

**CERTIFICATION OF APPROVAL**

**Internet QoS for DiffServ-Enabled Routers**

by

Mohd Shahridan bin Hassan

A project dissertation report submitted to the  
Information System Programme  
Universiti Teknologi PETRONAS  
in partial fulfillment of the requirements for the  
BACHELOR OF TECHNOLOGY (Hons)  
(INFORMATION SYSTEM)

Approved by,

---

(Anang Hudaya bin Muhamad Amin)

Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

June 2006

i

tk  
5105 875

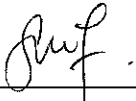
.m697

2006

1) Internet  
2) Computer communication network

## **CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



---

MOHD SHAHRIDAN BIN HASSAN

## ABSTRACT

Differentiated Service Model (DiffServ) is currently a popular research topic as a low-cost method to bring QoS to today's Internet backbone network. In this paper, the author introduces the techniques and methodologies that used to design and implement DiffServ-enabled (DS-enabled) routers. The adaptations of DS-enabled routers are designed to cater to the low Internet connectivity within Universiti Teknologi PETRONAS LAN. The author has implemented basic DiffServ setting using three CISCO3725 routers. Based on these DiffServ-enabled routers, the author set up a small scale lab network to study DiffServ QoS features: priority dropping (discrimination among different service classes), QoS guarantees and measuring QoS using various formal metrics (delay and throughput). Furthermore, the author present problems encountered during study, and the proposed solutions.

## ACKNOWLEDGEMENT

No book is the work only of its authors. I owe much to pioneers of networking who first identified its major issues and developed its concepts and techniques. When it became known what a tremendous UTP's Final Year project had become, I knew I need a great team to do it justice. Luckily I found Data Communication and Networking group. To me, there is no higher compliment than to be a part of Data Communication and Networking.

Million thanks and heartfelt appreciation to my project supervisor, Mr. Anang Hudaya Muhamad Amin for observed and directed the execution of my Final Year Project. Making sure that things got it right was Mr. Suhaimi Abdul Rahman who has been my core project supervisor, I should bow you.

For lab technician who gave me a great support during the system setup, Mr. Ruslan Idris, appreciate what you contribute to me. Not to forget all my friends for their cooperation during the system development and testing processes a bowl of cherries with you all.

One last note of appreciation, for all the people who contributed either direct or indirect support to my Year Project at Universiti Teknologi PETRONAS (UTP), thanks for your support. I hope my effort continue to be worthy.

## LIST OF FIGURES

- Figure 1: The Structure of DS Router [6]
- Figure 2: RIO Queuing Discipline
- Figure 3: Network Topology for the Simulation [6]
- Figure 4: The Methodology of the Project
- Figure 5: Sample Network Implementing DiffServ
- Figure 6: Existing UTP Network Overview
- Figure 7: Existing Paths for Internal and External TCP Traffic Flow
- Figure 8: Proposed TCP Flow Diagram
- Figure 9: Proposed Network Design by Employing DiffServ Technology
- Figure 10: CISCO3725 Setting
- Figure 11: A (a) Contiguous Network and (b) Discontiguous Network.
- Figure 12: Interface of RIP Configuration within Lab Network
- Figure 13: The Connected Interfaces through Routed IPs
- Figure 14: RIP Configuration
- Figure 15: DiffServ Configuration
- Figure 16: TCP Data Packets Captured Based on RIP Configuration
- Figure 17: Summary of TCP Packets Captured Based on RIP Configuration
- Figure 18: Ethereal IO Graphs Captured Based On RIP Setting
- Figure 19: Summary of TCP Packets Captured Based on DiffServ Configuration
- Figure 20: Ethereal IO Graphs Captured Based On DiffServ Setting

## LIST OF TABLES

- Table 1: CISCO3725 Interfaces  
Table 2: DSCP Values for Traffic Classes and Traffic Types  
Table 3: The Ethereal Results

## TABLE OF CONTENT

<b>CERTIFICATION OF APPROVAL.....</b>	<b>i</b>
<b>CERTIFICATION OF ORIGINALITY.....</b>	<b>ii</b>
<b>ABSTRACT.....</b>	<b>iii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iv</b>
<b>LIST OF FIGURES.....</b>	<b>v</b>
<b>LIST OF TABLES.....</b>	<b>vi</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Background of Study.....	1
1.2 Problem Statement and Identification.....	2
1.3 Objectives.....	3
1.4 Scope of Studies.....	3
1.5 Feasibility Study.....	3
<b>CHAPTER 2: LITERATURE REVIEW.....</b>	<b>4</b>
2.1 Introduction.....	4
2.2 Design of DS Routers.....	4
2.3 Implementation .....	6
2.4 DS Router.....	7
<b>CHAPTER 3: METHODOLOGIES / PROJECT WORK.....</b>	<b>10</b>
3.1 Project Approach.....	10
3.1.1 Conceptual Development / Information Gathering.....	10
3.1.2 Requirement Analysis.....	11
3.1.3 Architectural Design.....	16
3.1.4 Project Module Development.....	18
3.1.5 Operation and Maintenance.....	25
3.2 Tools/Equipment.....	25
<b>CHAPTER 4: RESULT AND DISCUSSION.....</b>	<b>26</b>
4.1 RIP Analysis.....	26
4.2 DiffServ Analysis .....	30
<b>CHAPTER 5: CONCLUSION AND RECOMMENDATIONS.....</b>	<b>33</b>
<b>LIST OF REFERENCES.....</b>	<b>35</b>

<b>APPENDICES.....</b>	<b>36</b>
Appendix A Interview Questions.....	37
Appendix B DiffServ Implementation.....	39
Appendix C Configuring RIP.....	54
Appendix D Implementing Quality of Services Policies with DSCP...	63
Appendix E Quality-Of-Service The Differentiated Services Model...	71
Appendix F Which Routing Protocol Should My Network Use ?.....	77



# CHAPTER 1

## INTRODUCTION

### 1.1 BACKGROUND OF STUDY

In the early 90's, in order to provide Quality of Service (QoS) guarantees in the Internet, the Integrated Service Model (IntServ) was proposed. It provides an integrated infrastructure to handle both conventional Internet and QoS-sensitive applications [1]. IntServ uses resource ReSerVation Protocol (RSVP) as its signaling protocol [2]. Although IntServ / RSVP can provide QoS guarantees, it has a scalability problem, since each IntServ enabled router has to maintain state information for each individual flow. To address the scalability issue, a new core stateless model, Differentiated Service Model (Diff-Serv), was proposed and has become a popular research topic as a low-cost method to bring QoS to today's Internet, especially in the backbone networks [3,4]. Although intensive research has been done on DiffServ, it is still difficult to measure the overall impact of DiffServ to the Internet without deploying DS-enabled routers (DS routers) to the Internet. In this work, the author introduces design and implementation of DS-enabled routers in the lab network environment. Intensive configurations and analysis are conducted to verify the design and implementation, and to study the TCP performance over DiffServ in a small-scale lab network through Ethereal analysis. The author also discusses several problems he has encountered during this work, and his solutions.

## **1.2 PROBLEM STATEMENT AND IDENTIFICATION**

As reported by Information Technology and Media Services (ITMS) Unit on 19<sup>th</sup> October 2005, there are several problems that lead to the slow Internet connectivity within UTP campus. The degradation of network performance occurred because of non-education related activities conducted over the network. This activity has generates unnecessary traffic to the internal gateway, caused bottleneck to the internal gateway and with heavy applications such as movies, audio, pictures, games distributed through network that consume most of the bandwidth. Next factor is the existence of illegal hosting servers which have bombarded network with unnecessary traffics such as junk mail, messages, signals so on and so forth. These servers wasted switches and proxy resources and decrease equipment performance, generated unnecessary heavy traffics to internal and external gateway and have attracted and created too many downloading activities. Misused of proxy servers is one of the determinants to the slow Internet connectivity within the campus at which most of proxy resources used by internal hosting servers that drop the proxy performance. These servers create heavy traffic request from internal hosting server thus leads to poor performance. It's also caused high processors utilization to the proxy servers and drops the servers' performance. The increases of network virus and worms within the network caused high processors utilizations rate and drop the network and servers equipment performance. These viruses and worms are distributed through files sharing and download materials from hosting servers.

As per conversation with Mr. Arfaishah from Network Department under ITMS Unit, he explained that UTP itself is using Routing Information Protocol (RIP), which may lead to the degradation of network performance. RIP may be slow to adjust for link failures and in some cases; the exchange of routing table could consume more than an acceptable of data bandwidth. So to cater to this shortage, DiffServ technology is proposed to be implemented.

### **1.3 OBJECTIVES**

Actually this project will be implemented in the Data Communication Lab at Building 2, Universiti Teknologi PETRONAS. The objectives of this project are:

- 1 To identify and analyze UTP traffic flow (TCP traffic)
- 2 To employ DiffServ for internal and external traffic flow
- 3 To propose a network framework to improve network connectivity

### **1.4 SCOPE OF STUDIES**

The project is done within UTP subnet, which is in the lab environment control because of the confidentiality. The scope only covers for internal and external TCP traffic flow.

### **1.5 FEASIBILITY STUDY**

This project conducted within the timeframe of about 1 year. It covers the view of QoS itself on how it works within DiffServ implementation by using three CISCO3725 routers. All the information needed also can be find from books in the library and from the Internet. With all the resources provided, it will be a feasible project in the time given.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 INTRODUCTION

The DiffServ model provisions end-to-end QoS guarantees by using service differentiations and works as follows. Incoming packets are classified and marked into different classes, using Differentiated Services CodePoint (DSCP) [5] (e.g., IPv4 TOS bits or IPv6 Traffic Class bits in a IP header). Complex traffic conditioning such as classification, marking, shaping, and policing are pushed to network edge routers or hosts. Therefore, the functionalities of the core routers are relatively simple - they classify packets and forward them using corresponding Per-Hop Behaviors (PHBs). From the administrative point of view, a DiffServ network could consist of multiple DS domains. To achieve end-to-end QoS guarantees, negotiation and agreement between these DS domains are needed. Although the boundary nodes need to perform complex conditioning like the edge nodes, the interior nodes within DS domains are simple [3,4]. In the current DiffServ model, three service classes have been proposed [6]: the premium class, the assured class, and the best-effort class. Different service classes are suitable for different types of applications.

#### 2.2 DESIGN OF DS ROUTERS

The Differentiated Service enabled routers (DS-enabled routers or DS routers) are key nodes in the DiffServ model. There are two types of DS-enabled routers: (1) edge routers, and (2) core routers. In this work, we focus on the design and implementation of the edge routers. Figure 1 shows the structure of a DS router.

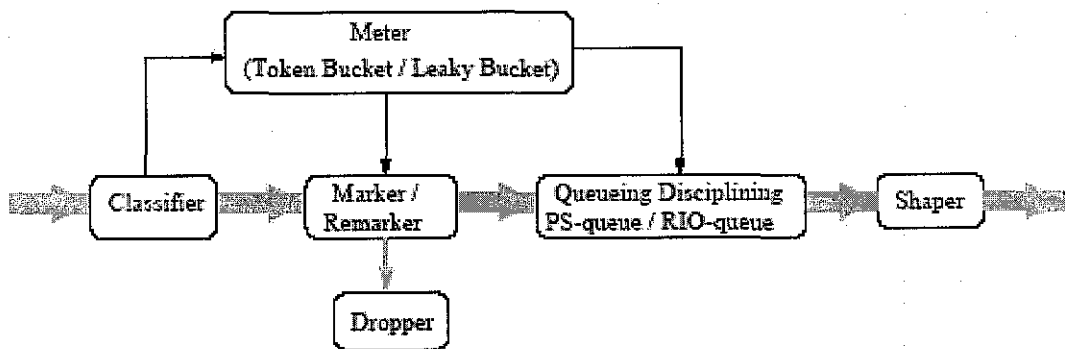


Figure 1: The Structure of DS Router [6]

In the figure, there are several key components to the DS router structure:

- **The Classifier.** The Classifier classifies packets according to their DSCP in the IP headers.
- **The Meter.** The Meter performs in-profile / out-of-profile checking on each incoming packet. Because burstiness exists in assured class traffic, the token bucket scheme is used to check the conformance of assured class traffic. While the leaky bucket scheme is used for premium class traffic, since burstiness is not allowed in premium class traffic.
- **The Marker/Re-marker.** After being classified, packets are marked into premium, assured and best-effort classes accordingly. Re-marking happens when assured packets violate the contracted traffic rate limit and become out-of-profile. These packets are remarked as best-effort packets.
- **The Dropper/Shaper.** If premium packets become out-of-profile, they are dropped immediately by the dropper. Shaping happens at the edge nodes or boundary nodes to eliminate jitters.
- **The Queueing Disciplining Modules.** The differentiation is achieved here. Two separated queues are used: the Premium Service Queue (PS-queue) for the premium packets and the RIO-queue 1 for both assured packets and best-effort packets. The PS-queue is a simple FIFO queue, while the RIO queue is more complicated. Figure 2 illustrates the

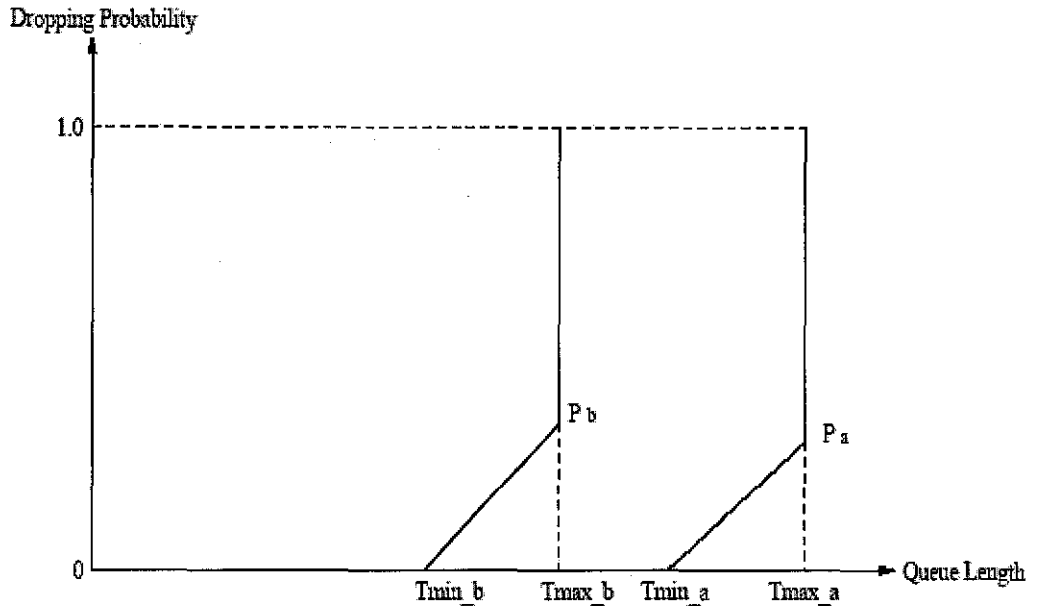


Figure 2: RIO Queuing Discipline

multi-class Random Early Detection (RED) algorithm which the RIO-queue is using. When RIO queue length exceeds the dropping threshold  $T_{min\_b}$ , new best-effort packets are dropped with increasing probability up to  $P_b$ . When RIO queue length exceeds  $T_{min\_a}$ , new assured packets are dropped with increasing probability up to  $P_a$ . When queue length exceeds  $T_{max\_b}$ , all new best-effort packets are dropped. When queue length exceeds  $T_{max\_a}$ , all incoming packets are dropped. By tuning the values of  $T_{min\_b}$ ,  $T_{min\_a}$ ,  $T_{max\_b}$ ,  $T_{max\_a}$ ,  $P_b$  and  $P_a$ , we can expect different dropping behaviors for both best-effort and assured packets.

### 2.3 IMPLEMENTATION

The simulation is implemented using OPNET Modeler 6.0.L running on Windows NT 4.0 Workstation with dual PentiumPro 200Mhz CPU and 128MB of RAM. Figure 3 shows the topology and scale of the simulation environment. The “clients” subnet comprises three client nodes, one switch and one DS edge router. The “INET CLOUD” consists of three DS-enabled / non-DS-enabled routers (it can be expanded to a more complicated topology). The “servers” subnet contains one server and one edge router.

