Technical Report

Report on the Quality of Service Translations (QoST) Project

Rameshbabu Prabagaran and Joseph Evans

ITTC-FY2001-TR-15661-01

January 2001

Project Sponsor:
Sprint Corporation

QoS Translations – Executive Summary

Value to Sprint

- Experimental results to determine the advantages and disadvantages of MPLS VPNs in the wide area using the Sprint backbone were obtained.
- Experimental results to evaluate the effects of various QoS technologies on diverse link layer technologies were obtained.
- An analysis of design and implementation issues in relation to an MPLS infrastructure in the wide area was performed.

Lessons Learned

MPLS

- MPLS integrates L2 switching and L3 forwarding to provide a true peer model, and can replace traditional IP over ATM for efficient delivery of a variety of services.
- MPLS requires a router switch combination at every node, but can be incorporated into an existing backbone with minimal changes.
- MPLS helps to solve the best path versus best hop problem.

CoS and QoS

- Coarse grained CoS is preferred over fine grained QoS due to implementation issues. For example, traditional IP over ATM requires special hardware to provide efficient QoS.
- MPLS is the best fit for CoS since it provides a per-class rather than per flow QoS.

Network management and configuration complexity

- Large configuration files resulting from complex CoS or QoS policies are difficult to configure and complex to maintain.
- Network design and management should accommodate changes to existing infrastructure and also addition of new sites/customers. Existing approaches are limited.

Table of Contents

1. Introduction

Given the dominance of IP in data traffic today and the increasing demands of customers for service guarantees, it is necessary that the network architecture support differentiation of IP services and provide the capability to support various quality requirements. Quality of service is the term used to denote the capability of a network to provide better service to selected network traffic over various technologies like Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet and 802.1 networks, SONET, MPLS and IP-routed networks that may use any or all of these underlying technologies. QoS does not involve increasing the bandwidth of networks to increase operational efficiency, but involves using the available bandwidth in a way that it achieves maximum efficiency for a wide range of applications. With today's Internet injecting eighty thousand prefixes into the core of the network, doing anything besides simple forwarding will incur a lot of processing overhead. Initially, the maxim was to keep the core of the network as simple as possible and push all the complexity to the edges. This gave rise to a multitude of algorithms and technologies that involved classification, queuing and scheduling of packets in different ways to achieve appropriate treatment and queuing behavior. Each link layer technology in addition placed different constraints on the way packet treatment had to be done. ATM was proposed with its own QoS capabilities that proved too strict and hard to be deployed. Integrated services and Differentiated services were two approaches to provide IP Quality of Service. IP had connectionless per packet precedence indicating priority for each packet and ATM had per-connection very strict QoS with traffic classes and traffic parameters and hence there were inherent problems in providing IP QoS over ATM with the overlay model. MPLS was developed to incorporate intelligence into the ATM switches in the core of the network and also enable the switches to participate in IP routing protocols. It also lead to the peer model of IP over ATM and provided a means to map the precedence in IP to the service categories in ATM. Given the plethora of networking and internetworking technologies and service guarantees, a clear understanding of the way in which various components of the QoS model interact is fundamental in providing a reliable and robust solution to today's Internet.

Section 2 gives an overview of the background, objectives and resources available for the work. Sections 3 and 4 explain the experiments and implementations that were carried out to test and evaluate the various work fields in different phases of the project. Section 5 outlines the two implementations that were done on Linux. Section 6 discusses the various QoS deployment issues and section 7 concludes this report.

2. Background

The Sprint QoST and Tag Trial is headed by Sprint TP & I with participation from ITTC, KU and Sprint ATL. The working group aims at evaluating and testing some of the QoS technologies available using Cisco gear and the Sprint backbone.

2.1 Objectives

IP services are increasingly important to both business and residential customers, and Quality of Service (QoS) features for IP will be sought by customers desiring service guarantees. For the network provider, IP QoS features will play a dominant role in the future of data communications offerings by offering a mechanism for differentiation of network service products. Strategies that can facilitate the wide variety of QoS architectures that will be available in the foreseeable future need to be developed and deployed.

The general objectives of the project and working group were to study and prototype QoS in IP networks using various technologies including ATM and native IP schemes. The specific objectives of this work were

1. To create a laboratory in which IP performance and management issues may be addressed.
2. To understand, though implementation and experimentation, the performance of IP network elements under stress when configured for various architectural options.
3. To understand and develop, through implementation and experimentation, methods by which IP and ATM QoS models might be made work in conjunction.
4. To understand the interaction between QoS-controlled networks and uncontrolled networks, as well as the issues in QoS configuration between Autonomous Systems.
5. To understand the management issues involved in the deployment of QoS features.

The initial approaches for the above objectives were focussed upon Cisco's Tag Switching architecture following general MPLS principles.

2.2 Resources

The following hardware were available for initial testing.

At ITTC, KU

1 Cisco 7507 RSP (jake), 2 Cisco 7206 Edge routers (snag and drag), 1 Cisco 12008 GSR (desi), 1 Cisco 8650 BPX (blutto), 4 Linux workstations (qost1-4).

At TP & I and ATL, Burlingame

1 Cisco 7206 Edge router each (kctagrouter, burlingame-tag), 1 Cisco 8650 BPX each, 1 Linux workstation at TP & I (tagtrial-pc), 1 Sun workstation at ATL (CAFine).

2.3 Experiments

The work was carried out in two phases. In the first phase, a laboratory for testing various Tag switching functionalities on Cisco Gear was created and TDP operation, re-routing and traffic engineering were tested and evaluated. In the second phase, BGP MPLS VPNs, MPLS CoS, Various mapping techniques for mapping IP to ATM QoS and IP to MPLS CoS and MPLS traffic engineering were studied, tested and evaluated. Two implementations namely MPLS traffic engineering and MPLS CoS in multi-VC LBR mode, both on Linux were also done. The following sections describe the design, testing and evaluation of the each of the above mentioned work fields.

3. Phase I experiments

The first phase of Tag Switching Trial studied the fundamental features and mechanisms pertaining to Tag Switching, including Tag Switching over ATM, Tag Distribution Protocol (TDP), explicit routing, and traffic engineering. All of the tests performed in Phase I are in reference to figure 1. Cisco IOS 12.0(5) T was used in the configuration of the same.

3.1 Initial Tests

1. Build the network as shown in the Figure 1. Configure the switches [22], Tag Switch Controllers, Tag Edge Routers and end-user devices appropriately. Use OSPF [9] as the routing protocol in the network core to configure a flat network, and attach the end-user devices statically. Make sure that there are atleast two subnets attached to each tag edge router.

2. Test the connectivity through an ATM cloud with ATM VPs.

3. Assign IP addresses to all interfaces and devices and check that these are being distributed by the IP routing protocol correctly ("show ip route" on any Tag Switch controller or Tag edge device).

4. Verify that TDP is running and has established the correct sets of tags on all interfaces according to the preferred tag encoding technique.

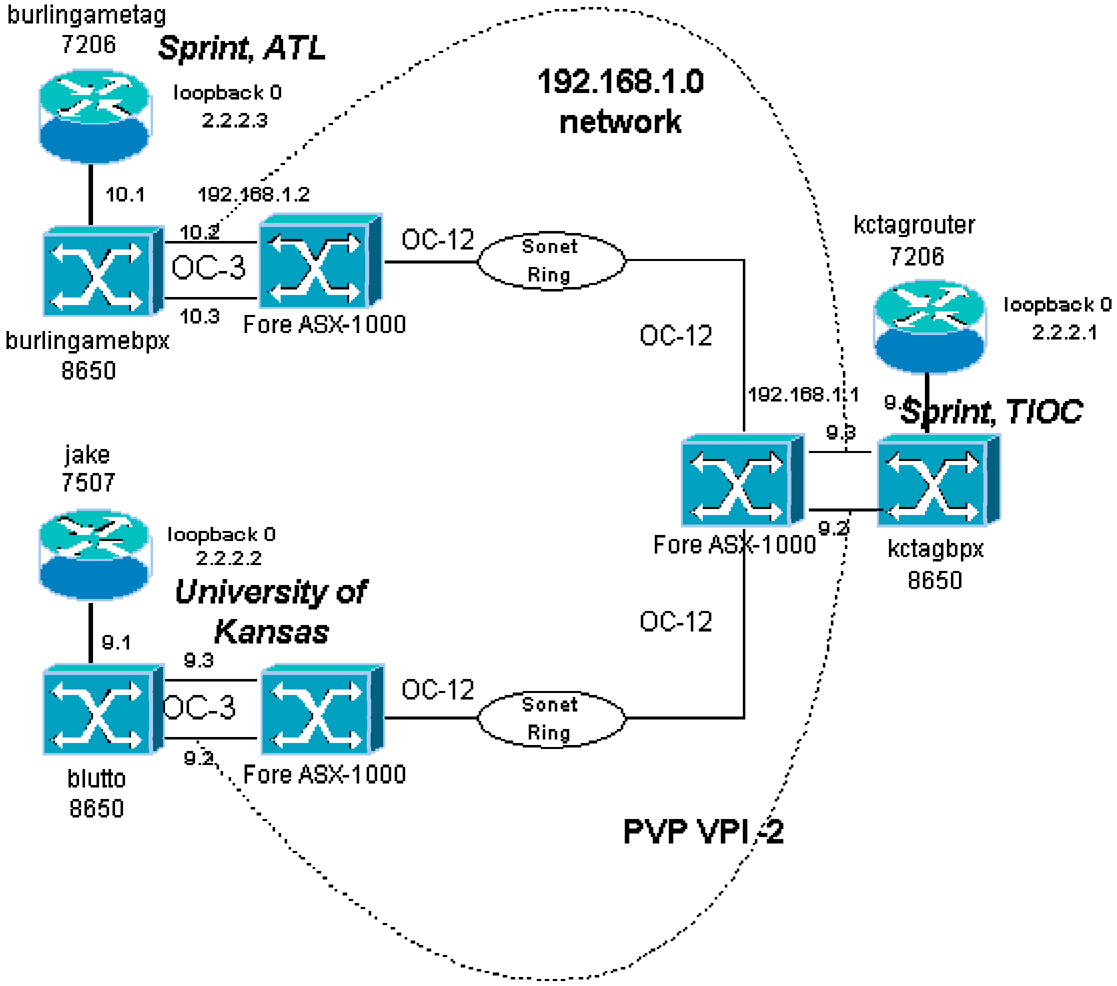

Figure 1. Network diagram for Phase I testing

5. Verify that end-customer devices can send IP traffic to each other. Verify that IP traffic passing through the network passes through the BPX 8650 shelves, but not the Tag Switch Controllers (by monitoring interface statistics).

6. Verify the TTL field is modified correctly through the tag cloud.

7. Check the granularity of the Forwarding Equivalence Class (FEC) and the number of TSPs (Tag Switched Path) in this configuration.

8. Check different possibilities of using different ATM service categories UBR, ABR with zero MCR, etc. when establishing TSPs.

3.2 TDP Protocol Operation

These tests were performed by monitoring TDP trace messages in the routers and Tag Switch Controllers, and/or by use of an appropriately programmed protocol analyzer supplied by Sprint. In this test, the Protocol Information Element Packets have been analyzed and the TDP protocol operation has been verified. The state machine was observed during establishment of the TCP connection between two tag-switching peers. After establishing the connection, the tags were obtained from the peer which was a nexthop router for a set of destinations.

TDP State Machine:

TDP provides the means for TSRs to distribute, request, and release tag-binding information for multiple network layer protocols. TDP also provides means to open, monitor and close TDP sessions and to indicate errors that occur during those sessions. TDP is a two party protocol that requires a connection oriented transport layer with guaranteed sequential delivery. Hence TCP is used as the transport for TDP. Initially the two Tag-Switching Routers (TSR) involved are in Initialized state. TDP_PIE_OPEN is the first Protocol Information Element (PIE) sent by a TSR initiating a TDP session to its peer. It is sent immediately after the TCP connection has been opened. The TSR receiving a TDP_PIE_OPEN responds either with a TDP_PIE_KEEPALIVE or with a TDP_PIE_NOTIFICATION. The state transition diagram of Tag distribution protocol can be found in [26].

3.3 Re-Routing

The Tag-Switched Router (TSR) obtains labels from the downstream next hop TSR for each destination. There might exist multiple paths to the same destination through different next hop TSRs. When an existing Tag-Switched Path (TSP) fails the TSR obtains labels from other downstream next hop TSR. This aspect of re-routing has been tested under this section. The setup that was used to test re-routing of packet on failure of tag-switched path is as given in figure 2.

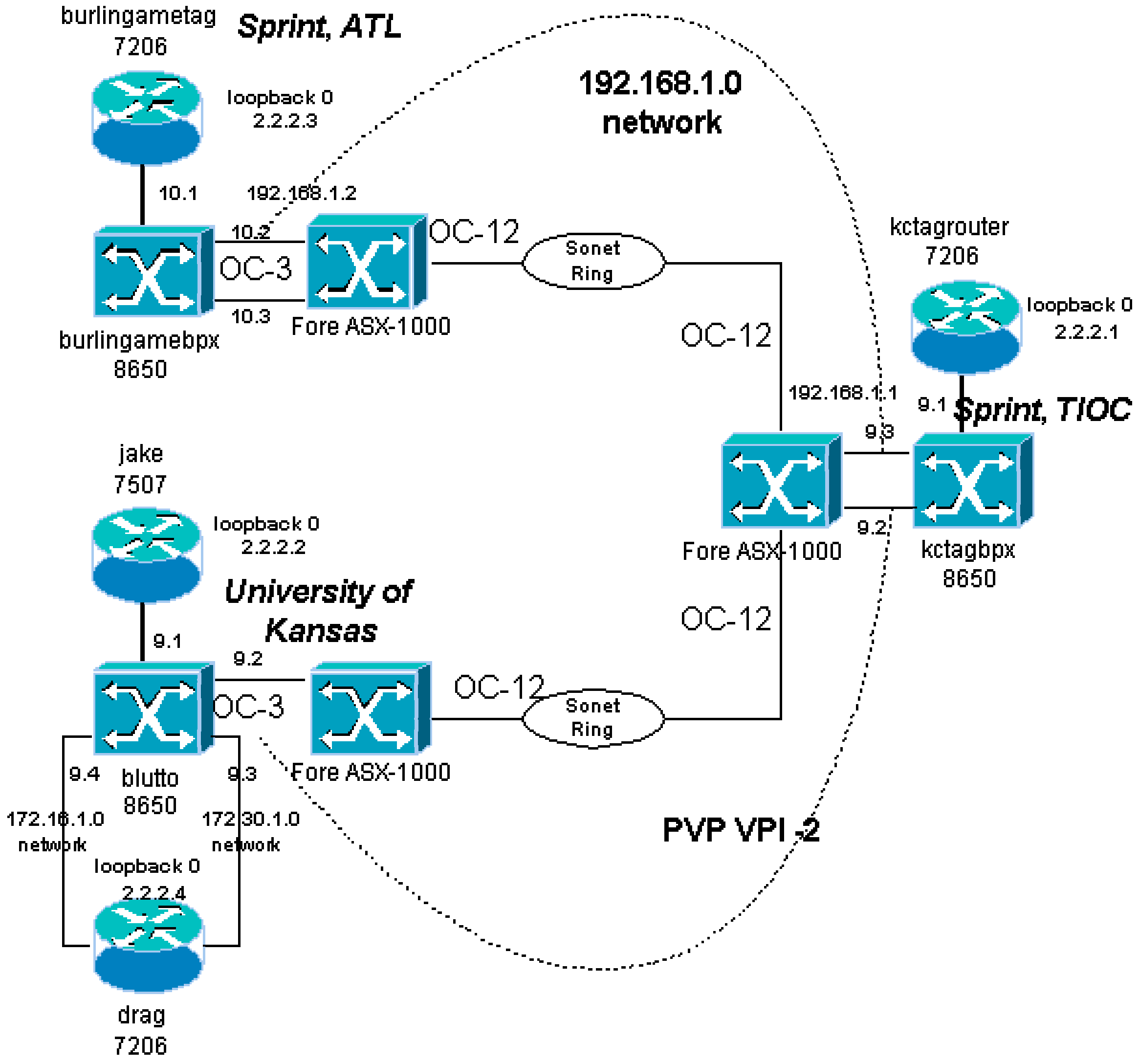

Figure 2. Re-routing setup

As shown above two networks were set up between "drag" (Cisco 7200) and "jake". There were two physical links to reach the destination "burlingame-tag". Re-Routing was tested by bringing down one of the physical links. This was achieved by just shutting down the corresponding interface.

