve new opportunities for regional collaboration. Our goal to achieve high bandwidth and low-latency, while simultaneously meeting stringent security requirements, was accomplished by the generation VPN that achieves end-to-end optimization of the network path based upon the requirements of individual applications. In addition to the development of the AA-VPN technology, we advanced and integrated sophisticated medical applications that require the high bandwidth, low-latency capabilities of the NGI. This included a regional medical archive system that enables healthcare institutions to archive and retrieve medical images in a realistic workflow environment for clinical care, research, education, public health, and disaster recovery. Multiple applications also allow pathologists to interact in real time over the AA-VPN including virtual microscopy and content-based image analysis.

14. SUBJECT TERMS

Next Generation Internet, Virtual Private Network, Quality of Service

15. NUMBER OF PAGES

36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

Section **Page**

List of Figures

Acknowledgments

The accomplishments realized during this Next Generation Internet program were the result of an impressive team effort. First, we wish to acknowledge the support and guidance of the DARPA/ITO Program Managers, Dr. Mari Maeda and Dr. Gary Koob. At Telcordia Technologies, Mr. Dan Daly and Dr. Nim Cheung were instrumental in the synthesis of the original Application-to-Application Virtual Private Network concept, and Mr. Daly served as the initial Telcordia Program Manager. At the Pittsburgh SuperComputing Center, Drs. Ralph Roskies and Mike Levine provided critical leadership and managerial support. Ms. Wendy Huntoon and her staff provided invaluable network support, and Mr. Nathan Stone contributed to the development of the Intelligent Archive application. Dr. Paul Chang (University of Pittsburgh) was the architect of the Intelligent Archive application, and Carlos Bentancourt was the principal developer. Also at the University of Pittsburgh, it was Dr. Michael Becich's vision to integrate multiple pathology applications into a collaborative environment.

Dr. Joel Saltz, who transferred from Johns Hopkins University to Ohio State University during this project, was responsible for the Virtual Microscope application, the first one to be integrated with the AA-VPN. At the University of Medicine and Dentistry New Jersey, Dr. David Foran was responsible for the Tele Microscope and Image Guided Decision Support System applications. Dr. Foran was also instrumental in executing the "3 site + 3 application" demonstration, which was hosted at UMDNJ.

1 Introduction

The objective of this study was to develop advanced technologies to exploit, enhance and demonstrate the capabilities of the Next Generation Internet (NGI). Our scope included both the development of novel network technology and demanding NGI applications that were integrated to achieve new opportunities for regional collaboration. Our goal to achieve high bandwidth and low-latency, while simultaneously meeting stringent security requirements, was accomplished by the development of an Application –to- Application Virtual Private Network (AA-VPN) architecture. This is a dynamic, next-generation VPN that achieves end-to-end optimization of the network path based upon the requirements of individual applications.

In addition to the development of the AA-VPN technology, we advanced and integrated sophisticated medical applications that require the high bandwidth, low-latency capabilities of the NGI. This included a regional medical archive system that enables healthcare institutions to archive and retrieve medical images in a realistic workflow environment for clinical care, research, education, public health, and disaster recovery. Multiple applications also allow pathologists to interact in real time over the AA-VPN including virtual microscopy and content-based image analysis.

This study was sponsored under DARPA's Next Generation Internet (NGI) Program. It was also a key element of the Hospitals, Universities, Businesses, and Schools (HUBS) initiative. The HUBS goal is to establish an advanced regional information technology infrastructure that enables new opportunities for collaboration between communities of interest. Initiated in the Four State Region of Delaware, Maryland, New Jersey and Pennsylvania, numerous technologies and applications are being developed to facilitate information sharing for economic growth, national defense, and quality of life improvements, including applications for enhanced educational and healthcare services.

The objective of the Application-to-Application Virtual Private Network (AA-VPN) system was to provide network management technology that enables demonstration of broadband applications in real-time medical imaging and electronic patient record archiving. In the this project, multiple distributed medical applications are running over NGI networks. These applications are:

- Intelligent archiving (IA) of electronic patient records (EPR). Patient records – including radiological images – are moved from regional archives to local servers where they are available to doctors during each patient's medical visit.
- Remote tele microscopy or virtual microscopy (TM or VM). Microscope servers enable users of the microscope client to manipulate and view medical samples remotely.
- Content based image retrieval (CBIR) and image guided decision support (IGDS). Microscopy images are compared against existing images stored in databases for diagnostic purposes.

These applications require network connectivity with predictable quality of service (QoS) and security attributes. Virtual Private Networks (VPNs) have evolved as the preferred enabling technology for distributed applications deployed over the Internet, mostly by virtue of their security features. Significant limitations nonetheless remain in current state of the art VPN implementations. An important restriction is that users do not have the means to flexibly allocate VPN resources between various applications. As a consequence, bandwidth must be overprovisoned when applications with stringent bandwidth, delay and jitter requirements are deployed on the VPN. Additionally, users do not have direct control over the VPN security configuration. Support for trust relationships that may change on a per session basis is impractical, due to the significant administrative overhead involved.

The Application to Application Virtual Private Network (AA-VPN) architecture extends current VPN capabilities to address these shortfalls. First, the AA-VPN provides dynamic QoS control for network aware applications, which enables QoS guarantees for applications with specific data transfer requirements ***as well as*** high utilization of VPN resources. To make such fine control scalable, ***local intelligence*** needs to deal with the complexity of dividing resources between users and applications. This local intelligence is split between the applications and AA-VPN. Network aware applications supply QoS parameters tailored for their respective traffic flows, while the AA-VPN distinguishes between various users and applications before passing their QoS requests to the network. The AA-VPN architecture allows scalable support for policy driven flow level QoS guarantees.

Second, the AA-VPN provides dynamic setup and teardown of secure VPN connections, as controlled by VPN users. Applications access AA-VPN encryption, message integrity, authentication and authorization services, and leverage this common infrastructure to meet their security requirements. By setting up the VPN connections dynamically, the AA-VPN enables a highly flexible trust model that does not require intervention from network administrators.

During this development project, a prototype version of the AA-VPN was designed, built, and integrated with multiple advanced medical imaging applications. The prototype was used to satisfy project deliverables, by conducting the following network experiments and demonstrations.

Table 1 – Application-to-Application Virtual Private Network Demonstrations

Date	Demonstration	Sites	Applications
7/01	Quality of Service network experiments	PSC	Virtual Microscope
1/02	Quality of Service network experiments	PSC	Intelligent Archive, Virtual Microscope
6/02	HUBS medical applications demonstration over Internet 2 backbone	PSC, UMDNJ, OSU	Intelligent Archive, Tele Microscope, Virtual Microscope
12/02	Security demonstration	Telcordia	prototype

The remainder of this document is divided into four sections. Section 2 presents the application network requirements and gives a detailed description of the AA-VPN architecture which was developed to meet these requirements. In section 3 the AA-VPN implementation and its status at the end of the project is documented. Section 4 contains details about the main project deliverables, with an emphasis on the Internet 2 and security demonstrations. Finally, section 5 describes technology transfer efforts related to the AA-VPN.

2 AA-VPN Architecture

Applications with demanding networking requirements were selected in order to establish the AA-VPN architecture. Several advanced medical applications were chosen due to their high bandwidth and low latency network requirements. In fact, both radiology and multiple pathology applications were selected in order to demonstrate that the AA-VPN is able to deliver the required quality of service and security capabilities ubiquitously to multiple applications.

The radiology and pathology departments in typical hospital systems are responsible for most medical imaging data and possess significant communications resources and domain expertise. Distributed applications running between users in these organizations need the basic characteristics of a virtual private network, i.e.:

- Security
- Performance
- Reliability

However, each application requires a different configuration for these properties. This means the virtual private network must also have a fourth characteristic:

- Customer Control

This section describes the requirements for application-based, dynamic control of VPN connectivity, and provides a detailed discussion of the Application-to-Application Virtual Private Network architecture which was developed to meet them.

2.1 HUBS Application Requirements

From a network performance perspective, the medical imaging applications involved in the project share a common client – server architecture, where clients download large amounts of digital data (1 MB or greater) from remote servers using the reliable transmission facility provided by the TCP protocol. These transfers generally occur as a result of user actions, which are infrequent on a networking timescale, and require an adequate response time (30 seconds or less). As an example, during the operation of the Tele Microscope (TM) when the operator of the TM client application moves the lens to a previously unviewed area of a pathology slide, a

request is sent to the TM server, and the corresponding data is sent back to the client, which fills in the detailed view window.

This communication requirement is shared by all the medical applications (TM, VM, IA, CBIR,

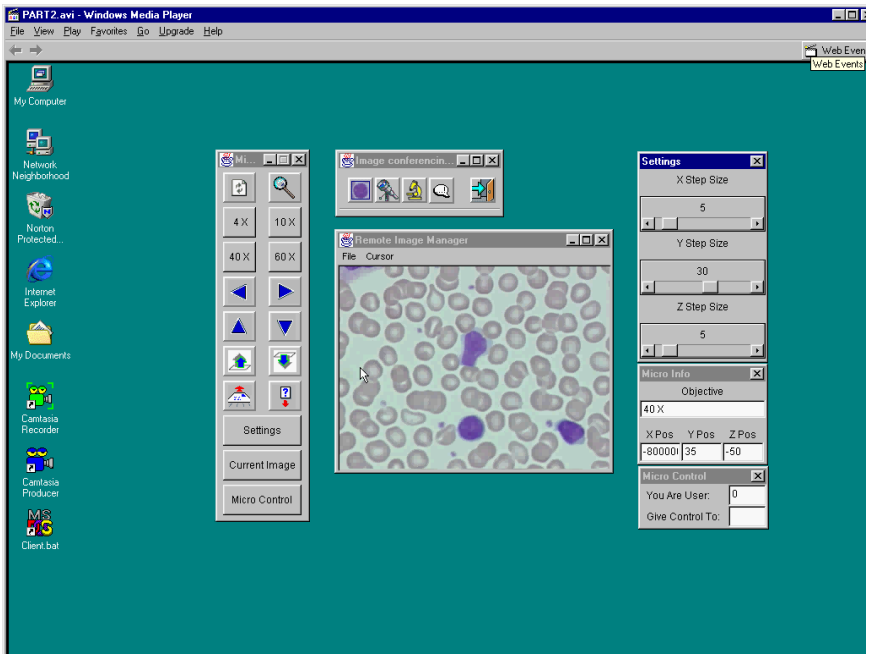

Figure 1. Tele Microscope Client Application

IGDS). The image guided decision support applications (CBIR and IGDS) impose the additional requirement of fast data upload to the server, since they also submit local imaging data for diagnostic analysis done at the server, which then returns gold standard database images along with its diagnosis. Finally, the IA application operates in two different modes. In "daytime" operation, radiology studies must be downloaded from the server for unscheduled patient visits. In "nighttime" operation, studies that have accumulated in the hospital database during the day are transferred to the archive server. In addition, studies needed by physicians for next day's scheduled patients must be downloaded from the archive into the hospital database. Therefore, IA "nighttime" operation requires sustained rates for bi directional bulk data transfers.