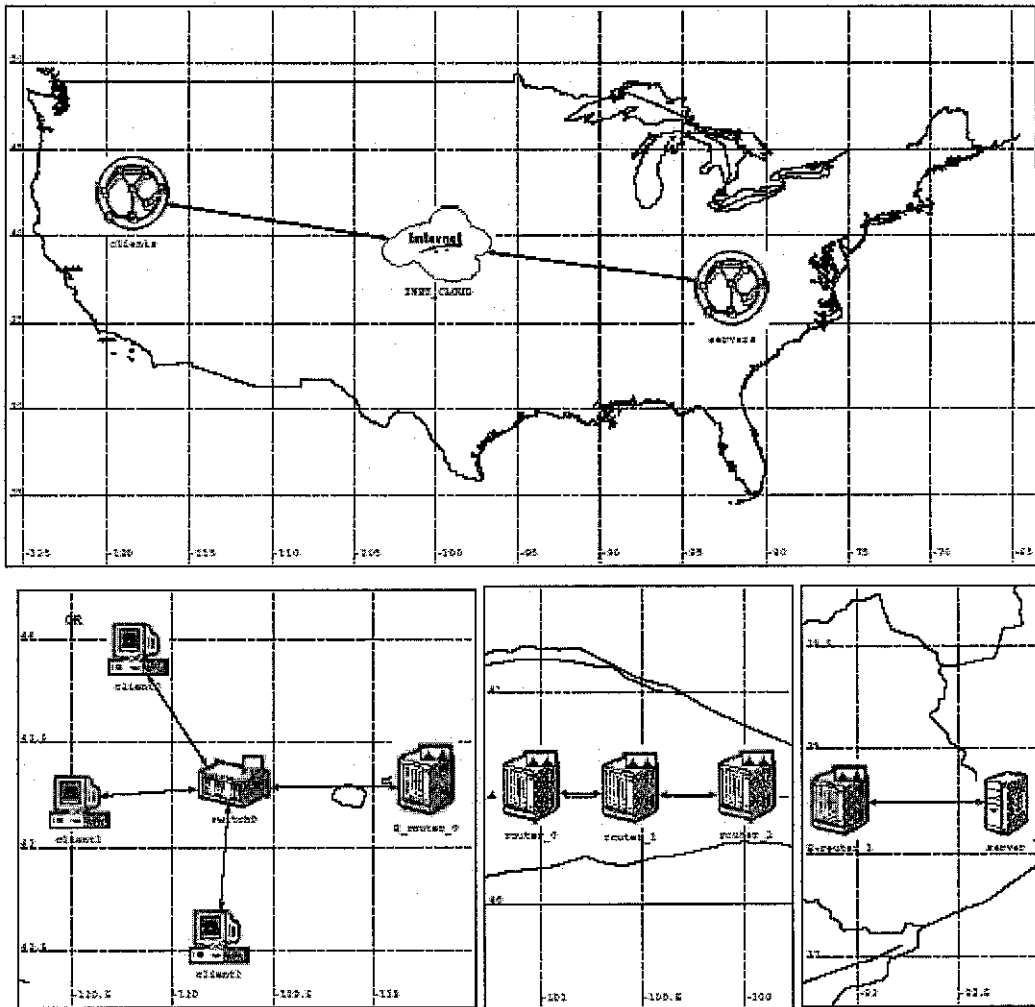


Figure 3: Network Topology for the Simulation [6]

## 2.4 DS ROUTER

To implement the DS scheme in a router, there are two options to choose from. One is to implement the DS-enabled IP module from scratch. The other one is to modify an existing router implementation available from the OPNET vendor's library. In this simulation, existing OPNET model for Cisco 7204 router has been modified. This decision is made based on the fact that most components needed for DS-enabled router exist in the Cisco 7204 router model. Only the IP process model needs to be modified in order to handle traffic from different classes. Therefore, the IP module is modified and put DS components in it. Although making the router DS-enabled is a significant enhancement with respect to functionality, the overall structure of the router has not been changed a lot. The reason is that in OPNET different modules (e.g., MAC, IP, TCP, OSPF, RIP and so on) are implemented as separated objects, which communicate with each other through modular interfaces. As long as DS-

enabled IP module maintains proper interface, it can be swapped into the router model, and the lower and higher protocol layers are able to communicate with the new IP process module properly. Figure 4 illustrates the process model for a DS enabled IP model. There are two different processes in this implementation. The top one `diff_ip_rte_v4` model is the main IP process, which implements main IP and DiffServ functionalities. And the lower one `diff_pq` model is the child process, which implements priority scheduling scheme for DiffServ. The IP `diff_ip_rte_v4` process model is implemented as follows:

- **Initializations.** All the initializations are done through “init”, “wait”, “`cmn_rte_tbl`” and “init\_too” states sequentially, which is the same as what the original OPNET IP process model does.
- **DS and non-DS-enabled states.** If the node is set to be DS-enabled, the transition labeled with “DIFFSERV” condition occurs. Otherwise, the transition labeled with “NO DIFFSERV” condition occurs.
- **Packet Classification.** The packet classification is implemented in the “DS\_schd” state.
- **Packet Monitoring and Policing.** Packet monitoring and policing are implemented in the “DS\_schd” state. After being classified, an incoming packet is monitored and policed according to the class it belongs to. If the packet is a premium class packet, it is monitored and policed by using the leaky bucket model. If the packet is an assured class packet or a best-effort class packet, it is monitored and policed by using the token bucket model (Section 2.2). If the packet is in the premium class and is conformed (in-profile), it is processed by the next state, the “IP\_serv” state, directly. If it is non-conformed (out-of-profile), it is discarded without any further process. If the packet is an in-profile assured packet, it is processed by the “IP\_serv” state, otherwise it is re-marked as a “best-effort” packet in “DS\_schd” state and processed by “IP\_serv” state later. If the packet is a “best-effort” packet, it goes ahead into the “IP\_serv” state and gets processed there.



- **Packet Routing and Forwarding.** After the classification and the conformance check, the packet enters the regular IP forwarding process, which is implemented by “IP\_serv”, “srv\_start”, “srv\_compl” and “idle” states. All of these states are implemented based on IP states provided by the OPNET library, except that the “idle” state is DiffServ-aware.

## CHAPTER 3

### METHODOLOGY / PROJECT WORK

#### 3.1 PROJECT APPROACH

The project will be divided into phases, using the System Development Life Cycle (SDLC), with required deliverables at the end of each phase. The reason why the author uses SDLC is because it provides guidelines that are easy to follow and understand.

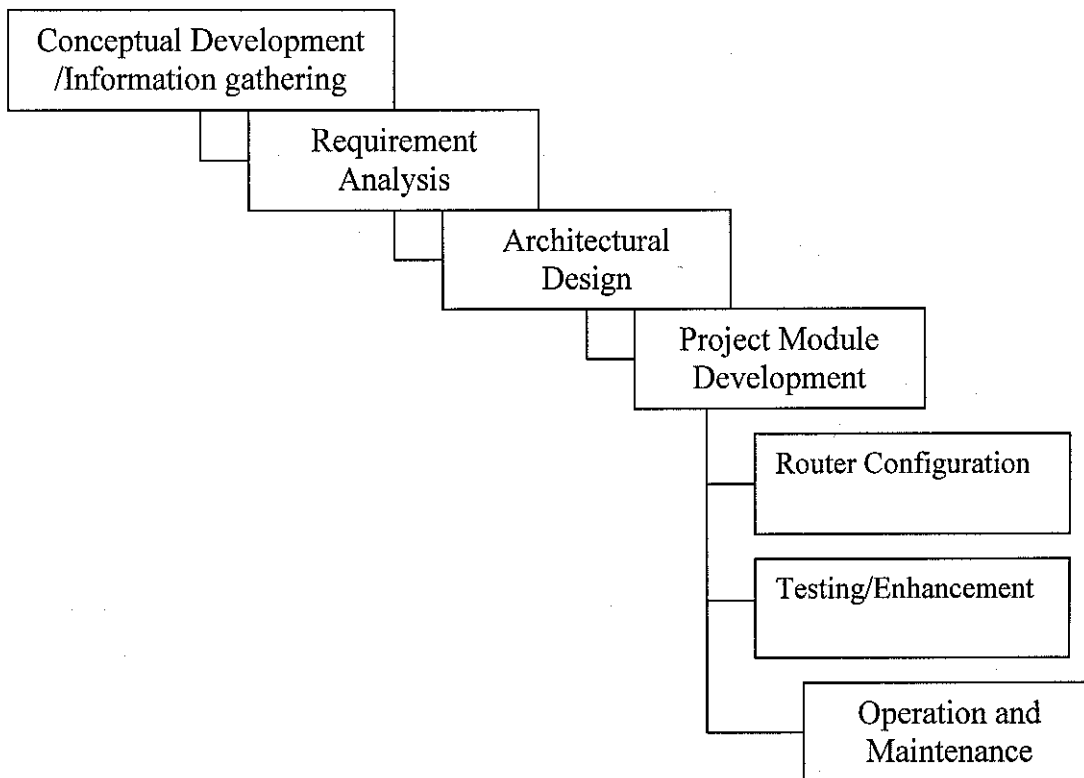


Figure 4: The Methodology of the Project

##### **3.1.1 Conceptual Development /Information gathering**

As shown on the Figure 4 above, the author will start the project by implementing the first step, which is Conceptual Development/Information gathering with the intention to do an observation of the material required for this project. The data gathering method for this project is through the collection of materials from the

library and Internet, interviewing experts in the related subject such as network administrator, lecturer as well as technicians.

### 3.1.2 Requirement Analysis

In the Requirement Analysis phase, author define several requirements need to be considered such as networking equipments. It includes hardware requirement (personal computer, CISCO router), software requirement (Ethereal) and network requirement (Intranet Connection, Internet, packet transfer). Under this stage the author has also identified DiffServ QoS configuration components for the proposed project. The setup for DiffServ will be similarly configured as shown in Figure 5.



Figure 5: Sample Network Implementing Diffserv

### **Differentiated Services Components**

The following components make up the foundation of a CISCO Differentiated Services implementation [7]:

- **Traffic conditioning (traffic policing and traffic shaping)**

Traffic conditioning is performed at the edges of a DiffServ domain. Traffic conditioners perform traffic shaping and policing functions to ensure that traffic entering the DiffServ domain conforms to the rules specified by the Traffic Conditioning Agreement (TCA), and comply with the service provisioning policy of the domain. Traffic conditioning may range from simple code point re-marking to complex policing and shaping operations.

- **Packet classification**

Packet classification uses a traffic descriptor (for example, the DSCP) to categorize a packet within a specific group in order to define that packet. After the packet has been defined (that is, classified), the packet is then accessible for QoS handling on the network.

Using packet classification, you can partition network traffic into multiple priority levels or classes of service. When traffic descriptors are used to classify traffic, the source agrees to adhere to the contracted terms and the network promises a quality of service. Traffic policers and traffic shapers use the traffic descriptor of the packet (that is, the classification of the packet) to ensure adherence to that agreement.

- **Packet marking**

Packet marking is related to packet classification. Packet marking allows author to classify a packet based on a specific traffic descriptor (such as the DSCP value). This classification can then be used to apply user-defined differentiated services to the packet and to associate a packet with a local QoS group.

Associating a packet with a local QoS group allows users to associate a group ID with a packet. The group ID can be used to classify packets into QoS groups based on prefix, autonomous system, and community string. A user can set up to 64 DSCP values and 100 QoS group markings.

- **Congestion management**

Congestion management (or scheduling) is achieved through traffic scheduling and traffic queueing. When there is network congestion, a scheduling mechanism such as class-based weighted fair queueing (CBWFQ) is used to provide guaranteed bandwidth to the different classes of traffic.

- **Congestion avoidance**

Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance is achieved through packet dropping. Among the more commonly used congestion avoidance mechanisms is Weighted Random Early Detection (WRED).

With WRED and Differentiated Services, the aauthor have the option of allowing WRED to use the DSCP value when WRED calculates the drop probability of a packet.

## Feature Sets

This section lists the feature sets that correspond to the Differentiated Services components listed earlier. These feature sets provide the necessary functionality that allows author to implement Differentiated Services.

This feature set includes the following features:

- **Modular Quality of Service Command-Line Interface.** This feature provides a command-line interface (CLI) structure that allows users to configure class-based QoS features.
- **Class-Based Packet Marking.** This feature provides a user-friendly command-line interface for efficient packet marking by which users can differentiate packets by designating them different identifying values. For example, this feature allows users to mark packets by setting the IP Precedence bits or the IP DSCP in the ToS byte.
- **Traffic Policing.** This feature allows you to limit the input or output transmission rate of a class of traffic based on user-defined criteria. It also enables the system to mark packets by setting the IP Precedence value, the QoS group, or the DSCP value.
- **Class-Based Shaping.** This feature allows author to configure Generic Traffic Shaping (GTS) on a traffic class, specify average rate or peak rate traffic shaping, and configure CBWFQ inside GTS.
- **CBWFQ.** This feature is a scheduling mechanism used to provide a minimum bandwidth guarantee to traffic classes during times of network congestion at an interface.

- **DiffServ Compliant Weighted Random Early Detection.** This feature provides support for the DiffServ standard. It enables WRED to use either the DSCP or the IP Precedence value when calculating the drop probability for a packet. This feature should be used in conjunction with CBWFQ.
- **Enhanced show policy-map interface Command Enhancements for Class-Based Accounting.** The show policy-map interface command now displays information such as the incoming traffic rate, the dropped packet rate, the number of matched packets, and the number of matched bytes for traffic classes that are attached to the specified interface. This feature collects and displays common statistics that are used for billing and accounting purposes.
- **Multiprotocol Label Switching (MPLS) Class of Service Enhancements.** This feature allows the service provider to set the MPLS experimental field instead of overwriting the value in the customer's IP precedence field (the first three bits of the DSCP field in the header of an IP packet).

### 3.1.3 Architectural Design

The architecture designs were develop according to the general architecture of a sample DiffServ implementation concept and conceptual model as shown in Requirement Analysis in Figure 5. Author decided to connect on both of Intranet and Internet connection. As illustrated in Figure 6, Figure 7, Figure 8 and Figure 9 below, the diagram shows the overview of UTP network, TCP flow diagram for internal and external traffic which employs the DS-enabled routers (R), the designation paths for internal and external traffic flows across the switches and the proposed network framework by employing DS routers.

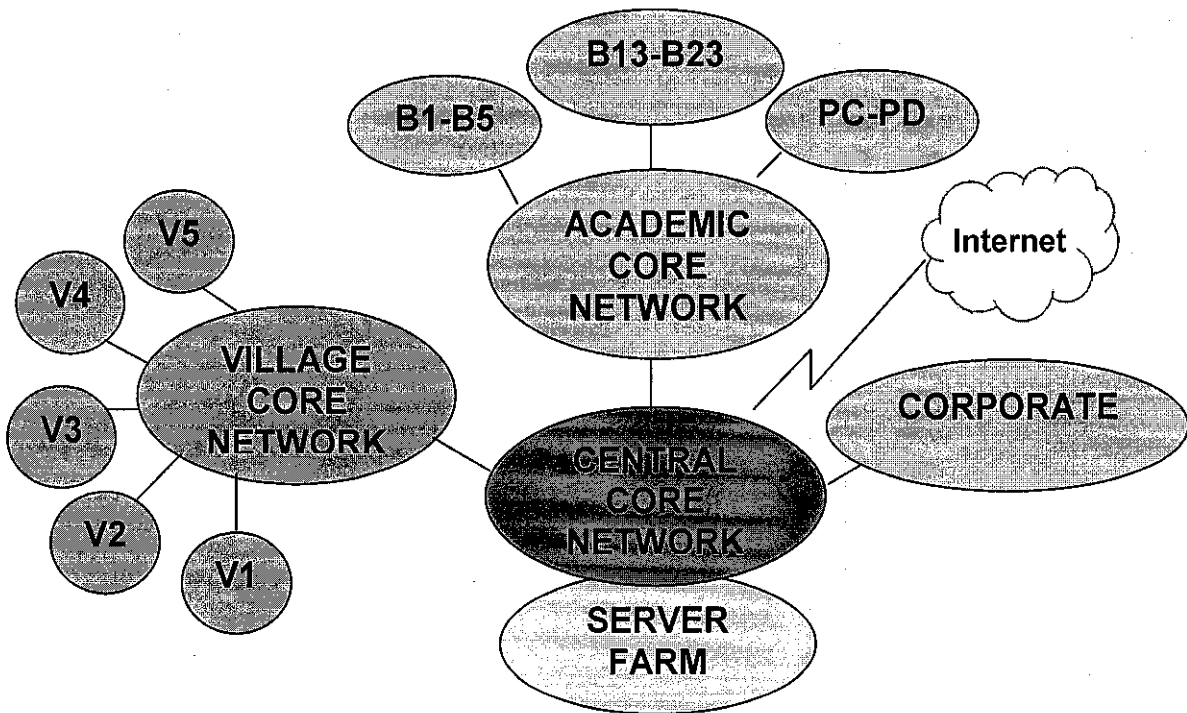


Figure 6: Existing UTP Network Overview



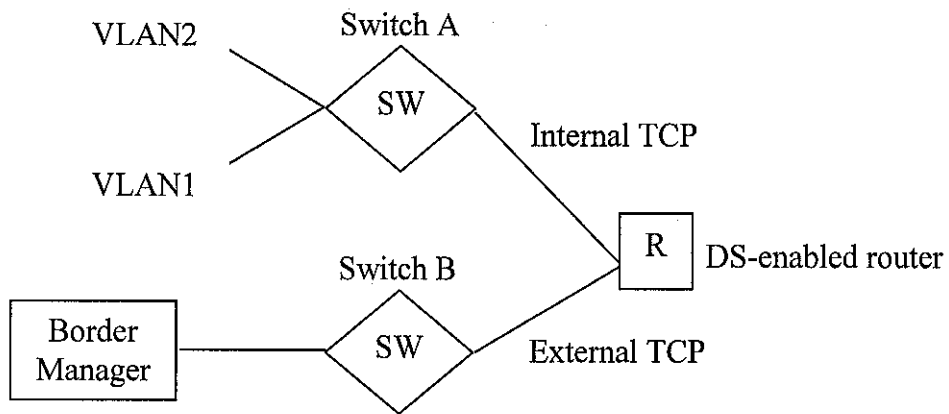


Figure 7: Existing Paths for Internal and External TCP Traffic Flow

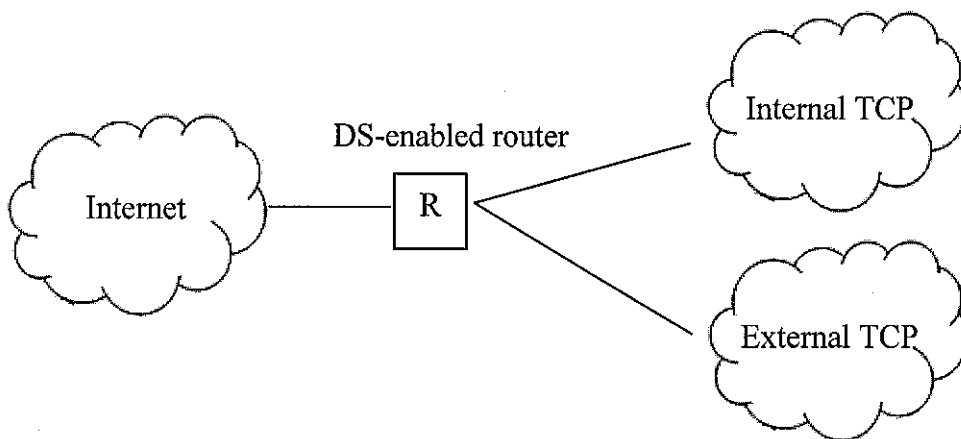


Figure 8: Proposed TCP Flow Diagram

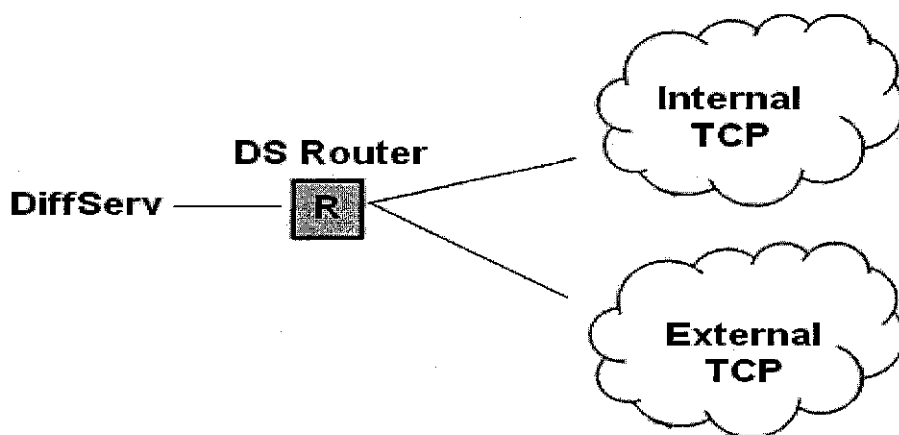


Figure 9: Proposed Network Design by Employing DiffServ Technology

### 3.1.4 Project Module Development

To make sure that the project is easy to develop, the author used the concept of divide and conquer. After the main architecture of the system had been developed, the project was developed according to the module that had been divided. The modules were based on the system architecture, functionality and the requirements. The modules of the project were as follows:

- Router configuration
- Testing and enhancement

One PC will be set up with Windows operating system. Set up the network properties and the tools needed.