Detailed analysis of data

```
drag#sh tag int
Interface         IP   Tunnel  Operational
ATM1/0.93            Yes   No      Yes         (ATM tagging)
ATM6/0.94            Yes   No      Yes         (ATM tagging)
```

The display shows all the active Tag Interfaces on "drag". As set up two interfaces are active.

```
 drag#sh tag tdp disc
Local TDP Identifier:
   2.2.2.4:0
TDP Discovery Sources:
   Interfaces:
      ATM1/0.93: xmit/recv
         TDP Id: 2.2.2.2:1; IP addr: 172.30.1.1
      ATM6/0.94: xmit/recv
         TDP Id: 2.2.2.2:3; IP addr: 172.15.1.1
```

The display shows the peer neighbors for each interface. The interface address of corresponding peer neighbors has been shown.

```
drag#sh tag tdp neighbor
Peer TDP Ident: 2.2.2.2:3; Local TDP Ident 2.2.2.4:2
        TCP connection: 172.15.1.1.711 - 172.15.1.2.11575
        State: Oper; PIEs sent/rcvd: 108/116; ; Downstream on demand
        Up time: 01:15:33
        TDP discovery sources:
          ATM6/0.94
Peer TDP Ident: 2.2.2.2:1; Local TDP Ident 2.2.2.4:1
        TCP connection: 172.30.1.1.711 - 172.30.1.2.11598
        State: Oper; PIEs sent/rcvd: 20/20; ; Downstream on demand
        Up time: 00:12:38
        TDP discovery sources:
          ATM1/0.93
```

A more detailed information relating to Tag-Switching interface has been displayed above.

```
drag#sh atm vc
           VCD /                          Peak Avg/Min Burst
Interface   Name      VPI  VCI  Type  Encaps   Kbps   Kbps  Cells Sts
1/0.93      10          4   32  PVC   SNAP    155000 155000        UP
1/0.93      40          4   34  TVC   MUX     155000 155000        UP
1/0.93      39          4   36  TVC   MUX     155000 155000        UP
1/0.93      42          4   38  TVC   MUX     155000 155000        UP
1/0.93      41          4   40  TVC   MUX     155000 155000        UP
6/0.94      2           5   32  PVC   SNAP    155000 155000        UP
6/0.94      4           5   34  TVC   MUX     155000 155000        UP
6/0.94      3           5   36  TVC   MUX     155000 155000        UP
6/0.94      6           5   38  TVC   MUX     155000 155000        UP
6/0.94      5           5   40  TVC   MUX     155000 155000        UP
```

The display shows the various VC's set up between "jake" and "drag".

```
drag#sh tag atm-tdp bindings
 Destination: 2.2.2.1/32
   Headend Router ATM1/0.93 (2 hops) 4/34  Active, VCD=40
   Headend Router ATM6/0.94 (2 hops) 5/34  Active, VCD=4
 Destination: 2.2.2.2/32
   Headend Router ATM1/0.93 (1 hop) 4/36  Active, VCD=39
   Headend Router ATM6/0.94 (1 hop) 5/36  Active, VCD=3
```

```
Destination: 2.2.2.3/32
  Headend Router ATM1/0.93 (3 hops) 4/38  Active, VCD=42
  Headend Router ATM6/0.94 (3 hops) 5/38  Active, VCD=6
Destination: 192.68.0.0/16
  Headend Router ATM1/0.93 (2 hops) 4/40  Active, VCD=41
  Headend Router ATM6/0.94 (2 hops) 5/40  Active, VCD=5
```

The above display shows the Tag VC's that exist between "drag" and "jake". For each destination "drag" has obtained a set of two tags, one through each physical link.

```
drag#sh tag for
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
26     5/62        2.2.2.1/32       0          AT6/0.94   point2point
       4/34        2.2.2.1/32       0          AT1/0.93   point2point
27     5/36        2.2.2.2/32       0          AT6/0.94   point2point
       4/36        2.2.2.2/32       0          AT1/0.93   point2point
28     5/64        2.2.2.3/32       0          AT6/0.94   point2point
       4/38        2.2.2.3/32       0          AT1/0.93   point2point
29     5/66        192.68.0.0/16    0          AT6/0.94   point2point
       4/40        192.68.0.0/16    0          AT1/0.93   point2point
```

The forwarding table displays tags that will be used to forward the packet to the destinations. Since the router debug messages did not display all information relating to the packet contents, transmission and reception, HP Internet Advisor (A WAN measurement tool) was used to analyze packets at the ATM level. This tool sniffs the data flowing in ATM link at the same link rate and provides the required statistics. When an ATM cell with an End of Message flag is encountered it decodes the content of packet upto TCP level. The following data has been collected in a similar way. In the above set up, the instrument acted as a sniffer between the link from BPX switch "blutto" to ForeASX 1000 switch. Using the command "show xtagATM cross-connect" on jake VC's connecting from one interface to other interface can be observed as shown below.

```
Phys Desc   VPI/VCI   Type   X-Phys Desc  X-VPI/VCI   State
9.3.0       4/38      ->     9.2.0        2/54        UP
9.3.0       4/40      ->     9.2.0        2/56        UP
9.3.0       4/34      ->     9.2.0        2/52        UP
9.3.0       4/36      ->     9.1.0        2/153       UP
9.3.0       4/32      <->    9.1.0        2/139       UP
9.4.0       5/64      ->     9.2.0        2/42        UP
9.4.0       5/66      ->     9.2.0        2/44        UP
9.4.0       5/62      ->     9.2.0        2/40        UP
9.4.0       5/36      ->     9.1.0        2/125       UP
9.4.0       5/32      <->    9.1.0        2/122       UP
```

From the above display it can be seen that 4/38 on XTagATM94 (Extended ATM Interface on "jake" connected to "drag") is connected to 2/54 on XTagATM92 which connects to destination "2.2.2.3". Thus if a "ping" operation is performed on drag packets flow through 4/38 on 9.4 trunk and switch over to 2/54 on 9.2. Similarly it can be observed that 5/64 on XTagATM94 is connected to 2/42 on 9.2 trunk.

```
    Phys Desc    VPI/VCI    Type  X-Phys Desc  X-VPI/VCI   State
    9.2.0       2/54        ->    9.3.0       3/61       UP
    9.2.0       2/42        ->    9.3.0       3/57       UP
```

The above display is from a similar show command at the "kctagrouter" which shows that the two incoming VC's from drag are switched over to 9.3 trunk connected to "burlingame-tag".

```
    28    5/64       2.2.2.3/32       0         AT6/0.94   point2point
          4/38       2.2.2.3/32       0         AT1/0.93   point2point
```

The above display is a part of Tag-Switching Forwarding table showing tags established to destination "2.2.2.3". To test Re-routing, interface ATM 1/0.93 was shutdown during a prolonged "ping".

```
Summary of: Record #1517 (P2) Captured on 08.30.99 at 11:04:37.6797405,
          ATM: VPI.VCI, 2.54; CLP = High; PTI = SDU Type 0; HEC = Good,
        AAL-5: Type, Not EOM,
    Summary of: Record #1518 (P2) Captured on 08.30.99 at 11:04:37.6797432,
          ATM: VPI.VCI, 2.54; CLP = High; PTI = SDU Type 1; HEC = Good,
        AAL-5: Type, EOM; UU = 0x00; CPI = 0x00; Length = 1004 ; CRC-32 = Good
         ICMP: echo_request,
    Summary of: Record #1519 (P2) Captured on 08.30.99 at 11:04:37.8396596,
          ATM: VPI.VCI, 2.42; CLP = High; PTI = SDU Type 0; HEC = Good,
        AAL-5: Type, Not EOM,
    Summary of: Record #1520 (P2) Captured on 08.30.99 at 11:04:37.8396629,
          ATM: VPI.VCI, 2.42; CLP = High; PTI = SDU Type 0; HEC = Good,
        AAL-5: Type ,Not EOM,
```

The data displayed above shows that when the interface ATM 1/0.93 on "drag" goes down packets are switched from one Tag switched Path to another TSP on ATM 6/0.94. The approximate time of switching was determined to be 160 msec. After the TSP comes back, the data is again switched back to the original TSP.

3.4 Traffic Engineering

Traffic Engineering (TE) is concerned with performance optimization of operational networks. In general, it encompasses the application of technology and scientific principles to the measurement, modeling, characterization, and control of Internet traffic, and the application of such knowledge and techniques to achieve specific performance objectives. The aspects of Traffic Engineering that are of interest concerning Tag-Switching are measurement and control. Congestion problems resulting from inefficient resource allocation can be addressed through Traffic Engineering. In general, congestion can be reduced by adopting load-balancing policies. The objective of such strategies is to minimize maximum congestion or alternatively to minimize maximum resource utilization, through efficient resource allocation. When congestion is minimized through efficient resource allocation, packet loss decreases, transit delay decreases, and aggregate throughput increases. Thereby, the perception of network service quality experienced by end users becomes significantly enhanced. Tag Switching is strategically significant for Traffic Engineering [24] because of the following factors:

(1) Explicit tag switched paths which are not constrained by the destination based forwarding paradigm can be easily created through manual administrative action or through automated action by the underlying protocols, (2) TSPs can potentially be efficiently maintained, (3) Traffic trunks can be instantiated and mapped onto TSPs, (4) A set of attributes can be associated with traffic trunks which modulate their behavioral characteristics, (5) A set of attributes can be associated with resources which constrain the placement of TSPs and traffic trunks across them, (6) Tag-Switching allows for both traffic aggregation and disaggregation whereas classical destination only based IP forwarding permits only aggregation, (7) It is relatively easy to integrate a "constraint-based routing" framework with Tag-Switching. The traffic engineering tests are in relation to the network diagram (figure 3). The following steps were performed in router configuration mode to engineer traffic from "jake" to "burlingame-tag".

Step 1: The TSP tunnel signalling support was configured all along the path. i.e. on each interface through which the path is established TSP signalling was enabled.

```
jake(config)# ip cef distributed
jake(config)# tag-switching tsp-tunnels
jake(config)# interface XTagATM93
jake(config-if)# tag-switching tsp-tunnels
jake(config-if)# exit
```

Similar configuration was done on all interfaces along the path.

Step 2: Four TSP tunnels were configured at the headend i.e. "jake". Two were configured through the direct path to "burlingame-tag" and other two with a hop at "kctagrouter".

```
jake(config)# interface tunnel 22231 (1st Tunnel to 2.2.2.3)
jake(config-if)# ip unnumbered XTagATM93
jake(config-if)# tunnel mode tag-switching
jake(config-if)# tunnel tsp-hop 1 172.30.1.2 lasthop
jake(config-if)# exit
jake(config)# interface tunnel 22232 (2nd Tunnel to 2.2.2.3)
jake(config-if)# ip unnumbered Loopback0
jake(config-if)# tunnel mode tag-switching
jake(config-if)# tunnel tsp-hop 1 2.2.2.1 (IP address of XTagATM92 on kctagrouter)
jake(config-if)# tunnel tsp-hop 2 192.168.1.2 lasthop
```

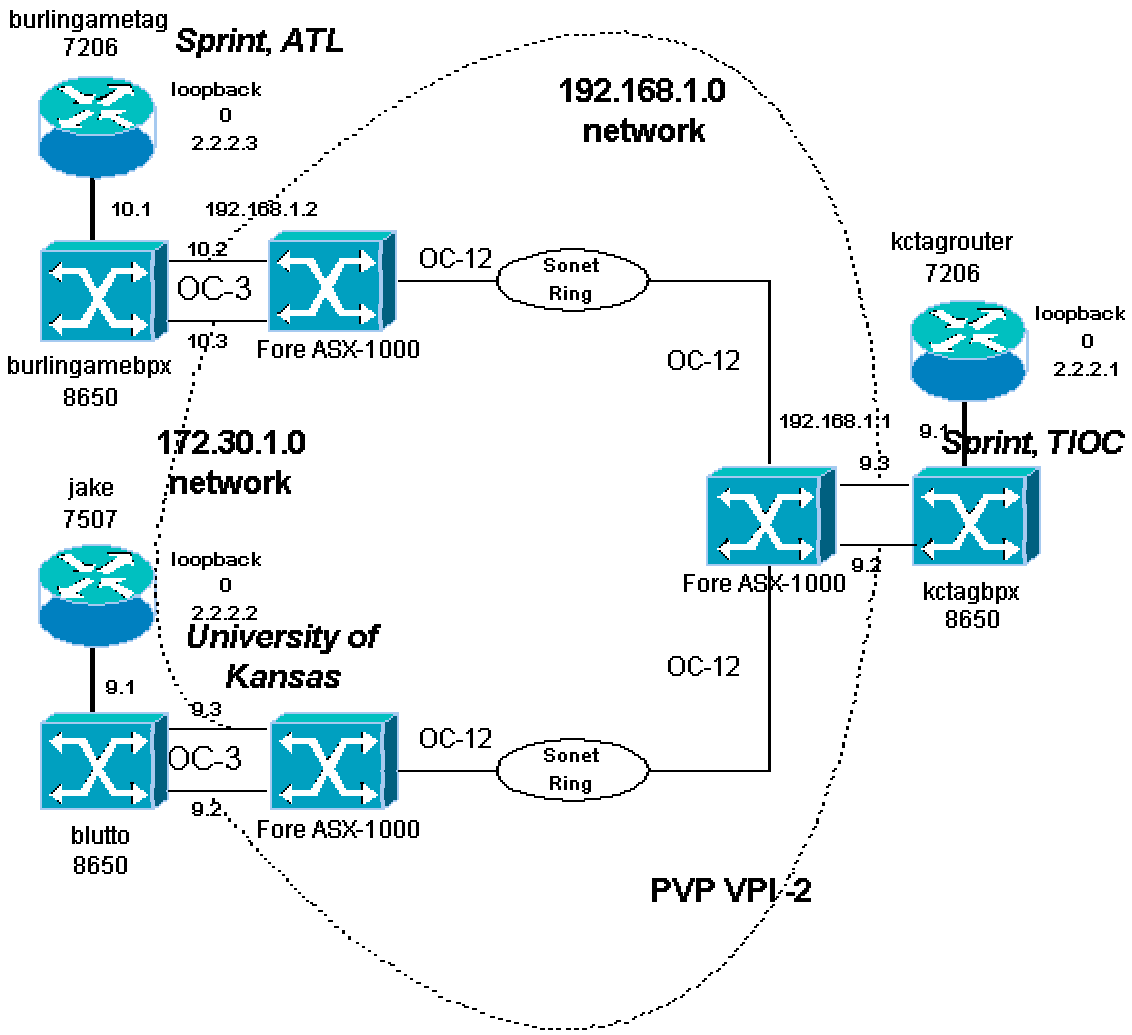

Figure 3. Traffic Engineering test setup

Similarly two more tunnels were configured.

Step 3: A traffic engineering filter was configured to classify the traffic to be routed. The filter selects all traffic where the egress router is 2.2.2.3 (burlingame-tag).

```
jake(config)# router traffic-engineering
jake(config)# traffic-engineering filter 1 egress 2.2.2.3 255.255.255.0
```

Step 4: The traffic engineering route is configured to send the traffic down the tunnel. The tunnel with least preference is selected when sending the traffic.

```
jake(config)# router traffic-engineering
jake(config)# traffic-engineering route 1 Tunnel22231 preference 10
jake(config)# traffic-engineering route 1 Tunnel22232 preference 20

jake#sh tag forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     Untagged    192.168.70.0/24   0          PO4/0/0    point2point
27     2/34        2.2.2.1/32        0          XT92       point2point
28     4/35        172.30.1.2/32     0          XT93       point2point
29     5/33        172.16.1.1/32     0          XT94       point2point
30     Untagged[T] 2.2.2.3/32        0          Tu22233    point2point
31     2/36        192.168.1.0/24    0          XT92       point2point
       4/37        192.168.1.0/24    0          XT93       point2point
32     2/38        192.168.10.0/24   0          XT92       point2point

[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

The above command displays tag-switching forwarding table. The table displays that a traffic-engineered route is associated with destination 2.2.2.3.

```
jake#sh ip traffic-engineering configuration detail
Traffic Engineering Configuration
   Filter 1: egress 2.2.2.3/32, local metric: ospf-10/2
      Tunnel22233 route installed
        interface up, preference 5
        loop check off
      Tunnel22231 route not installed
        interface up, preference 10
        loop check off
      Tunnel22232 route not installed
        interface up, preference 20
        loop check off
      Tunnel22234 route not installed
        interface up, preference 40
        loop check off
```

The configuration details of the tunnels are displayed. The display shows that Tunnel 22233 (3rd tunnel to 2.2.2.3) has the lowest preference number. Hence this tunnel is used to send traffic to 2.2.2.3.

```
jake#sh ip route
        2.0.0.0/32 is subnetted, 3 subnets
C       2.2.2.2 is directly connected, Loopback0
O       2.2.2.3 [has traffic engineered override]
             [110/2] via 172.30.1.2, 2d19h, XTagATM93
S       2.2.2.1 is directly connected, XTagATM92
O    192.168.10.0/24 [110/11] via 2.2.2.1, 2d19h, XTagATM92
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       172.16.1.1/32 is directly connected, XTagATM94
C       172.16.1.0/24 is directly connected, XTagATM94
     172.30.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       172.30.1.2/32 is directly connected, XTagATM93
C       172.30.1.0/24 is directly connected, XTagATM93
```

```
O    192.168.1.0/24 [110/2] via 2.2.2.1, 2d19h, XTagATM92
                    [110/2] via 172.30.1.2, 2d19h, XTagATM93
```

The above routing table displayed shows that there exists a traffic engineering override to the destination 2.2.2.3. Before displaying this routing table the router checks the traffic-engineering routing table for any traffic engineered routes. Since the router debug messages did not display all information relating to the packet contents, transmission and reception, HP Internet Advisor (A WAN measurement tool) was used to analyze packets at the ATM level. The instrument acted as a sniffer between the link from "jake" to BPX switch "blutto". Using the command "show xtagATM cross-connect" on jake VC's connecting from one interface to other interface can be observed as shown below.

```
Phys Desc    VPI/VCI    Type  X-Phys Desc  X-VPI/VCI   State
9.1.0        2/67       ->    9.3.0        4/41        UP
9.1.0        2/66       ->    9.3.0        4/33        UP
9.1.0        2/62       <->    9.4.0        5/32        UP
9.1.0        2/59       ->    9.2.0        2/42        UP
9.1.0        2/58       ->    9.2.0        2/38        UP
9.1.0        2/57       ->    9.2.0        2/34        UP
9.1.0        2/54       <->    9.3.0        4/32        UP
9.1.0        2/53       <->    9.2.0        2/32        UP
9.1.0        2/69       <-    9.4.0        5/40        UP
9.1.0        2/68       <-    9.4.0        5/34        UP
9.1.0        2/65       <-    9.3.0        4/36        UP
9.1.0        2/64       <-    9.3.0        4/34        UP
9.1.0        2/63       <-    9.2.0        2/39        UP
9.1.0        2/56       <-    9.2.0        2/35        UP
9.1.0        2/55       <-    9.2.0        2/33        UP
```

From the above display it can be seen that 2/42 on XTagATM92 is connected to 2/59 on VSI control trunk 9.1 on BPX. Thus if a "ping" operation is performed on jake, its packets flow through 2/59 on 9.1 trunk and switch over to 2/42 on 9.2. Similarly it can be observed that 4/41 on XTagATM93 is connected to 2/67 on 9.1 trunk. Based on the above information, two VC filters were set up to observe the switching the between the two Tag-Switching Tunnels 22233 and 22231. The following data was observed on the Internet Advisor when the Tunnel 22233 was shutdown during a "ping" operation.