The actual bandwidth requirements of the particular applications depend on the size of the image data being transferred, and on the desired response times. These characteristics were studied, and the derived bandwidth requirements are presented in Table 2.

Table 2 – Medical Application Performance Requirements

Application	Bandwidth	Comments
IA nighttime	40 Mbits/sec	Must maintain high average TCP throughput.
IA daytime	25 Mbits/sec	Short bursts.
VM	10 Mbits/sec	Short bursts.
TM	10 Mbits/sec	Short bursts. Real time control of microscope also requires 50 ms one way delay for control flows.
CBIR	10 Mbits/sec	Short bursts. Upload and download.
IGDS	10 Mbits/sec	Short bursts. Upload and download.

Because imaging applications are generally characterized by very bursty traffic flows, performance guarantees are mutually exclusive with a reasonable level of resource utilization when static resource allocation is used. When multiple imaging sessions are carried over a statically allocated channel, one of two conditions are likely to occur. If the channel is relatively "narrow", the application gets unpredictable performance, as concurrent transfer of large data objects results in sharply lower throughput during ***and*** post contention, due to TCP dynamics. If the channel is relatively "wide", very low overall resource utilization will be observed. **In the AA-VPN approach, advanced applications and the network manage resources dynamically in order to achieve an effective utilization of capacity.** This maintains consistent TCP throughput, which in turn enables meeting the application response time requirements while maintaining a reasonable level of network utilization.

From a security standpoint, applications must meet the bandwidth requirements discussed above, while at the same time ensuring the privacy and integrity of sensitive medical information. COTS technology can be used to satisfy these requirements – namely hardware based implementations of the standard IPSec protocol. To maintain usability, the applications must perform **dynamic setup and teardown of IPSec tunnels between trusted participants**, without requiring intervention from the network administrator.

Finally, another common characteristic of the applications is that they are network aware. Application developers have access to source code and are able to modify them to take advantage of any network related functionality available at the API level.

2.2 AA-VPN Architecture

Today virtual private networks are composed of a defined group of end user sites and secure tunnels between these sites that have been provisioned by the network service provider. An important distinguishing service offered by the VPN is security, i.e. data privacy and user authentication. Service provider IP networks use a combination of provisioning and QoS related technologies such as Diff Serv, MPLS and policy based routing to overlay VPNs onto their networks. In most VPNs, individual users and applications on the same VPN are given identical QoS. Although not widely deployed, some service providers are also offering dedicated paths implementing a higher level of service, and can customize their CPE to route critical applications over those paths. In either case, this is a static configuration that can only be changed by making a service request to the network provider. A similar static approach is used with regard to security configuration, where VPNs usually implement a trust model that is defined at site granularity.

The type of networking technology provided by the AA-VPN is revolutionary in that it manages QoS and security parameters for the individual applications, not just the sites on which the applications are running. The AA-VPN provides users with the ability to manage network performance trade-offs. It enables the establishment of a VPN that is specified at a "coarse grain" level to the network, through management systems such as a bandwidth broker, but specified at a "fine grain" level to the application. This provides a scalable way for the VPN to serve the detailed performance and security needs of the applications without involving the network management systems of the network provider in various fine level details.

To achieve its objective of fine-grained, application controlled resource management, the AA-VPN distributes more of the network intelligence and control to the applications and to the group of users affiliated with the AA-VPN. This is accomplished using a single entity for each VPN – called the AA-VPN Services Manager (SM) – to translate the VPN user service requirements into network control parameters and then to communicate those parameters to the network infrastructure. The Services Manager acts as a management proxy for the application, and is identified in Figure 2 which presents the high level AA-VPN architecture. The architecture of the AA-VPN consists of:

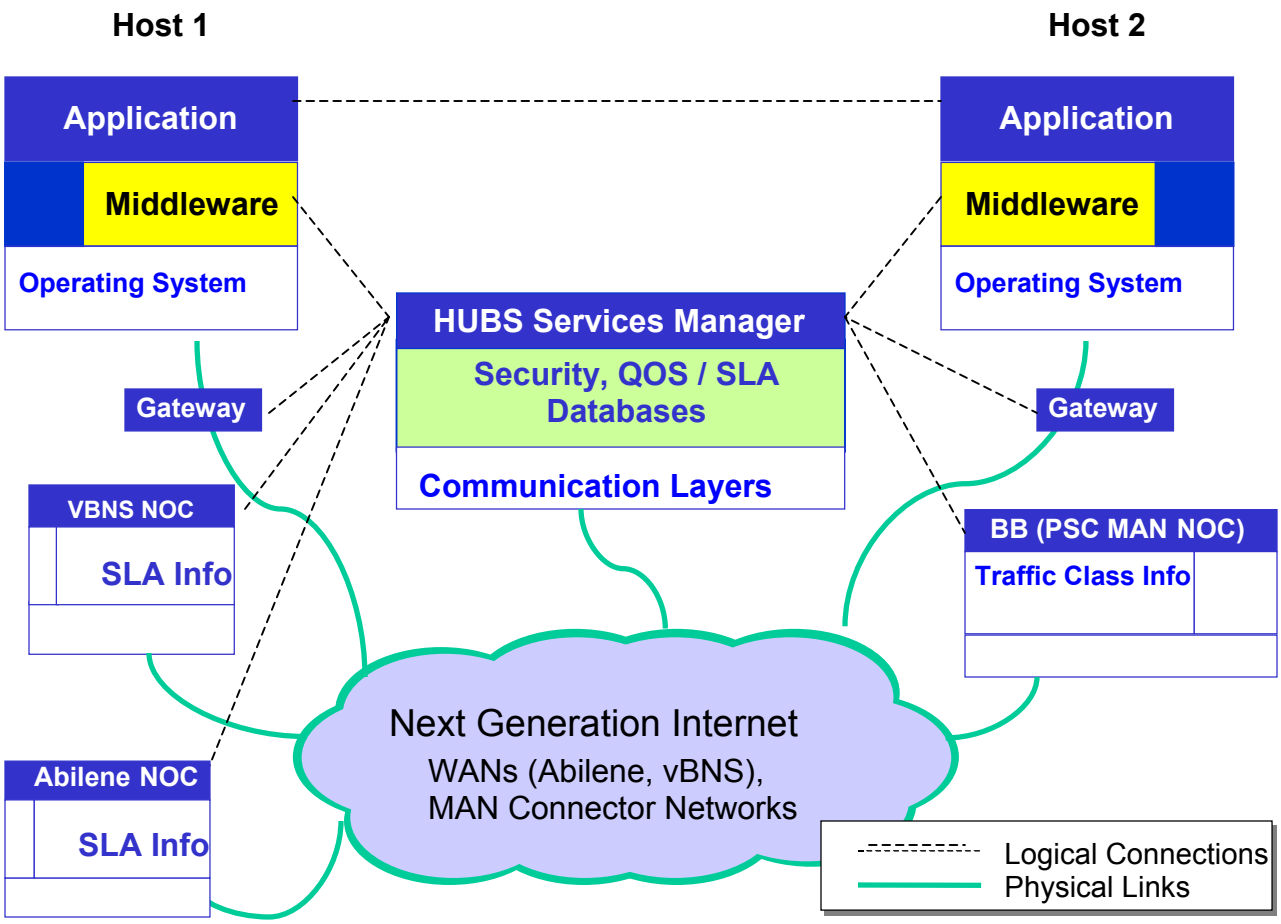

Figure 2 - AA-VPN Architecture

- The Services Manager that supports the establishment and teardown of the VPN connections.
- The Bandwidth Broker (BB), which is a resource manager which provides admission control and bandwidth management functionality for router based IP networks that support Differentiated Services (Diff Serv).
- Middleware on the application hosts that implements communication with the SM and the packet marking required for AA-VPN QoS services.
- Gateways that support high speed cryptographic functions required for AA-VPN security services
- Network elements that support Diff Serv and are configured according to either Service Level Agreements (SLAs) or according to traffic classes known to the Bandwidth Broker.

The function of each component is described in Section 2.3.

2.3 AA-VPN QoS Components and Data Flow

The AA-VPN middleware is implemented as a library that is linked with the applications. The middleware connects to the Services Manager at initialization, and performs a signaling function and a QoS policy enforcement function. Applications initiate the data flows required for AA-VPN by making AA-VPN middleware API calls. The API is similar to the well-known socket

```
enum FlowType {DefaultFlowType, Command, Video, Image, Bulk} ;
enum FlowDuration {DefaultFlowDuration, NoLimit,
                   DataSizeLimit} ;
int  aavpn_qos( int flow_id, int fd,
                FlowType flow_type,
                FlowDuration flow_duration,
                int data_rate, int data_size ) ;
```

Figure 3 – AA-VPN QoS Request C++ API Call

interface and also contains additional QoS related calls. Each API function follows the corresponding socket function, but requires additional parameters relevant to the AA-VPN. To make programming over the API as easy as possible, the aavpn_qos() function uses a flow type abstraction, which abstracts the desired delay, loss and burst size characteristics, as illustrated in Figure 3.

The policy enforcement function sets up the tagging of application packets for QOS purposes, and comes into play at the end of the QoS data flows. After the application initiates a QoS request via an API call, shown as step 1 of the data flow in Figure 4, the middleware initiates communication with the Services Manager to get the needed VPN resources. These middleware to Services Manager transactions are similar to out of band signaling in the public switched phone network. Application data does not go through the SM, but the VPN connection is set up dynamically before the data transfer occurs.

The AA-VPN offers several types of service to the applications, and maps each AA-VPN service to the most appropriate network service. It does that for each of the underlying networks.

