# Design and Implementation of a Management System for Differentiated Services Using the SNMP Framework

Jae-Young Kim, Myung-Sup Kim, Won-Ki Hong
Department of Computer Science and Engineering
Pohang University of Science and Technology
{jay, mount, jwkhong}@postech.ac.kr

Tae-Sang Choi, Yoon-Hee Jung, and Sung-Won Sohn
Internet Architecture Team
Intenet Technology Department
Electronics and Telecommunications Research Institute
{choits, yhjung, swsohn}@etri.re.kr

**Abstract**

Differentiated Services (DiffServ), which is being standardized by IETF, is considered to be a promising method for supporting different service characteristics to different classes of network users on the Internet. DiffServ is useful when supporting distributed applications that deal with mutlimedia or real-time network traffic flows. The IETF DiffServ working group has defined a general architecture of DiffServ and is elaborating more detailed features. Network vendors have started developing DiffServ-enabled network devices. However, the management aspect of DiffServ is not fully standardized yet. A simple but powerful management mechanism is needed in order to operate, provision, monitor and control differentiated services on the Internet. In this paper, we present our work on the design of an architecture for managing DiffServ using the SNMP framework. We have analyzed IETF DiffServ MIB and designed a DiffServ management system. In order to realize and validate our design, DiffServ routers have been developed in Linux systems and the DiffServ agents have been implemented for these DiffServ routers. Based on our design, we have also developed a Web-based management system that can configure, monitor and control DiffServ and that provides a user-friendly and ubiquitous access to the resources being managed.

**Keywords**

Differentiated Services, SNMP Agent, Internet Management, Web-based Management

# I. INTRODUCTION

Over the past decade, the number of devices and the number of Internet users have increased at an exponential rate. As the Internet boundary continues to widen, the network traffic caused by data transfers over the Internet will continue to rise. While previous bandwidths were sufficient to carry text-based application data, curent bandwidths are no longer sufficient to handle current multimedia and real-time network traffic flows.

As the increased rate of network bandwidth is quite slower than the increased rate of network usage, there are always bottleneck points where bandwidth is insufficient for network users. In such situations, every packet competes for access to the bandwidth and the result is packet loss, unexpected delays, and jitter. However, both Transmission Control Protocol (TCP) and Internet Protocol (IP), which are two network protocols for delivering packets in the Internet, were originally designed in the best-effort service model. They handle all packets with the same priority: first come, first serve.

But users' requirements are changing. They want to get different service qualities for different types of services they obtain. Integrated Services (IS) with Resource reServation Protocol (RSVP) signaling is the first approach to provide such a service on the Internet [22, 26]. RSVP attempts to provide per-flow QoS support assurances with dynamic resource reservation. A flow is defined by the 5-tuple, consisting of source and destination IP address, transport protocol, and source and destination port. However, since RSVP/IS relies on per-flow states and per-flow processing in every network node, it is difficult to deploy RSVP/IS in large carrier networks like the Internet.

Differentiated Services (DiffServ) is an alternative approach to provide differentiated service qualities to different classes of users. DiffServ uses aggregation of traffics in each routing decision point. Type of Service (ToS) field is used for distinguishing these traffic aggregates. Since the ToS is much simpler than the 5-tuple information, it is easier to implement DiffServ than RSVP/IS [3, 9, 28].

DiffServ applies administrative domain concepts. Within one domain, core routers forward traffics according to the ToS field of the traffic aggregates. Between two different domains, there are edge routers which perform classification of flows based on 5-tuple information like RSVP/IS. Since the edge routers mark the ToS field of the incoming traffic, core routers do not need to handle complex information in such traffic. The example of a DiffServ domain is illustrated in Figure 1.

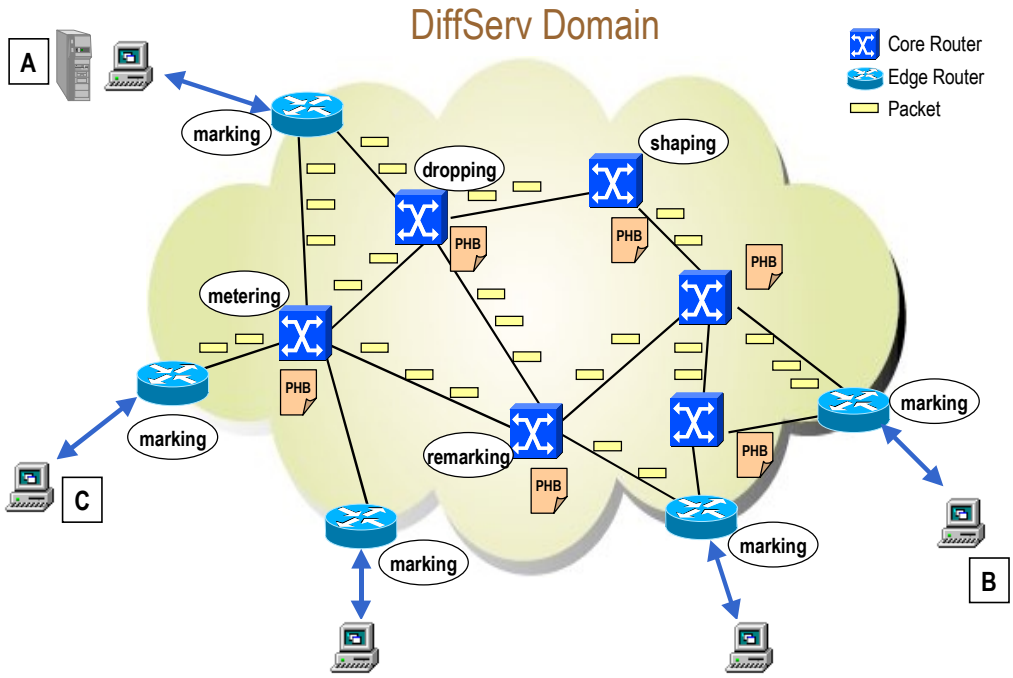



Figure 1 : Example of a DiffServ Domain

At the boundary of the DiffServ domain there are edge routers marking ToS fields of incoming packets. Within the domain, there are core routers which perform various differentiating actions such as dropping, metering, shaping, and remarking. If host A is a Video-on-Demand (VOD) server and host B wants to connect to the host A with a relatively high priority, the host B requests differentiated service for its connection to the host A and the packets between host A and B have high-priority ToS field marking when entering the DiffServ domain. The core routers can distinguish the packets from general packets such as file transfers between the host B and the host C. The core routers treat packets differently according to the packets' ToS field and consequently provides different service qualities to different network connections. The basic architecture of DiffServ is further explained in Section 2.

Although the IETF DiffServ working group has defined several standards in DiffServ, management of DiffServ is not yet fully standardized. Current standards have defined only the operational aspects of DiffServ. When deploying DiffServ in network nodes, various management functions are needed for remote control of a large number of DiffServ nodes. From the management point of view, network element management, network management,

and service management need to be defined. Some considerations are how to configure each DiffServ router, how to change its configuration, and how to monitor or meter traffics each router handles.

In this paper, we have constructed an SNMP management framework for managing DiffServ. The IETF DiffServ working group has defined DiffServ MIB for managing DiffServ-enabled network devices. Based on this MIB, we have developed an SNMP agent system that operates in Linux-based DiffServ routers. Furthermore, we have developed a Web-based DiffServ manager that provides easy-to-use interfaces which can operate in any Web browser.

The rest of this paper is organized as follows: Section 2 explains the architecture of differentiated services proposed by IETF. Section 3 presents the design of a DiffServ management system. Section 4 describes the detailed implementation on Linux systems and Section 5 summarizes our work and discusses possible future research.

## II. ARCHITECTURE OF DIFFSERV

The IETF DiffServ working group has defined the basic architecture of DiffServ. In this section, we summarize current standards and investigate several issues.

### 2.1. GENERAL CONCEPTS OF DIFFSERV

Internet traffic is defined as a collection of equal priority packet streams treated in best-effort forwarding schemes at network nodes, and it has explosively increased in recent years.

DiffServ proposes a basic method to differentiate a set of traffics among network nodes. The method is based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different behavior aggregates. Each behavior is identified by a single Differentiated Services Code Point (DSCP).