```
Summary of: Record #1231 (P1) Captured on 09.20.99 at 05:24:42.9758458
      ATM: VPI.VCI 2.59; CLP = High; PTI = SDU Type 0; HEC = Good
    AAL-5: Type Not EOM

Summary of: Record #1232 (P1) Captured on 09.20.99 at 05:24:42.9758485
      ATM: VPI.VCI 2.59; CLP = High; PTI = SDU Type 1; HEC = Good
    AAL-5: (reassembly complete: 22 cells)Type EOM; UU = 0x00; CPI = 0x00; Length = 1004; CRC-32 = Good
     ICMP: echo_request
 Internet: 172.30.1.1 -> 2.2.2.3 hl: 5 ver: 4 tos: 0 len: 1000 id: 0xc85 fragoff: 0 flags: 00 ttl: 255 prot: ICMP(1)

Summary of: Record #1233 (P1) Captured on 09.20.99 at 05:24:43.0148944
      ATM: VPI.VCI 2.67; CLP = High; PTI = SDU Type 0; HEC = Good
```

```
   AAL-5: Type Not EOM

Summary of: Record #1234 (P1) Captured on 09.20.99 at 05:24:43.0148971
     ATM: VPI.VCI 2.67; CLP = High; PTI = SDU Type 0; HEC = Good
   AAL-5: Type Not EOM
```

Thus it can be concluded that the data is switched from one tunnel to the other. In similar manner, it was observed that data was switched through the path specified by OSPF when all the tunnels were shutdown. Thus TDP protocol operation, re-routing and traffic engineering with Tag-switching were tested as part of the work fields in Phase I.

4. Phase II Experiments

Phase II of the working group focussed on features of MPLS, deployment issues involved in IP to ATM QoS translation and viceversa, deployment issues involved in IP to MPLS CoS translation over ATM and viceversa and implementations on Linux to test various MPLS functionalities. The work items that were tested and evaluated were

(i) BGP MPLS VPNs
(ii) MPLS CoS
(iii) MPLS traffic engineering
(iv) IP to ATM QoS translation and IP to MPLS CoS translation over ATM

Each of the above work item will be discussed in good detail in the following sections.

4.1 BGP MPLS VPNs

The following section gives background information on MPLS and BGP-MPLS VPNs.

Background on MPLS

The forwarding function of a conventional router involves a capacity-demanding procedure that is executed per packet in each router in the network. As line speeds increase, the forwarding function may constitute a bottleneck. This demands more efficient algorithms, data structures, faster processors and memory. MPLS [4] takes another approach by simplifying the forwarding function in the core routers, i.e. by introducing a connection oriented mechanism inside the connectionless IP network. In an MPLS network, a Label Switched Path (LSP) is setup for each route or path through the network. The edge routers

(i) Analyze the header to decide which label switched path to use

(ii) Add a corresponding LSP identifier in the form of a label, to the packet as it is forwarded to the next hop.

Once this is done, all subsequent nodes may simply forward the packet along the label switched path identified by the label in the front of the packet. This enables a connection-oriented mechanism that not only helps in faster forwarding but also helps the switches to take part in routing and forwarding.

MPLS integrates a label-swapping framework with network layer routing. The basic idea involves assigning short fixed length labels to packets at the ingress to an MPLS cloud (based on the concept of forwarding equivalence classes). Throughout the interior of the MPLS domain, the labels attached to packets are used to make forwarding decisions (usually without recourse to the original packet headers). MPLS consists of two components: forwarding and control. The forwarding component uses the labels carried by packets and the Label Information Base (also called LFIB) maintained by a Label Switch Router (LSR) to perform packet forwarding. The control component is responsible for maintaining correct label forwarding information among a group of interconnected label switches. Label Distribution Protocol (LDP) is one of the realizations of the control component. MPLS integrates the performance and traffic management capabilities of Data Link Layer 2 with the scalability and flexibility of Network Layer 3 routing. It is applicable to networks using any Layer 2 switching, but has particular advantages when applied to ATM networks [6]. It integrates IP routing with ATM switching to offer scalable IP over-ATM networks. MPLS has a few advantages over traditional IP over ATM.

(i) In the IP over ATM overlay model, each edge device is just one IP hop away from every other edge device. This is because the switches are just seen as points of crossconnects by the IP routers that take place in routing protocols. As a consequence, the conventional IP routing protocols, RIP, OSPF, ISIS, BGP do not take the number of ATM hops into account while doing a routing table calculation. MPLS unifies this IP/ATM paradigm and makes it possible for ATM switches to take part in routing protocols. This adds intelligence into the ATM switches.

(ii) MPLS supports explicit routing which makes it possible to do traffic engineering to off-load congested links and routers. This is in general referred to as load balancing.

(iii) MPLS can be used as a basis in the construction of Virtual Private Network (VPN)-aware networks using the connectionless IP routing protocol with better scalability and manageability features than the traditional connection-oriented VPN's, e.g., Frame Relay and ATM.

4.1.1 BGP MPLS VPNs

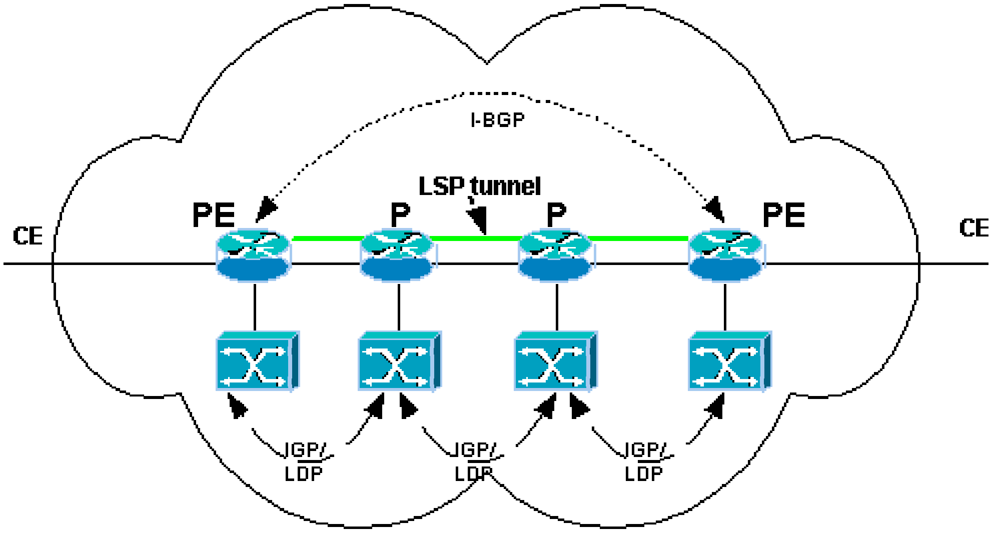

Figure 4. BGP MPLS VPNs

As shown in the Figure, a customer edge router (CE) belonging to a particular VPN populates information about VPN membership to the Provider Edge (PE) router. A PE-to-PE tunnel [1] is established using an MPLS Label Switched Path (LSP). Membership advertisement and route distribution across the MPLS backbone is done using BGP extended communities attributes. Each VPN is assigned a unique identifier called a Route Distinguisher (RD), which is appended to the IP address to form a unique VPN-IPv4 address. Per-VPN forwarding tables are maintained for each node in the VPN. During provisioning, a specific VPN is associated with a specific interface. The RD as well as the use of MPLS labels to route traffic to each site in a VPN allows customers to keep their private addressing schemes, without needing NAT. Two levels of MPLS labels are maintained - the outer one carries the VPN-IPv4 address information between two PE routers and the inner label (top label) is used for label switching at each hop within the network.

4.1.2 MPLS VPN OPERATION

The following points summarize VPN operation.

1. VPN routing/forwarding instances (VRFs) - Each VPN is associated with one or more VRFs. A VRF table defines a VPN at a customer site attached to a PE router. Each VPN is associated with one or more VRFs. A VRF table consists of an IP routing table and a set of interfaces that use the forwarding table (Cisco Express Forwarding) derived from the routing table. It also contains a set of rules and routing protocol

variables that determine what goes into the forwarding table. A given site may belong to one or more VPNs, but can be associated with only one VRF. Packet forwarding information is stored in the IP routing table and the CEF table for each VRF. These tables prevent information from being forwarded outside a VPN, and also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

2. VPN Route Target Communities - Distribution of VPN routing information is done using BGP extended communities [2]. When a VPN route is injected into BGP, it is associated with a list of VPN target communities. This list is typically formed through an export list of extended community-distinguishers associated with the VRF from which the route was learned. Associated with each VRF is an import list of route-target communities. This list defines values to be verified by the VRF table before a route is eligible to be imported into the VPN routing instance.

3. Distribution of routing information - The PE router can learn an IP prefix from a CE router (by static configuration, RIP or BGP). It then generates a VPN-IPv4 prefix by appending an 8 byte RD. The RD is specified by a configuration command on the PE. The VPN-IPv4 addresses are used to distribute network reachability information per VPN via BGP [5].

4. Label forwarding - Based on the routing information stored in each VRF (which includes the IP routing table and the CEF table), MPLS (Cisco Label Switching) uses the VPN-IPv4 addresses to forward packets to the destinations. An MPLS label is associated with each customer route. The label is assigned by the PE. The label forwarding across the testbed network may be based on dynamic IP paths (obtained using OSPF in the present network). As an extension, the paths can be generated based on Traffic Engineering algorithms. Two levels of labels are used for forwarding - one to direct the packet to the correct PE router using VPN-IPv4 addresses, and the second to forward the label through the network.

4.1.3 Testing and evaluation

The test report is in reference to the connectivity diagram shown in figure 5. The tag cloud consisted of Cisco 7200 Routers controlling the BPX 8650 ATM Switches at the three sites - University of Kansas, Sprint TIOC and Sprint ATL. These Tag-Switched Routers were Provider Edge (PE) routers of the tag cloud. Three Virtual Private Networks were set up in this trial. One VPN having members at all three sites was set up and was referred by "QoS VPN". A second VPN referred to as "KU-ATL" VPN had VPN members at the two sites KU and Sprint ATL. The third VPN set up had two members from KU and Sprint TIOC and was referred as "KU-TIOC" VPN. Cisco IOS 12.0(5) T1 was used in the testing.

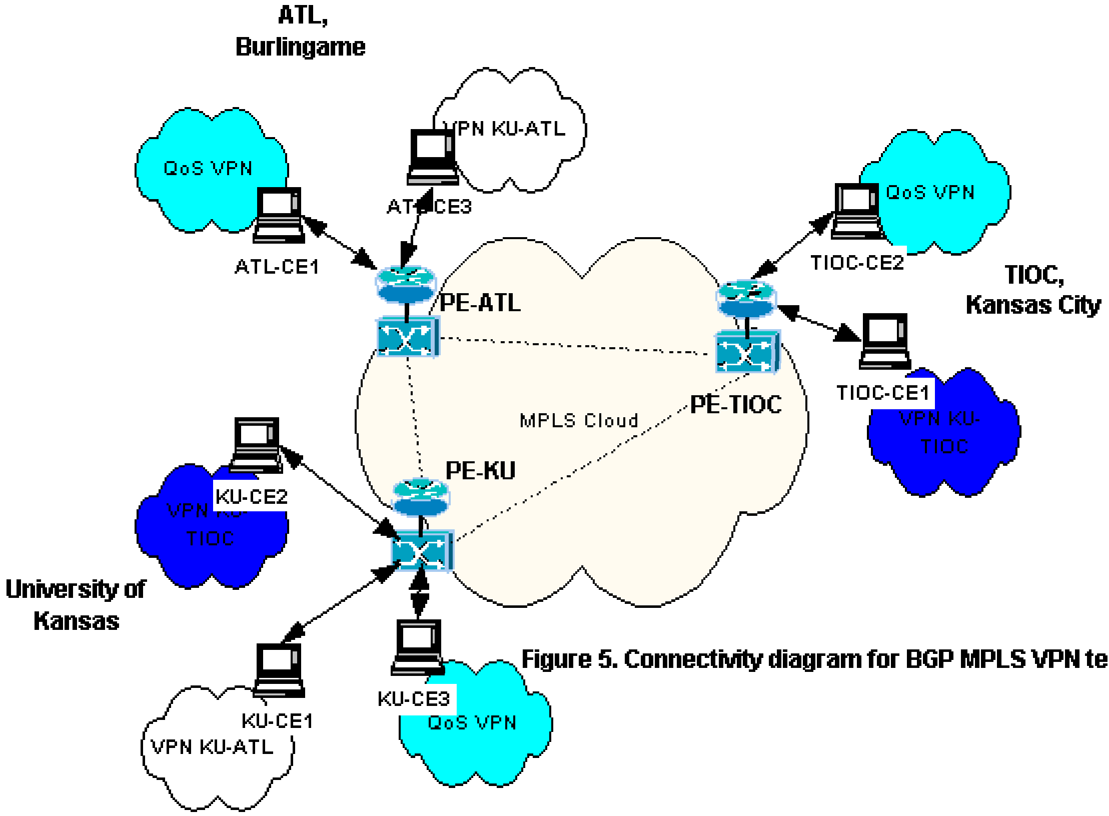

Figure 5. Connectivity diagram for BGP MPLS VPN tests

QoS VPN was first set up and tested. This was followed by setting up of two member VPNs. This was done to test the VPNs within the resources available. The VPN members were PCs and Customer Edge (CE) routers. At ATL a Solaris host (CAFine) was statically connected to PE router while CE router (snag - CISCO 7200) at KU and Linux host (tagtrial-pc) at TIOC were connected using BGP sessions to the PE router. The Linux PC at TIOC used Zebra to establish BGP sessions with Cisco gear. QoS VPN was used to test the QoS capabilities associated with VPNs and constrained distribution of VPN information using BGP. KU-ATL and KU-TIOC VPNs were used to verify if the same site could belong to multiple VPNs along with other common tests.

The following network diagram gives detailed information about the test setup.

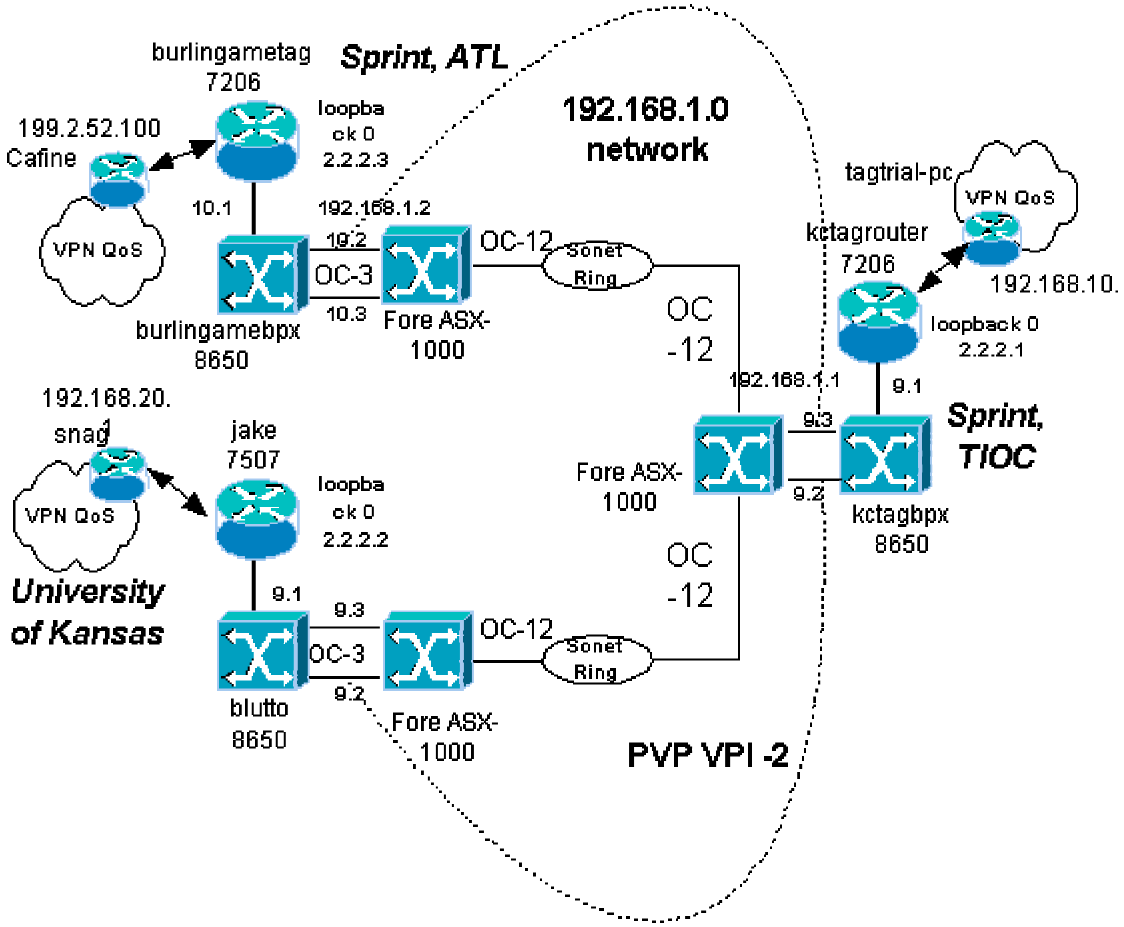

Figure 6. Network diagram for BGP MPLS VPN testing

4.1.3.1 Building VPN's

The following steps were carried out in building the VPNs. The Route Distinguishers (RD) were defined. They are shown in the table below.

No.	Name of the VPN Route Distinguisher	ASN:VPN-ID
1.	QoS VPN	100:10
2.	KU-TIOC VPN	100:30
3.	KU-ATL VPN	100:20

Table 1.Route distinguishers for BGP MPLS VPN testing

The following steps were performed in router configuration mode to set up the VPNs.

Step 1: A Virtual Route/Forwarding (VRF) instance was defined for each VPN. A RD uniquely identifies a VPN. Only routes belonging to a particular VRF is imported or exported.

```
jake(config)#ip vrf qos-vpn
jake(config-vrf)# rd 100:10
jake(config-vrf)# route-target export 100:10
jake(config-vrf)# route-target import 100:10
jake(config-vrf)#end

jake(config)#ip vrf ku-tioc-vpn
jake(config-vrf)# rd 100:30
jake(config-vrf)# route-target export 100:30
jake(config-vrf)# route-target import 100:30
jake(config-vrf)#end

jake(config)#ip vrf ku-atl-vpn
jake(config-vrf)# rd 100:20
jake(config-vrf)# route-target export 100:20
jake(config-vrf)# route-target import 100:20
jake(config-vrf)#end
```

Step 2: The VRF defined above was associated with an interface and an IP address was assigned to it.

```
jake(config)#interface atm1/0/0.100 tag-switching
jake(config-subif)#ip vrf forwarding qos-vpn
jake(config-subif)# ip address 192.168.20.2 255.255.255.0
jake(config-subif)#end
```

Step 3: Provider Edge (PE) router to Provider Edge router static/BGP sessions were established for distribution of VRF information for each VPN.

```
jake(config)#router bgp 20
jake(config-router)# no synchronization
jake(config-router)# no bgp default ipv4-unicast
jake(config-router)# neighbor 2.2.2.1 remote-as 20
jake(config-router)# neighbor 2.2.2.3 remote-as 20
jake(config-router)# address-family vpnv4
jake(config-router-af)# neighbor 2.2.2.1 activate
jake(config-router-af)# neighbor 2.2.2.1 send-community extended
jake(config-router-af)# neighbor 2.2.2.3 activate
jake(config-router-af)# neighbor 2.2.2.3 send-community extended
jake(config-router-af)# exit-address-family
jake(config-router)#end
```

Step 4: PE to Customer Edge (CE) router static/BGP sessions were established for exchange of VPN member reachability information.

```
jake(config)#router bgp 20
jake(config-router)#address-family ipv4 vrf qos-vpn
jake(config-router-af)#redistribute connected
jake(config-router-af)# redistribute static
jake(config-router-af)# no auto-summary
jake(config-router-af)# no synchronization
jake(config-router-af)# exit-address-family
jake(config-router)#end
```

Similar configuration steps were carried out at kctagrouter (TIOC) and burlingame-tag (ATL) routers.

4.1.3.2 Results

The following are some of the results that were got using the test topology as described above.

1. It was observed that only one VRF could be associated with one physical interface. Hence within the resources available, it was only possible to set up QoS VPN and the other two VPNs (KU-ATL and KU-TIOC) independently.

2. Display of a set of defined VRFs and interfaces.

```
jake#show ip vrf interfaces
   Interface          IP-Address          VRF          Protocol
   ATM1/0/0.100          192.168.20.2          qos-vpn        up
```

The above command shows that sub-interface ATM1/0/0.100 has been associated with the qos-vpn VRF and that the sub-interface has an IP address of 192.168.20.2.below.

3. Display of VRF information including Import and Export community lists.