The Services Manager receives messages corresponding to each AA-VPN API call, which is shown as step 2 of the QoS Data Flow diagram (Figure 4). In processing QoS requests, the first function is to map the high-level service request contained in the API call into one or more low level requests made to the network. The Services Manager uses provisioning data to perform this

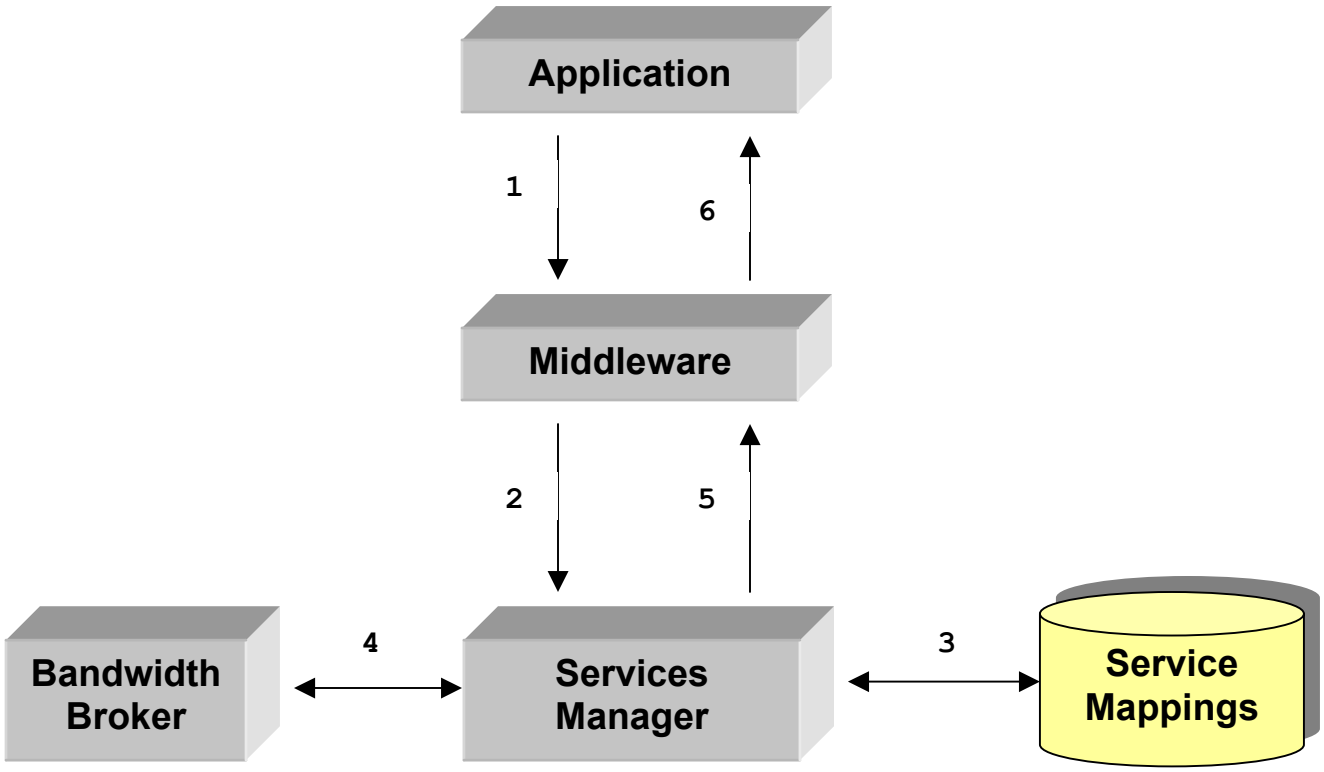

Figure 4 - AA-VPN QoS Data Flow

mapping, shown as step 3 in the QoS Data Flow diagram. The processing steps required to perform the service mappings are shown in Figure 5.

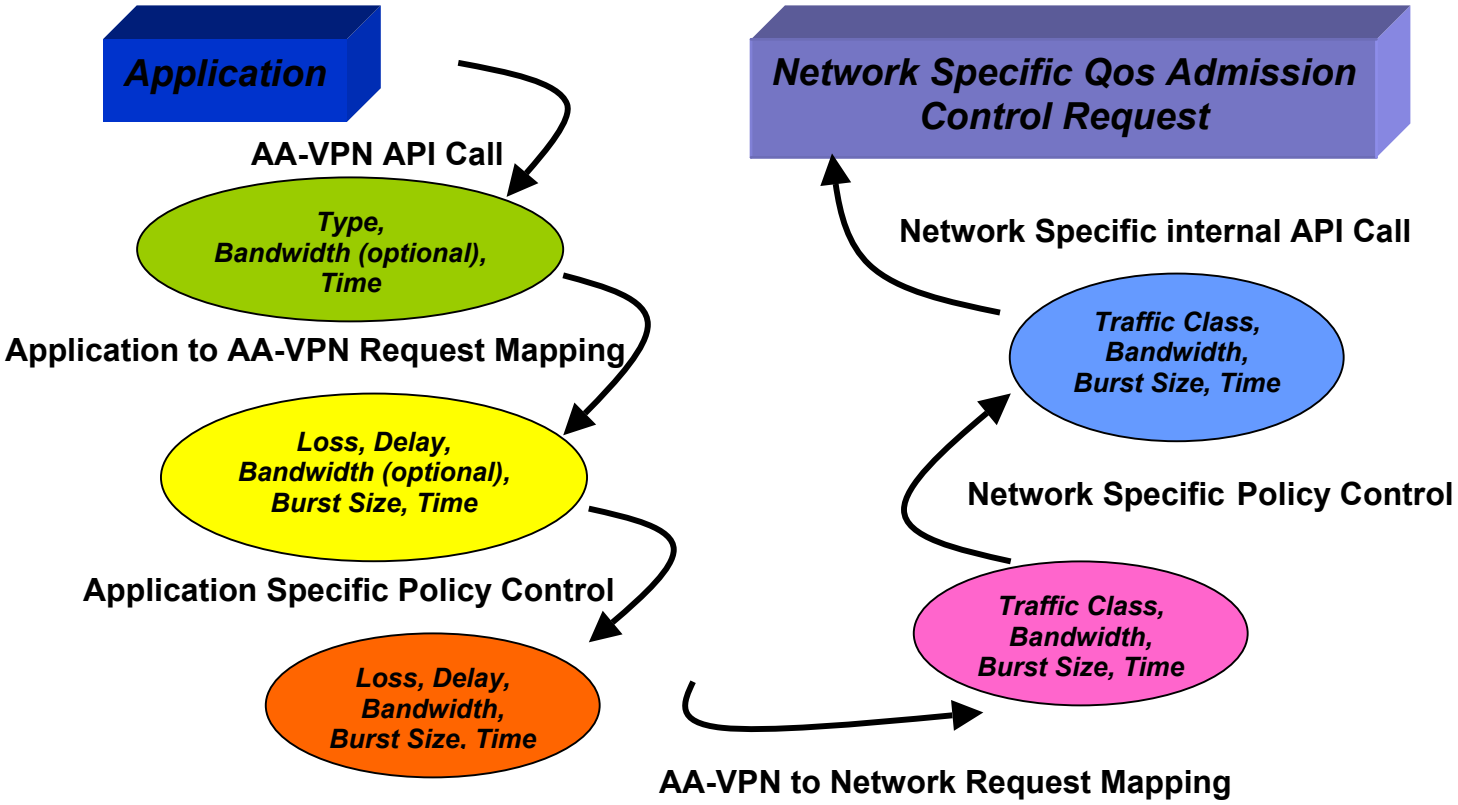

Figure 5 - AA-VPN Service Mappings

Once the application request is received, the flow type parameter is converted to loss, delay and burst size parameters. The application may specify its bandwidth requirement but is not required to. This recognizes the approach implicit in network programming that is to request data transmission at a rate that is "as fast as possible". The Services Manager then performs a second mapping step, which is to convert "as fast as possible" to an actual rate that gives the application its desired response time. At this point, the Services Manager has a generic request, which it converts to a specific network request (e.g. video application should use premium udp service). Finally, the Services Manager applies network specific policy control (e.g. premium udp is too expensive for this network, send best effort) before sending the request to the network.

Because application traffic flows may traverse several networks, the network specific mapping steps are executed for each of the traversed networks. The AA-VPN provides end-to-end QoS that is contingent on the support of the underlying network. **The QoS management capability is extended to cover a heterogeneous network environment, composed of multiple service provider networks, or of networks with various QoS management mechanisms**. The Services Manager is provisioned with network topology information and uses that information to

determine which networks it must negotiate with. In this architecture, we assume the networks support Differentiated Services, but it is not assumed that a common end-to-end service model exists between them.

After the service mapping steps are completed, the Service Manager has a complete picture of the QoS request for the specific data flows involved in the application. It knows the endpoints (IP address and port), the traversed networks, and the traffic class and flow parameters (rate, duration). Communication with "the network" is another function of the Services Manager. This function is exercised in step 4 of the QoS Data Flow diagram (Figure 4), and consists of an admission control request passed on to the Bandwidth Broker. The Services Manager also maintains state information about admitted flows, in order to make sure only valid requests are forwarded on to the Bandwidth Broker.

The AA-VPN bandwidth management capability is implemented in the Bandwidth Broker (BB). The BB provides admission control and bandwidth management for Diff Serv enabled IP networks. To do so, it relies on Class based WFQ in the routers which is provisioned according to guidelines known to the Bandwidth Broker. The admission control algorithms take into account factors such as flow multiplexing gain, which were evaluated for the applications through simulation studies. The Bandwidth Broker provides positive control over allocation of network resources by limiting the number of concurrent traffic flows that use the resources of a specific traffic class, therefore enabling performance guarantees. This approach stands in contrast with network aware applications, which adapt to changing network conditions by reducing their required bandwidth. Adaptive applications can operate more independently from network management systems, but will still fail when the network is under massive load.

If the admission control request succeeds, the Bandwidth Broker returns the DSCP value, which will be used to mark the packets for that flow. The Services Manager returns the DSCP to the middleware, which establishes the packet marking on the host platform. After packets enter the network, Diff Serv enabled IP routers use the DSCP to recognize packets as being part of the appropriate traffic class; and packets are queued appropriately to ensure the desired delay and loss characteristics.

A similar data flow occurs when QoS is released for a particular application. The only difference is the Services Manager uses state information about each flow, and avoids the service mapping steps described above.

2.4 AA-VPN Security Components

The AA-VPN provides a comprehensive set of security services to applications, including:

- Encryption
- Message integrity
- User authentication
- User authorization

The dynamic setup and teardown of secure VPN connections, as controlled by VPN users through the Services Manager allows users to establish a match between the desired security configurations of session participants. The secure VPN connections are based on the standard IP Sec protocol. Transform sets (i.e. choice of algorithms for encryption and message integrity) for particular traffic flows are chosen based on application demand. This helps avoid the problem of a host that is short of security resources (e.g. only implements DES but not 3-DES) bringing the whole VPN to the lowest common denominator where all VPN participants must match the limited capabilities of that host.