#### **Constructing Services Using DiffServ and RIP**

Basically, there are two types of implementation needed to set up and compare in this project, which is the RIP configuration and DiffServ configuration. Both configurations used three CISCO3725 routers. Below figures show how the implementation for both are done.

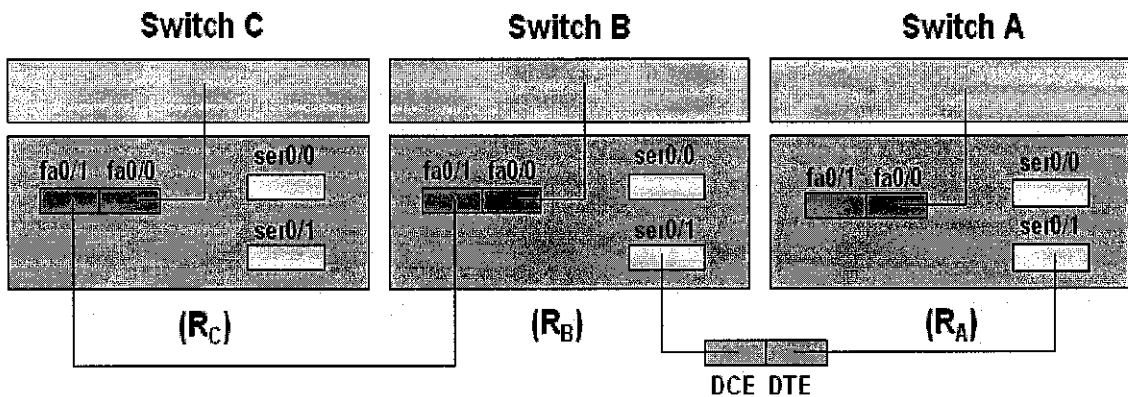


Figure 10: CISCO3725 Setting

The above figure shows the basic setup for both DiffServ and RIP. Each of the routers (R<sub>A</sub>, R<sub>B</sub> and R<sub>C</sub>) is connected through three switches; Switch A, Switch B and Switch C. Table 1 shows the description of each interface connected through these routers.

<b>Interface</b>	<b>Router C (R<sub>C</sub>)</b>	<b>Router B (R<sub>B</sub>)</b>	<b>Router A (R<sub>A</sub>)</b>
fa0/0	223.8.151.1 (int)	172.17.7.1 (int)	10.20.20.1 (int)
fa0/1	199.6.13.2 (ext)	199.6.13.1 (ext)	160.0.53.20 (ext)
ser0/0	-	-	-
ser0/1	-	20.0.0.2 (ext)	20.0.0.1 (ext)

Table 1: CISCO3725 Interfaces

### *Configuring RIP*

The first step for configuring the router for RIP is to set up the interfaces. This includes assigning an IP address and a subnet mask to the interface using the command *ip address A.B.C.D subnet mask*. Next the interface is enabled using the *no shut* command.

The RIP routing protocol is enabled on the router by entering the command *router RIP* at the **Router(config)#** prompt in the privileged EXEC mode. Next network statements are required to declare what networks will be advertised by the RIP routing protocol. To **advertise** the network means the routing table containing the network is shared with its neighbors. The network command requires the use of a **class network address** (class A, class B, class C) after the **network** command. This is call **classful addressing**. A class network address or classful address is the network portion of the address for the particular class of the network. For example VLAN A in our campus network is on the 10.0.1.41 NET, as shown in Figure 11. This is a class A network and the network portion of the address is 10.0.0.0. The structure of the network command is *network [network address]* where the network address is the network IP address for the router; therefore, the command in RIP will be **network 10.0.0.0**.

The following discussion explains how to initialize RIP and how to set the networks attached to the router. Any interfaces that are part of the 10.0.0.0 network will run the RIP routing protocol once these commands are entered. Note

that subnets or subnet masks are not specified in the RIP network command. This is because the class network address is used and all IP addresses in the network (e.g., 10.0.0.0) are enabled to use RIP.

```
Router (config) #router rip
```

```
Router (config-router) #network 10.0.0.0
```

RIP can only be used in contiguous networks, meaning that the networks and routes must have the same class network address. This means that the router addresses for the network connecting the routers must be the same class as the LAN connected to the router. This is shown in Figure 11(a) and (b). LAN A and LAN B have a 10.### address (also called a “10 network” address). The network address connecting the two routers must also be a “10” network address. The IP address for the network connecting the two routers in Figure 10 is 10.0.1.41. This is a “10” network address. The network shown in Figure 11(b) uses the IP address of 192.168.10.0 for the network connecting the two routers. An address of 192.168.10.0 is in the 192.168.10.0 network. This is not part of the “10” network; therefore, the 192.168.10.0 address is not suitable for use in RIP.

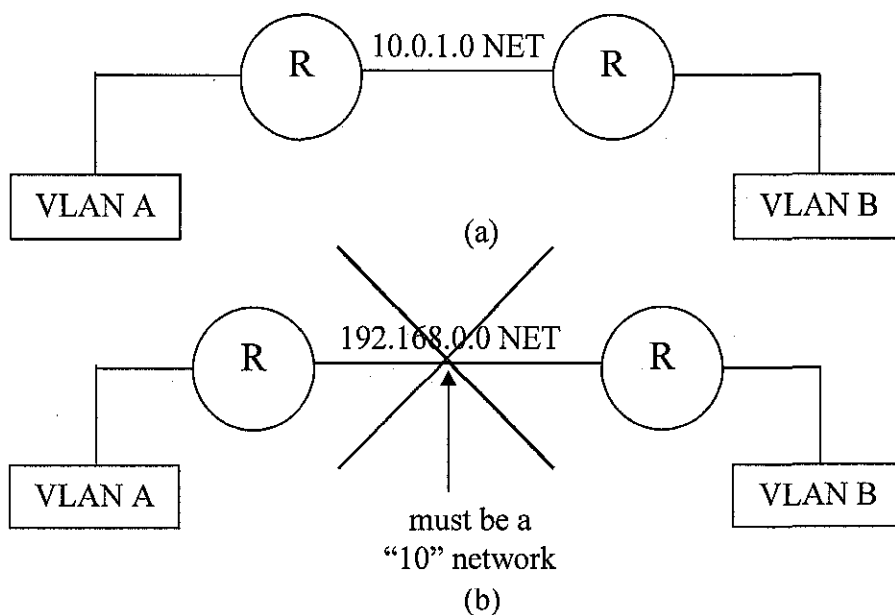


Figure 11: A (a) Contiguous Network and (b) Discontiguous Network.

RIP is relatively simple routing protocol to configure. However, RIP is only good for very small networks that have a limited size to manage the network and not suited for network that need fast convergence. RIP is a standard protocol, not a proprietary protocol, meaning that the use of the protocol is not limited to certain equipments manufacturers. The actual experimentation has been done in the Data Communication lab and the results have been reflected as in the Figure 12, Figure 13 and Figure 14.

```
RIPRouter#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 7 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
  Interface          Send Recv Triggered RIP Key-chain
  FastEthernet0/0    1    1 2
  FastEthernet0/1    1    1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    30.0.0.0
    199.6.13.0
  Routing Information Sources:
    Gateway         Distance   Last Update
    199.6.13.1      120       00:00:13
  Distance: (default is 120)
```

Figure 12: Interface of RIP Configuration within Lab Network

```

RIPRouter#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C 199.6.13.0/24 is directly connected, FastEthernet0/1
R 172.17.0.0/16 [120/1] via 199.6.13.1, 00:00:04, FastEthernet0/1
C 30.0.0.0/8 is directly connected, FastEthernet0/0

```

Figure 13: The Connected Interfaces through Routed IPs

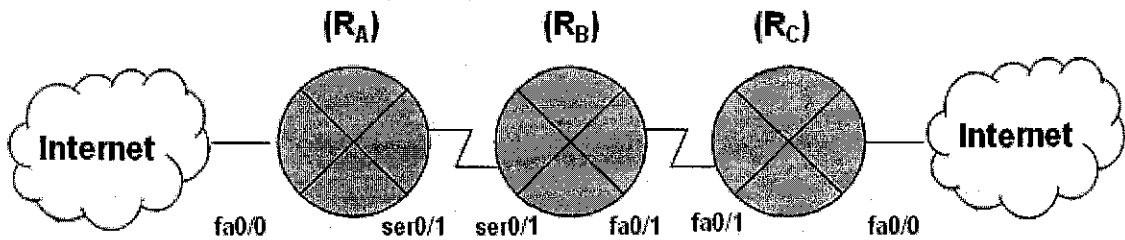


Figure 14: RIP Configuration

## Configuring DiffServ

The following section provides a Diffserv implementation. Figure 15 shows DiffServ implementation with three routers; Router A ( $R_A$ ), Router B ( $R_B$ ) and Router C ( $R_C$ ).

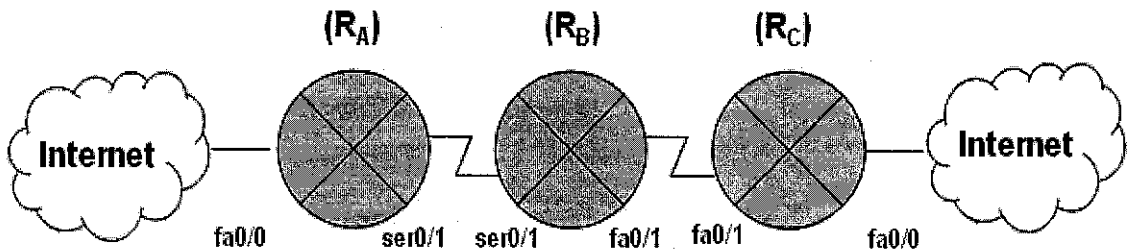


Figure 15: DiffServ Configuration

In this setup, the author tends to create end-to-end QoS to several different types of traffic classes using the CISCO IOS Differentiated Services feature set.

- Voice is considered premium class. The gold class of traffic consists of TACACS (Transmission Access Controller Access System) sessions, along with traffic marked with DSCP values 12 and 14. The silver traffic class consists of Telnet, Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP) sessions. The bronze traffic class consists of web traffic and traffic marked with DSCP values 28 and 30. Anything else is considered as belonging to "best-effort" traffic class.
- The premium class should be forwarded with the lowest delay possible up to a maximum of 500 kbps during periods of congestion. The gold class should be treated preferentially over the silver class, which in turn should be treated preferentially over the bronze class. The gold, silver, and bronze classes should have 35 percent, 25 percent, and 15 percent, respectively, of the interface bandwidth as the minimum bandwidth guarantees. Bronze class should be shaped to 320 kbps, and the best-effort class should be policed to 56 kbps.

- To provision for the various traffic classes, the traffic needs to be classified based on DSCP values in a DiffServ domain. So that traffic can be classified based on DSCP values, the traffic should be pre-marked with the appropriate DSCP values at the time of entering the network.

In Figure 1, the correct place to do this kind of traffic marking is in the incoming direction of Fast Ethernet interface 0/0 of Router A (R<sub>A</sub>) and the incoming direction of Serial interface 0/1 of remote Router B (R<sub>B</sub>). This marking can be achieved through an input service policy. Table 2 lists the DSCP values used to mark different classes of traffic entering into the designed network.

<b>Traffic Class</b>	<b>Traffic Type</b>	<b>DSCP Value</b>
Premium	Voice	46
Gold	TACACS	10
Silver	Telnet	18
	SMTP	20
	FTP	22
Bronze	HTTP	26

Table 2: DSCP Values for Traffic Classes and Traffic Types [7]



The configurations demonstrate how marking, shaping, policing, and monitoring is done through Modular QoS Command-Line Interface (MQC) and this is attached under Appendices (Refer to Appendix B).

### **3.1.5 Operation and maintenance**

Finally, in operation and maintenance, it prepares the project to be delivered. The project operation and maintenance base on this project is involving system integration, implementation and enhancement where applicable. Finally the steps taken were project maintenance and monitoring.

## **3.2 TOOLS / EQUIPMENTS**

The tools needed to implement the configuration are:

- 1 Ethereal - Data can be captured "off the wire" from a live network connection, or read from a capture file.
- 2 Three CISCO3725 routers
- 3 Switches

## **CHAPTER 4**

### **RESULT AND DISCUSSION**

In this chapter, the author had written the result and the discussion for the analysis that had been made especially on the requirement analysis. On this chapter, author will also discuss the output that will be made throughout the project implementation and research itself.

The requirement analysis focuses on the system requirement, which include the hardware. The overall output will be triggered through the observation on the packets sent, throughput, and time-to-completion in the Ethereal interface. Each of data packets will be analyzed separately based on the internal and external flow of TCP. As we know the UTP applies RIP (Routing Information Protocol) as it's routing protocol. This routing protocol has been explained in the Chapter 3: Methodology/ Project Work under Project Development phase. It is based on the dynamic routing protocols. Same goes to the explanation for DiffServ technology.

#### **4.1 RIP Analysis**

Below results show how the data packets have been captured using Ethereal and the explanation tells the scenario occurred. Basically, the data sized 175MB have been copied and sent through the three CISCO3725 routers. Two hosts are available between these routers, which act as a sender and receiver for the assigned data. The hosts labeled as 10.20.20.15 and 223.8.151.10.

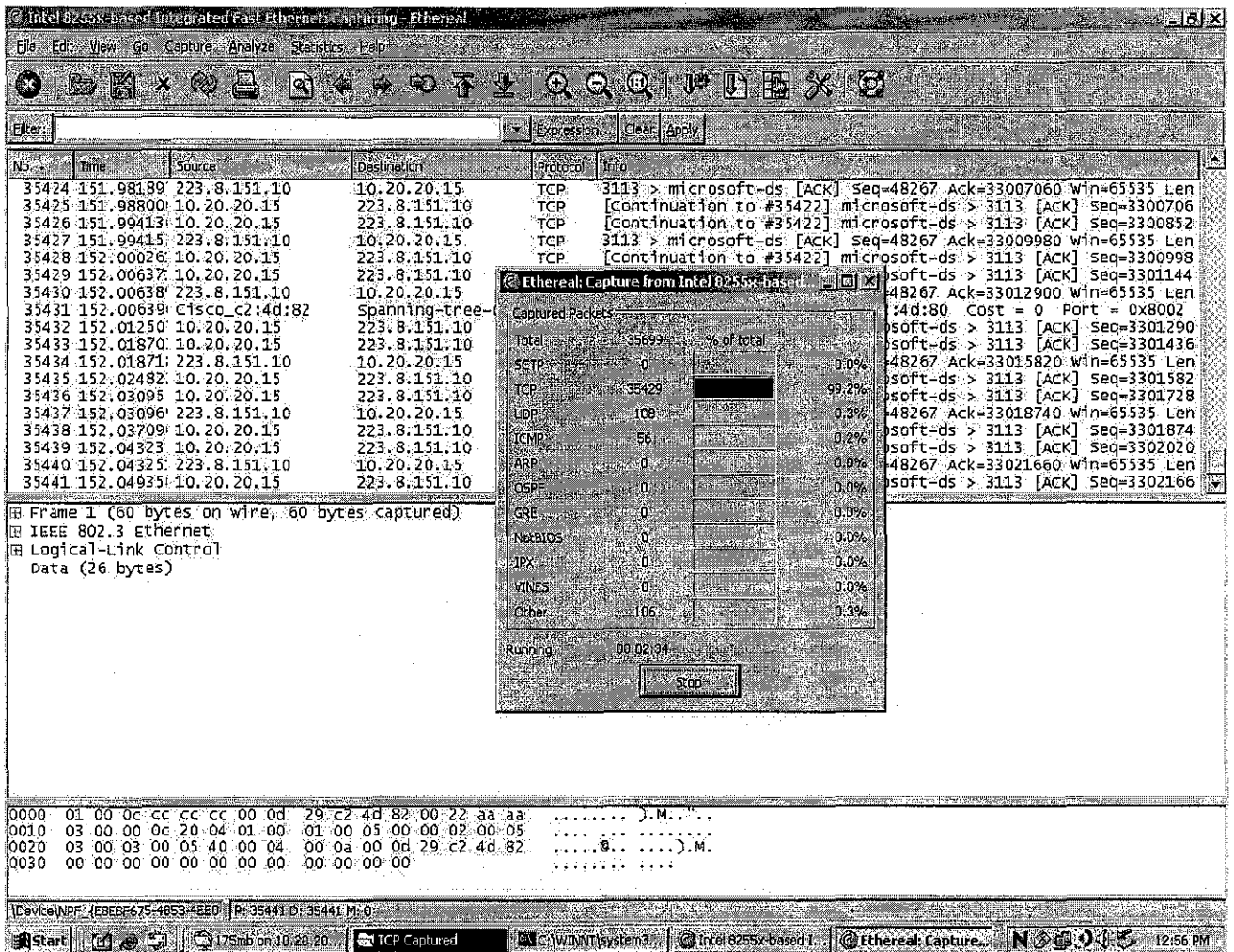


Figure 16: TCP Data Packets Captured Based on RIP Configuration

As shown on the above figure, almost 99% of network process captured based on TCP layer because it involves the transmission and receiving processes. While RIP setup is running, the DiffServ setting has been removed from routers' interfaces. This is to make sure that the RIP configuration running smoothly without any other restrictions. Below figure shows the details of successful data that has been captured.