```
jake#show ip vrf detail
   VRF ku-atl-vpn; default RD 100:20
     No interfaces
     Connected addresses are not in global routing table
     Export VPN route-target communities
       RT:100:20
     Import VPN route-target communities
       RT:100:20
     No import route-map
   VRF ku-tioc-vpn; default RD 100:30
     No interfaces
     Connected addresses are not in global routing table
     Export VPN route-target communities
       RT:100:30
     Import VPN route-target communities
       RT:100:30
     No import route-map
   VRF qos-vpn; default RD 100:10
     Interfaces:
       ATM1/0/0.100
     Connected addresses are not in global routing table
     Export VPN route-target communities
       RT:100:10
     Import VPN route-target communities
       RT:100:10
     No import route-map
```

The above 'show' command gives details of the set of defined VRFs, the RDs associated with the VRFs, the interface associated with a particular VRF and the RDs of the VPNs to which and from which routes can be imported and exported respectively. The router maintains a separate routing table for every VRF and this routing table is not a part of the global routing table that is maintained by the router. This can be seen from the above display as 'Connected addresses are not in global routing table'.

4. Display of IP routing table for each VRF.

```
    jake#show ip route vrf qos-vpn
    B    192.168.10.0/24 [200/0] via 2.2.2.1, 16:17:00
    C    192.168.20.0/24 is directly connected, ATM1/0/0.100
    B    199.2.52.0/24 [200/0] via 2.2.2.3, 15:19:00
```

The above command gives the IP routing table associated with a particular VRF. The above display shows three entries corresponding to the three sites that are part of this VPN. The 'B' against the 192.168.10.0 network (TIOC) and 199.2.52.0 network (ATL) denote that these are connected via BGP to this network. The display also shows that the 192.168.20.0 network is directly connected to this router via the atm1/0/0.100 sub-interface can be seen at TIOC and ATL and are shown below.

5. Display of CEF table associated with a VRF.

```
jake#show ip cef vrf qos-vpn
    Prefix              Next Hop            Interface
    0.0.0.0/32          receive
    192.168.10.0/24     2.2.2.1             XTagATM92
    192.168.20.0/24     attached            ATM1/0/0.100
    192.168.20.0/32     receive
    192.168.20.2/32     receive
    192.168.20.255/32   receive
    199.2.52.0/24       2.2.2.3             XTagATM93
    224.0.0.0/24        receive
    255.255.255.255/32  receive
```

The above command (for jake) shows the CEF table associated with the VRF jake. It can be seen above that the 192.168.10.0 network is connected to this router via the XTagATM92 interface with 2.2.2.1 (Kctagrouter) as the next hop. Also, the 199.2.52.0 network can be reached via XTagATM93 interface with 2.2.2.3 (Burlinghametag) as the next hop. The display also shows that network 192.168.20.0 is attached i.e. directly connected via ATM1/0/0.100 sub-interface.

6. Display of VPN-IPv4 information in the Network Layer Reachability Information (NLRI) of the BGP update messages.

```
jake#show ip bgp vpnv4 all
BGP table version is 13, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network          Next Hop            Metric LocPrf Weight Path
   Route Distinguisher: 100:10 (default for vrf qos-vpn)
   *>i192.168.10.0     2.2.2.1                0    100      0 i
   *> 192.168.20.0     0.0.0.0                0         32768 ?
   *>i199.2.52.0       2.2.2.3                0    100      0 i
```

The above command shows the VPN-IPv4 information of the BGP update messages at the three PE routers. Since 192.168.10.0 and 199.2.52.0 VPNs are exported routes for jake, the corresponding entries have an 'i' against them denoting that they are IGP learnt routes. The next hops through which the networks can be reached are also displayed in the same.

7. Display of label forwarding entries that correspond to VRF routes advertised by the router

```
jake#show ip bgp vpnv4 all tags
     Network          Next Hop      In tag/Out tag
     Route Distinguisher: 100:10 (qos-vpn)
     192.168.10.0     2.2.2.1        notag/34
     192.168.20.0     0.0.0.0        27/aggregate(qos-vpn)
     199.2.52.0       2.2.2.3        notag/30
```

The above command displays the VPN IPv4 labels/tags in the label stack. The In tag refers to the incoming tag and out tag refers to the outgoing tag. At jake, since the packet from 2.2.2.1 has no incoming tag in the VPN layer of the stack, the corresponding entry has a 'notag'. However when it has to be forwarded to the VPN member (192.168.20.0) it is given a tag '34'. For the directly connected VPN, the stack will have a VPN-IPv4 label and is marked as In tag '27'.

8. Forwarding:

Two set of labels are used for data forwarding - one (base label) to forward the packet across the MPLS cloud and another (nested label) to carry the data to correct destination site from the edge router. The QoS VPN was used to verify that the nested label is not changed during the packet transfer across the tag-switching cloud. Debug message were enabled on the two routers (jake and kctagrouter) to observe the same. The ping packets were sent from the end host of the QoS VPN. It was verified that the nested label remained unchanged while it was transferred between the PE routers (Figure 7).

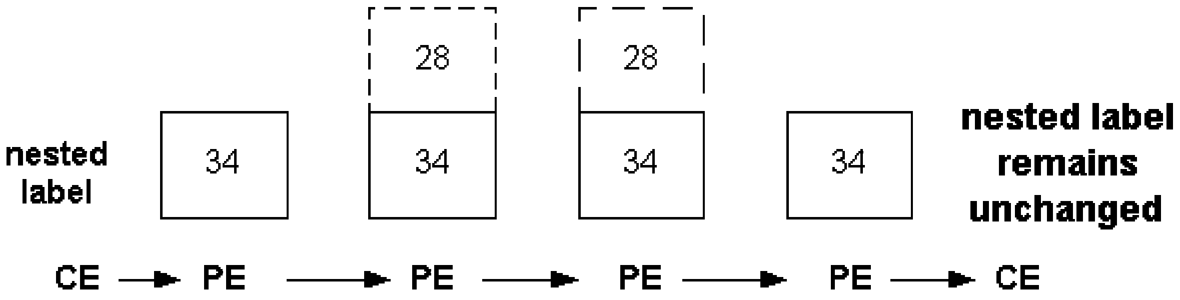

Figure 7. MPLS VPN two level label stack

9. Constrained distribution of Routing information:

In an extranet scenario involving two organizations, distribution of routing information from one organization to the other can be constrained to follow a particular path. This was tested for the KU-ATL VPN in which the distribution of VPN member reachability information was constrained through TIOC. Here TIOC was setup as a Router Reflector [3] and the BGP peering between KU and ATL was removed. A route reflector eliminates the need for having a full-mesh BGP network. Routes from KU and ATL were sent to TIOC and then distributed to the appropriate router. Hence all route exchanges took place through TIOC while data traffic flowed directly from KU to ATL through the direct link.

The configuration changes needed to effect the same at TIOC were

```
   kctagrouter(config)#router bgp 20
   kctagrouter(config-router)#address-family vpnv4
   kctagrouter(config-router-af)#neighbor 2.2.2.2 activate
   kctagrouter(config-router-af)#neighbor 2.2.2.2 route-reflector-client
   kctagrouter(config-router-af)#neighbor 2.2.2.2 send-community extended
   kctagrouter(config-router-af)#neighbor 2.2.2.3 activate
   kctagrouter(config-router-af)#neighbor 2.2.2.3 route-reflector-client
   kctagrouter(config-router-af)#neighbor 2.2.2.3 send-community extended
   kctagrouter(config-router-af)#exit-address-family
   kctagrouter(config-router)#end
```

The BGP neighbor information at KU is as given below.

```
jake#show ip bgp neighbors
   BGP neighbor is 2.2.2.1,  remote AS 20, internal link
    BGP version 4, remote router ID 2.2.2.1
    BGP state = Established, up for 00:13:59
    Last read 00:00:59, hold time is 180, keepalive interval is 60 seconds
    Neighbor capabilities:
      Route refresh: advertised and received
      Address family VPNv4 Unicast: advertised and received
    Received 1296 messages, 0 notifications, 0 in queue
    Sent 1297 messages, 0 notifications, 0 in queue
    Route refresh request: received 0, sent 0
    Minimum time between advertisement runs is 5 seconds
```

```
 For address family: VPNv4 Unicast
  BGP table version 19, neighbor version 19
  Index 1, Offset 0, Mask 0x2
  2 accepted prefixes consume 120 bytes
  Prefix advertised 4, suppressed 0, withdrawn 1
```

It can be seen from the above display that the PE router at ATL and KU establish a BGP session with TIOC. The VPN routes are exchanged between the above peers and router reflector reflects the same to the other peer. Hence this verifies constrained distribution of routing information.

10. Verification of multiple VPN membership of a single site using a single interface.

A site can be a member of multiple VPNs. In typical cases, a site can belong both to an Extranet and an Intranet. However Cisco's implementation of the same allows only one VRF to be associated with an interface/sub-interface. Hence a common host has to connect to a PE router using multiple interfaces if it has to be a member of multiple VPNs. This multiple VPN membership can be obtained over the same physical link using different sub-interfaces. When multiple VRFs are configured on the same interface/sub-interface, the router over-writes the previous VRF entry and accounts for only the latest VRF forwarding entry. The configuration changes needed to effect the same were

```
jake(config)#interface ATM1/0/0.100 tag-switching
jake(config-subif)#ip vrf forwarding qos-vpn
jake(config-subif)#ip address 192.168.20.2 255.255.255.0
jake(config-subif)#no ip directed-broadcast
jake(config-subif)#tag-switching atm control-vc 100 32
jake(config-subif)#tag-switching ip
jake(config-subif)#exit

jake(config)#interface ATM1/0/0.200 tag-switching
jake(config-subif)#ip vrf forwarding qos-vpn
jake(config-subif)#ip address 192.168.30.2 255.255.255.0
jake(config-subif)#no ip directed-broadcast
jake(config-subif)#tag-switching atm control-vc 200 32
jake(config-subif)#tag-switching ip
jake(config-subif)#end
```

Similar configuration was done on snag. The BGP route distribution information and the label forwarding table is as shown below.

```
jake#sh ip bgp vpnv4 all tags
    Network          Next Hop     In tag/Out tag
  Route Distinguisher: 100:10 (qos-vpn)
    129.237.120.0/21 2.2.2.4        notag/29
    192.168.10.0     2.2.2.1       notag/29
    192.168.20.0     0.0.0.0       27/aggregate(qos-vpn)
    192.168.30.0     0.0.0.0       30/aggregate(qos-vpn)
```

```
   199.2.52.0      2.2.2.3        notag/30

 jake#sh ip vrf brief
   Name              Default RD        Interfaces
   ku-atl-vpn         100:20
   ku-tioc-vpn         100:30
   qos-vpn           100:10            ATM1/0/0.100
                                       ATM1/0/0.200
```

Ping packets were sent from the end host at TIOC to the different VPN interfaces and were found to be successful.

11. Verification of VPN membership information dissemination

A CE router can connect to a PE router either by establishing static routing or by BGP. To observe the same, the Solaris PC (CAFine) at ATL was statically connected to the burlingame-tag router, the Linux PC (tagtrial-pc) at TIOC was connected to the router via BGP using 'Zebra' and snag and drag (7200) at KU were connected to jake via BGP. Two VPNs KU-TIOC and KU-ATL were setup between the respective places and connected to the CE routers/hosts as mentioned above. To verify proper membership information dissemination, the following tests were conducted.

(i) Display of set of defined VRFs and interfaces

```
jake#show ip vrf interfaces
    Interface            IP-Address      VRF            Protocol
    XTagATM94               192.168.20.2    ku-atl-vpn      up
    ATM1/0/0.100            192.168.20.2    ku-tioc-vpn     up
```

The above display shows that there are two CE sites attached to the same PE router at KU through different interfaces and that both of them are assigned the same IP address. They however belong to different VPNs.

(ii) Display of IP routing tables for a particular VRF

```
jake#show ip route vrf ku-tioc-vpn
    B    192.168.10.0/24 [200/0] via 2.2.2.1, 00:03:25
    C    192.168.20.0/24 is directly connected, ATM1/0/0.100

jake#show ip route vrf ku-atl-vpn
    C    192.168.20.0/24 is directly connected, XTagATM94
    B    199.2.52.0/24 [200/0] via 2.2.2.3, 00:03:52
```

From the above show messages, it can be seen that the PE routers get information from the connected CE routers and identify the members belonging to a corresponding VPN by importing/exporting routes. Also it can be seen that the routes/updates are received by peer VPN members via BGP.

(iii) Display of VPN-IPv4 information in the NLRI of the BGP update messages.

```
jake#show ip bgp vpnv4 all
   BGP table version is 18, local router ID is 2.2.2.2
Network          Next Hop             Metric LocPrf Weight Path
   Route Distinguisher: 100:20 (default for vrf ku-atl-vpn)
   *> 192.168.20.0     0.0.0.0                 0         32768 ?
   *>i199.2.52.0       2.2.2.3               0     100       0 ?
   Route Distinguisher: 100:30 (default for vrf ku-tioc-vpn)
   *>i192.168.10.0     2.2.2.1                 0     100       0 ?
   *> 192.168.20.0     0.0.0.0                 0         32768 ?
```

(iv) Display of the label forwarding entries that correspond to the VRF routes advertised by the router

```
   kctagrouter#show ip bgp vpnv4 all tags
     Network          Next Hop      In tag/Out tag
   Route Distinguisher: 100:20
     192.168.20.0     2.2.2.2         notag/29
   Route Distinguisher: 100:30 (ku-tioc-vpn)
     192.168.10.0     0.0.0.0         27/aggregate(ku-tioc-vpn)
     192.168.20.0     2.2.2.2         notag/28
```

The VPN-IPv4 labels are displayed above. Though kctagrouter has one peer VPN sites at KU, it reserves tags for the other site, which is not part of the VPN.

(v) Display of global IP routing table.

```
   jake#show ip route
       2.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
   O       2.2.2.3/32 [110/2] via 2.2.2.3, 01:12:57, XTagATM93
   C       2.2.2.0/24 is directly connected, Loopback0
   O       2.2.2.1/32 [110/2] via 2.2.2.1, 01:12:57, XTagATM92
       129.237.0.0/21 is subnetted, 1 subnets
   C       129.237.120.0 is directly connected, Ethernet6/1/0
```

It can be seen from the above routing table that the VPN routes are not part of the global routing table and so any external host cannot reach the members of the VPN unless they are part of the VPN.

(vi) Display of PE-CE BGP information

```
   snag#show ip route
       2.0.0.0/24 is subnetted, 1 subnets
   C       2.2.2.0 is directly connected, Loopback0
   C    192.168.60.0/24 is directly connected, ATM1/0
   S    192.168.10.0/24 [1/0] via 192.168.20.2
   C    192.168.20.0/24 is directly connected, ATM5/0.100
       129.237.0.0/21 is subnetted, 1 subnets
   C    129.237.120.0 is directly connected, Ethernet3/0
   S*      0.0.0.0/0 [1/0] via 129.237.127.254
```

Since snag (CE router) is connected to jake via BGP, all the routes associated with snag is distributed via BGP to jake and this can be observed in jake (below).

```
jake#sh ip bgp vpnv4 all
BGP table version is 31, local router ID is 2.2.2.2
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:20 (default for vrf ku-atl-vpn)
*> 192.168.20.0     0.0.0.0                  0         32768 ?
*>i199.2.52.0       2.2.2.3                0    100      0 ?
Route Distinguisher: 100:30 (default for vrf ku-tioc-vpn)
*>i2.0.0.0          192.168.20.1             0    100      0 ?
*>i129.237.0.0      192.168.20.1              0    100      0 ?
*>i192.168.10.0     2.2.2.1                  0    100      0 ?
*>i192.168.20.0     192.168.20.1              0    100      0 ?
*>i192.168.60.0     192.168.20.1              0    100      0 ?
```

It is seen here that the 2.0.0.0, 129.237.0.0 routes are also distributed hence verifying the distribution of all CE site routes to the PE for a particular VRF. At TIOC, however, a Linux PC was used to establish BGP sessions with the kctagrouter directly using Zebra. The configuration on kctagrouter is similar to the previous case. It was verified that routes learnt from both static and BGP sessions between PE and CE routers are distributed to all members of the VPN. Also it was observed that same IP address can be used in different VPNs as long as it belongs to a different interface.

12. Reliability tests:

An MPLS network is capable of re-routing in a network as a result of a broken or a damaged link. This feature can be tested by providing two paths to the same destination from a host and then testing connectivity by bringing down the active link. Two paths were set up between the end hosts of KU-TIOC, one direct and one through ATL. Initially all traffic went from KU to TIOC through the direct link. Then the interface that connected KU and TIOC was shutdown. The forwarding table given below indicates that both TIOC and ATL are now reached through the XTagATM 93 interface (the interface that connects KU and ATL).

```
jake#show tag-switching forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     4/47        2.2.2.1/32        0          XT93       point2point
27     4/33        2.2.2.3/32        0          XT93       point2point
28     Aggregate   192.168.20.0/24[V] 0
29     Aggregate   192.168.20.0/24[V] 0
30     Untagged    129.237.0.0/16[V] 0          AT1/0/0.100 point2point
31     Untagged    2.0.0.0/8[V]      0          AT1/0/0.100 point2point
32     Untagged    192.168.60.0/24[V] 0         AT1/0/0.100 point2point
33     4/47        192.168.10.0/24[V] 0         XT93       point2point
```

The TDP discovery corresponding to this is given below.

```
jake#show tag-switching tdp discovery
Local TDP Identifier:
    2.2.2.2:0
TDP Discovery Sources:
    Interfaces:
        ATM1/0/0.100: xmit
        XTagATM93: xmit/recv
            TDP Id: 2.2.2.3:2
        XTagATM94: xmit
```

It can be seen that XTagATM92 does not exist and from the forwarding table, it can be observed that all traffic to TIOC is re-routed to ATL. Ping packets from KU to TIOC give us the following statistics

```
jake#ping 2.2.2.1
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/73/76 ms
```

Recovery:

When the link was brought up again, the traffic bound for TIOC from KU was seen to leave through the XTagATM 92 interface. The forwarding table of jake and the ping statistics are now given as

```
jake#show tag-switching forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     2/34        2.2.2.1/32        0          XT92       point2point
27     4/33        2.2.2.3/32        0          XT93       point2point
28     Aggregate   192.168.20.0/24[V] 0
29     Aggregate   192.168.20.0/24[V] 0
30     Untagged    129.237.0.0/16[V] 0          AT1/0/0.100 point2point
31     Untagged    2.0.0.0/8[V]      0          AT1/0/0.100 point2point
32     Untagged    192.168.60.0/24[V] 0         AT1/0/0.100 point2point
33     2/34        192.168.10.0/24[V] 0         XT92       point2point
```

```
jake#show tag-switching tdp discovery
Local TDP Identifier:
    2.2.2.2:0
TDP Discovery Sources:
    Interfaces:
        ATM1/0/0.100: xmit
        XTagATM92: xmit/recv
            TDP Id: 2.2.2.1:1
        XTagATM93: xmit/recv
            TDP Id: 2.2.2.3:2
        XTagATM94: xmit

jake#ping 2.2.2.1
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

```
jake#ping 2.2.2.3
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/38/40 ms
```

The ping time when traffic was re-routed was 73 ms owing to roundtrip from KU to ATL and then from ATL to TIOC. When the link was brought up, the ping time changed to 2 ms. Also the forwarding table now shows that all traffic bound for TIOC goes through XTagATM 92 interface. The above series of tests verify reliability of traffic being re-routed when a link goes down owing to a failure.