Application traffic flows are directed into IP Sec tunnels that match the application security requirements, as illustrated in Figure 6. The Services Manager is responsible for the creation and configuration of these tunnels, and this occurs when aavpn_connect() calls are issued by the application to the middleware. The Services Manager implementation maintains CLI control over the AA-VPN gateways by establishing a telnet session to each gateway. Tunnels are torn down when the aavpn_close() call is issued by the application. Because the medical applications generate high data rates, the cryptographic processing associated with IP Sec must be performed in hardware. For that reason, the IP Sec tunnels only extend between dedicated AA-VPN gateways at the respective sites, rather than end-to-end between hosts. The gateways may be either routers with high-speed crypto hardware or dedicated VPN devices.

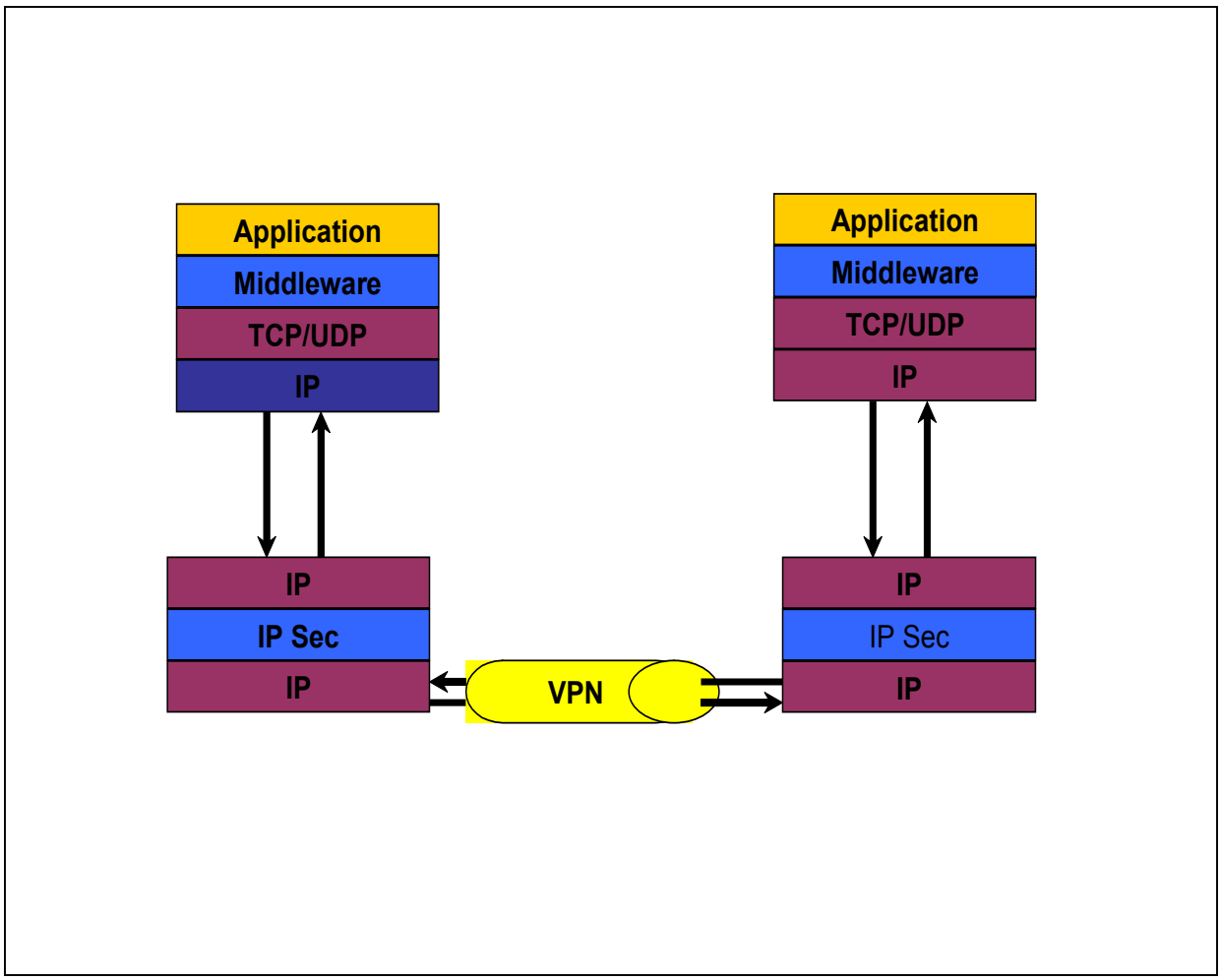

Figure 6. Gateway Based IP Sec Architecture

In addition to setting up IP Sec tunnels, the SM maintains databases with application-level authentication and authorization information.

3 AA-VPN Implementation

This section describes the current implementation status. Block level diagrams of the AA-VPN middleware and Services Manager are included. Additional functionality is included to support standalone applications, required for AA-VPN 2.0. *(Version 2.0 of the AA-VPN was developed with funding from the U.S. Army Space and Missile Defense Command (SMDC) to provide QoS for applications that are part of a collaborative engineering environment.)*

3.1 AA-VPN Middleware

The AA-VPN API supports the following functions:

- **aavpn_listen()**
- **aavpn_connect()**
- **aavpn_accept()**
- **aavpn_close()**
- **aavpn_qos()**
- **aavpn_qos_delete()**
- **aavpn_qos_recvdelete()**

The application owns the file descriptors associated with the various socket connections. The AA-VPN middleware is not involved in the **socket()** and **bind()** socket API calls. The middleware keeps track of all the sockets for which AA-VPN functions have been called, and of the corresponding source and destination host/port pairs, as indicated in Figure 7.

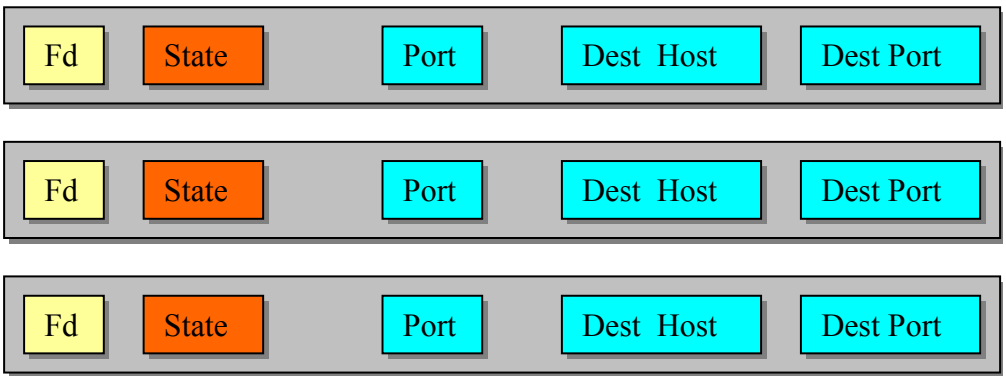

Figure 7. AA-VPN Middleware State Table

AA-VPN specific security processing occurs during the **listen(), connect(), accept()** and **close()** calls, which is why the application invokes the corresponding AA-VPN middleware functions. After the IP Sec tunnels are established, the middleware does not do any additional security processing; consequently the **send()** and **recv()** calls are made directly into the socket API. The **aavpn_qos()** function is called at the beginning of the session for long lived flows and at any time during a session of short lived flows.

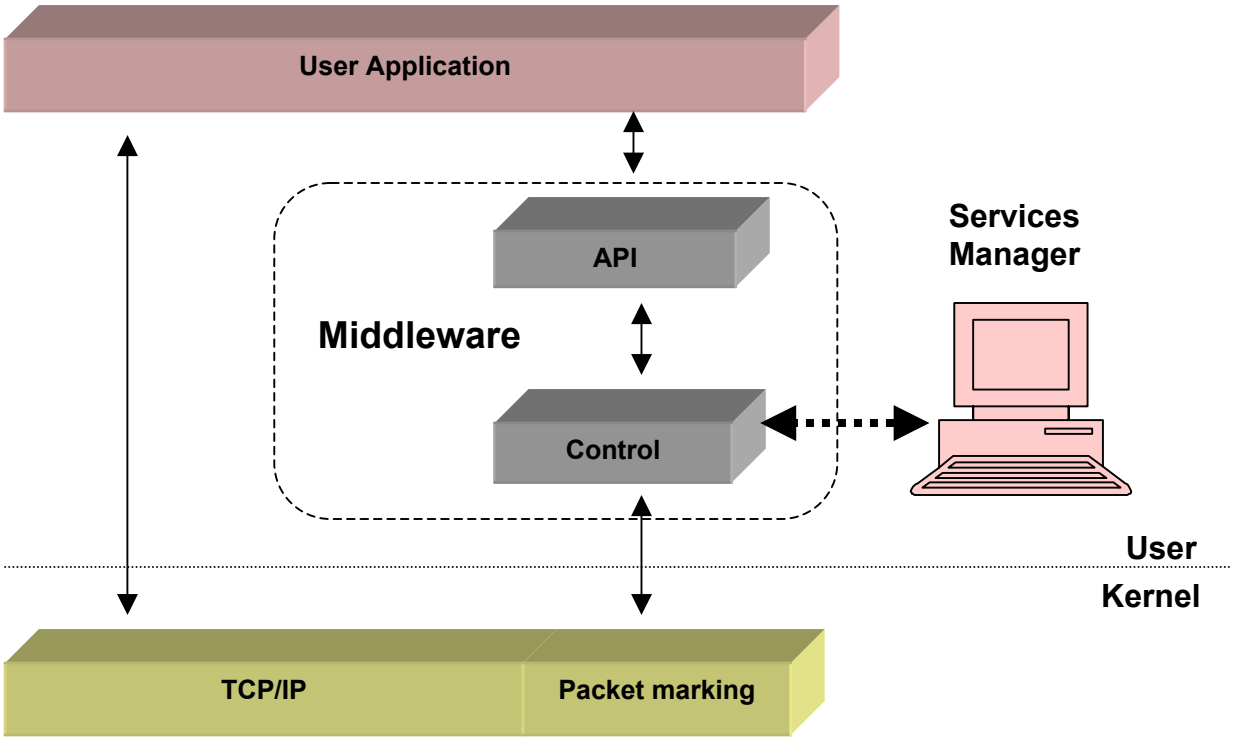

Figure 8 – AA-VPN Middleware Implementation

The AA-VPN middleware implementation is depicted in Figure 8. The middleware initiates interaction with the Services Manager, and protects the SM from application programming errors. It does that by keeping a state associated with each application socket, and only initiating interaction with the Services Manager when the socket is in a correct state (for example, if the application calls aavpn_accept() on a client-side socket, the error is detected and no messages are sent to the Services Manager). The allowed socket states reflect the last call made in the socket API sequence.