DSCP is the most-significant 6 bits from the IPv4 Type-Of-Service (ToS) octet or IPv6 traffic class octet. This 6-bit field indicates how each router should treat the packet. This treatment is called a Per-Hop Behavior (PHB). PHB defines how an individual router will treat an individual packet when sending it over the next hop through the network. Being 6 bits long, the DSCP can have one of 64 different binary values.

Four overall types of PHBs have been defined as standard thus far [9, 10, 13, 14]. They are default, class-selector, Assured Forwarding (AF), and Expedited Forwarding (EF). Table 1 summarizes the standard PHBs and DSCP values accordingly.

Table 1 : Standard PHBs

| PHB Name | DSCP | Description |
|---|---|---|
| Default | 000000 | Best-effort (RFC 1821) |
| class-selector | xxx000 | 7 classes (RFC 2474) |
| Afxy | xxxyy0 | 4 classes with 3 drop probabilities (RFC 2597) |
| EF | 101110 | no drop (RFC 2598) |

DiffServ-enabled network nodes handle classes of network packets differently by using the DSCP values in the packet header within an administrative domain. The DS domain is a contiguous set of DiffServ-enabled nodes which operate with a common service provisioning policy and a set of PHB groups implemented in each node. There are two kinds of DiffServ nodes in a DS domain. To select a PHB from one of the PHB groups supported within the domain, edge routers, located at the boundary of the DS domain, classify (and possibly condition) ingress traffic to ensure that packets crossing the domain are appropriately marked. Internal routers, called core routers, only check the DSCP values of forwarded packets and perform predefined PHB actions on the packets. It requires no per-flow state in backbone and trunk routers. This internal behavior saves times and resources for providing different service qualities to different classes of users.

### 2.2. COMPONENTS

A DiffServ-enabled network node needs to have several components for handling DiffServ. Figure 2 explains five components of DiffServ architecture; classifier, meter, marker, shaper, and dropper [11, 12] in a traffic conditioning block (TCB).

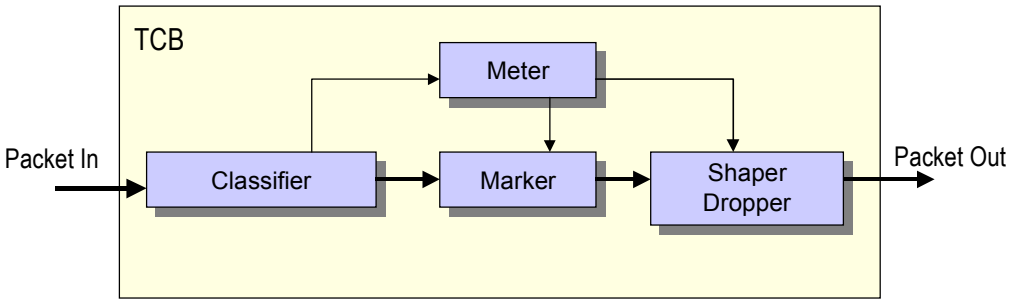



Figure 2 : Basic Traffic Control Block of DiffServ

A classifier selects network packets in a traffic stream based on the content of some portion of the packet header. There are two types of classifiers, the Behavior Aggregate (BA) classifier based on the DiffServ values, and the Multi-Field (MF) classifier based on the

value of a combination of 5-tuple information.

A meter measures the temporal properties of the stream of packets selected by a classifier. It passes the state information to other conditioning actions to trigger a particular action for each packet.

A marker sets the DSCP of a packet and a shaper delays some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. A dropper discards some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile.

DiffServ router is a fundamental DiffServ-enabled network node. The conceptual model and requirements of the DiffServ routers are discussed in IETF [16, 20]. The DiffServ router is considered to have routing component, set of TCBs, queuing component, and configuration and monitoring interfaces that are organized as in

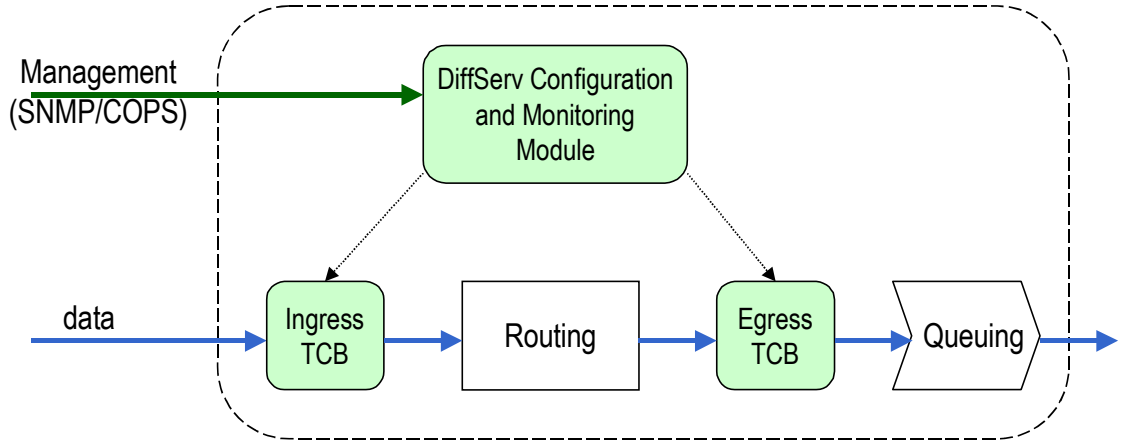



Figure 3 : Conceptual Model of a DiffServ Router

DiffServ-related components are separated from the routing component to make it easy to add DiffServ capability to the existing router. There are set of TCBs cascaded both in ingress point and egress point. Traffic conditioning can be performed either at the ingress point or at the egress point, or both. Queuing component is a set of underlying packet queues keeps packets before the routers sends them out. Management module for DiffServ router can be operated in several ways such as SNMP or COPS. The management module configure TCB parameters and monitor the peformance of each TCB.

### 2.3. CURRENT OPEN ISSUES

Much work remains to be done in DiffServ architecture [3, 28]. The following issues are currently being researched.

- Network management

  The current DiffServ architecture concerns only how to operate with the DSCP and PHB in each network node. There are not enough so-called FCAPS management issues in the current standards. Recently the IETF DiffServ working group proposed an SNMP MIB definitions for DiffServ network nodes.

- End-to-end flow management

  DiffServ network nodes handles traffic without any interaction with neighbor nodes. Although this simplifies functions of a network node, no interaction among cooperating network nodes makes it difficult to understand end-to-end flow behavior. In order to control the end-to-end traffic characteristics in one domain, a kind of overall control function is needed.

- Integration with the policy framework

  Common Open Policy Service (COPS) protocol in the IETF policy framework has been proposed for flexible and configurable control of network nodes [23, 24]. Integrating the policy framework in Integrated Services and DiffServ is a major research topic of late.

- Interdomain QoS negotiation

  A set of DSCPs and PHBs is meaningful and consistent in only one DS domain. They are not valid or totally different in another DS domain. Thus edge routers located in domain borders need to perform a kind of QoS negotiations with the neighbor edge routers to keep the service quality across multiple DS domains. Bandwidth Broker (BB) concept is introduced and BBs negotiate with each other to determine the QoS parameter among different DS domains or Integrated Services. These operations are not yet completely specified.

In this paper we focus on adding network management architecture and functionality to the current DiffServ architecture. We have designed and implemented a DiffServ management system in the SNMP management framework.

# III. Design of a DiffServ Management System

In this section, we present a detailed design processes of a DiffServ management system which is based on the SNMP framework. First, we define and categorize the management of DiffServ. Next, the requirements of a DiffServ management system are listed. For our DiffServ management system we choose the SNMP framework. The SNMP framework concept is overviewed and the IETF DiffServ MIB is analyzed from the operational viewpoint. Finally, the design architecture of our DiffServ management system is described.