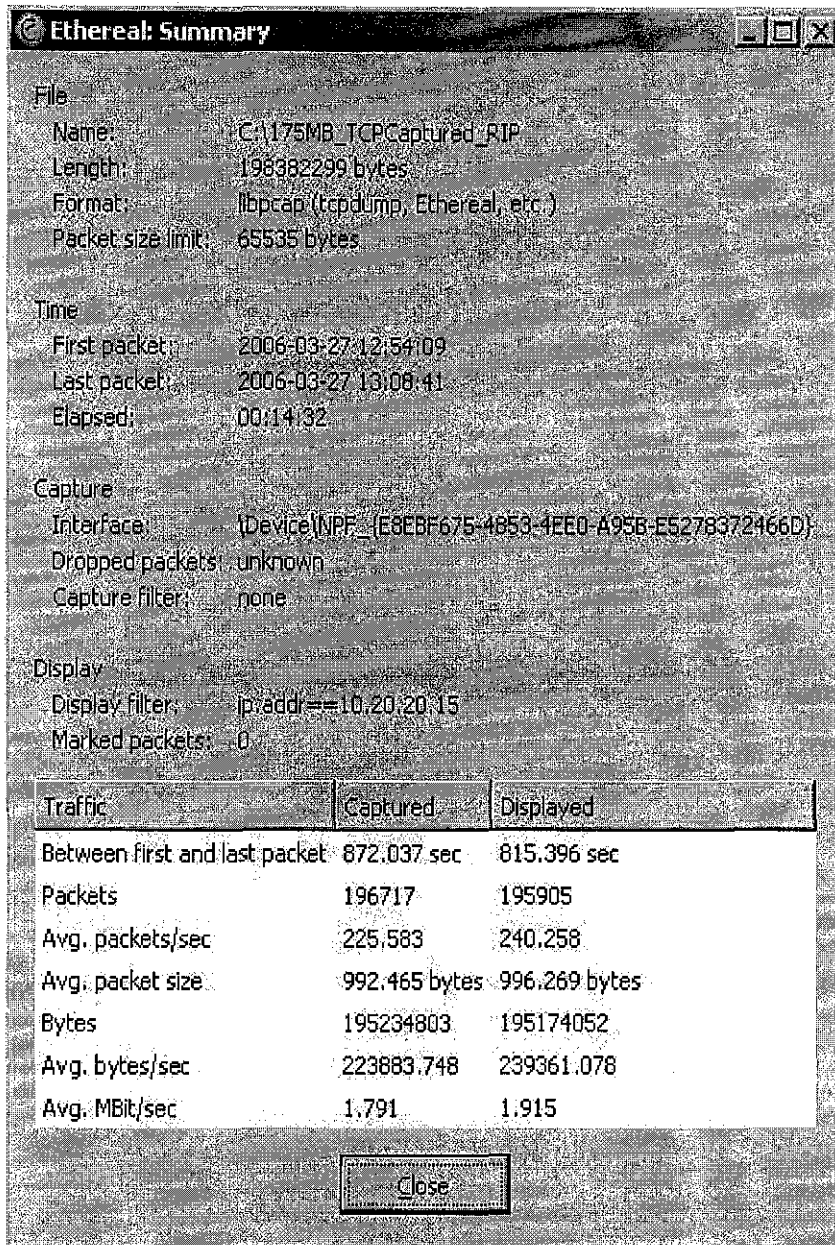


Figure 17: Summary of TCP Packets Captured Based on RIP Configuration

The host that captured the data packets has been labeled as 10.20.20.15, which filters all the activities done through downloading process from the other host. The downloading process – the delay; started at 12.54 p.m. and finished at 1.08 pm which equivalent to 14 minutes and 32 seconds (872.037 seconds). Average packets that have been captured per second are 225.583. This is also known as throughput; the rate at which packets go through the network.

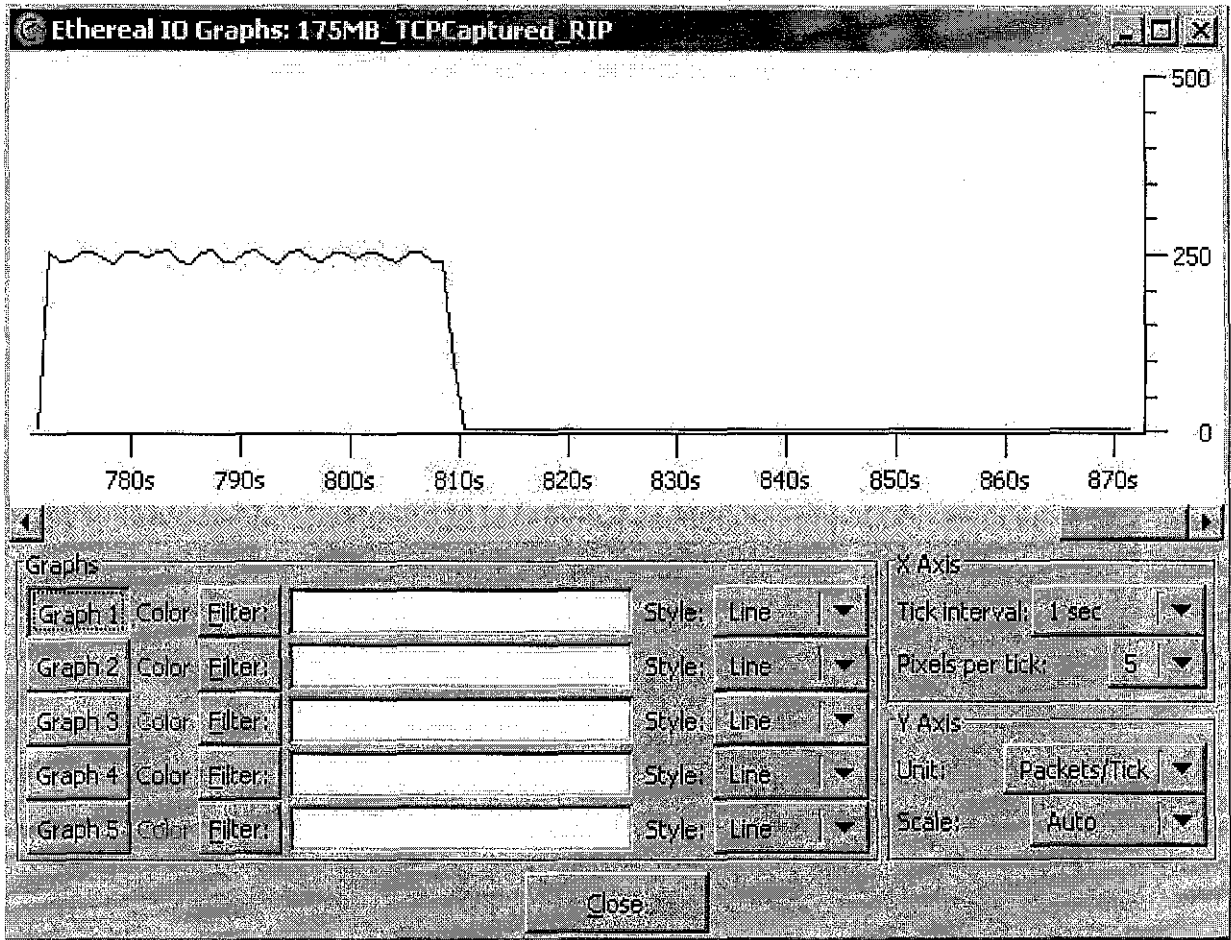


Figure 18: Ethereal IO Graphs Captured Based On RIP Setting

The graph shows the number of packets sends per second and Figure 16 already explains this on the previous page. It is estimated that 250 packets have been captured every seconds. The elapsed time need to complete the capturing process between the first and the last packets are 14 minutes and 32 seconds.

## 4.2 DiffServ Analysis

There is a slightly different between results obtained from RIP with the results captured by DiffServ setting. The different is on the delay and the throughput itself. For this purpose, RIP configuration has been cleared off from the routers' interfaces. Figure 18 and Figure 19 explained how the scenario occurred.

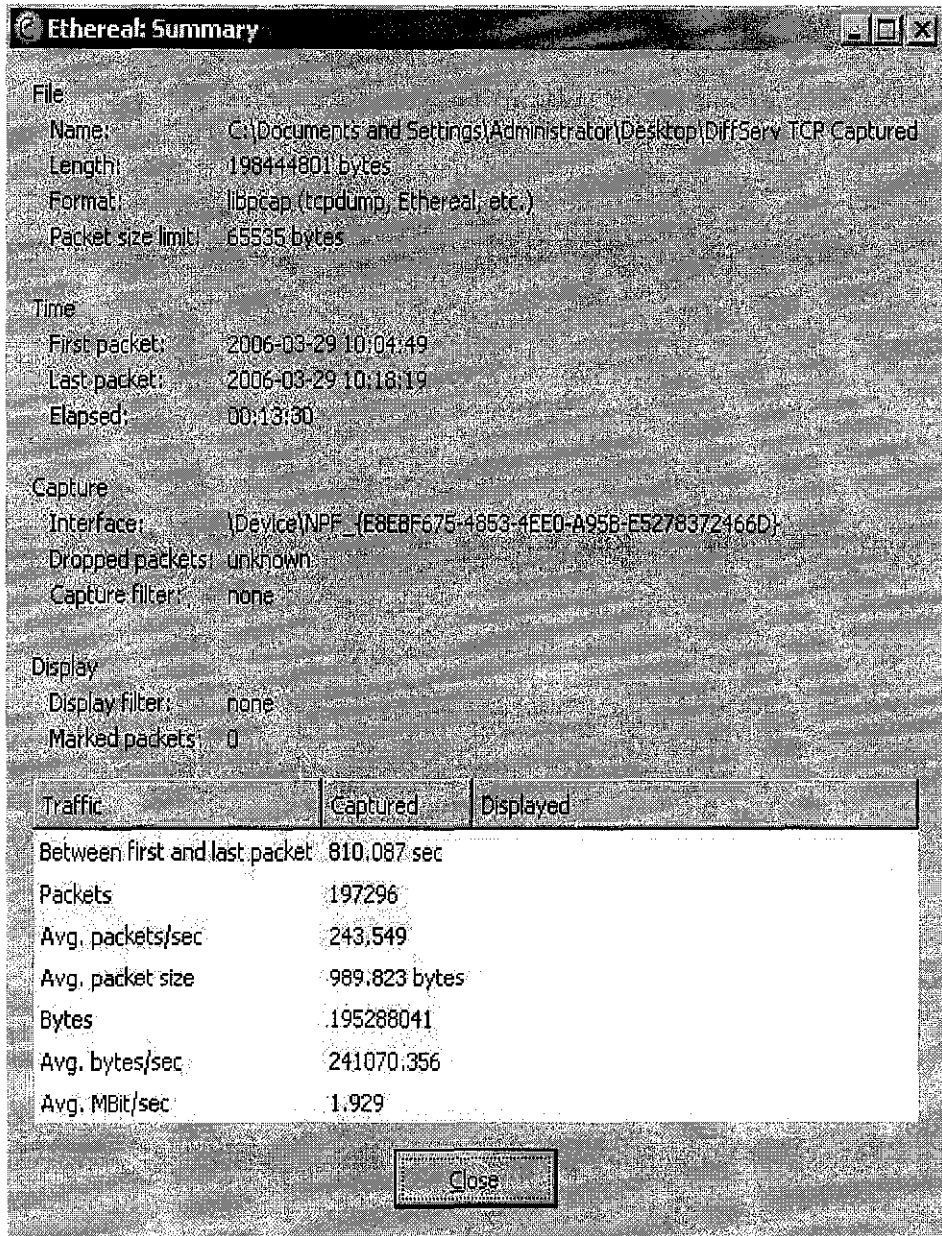


Figure 19: Summary of TCP Packets Captured Based on DiffServ Configuration

If compared to RIP analysis, the time taken to complete the downloading activity under DiffServ setting is 61.95 seconds (1.0325 minutes), which is considered much faster. Average packets sent per second or throughput is 243.549 packets, which is much higher than TCP packets captured under RIP configuration. The graph below explains the packets' trend.

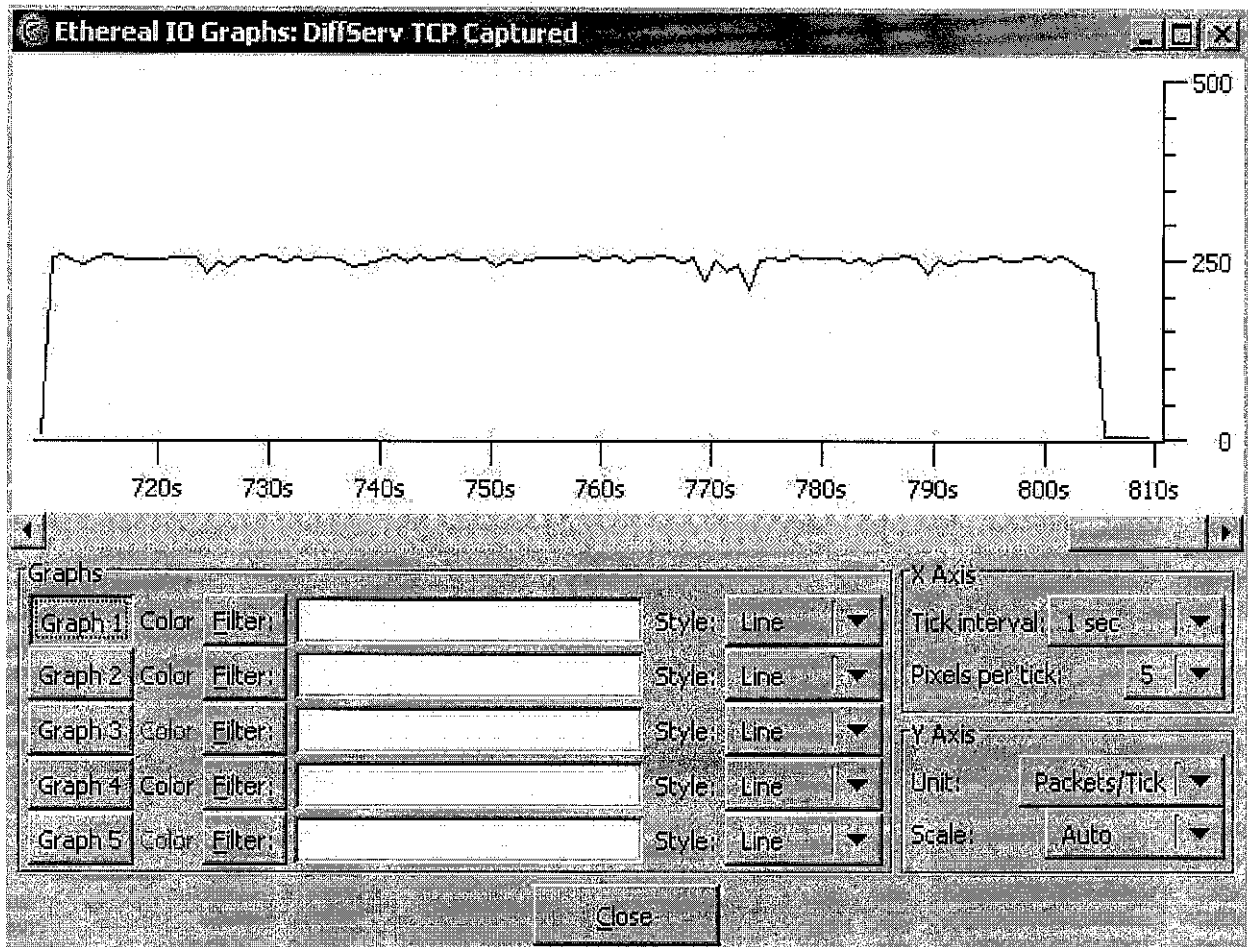


Figure 20: Ethereal IO Graphs Captured Based On DiffServ Setting

The packets are consistently downloaded within 13 minutes and 30 seconds, which can be concluded as a bit faster than the TCP packets downloaded lies under RIP setting.

The author can say that the project has achieved its objectives as the proven shows by both analysis. The DiffServ served better packet forwarding compared to RIP in terms of the delay and the throughput. The routers are indeed fulfilling the QoS requirements at

which it can deliver multiple classes of service. It is scalable which can support emerging network intensive and mission critical applications.

Table 3 below shows the results for both RIP and DiffServ based on Ethereal analysis.

Type of formal metrics to measure QoS	Result	
	RIP	DiffServ
1) Delay	14 minutes 32 seconds	13 minutes 30 seconds
2) Throughput	225.583 packets/sec	243.549 packets/sec

Table 3: The Ethereal Results

As written above, the author can conclude that DiffServ is better than RIP in terms of its capabilities to reduce delay and maximize output. More packets are sent in a second with faster time-to-completion.



## CHAPTER 5

### CONCLUSION AND RECOMMENDATIONS

In this paper, the author introduced the design and implementation of DS-enabled routers in the Internet focusing both on internal and external of TCP. The author also conducted a small scale lab network based on provided DS-enabled routers. Through these analyses, the author not only verified the correctness of the design and implementation, but also studied some DiffServ QoS features within the lab network, such as multiple classes of services; QoS guarantees also various formal metric likes delay and throughput.

Based on the analysis performed, it is proven that DiffServ can satisfy different QoS requirements. With Differentiated Services, the network tries to deliver a particular kind of service based on the QoS specified by each packet. It is proposed and suggested that the UTP Management can adapt this technology to provide better Internet connectivity within UTP LAN. It is low-cost method as we can only focus on the existence routers resided in the data center.

As proposed by ITMS Unit, the network team leads by their Network Administrator is in the process of upgrading the bandwidth to cater to the degradation of UTP network performance. Alternatively, they aimed to have several gateway to reduce the congestion and the possibility of wireless accessibility by ISP. To make it happens, they hope students stop hosting any non-educated activities such as servers hosting, video streaming, do not abuse proxy, update antivirus regularly, release the Internet bandwidth whenever not in use and the most important thing use the provided Internet for only educational purposes.

To have DiffServ successfully implemented, the author suggests that the ITMS Unit, first thing first; should enforce proper configuration on proxy and with proper bandwidth management policy. Used antivirus solution for such DDoS attack, worms and viruses spread through network. In term of routing design, isolate all students out and force them to use link to go out Internet only and the major link use for other important applications

or executives group of user. For better Intranet connection, can force user accessing server through LAN and this depend on server performance and routing.

## LIST OF REFERENCES

- [1] Roland Bless and Klaus Wehrle. Evaluation of differentiated services using an implementation under linux. In IWQoS'99, London, 1999.
- [2] R. Braden and L. Zhang. Resource ReSerVation Protocol (RSVP) - Verion 1 Functional Specification. RFC 2205, September 1997.
- [3] S.Blake et. al. An Architecture for Differentiated Services. RFC 2475, December 1998.
- [4] S.Shenker et. al. Integrated Services in the Internet Architecture: an Overview. RFC 1633, June 1994.
- [5] F.Baker, D.Black, S.Blake, and K.Nichols. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474, December 1998.
- [6] Jun Wang, Klara Nahrstedt and Yuxin Zhou. Design and Implementation of DiffServ Routers in OPNET, Department of Computer Science, University of Illinois at Urbana-Champaign.
- [7] <http://www.cisco.com/warp/public/105/dscpvalues.html>
- [8] <http://www.ciscopress.com/articles/printerfriendly.asp?p=417092>

## **APPENDICES**

<b>Appendix A</b>	<b>Interview Questions</b>
<b>Appendix B</b>	<b>DiffServ Implementation</b>
<b>Appendix C</b>	<b>Configuring RIP</b>
<b>Appendix D</b>	<b>Implementing Quality of Services Policies with DSCP</b>
<b>Appendix E</b>	<b>Quality-Of-Service The Differentiated Services Model</b>
<b>Appendix F</b>	<b>Which Routing Protocol Should My Network Use?</b>

**APPENDIX A**

**INTERVIEW QUESTIONS**

## INTERVIEW QUESTIONS

Below are the questions prepared for UTP Network Administrator, Mr. Arfaishah:

1. What are the causes lead to low Internet connectivity within the UTP campus? Why is it happens frequently?
2. Is it because of ISP or the network devices fail to accommodate high-bandwidth?
3. How do you assign a bandwidth usage to certain task? Internet connectivity is much better at new academic complex as compared to students' villages. Why?
4. What are the solutions need to encounter such problem as mentioned above? Justify through theory and technical aspects.
5. How do you prioritize the traffic comes in?
6. Based on the basic architecture of designing a network traffic, it should be 80 (Internal):20 (External). Do UTP apply this design?