Before a message is sent to the Services Manager, the middleware encodes the data structures that are passed on into a corresponding message buffer. The message buffer is sent on to the Services Manager over a TCP socket to the Services Manager. The Service Manager responses are then received and decoded. Finally the middleware uses the **setsockopt()** system call to set

up packet marking on the application socket, according to the DSCP value returned by the SM. This functionality is available on the Solaris, Linux, Windows NT 4.0 *(With Service Pack 6)* and Windows 2000 operating systems.

The middleware implementation consists of C++ code, and the corresponding C++ API function calls. Solaris, Windows NT 4.0 and Windows 2000 implementations were developed as part of the project. A Java API was also developed that is mapped onto the C++ API calls via JNI. The Java API is used by the Tele Microscope application.

3.2 AA-VPN Services Manager

Figure 9 presents the block level implementation of the Services Manager. The Services Manager is implemented as a single-threaded JAVA application running over the Windows 2000 operating system. Native code methods are used to access a C++ layer that implements network communication with the AA-VPN middleware.

Application flows served by the AA-VPN may traverse multiple administrative domains, each one of which may offer different QoS support. Different QoS support characteristics can include:

- The network may be over provisioned, so that congestion rarely or never occurs
- The network may have a static prioritization and resource allocation scheme that enforces Service Level Agreements (SLAs), or
- The network may have a dynamic QoS capability implemented in a Bandwidth Broker (BB).

In order to establish connectivity between the server and clients located at different medical schools, our application flows typically traversed five networks as indicated in Figure 10.

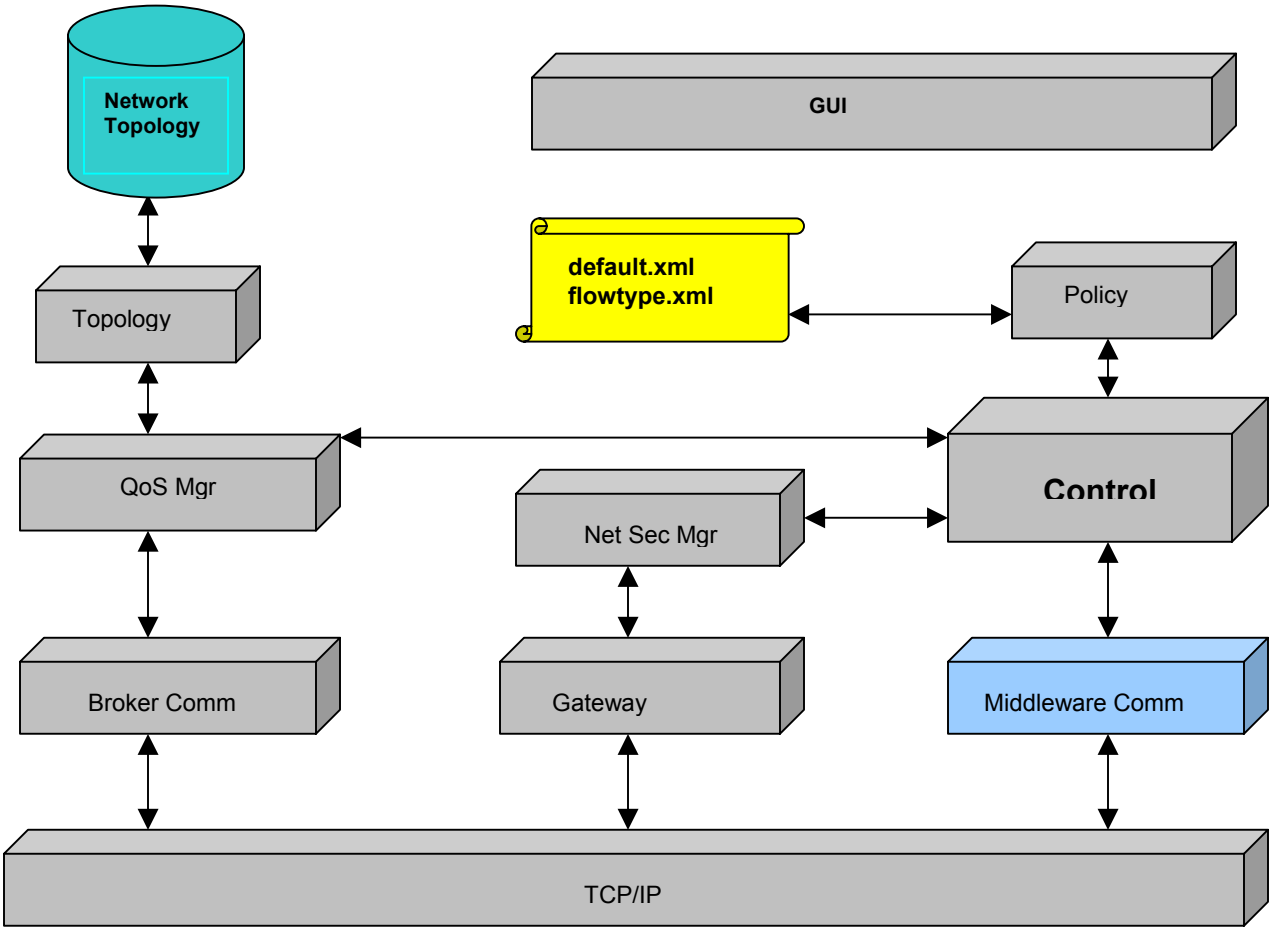

Figure 9 – AA-VPN Services Manager Implementation

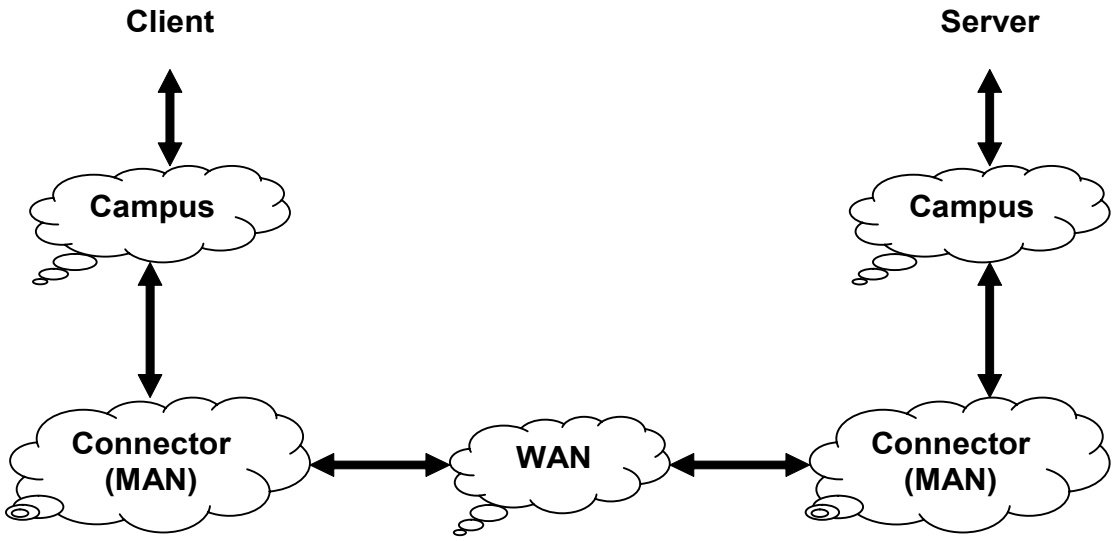

Figure 10. Application Network Connectivity

When the QOS module processes a request, it invokes the services of the Topology Module to obtain the list of networks traversed by the application flow. For each network, the QOS module maintains a NetworkQOS object. All NetworkQOS objects implement an interface for admission control requests stated in terms of AA-VPN traffic flow attributes. The service flow attribute contains the desired delay and loss characteristics for the flow. This is shown below:

dscp = admit (ingress, egress, duration, data_size, data_rate, service)

The QOS module is based on a class hierarchy where different implementation classes provide admission control functionality for different types of underlying networks. Each implementation class performs a mapping of the AA-VPN service to the "local" network service, and interfaces with network control and management systems when appropriate. The high level design is shown in Figure 11.

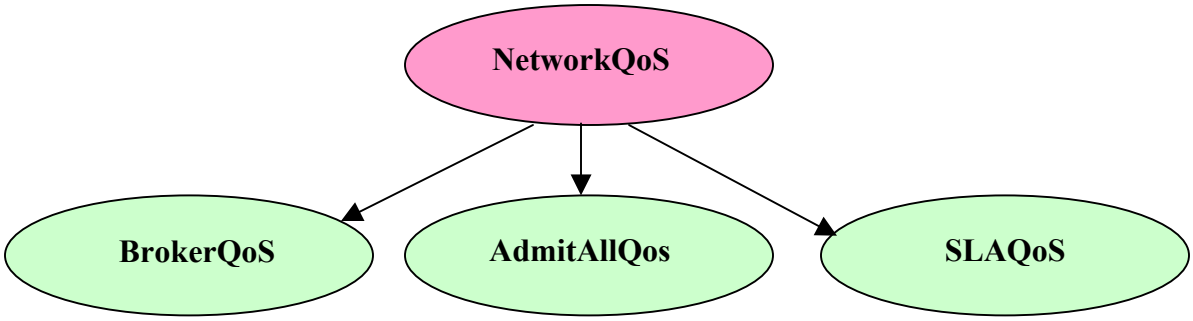

Figure 11. QoS Module Class Hierarchy

In addition to the service mapping, implementation classes keep track of various other network specific information. For example, SLAQOS objects will keep track of existing SLAs, and of currently allocated bandwidth for that specific traffic class.

The Policy Module manages default sets of QOS parameters, which are referenced by the application id and flow id parameters. These default parameter sets are contained in xml files, and are provided as a convenience feature to the application. This information is required for those applications that run over their own middleware layer (CORBA, HLA/RTI), and thus do not have direct access to the sockets used for network communication. In such cases, applications cannot specify the server port number their clients connect to, because that information is not accessible at the top layer of the application. They must instead rely on default configuration information that maps the application id and flow id to the actual server port (and even server host if needed).