## 3.1. Management of Differentiated Services

Current DiffServ architecture does not have fully standardized management functionality. As we explained in Section 2, only operational and functional models are now standardized. Currently a number of management concepts are introduced and discussed by the IETF DiffServ working group, yet, most management issues are related to network element management. From a sophisticated management viewpoint such as Telecommunication Management Network (TMN) [32, 33, 34], a complete vertical management framework is required.for managing DiffServ. By following TMN logical layered architecture, we have itemized several DiffServ management issues in each layer as in Figure 4.

| | |
|---|---|
| Business Management | Metering / Accounting  Interdomain Traffic Management Policy<br>Traffic Shaping Policy  DiffServ Marking Policy |
| Service Management | SLA Negotiation / Monitoring / Enforcement  Session Management<br>Customer Service Management  Interdomain Service |
| Network Management | Network Topology Management  Link / Path Performance Metering<br>Link / Path Monitoring  End-to-end Traffic Flow Management |
| Element Management | PHB Configuration Provisioning  TCB Performance Metering<br>Fault Monitoring  DiffServ Policing  Traffic Shaping |

Figure 4 : Management Layers for Differentiated Services

The element management layer manages individual network nodes in a DiffServ domain.

Configuration of DiffServ routers, performance and fault monitoring, traffic shaping, and traffic policing are examples of element management operations. Basically, management operations on this layer are executed on each DiffServ-enabled network node. Current IETF discussions cover management issues on this layer partially.

The network management layer manages DiffServ networks as a whole. Network topology management, link and path performance metering or monitoring, and end-to-end flow management are examples of network management operations. DiffServ network is logically constructed by combining all the DiffServ routers in a DiffServ domain. Handling DiffServ network as a whole is very useful for end-to-end traffic management. We can oversee and control end-to-end DiffServ flows with the constructed network information.

The service management layer manages high-level DiffServ services. Service Level Agreement (SLA) management, session management, customer service management, and interdomain service management are examples of service management operations. A set of DiffServ domains are interconnected to link service providers and service consumers. Managing end-to-end service quality is a key role in this layer.

The business management layer manages business-oriented considerations. Metering or accounting and business policies for various operations are examples of business management operations. DiffServ will open a new market to provide different levels of service qualities at different prices. Business management of DiffServ will be important in the near future.

Operations in the upper layer are composed of operations in the lower layer. For providing complete DiffServ management operations covering every logical layer, management operations in each layer are needed to be created and stabilized. In this paper, our research efforts focus on element management and network management of DiffServ. Service and business management issuses are subjects for future research work.

### 3.2. REQUIREMENTS OF A DIFFSERV MANAGEMENT SYSTEM

For developing a DiffServ management system, we have analyzed requirements of a DiffServ management system. The followings are selected requirements.

- Management in the SNMP framework

  SNMP is considered as a standard management framework for Internet management. Although COPS and policy management framework has been proposed for the flexible management recently, in order to simplify the system, we have chosen the SNMP

framework for our management system. The IETF DiffServ MIB is the key management information between the SNMP manager and agent.

- Web-based management interfaces

We have adopted Web-based management interfaces for our management system. Web-based management has quite a few advantages [29]. Since Web browsers are ubiquitous, easy-to-use, and platform-independent, the management interfaces running in Web browsers provide cost-effective, portable, and convenient interfaces to human managers. The same Web browser both receives managemnet operations and displays operation results.

- Remote configuration provisioning

When DiffServ is deployed in an organization, there might be lots of DiffServ-enabled network nodes working together in the organization's backbone network. A DiffServ management system should be able to configure numbers of DiffServ nodes remotely to provision network resources.

- Real-time metering and status monitoring

Network traffic changes its characteristics dynamically. In order to make DiffServ nodes to be cope with the dynamic traffic variations, a DiffServ management system needs to have realtime metering and status monitoring capability. Periodic SNMP polling and history keeping should be performed in a DiffServ manager. Further, graphical representation of status changes should be displayed. Based on the metering and monitoring results, a DiffServ management system can be able to modify configurations of a set of DiffServ nodes.

- End-to-end flow management

A DiffServ flow is defined as a traffic flow with the same DSCP value in a DiffServ domain. There is a set of different DiffServ flows coming in and out in a DS domain. However, as descibed in the previous section, DiffServ nodes do not interact with each other to keep end-to-end statistics of each DiffServ flow. From the management point of view, the end-to-end characteristics of DiffServ flows are important because the network service quality is determined by the end-to-end performance parameters, not by the individual performance paramters in each DiffServ node. Therefore, a DiffServ

management system should be able to summarize performance parameters of each DiffServ node and report the end-to-end DiffServ flow characteristics.

### 3.3. SNMP FRAMEWORK

Simple Network Management Protocol (SNMP) framework is chosen for the DiffServ management platform. SNMP is simple, cost-effective, and the *de facto* standard for managing Internet-related systems [2].

The SNMP framework consists of Management Information Base (MIB) and SNMP operations. MIB is the specification containing definitions of management information so that networked systems can be remotely monitored, configured, and controlled [1]. There are three typical SNMP operations which manipulate MIB variables between an SNMP manager and an SNMP agent. A GET operation reads a value of a MIB variable from an SNMP agent, a SET operation sets a value of a MIB variable in an SNMP agent, and a TRAP operation reports asynchronous events to an SNMP manager from an SNMP agent.

Defining a MIB is the first step for developing an SNMP management system. The following subsection describes the DiffServ MIB.

### 3.4. DIFFSERV MIB

The IETF DiffServ working group currently suggests a Management Information Base (MIB) for the DiffServ architecture [5]. The MIB is designed according to the DiffServ implementation conceptual model [20]. Table 2 summarizes six object tables defined in the DiffServ MIB.

Table 2 : DiffServ MIB Structure

| Table Name | # of Objects | Description |
|---|---|---|
| Aggregate | 1 | DSCP value |
| MFClassifier | 13 | multi-field classification value |
| Classifier | 8 | classifier list |
| TBMeter | 6 | token-bucket meter value |
| Action | 15 | mark / count / drop |
| Queue | 7 | priority queue |

Each table contains several MIB objects to configure, monitor, and modify DiffServ characteristics in a network node. By getting and setting these object values via SNMP, SNMP managers are able to manage DiffServ-enabled network nodes remotely.

The DiffServ table entries are linked each other with the RowPointer textual convention.

RowPointer object is used for pointing an entry in the same or different table [2]. A traffic control block (TCB), as illustrated in Figure 2, consists of a number of different classifiers, meters, actions, and queues. The DiffServ MIB represents a TCB as a series of table entries linked together by RowPointers. With this scheme a lot of different TCBs can be represented in the six tables efficiently. For example, several TCBs have the same BA and MF classifiers, the same actions or meters on the same queues. Thus the same parameters need not be defined separately. The current MIB design provides parameter sharing for different TCBs. If the DiffServ MIB defines a TCB table which contains entries for all the TCBs, the size of the table would be much more than the sum of the current six table entries. Figure 5 shows the linked relationship of six tables in the DiffServ MIB.