**APPENDIX B**

**DIFFSERV IMPLEMENTATION**

# DIFFSERV IMPLEMENTATION FOR ROUTER A, ROUTER B AND ROUTER C

## Router A (RA) Configuration

```
ROUTER_A#show running-config
Building configuration...

Current configuration : 4783 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname ROUTER_A
!
boot system flash c3725-js-mz.122-15.T2.bin
logging queue-limit 100
enable secret 5 $1$VYTo$fuaEKUJNy4fhjs6/n4mi20
enable password 7 04521F16
!
ip subnet-zero
!
!
no ip dhcp conflict logging
ip dhcp excluded-address 10.20.20.0 10.20.20.10
!
ip dhcp pool amie
    network 10.20.20.0 255.255.255.0
    default-router 10.20.20.1
!
mpls ldp logging neighbor-changes
!
!
voice rtp send-recv
!
voice service pots
!
voice service voip
    h323
!
!
!
!
!
```



```

no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
class-map match-all voip
class-map match-all gold
  match ip dscp af11 af12 af13
class-map match-all ef
  match access-group 101
class-map match-all EF
  match access-group 101
class-map match-all AF21
  match access-group 108
class-map match-all AF23
  match access-group 110
class-map match-all AF22
  match access-group 109
class-map match-all bronze
  match ip dscp af31 af32 af33
class-map match-all platinum
  match ip dscp ef
class-map match-all silver
  match ip dscp af21 af22 af23
class-map match-all best-effort
  match access-group 105
class-map match-all AF3
  match access-group 104
class-map match-all af1
class-map match-all AF1
  match access-group 102
class-map match-all premium
  match ip dscp ef
!
!
policy-map VOIP
class premium
  priority 500
class gold
  bandwidth percent 50
class silver
  shape average 320000
  bandwidth percent 35

```

```

class bronze
bandwidth percent 15
shape average 320000
class best-effort
police cir 56000 bc 1750 be 1750
conform-action set-dscp-transmit default
exceed-action drop
violate-action drop
class platinum
priority 500
policy-map SETDSCP
class EF
set dscp ef
class AF1
set dscp af11
class AF21
set dscp af21
class AF22
set dscp af22
class AF23
set dscp af23
class AF3
set dscp af31
!
!
!
!
interface FastEthernet0/0
ip address 10.20.20.1 255.255.255.0
ip helper-address 10.0.0.1
no ip mroute-cache
load-interval 60
speed auto
half-duplex
no mop enabled
!
interface Serial0/0
ip address 10.0.0.2 255.255.255.0
no ip mroute-cache
clockrate 2000000
no fair-queue
!
interface FastEthernet0/1
ip address 160.0.53.201 255.255.252.0
no ip mroute-cache

```

```

speed 100
full-duplex
!
interface Serial0/1
bandwidth 2000
ip address 20.0.0.1 255.0.0.0
no ip mroute-cache
load-interval 60
!
interface ATM2/0
ip address 192.22.74.3 255.255.255.0
no ip mroute-cache
atm clock INTERNAL
atm esi-address 345678901234.12
no atm ilmi-keepalive
pvc 1/5 qsaal
!
pvc 1/16 ilmi
!
pvc 1/32
protocol ip 192.22.74.2 broadcast
!
!
interface Dialer1
no ip address
!
router rip
network 10.0.0.0
network 172.17.0.0
network 192.22.74.0
neighbor 192.22.74.4
!
ip http server
ip classless
ip route 160.0.52.0 255.255.252.0 160.0.55.254
ip route 172.17.0.0 255.255.0.0 20.0.0.2
ip route 192.22.74.0 255.255.255.0 192.22.74.4
ip route 199.6.13.0 255.255.255.0 20.0.0.2
ip route 223.8.151.0 255.255.255.0 199.6.13.2
!
!
!
access-list 101 permit udp any any range 16384 32768
access-list 101 permit tcp any any eq tacacs
access-list 101 permit tcp any any eq www
access-list 102 permit tcp any any eq tacacs
access-list 104 permit tcp any any eq www

```

```
access-list 105 permit ip any any
access-list 108 permit tcp any any eq telnet
access-list 109 permit tcp any any eq smtp
access-list 110 permit tcp any any eq ftp
!
!
snmp-server community public RO
snmp-server enable traps tty
tftp-server flash:c3725-js-mz.122-15.T2.bin
!
call rsvp-sync
!
voice-port 1/0/0
!
voice-port 1/0/1
echo-cancel coverage 32
!
voice-port 1/1/0
!
voice-port 1/1/1
!
!
mgcp profile default
!
!
!
dial-peer cor custom
!
!
!
dial-peer voice 1 pots
destination-pattern 5551001
port 1/0/1
!
dial-peer voice 3 voip
destination-pattern 556....
session target ipv4:20.0.0.2
!
dial-peer voice 2 voip
destination-pattern 4444444
session target ipv4:172.17.7.10
codec g711ulaw
!
dial-peer voice 5 voip
```

```
destination-pattern 777....
session target ipv4:20.0.0.2
!
dial-peer voice 6 voip
destination-pattern 9999999
session target ipv4:172.17.7.10
codec g711ulaw
!
dial-peer voice 7 voip
destination-pattern 1111111
session target ipv4:10.20.20.12
!
dial-peer voice 4 voip
destination-pattern 555....
session protocol sipv2
session target ipv4:20.0.0.1
!
dial-peer voice 8 voip
destination-pattern 1112222
session protocol sipv2
session target ipv4:10.20.20.15
!
dial-peer voice 9 voip
destination-pattern 5551111
session protocol sipv2
session target ipv4:10.20.20.120
!
sip-ua
!
!
line con 0
line aux 0
line vty 0 4
password 7 071A355C
login
!
end
```

## Router B (R<sub>B</sub>) Configuration

```
ROUTER_B#show running-config
Building configuration...

Current configuration : 3136 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ROUTER_B
!
logging queue-limit 100
enable secret 5 $1$TXII$dZr8JUHMz/vbB/XPoHLNP.
enable password cosig.2005
!
ip subnet-zero
!
!
!
mpls ldp logging neighbor-changes
!
!
!
!
!
!
!
!
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
class-map match-all gold
 match ip dscp af11 af12 af13
class-map match-all bronze
 match ip dscp af31 af32 af33
class-map match-all platinum
```

```

match ip dscp ef
class-map match-all silver
match ip dscp af21 af22 af23
class-map match-all best-effort
match ip dscp default
!
!
policy-map AVVID
class silver
bandwidth percent 35
random-detect dscp-based
random-detect dscp 18 20 40 10
random-detect dscp 20 20 40 30
random-detect dscp 22 2 3 3
class gold
bandwidth percent 50
random-detect dscp-based
random-detect dscp 10 20 40 10
random-detect dscp 14 20 40 20
random-detect dscp 20 12 40 15
class bronze
bandwidth percent 15
random-detect dscp-based
random-detect dscp 26 20 40 10
random-detect dscp 28 20 40 20
random-detect dscp 30 20 40 30
class platinum
priority 500
!
!
!
!
interface FastEthernet0/0
ip address 172.17.7.1 255.255.0.0
speed auto
full-duplex
no mop enabled
!
interface Serial0/0
no ip address
ip broadcast-address 0.0.0.0
shutdown
clockrate 2000000
!

```

```

interface FastEthernet0/1
bandwidth 2000
ip address 199.6.13.1 255.255.255.0
no ip mroute-cache
load-interval 60
speed auto
full-duplex
!
interface Serial0/1
ip address 20.0.0.2 255.0.0.0
no ip mroute-cache
clockrate 2000000
!
interface ATM2/0
ip address 192.22.74.4 255.255.255.0
atm clock INTERNAL
atm esi-address 345678901234.12
no atm ilmi-keepalive
pvc 1/5 qsaal
!
pvc 1/32
protocol ip 192.22.74.2 broadcast
!
!
router rip
network 10.0.0.0
network 172.17.0.0
network 192.22.74.0
neighbor 192.22.74.3
!
no ip http server
ip classless
ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 192.22.74.0 255.255.255.0 172.17.7.0
ip route 192.22.74.0 255.255.255.0 192.22.74.3
ip route 223.8.151.0 255.255.255.0 199.6.13.2
!
!
!
!
!
snmp-server community public RO
snmp-server enable traps tty
!
call rsvp-sync

```



```
!  
voice-port 1/0/0  
!  
voice-port 1/0/1  
!  
voice-port 1/1/0  
!  
voice-port 1/1/1  
!  
!  
mgcp profile default  
!  
!  
!  
dial-peer cor custom  
!  
!  
!  
dial-peer voice 1 pots  
destination-pattern 5561002  
port 1/0/1  
!  
dial-peer voice 2 voip  
destination-pattern 555....  
session target ipv4:20.0.0.1  
!  
dial-peer voice 3 voip  
destination-pattern 1234567  
session target ipv4:172.17.7.10  
codec g711ulaw  
!  
dial-peer voice 4 pots  
destination-pattern 7777777  
port 1/0/1  
!  
dial-peer voice 5 voip  
destination-pattern 888....  
session target ipv4:10.20.20.11  
codec g711ulaw  
!  
dial-peer voice 9 voip  
destination-pattern 9999999  
session target ipv4:10.20.20.11  
codec g711ulaw  
!
```

```
!  
line con 0  
line aux 0  
line vty 0 4  
password cosig.2005  
login  
!  
end
```

## Router C (R<sub>C</sub>) Configuration

```
ROUTER_C#show running-config
Building configuration...

Current configuration : 3151 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname ROUTER_C
!
boot system flash c3725-js-mz.122-15.T2.bin
logging rate-limit 10 except errors
no logging console
enable secret 5 $1$9KJJ$1Sh56KqPTSAF9zJDgpeFW.
enable password 7 03145A1815182E5E4A
!
clock timezone UTP -8
ip subnet-zero
!
!
ip dhcp excluded-address 160.0.57.245 160.0.57.254
ip dhcp excluded-address 160.0.57.0 160.0.57.204
ip dhcp smart-relay
!
ip dhcp pool LABNET
  network 160.0.57.0 255.255.255.0
  dns-server 160.0.226.202
  default-router 160.0.57.205
  lease 100
  update arp
!
ip dhcp pool Datacomm
  lease 100
!
ip dhcp pool LABNET2
  network 160.0.56.0 255.255.255.0
  default-router 160.0.56.205
  lease 100
!
!
```

```

!
!
class-map match-all gold
  match ip dscp af11 af12 af13
class-map match-all EF
  match access-group 101
class-map match-all AF21
  match access-group 108
class-map match-all AF23
  match access-group 110
class-map match-all AF22
  match access-group 109
class-map match-all bronze
  match ip dscp af31 af32 af33
class-map match-all platinum
  match ip dscp ef
class-map match-all silver
  match ip dscp af21 af22 af23
class-map match-all best-effort
  match access-group 105
class-map match-all AF3
  match access-group 104
class-map match-all AF1
  match access-group 102
class-map match-all premium
  match ip dscp ef
!
!
policy-map VOIP
  class platinum
    priority 500
  class gold
    bandwidth percent 50
  class bronze
    shape average 320000
    bandwidth percent 15
  class silver
    bandwidth percent 9
  class AF3
policy-map SETDSCP
  class EF
    set dsep ef
  class AF1
    set dscp af11
  class AF3
    set dscp af31

```

```

class AF21
  set dscp af21
class AF22
  set dscp af22
class AF23
  set dscp af23
!
!
!
interface FastEthernet0/0
  ip address 223.8.151.1 255.255.255.0
  service-policy input SETDSCP
  load-interval 60
  speed 100
  full-duplex
!
interface Serial0/0
  no ip address
  no ip mroute-cache
  shutdown
  clockrate 2000000
  no fair-queue
!
interface BRI0/0
  no ip address
  shutdown
!
interface FastEthernet0/1
  ip address 199.6.13.2 255.255.255.0
  service-policy output VOIP
  load-interval 60
  duplex auto
  speed auto
!
interface Serial0/1
  bandwidth 2000
  no ip address
  no ip mroute-cache
  shutdown
  clockrate 2000000
!
interface BRI0/1
  no ip address
  shutdown
!

```

```
router rip
 network 223.8.151.0
 !
ip http server
ip classless
ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 20.0.0.0 255.0.0.0 199.6.13.0
ip route 172.17.7.0 255.255.255.0 199.6.13.1
ip route 199.6.13.0 255.255.255.0 20.0.0.0
ip route 223.8.151.0 255.255.255.0 199.6.13.0
 !
 !
access-list 101 permit udp any any range 16384 32768
access-list 102 permit tcp any any eq tacacs
access-list 104 permit tcp any any eq www
access-list 105 permit tcp any any
access-list 108 permit tcp any any eq telnet
access-list 109 permit tcp any any eq smtp
access-list 110 permit tcp any any eq ftp
snmp-server community public RO
snmp-server enable traps tty
 !
line con 0
 exec-timeout 0 0
 transport output none
line aux 0
line vty 0 4
 password 7 0213104B
 login
 !
 !
end
```

# **APPENDIX C**

## **CONFIGURING RIP**

# Configuring RIP

---

This chapter describes how to configure RIP. For a complete description of the RIP commands that appear in this chapter, refer to the “RIP Commands” chapter of the *Network Protocols Command Reference, Part 1*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

The Routing Information Protocol (RIP) is a relatively old, but still commonly used, interior gateway protocol (IGP) created for use in small, homogeneous networks. It is a classical distance-vector routing protocol. RIP is documented in RFC 1058.

RIP uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. The Cisco IOS software sends routing information updates every 30 seconds; this process is termed *advertising*. If a router does not receive an update from another router for 180 seconds or more, it marks the routes served by the nonupdating router as being unusable. If there is still no update after 240 seconds, the router removes all routing table entries for the nonupdating router.

The metric that RIP uses to rate the value of different routes is *hop count*. The hop count is the number of routers that can be traversed in a route. A directly connected network has a metric of zero; an unreachable network has a metric of 16. This small range of metrics makes RIP an unsuitable routing protocol for large networks.

If the router has a default network path, RIP advertises a route that links the router to the pseudonetwork 0.0.0.0. The network 0.0.0.0 does not exist; RIP treats 0.0.0.0 as a network to implement the default routing feature. The Cisco IOS software will advertise the default network if a default was learned by RIP, or if the router has a gateway of last resort and RIP is configured with a default metric.

RIP sends updates to the interfaces in the specified networks. If an interface’s network is not specified, it will not be advertised in any RIP update.

Cisco’s implementation of RIP Version 2 supports plain text and MD5 authentication, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs).

For protocol-independent features, which also apply to RIP, see the chapter “Configuring IP Routing Protocol-Independent Features” in this document.

## IP Configuration Task List

To configure RIP, complete the tasks in the following sections. You must enable RIP. The remaining tasks are optional.

- Enable RIP
- Allow Unicast Updates for RIP



- Apply Offsets to Routing Metrics
- Adjust Timers
- Specify a RIP Version
- Enable RIP Authentication
- Disable Route Summarization
- Run IGRP and RIP Concurrently
- Disable the Validation of Source IP Addresses
- Enable or Disable Split Horizon
- Configure Interpacket Delay

For information about the following topics, see the “Configuring IP Routing Protocol-Independent Features” chapter:

- Filtering RIP information
- Key management (available in RIP Version 2)
- VLSM

## nable RIP

To enable RIP, perform the following tasks, starting in global configuration mode:

Task	Command
<b>Step 1</b> Enable a RIP routing process, which places you in router configuration mode.	<b>router rip</b>
<b>Step 2</b> Associate a network with a RIP routing process.	<b>network <i>network-number</i></b>

## llow Unicast Updates for RIP

Because RIP is normally a broadcast protocol, in order for RIP routing updates to reach nonbroadcast networks, you must configure the Cisco IOS software to permit this exchange of routing information. To do so, perform the following task in router configuration mode:

Task	Command
Define a neighboring router with which to exchange routing information.	<b>neighbor <i>ip-address</i></b>

To control the set of interfaces with which you want to exchange routing updates, you can disable the sending of routing updates on specified interfaces by configuring the **passive-interface** command. See the discussion on filtering in the “Filter Routing Information” section in the “Configuring IP Routing Protocol-Independent Features” chapter.

## Apply Offsets to Routing Metrics

An offset list is the mechanism for increasing incoming and outgoing metrics to routes learned via RIP. This is done to provide a local mechanism for increasing the value of routing metrics. Optionally, you can limit the offset list with either an access list or an interface. To increase the value of routing metrics, perform the following task in router configuration mode:

Task	Command
Apply an offset to routing metrics.	<code>offset-list [access-list-number   name] {in   out} offset [type number]</code>

## Adjust Timers

Routing protocols use several timers that determine such variables as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs. You can make the following timer adjustments:

- The rate (time in seconds between updates) at which routing updates are sent
- The interval of time (in seconds) after which a route is declared invalid
- The interval (in seconds) during which routing information regarding better paths is suppressed
- The amount of time (in seconds) that must pass before a route is removed from the routing table
- The amount of time for which routing updates will be postponed

It also is possible to tune the IP routing support in the software to enable faster convergence of the various IP routing algorithms, and, hence, quicker fallback to redundant routers. The total effect is to minimize disruptions to end users of the network in situations where quick recovery is essential.

To adjust the timers, perform the following task in router configuration mode:

Task	Command
Adjust routing protocol timers.	<code>timers basic update invalid holddown flush [sleepime]</code>

## Specify a RIP Version

Cisco's implementation of RIP Version 2 supports authentication, key management, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs). Key management and VLSM are described in the chapter "Configuring IP Routing Protocol-Independent Features."

By default, the software receives RIP Version 1 and Version 2 packets, but sends only Version 1 packets. You can configure the software to receive and send only Version 1 packets. Alternatively, you can configure the software to receive and send only Version 2 packets. To do so, perform the following task in router configuration mode:

Task	Command
Configure the software to receive and send only RIP Version 1 or only RIP Version 2 packets.	<code>version {1   2}</code>

The preceding task controls the default behavior of RIP. You can override that behavior by configuring a particular interface to behave differently. To control which RIP version an interface sends, perform one of the following tasks in interface configuration mode:

Task	Command
Configure an interface to send only RIP Version 1 packets.	<b>ip rip send version 1</b>
Configure an interface to send only RIP Version 2 packets.	<b>ip rip send version 2</b>
Configure an interface to send RIP Version 1 and Version 2 packets.	<b>ip rip send version 1 2</b>

Similarly, to control how packets received from an interface are processed, perform one of the following tasks in interface configuration mode:

Task	Command
Configure an interface to accept only RIP Version 1 packets.	<b>ip rip receive version 1</b>
Configure an interface to accept only RIP Version 2 packets.	<b>ip rip receive version 2</b>
Configure an interface to accept either RIP Version 1 or 2 packets.	<b>ip rip receive version 1 2</b>

## Enable RIP Authentication

RIP Version 1 does not support authentication. If you are sending and receiving RIP Version 2 packets, you can enable RIP authentication on an interface.