The second type of information is a QOS definition of what is "as fast as possible" for a particular application, such as an image transfer. Since the high level performance requirements for the image transfer applications is to transfer a large amount of data over TCP "as fast as possible", policies provide a better definition, dependent on application, site, user, time of day, etc.

The Topology Module provides two services, one each to the Network Security Manager Module, and to the QOS Manager Module. The first service returns the pair of AA-VPN gateways that handle traffic for a given connection. The second service returns a list of networks traversed by a connection between two addresses. The QOS Manager Module needs this information to request admission control from each of the networks traversed by the connection.

Dedicated database tables are used for both services. Because the current deployment does not support a large number of hosts and AA-VPN gateways, the tables can be populated manually. Any database supporting JDBC can be used, and the Pointbase Java database, which comes with the Visual Café development environment, was selected.

Table 3 - Connection to AA-VPN Gateway Mapping Table

Server Host	Client Host	Server gateway	Server gateway interface	Client gateway	Client gateway interface
a.b.c.d	x.y.z.w	a.b.c.e	f.g.h.i	x.y.z.v	m.n.p.q

The host to gateway mapping table, shown in Table 3, contains the IP address of the gateway and also the IP address of the outgoing interface that will carry the IP Sec tunnel. When the Network Security Manager Module sets up a tunnel between two hosts, it gets the information for both hosts. The two addresses from the gateway interface column are the endpoints of the IP Sec tunnel.

Table 4 - Traversed Networks Table

Src Host	Dest Host	Network List
a.b.c.d	x.y.z.w	a.b.c.0, p.q.r.0, m.n.0.0, f.g.h.0, x.y.z.0

The traversed networks table, depicted in Table 4, contains a comma-separated list of network addresses for every pair of hosts in the table. The QOS Manager Module uses this information to request admission control for application flows from each network crossed by those flows. Several enhancements to this design are required as the number of supported hosts is scaled up. The table was simplified by keying the network list with subnet addresses instead of host addresses, and it could also be maintained dynamically by periodically running the traceroute program and analyzing its output.

The Topology Module accesses the database tables via JDBC, and provides an API for its services to the other Services Manager modules. A utility was provided to facilitate direct user control of the database tables, which is required for initially populating the tables, and is useful for diagnostic purposes.

Finally, the Services Manager GUI implements an Exit button, various menus for editing the xml files containing provisioning information, and a window which displays the Services Manager trace output.

4 Network Experiments and Demonstrations

As part of this project, two sets of network QoS experiments were carried out over the Pittsburgh GigaPoP network, followed by a 3 site + 3 application QoS demonstration where AA-VPN enabled HUBS applications were run over the Internet 2 backbone between several sites. Finally, an AA-VPN security demonstration was conducted in the lab at Telcordia. These experiments and demonstrations are summarized in this section.

4.1 Pittsburgh GigaPop Network Experiments

4.1.1 Bandwidth Broker Admission Control Guidelines

The network QoS experiment conducted over the Pittsburgh GigaPop network considered this network as one IP domain enabled for Differentiated Services. To provide dynamic QoS management in such a network, a service provider would use a bandwidth broker system. As part of the experiment, we derived the appropriate traffic classes and admission control guidelines from simulations, as described below. We then customized the Telcordia Bandwidth Broker to support the medical imaging applications. The experiment used Telcordia's methodology for provisioning IP QoS for the image transfer applications. The simulation results were used to implement the appropriate QoS configuration for the routers in the GigaPop network**.**

Simulations are used to determine the multiplexing gain that can be extracted from a traffic class, subject to their QoS requirements being satisfied. In this case, a single traffic class was used for all the TCP-based image transfer applications (Virtual Microscope [VM], Integrated Guidance and Decision System [IGDS], Content Based Image Retrieval [CBIR], Intelligent Archiving with real-time update [IA-daytime]). An analysis of the various image transfer applications revealed that an average bandwidth of 8 Mbps per flow was required for a successful user experience. Class-based Weighted Fair Queuing was used in core routers to assure minimum bandwidth to various traffic classes. Most COTS routers support some version of WFQ for class-based resource allocation. The ns-2 simulation captured the network topology that would be used for the AA-VPN experiments. Average, 95 percentile, and maximum image transfer delay, and the achieved throughput were used as metrics, while the average packet loss rate was used as an indication of network load.

The simulation was also used to determine the buffer size to be used for the WFQ mechanism in the core routers for the image transfer application class. The image transfer applications do not offer significant multiplexing gain, due to the large bandwidth requirement of individual image transfer flows, as compared to the link capacity allocated for the image transfer traffic class. Also, while the performance of all applications was similar during normal network conditions, they were affected differently by network congestion. It is thus important to ensure that the network load does not reach the point where applications in the same traffic class do not receive uniform service. While all the image transfer applications have the same average bandwidth

requirement, the IA-daytime application has a significantly larger image size compared to the other applications. An interesting effect of this fact is that IA-daytime exerts a significantly larger influence on overall traffic class performance as compared to other application flows. This is because the IA-daytime flows are longer lived, hence they manage to exist long enough to attempt stealing bandwidth from existing flows belonging to other applications. This phenomenon may argue for considering flow-duration as well as average bandwidth requirement for TCP-based applications, when assigning applications to traffic classes.

4.1.2 QoS Experiment Design

The experiments conducted on the Pittsburgh GigaPop network consisted of running multiple instances of the Virtual Microscope (VM). Two VM server workstations were located at the

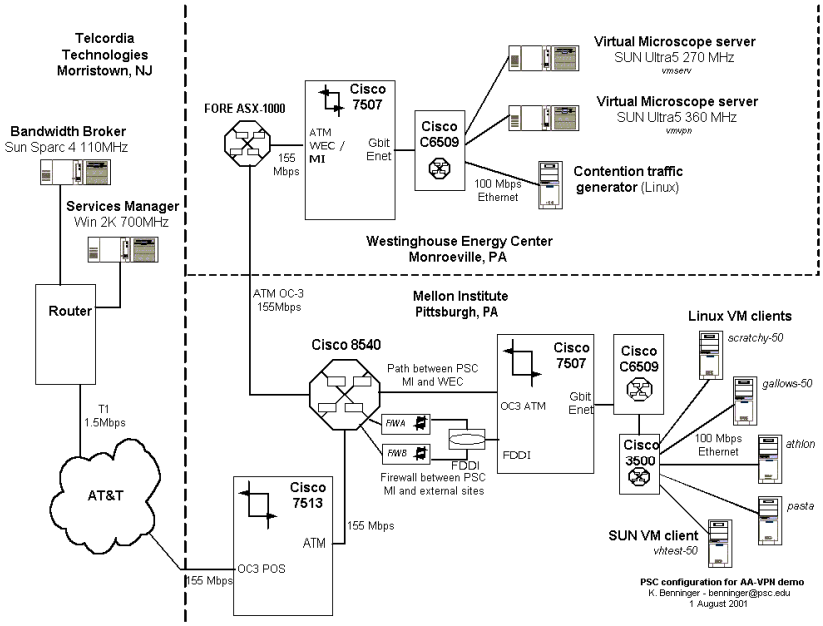

Figure 12 – PSC Experiment Network

Westinghouse Energy Center (WEC) and five VM client workstations were located at Mellon Institute (MI). The two facilities are located approximately 20 miles apart, and are connected via an ATM link. The network configuration is shown in Figure 11:

The AA-VPN Services Manager (and the Telcordia Bandwidth Broker) were located remotely at Telcordia Technologies, in Morristown, NJ, and were accessed via the Internet (through the AT&T commodity Internet connection). This scheme was feasible, because only small amounts

of control data need to be exchanged between the high-speed network and these management entities.

To facilitate the experiments, the Cisco 7507 routers were augmented by installing in each a VIP2-50 carrier with enhanced ATM port adapter (PA-A3) on loan from Cisco Systems. These routers were interconnected via a UBR PVC that was constrained on the interfaces to a peak cell rate (PCR) of 40000 kb/s. This PVC was provisioned on a PVP which was burstable to a full 155 Mb/s; the normal traffic on the PVC is on the order of 10 Mb/s. Per-VC class-based weighted fair queuing was enabled on these interfaces to support the QoS model for the experiments. We provisioned a Premium TCP traffic class with 24 Mbits/sec bandwidth, a Premium UDP class with 8 Mbits/sec and a Best Effort class with the remaining 8 Mbits/sec. The Virtual Microscope application used the Premium TCP class. This feature required an upgrade of the system software on both routers, and the installation of additional memory on the router at the Mellon Institute site.

The original experimental setup consisted of up to 5 Virtual Microscope clients running concurrently and downloading images over the network. A bandwidth of 8 Mbps was deemed adequate to support each Virtual Microscope session. This number was derived from a typical Virtual Microscope client cache refill event which causes the transfer of approximately 1 MB of data, and for which a 1 second response time is considered adequate. Therefore, given 40 Mbps overall bandwidth, an unloaded link should provide adequate performance to all Virtual Microscope clients. During the experiments we varied the number of active Virtual Microscope clients from 1 to 4, and also varied the amount of contention traffic as 0% (none), 20% (8 Mb/s), 50% (20 Mb/s) and 80% (32 Mb/s) of the overall bandwidth. This was done for both the original Virtual Microscope application (which does not use the QoS provisioned in the network), as well as for the AA-VPN enabled VM application. Because the Premium TCP traffic class used by the Virtual Microscope application is provisioned with 24 Mbps bandwidth, the expected outcome is to show the AA-VPN can provide adequate performance for up to 3 Virtual Microscope sessions even under extreme contention traffic.

4.1.3 QoS Experiment Results

The experiments showed that the AA-VPN and Bandwidth Broker were successful in providing the desired quality of service for high bandwidth medical imaging applications. Without the AA-VPN, these applications would not function as required on a heavily loaded network. We determined the desirable response times for various number of clients, taking into account the difference in processor speed between the two servers, and the sequential handling of client requests by each server. The desirable response times for various numbers of clients are indicated by the "adequate performance" line on Figure 13. We then compared the desirable performance to what was actually observed with and without the AA-VPN.