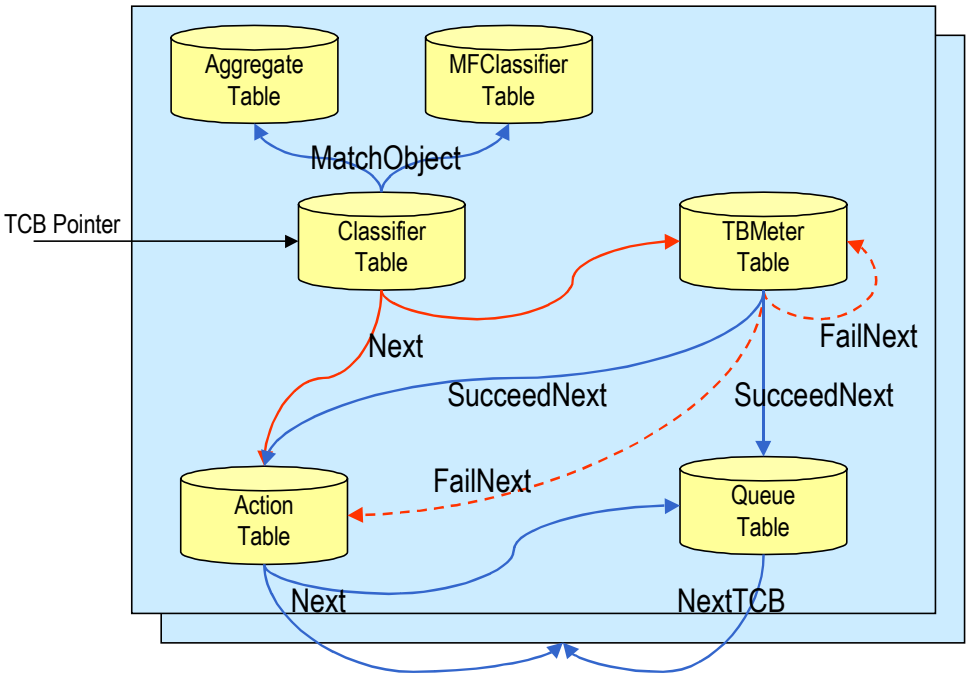



Figure 5 : MIB Table Relationship

All the TCB pointers in a DiffServ network node are contained in the classifier table. The TCB configuration linked list starts from the table entry. Each entry in the classifier table has two RowPointer objects, MatchObject and Next. The MatchObject points either to an Aggregate table entry or an MFClassifier table entry and Next points to either a TBMeter table entry or an Action table entry. The TBMeter table entry is used for determining the conformance of packet flows and the next TCB component is chosen on the basis of whether the current packet flow succeeds or fails the meter parameters. When it succeeds, the packet flow is sent to the appropriate Action or Queue table entry, and when it fails, the packet flow

is sent to another Meter table entry or the appropriate Action table entry. The Action and Queue table entry has a pointer to the next TCB classifier and this closes a TCB configuration.

### 3.5. DESIGN ARCHITECTURE

We have designed a DiffServ management system satisfying the previous requirements. The architecture consists of three distinct layers as depicted in Figure 6. The three-tier architecture includes a network management system (NMS) client running in a Web browser, an NMS server containing a Web server and DiffServ manager, and a network element performing DiffServ routing and SNMP management.

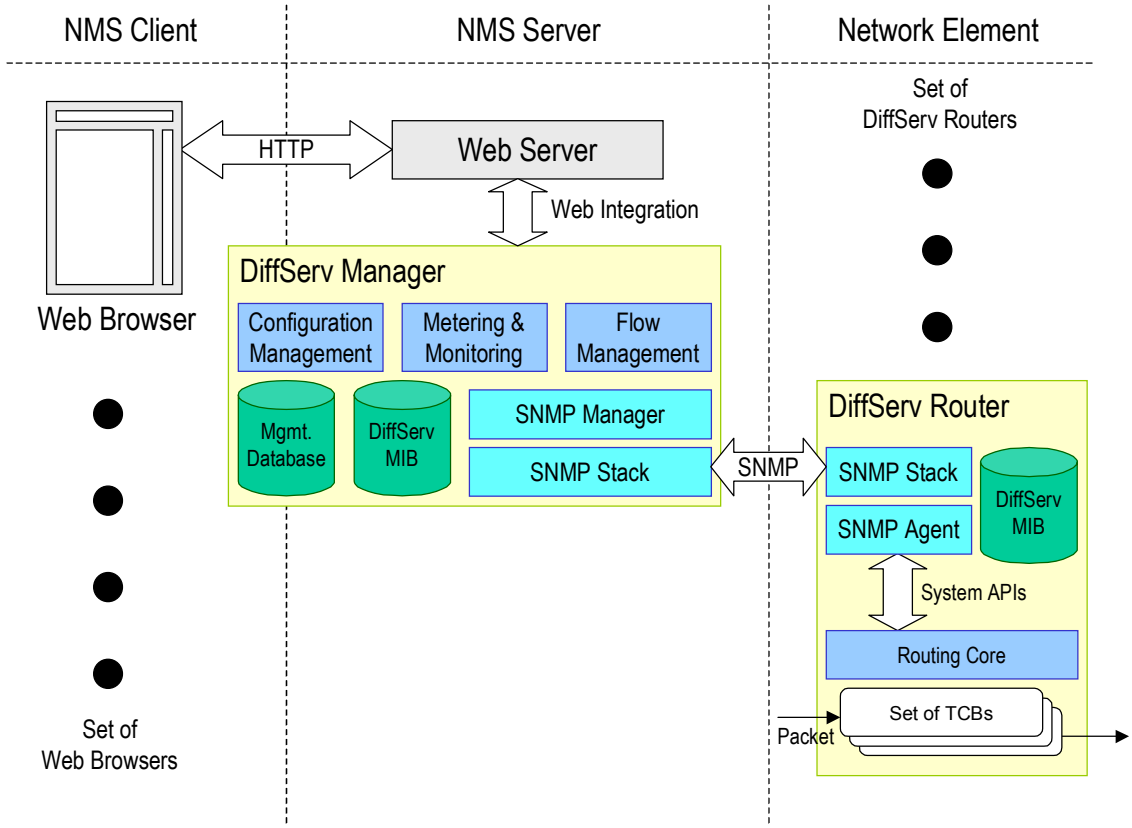



Figure 6 : Design Architecture of the DiffServ Management System

The NMS server is the central server for managing a set of DiffServ routers and providing management interfaces to a set of Web browsers. The Web server located in the NMS server layer has a role to provide a Web-based management interfaces in Web browsers. The integration with Web server and DiffServ manager can be achieved in various ways such as a basic HTML file access method, a Common Gateway Interface (CGI) method, and a Java

applet/servlet method.

The DiffServ manager performs three high-level DiffServ management functions – configuration management, metering and monitoring, and flow management. The management database for storing and retrieving additional information other than the DiffServ MIB is needed in order to provide high-level DiffServ management functions. At the bottom of the DiffServ manager, an SNMP communicates with a set of SNMP agents running in different DiffServ routers within a DS domain.

Three high-level DiffServ management functions performs sophisticated and extended management functions for providing value-added management operations to users. Configuration management function performes remote configuration provisioning. Every DiffServ paramteter is determined and enforced via the configuration management function. Metering and monitoring function periodically observes the status of DiffServ routers and compares the results with predefined desirable peformance metrics. Such conformance test results are necessary for modifying DiffServ router behaviors in the configuration management function. The flow management function summarizes all the DiffServ flows in a DS domain and provides the end-to-end DiffServ flow characteristics. The function collects routing tables and DiffServ flow information from every DiffServ router and constructs an overall end-to-end parameters of each DiffServ flow. The parameters are useful for understanding current quality of network services.

The DiffServ routers are managed network elements in the design architecture. A DiffServ router contains a routing core module for controlling a set of TCBs that execute packet forwarding according to various DSCP values and an SNMP agent module for handling SNMP manager requests for the DiffServ MIB. System-dependent APIs are used for connecting the SNMP agent module and the routing core module. The values of DiffServ MIB variables are determined by specific system-dependent system calls. The methods of retrieving and setting DiffServ parameters in the routing core module need not be the same among different implementation architectures.

Within a DS domain there might be lots of both DiffServ routers and DiffServ management clients interworking with each other. The three-tier architecture has advantages in such network management environments. One centralized DiffServ manager controls a set of DiffServ routers while providing management interfaces to a set of management clients at the same time. However, by separating the management user interfaces from the manager itself, the DiffServ manager is able to concentrate on the management functions and thus the performance of the DiffServ manager can be improved. To address the scalability problem,

the three-tier architecture can be easily extended to support distributed management functionality with multiple DiffServ managers located in the middle layer. Obviously, there must be a concrete way of combining the multiple DiffServ managers in this case.

## IV. IMPLEMENTATION

Based on the previous design architecture, we have implemented a DiffServ management system. DiffServ routers have been constructed in Linux systems and we have added an SNMP agent for the IETF DiffServ MIB in each router. A centrallized DiffServ manager that includes an SNMP manager for the DiffServ MIB has been implemented in a dedicated Linux system as well. A Web-based DiffServ management interfaces has been also developed for delivering various management services to users very conveniently.

A DiffServ agent is running on a Linux DiffServ router. The agent has DiffServ MIB values that are extracted from the Linux traffic controller via NetLink sockets. A DiffServ manager is integrated with Web technologies. The manager interacts with several agents via SNMP and interacts with human managers via HTTP. Human managers can operate every DiffServ routers in their control within easy-to-use Web browsers. Implementation details of the manager and the agent are described in the following subsections.