13. Scalability tests:

The scalability tests have been carried out based on the number of lines of configuration file that needs to be added/changed. For establishing a new VPN on a PE router, the following number of lines needs to be added.

Per - PE:

Function.	Number of Lines
Defining the new VPN with a VRF	4
Associating the VRF with an interface	2
Establishing PE-PE BGP sessions	3 per PE
Establishing PE-CE static sessions	2

Per CE:

Function.	Number of Lines
Establishing PE-CE static sessions	1
Establishing PE-CE BGP sessions	3

Example:

For introducing 10 Customers with 20 sites each, the number of lines of configuration file needed are

1 line per CE (for BGP) + Normal configuration tasks for connectivity to each of the 20 sites.

```
router(config)#router bgp 25
router(config-router)#address-family ipv4 vrf site1-vpn ! name of vpn
router(config-router-af)#redistribute connected
```

10*(4+2+2)+3 = 83 lines per PE

4 lines for defining new VRF

```
 router(config)#ip vrf site1-vpn
router(config-vrf)#rd 100:30
router(config-vrf)#route-target export 100:30
router(config-vrf)#route-target import 100:30
```

2 lines for associating the VRF with the interface

```
router(config)#interface atm1/0.100 tag-switching
router(config-if)#ip vrf forwarding site1-vpn
router(config-if)#ip address 192.168.10.1
```

3 lines for PE-PE BGP peering & 2 lines for PE-CE BGP sessions

```
router(config)#router bgp 20
router(config-router)#no synchronization
router(config-router)#no bgp default ipv4-unicast ! for PE-PE
router(config-router)#neighbor 2.2.2.2 remote-as 20 ! for PE-PE

router(config-router)#address-family ipv4 vrf ku-tioc-vpn
router(config-router-af)#redistribute connected ! for PE-CE
router(config-router-af)#neighbor 192.168.10.10 remote-as 120 ! for PE-CE
router(config-router-af)#no auto-summary
router(config-router-af)#no synchronization
router(config-router-af)#exit-address-family

router(config-router)#address-family vpnv4
router(config-router-af)#neighbor 2.2.2.2 activate ! for PE-PE
router(config-router-af)#neighbor 2.2.2.2 send-community extended
router(config-router-af)#exit-address-family
```

Except for PE-PE BGP peering commands, all the others have to be repeated per customer and so amounts to 83 lines as described above

4.1.3.3 Observations

For introducing changes, only a few lines associated with the interface needs to be changed. There is a linear increase (O (1)) in the number of lines with the increase in the number of VPNs. There is a linear increase in the number of lines with the increase in the number of sites on the CE router and there is no configuration change required on the PE router if its not a new VPN.

4.1.4 Conclusions

From the experiments and results it was found that

(i) MPLS can be used as a basis in the construction of VPN aware networks using connectionless IP routing protocols with better manageability and scalability than the traditional layer two connection oriented VPNs that involved authentication and authorization overhead.

(ii) MPLS VPNs offer the same level of security that layer two VPNs offer if configured properly.

(iii) MPLS VPNs eliminate routing complexity in the core of the network by enabling only the PEs connected to the VPN sites to take part in routing information dissemination.

(iv) MPLS VPNs offer a very scalable framework for building VPNs by eliminating the need to maintain state and routing information about VPNs at the Provider core routers.

(v) MPLS VPNs also offer address reuse capabilities enabling sites that do not have any common hosts to reuse the address space.

(vi) MPLS VPNs enable constrained distribution of routing information and hence can be extended to the extranet scenario with tremendous ease.

(vii) MPLS VPNs can offer most of the IP services that are offered in the current day Internet.

4.2. QoS in VPNs and MPLS CoS

One of the important requirements for IP based VPNs is, to obtain differentiated and dependable Quality of Service for flows belonging to a VPN. Two performance abstractions are defined as building blocks in the QoS framework. These performance abstractions relate to how a customer would specify or think of the performance requirements of a VPN.

1. Pipe: A pipe provides performance guarantees for traffic between a specific origin and destination pair depending on the service level agreements made with the provider.

2. Hose: A hose provides performance guarantees between an origin and a set of destinations (going into the VPN) and between a node and a set of origins (coming from the VPN).

A hose is characterized by (i) the aggregate traffic from the origin to any of the destination nodes that are part of the VPN, (ii) the aggregate traffic from all the other nodes in the VPN to a particular sink node in the VPN. A hose provides performance guarantees based on such aggregate traffic specifications. These performance abstractions can be managed in the following two ways.

A. Resources are managed on a VPN specific basis. All of the different flows associated with different QoS's within a VPN have their resources allocated from the resources specific to that VPN. **B**. Resources are managed on an individual QoS basis. Thus, the traffic associated with a VPN for a specific QoS would use the share of resources allocated for the QoS.

4.2.1 MPLS Class of Service

The MPLS CoS [8] feature enables network administrators to provide differentiated types of service across an MPLS Switching network. MPLS CoS in Cisco gear offers packet classification, congestion avoidance and congestion management. In order to use MPLS CoS features, the following features are prerequisites

1.CEF switching in every MPLS enabled router
2.MPLS
3.ATM functionality with ATM switches (BPX).
4.Appropriate software and firmware in the associated ATM switch

Since MPLS is both a routing and switching technology, it depends on the layer 2 mechanisms for QoS. At present, MPLS CoS support is provided by underlying ATM technology. Hence ATM functionality is a fundamental requirement for the support of CoS in MPLS.

4.2.2 Advantages of MPLS CoS over native ATM QoS

(i) Point to Point VCs are used in traditional ATM and frame relay networks to implement CoS. Substantial amount of provisioning and management overhead is involved in this QoS support. Compared to this per VC management MPLS offers QoS support with far less complexity by doing per service management.

(ii) In native ATM, ordinary VCs drop cells in over-subscribed classes even when bandwidth is available. In MPLS the class based weighted fair queueing enables efficient bandwidth utilization by borrowing unused bandwidth from one class and allocating it to other classes. MPLS uses pre-defined sets of labels for each service class. A different label is used IP per destination to designate each service class. There can be upto four labels per IP source-destination pair. Using these labels core LSRs implement Class based WFQ to allocate specific amounts of bandwidth and buffer to each service class. Cells are queued by class to implement latency guarantees. The default mapping from ToS to CoS is as shown in table 2.

CoS mapping	ToS
Available	0/4
Premium	1/5
Control	2/6
Standard	3/7

Table 2: IP precedence to ToS mapping

4.2.3 MPLS CoS Operation

Briefly, the following steps are involved in CoS operation.

Step 1: The IP Type of Service (ToS) for a packet is set in the host (or router). The precedence bits define CoS as shown in the table above.

Step 2: One or more labels are copied from the IP ToS to Label CoS in the label header at the label edge router (LER).

Step 3: The packet is queued in the Label Switch Router (LSR) according to its CoS.

Step 4: The MPLS CoS bits are mapped to an ATM label VC in LSR at edge of ATM cloud.

Step 5: Queuing to ATM cells is based on their CoS in the ATM LSR (BPX 8650, for example).

Step 6: At the edge of the ATM cloud, the packet is forwarded with appropriate Label CoS.

Step 7: The labeled packet is received at the LER and after removing the label, it is forwarded with appropriate CoS.

4.2.4 Testing and evaluation

4.2.4.1 Test Setup

The test setup for testing MPLS CoS is as shown in the network diagram (figure 8).

Three hosts and four Label Switched Routers were used in the test. Two routers (jake and kctagrouter) were used as the Label Switch Controllers (7500/7200) controlling the ATM Switches BPX 8650. Netspec was used to generate traffic between the end hosts and measure the throughput. Qost1 was used as the sink while qost2 and qost3 were used as the sources. Cisco IOS 12.0(5) T1 was used on all the routers.

Bandwidth allocation on the BPX was done via an XTagATM interface on the Label Switched Controller. The four classes of service are offered by MPLS by default. They are available, standard, premium and control. The four different classes were given different bandwidths and the throughput was observed.

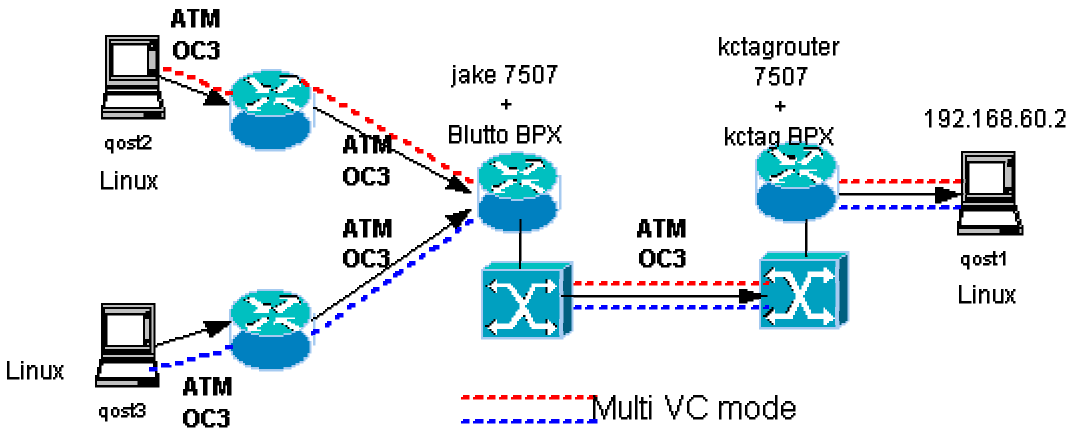

Figure 8. Network diagram for MPLS CoS testing.

4.2.4.2 QoS test results

Destination = qost1 (TIOC); Sources = qost2 and qost3; Traffic type = UDP

Test	Bandwidth Allocation	Source	Precedence	Throughput Tx (Mbps)	Throughput Rx (Mbps)
1	100 %	qost2	0	133.851	22.879
	available	qost3	0	133.834	23.284
2	100 %	qost2	2	133.827	15.446
	available	qost3	3	133.833	29.261
3	100%	qost2	2	133.841	19.693
	control	qost3	3	133.852	27.550
4	100%	qost2	2	133.845	32.254
	premium	qost3	3	133.848	11.744
5	50% available	qost2	2	133.847	22.934
	50% premium	qost3	3	133.836	23.583

4.2.4.3 Observations

1. Class based WFQ and WEPD are enabled by default for MPLS CoS. The Tag-switching interfaces do not explicitly support WFQ.

2. Low UDP throughputs were due to the buffering at the routers and the capabilities of the 7200s.

3. It has been observed that any bandwidth allocation whose total exceeds 100% results in a software crash of the router.

4. Disabling an operational tag-switching interface on a tag-switched controller makes it incapable of tag switching if it is re-enabled. To restore tag-switching operation, the BPX VSI shelf and interface resources should be reset.

4.2.5 Conclusions

MPLS CoS provides only relative QoS support for various service classes. Cisco IOS does not rigidly allocate bandwidth to the service classes. This can be observed from the tests 3 and 4 where there has been 100% bandwidth allocation to control and premium classes respectively.

4.3 IP to ATM and IP to MPLS CoS translation (over ATM)

IP CoS mechanisms are built upon the connectionless per packet precedence paradigm called priority. The ToS byte in the IP header is usually an indication of the priority a packet should receive when being forwarded. Since precedence varies from 0 to 7, the number of CoSs is limited in IP. ATM provides a per-connection very strict QoS with its own traffic classes and traffic parameters. Since most of the ISP backbones are ATM, mapping from IP to ATM QoS and vice versa becomes a necessity if a particular packet is to be given appropriate treatment end to end. The following section identifies the QoS techniques on ATM, IP and MPLS systems and then describes the mapping techniques between them.

4.3.1 QoS in IP systems

Many approaches have been suggested for providing QoS in the Internet Protocol. Some of the approaches are ToS routing [10], Integrated Services [14] and Differentiated services [13].

4.3.1.1 ToS routing

IP precedence utilizes the three precedence bits in the IPv4 header's ToS (Type of Service) field to specify class of service for each packet. Traffic can be partitioned in upto six classes of service using IP precedence, two of them being reserved for internal network use. The queueing technologies throughout the network can then use this signal to provide expedited handling.

Features such as policy-based routing and committed access rate (CAR) can be used to set precedence based on extended access-list classification. This allows considerable flexibility for precedence assignment, including assignment by application or user or by destination and source subnet, and so on. Typically this functionality is deployed as close to the edge of the network (or administrative domain) as possible, so that each subsequent network element can provide service based on the determined policy. IP precedence can also be set in the host or network client, and this signaling can be used optionally; however, this can be overridden by policy within the network. IP precedence enables service classes to be established using existing network queuing mechanisms (for example, WFQ or WRED), with no changes to existing applications or complicated network requirements. The routers that support precedence bits need to implement precedence ordered queue service and precedence based congestion control along with a mechanism to select the priority features of the link layer. The precedence bits serve to differentiate between the various traffic flows based on the relative importance of the individual flows. When a router implements precedence ordered queue service, it ensures that a packet with a certain priority is not transmitted until and unless all packets with higher precedence values are transmitted. Similarly the lower layer precedence mapping ensures that the packet priority is maintained at the link level as well. OSPF and ISIS are two protocols that are capable of ToS routing.

4.3.1.2 Integrated services

In the Integrated services (IntServ) model, network resources are apportioned according to an application's QoS request and subject to bandwidth management policy. The most popular protocol that is used in this community is Resource Reservation Protocol (RSVP).

The ReSerVation Protocol (RSVP) is a signaling protocol that provides reservation setup and control to enable the integrated services (IntServ), which is intended to provide the closest thing to circuit emulation on IP networks. RSVP is the most complex of all the QoS technologies, for applications (hosts) and for network elements (routers and switches). As a result, it also represents the biggest departure from standard best-effort IP service and provides the highest level of QoS in terms of service guarantees, granularity of resource allocation and detail of feedback to QoS-enabled applications and users. There are two types of services in the IntServ model

(i) Guaranteed: This comes as close as possible to emulating a dedicated virtual circuit. It provides firm (mathematically provable) bounds on end-to-end queuing delays by combining the parameters from the various network elements in a path, in addition to ensuring bandwidth availability according to the TSpec parameters (IntServ Guaranteed).

(ii) Controlled Load: This is equivalent to best effort service under unloaded conditions. Hence, it is better than best effort, but cannot provide the strictly bounded service that Guaranteed service promises (IntServ Controlled).

Integrated Services use a token-bucket model to characterize its input/output queuing algorithm. A token-bucket is designed to smooth the flow of outgoing traffic, but unlike a leaky-bucket model (which also smoothes the out-flow), the token-bucket model allows for data bursts-higher send rates that last for short periods. Some of the salient characteristics of RSVP are

(i) Reservations are "soft" and hence need to be refreshed periodically by the receivers

(ii) Applications require APIs to specify the flow requirements, initiate the reservation request and receive notification of success or failure.

(iii) RSVP traffic can traverse non-RSVP routers and this creates a weak-link in the QoS chain where the service falls back to best effort.

(iv) RSVP provides the highest level of IP QoS possible allowing an application to request QoS with a high level of granularity and with the best guarantees of service delivery possible. However implementation and deployment involves a lot of overhead and complexity that might not be suitable for many applications.

4.3.1.3 Differentiated services

Differentiated Services (DiffServ) provides a simple and coarse method of classifying services of various applications. DiffServ assumes the existence of a service level agreement (SLA) between networks that share a border. The SLA establishes the policy criteria, and defines the traffic profile. It is expected that traffic will be policed and smoothed at egress points according to the SLA, and any traffic out of profile (i.e. above the upper-bounds of bandwidth usage stated in the SLA) at an ingress point have no guarantees (or may incur extra costs, according to the SLA). The policy criteria used can include time of day, source and destination addresses, transport, and/or port numbers (i.e. application Ids). Basically, any context or traffic content (including headers or data) can be used to apply policy. Differentiated services are intended to provide scaleable service discrimination in the Internet without a need for maintaining per flow state or doing per hop signaling. This approach employs a small set of building blocks from which a variety of services can be built. These services can be either end-to-end or intra domain. Differentiated Services provide a wide range of services through a combination of setting bits in the ToS octet at network edges and administrative boundaries, using those bits to determine how packets are treated by the routers inside the network, and conditioning the marked packets at network boundaries in accordance with the requirements of each service.

DiffServ uses a different format of the earlier ToS Octet to identify classes and/or precedences. Six bits of the eight bits are used to identify a DiffServ Code Point (DSCP) that identifies the class that a packet belongs to and also the drop precedence of the packet. There are two main classes defined by the IETF. Expedited Forwarding (EF) [11] aims at providing a low loss, low latency, low jitter, assured bandwidth and end-to-end service through DS domains. Such a service appears to the endpoints like a point-to-point connection or a "virtual leased line". Assured Forwarding (AF) [12] is a means for a provider DS domain to offer different levels of forwarding assurances for IP packets received from a customer DS domain. This is used in places where a customer wants his packets to be forwarded with a high probability as long as the aggregate traffic from each site does not exceed the subscribed rate.

4.3.1.4 QoS in ATM

The ATM forum has specified certain ATM traffic classes that have a set of defined traffic parameters. The traffic classes are called service categories. The traffic classes are Constant Bit Rate (CBR), Variable Bit Rate - real time (rt-VBR), Variable Bit Rate - non-real time (nrt-VBR), Available Bit Rate (ABR) and Unspecified Bit Rate (UBR). The traffic parameters that are considered are Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), Burst Tolerance (BT), Minimum Cell Rate (MCR) and Cell Delay Variation Tolerance (CDVT). The details regarding the ATM traffic classes and ATM traffic parameters can be found in the ATM forum's traffic management specification.

4.3.1.5 QoS in MPLS - This has been discussed in section 4.2.1.

4.2.1.6 QoS techniques available on different systems

This section discusses the various QoS techniques that are available in the three technologies IP, ATM and MPLS.

4.2.1.6.1 IP

The differentiated services model described above is the popular model for providing IP Class of Service. The key components or elements of this model is as shown in the following figure.