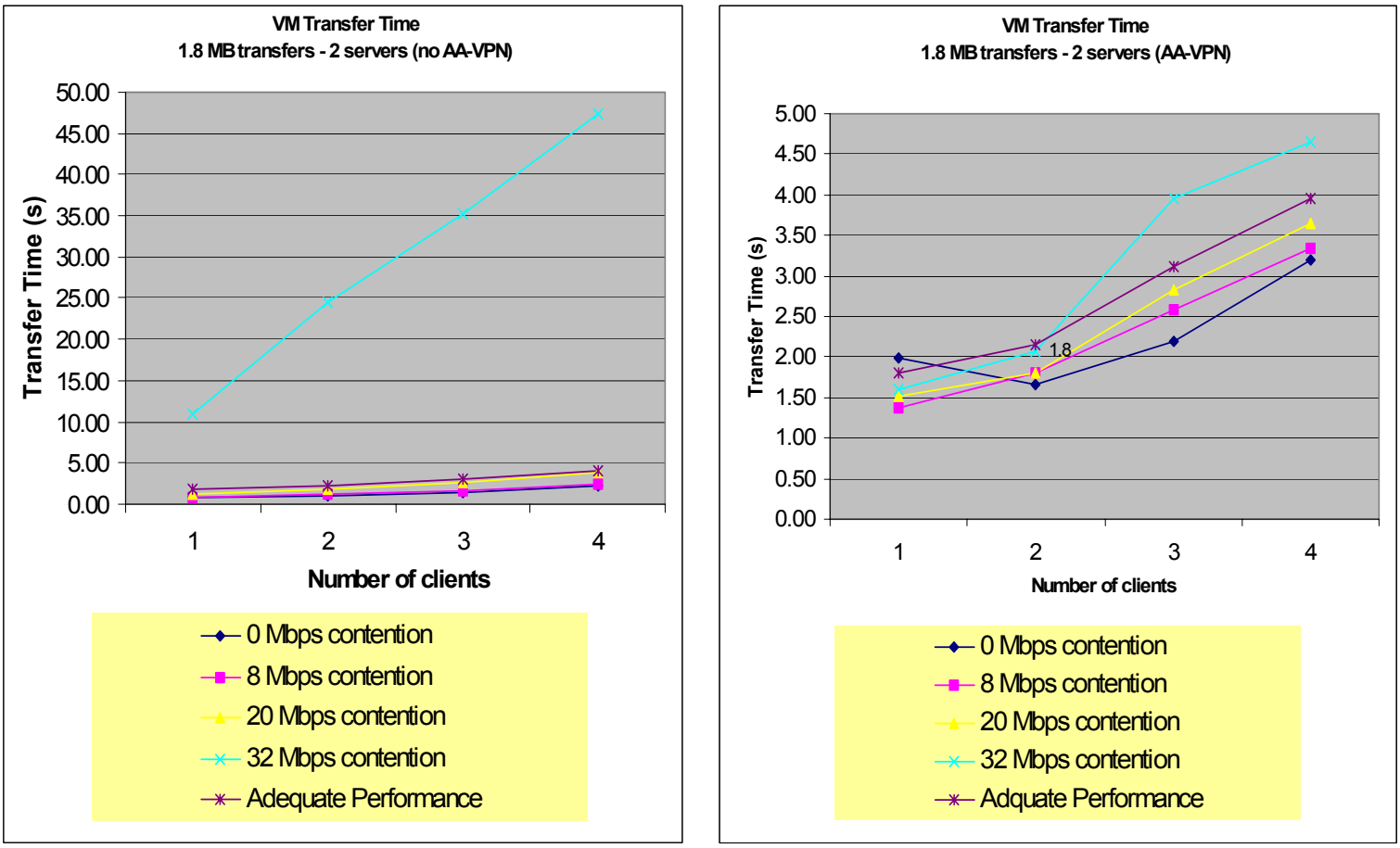

Figure 13 – PSC Network Experiment Results

Without the AA-VPN the applications stayed within desirable performance bounds with up to 50% contention. This corresponds to the expected behavior. At 50% network contention we have 20 Mbps of unused bandwidth which is adequate for 2 concurrent image downloads. Because we can have at most one active image download per server, we expect to see adequate performance. At 80% contention, however, the AA-VPN provides an improvement of about 1 order of magnitude, and maintains the application response times within or close to desirable values. We observe that the AA-VPN imposes a processing overhead of about 1 sec for each transaction, which is worthwhile to ensure desired performance when the network is heavily loaded.

4.2 3 Site + 3 Application Demonstration

In June 2002, a demonstration of AA-VPN enabled high bandwidth medical imaging applications was carried out over Internet 2. This was the culmination of the work conducted in the project networking task. We succesfully integrated the efforts of the network connectivity task, the AA-VPN task, and the medical application tasks. In order to carry out the demonstration, we ensured end-to-end high bandwidth connectivity over Internet 2 between participating sites, which required debugging some problems found in the campus network infrastructures. The AA-VPN was successfully deployed in a WAN environment, and it performed as expected. The original project goal of running advanced distributed medical applications over the Internet 2 backbone was met by executing the "3 site + 3 application" demonstration.

The demonstrated applications were:

- Virtual Microscope (VM)
- Tele Microscope (TM)
- Intelligent Archive (IA)

All three applications were demonstrated with and without the AA-VPN, and in the presence of contention traffic. The AA-VPN effectively maintained throughput for the applications, while operating over a wide area network environment. The network topology is shown in Figure 14.

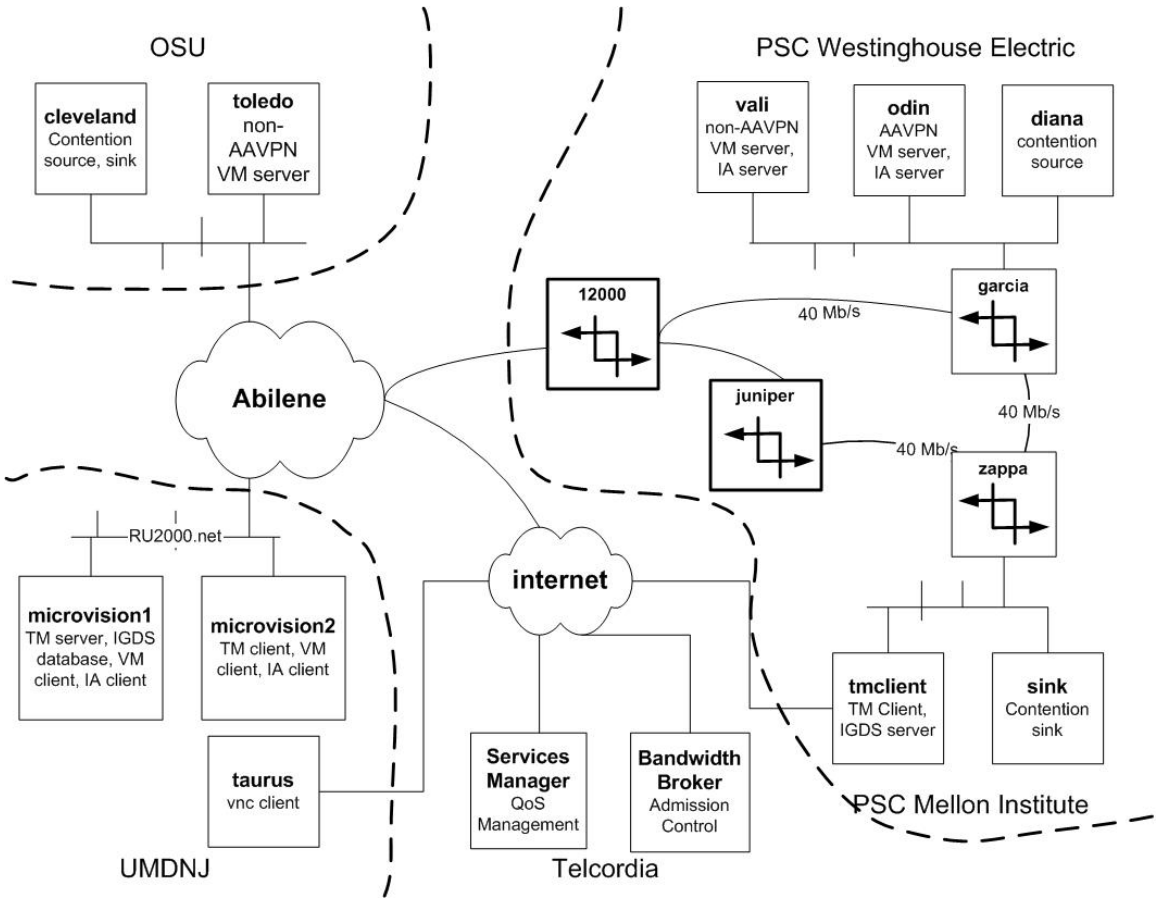

Figure 14 - 3 Site + 3 Application Demonstration Network Topology

Sustained data rates of more than 30 Mbps were achieved between UMDNJ / Rutgers and PSC, over the Abilene backbone. While the WAN connections are more than able to support this bandwidth requirement, several problems had to be resolved in the UMDNJ / Rutgers campus infrastructure. Contention traffic was injected into the network and competed with all three applications for bandwidth over the PSC network. As an example, Intelligent Archive (IA) application flows used during the demonstration are shown in Figure 15.