### 4.1. DIFFSERV IN LINUX SYSTEMS

Linux, a shareware operating system, supports QoS features in its networking kernel from the kernel version 2.1.90 [30]. The QoS support offers a wide variety of traffic control functions, which can be combined in a modular way. The traffic control code in the Linux kernel consists of four conceptual components (queuing discipline, class, filter, and policing) and configured logically as in Figure 7 [7, 8].

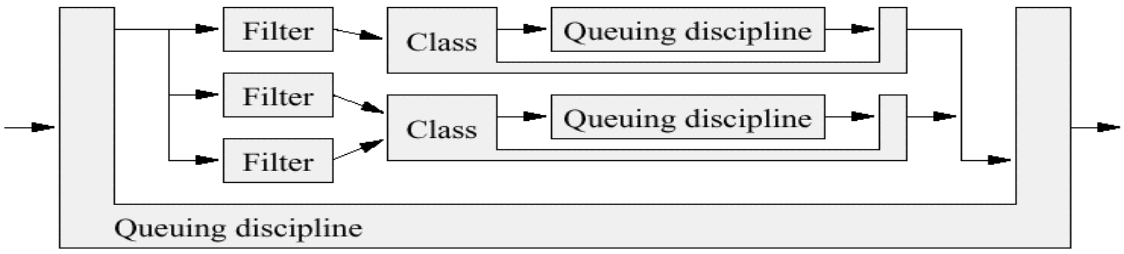



Figure 7 : Linux Traffic Control Blocks

Each network device has a queuing discipline associated with it, which controls how packets enqueued on that device are treated. Queuing disciplines may use filters to

distinguish among different classes of packets and process each class in a specific way. Policing resides within a filter and performs the conformance test on classes of packets and appropriate actions on each class depending on the test results.

Based on this Linux traffic control framework, W. Almesberger et al. have designed and implemented basic DiffServ-field classification and manipulation functions required by DiffServ network nodes [8]. The extended DiffServ features are freely available in the form of a kernel patch. The latest DiffServ package (DS-6) available from 'DiffServ on Linux' Web site [28] contains the kernel patch, a control program (tc), and several PHB scripts written with the control program. By installing the DiffServ package, a Linux system is able to perform DiffServ router functions.

However, the current Linux DiffServ implementation does not have sufficient management functionality. There is no management architecture and every script setup needs to be manually configured and modified in local machines. Further, metering and monitoring functions of DiffServ are not fully supported. Our work focuses on this lack of management functionality.

### 4.2. LINUX DIFFSERV AGENT

The DiffServ agent is an SNMP agent with DiffServ MIB running on the Linux DiffServ router. Basically the agent extracts DiffServ parameters from the Linux traffic control kernel and modifies the appropriate MIB values on the request from a DiffServ manager. The agent also receives management operations from a DiffServ manager and performs the appropriate parameter changes in the Linux traffic control kernel.

The organization of the Linux DiffServ router implementation is explained in Figure 8. There are two process spaces in the Linux operating system, the user space and the kernel space. Extending from Linux traffic control framework, the Linux DiffServ implementation resides in the kernel space. In the user space, the DiffServ SNMP agent is implemented in the user space, combined with the Linux traffic control program. Communication between the DiffServ agent and the Linux traffic control kernel is effected via NetLink sockets [32]. The NetLink socket is a socket-type bidirectional communication link located between kernel space and user space. It transfers information between them.

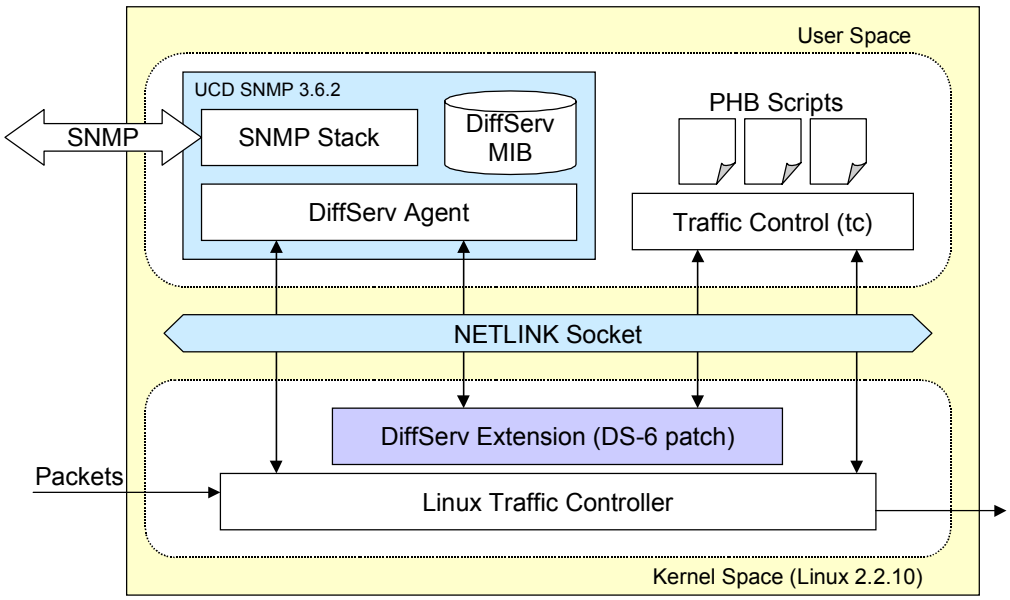



Figure 8 : Organization of Linux DiffServ Router Implementation

The agent has been implemented by using UCD SNMP agent extension package [6]. UCD SNMP 3.6.2 provided the agent development environment. The DiffServ agent uses the traffic control program (tc) or NETLINK socket directly for accessing DiffServ parameters in kernel space and manipulates the values of DiffServ MIB.

Mapping of Linux traffic control parameters between DiffServ MIB objects is a key role of DiffServ agent. There are three kinds of traffic control parameters in Linux kernel and six MIB tables in DiffServ MIB. Figure 9 shows how two different values are mapped in our DiffServ agent.

DSCP values in the aggregate table are extracted from filter parameters. Values of the MFClassifer table are obtained from match-ip parameters in u_32 filters. The classifer table is constructed by qdisc and filter parameters. The TBMeter table objects are mapped to sentBytes and sentPackets in qdisc and filter paramters. The action table objects are from class parameters. And finally, the queue table is filled with qdisc parameters.

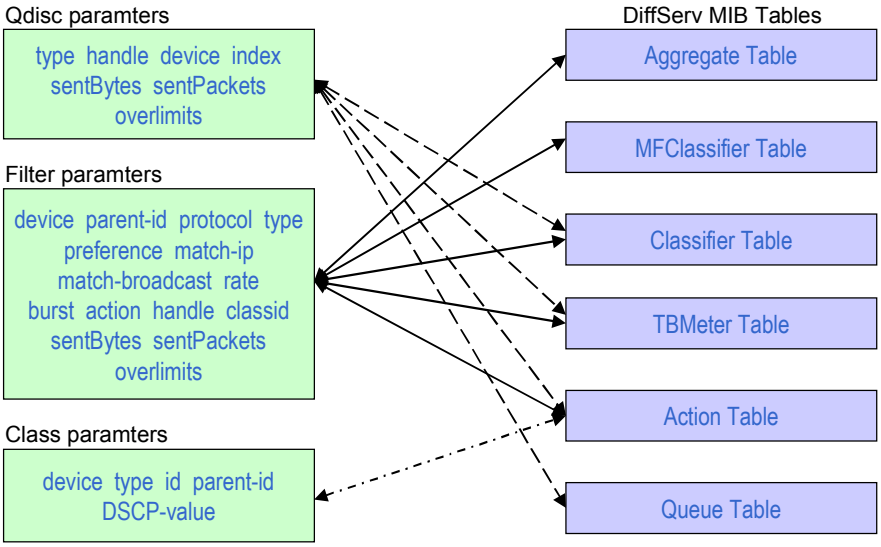



Figure 9 : Mapping Relationship Between TC Paramters and DiffServ MIB Tables

### 4.3. WEB-BASED DIFFSERV MANAGER

We have developed a DiffServ manager as an NMS server and a Web-based GUI application as an NMS client. The integration of Web technology within network management systems has benefits of ubiquitous, easy-to-use, and user-friendly interfaces. A Web server has been integrated in the NMS server (see Figure 6). Static HTML files and dynamic CGI scripts organize the Web-based user interfaces in the client's Web browser. User requests are collected via CGI scripts and processed in the DiffServ manager. Figure 10 shows a top-level Web interface of our DiffServ manager.