The key chain determines the set of keys that can be used on the interface. If a key chain is not configured, no authentication is performed on that interface, not even the default authentication. Therefore, you must also perform the tasks in the section “Manage Authentication Keys” in the “Configuring IP Routing Protocol-Independent Features” chapter.

We support two modes of authentication on an interface for which RIP authentication is enabled: plain text authentication and MD5 authentication. The default authentication in every RIP Version 2 packet is plain text authentication.

---

**Note** Do not use plain text authentication in RIP packets for security purposes, because the unencrypted authentication key is sent in every RIP Version 2 packet. Use plain text authentication when security is not an issue, for example, to ensure that misconfigured hosts do not participate in routing.

---

To configure RIP authentication, perform the following tasks in interface configuration mode:

Task	Command
<b>Step 1</b> Enable RIP authentication.	<b>ip rip authentication key-chain <i>name-of-chain</i></b>
<b>Step 2</b> Configure the interface to use MD5 digest authentication (or let it default to plain text authentication).	<b>ip rip authentication mode {text   md5}</b>

Task	Command
<b>Step 3</b> Perform the authentication key management tasks.	See the section “Manage Authentication Keys” in the “Configuring IP Routing Protocol-Independent Features” chapter.

See the “Key Management Examples” section of the “Configuring IP Routing Protocol-Independent Features” chapter for key management examples.

## Disable Route Summarization

RIP Version 2 supports automatic route summarization by default. The software summarizes subprefixes to the classful network boundary when crossing classful network boundaries.

If you have disconnected subnets, disable automatic route summarization to advertise the subnets. When route summarization is disabled, the software transmits subnet and host routing information across classful network boundaries. To disable automatic summarization, perform the following task in router configuration mode:

Task	Command
Disable automatic summarization.	<b>no auto-summary</b>

## Run IGRP and RIP Concurrently

It is possible to run IGRP and RIP concurrently. The IGRP information will override the RIP information by default because of IGRP’s administrative distance.

However, running IGRP and RIP concurrently does not work well when the network topology changes. Because IGRP and RIP have different update timers, and because they require different amounts of time to propagate routing updates, one part of the network will end up believing IGRP routes and another part will end up believing RIP routes. This will result in routing loops. Even though these loops do not exist for very long, the time to live (TTL) will quickly reach zero, and ICMP will send a “TTL exceeded” message. This message will cause most applications to stop attempting network connections.

## Disable the Validation of Source IP Addresses

By default, the software validates the source IP address of incoming RIP routing updates. If that source address is not valid, the software discards the routing update.

You might want to disable this feature if you have a router that is “off network” and you want to receive its updates. However, disabling this feature is not recommended under normal circumstances. To disable the default function that validates the source IP addresses of incoming routing updates, perform the following task in router configuration mode:

Task	Command
Disable the validation of the source IP address of incoming RIP routing updates.	<b>no validate-update-source</b>

## Enable or Disable Split Horizon

Normally, routers that are connected to broadcast-type IP networks and that use distance-vector routing protocols employ the *split horizon* mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routers, particularly when links are broken. However, with nonbroadcast networks (such as Frame Relay and SMDS), situations can arise for which this behavior is less than ideal. For these situations, you might want to disable split horizon. This applies to IGRP and RIP.

If an interface is configured with secondary IP addresses and split horizon is enabled, updates might not be sourced by every secondary address. One routing update is sourced per network number unless split horizon is disabled.

To enable or disable split horizon, perform the following tasks in interface configuration mode:

Task	Command
Enable split horizon.	<b>ip split-horizon</b>
Disable split horizon.	<b>no ip split-horizon</b>

Split horizon for Frame Relay and SMDS encapsulation is disabled by default. Split horizon is not disabled by default for interfaces using any of the X.25 encapsulations. For all other encapsulations, split horizon is enabled by default.

See the “Split Horizon Examples” section at the end of this chapter for examples of using split horizon.

---

**Note** In general, changing the state of the default is not recommended unless you are certain that your application requires making a change in order to advertise routes properly. Remember: If split horizon is disabled on a serial interface (and that interface is attached to a packet-switched network), you *must* disable split horizon for all routers in any relevant multicast groups on that network.

---

## Configure Interpacket Delay

By default, the software adds no delay between packets in a multiple-packet RIP update being sent. If you have a high-end router sending to a low-speed router, you might want to add such interpacket delay to RIP updates, in the range of 8 to 50 milliseconds. To do so, perform the following task in router configuration mode:

Task	Command
Add interpacket delay for RIP updates sent.	<b>output-delay <i>delay</i></b>

## RIP Configuration Examples

This section contains the following RIP configuration examples:

- Split Horizon Examples

## plit Horizon Examples

Two examples of configuring split horizon are provided.

### Example 1

The following sample configuration illustrates a simple example of disabling split horizon on a serial link. In this example, the serial link is connected to an X.25 network.

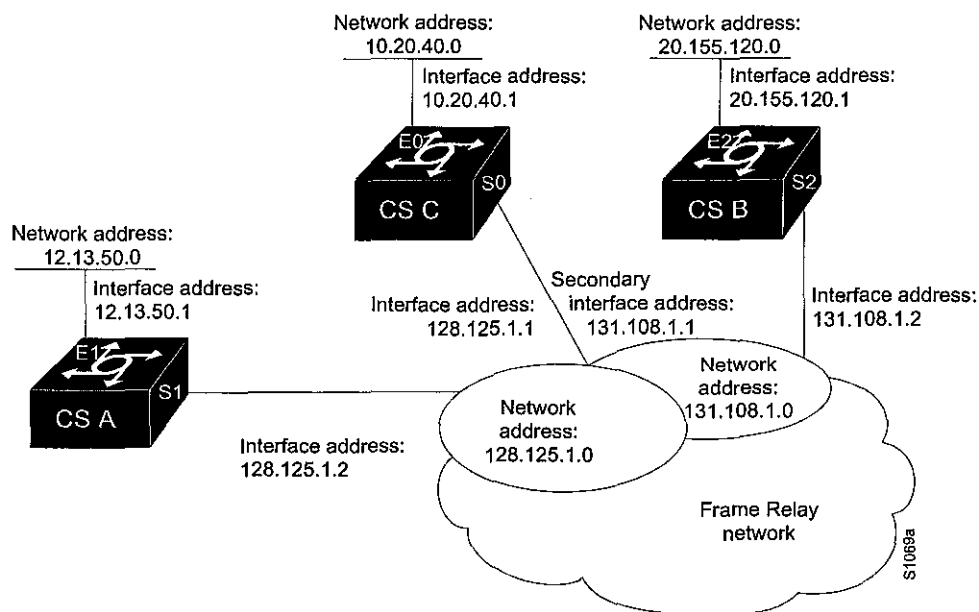
```
interface serial 0
encapsulation x25
no ip split-horizon
```

### Example 2

In the next example, Figure 16 illustrates a typical situation in which the **no ip split-horizon** interface configuration command would be useful. This figure depicts two IP subnets that are both accessible via a serial interface on Router C (connected to Frame Relay network). In this example, the serial interface on Router C accommodates one of the subnets via the assignment of a secondary IP address.

The Ethernet interfaces for Router A, Router B, and Router C (connected to IP networks 12.13.50.0, 10.20.40.0, and 20.155.120.0) all have split horizon *enabled* by default, while the serial interfaces connected to networks 128.125.1.0 and 131.108.1.0 all have split horizon *disabled* by default. The partial interface configuration specifications for each router that follow Figure 16 illustrate that the **ip split-horizon** command is *not* explicitly configured under normal conditions for any of the interfaces.

Figure 16 Disabled Split Horizon Example for Frame Relay Network



In this example, split horizon must be disabled in order for network 128.125.0.0 to be advertised into network 131.108.0.0, and vice versa. These subnets overlap at Router C, interface S0. If split horizon were enabled on serial interface S0, it would not advertise a route back into the Frame Relay network for either of these networks.

### Configuration for Router A

```
interface ethernet 1
  ip address 12.13.50.1
!
interface serial 1
  ip address 128.125.1.2
  encapsulation frame-relay
  no ip split-horizon
```

### Configuration for Router B

```
interface ethernet 2
  ip address 20.155.120.1
!
interface serial 2
  ip address 131.108.1.2
  encapsulation frame-relay
  no ip split-horizon
```

### Configuration for Router C

```
interface ethernet 0
  ip address 10.20.40.1
!
interface serial 0
  ip address 128.124.1.1
  ip address 131.108.1.1 secondary
  encapsulation frame-relay
```

## **APPENDIX D**

# **IMPLEMENTING QUALITY OF SERVICES POLICIES WITH DSCP VALUES**



# Implementing Quality of Service Policies with DSCP

Document ID: 10103

---

**Introduction**

**Prerequisites**

Requirements

Components Used

Background Theory

Conventions

**Differentiated Services Code Point**

**Assured Forwarding**

**Expedited Forwarding**

**Using the DSCP Field**

**Packet Classification**

**Marking**

**Using Committed Access Rate or Class-Based Policing**

**DSCP-Compliant WRED**

**Known Issues in Cisco IOS Software 12.2 Release Trains**

**Related Information**

---

## Introduction

This document describes how to set the Differentiated Services Code Point (DSCP) values in Quality of Service (QoS) configurations on a Cisco router, and it summarizes the relationship between DSCP and IP precedence.

## Prerequisites

### Requirements

You should be familiar with the fields in the IP header and Cisco IOS® CLI

### Components Used

This document is not restricted to specific software and hardware versions.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Background Theory

Differentiated Services (DiffServ) is a new model in which traffic is treated by intermediate systems with relative priorities based on the type of services (ToS) field. Defined in RFC 2474 and RFC 2475, the DiffServ standard supersedes the original specification for defining packet priority described in RFC 791. DiffServ increases the number of definable priority levels by reallocating bits of an IP packet for priority marking.

The DiffServ architecture defines the DiffServ (DS) field, which supersedes the ToS field in IPv4 to make per-hop behavior (PHB) decisions about packet classification and traffic conditioning functions, such as metering, marking, shaping, and policing.

The RFCs do not dictate the way to implement PHBs; this is the responsibility of the vendor. Cisco implements queuing techniques that can base their PHB on the IP precedence or DSCP value in the IP header of a packet. Based on DSCP or IP precedence, traffic can be put into a particular service class. Packets within a service class are treated the same way.

## Conventions

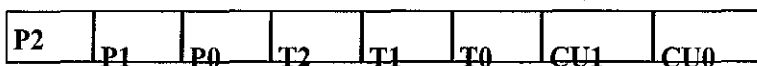
For more information on document conventions, refer to Cisco Technical Tips Conventions.

## Differentiated Services Code Point

The six most significant bits of the DiffServ field is called as the DSCP. The last two Currently Unused (CU) bits in the DiffServ field were not defined within the DiffServ field architecture; these are now used as Explicit Congestion Notification (ECN) bits. Routers at the edge of the network classify packets and mark them with either the IP Precedence or DSCP value in a Diffserv network. Other network devices in the core that support Diffserv use the DSCP value in the IP header to select a PHB behavior for the packet and provide the appropriate QoS treatment.

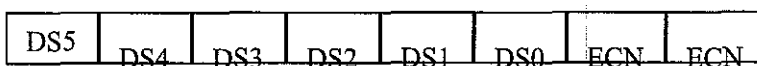
The diagrams in this section show a comparison between the ToS byte defined by RFC 791 and the DiffServ field.

### ToS Byte



- IP precedence three bits (P2 to P0)
- Delay, Throughput and Reliability three bits (T2 to T0)
- CU (Currently Unused) two bits(CU1–CU0)

### DiffServ Field



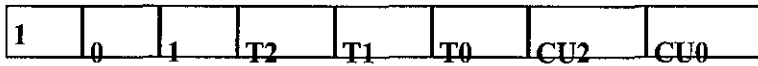
- DSCP six bits (DS5–DS0)
- ECN two bits

The standardized DiffServ field of the packet is marked with a value so that the packet receives a particular forwarding treatment or PHB, at each network node.

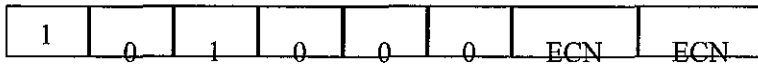
The default DSCP is 000 000. Class selector DSCPs are values that are backward compatible with IP precedence. When converting between IP precedence and DSCP, match the three most significant bits. In other words:

IP Prec 5 (101) maps to IP DSCP 101 000

### ToS Byte



**DiffServ Field**



The DiffServ standard utilizes the same precedence bits (the most significant bits DS5, DS4 and DS3) for priority setting, but further clarifies the definitions, offering finer granularity through the use of the next three bits in the DSCP. DiffServ reorganizes and renames the precedence levels (still defined by the three most significant bits of the DSCP) into these categories (the levels are explained in greater detail in this document):

Precedence Level	Description
7	Stays the same (link layer and routing protocol keep alive)
6	Stays the same (used for IP routing protocols)
5	Express Forwarding (EF)
4	Class 4
3	Class 3
2	Class 2
1	Class 1
0	Best effort

With this system, a device prioritizes traffic by class first. Then it differentiates and prioritizes same-class traffic, taking the drop probability into account.

The DiffServ standard does not specify a precise definition of "low," "medium," and "high" drop probability. Not all devices recognize the DiffServ (DS2 and DS1) settings; and even when these settings are recognized, they do not necessarily trigger the same PHB forwarding action at each network node. Each node implements its own response based on how it is configured.

## Assured Forwarding

RFC 2597 defines the assured forwarding (AF) PHB and describes it as a means for a provider DS domain to offer different levels of forwarding assurances for IP packets received from a customer DS domain. The Assured Forwarding PHB guarantees a certain amount of bandwidth to an AF class and allows access to extra bandwidth, if available. There are four AF classes, AF1x through AF4x. Within each class, there are three drop probabilities. Depending on a given network's policy, packets can be selected for a PHB based on required throughput, delay, jitter, loss or according to priority of access to network services.

Classes 1 to 4 are referred to as AF classes. The following table illustrates the DSCP coding for specifying the AF class with the probability. Bits DS5, DS4 and DS3 define the class; bits DS2 and DS1 specify the drop probability; bit DS0 is always zero.

Drop	Class 1	Class 2	Class 3	Class 4
Low	001010	010010	011010	100010

	AF11 DSCP 10	AF21 DSCP 18	AF31 DSCP 26	AF41 DSCP 34
Medium	001100 AF12 DSCP 12	010100 AF 22	011100 AF32	100100 AF42
High	001110 AF13 DSCP 14	DSCP 20 010110 AF23	DSCP 28 011110 AF33	DSCP 36 100110 AF43

DSCP 22    DSCP 30    DSCP 38

## Expedited Forwarding

RFC 2598 defines the Expedited Forwarding (EF) PHB: "The EF PHB can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DS (Diffserv) domains. Such a service appears to the endpoints like a point-to-point connection or a "virtual leased line." This service has also been described as Premium service." Codepoint 101110 is recommended for the EF PHB.

Again, vendor-specific mechanisms need to be configured to implement these PHBs. Refer to RFC 2598 for more information about EF PHB.

## Using the DSCP Field

There are three ways you can use the DSCP field:

- Classifier Select a packet based on the contents of some portions of the packet header and apply PHB based on service characteristic defined by the DSCP value.
- Marker Set the DSCP field based on the traffic profile.
- Metering Check compliance to traffic profile using either a shaper or dropper function.

Cisco IOS software considers the precedence bits of the ToS field if there is traffic that is queued in Weighted Fair Queuing (WFQ), Weighted Random Early Detection (WRED), or Weighted Round Robin (WRR). The precedence bits are not considered when Policy Routing, Priority Queuing (PQ), Custom Queuing (CQ), or Class Based Weighted Fair Queuing (CBWFQ) are configured.

## Packet Classification

Packet classification involves using a traffic descriptor to categorize a packet within a specific group and making the packet accessible for QoS handling in the network. Using packet classification, you can partition network traffic into multiple priority levels or a class of service (CoS).

You can use either access lists (ACLs) or the **match** command in the modular QoS CLI to match on DSCP values. For more information on how to use ACLs, refer to Quality of Service for the Cisco 7200/7500. Selecting a DSCP value in the match command was introduced in Cisco IOS Software Release 12.1(5)T.

```
Router1(config)# access-list 101 permit ip any any ?
```

Cisco – Implementing Quality of Service Policies with DSCP

dscp	Match packets with given dscp value
fragments	Check non-initial fragments
log	Log matches against this entry
log-input	Log matches against this entry, including input interface
precedence	Match packets with given precedence value
time-range	Specify a time-range
tos	Match packets with given TOS value

When you specify the *ip dscp* value in the **class map** command, you have these:

```
Router(config)# class-map match-all VOIP
1751-utl1(config-cmap)# match ip dscp ?
<0-63> Differentiated services codepoint value
af11 Match packets with AF11 dscp (001010)
af12 Match packets with AF12 dscp (001100)
af13 Match packets with AF13 dscp (001110)
af21 Match packets with AF21 dscp (010010)
af22 Match packets with AF22 dscp (010100)
af23 Match packets with AF23 dscp (010110)
af31 Match packets with AF31 dscp (011010)
af32 Match packets with AF32 dscp (011100)
af33 Match packets with AF33 dscp (011110)
af41 Match packets with AF41 dscp (100010)
af42 Match packets with AF42 dscp (100100)
af43 Match packets with AF43 dscp (100110)
cs1 Match packets with CS1 (precedence 1) dscp (001000)
cs2 Match packets with CS2 (precedence 2) dscp (010000)
cs3 Match packets with CS3 (precedence 3) dscp (011000)
cs4 Match packets with CS4 (precedence 4) dscp (100000)
cs5 Match packets with CS5 (precedence 5) dscp (101000)
cs6 Match packets with CS6 (precedence 6) dscp (110000)
cs7 Match packets with CS7 (precedence 7) dscp (111000)
default Match packets with default dscp (000000)
ef Match packets with EF dscp (101110)
Router1(config-cmap)# match ip dscp af31
```

## Marking

The DSCP can be set to a desired value at the edge of the network in order to make it easy for core devices to classify the packet as shown in the Packet Classification section and provide a suitable level of service. Class-Based Packet Marking can be used to set the DSCP value as shown here:

```
policy-map pack-multimedia-5M

!--- Creates a policy map named pack-multimedia-5M.

class management

!--- Specifies the policy to be created for the
!--- traffic classified by class management.

bandwidth 50
set ip dscp 8

!--- Sets the DSCP value of the packets matching
!--- class management to 8.
```

```

class C1
  priority 1248
  set ip dscp 40
class voice-signalling
  bandwidth 120
  set ip dscp 24

```

## Using Committed Access Rate or Class-Based Policing

Committed Access Rate and Class-Based Policing are traffic regulation mechanisms, used to regulate traffic flow to conform with the agreed upon service parameters. These mechanisms along with DSCP can be used to provide different levels of service to conforming and non-conforming traffic by appropriately modifying the DSCP value, as shown in this section.