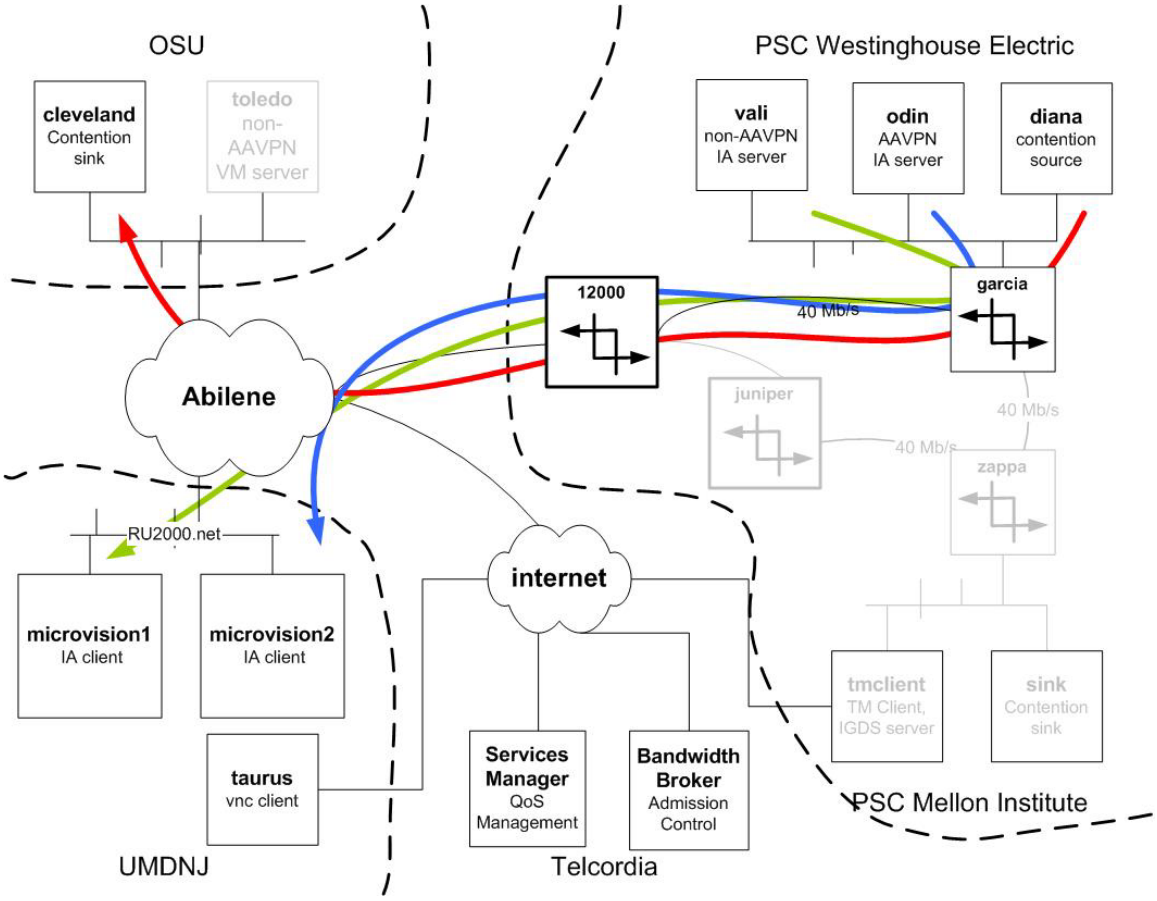

Figure 15 – 3 Site + 3 Application Demonstration IA Flows

Contention traffic is shown in red, non-protected IA traffic is shown in green, and AA-VPN protected traffic – generated from a server that is linked with the AA-VPN middleware – is shown in blue. Both the AA-VPN enabled server and client were communicating with the AA-VPN Services Manager located in the Telcordia DMZ. *(The AA-VPN components were distributed over the WAN. QoS was supported over the Pittsburgh GigaPop network only.)*

4.3 Security Demonstration

A demonstration of the AA-VPN security features was held at Telcordia in December 2002. The pseudo-app server and client were enhanced to support the following two security configurations, specified via a command line argument.**Security <1> : IP Sec ESP with**

- **TDES-MD5 transform**

- **Security <2> : IP Sec AH with SHA1 transform.** The application transferred 10 MB of data are transferred over TCP. IP Sec security associations (tunnels) are created and deleted as a result of AA-VPN middleware calls. IP Sec shared secret crypto maps are pre-configured on the routers, but the corresponding access control lists are updated during aavpn_connect() and cleared during aavpn_close(). Whenever such an update occurs, the crypto maps are reapplied to the respective interfaces, and the new security associations are established. The lab network topology is shown below.

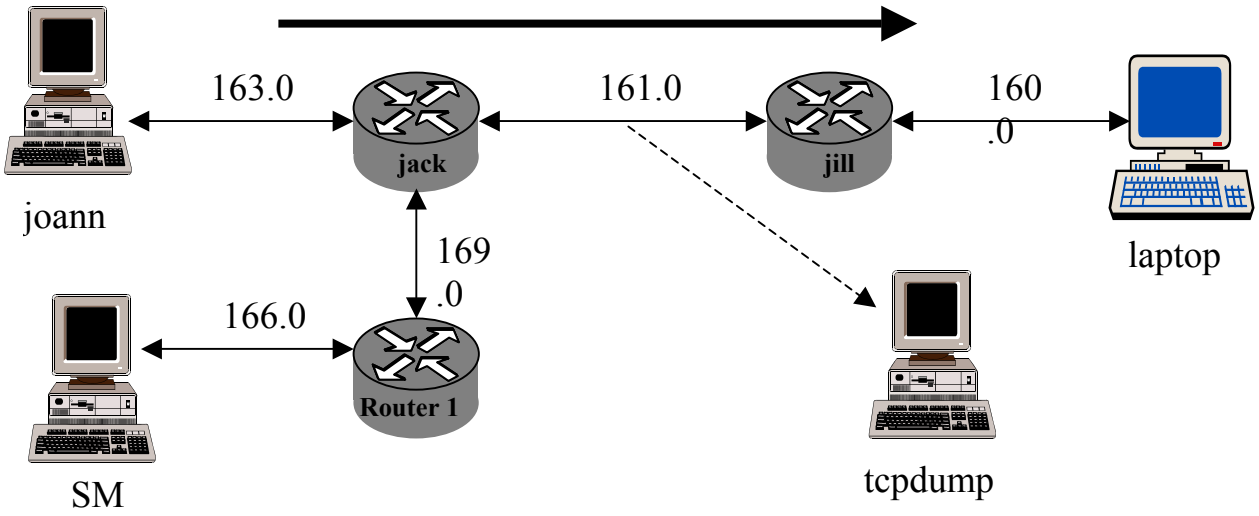

Figure 16 – Security Demonstration Lab Network Topology

When the AA-VPN is not active, the tcpdump packet sniffer detects TCP traffic between the two communicating hosts (joann, laptop). We also see an identical payload in the data portion of every packet. When the AA-VPN is active, we only see IP Sec ESP and AH traffic between the IP Sec tunnel mode endpoints (peer1, peer2), which hides the ip address of the communicating hosts, as well as the higher layer protocol information (TCP port numbers, etc.). In addition the payload is different in the data portion of every packet, which is a result of the ESP encryption that takes place.

This demonstration showed succesful setup and teardown of IP Sec security associations under application control.

5 Technology Transfer

The AA-VPN QoS functions, named Session-Level QoS Management System (SL-QMS), are being enhanced in a Missile Defense Agency (MDA) funded project as an enabling technology for distributed real-time simulation and for collaborative engineering applications running over the Missle Defense Agency network (MDAnet). The main difference between the HUBS project and the Wide Bandwidth Information Infrastructure (WBII) / Wide Bandwidth Technology (WBT) projects is the additional requirement of supporting standalone applications which are not network aware, and cannot be linked with the AA-VPN middleware.

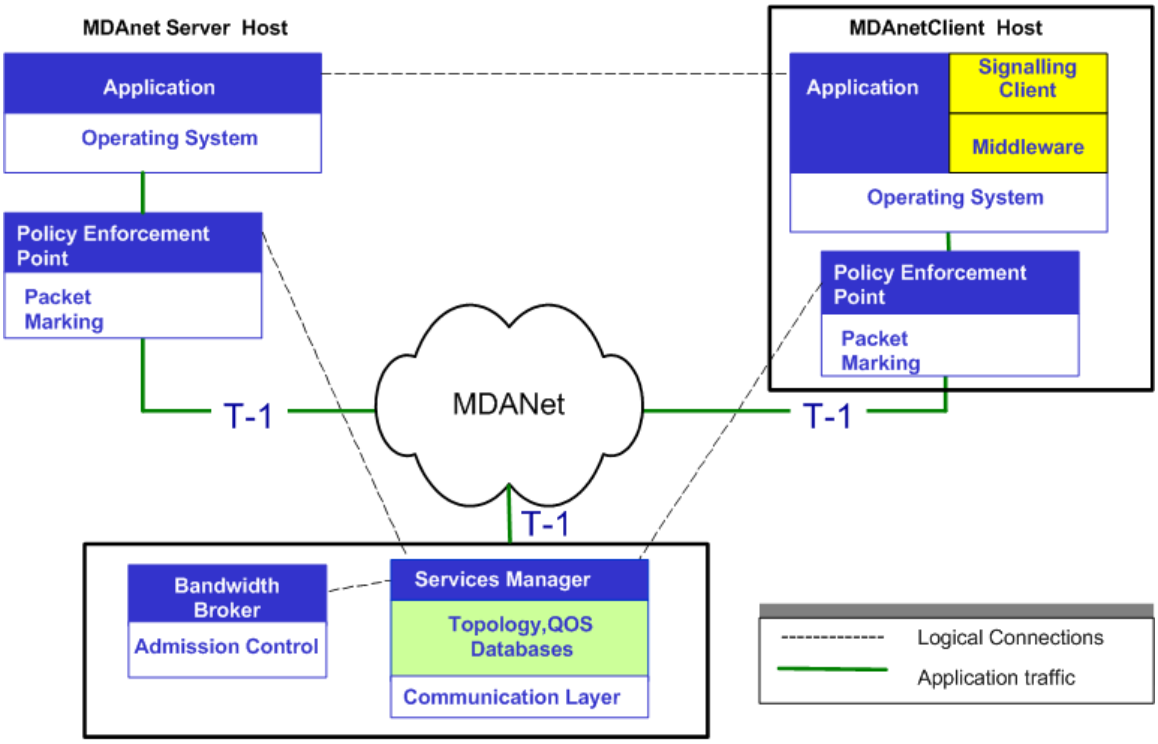

Figure 17 – AA-VPN Support for Standalone Applications

MDAnet users are running a set of distributed commercial off-the-shelf (COTS) applications which are part of a Collaborative Unified Environment (CUE) and, which are running over a secure IP network with links of various speeds in the T1 to T3 range. These applications have varying QoS requirements. To meet these QoS requirements, control over the allocation of network bandwidth between the different CUE applications is desired. The AA-VPN provides an "always-on" QoS management capability that will be used to ensure that the high value traffic from the CUE, particularly the Video Teleconferencing data has preferential access to the T-1 lines connecting the CUE workstations to the MDAnet.

MDAnet applications operate without any knowledge of the SL-QMS. Users initiate a series of QoS requests on behalf of the applications by running the Signaling Client. The Signaling Client is a user level program that interfaces with the Services Manager (SM) via the SL-QMS API. In the current SL-QMS implementation (release 2.0) the SM interfaces with the PEP that resides on a dedicated hardware platform that acts as a packet marking router.

.

In the WBII project, the SL-QMS was used to support a distributed hardware in the loop simulation over a cross country ATM link. In the WBT project, a feasability demonstration, showing support for collaborative engineering applications running over the MDAnet is slated for the near future.