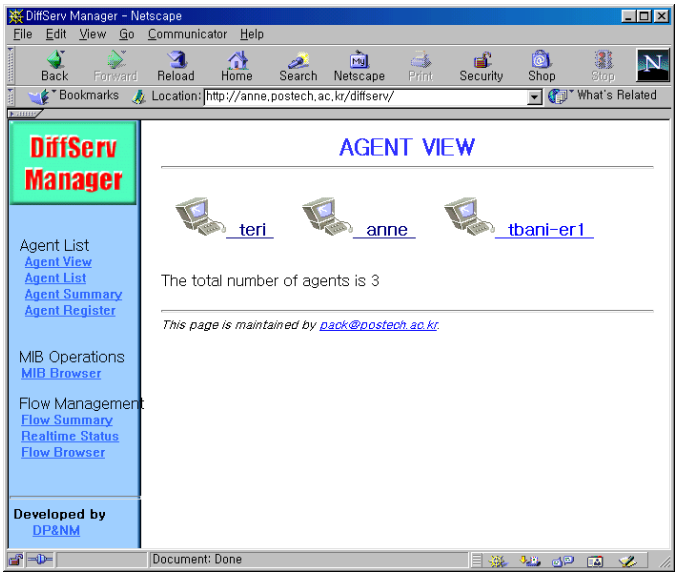



Figure 10 : Web-Based DiffServ Manager Interface

The manager functions consist of the following subfunctions.

- Agent list management
- Agent information reporting
- MIB browser
- Flow information reporting
- Network flow browser
- Realtime statistics reporting

Agent list management keeps a record of DiffServ agents managed by a DiffServ manager. The manager is able to add, modify, or delete an entry for a DiffServ agent. Figure 11 is a screenshot of the agent list management function. The DiffServ manager has system name, IP address, SNMP port number, and community name of each DiffServ agent.

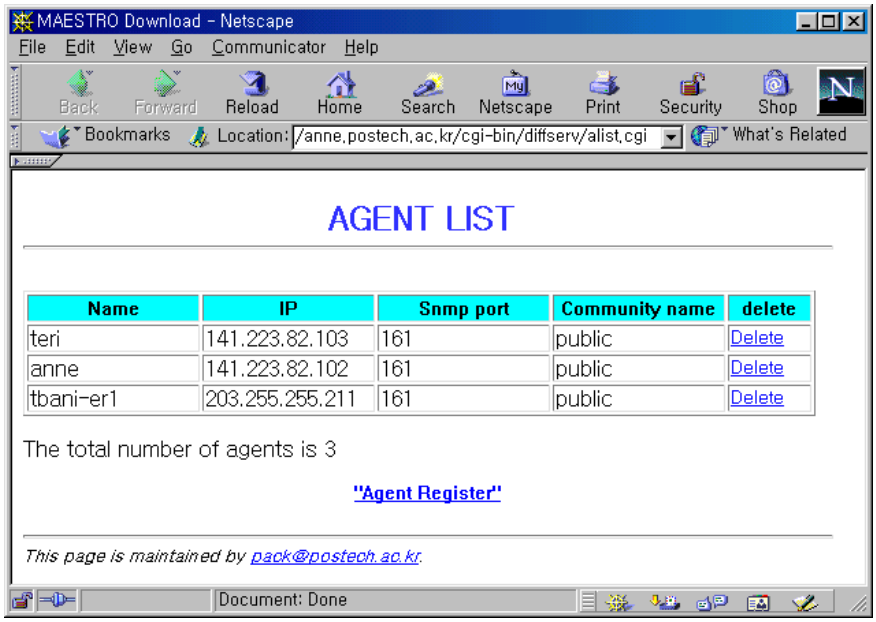



Figure 11 : Agent List Management

Agent information reporting provides an overall agent status to users. The reporting contains general information such as an IP address, a system name, and installed network interfaces of a DiffServ router. Flow descriptions summary information and MIB browser also included in this function as in Figure 12.

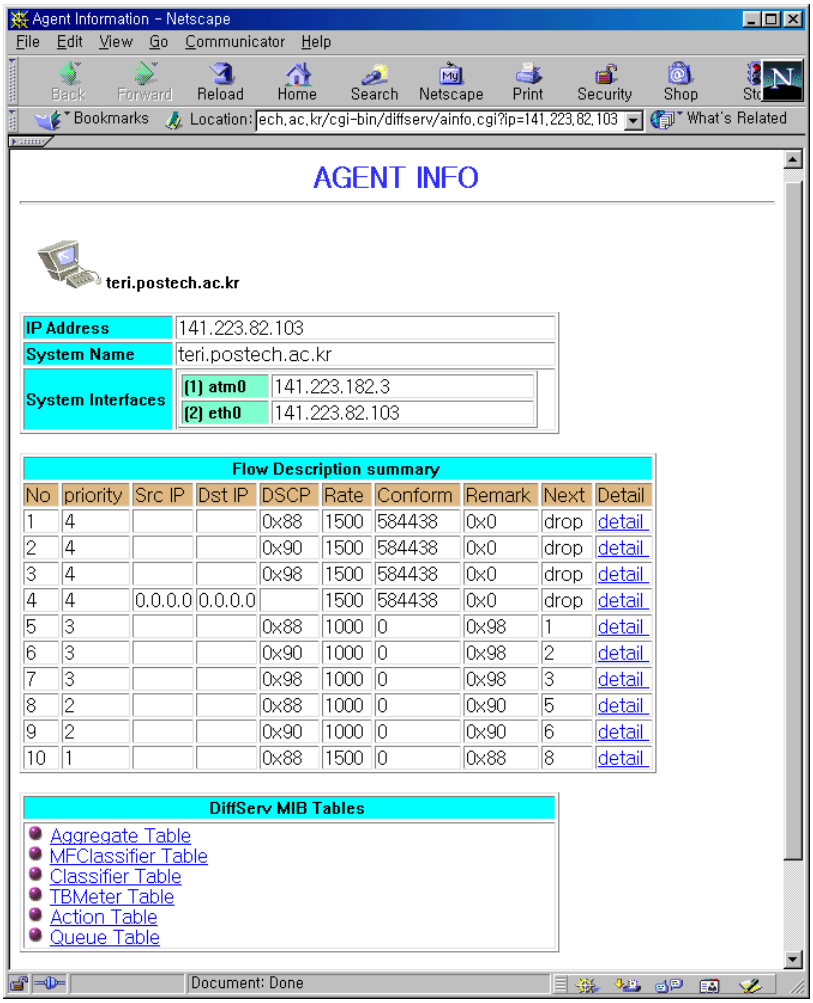
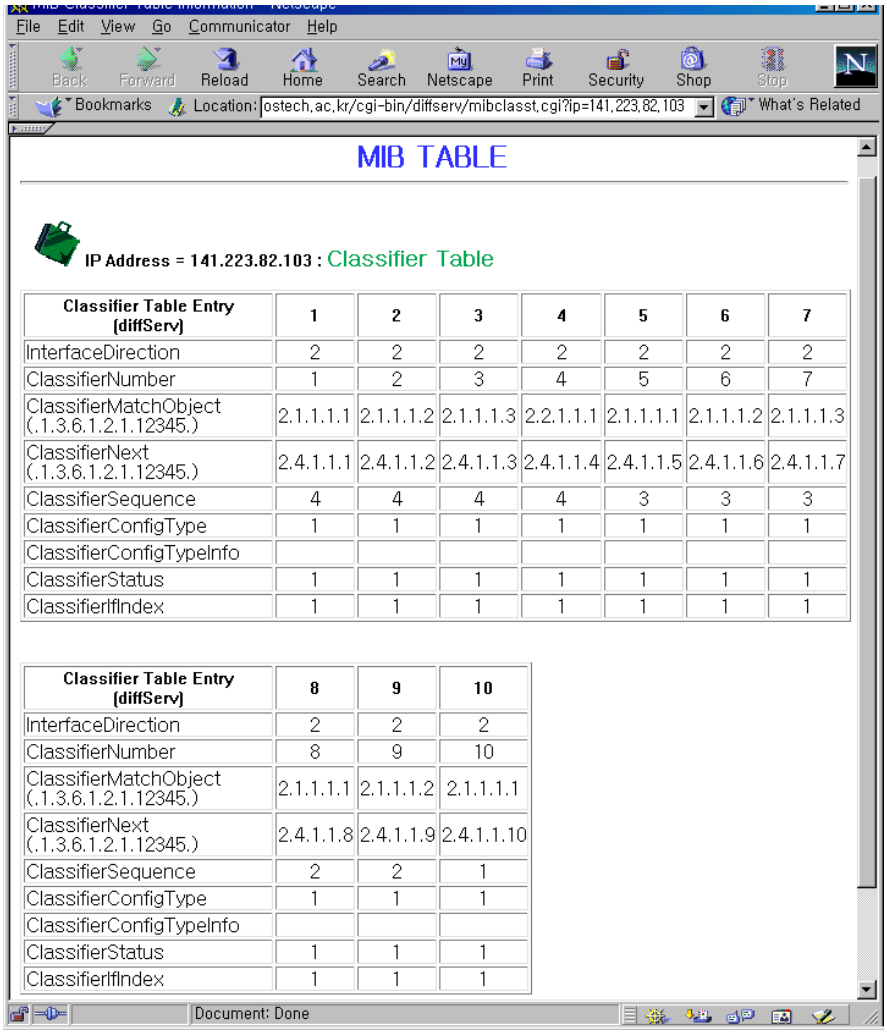