Refer to [Configuring Traffic Policing and Comparing Class-Based Policing and Committed Access Rate](#) for more information.

```

interface Serial1/0.1 point-to-point

bandwidth 5000
ip address 192.168.126.134 255.255.255.252
rate-limit output access-group 150 8000 1500 2000 conform-action
  set-dscp-transmit 10 exceed-action set-dscp-transmit 20

!--- For traffic matching access list 150, sets the DSCP value of conforming traffic
!-- to 10 and that of non-conforming traffic to 20.

rate-limit output access-group 152 8000 1500 2000 conform-action
  set-dscp-transmit 15 exceed-action set-dscp-transmit 25
rate-limit output access-group 154 8000 1500 2000 conform-action
  set-dscp-transmit 18 exceed-action set-dscp-transmit 28
frame-relay interface-dlci 17
class shaper-multimedia-5M

```

## DSCP-Compliant WRED

Weighted Random Early Detection (WRED), selectively discards lower-priority traffic when the interface begins to get congested. WRED can provide differentiated performance characteristics for different CoS. This differentiated service can be on basis of the DSCP, as shown here:

```

class C2
  bandwidth 1750
  random-detect dscp-based

!--- Enable dscp-based WRED as drop policy.

  random-detect exponential-weighting-constant 7

!--- Specifies the exponential weight factor for the
!-- average queue size calculation for the queue.

  random-detect dscp 16 48 145 10

!--- Specifies the minimum and maximum queue thresholds
!-- for each DSCP value.

  random-detect dscp 32 145 435 10

```

Refer to the DiffServ Compliant WRED section of Congestion Avoidance Overview for more information.

## Known Issues in Cisco IOS Software 12.2 Release Trains

For more information on the following bugs, You can use the Bug Toolkit (registered customers only) for more information on these bugs:

- CSCdt63295 (registered customers only) If you fail to set the ToS byte with the new DSCP marking commands on the dial peers (set to 0) in Cisco IOS Software Release 12.2.2T, then packets will not be marked and they will remain with a ToS set to 0.
- CSCdt74738 (registered customers only) Support for the **set ip dscp** command on the Cisco 7200 router and lower-end platforms for the for multicast packets should be available as of Cisco IOS Software Release 12.2(3.6) and later.

---

## Related Information

- [Implementing DiffServ for End-to-End Quality of Service Overview](#)
- [Using Content Networking to Provide Quality of Service](#)
- [Cisco IOS Software: Quality-of-Service: The Differentiated Services Model \(DiffServ\)](#)
- [Control Plane DSCP Support for RSVP](#)
- [Diff-Serv-aware Traffic Engineering \(DS-TE\)](#)
- [Differentiated Services Compliant Distributed Weighted Random Early Detection](#)
- [RFC 3168: The Addition of Explicit Congestion Notification \(ECN\) to IP](#)
- [Quality of Service \(QoS\) Support Pages](#)
- [Technical Support – Cisco Systems](#)

---

All contents are Copyright © 1992–2006 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

---

Updated: May 24, 2006

Document ID: 10103

---

## **APPENDIX E**

# **QUALITY-OF-SERVICE THE DIFFERENTIATED SERVICES MODEL**





A SHEET

# QUALITY-OF-SERVICE THE DIFFERENTIATED SERVICES MODEL

Updated: August 2005

## CHALLENGE

Organizations delivering network-based services need powerful end-to-end solutions to effectively and predictably deliver the differing Quality-of-Service (QoS) requirements of voice, video, and data applications. Voice, for example, requires a small but assured amount of bandwidth, low delay, jitter and low packet loss. A data application such as File Transfer Protocol (FTP) needs more bandwidth, but can tolerate the delay and jitter.

## SOLUTION

Cisco IOS® Software DiffServ offers application-level QoS and traffic management in an architecture that incorporates mechanisms to control bandwidth, delay, jitter and packet loss. Cisco Diffserv complements the Cisco IntServ offering by providing a more scalable architecture for end-to-end QoS. This scalability is achieved by the mechanisms controlling QoS at an aggregate level. Application traffic can be categorized into multiple classes (aggregates), with QoS parameters defined for each class. A typical arrangement would be to categorize traffic into premium, gold, silver, bronze, and best-effort classes.

## STANDARDS-BASED

Cisco IOS Software DiffServ is fully compliant with the Internet Engineering Task Force (IETF) standards defined in RFC 2474, RFC 2475, RFC 2797 and RFC 2598. This solution leverages the new IETF definition of the IPv4 Type of Service (ToS) octet in the IP packet-header by utilizing the Differentiated Services Code Point (DSCP) field to classify packets into any of the 64 possible classes. When the packets are classified, IETF-defined Per-Hop Behaviors (PHBs) including Assured Forwarding (AF) and Expedited Forwarding (EF) are implemented using Cisco QoS tool sets. Traffic characterized as EF will receive the lowest latency, jitter and assured bandwidth services which is suitable for applications such as Voice over VoIP). AF allows carving out the bandwidth between multiple classes in a network according to desired policies. As a value-add, this implementation also allows a construction of user-defined PHBs, beyond the scope of AF and EF. DSCP code points other than the ones reserved for EF and best effort service can be associated with an arbitrary PHB.

## CISCO IOS SOFTWARE: QUALITY-OF-SERVICE APPLICATIONS

In an enterprise environment, QoS policies must allow critical business applications to receive requisite resources, while ensuring other applications are not neglected. By classifying the application traffic into premium, gold, silver and other classes, a baseline methodology is set to provide end-to-end QoS. Diffserv enables this classification by utilizing the DSCP field. Using Cisco DiffServ, a properly designed network can deliver assured bandwidth, low latency, low jitter and packet loss for voice while simultaneously ensuring slices of available bandwidth to other classes.

Service Providers want to provide value-added services to their customers by providing blanket Service Level Agreements (SLAs), as well as application-specific assurances (aSLAs). It is possible, for example, to divide a customer's traffic at the network edge into gold, silver, and bronze services (also referred to as olympic service) and provide relative / absolute assurances to each. Cisco also provides for mapping the per-class IP QoS requirements into Asynchronous Transfer Mode (ATM) Classification on Flows (CoS) parameters, providing mechanisms both at the edge and the core. Within the Service Provider network, Cisco enables end-to-end QoS, via Multi Protocol Label Switching (MPLS)-Diffserv. MPLS could also be used as a reference mechanism to translate the IP QoS to MPLS QoS.

## Architectural Components

### Packet Classification

Packets entering a DiffServ domain or region (collection of DiffServ routers) can be classified in a variety of ways—from IP source and destination addresses, Layer 4 protocol and port numbers, incoming interface, MAC address, IP precedence, the DSCP value, Layer 2 information (such as IEEE 802.1p bits, Ethernet 802.1p bits), and the Cisco value-added mechanism Network Based Application Recognition (NBAR). Once these packets are classified on the basis of the criteria mentioned above, they can be processed, conditioned and marked. Packet classification and other mechanisms can all be performed within the Cisco Modular QoS CLI (MQC), a modular technique to separate packet classification from the policy applied to the classes, from the application of the policy on an interface or sub-interface.

### Packet Marking

IPv4 Type of Service (ToS) octet has been redefined from the 3-bit IP-precedence to a 6-bit DSCP field (Figure 1). Packets can be marked with arbitrary DSCP value / standard values, corresponding to the appropriate AF (Figure 2), EF or user defined class. For example, EF is designated the codepoint "101110". Cisco IOS Software also supports class-selector codepoints, which is a method of marking the 6 DSCP bits that are compatible with systems that only support the IP-precedence scheme. These codepoints are in the form of "xyz000", where x, y, and z can represent a 0. The codepoint for best-effort traffic will be set to "000000". This implementation brings additional value-add by also allowing packets to be marked with an arbitrary DSCP, and mapping them to a locally significant (non-AF/EF/default) PHB. This allows for construction of new and previously unavailable services.

Figure 1. DiffServ Codepoint Field

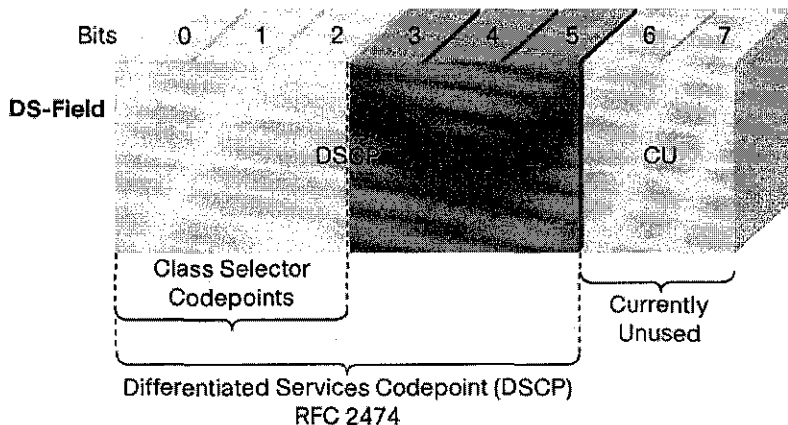
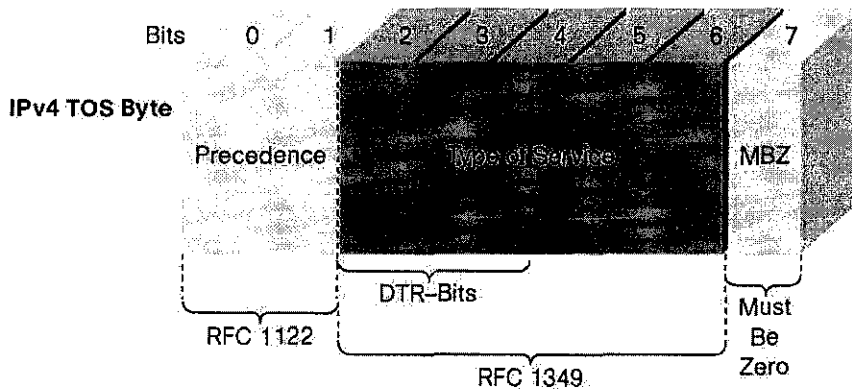


Figure 2. The Original IPv4 ToS Byte



**Bits (0–2): IP-Precedence Defined**

- 111 – Network Control
- 110 – Internetwork Control
- 101 – CRITIC/ECP
- 100 – Flash Override
- 011 – Flash
- 101 – Immediate
- 001 – Priority
- 000 – Routine

**Bits (3–6): The Type of Service Defined**

- 0000 – [all normal]
- 1000 – [minimize delay]
- 0100 – [maximize throughput]
- 0010 – [maximize reliability]
- 0001 – [minimize monetary cost]

Table 1. DiffServ AF Codepoint Table

OP Precedence	Class #1	Class #2	Class #3	Class #4
Low Drop Precedence	(AF11) 001010	(AF21) 010010	(AF31) 011010	(AF41) 100010
Medium Drop Precedence	(AF12) 001100	(AF22) 010100	(AF32) 011100	(AF42) 100100
High Drop Precedence	(AF13) 001110	(AF23) 010110	(AF33) 011110	(AF43) 100110

**DiffServ Traffic Conditioning**

At the edge of the network, this component is logically responsible for classifying, marking, metering, and shaping or policing the packets entering the network. In the Cisco IOS DiffServ model, classification and marking are done using the MQC. Metering is done using a token bucket algorithm, shaping is done using class-based traffic shaping or class-based frame relay traffic shaping and policing is done using class-based policing. On the other hand, Cisco also provides for the per-class accounting Management Information Base (MIB) and statistics for each class (regardless of DiffServ domain) can be gathered for management purposes.

**DiffServ PHB Enforcing**

As a packet leaves the Ingress Router, and enters the network core, PHBs are enforced, depending on the packet marking, with the appropriate DiffServ PHB. EF can be implemented using Low Latency Queuing (LLQ). AFxy PHBs can be implemented using Class Based Weighted Fair Queuing (CBWFQ) and Weighted Random Early Detect (WRED), Class Based Policing, or Class Based Traffic Shaping (CBTS). Locally defined PHBs can also be constructed using the same tools—LLQ, CBWFQ, and WRED.

## Table 2. Key Cisco IOS Diffserv Features and Benefits

Features	Benefits
Full IETF Compliancy	Standards based QoS that can be applied end-to-end
Packet Classifications via DSCP	Scalability—Fewer states are stored at the core of the network
Standard and User Defined PHBs	End-to-end construction of well-defined services for applications
Modular QoS CLI	Granular traffic control and flexible management
CEF and Arbitrary Classes	Flexible classification and service offerings

### SOFTWARE SUPPORT

Available in Cisco IOS Software Release 12.1(5)T and later versions

### ADDITIONAL INFORMATION

Additional information about Cisco IOS QoS technology can be found at <http://www.cisco.com> or by contacting your local Cisco representative.



**Corporate Headquarters**  
Cisco Systems, Inc.  
20 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
[www.cisco.com](http://www.cisco.com)  
Tel: 408 526-4000  
1 800 553-NETS (6387)  
Fax: 408 526-4100

**European Headquarters**  
Cisco Systems International BV  
Haarlerbergpark  
Haarlerbergweg 13-19  
1101 CH Amsterdam  
The Netherlands  
[www-europe.cisco.com](http://www-europe.cisco.com)  
Tel: 31 0 20 357 1000  
Fax: 31 0 20 357 1100

**Americas Headquarters**  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
[www.cisco.com](http://www.cisco.com)  
Tel: 408 526-7660  
Fax: 408 527-0883

**Asia Pacific Headquarters**  
Cisco Systems, Inc.  
168 Robinson Road  
#28-01 Capital Tower  
Singapore 068912  
[www.cisco.com](http://www.cisco.com)  
Tel: +65 6317 7777  
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on [the Cisco Website at www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Copyright © 2005 Cisco Systems, Inc. All rights reserved. CCSP, CCVP, the Cisco Square Bridge logo, Follow Me Browsing, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Access Registrar, Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CIE, CCIP, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Partnerise, the iQ logo, iQ Net Readiness Scorecard, LightStream, Linksys, MeetingPlace, MGX, the Networkers logo, Networking Academy, Network Registrar, Packet, PIX, Post-Routing, Pre-Routing, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StrataView Plus, TeleRouter, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0502R) 205274.W\_ETMG\_PL\_8.05

Printed in the USA



## **APPENDIX F**

# **WHICH ROUTING PROTOCOL SHOULD MY NETWORK USE?**

# Which Routing Protocol Should My Network Use?

Date: Sep 30, 2005 By Russ White, Alvaro Retana, Don Slice. Sample Chapter is provided courtesy of Cisco Press.

One of the toughest questions facing network engineers is which routing protocol to use. Each has its strengths, and each works better with some network designs than with others. This chapter will help you decide which routing protocol is best for your particular network.

Among all the thorny questions that network engineers are asked on a regular basis, probably among the hardest is this one:

- My network currently runs Enhanced Interior Gateway Routing Protocol (EIGRP). Would I be better off if I switched to Open Shortest Path First (OSPF)?

You can replace the two protocols mentioned in this sentence with any pair of protocols among the advanced interior gateway protocols (OSPF, Intermediate System-to-Intermediate System [IS-IS] and EIGRP), and you have described a question that routing protocol engineers are asked probably thousands of times a year. Of course, convergence is always faster on the other side of the autonomous system boundary, so to speak, so it is always tempting to jump to another protocol as soon as a problem crops up with the one you are running.

How do you answer this question in real life? You could try the standard, "It depends," but does this really answer the question? The tactic in the Routing Protocols Escalation Team was to ask them questions until they went away, but none of these answers really helps the network operator or designer really answer the question, "How do you decide which protocol is the best?"

Three questions are embedded within this question, really, and it is easier to think about them independently:

- Is one protocol, in absolute terms, "better" than all the other protocols, in all situations?
- If the answer to this first question is "No," does each routing protocol exhibit some set of characteristics that indicate it would fit some situations (specifically, network topologies) better than others?
- After you have laid out the basics, what is the tradeoff in living with what you currently have versus switching to another routing protocol? What factors do you

need to consider when doing the cost/benefit analysis involved in switching from one routing protocol to another?

This appendix takes you through each of these three questions. This might be the first and last time that you hear a network engineer actually answer the question, "Which routing protocol should I use?" so get ready for a whirlwind tour through the world of routing.

## Is One Protocol "Better" Than the Others?

The first thing you need to do with this sort of question is to qualify it: "What do you mean by better?" Some protocols are easier to configure and manage, others are easier to troubleshoot, some are more flexible, and so on. Which one are you going to look at?

This appendix examines ease of troubleshooting and convergence time. You could choose any number of other measures, including these:

- **Ease of management**—What do the Management Information Bases (MIBs) of the protocol cover? What sorts of **show** commands are available for taking a network baseline?
- **Ease of configuration**—How many commands will the average configuration require in your network configuration? Is it possible to configure several routers in your network with the same configuration?
- **On-the-wire efficiency**—How much bandwidth does the routing protocol take up while in steady state, and how much could it take up, at most, when converging in response to a major network event?

### Ease of Troubleshooting

The average uptime (or reliability) of a network is affected by two elements:

- How often does the network fail?
- How long does it take to recover from a failure?

The network design and your choice of equipment (not just the vendor and operating system, but also putting the right piece of equipment into each role and making certain that each device has enough memory, and so on) play heavily into the first element. The design of the network also plays into the second element. The piece often forgotten about when considering the reliability of a network is how long it takes to find and fix, or troubleshoot, the network when it fails.

Ease of management plays a role in the ease of troubleshooting, of course; if it is hard to take a baseline of what the network is supposed to look like, you will not do so on a regular basis, and you will have a dated picture to troubleshoot from. The tools available for troubleshooting are also important. Of course, this is going to vary between the implementations of the protocols; here, implementations in Cisco IOS Software illustrate



the concepts. Table G-1 outlines some of the troubleshooting tools that are available in EIGRP, OSPF, and IS-IS, in Cisco IOS Software.