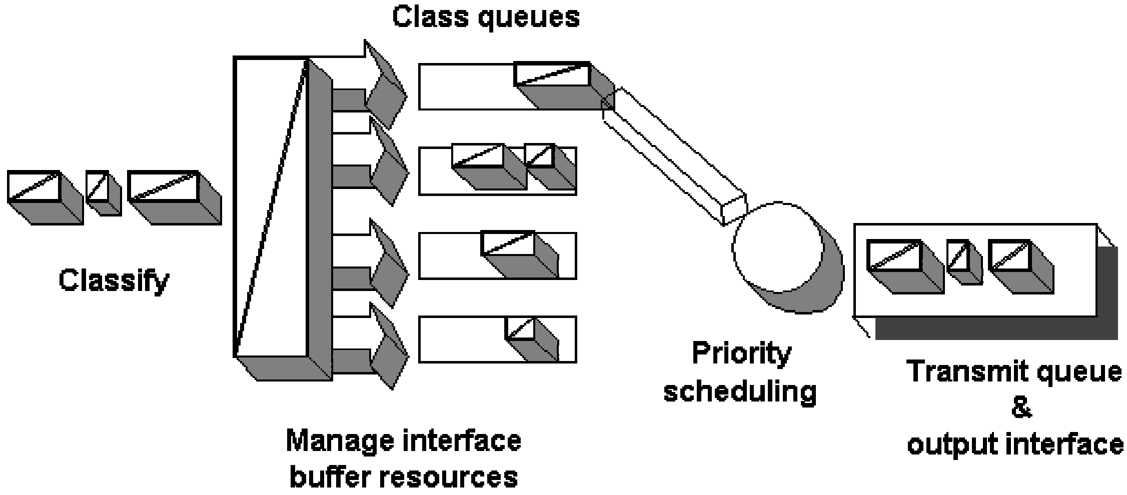
Class queues
Classify
Priority scheduling
Transmit queue & output interface
Manage interface buffer resources

Figure 9. Key components of the IP CoS model

It can be seen that the Traffic Conditioning (TC) is done before packets are put on the output line. IP CoS is provided by the following features.

Classification - Classification is done so that packets can be placed in appropriate queues before being scheduled. Classification is a means of managing bandwidth at the edge of a network. Some of the classifiers that are available on Cisco gear are Committed Access Rate (CAR), QoS policy propagation via BGP (QPPB) and Policy based routing.

Congestion management and avoidance - Congestions in the network are a primary bottleneck to achieve desired throughput. Queueing and scheduling are usually done to manage congestion. Some of the congestion management techniques are Weighted Fair Queueing (WFQ), Priority Queueing (PQ) and Class Based Queueing (CBQ). Congestion avoidance is generally done for TCP traffic using Random Early Detect (RED) [16, 17]. Weighted RED (WRED) is a mechanism by which a router can assign weights to the different traffic classes so that packets can be dropped based on class when congestion manifests itself.

Policing and Shaping - A shaper typically delays excess traffic using a buffer, or queueing mechanism, to hold packets and shape the flow when the data rate of the source is higher than expected. Thus, traffic adhering to a particular profile can be shaped to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches. Generic Traffic Shaping (GTS) and Frame Relay Traffic Shaping (FRTS) are the shapers supported on Cisco gear.

4.2.1.6.2 ATM

ATM QoS is provided by combining some of the above mentioned operations (on IP) on Virtual Circuits. The QoS techniques available on ATM are

Per-VC WRED - WRED is configured on a per-VC basis. Thus packets can be dropped inside a VC based on its precedence when multiple streams are multiplexed onto a single VC.

Class Based WFQ (CBWFQ) and per-VC WFQ - CBWFQ is similar to WFQ but is per class. Per-VC WFQ is again WFQ done on a VC by VC basis. The disadvantage of per-VC WFQ is the cost of maintaining per-VC queues in an environment with a reasonably large number of VCs.

Early Packet Discard (EPD) and Partial Packet Discard (PPD) - When ATM cells are discarded in an ATM network, its upto the higher layers to request for re-transmission. Hence, the loss of a single cell can waste a huge amount of bandwidth. To prevent this, PPD drops the rest of the AAL packet when a cell is to be discarded and when the buffer becomes full. This has advantages of saving bandwidth and increasing throughput. EPD on the other hand drops a complete AAL packet when the buffer occupancy threshold is reached.

4.2.1.6.3 MPLS

Since MPLS is both a routing and switching technology, it depends on the layer 2 mechanisms for QoS. At present, MPLS CoS support is provided by underlying ATM technology. Hence ATM functionality is a fundamental requirement for the support of CoS in MPLS. MPLS CoS support is provided by the following features.

IP Precedence - This feature uses three bits in the IP header to indicate the service class of a packet (up to eight classes). This value is set at the edge and enforced in the core. In IP+ATM networks, different labels are used to indicate precedence levels.

Committed Access Rate (CAR) - CAR manages bandwidth allocation for certain traffic types. CAR uses the type of service (ToS) bits in the IP header to classify packets according to input and output transmission rates. CAR is often configured on interfaces at the edge of a network in order to control traffic into or out of the network.

Class-Based Weighted Fair Queuing (CBWFQ) - This feature provides the ability to reorder packets and control latency at the edge and in the core. By assigning different weights to different service classes, a switch manages buffering and bandwidth for each service class. Because weights are relative and not absolute, under utilized resources are shared between service classes for optimal bandwidth efficiency.

Weighted Early Packet Discard (WEPD) - WEPD drops packets intelligently when congestion occurs. Packets are scheduled by class during congestion.

4.3.2 IP to ATM CoS mapping

In Phase II of the trial, study and analysis of IP to ATM CoS mapping was done. The different mapping techniques were analyzed and the performance, design and scalability issues were evaluated. The following section describes the work of the working group in relation to the same. No experiments were conducted due to resource constraints and the non-availability of the PA-A3 adapters on Cisco gear. IP to ATM CoS feature provides a complete working solution to class-based services without the investment of a new ATM infrastructure. This enables provision of `differentiated services' across the entire wide-area network and not just the routed portion alone.

Features:

The IP to ATM CoS mapping can be of two types.

(i) Mapping to a single VC and providing differentiated service within the VC.

(ii) Mapping to a VC bundle and providing differentiated services within the bundle.

Cisco routers (7200s and 7500s) provide this functionality using the enhanced ATM PA-A3 port adapters. It is fundamental to understand the features available on the PA-A3 and the collaboration between the PA-A3 and the processor for proper deployment of the IP to ATM CoS functionality. Some of the features that the PA-A3 provides relevant to the IP to ATM CoS mapping are given below

- Traffic shaping and rich ATM service category support
- Per-VC queueing, per-VC WRED and per-VC backpressure
- Flexible VC bundle management

The processor in the router and the PA-A3 collaborate in the following way to provide proper and stable IP to ATM CoS functionality

- The PA-A3 transmits ATM cells on each ATM PVC according to the ATM shaping rate.

- The PA-A3 maintains a per-VC first-in, first-out (FIFO) queue for each VC where it stores the packets waiting for transmission onto that VC.

- The PA-A3 provides explicit back pressure to the processor so that the processor only transmits packets to the PA when the PA has sufficient buffers available to store the packets which ensure that the PA-A3 will never need to discard any packets regardless of the level of congestion on the ATM PVC.

- The PA-A3 provides backpressure both at the VC level and at the aggregate all-VC level.

The processor then monitors the level of congestion independently on each of its per-VC queues and performs a WRED selective congestion avoidance algorithm independently on each of these queues that enforces service differentiation across the IP CoS.

4.3.2.1 Single ATM-VC Support

IP traffic having different IP precedence are mapped onto a single VC in this case. Weighted Random Early Detection (WRED) is used to classify packets based on the IP precedence and subject them to different drop probabilities and hence different priorities. The following takes place when an IP packet arrives at a router in addition to the normal routing/forwarding process.

- Commited Access Rate (CAR) is used to classify and mark packets depending on the agreement with the domain.

- The packets are stored in the per-VC queue in the PA-A3 till it is full. If the PA-A3 queue is full, it gives backpressure to the processor and the packets are now queued in the processor's queue.

- WRED is used to subject packets to different drop probabilities and hence different priorities. When the link is free for transmission, the packets are sent out.

It is to be noted that the congestion is managed totally at the IP layer using WRED and hence the congestion in one VC does not affect other VCs. Also, congestion management at the IP layer prevents ATM cells from being dropped in the core and hence prevents re-transmission of a complete packet when a single cell is dropped.

4.3.2.2 VC Bundle support

IP traffic having different IP precedence are mapped to multiple VCs in this case. QoS is provided over this entire VC bundle and/or individual VCs. Using VC bundles, differentiated services can be provided by distributing IP precedence levels over different VC bundle members. Two types of mapping are provided in this case

- Map single precedence level to a discrete VC and hence provide individual VCs in the bundle to carry packets with a particular IP precedence.

- Use WRED to differentiate across traffic and hence enable single VCs to carry multiple precedence with different drop probabilities.

The following actions take place in a router configured to provide bundle-VC support in addition to the normal routing/forwarding.

1. To determine the VC to be used to forward a packet to the destination, matching between precedence levels of packets and VCs take place. The bundle management part of the router takes care of matching between a packet's IP precedence and the IP precedence value or range of values assigned to a VC.

2. The packet is now forwarded on that matching VC(s).

3. In case of a VC failure, traffic is re-directed to a previously configured VC.

Multiple parallel VCs allow stronger differentiation at the IP layer hence allowing for rt-CoS and nrt-Cos. Having seen what the features underlying the mapping are, we specifically concentrate on three major issues

4.3.2.3 Performance issues

Performance of a technique is addressed mainly by stability, management & control and throughput. The following are the issues related to performance

- ♦ Stability: Due to explicit per-VC backpressure given by the PA-A3 to the processor the stability of the system does not degrade rapidly (i.e. causing packets to be dropped heavily during times of heavy congestion).

- ♦ Management of VC bundle: In case of a single PVC failure inside a bundle, all traffic destined for that particular VC can be redirected to a previously configured PVC. This prevents the failure of the whole bundle in case a bundle member goes down. This feature also allows policy administration.

- ♦ Congestion: As mentioned previously, the congestion of a single VC does not affect the link as a whole and hence eliminates the problem that was inherent with ATM during congestion. Also discarding packets at the IP level ensures that cells don't get dropped during periods of congestions hence not reducing throughput drastically.

- Bandwidth allocation: WRED is responsible for marking packets with different drop probabilities and so there is no strict bandwidth allocation. However bandwidth guarantee can be given provided the traffic remains well below a certain profile.

- Processing delay: Commited Access Rate claims to perform classification, rate measurement, enforcement and marking at a very high speed and so it does not prove to be a bottleneck to processing at line-speeds.

4.3.2.4 Design issues

Deployment of the IP to ATM CoS feature needs careful design and some of the issues related to design are as follows.

- The feature uses the existing ATM infrastructure hence eliminating the need for a new backbone. Service providers can continue to use their established ATM backbone to provide IP CoS to their customers.

- The feature uses already established software features like CAR and WRED to provide the mapping.

- The main constraint in deploying the feature in the Internet is that the routers have to be equipped with the ATM enhanced PA-A3 port adapters. As mentioned previously, the PA-A3 adapters provide for per-VC queueing, per-VC WRED and traffic shaping on per-VC basis.

- This feature also provides the conventional ATM CBR, rt-VBR, ABR and the ATM analogue of IP's best-effort i.e. UBR. The main advantage of this feature is that congestion is pushed to the edge and all processing/pre-processing takes place at the edge and only packet-treatment is taken care of by the core.

4.3.2.5 Scalability issues

One of the primary problems in the growing Internet is scalability. With multiple VCs running from source to destination, the configuration of the edge and the core routers become a real problem. Some of the issues related to scalability in the IP to ATM CoS feature are as follows.

- For each source-destination pair, a VC or VC-bundle is created with desired features. With a linear increase in the number of source-destination pairs, there is a linear increase in the number of VCs or VC-bundle hence giving rise to scalability problems. A few thousand customers connected to a SP can almost deplete the VC-bundles configurable on a particular link.

- During periods of congestion, the backpressure given by the PA-A3 causes packets to be queued in the processor. This gives rise to increased processor load during periods of heavy congestion. This increase is linear with the increase in size of the processor buffer.

- There is also a constraint placed on the number of VCs that can be managed by a given PA. For each VC that is created on the PA, a buffer is allocated from the buffer pool. This places a limitation on the number of VCs created as mentioned earlier.

- Lines of configuration code.

In single-VC mode, the number of lines of configuration code required on the router for configuring a single PVC are as follows

definition of WRED group = 1
definition of WRED parameters = 1 * number of classes reqd (max = 8)
attaching a WRED group with a PVC = 2
total = 3 + (number of classes required)

In the bundle VC configuration mode, the number of lines of configuration code required on the router for configuring a single VC-bundle are as follows

definition of vc-class = 4 + 'n' optional parameters
definition of a VC-bundle = 1
configuration of individual VCs in the bundle = p (# PVCs)
configuration of parameters in the bundle = p * m (optional params)
attaching a vc-class with the each corresponding PVC = p

total : required = 5 + 2p lines & optional => 5 + n + 2p + p*m

For example, to configure a single PVC with 8 precedence value, we require the following configuration (IOS 12.0(7)T).

```
jake(config)#interface ATM1/0/0.100 multipoint
jake(config-subif)#ip address 192.168.101.1 255.255.255.0
jake(config-subif)#pvc ku 100
jake(config-if-atm-vc)#encapsulation aal5nlpid
jake(config-if-atm-vc)#random-detect attach ku-single-pvc
!
jake(config)#random-detect-group ku-single-pvc
jake(cfg-red-group)#precedence 0 200 1000 10
jake(cfg-red-group)#precedence 1 300 2000 10
jake(cfg-red-group)#precedence 2 400 2000 10
jake(cfg-red-group)#precedence 3 500 2000 10
jake(cfg-red-group)#precedence 4 600 2000 10
jake(cfg-red-group)#precedence 5 700 4000 10
jake(cfg-red-group)#precedence 6 800 4000 10
jake(cfg-red-group)#precedence 7 900 4000 10
!
```

For N PVCs, we require N such WRED groups hence making the configuration unscalable. Thus, it can be seen that there are inherent advantages and disadvantages in deploying this IP to ATM CoS feature. For SPs that already have a well established ATM backbone, this feature will be really helpful in providing differentiated services to customers at the price of scalability.

4.3.3 IP to MPLS CoS mapping over ATM

MPLS was proposed to remove the per-VC overhead and inefficient bandwidth utilization in mapping IP to ATM QoS using the overlay model. In the MPLS peer model, there is IP intelligence at every hop and hence a possible discard and efficient bandwidth utilization at every hop. The IETF has proposed two ways in which IP CoS can be mapped to MPLS CoS.

- In one model, the ToS octet in the IP header is copied onto the EXP field of the MPLS shim header and appropriate packet treatment is given based on the value contained in the EXP field. When spanning multiple domains, either the pipe model or the uniform model can be used consistently to provide appropriate treatment to the packet.

- In another model, LDP signals N labels per precedence per IP source-destination pair as described in section 4.2.3. This model provides treatment to the packet by providing IP treatment to packets at the edge and MPLS over data-link treatment at the core. Since ATM is the preferred data-link layer mechanism, the core implements ATM QoS in the form of per CoS WFQ and per CoS WEPD.

Tests and results

4.3.3.1 Test setup

Tests were conducted to study and evaluate the effects of CoS mapping when a packet traverses diverse domains. Parameters specific to a domain were changed and its effects on traffic characteristics were observed. Since DiffServ support was not available on the Cisco gear, mapping from DiffServ to MPLS CoS could not be tested. The test setup that was used to conduct the tests is as shown in the figure below. Cisco IOS 12.0(7) T was the image used on all the routers for testing.

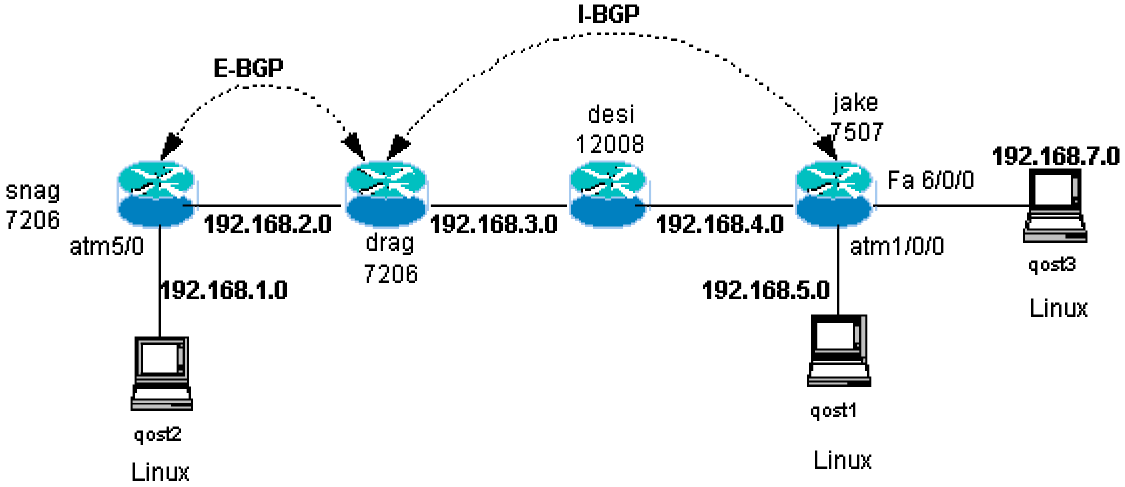

Figure 10. Test setup for testing MPLS to IP CoS translation

The configuration needed at a node (interface) to enable CoS is as given below

```
interface ATM0/0/0
 <standard configuration>
 fair-queue tos
 fair-queue tos 1 weight 20
 fair-queue tos 2 weight 30
 fair-queue tos 3 weight 40
```

The above enables WFQ with weights of 20, 30 and 40 assigned to ToS 1, 2 and 3 respectively.

4.3.3.2 Results

1. Baseline testing - Two full blast streams were generated using Netspec and was sent from qost1 and qost3 respectively to qost2 connected to snag. Also a constant 10 Mbps bursty stream was sent from a host attached to drag to qost2. No CoS feature was configured on any of the routers and the result was as follows

Tx Node	ToS set	Throughput Transmitted Mbps	Throughput Received Mbps
qost1	0	131.711 (UDP)	39.445
qost3	6	95.517 (UDP)	35.230

The low received throughputs are due to buffering capacity of the 7200s (drag and snag) and congestion at jake.

2. CoS on snag alone - Snag was configured with WFQ on its outgoing and incoming interface. Traffic was sent as before through the network.

Tx Node	ToS set	Throughput Transmitted Mbps	Throughput Received Mbps
qost1	0	131.708 (UDP)	39.873
qost3	6	95.539 (UDP)	34.853

The behavior was quite expected because of the normal treatment of packets in the MPLS cloud, i.e. packets are given best effort treatment in that domain and hence CoS treatment at snag does not have any effect.

3. CoS on drag -> snag interface alone - The interface between drag and snag on drag was configured to provide preferential treatment, i.e. WFQ was configured on drag's outgoing interface to snag.

Tx Node	ToS set	Throughput Transmitted Mbps	Throughput Received Mbps
qost1	0	131.718 (UDP)	40.345
qost3	6	95.539 (UDP)	35.542

This behavior was also expected because of the best effort treatment given to packets in the MPLS domain. Similar results were observed with CoS on drag and snag alone.

4. CoS on drag, jake and snag - WFQ was configured on all the routers and the same test was conducted. The MPLS domain was not configured to operate in multi-VC Label Bit Rate (LBR) mode.