Figure 12 : Agent Information Reporting

The MIB browser is a basic SNMP management function. When a manger wants to look up specific MIB object values, this MIB browser can be used efficiently. All the six DiffServ tables are listed as in Figure 12. Selecting each table shows current values of MIB objects in each DiffServ MIB table. Figure 13 illustrates the MIB browser for the classifier table with 10 different classifiers.




| Classifier Table Entry [diffServ] | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| InterfaceDirection | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| ClassifierNumber | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| ClassifierMatchObject (.1.3.6.1.2.1.12345.) | 2.1.1.1.1 | 2.1.1.1.2 | 2.1.1.1.3 | 2.2.1.1.1 | 2.1.1.1.1 | 2.1.1.1.2 | 2.1.1.1.3 |
| ClassifierNext (.1.3.6.1.2.1.12345.) | 2.4.1.1.1 | 2.4.1.1.2 | 2.4.1.1.3 | 2.4.1.1.4 | 2.4.1.1.5 | 2.4.1.1.6 | 2.4.1.1.7 |
| ClassifierSequence | 4 | 4 | 4 | 4 | 3 | 3 | 3 |
| ClassifierConfigType | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ClassifierConfigTypeInfo | | | | | | | |
| ClassifierStatus | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ClassifierIfIndex | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Classifier Table Entry [diffServ] | 8 | 9 | 10 |
|---|---|---|---|
| InterfaceDirection | 2 | 2 | 2 |
| ClassifierNumber | 8 | 9 | 10 |
| ClassifierMatchObject (.1.3.6.1.2.1.12345.) | 2.1.1.1.1 | 2.1.1.1.2 | 2.1.1.1.1 |
| ClassifierNext (.1.3.6.1.2.1.12345.) | 2.4.1.1.8 | 2.4.1.1.9 | 2.4.1.1.10 |
| ClassifierSequence | 2 | 2 | 1 |
| ClassifierConfigType | 1 | 1 | 1 |
| ClassifierConfigTypeInfo | | | |
| ClassifierStatus | 1 | 1 | 1 |
| ClassifierIfIndex | 1 | 1 | 1 |

Figure 13 : MIB Browser

Flow information reporting enables a user to easily understand characteristics of DiffServ flows a DiffServ router handles. Since DiffServ MIB tables are linked together as in Figure 5, it is difficult to see a specific traffic flow in the MIB tables. The flow information reporting makes it easy by extracting MIB values according to flow concepts. Every different flow is listed in the flow information reporting window as in Figure 14 and users can also see the real-time statistics of flows.

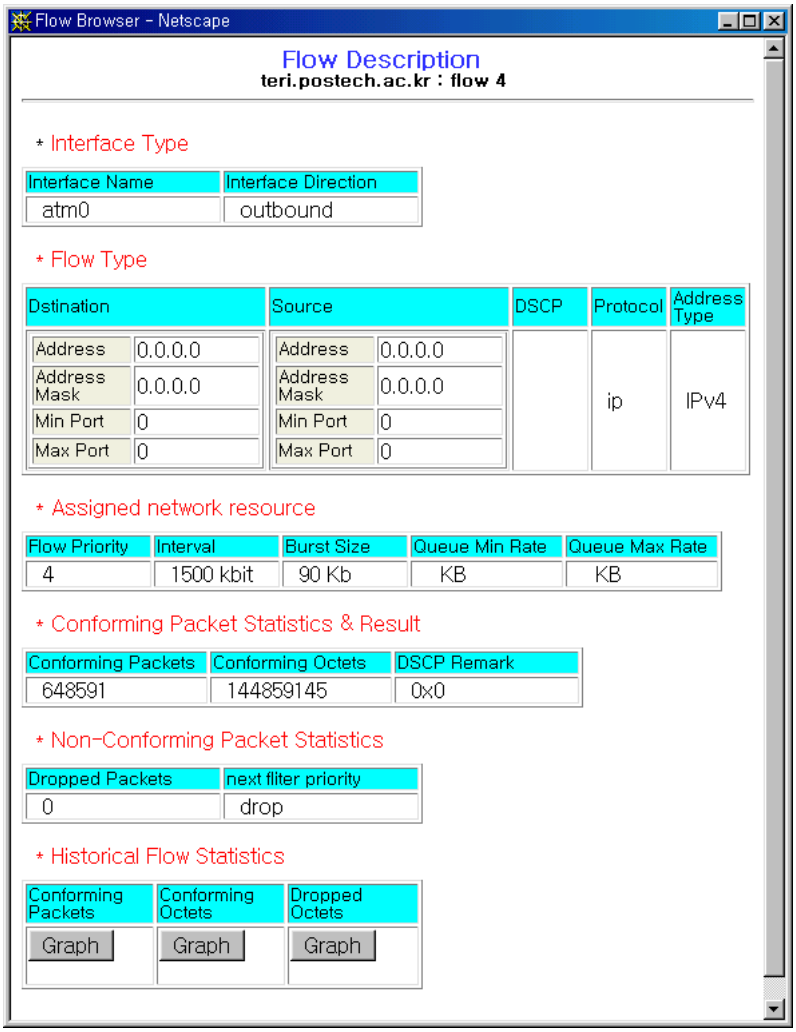



Figure 14 : Flow Information Reporting

The network flow browser provides network management viewpoint of current DiffServ flows. By combining routing information and DiffServ MIB values from every DiffServ router within a DiffServ domain, the manager is able to construct a topological status of current DiffServ flows. Classified by all the available DHCP values, every DiffServ flow in a DiffServ domain are easily represented in the network flow browser window, as in Figure

15.