**Table G-1 Cisco IOS Software Troubleshooting Tools for EIGRP, OSPF, and IS-IS**

	EIGRP	OSPF	IS-IS
Debug Neighbors	Neighbor formation state; hello packets.	Neighbor formation state; hello packets.	Packets exchanged during neighbor formation.
Log Neighbor State	Yes.	Yes.	No.
Debug Database Exchange	Packets exchanged (updates, replies, and so on), with filters per neighbor or for a specific route.	Packets flooded, with filters for specific routing information. Packets retransmitted.	Packets flooded.
Debug Interactions with the Routing Table	Yes.	No.	No.
Debug Route Selection Process	Yes (DUAL <sup>1</sup> FSM <sup>2</sup> events).	Yes (SPF <sup>3</sup> events).	Yes (SPF events).
Show Database	Yes, by specific route and route state.	Yes, by LSA <sup>4</sup> type and advertising router.	Yes, by LSP <sup>5</sup> ID or type of route.
Event Log	Yes; understandable if you comprehend DUAL and its associated terminology.	Yes; only understandable if you have access to the source code.	No.

<sup>1</sup> DUAL = Diffusing Update Algorithm

<sup>2</sup> FSM = finite state machine

<sup>3</sup> SPF = shortest path first

<sup>4</sup> LSA = link-state advertisement

<sup>5</sup> LSP = link-state packet

From this chart, you can see that EIGRP generally provides the most tools for finding a problem in the network quickly, with OSPF running a close second.

## Which Protocol Converges Faster?

I was once challenged with the statement, "There is no way that a distance vector protocol can ever converge faster than a link-state protocol!" An hour and a half later, I think the conversation tapered off into, "Well, in some situations, I *suppose* a distance vector protocol *could* converge as fast as a link-state protocol," said without a lot of conviction.

In fact, just about every network engineer can point to reasons why he thinks a specific routing protocol will *always* converge faster than some other protocol, but the reality is that all routing protocols can converge quickly or slowly, depending on a lot of factors strictly related to network design, without even considering the hardware, types of links, and other random factors that play into convergence speed in different ways with each protocol. As a specific example, look at the small network illustrated in Figure G-1 and consider the various options and factors that might play into convergence speed in this network.



Figure G-1 Simple Network

This figure purposefully has no labels showing anything concerning routing protocols configuration or design; instead, this section covers several possible routing configurations and examines how the same protocol could converge more or less quickly even on a network this small through just minor configuration changes.

Start with EIGRP as an example:

- The Router A to C link has a bandwidth of 64 kbps.
- The Router A to B link has a bandwidth of 10 Mbps.
- The Router B to D and Router C to D links have equal bandwidths.

With this information in hand, you can determine that Router D is going to mark the path to 10.1.1.0/24 through Router B as the best path (the *successor* in EIGRP terms). The path through Router C will not be marked as a *feasible successor*, because the differential in the metrics is too great between the two paths. To the EIGRP process running on Router D, the path through Router C cannot be proven based on the metrics advertised by Routers B and C, so the path through Router C will not be installed as a possible backup route.

This means that if the Router B to D link fails, Router D is forced to mark 10.1.1.0/24 as *active* and send a query to Router C. The convergence time is bounded by the amount of time it takes for the following tasks:

- Router D to examine its local topology table and determine that no other known loop-free paths exist.
- Router D to build and transmit a query toward Router C.
- Router C to receive and process the query, including examining its local EIGRP topology table, and find it still has an alternate path.
- Router C to build a reply to the query and transmit it.
- Router D to receive the reply and process it, including route installation time and the time required to change the information in the forwarding tables on the router.

Many factors are contained in these steps; any one of them could take a long time. In the real world, the total time to complete the steps in this network is less than two or three seconds.

Now change the assumptions just slightly and see what the impact is:

- The Router A to C link and A to B links have equal bandwidth.
- The Router B to D link has a bandwidth of 64 kbps.
- The Router B to C link has a bandwidth of 10 Mbps.

As you can tell, the network conditions have been changed only slightly, but the results are altered dramatically. In this case, the path to 10.1.1.0/24 through Router C is chosen as the best path. EIGRP then examines the path through Router B and finds that it is a loop-free path, based on the information embedded in EIGRP metrics. What happens if the Router B to C link fails?

The process has exactly one step: Router D examines its local EIGRP topology table and finds that an alternate loop-free path is available. Router D installs this alternate route in the local routing table and alters the forwarding information as needed. This processing takes on the order of 150 milliseconds or less.

Using the same network, examine the various reactions of OSPF to link failures. Begin with these:

- The Router B to D link has a cost of 20.
- All other links in the network have a cost of 10.
- All routes are internal OSPF routes.

What happens if the Router B to C link fails?

1. Router B and C detect the link failure and wait some period of time, called the link-state advertisement (LSA) generation time. Then they flood modified router LSAs with this information.
2. The remaining routers in the network receive this new LSA and place it in their local link-state databases. The routers wait some period of time, called the shortest path first (SPF) wait time, and then run SPF.

3. In the process of running SPF, or after SPF has finished running (depending on the implementation), OSPF will install new routing information in the routing table.

With the default timers, it could take up to one second (or longer, in some situations) to detect the link failure and then about three and a half seconds to flood the new information. Finally, it could take up to two and a half seconds before the receiving routers will run SPF and install the new routing information. With faster times and various sorts of tuning, you can decrease these numbers to about one second or even in the 300-millisecond range in some specific deployments.

Making Router D an area border router (ABR) dramatically impacts the convergence time from the Router E perspective because Router D has to perform all the preceding steps to start convergence. After Router D has calculated the new correct routing information, it must generate and flood a new summary LSA to Router E, and Router E has to recalculate SPF and install new routes.

Redistributing 10.1.1.0/24 into the network and making the area that contains Routers A, B, C, and D into a not-so-stubby area (NSSA) throws another set of timers into the problem. Router D now has to translate the Type 7 external LSA into an external Type 5 LSA before it can flood the new routing information to Router E.

These conditions do not even include the impact of multiple routes on the convergence process. EIGRP, for instance, can switch from its best path to a known loop-free path for 10,000 routes just about as fast as it can switch 1 route under similar conditions. OSPF performance is adversely impacted by the addition of 10,000 routes into the network, possibly doubling convergence time.

You can see, then, that it is not so simple to say, "EIGRP will always converge faster than OSPF," "IS-IS will always converge faster than EIGRP," or any other combination you can find. Some people say that OSPF always converges faster than EIGRP, for instance, but they are generally considering only intrarea convergence and not the impact of interarea operations, the impact of various timers, the complexity of the SPF tree, and other factors. Some people say that EIGRP always converges faster than any link-state protocol, but that depends on the number of routers involved in the convergence event. The shorter the query path, the faster the network converges.

If you align all the protocol convergence times based on the preceding examination, you generally find the convergence times in this order, from shortest to longest:

1. EIGRP with feasible successors.
2. Intrarea OSPF or IS-IS with fast or tuned timers.
3.
  - a. EIGRP without feasible successors.
  - b. Intrarea OSPF or IS-IS with standard timers.
  - c. Interarea OSPF or IS-IS.

The last three are highly variable, in reality. In any particular network, OSPF, IS-IS, and EIGRP without feasible successors might swap positions on the list. The network design, configuration, and a multitude of other factors impact the convergence time more than the routing protocol does. You get the best convergence time out of a routing protocol if you play the network design to the strengths of the protocol.

## Which Designs Play to the Strength of Each Protocol?

The natural question, after you have decided that network design plays into the suitability of the protocol (you have seen this to be the case for convergence speed, but the same is also true of any other factor you might consider for a given routing protocol, including management, troubleshooting, configuration, and so on) is this:

- What sorts of network designs play into the strengths of any given routing protocol?

This is not an easy question to answer because of the numerous ways to design a network that works. Two- and three-layer network designs, switched cores versus routed cores, switched user access versus routed user access—the design possibilities appear to be endless. To try to put a rope around this problem, the sections that follow examine only a few common topological elements to illustrate how to analyze a specific topology and design and try to determine how a routing protocol will react when running on it.

The specific types of network topologies considered here are as follows:

- Hub-and-spoke designs
- Full mesh designs
- Highly redundant designs

After you consider each of these specific topology elements, you learn the general concepts of hierarchical network design and how each protocol plays against them.

### Hub-and-Spoke Topologies

Hub-and-spoke network designs tend to be simple in theory and much harder in implementation. Scaling tends to be the big problem for hub-and-spoke topologies. The primary focus here is the capability of a routing protocol to maintain a multitude of routing neighbors and to converge to massive network events in an acceptable amount of time. Assume, throughout this section, that you are always dealing with dual-homed hub-and-spoke networks, as Figure G-2 illustrates.



Figure G-2 Dual-Homed Hub-and-Spoke Network

Start by considering the following simple question:

- How many spokes or remote routers does it take to really start stressing any routing protocol that is running over a hub-and-spoke network design?

The answer to this question always depends on various factors, including link speed and stability, router processing speed and packet switching speeds, and other factors. However, general experience shows that a high-speed router (in terms of processing power) with reasonably good design supports at least 100 remote sites with any modern routing protocol.

When considering network designs in which hundreds of remote sites are available, however, you need to use special techniques with each protocol to scale the number of remote sites attached to a single pair of hub routers. Look at each protocol to see what types of problems you might encounter and what types of tools are available to resolve those problems:

- OSPF floods topology information to each router within an area and summaries of reachability information into the area. You can place all the remote site routers into one or more OSPF *stub areas*, which cuts down on the amount of information flooded out to each remote site. Any change on a remote site is still flooded to every other remote site within the same area. For that reason, the design becomes a tradeoff between the number of areas that you want to manage and that the hub routers support and the amount of information that you can flood through the low-speed links connecting the remote stub sites.
- IS-IS also floods information to each router within an area. It does not, by default, flood information from the core of the network (the L2 routing domain) into each area. Again, you still face the tradeoff of how many level 1 routing domains you want to support at the hub routers versus how much information you can flood toward each remote router.
- The primary factor in determining scaling and convergence time in an EIGRP hub-and-spoke network is the number of queries the hub router needs to generate or process when the network changes, and the number of updates the hub router needs to generate toward the remote. Normally, if a hub loses several routes, for instance, it needs to generate queries for each of those routes to each of the remote sites. The remote sites then query the other hub router, which must process and reply to each of the queries. If the number of routes is high, this can be a processor- and memory-intensive task, causing the network to converge slowly, especially if the links between the remote sites and the hub routers are low speed. In this situation, you can summarize routers at the core toward the remote routers and block the routing information transmitted up toward the core routers. You can also cut down on the query range into the hub-and-spoke network dramatically. EIGRP, however, also provides a special operational mode for the remote sites; you can configure the remote sites as *stubs*, which indicates to the hub routers that the remote sites are never used for transiting traffic. If the remote sites are

configured as stub routers, the hub router never queries them for lost routes, and the scaling properties change dramatically.

EIGRP, in theory, scales much better in a hub-and-spoke topology—and this is true in real networks, too. You often find EIGRP hub-and-spoke networks that have more than 500 remote sites attached to a pair of hub routers, over low bandwidth links, in the wild. In contrast, you tend to see OSPF and IS-IS hub-and-spoke networks top out at around 200 remote sites, even if higher bandwidth links are involved.

## Full Mesh Topologies

Full mesh topologies are a less common design element in networks, but they are worth considering because the scaling properties of a routing protocol in a full mesh design indicate, to some degree, the scaling properties of the same protocol in a partial mesh design. You can think of a full mesh topology as a special case of a partial mesh topology. Again, look at the challenges and tools that are available for each protocol. Use the network illustrated in Figure G-3 throughout this discussion.



Figure G-3 Full Mesh Network

- Each OSPF router sends topology information to each adjacent neighbor within an area (flooding domain). If Router A receives a new link-state advertisement (LSA), Router D receives three copies of this new LSA: one from Router A, one from Router B, and one from Router C. The Cisco IOS Software implementation of OSPF does have an option to control the flooding through a full mesh network, using the **database filter-out** command.
- IS-IS is similar to OSPF; each router sends topology information to each adjacent neighbor. Cisco IOS Software enables you to control flooding through *mesh groups*.
- Each router in an EIGRP network sends each of the routes it is using to forward traffic to each neighbor. In this network, Router D is going to receive three copies of any new routing information that Router A receives, one copy from Router A, one from Router B, and one from Router C. These three copies of the routing information might be the same, but they indicate reachability through three different next hops (or neighbors). Reducing the information propagated through the mesh is difficult, at best. You can filter these routing updates through some paths within the mesh to decrease the amount of information flooded through the mesh, but that also reduces the number of paths usable through the mesh for any specific destination.

OSPF and IS-IS flood extra information through a mesh topology by default, but you can use tools to reduce the amount of flooding in highly meshed topologies. EIGRP sends updates through each router in the mesh, but it is difficult to reduce the number of these updates unless you want to decrease the number of paths that the network actually uses through the mesh.

In the real world, OSPF and IS-IS scale better in highly meshed environments, especially if you implement flooding reduction techniques. This is a matter of scale, of course; networks that have a mesh network of 20 or 30 routers work fine with any of the three routing protocols. However, when the mesh starts surpassing this number of routers, the special techniques that OSPF and IS-IS offer to scale further can make a difference.

## Interaction with Hierarchical Designs

Traditional network design is based on layers, either two or three, that abstract the network details into "black boxes" and divide functionality vertically through the network to make management and design easier:

- The two-layer model has *aggregation* and *core layers*, or *areas*, within the network.
- The three-layer model has *access*, *distribution*, and *core layers*.

How do these layered network designs interact with each protocol? Consider each protocol in turn.

OSPF splits flooding domains into areas that are separated by ABRs. Because every router within an area must share the same link-state database to calculate loop-free paths through the network, the only place that route aggregation can be performed is at an ABR. ABRs actually aggregate two types of information:

- Information about the topology of an area that is hidden from other areas at these border edges
- Aggregation of reachability information that can be configured at these border edges

This combination of route aggregation points and flooding domain boundaries in the network implies several things:

- In all three-layer network designs with OSPF, you should place the ABR in the distribution layer of the network.
- In all two-layer network designs with OSPF, you should place the ABR at the aggregation to core layer edge of the network.
- The most aggregation points that you can cross when passing from one edge of the network to the opposite edge of the network is two.



These topological limitations might not be major in smaller networks, but in networks that have thousands of routers, they could impose severe restrictions on the network design. Network designers and operators normally break up OSPF networks at this size into multiple administrative domains, connecting the separate domains through BGP or some other mechanism.

IS-IS is similar to OSPF in its restrictions, except that IS-IS allows the core and outlying flooding domains to overlap. This introduces a degree of flexibility that OSPF does not provide, but you can still only aggregate routing information at the edges where two flooding domain meet, and you cannot build more than two levels of routing into the network.

EIGRP, as a distance vector protocol, does not divide the concepts of topology summarization and routing aggregation; topology beyond one hop away is hidden by the natural operation of the protocol. Figure G-4 illustrates the conceptual difference among EIGRP, OSPF/IS-IS, and RIP in terms of topology information propagated through the network.



Figure G-4 Topological Awareness in Routing Protocols

If you examine the scope through which routing information is transmitted (or known) within a network, you find the following:

- The Bellman-Ford algorithm, used by the Routing Information Protocol (RIP) and the Interior Gateway Routing Protocol (IGRP), uses only information about the local cost to reach a given destination. If Router B is running RIP, it considers only the total cost of the path to reach a destination at Router E when deciding on the best (loop-free) path.
- Diffusing Update Algorithm (DUAL), used by EIGRP, considers the local cost to reach a given destination and the cost of each neighbor to reach the same destination when calculating which available paths are loop free. EIGRP uses an awareness of the topology that is one hop away from the calculating router.
- OSPF and IS-IS, which are link-state protocols, do not use information about the metrics of a neighbor; rather, they count on being aware of the entire topology when calculating a loop-free path. At a flooding domain border, OSPF and IS-IS act much like distance vector protocols. Router A does not know about the topology behind Router B; it only knows the cost of Router B to reach destinations that are attached to Router E.

Because topology information is hidden in the natural processing of EIGRP routing updates, EIGRP is not restricted in where it can aggregate routing information within the network. This provides a great deal of flexibility to network designers who are running EIGRP. Multiple layers of aggregation can be configured in the network. This means that

moving from one edge of the network to the opposite edge of the network could mean encountering many more than two aggregation points.

The practical result of the EIGRP capability to aggregate routing information anywhere in the network is that many existing large-scale (2000 router and larger) networks run within a single EIGRP process or administrative domain. The feasibility of building networks this large is based on the capability to use route aggregation to divide the network into multiple layers, or sections, each acting fairly independently of the other. Although it is possible to build an OSPF or IS-IS network this large, designing and managing this network is more difficult because of the restrictions that link-state protocols place on aggregation points.

In general, up to some relative size, the protocols are relatively equal in their capability to work with hierarchical network designs. OSPF and IS-IS tend to be less flexible about where route aggregation can be placed in the network, making it more difficult, in some situations, to fit the network design and the protocol design together. EIGRP excels at fitting into hierarchical network design.

## **Topological Rules of Thumb**

After examining these various network topologies and how each routing protocol tends to react, you can see that when a network does not reach the edge of a specific protocol capability on any given topology, any of the routing protocols is fine. If your network has a specific predominant topology type, however, such as large-scale hub-and-spoke or large-scale full mesh topologies, choosing a protocol to fit those topologies makes sense. You can always compromise in complex areas of your network design by making effective and stable topological design areas in which the routing protocol is really stretched to the edge of its capabilities.

## **What Are the Tradeoffs?**

In many networks, the final decision of which routing protocol is "best" comes down to these issues:

- **Convergence speed**—How important is convergence speed? How much flexibility do you have in the design of your network around convergence speeds?
- **Predominant topologies**—Does your network design have one dominant type of topology? Would a full mesh or large-scale hub-and-spoke topology benefit from running one protocol over another?
- **Scaling strategy**—Does your scaling strategy call for dividing the network into multiple pieces, or does it call for a single IGP domain, with the network broken up into pieces through route aggregation and other techniques?
- **Maintenance and management**—Which routing protocol fits the network management style of your day-to-day operations? Which one seems easier to troubleshoot and manage in your environment?

Beyond the technical factors are some nontechnical ones. For instance, if you decide to switch protocols, what is the cost for the long term? You need to consider training costs, the cost of revised procedures, design effort, and possible downtime while you convert the network from one protocol to another.

In some situations, this might not be an issue. For instance, if two networks are being merged because of a corporate merger, and each runs a different protocol, the decision might be more open to consideration. If you are going to need to convert one half of the network or the other, you can more carefully consider the technical considerations and make a decision based on those considerations alone. However, if your network is stable today, you should think twice about switching protocols unless a change in the business environment or some major shift in the way the network is built indicates it is an important move to make to meet the needs of the enterprise.

---

© 2006 Pearson Education, Inc. Informit. All rights reserved.

800 East 96th Street Indianapolis, Indiana 46240