Tx Node	ToS set	Throughput Transmitted Mbps	Throughput Received Mbps
qost1	0	131.686 (UDP)	38.107
qost3	6	95.526 (UDP)	41.797

Since CoS treatment is given to packets throughout the network, the stream with higher priority (6) got better throughput than the stream with zero precedence.

5. Effects of varying allotted bandwidths to different classes - WFQ on jake was tuned with different weights for ToS. Here too, the MPLS domain was not configured to operate in multi-VC LBR mode. The results are as given below

Tx Node	ToS set	Throughput Tx Mbps	Throughput Rx Mbps
ToS 1 => 20 %, ToS 2 => 30%, ToS 3 => 40%			
qost1	0	131.725	42.852
qost3	6	95.517	44.315

ToS 1 => 10%, ToS 2 => 10%, ToS 3 => 70%			
qost1	0	131.725	42.832
qost3	6	95.765	44.297
ToS 1 => 0%, ToS 2 => 0%, ToS 3 => 99%			
qost1	0	130.987	42.742
qost3	6	95.653	44.363
ToS 1 => 0%, ToS 2 => 0%, ToS 3 => 1%			
qost1	0	131.733	46.597
qost3	6	95.512	39.840

From the above tests it can be seen that the unused bandwidth allocated to a class is shared by the packets belonging to other classes. There is no strict allocation of bandwidth and hence it is relative as is required for MPLS CoS. Also, it can be seen that allocating 99% of the bandwidth to class 0 does not starve packets belonging to class 3.

6. Tests with bursty traffic in multi-VC LBR mode - The above tests were conducted for full stream traffic. Since the majority of the Internet traffic is burst, certain tests were conducted with bursty traffic to observe the effects of CoS on bursty traffic. The MPLS domain in this case was configured to operate in multi-VC LBR mode. The MPLS LFIB is as given below on jake and drag

```
jake#show tag-switching forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     Multi-VC    2.2.2.4/32        0          AT0/0/0.100 point2point
27     Multi-VC    2.2.2.6/32        0          AT0/0/0.100 point2point
28     Multi-VC    192.168.3.0/24    0          AT0/0/0.100 point2point
```

```
drag#show tag-switching forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     Multi-VC    2.2.2.6/32        0          AT5/0.100  point2point
27     Multi-VC    192.168.4.0/24    0          AT5/0.100  point2point
28     Multi-VC    2.2.2.2/32        0          AT5/0.100  point2point
```

The test involved sending three bursty streams confined to a bandwidth of 20 Mbps using Netspec and one bursty stream confined to a bandwidth of 10 Mbps from qost3 to qost2. A full stream was sent from qost1 to qost2. Also, a bursty stream confined to 10 Mbps was constantly filling the pipe between drag and snag. The test results are as shown below.

Tx node	ToS	Throughput Tx Mbps	Throughput Rx Mbps
i. MPLS domain = 1 => 20, 2 => 30, 3 => 40, no Cos on snag			
qost3	0	20.001	17.133
qost3	0	20.002	17.274
qost3	0	20.002	17.195
qost3	0	6.001	5.289

qost1	6	131.701	51.276

ii. MPLS domain = 1 => 20, 2 => 30, 3 => 40, WFQ configured on snag's interfaces

qost3	0	19.985	17.112
qost3	0	19.985	17.215
qost3	0	19.986	17.134
qost3	0	5.996	5.305
qost1	6	131.701	53.274

iii. MPLS domain = 1 => 20, 2 => 30, 3 => 40, WFQ configured on snag's interfaces

qost3	1	20.001	17.122
qost3	2	20.001	17.149
qost3	4	20.002	17.226
qost3	0	6.001	5.292
qost1	6	131.721	45.581

iv. MPLS domain = 1 => 20, 2 => 30, 3 => 40, WFQ configured on snag's interfaces

qost3	1	19.989	17.135
qost3	2	19.989	17.123
qost3	0	19.990	17.265
qost3	0	5.997	5.298
qost1	6	131.705	48.163

v. MPLS domain = 1 => 20, 2 => 30, 3 => 40, WFQ configured on snag's interfaces

qost3	6	20.001	17.154
qost3	6	20.002	17.223
qost3	6	20.002	17.078
qost3	6	6.001	5.293
qost1	0	131.702	46.700

vi. MPLS domain = 1 => 20, 2 => 30, 3 => 40, no CoS on snag

qost3	6	20.001	17.229
qost3	6	20.001	17.098
qost3	6	20.002	17.211
qost3	6	6.001	5.303
qost1	0	131.714	46.681

From the above tests it can be confirmed that the unused bandwidth is shared unequally among the classes and that higher priority classes get more share of the unused bandwidth than the others. This is evident from tests ii, iii and iv. Test v shows the effect of CoS on zero precedence full stream traffic when it shares the same pipe with other higher priority bursty flows. Test vi shows the effect of having no CoS in the second domain. No observable changes were got when CoS was turned off on snag owing to the processing capabilities of the 7200s at greater than OC-3 speeds.

9. Class Based WFQ - CBWFQ was configured on all the routers assigning fixed bandwidth to fixed precedence. The way this is configured is as follows

(i) Create access lists to filter traffic

```
jake(config)# access-list 110 permit ip any any precedence network
jake(config)# access-list 116 permit ip any any precedence priority
```

(ii) Create class maps to assign access-lists to classes

```
jake(config)#class-map prec7
jake(config-cmap)# match access-group 110
jake(config)# class-map prec1
jake(config-cmap)# match access-group 116
```

(iii) Create policy map to assign bandwidths to classes

```
jake(config)# policy-map cbwfq
jake(config-pmap)# class class-default
jake(config-pmap-c)# bandwidth 10000
jake(config-pmap)# class prec7
jake(config-pmap-c)# bandwidth 50000
jake(config-pmap)# class prec1
jake(config-pmap-c)# bandwidth 20000
```

(iv) Attach the policy map to an interface

```
jake(config)# interface ATM0/0/0
jake(config-if)# <standard configuration>
jake(config-ig)# service-policy output cbwfq
```

As before, four bursty streams were sent from qost3 to qost2 and one full stream was sent from qost1 to qost2. A background bursty traffic of 10 Mbps was also sent to fill the pipe between snag and drag. The results for UDP are as shown below.

Bandwidth of 50 Mbps to prec 7, 10 to prec 0 on all routers

Tx node	ToS	Throughput Tx	Throughput Rx
(i)		Mbps	Mbps
qost3	0	20.001	17.638
qost3	0	20.002	17.655
qost3	0	20.002	17.658
qost3	1	4.000	3.568
qost1	7	131.724	59.693
(ii)			
qost3	1	19.993	17.627
qost3	1	19.993	17.630
qost3	1	19.994	17.643
qost3	1	5.998	5.307
qost1	7	131.714	60.029

(iii)

qost3	7	20.001	17.634
qost3	7	20.001	17.634
qost3	7	20.002	17.633
qost3	7	6.001	5.304
qost1	0	131.721	52.527

The above three tests illustrate the effects of assigning fixed bandwidths to various precedence. Here too, the bandwidth assignment is relative and not fixed to prevent starvation of lower priority flows. This is evident from tests ii and iii. It was also observed that VCs were not teared down and signaled again when bandwidth parameters corresponding to an interface was changed.

8. Effects of tuning CoS parameters on snag - The router running IP was configured with different CoS parameters and different weights and flows with different precedences were run through it. No observable change in throughput could be seen for any of the configurations. A reason for this was the buffering capabilities of the 7200s. Another reason was the inappropriate MPLS CoS treatment given to packets by the GSR which acts as the core router.

4.3.3.3 Observations

From the above tests, it was observed that

(i) MPLS CoS does fair bandwidth allocation in a way that lower priority flows are not starved and at the same time giving a relatively higher proportion of the unused bandwidth to higher priority flows.

(ii) TCP flows were not found to get any preferential treatment irrespective of their priorities even when congestion avoidance mechanisms were configured with appropriate thresholds.

(iii) When bandwidth parameters corresponding to the different classes changed, signaling took place to tear down old VCs and create new VCs with the new bandwidth parameters.

(iv) The aggregate throughput through the network did not go beyond 80 Mbps when full stream traffic was sent and did not go beyond 120 Mbps when bursty traffic was sent.

4.4 MPLS Traffic Engineering

4.4.1 Background

Traffic Engineering (TE) is concerned with performance optimization of operational networks. In general, it encompasses the application of technology and scientific principles to the measurement, modeling, characterization and control of Internet traffic, and the application of such knowledge and techniques to achieve specific performance objectives. The key performance objectives associated with Traffic Engineering can be classified as being either

1. Traffic oriented (or)
2. Resource oriented.

Traffic oriented performance objectives include the aspects that enhance the QoS of traffic streams. In a single class, best effort Internet service model, the key traffic oriented performance objectives include minimization of packet loss, minimization of delay, maximization of throughput, and enforcement of Service Level Agreements. Resource oriented performance objectives include aspects pertaining to the optimization of resource utilization. In particular, it is desirable to ensure that subsets of network resources do not become over utilized and congested while other subsets along alternate feasible paths remain underutilized. Bandwidth is a crucial resource in contemporary networks. Therefore, a central function of Traffic Engineering is to efficiently manage bandwidth resources. Minimizing congestion is a primary traffic and resource oriented performance objective. The interest here is on congestion problems that are prolonged rather than on transient congestion resulting from instantaneous bursts. Congestion typically manifests under the two scenarios:

1. When network resources are insufficient or inadequate to accommodate offered load.
2. When traffic streams are inefficiently mapped onto available resources; causing subsets of network resources to become over-utilized while others remain underutilized.

The second type of congestion problem, namely those resulting from inefficient resource allocation, can usually be addressed through Traffic Engineering. The control capabilities offered by existing Internet Interior Gateway Protocols (IGP) are not adequate for Traffic Engineering. Indeed, IGPs based on shortest path algorithms contribute significantly to congestion problems in Autonomous Systems within the Internet.

Shortest Path First algorithms generally optimize based on a simple additive metric. These protocols are topology driven, so bandwidth availability and traffic characteristics are not factors considered in routing decisions. Consequently, congestion frequently occurs when (i) the shortest paths of multiple traffic streams converge on specific links or router interfaces. (ii) a given traffic stream is routed through a link or router interface which does not have enough bandwidth to accommodate it. These scenarios manifest even when feasible alternate paths with excess capacity exist. It is this aspect of congestion problems that Traffic engineering aims to vigorously obviate. The MPLS WG of the IETF has been working on extensions to IGPs to calculate paths that will help in load balancing. The WG is also working on signaling bandwidth parameters while setting up LSPs and provide capabilities for explicit and constraint based routing. RSVP with TE extensions and CR-LDP are two main signaling protocols that are used to set up LSPs in an MPLS TE environment. A detailed working of the same can be found in [ref].

4.4.2 Testing and evaluation

The testing of MPLS TE is in reference to the following figure.

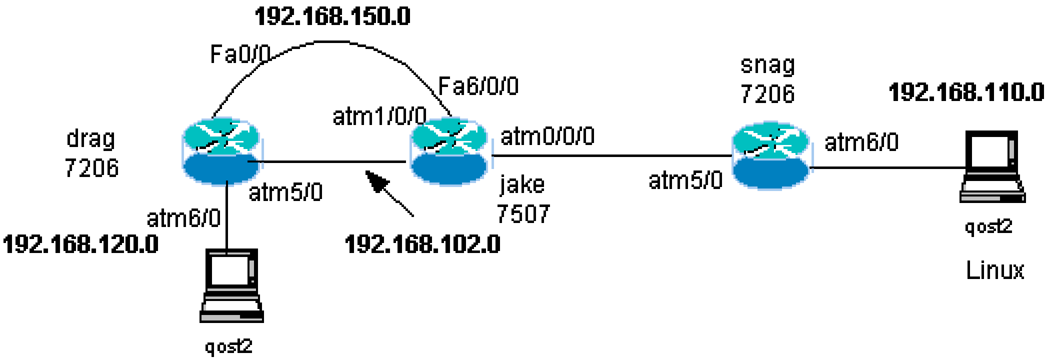

Figure 11. Test setup for testing MPLS TE.

4.4.2.1 Configuration

The configuration needed for a network to support MPLS TE in addition to conventional MPLS is as follows Cisco IOS 12.0(7) T with automated MPLS TE features was used in the testing.

(i) Enabling MPLS traffic engineering tunnels - This is needed to enable MPLS traffic engineering on a router. This is done in global configuration mode and on all the interfaces that need to support MPLS TE.

```
drag(config)# mpls traffic-eng tunnels
drag(config)# interface FastEthernet0/0
drag(config-if)# mpls traffic-eng tunnels
```

(ii) Allocating RSVP bandwidth on an interface - RSVP is reserved some bandwidth on an interface. All the interfaces that take part in MPLS TE should be configured with some bandwidth for RSVP.

```
drag(config)# interface FastEthernet0/0
drag(config-if)# ip rsvp bandwidth 1000 1000
```

(iii) Enabling MPLS traffic engineering calculations on IGP: OSPF was used as the IGP for MPLS TE. To enable modified SPF calculation in a network, all the routers participating in MPLS TE have to be configured with a router ID and an area as follows.

```
drag(config)# router ospf 10
drag(config-router)# mpls traffic-eng router-id Loopback0
drag(config-router)# mpls traffic-eng area 10
```

Loopback is used in this case, as it is an interface that is always up.

(iv) Creation of tunnels - Tunnels are finally created that will serve as traffic trunks/LSPs. A tunnel can be created with a specific bandwidth, priority for preemption, path and destination. The path can be either dynamic (calculated by the IGP) or can be explicitly mentioned at the head end. An example configuration is shown below.

```
drag(config)# interface Tunnel2
drag(config-if)# ip unnumbered Loopback0
drag(config-if)#no ip directed-broadcast
drag(config-if)# tunnel destination 2.2.2.5
drag(config-if)# tunnel mode mpls traffic-eng
drag(config-if)# tunnel mpls traffic-eng autoroute announce
drag(config-if)# tunnel mpls traffic-eng priority 7 7
drag(config-if)# tunnel mpls traffic-eng bandwidth 100
drag(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
```

4.4.2.2 Results

Tests were conducted to verify connectivity, resource allocation, re-routing of traffic when a LSP goes down and load balancing of traffic on multiple LSPs. Two tunnels with dynamic option and one tunnel with explicit path option was set up and higher priority was assigned to the dynamic path option tunnels. Each of the tunnels was assigned a bandwidth of 100 kbps.

1. Display of LFIB at the headend - drag was configured with tunnels and was the headend for three tunnels. The LFIB at drag is as shown below before and after setting up tunnels.

```
drag#show tag-switching forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     1/51        2.2.2.2/32        0         AT5/0.100  point2point
       Pop tag     2.2.2.2/32        0         Fa0/0      192.168.150.2
27     28          2.2.2.5/32        0         Fa0/0      192.168.150.2
       1/87        2.2.2.5/32        0         AT5/0.100  point2point
28     1/80        192.168.101.0/24  0         AT5/0.100  point2point
       Pop tag     192.168.101.0/24  0         Fa0/0      192.168.150.2
29     29          192.168.110.0/24  0         Fa0/0      192.168.150.2
       1/88        192.168.110.0/24  0         AT5/0.100  point2point
```

```
drag#show tag-switching forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     1/51        2.2.2.2/32        0         AT5/0.100  point2point
       Pop tag     2.2.2.2/32        0         Fa0/0      192.168.150.2
27     Untagged[T] 2.2.2.5/32        0         Tu2        point2point
       Untagged[T] 2.2.2.5/32        0         Tu25       point2point
       Untagged[T] 2.2.2.5/32        0         Tu2225     point2point
28     1/80        192.168.101.0/24  0         AT5/0.100  point2point
       Pop tag     192.168.101.0/24  0         Fa0/0      192.168.150.2
29     Untagged[T] 192.168.110.0/24  0         Tu2        point2point
       Untagged[T] 192.168.110.0/24  0         Tu25       point2point
       Untagged[T] 192.168.110.0/24  0         Tu2225     point2point
```

From the above displays, it can be seen that 3 tunnels were created at the headend and were treated as separate traffic trunks/LSPs.

2. Display of traffic engineering tunnels:

```
drag#show mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:           running
    RSVP Process:                  running
    Forwarding:                    enabled
    Periodic reoptimization:       every 3600 seconds, next in 2139 seconds
TUNNEL NAME                      DESTINATION      STATUS      STATE
drag_t2                          2.2.2.5          up          up
drag_t25                         2.2.2.5          up          up
drag_t2225                       2.2.2.5          up          up
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

The above display shows the three tunnels and their corresponding status at the headend.

3. Display of traffic engineering topology

```
drag#show mpls traffic-eng topology brief
My_System_id: 2.2.2.4, Globl Link Generation 156
```

```
IGP Id: 2.2.2.2, MPLS TE Id:2.2.2.2 Router Node
    link[0 ]:Intf Address: 192.168.102.1    Generation 125
            Nbr IGP Id: 2.2.2.4, Nbr Intf Address:192.168.102.2
    link[1 ]:Intf Address: 192.168.150.2    Generation 125
            Nbr IGP Id: 192.168.150.1, Nbr Intf Address:192.168.150.1
    link[2 ]:Intf Address: 192.168.101.1    Generation 154
            Nbr IGP Id: 2.2.2.5, Nbr Intf Address:192.168.101.2
IGP Id: 2.2.2.4, MPLS TE Id:2.2.2.4 Router Node
    link[0 ]:Intf Address: 192.168.102.2    Generation 155
            Nbr IGP Id: 2.2.2.2, Nbr Intf Address:192.168.102.1
    link[1 ]:Intf Address: 192.168.150.1    Generation 156
            Nbr IGP Id: 192.168.150.1, Nbr Intf Address:192.168.150.1
IGP Id: 2.2.2.5, MPLS TE Id:2.2.2.5 Router Node
    link[0 ]:Intf Address: 192.168.101.2    Generation 126
            Nbr IGP Id: 2.2.2.2, Nbr Intf Address:192.168.101.1
IGP Id: 192.168.150.1, Network Node
    link[0 ]:Intf Address: 0.0.0.0    Generation 101
            Nbr IGP Id: 2.2.2.4,
    link[1 ]:Intf Address: 0.0.0.0    Generation 101
            Nbr IGP Id: 2.2.2.2,
```

Details regarding the outcome of the SPF algorithm and the various traffic trunks can be found in the above display.

4. Display of explicit path for explicitly routed tunnel:

```
jake#show ip explicit-paths
PATH 1 (strict source route, path complete, generation 4)
   1: next-address 2.2.2.5
   2: next-address 2.2.2.4
```

The above display shows the explicit path taken by the explicitly routed tunnel in the MPLS TE network.

5. Re-routing- Ping packets were constantly being sent on one tunnel from qost2 to qost1 and that tunnel was brought down and using debug messages and traceroute from the end hosts, the new path taken by the ping packets was observed. The following were also observed

(i) The time taken for re-routing was considerably large, on an average it was 3 seconds.

(ii) During re-routing, ping packets were dropped. Around 6-7 packets were dropped on an average

(iii) Re-routing always chose an alternative interface even when trunks were available on the interface on which the earlier trunk failed.

6. Load balancing - Traffic was run through from qost1 to qost2 and multiple streams were sent to fill the tunnels. No load balancing of traffic was observed even when the aggregate traffic bandwidth exceeded the

link bandwidth. Also, it was observed that all the streams were sent on a particular tunnel even when an alternate interface and tunnel were available to load balance traffic. Due to resource constraints, further experiments to test load balancing could not be conducted.

4.4.3 Observations

From the above tests, it was observed that, with MPLS TE,

1. Explicit Label Switched Paths (LSP) not constrained by the destination based forwarding paradigm can be set up.

2. LSPs can be efficiently maintained and preemption priorities can be assigned to LSPs.

3. A set of attributes can be associated with resources, which constrain the placement of LSPs and traffic flows across them.

4. MPLS allows for both traffic aggregation and de-aggregation whereas classical destination only based IP forwarding permits only aggregation.

5. Implementations on Linux

Two implementations were done by the working group. One was the implementation of MPLS traffic engineering on Linux. This was done as part of the thesis work by a member of the working group. The other was an implementation of IP to MPLS CoS translation over ATM in multi-VC LBR mode on Linux. The software architectures and descriptions of the two implementations are discussed in the following sections.

5.1 MPLS TE on Linux

The various modules used in this implementation and their interaction are as shown in the figure above. The OSPF daemon executes the SPF algorithm after establishing adjacency and link state database. During this execution, the multiple routes module determines multiple routes to all destinations. Opaque capability is advertised to the OSPF peers in the OSPF hello exchange process. Traffic engineering LSA are advertised to the OSPF peers and contain the Router address TLV and link TLV. Bandwidth is measured at outgoing ports of the ATM switch using SNMP. The SNMP module calculates used link bandwidth averaged over a

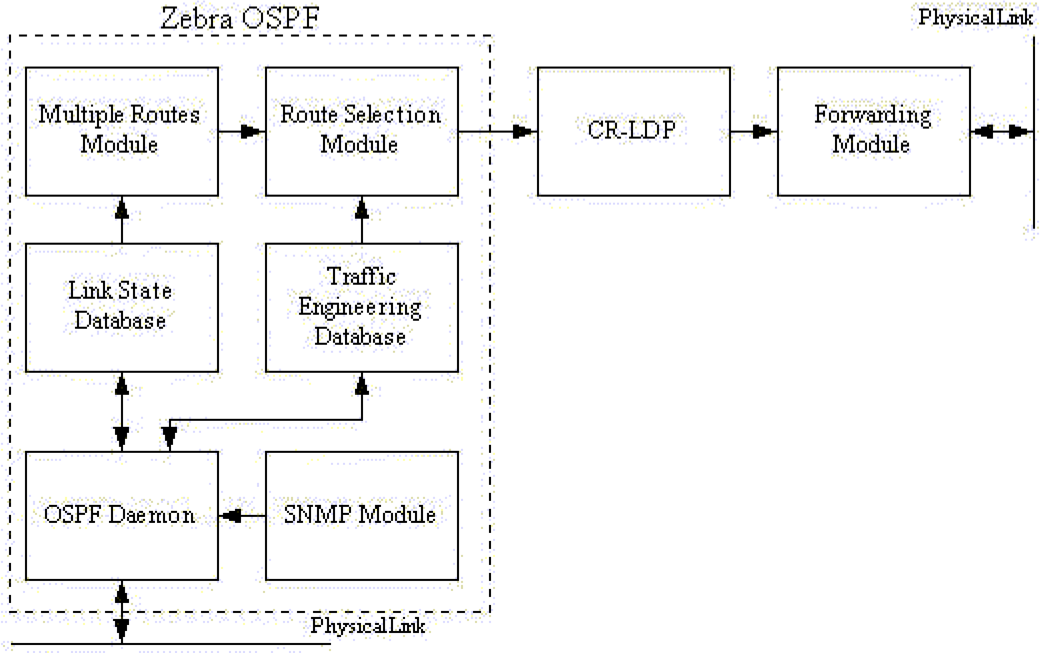

Figure 12. Software architecture for the implementation of MPLS TE on Linux

particular period to accommodate bursts. TE LSAs are sent when the used link bandwidth exceeds a particular threshold and when there is a significant bandwidth usage. These events cause the available bandwidth to be set to zero and drop the route from the route selection algorithm. CR-LDP is used to set up the LSP. MPLS forwarding takes place when there is a label for the destination prefix. If the precedence of the packet does not match the pre-configured precedences of any of the LSPs, an LSP with the least precedence is used to forward the packet. The implementation made use of GNU Zebra and integrated a number of commands to the existing zebra OSPF configuration tool set.

5.2 IP to MPLS CoS translation over ATM on Linux in multi-VC LBR mode:

The modules used in this implementation and their interactions are as shown in the figure above. OSPF is run on three nodes that also run MPLS and it builds a routing table. BGP is used to advertise network prefixes from one ASBR to another and hence update routing tables. These routes are then sent to the LIB module that does LDP signaling and establishes four labels per source destination pair. The eight IP precedences are distributed among the four labels as described in section x. Due to the limitations with the ENI ATM driver that comes with Linux, the implementation does a mapping of three of the classes onto

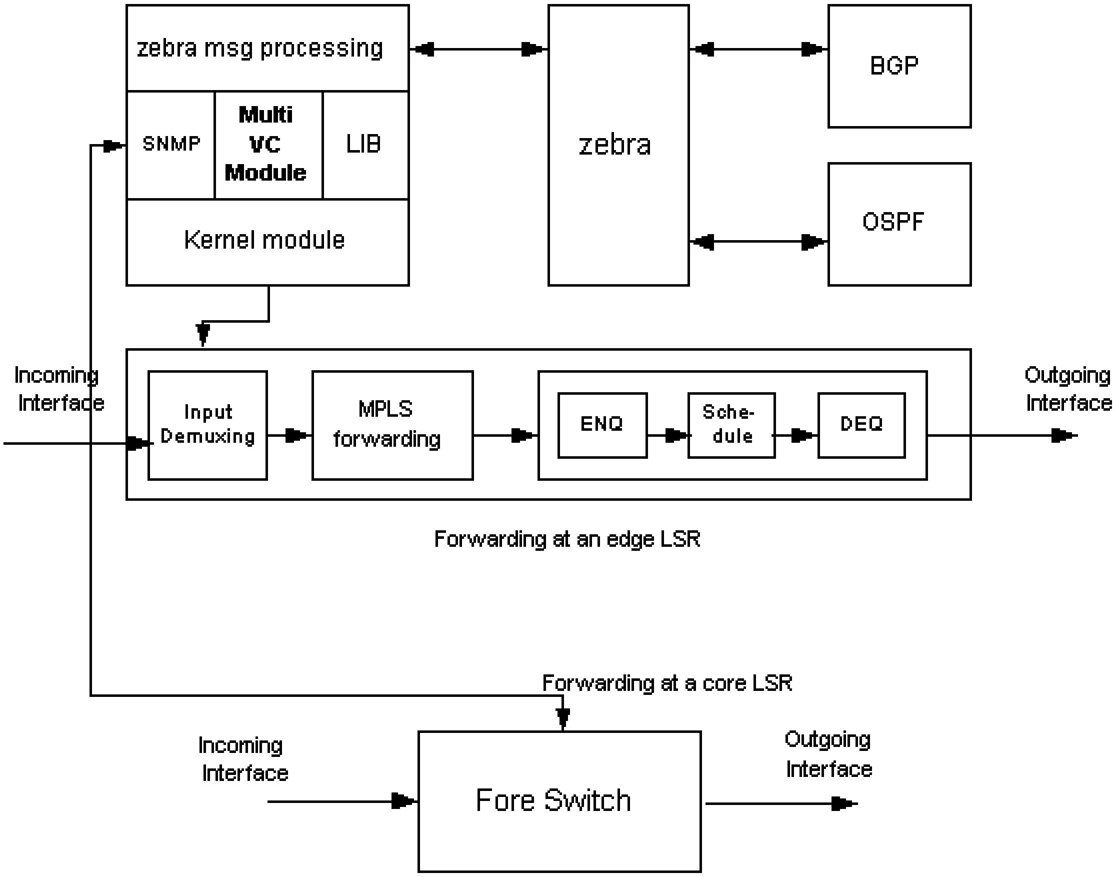

Figure 13. Software architecture of IP to MPLS CoS translation implementation in Multi-VC LBR mode

UBR and the highest priority class onto CBR. SNMP is used to set up VCs on the switch and also set appropriate QoS parameters on the switch for a VC using UPC contracts. The LIB is then sent to the kernel that does appropriate forwarding. The multi-VC module also takes care of assigning appropriate CoS bandwidth to the four classes and configuring filters and qdiscs in the kernel that lead to appropriate CoS while forwarding a packet. CLP bit is set or not set depending on its IP precedence. The architecture employs CBQ at the edge and CBQ and EPD at the core. The implementation made use of GNU Zebra, TC tool and iproute2 and integrated a number of commands to the existing zebra configuration tool set.

6. QoS deployment issues

This section evaluates some of the management and deployment requirements of deploying QoS architecture(s) within an existing network infrastructure. The key issues considered are scalability, router/switch configuration complexity, processor loads, stability of the network and individual nodes, peer and overlay models in QoS deployment and PHB design in relation to an MPLS infrastructure.

6.1 Peer vs Overlay model

The overlay model consists of IP routers at the edge of the network and ATM switches at the core of the network. From a routing viewpoint, this creates a cut-through in the network wherein every IP router will see every other IP router at the edge of the network as being one hop away. This is primarily because the core switches do not take part in the routing protocols and hence are seen as just cross-connects by the IP routers. Hence, QoS deployment in the overlay model involves provisioning resources at the edge and signaling at the core to provision resources. Signaling involves a lot of overhead and also has to take place link by link.

The peer model consists of IP intelligence at every hop inside the core of the network. This leads to ATM switches taking part in IP routing protocols and hence lead to better route selection and bandwidth management. This also has the disadvantage that ATM switches need to understand the mechanisms of IP routing protocols. With tens of thousands of routes injected into the core of the network, the switch's efficiency breaks down. MPLS addresses this issue by letting IP friendly routers take care of routing and signaling and letting the switch do the forwarding at the core of the router by establishing crossconnects based on the routing table built.

6.2 Scalability

The key to scalability is distribution. Scalability is a very important issue when deploying QoS in the wide area. The Integrated services model does not scale very well because of the infinite amount of state variables that has to be stored at a node. Inside the Service Provider backbone, maintaining this infinite state (node to which a RESV has to be sent, node to which the request has to be sent etc.) does not scale very well. The Differentiated Services model eliminates this problem by introducing the peer model as described above. But it has most of the disadvantages that forwarding using IP has at the provider core. There isn't a scalability problem here but there is an efficiency problem especially at speeds of the order of OC-48 or OC-192. The peer model that makes use of MPLS eliminates both the above problems. MPLS has its disadvantages too. It requires that a router be present at every node in the provider core to control the switch. However, the benefits, i.e. scalable MPLS VPNs, re-routing and load-balancing using TE might be a tradeoff for providing a switch and router at every node.

MPLS VPNs scale extremely well compared to the conventional layer 2 VPNs. Security equal to that offered by the layer 2 VPNs is achieved through route filtering and maintenance of per site forwarding tables. The addition of a new customer will not affect the entire topology but will only require changes to the PE to which the customer connects.

IP CoS requires the presence of a Bandwidth Broker to validate and set precedence for flows from the customer. The Bandwidth Broker again requires knowledge of the topology of the network to provision resources. Dynamic routing changes have to be accommodated into the Bandwidth Broker implementation and hence constitutes a problem. Also, if individual flows were to be taken care of by the Bandwidth Broker, then this model would have severe scalability problems. Hence, only aggregates should provision resources.

6.3 Router/switch configuration complexity

An issue closely related to scalability is the configuration that is needed at a node to implement an appropriate technology. A configuration file that spans thousands of lines is both hard to maintain and difficult to configure. Considering the enormous amount of filtering that is needed at the service provider boundaries, if the technology in use demands additional constraint on the earlier filters, then the configuration file with grow without bounds. Such configuration files give rise to additional complexity when faults have to be rectified. The interaction between the router and the switch also needs to take place via some protocol. The simplest of them all, SNMP, does not scale well when the number of requests are large. VSI and GSMP are two protocols that were introduced to control a switch using a router. Each has its own advantages and disadvantages.

6.4 Stability of the network and individual nodes

Routers and switches have to be chosen properly so as to withstand the traffic flowing through them. It was seen from the MPLS CoS testing that the 7200s had buffering problems when traffic through them exceeded OC-3 rates. A common problem that was encountered while configuring and testing was that OSPF routes were not exchanged when an interface was taken down and brought up. Another problem that was encountered was that disabling an operational MPLS interface disabled signaling on that interface which did not get enabled even after the interface was brought up. The solution was to bring down the shelves on the switch and then bring it up. Such problems can cause long outages in the network. The instability of routes in the Internet, fiber cuts, the prolonged time for network nodes to react can cause severe instabilities in the network. There are proposed solutions for route flap dampening but all this depends on the stability of the network itself.

6.5 Processor loads

The bulk of Internet routes in the Service Provider core give rise to a lot of processing overhead when forwarding a packet using plain IP. Choosing a destination from less than 10 entries would involve searching through the forwarding table for a matching entry, but with tens of thousands of routes, choosing an entry would involve complicated hashing and tree algorithms. If the forwarding were based on the precedence, then this would give rise to additional overhead. In addition to forwarding, queueing, scheduling and other elements of traffic conditioning might take a lot of processor cycles. The issue here is to minimize the CPU load when forwarding a packet and also minimize drops due to buffer occupancy on a node. MPLS aims at reducing the processor load by decoupling routing and forwarding paradigms. The switch will be set with certain QoS parameters and it will do the function of forwarding while the router will take care of building routing tables, establishing cross-connects, configuring the switch with appropriate buffer size, queue size etc.

6.6 PHB design in relation to an MPLS infrastructure

Different applications require different PHBs at a network node depending on the time, resource requirements and nature of the application. Enabling per-CoS treatment at a network node is not an easy task given the constraints. Some of the issues involved in providing MPLS CoS treatment are

(i) Appropriate mapping from IP to MPLS CoS, E-LSPs vs L-LSPs, Pipe model vs Uniform model.

(ii) Mapping CoS over ATM, i.e. given the precedences and application requirements, how can the traffic be mapped onto ATM. Can EF be mapped to CBR, AF to ABR etc. How to provide differentiation among AF classes if mapped so. When to set the CLP bit?

(iii) Is a Bandwidth Broker needed? If so, how will it maintain the dynamic routing changes and a correct view of the network topology? Also, how easy is it to provision resources end to end when the flow as to traverse multiple domains?

(iv) How to translate application requirements to PHBs?

(v) How to provide PHB treatment when traffic has to pass through multiple domains. How can it be ensured that all domains will give the same kind of treatment to the packet?

(vi) SLAs between two Autonomous Systems. Will the same set of PHBs be available in the next domain too?

(vii) Given the relatively large number of routes advertised in the Internet, will establishing multi-VCs per source destination pair be a good solution. Is aggregation necessary?

(vii) If VC merge is supported, how to de-aggregate flows and also prevent cell interleaving problem when used over ATM.

A good network design should encompass atleast some of the above issues and should provide cost efficient solutions to the different problems. The solutions to the above issues can be found in [7, 8, 15, 23 and 24]. There also has to be tradeoffs between price paid by customers, cost of buying and maintaining equipment, provisioning resources in the network and maintaining the network, preventing outages, satisfying customer needs etc.

7. Conclusion

The ever-growing demands of applications running on IP and the wide variety of Network access media available has made IP become the end-to-end QoS enabler. The working group has, in this project, studied, analyzed, tested and evaluated the features of some of the currently available CoS technologies on IP, ATM and MPLS within the constraints of the resources available. Issues related to interaction between IP, ATM and MPLS CoS components have been tested and evaluated. A MPLS VPN was implemented in the wide area using the Sprint backbone and Cisco gear and issues related to the scalability, ease of deployment and security have been elaborated. Observations, challenges and results related to the various experiments done on Cisco gear have also been discussed. Two implementations related to MPLS operations on Linux have been done and explained in this report. Deployment scenarios in the wide area have been illustrated and the various QoS deployment issues that are to be considered when designing a QoS aware network have been discussed. Issues related to deploying MPLS in the wide area have been elaborated in detail.

Reference:

[1] E. Rosen, Y. Rekther, "BGP/MPLS VPNs", draft-rosen-rfc2547bis-02.txt, July 2000.

[2] T. Bates, R. Chandra, D. Katz, Y. Rekther, "Multiprotocol Extensions for BGP-4", RFC 2858, Feb 1998.

[3] Bates and Chandrasekaran, "BGP Route Reflection: An alternative to full mesh IBGP", RFC 2796, April 2000.

[4] Rosen, Viswanathan, and Callon, "Multiprotocol Label Switching Architecture", July 2000, work in progress.

[5] Rekhter and Rosen, "Carrying Label Information in BGP4", January 2000, work in progress

[6] B. Davie et. al, "MPLS using LDP and ATM VC Switching", draft-ietf-mpls-atm-04.txt, June 2000.

[7] Bilel Jamoussi et. al, "Constraint-Based LSP Setup using LDP", draft-ietf-mpls-cr-ldp-04.txt, July 2000.

[8] L. Faucheur et. al, "MPLS Support of Differentiated Services", draft-ietf-mpls-diff-ext-07.txt, August, 2000.

[9] J. Moy, "OSPF Version 2", RFC 1583, Mar 1994.

[10] P. Almquist, "Type of Service in the Internet Protocol Suite", RFC 1349, Jul 1992.

[11] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999.

[12] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, Jun 1999.

[13] S. Blake et. al, "An Architecture for Differentiated Services", RFC 2475, Dec 1998.

[14] L. Zhang et. al, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, Sep 1997.

[15] Xipeng Xiao and Lionel M. Ni, "Internet QoS: A Big Picture", IEEE Network Magazine, March 1999.

[16] Sally Floyd, Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993.

[17] Elloumi, O., and Afifi, H., "RED Algorithm in ATM Networks", Tech report, June 1997.

[18] GAIT Internal Document, " CA*net II Differentiated Services - BB system Specification " Oct 1998.

[19] P. Ferguson, and G. Huston, "What is a VPN?", Revision 1, April 1 1998.

[20] Netspec traffic generator tool, www.ittc.ukans.edu/netspec

[21] Zebra routing software, www.zebra.org

[22] Cisco configuration pages.

[23] Rüdiger Geib, "Differential Services for the Internet and ATM", Jul 1999.

[24] D. Awduche et. al, "Requirements for Traffic Engineering Over MPLS", RFC 2702, Sept 1999.

[25] Li, T. and Y. Rekhter, "Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)", RFC 2430, October 1998.

[26] Andersson, et al, “LDP Specification”, August 2000.