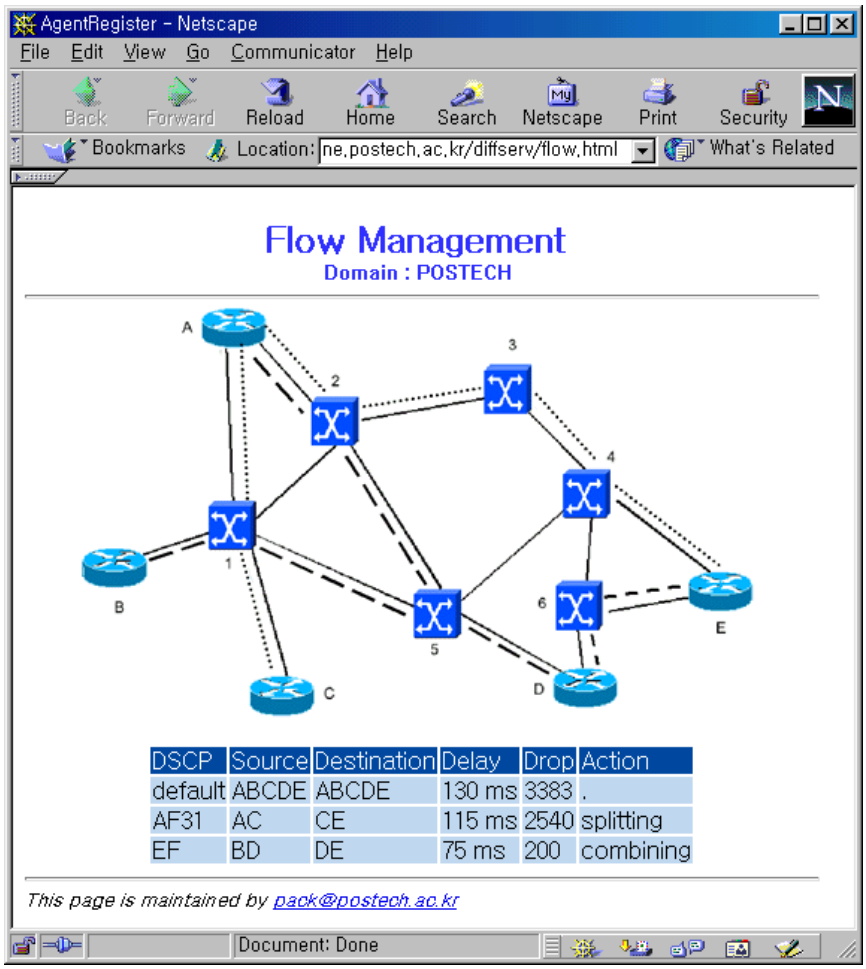
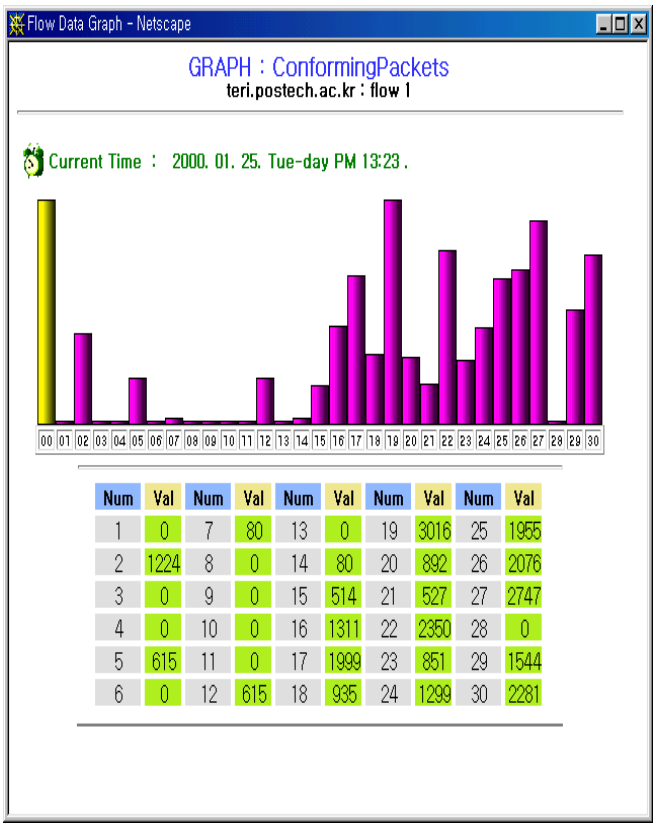



Figure 15 : Network Flow Browser

In Figure 15, there are five edge routers and six core routers under a domain named POSTECH. Current DiffServ flows in this domain are default, AF31, and EF. For each flow, both source and destination routers, as well as the topological interconnection status is represented as a dynamically modified map in Web browsers. Delay and drop performance are recorded as well. The performance parameters of a DiffServ network flow are calculated and summarized in the manager by collecting performance parameters of routers participating in the DiffServ flow path. Action represents intermediate flow characteristics when two or more flows affects each other. Splitting means a DiffServ flow is divided to several DiffServ flows in a certain point and combining means several DiffServ flows are joined in a certain point. These operations will be further refined and extended.

Realtime statistics reporting generates realtime graphs of statistical MIB variables. For every DiffServ flow in every DiffServ router, the Action MIB table contains the number of

conforming packets, the number of conforming octets, and the number of dropped packets. The DiffServ manager periodically retrieves values from every DiffServ router it manages and stores them as history records. When the DiffServ clients request to view the data, the DiffServ manager generates a statistical graph for each MIB variable. Figure 16 shows 30 realtime records and a bar graph of number of conforming packets of a DiffServ flow.




| Num | Val | Num | Val | Num | Val | Num | Val | Num | Val |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 7 | 80 | 13 | 0 | 19 | 3016 | 25 | 1955 |
| 2 | 1224 | 8 | 0 | 14 | 80 | 20 | 892 | 26 | 2076 |
| 3 | 0 | 9 | 0 | 15 | 514 | 21 | 527 | 27 | 2747 |
| 4 | 0 | 10 | 0 | 16 | 1311 | 22 | 2350 | 28 | 0 |
| 5 | 615 | 11 | 0 | 17 | 1999 | 23 | 851 | 29 | 1544 |
| 6 | 0 | 12 | 615 | 18 | 935 | 24 | 1299 | 30 | 2281 |

Figure 16 : Realtime Statistics

## V. CONCLUSION AND FUTURE WORK

Differentiated Services (DiffServ) is gaining acceptance as a promising solution for providing QoS support in the Internet. Although the operational architecture is now standardized by IETF, the management architecture is not yet fully standardized and needs to be refined and extended. This paper proposes a way to manage DiffServ in the SNMP framework. Since the current research efforts from the IETF DiffServ working group mainly focuses on the operational and functional descriptions of DiffServ, a detailed management framework for DiffServ is urgently needed. First, we have overviewed management concepts for DiffServ by categorizing management operations in the layered architecture and we then

selected several requirements for the DiffServ management system. The proposed DiffServ management system has a three-tier architecture and uses the DiffServ MIB in SNMP framework. Further, we have designed and developed a DiffServ agent system working in Linux platform and a Web-based manager system which manages several DiffServ agent systems. Management interfaces running in any Web browser enable users to control DiffServ routers conveniently.

In order to improve DiffServ management system, we are currently working on the following topics [4].

Performance evaluation of the developed management system needs to be considered. Because general DiffServ routers handle a huge amount of high-speed traffics, the DiffServ agent must not affect the routing performance of the DiffServ routers. A DiffServ management system needs to be implemented in such a way as to minimize performance degradation factors.

Integrating with the policy framework is highly recommended. To simplify the system, we have excluded the policy management features in this paper, but the policy framework needs to be integrated with the current SNMP framework for flexible and intelligent configuration and adaptation of DiffServ routers. Future work includes studying the meta-information model for policy representation and designing policy operational modules.

And finally, there are already several proprietary implementation of DiffServ routing functions from hardware vendors. We designed plans to integrate our system with commercial products so that our system will be a feasible solution for managing DiffServ in the next Internet structure.

## REFERENCES


[1] D. Perkins and E. McGinnis, Understanding SNMP MIBs, Prentice-Hall, 1997.

[2] W. Stalling, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, 3rd Edition, Addison-Wesley, 1999.

[3] R. Rajan et al., "A Policy Framework for Integrated and Differentiated Services in the Internet," IEEE Network, September/October 1999, pp.36-41.

[4] T. Choi, Y. Jung, and S. Sohn, "An Architecture of a QoS Management System for Next Generation Internet," KNOM Review, Vol. 3, No. 1, December 1999, pp.1-9.

[5] F. Baker, K. H. Chan, and A. Smith, "Management Information Base for Differentiated Services Architecture," IETF Internet-Draft, draft-ietf-diffserv-mib-01.txt, October 1999.

[6] UCD-SNMP Homepage, http://ucd-snmp.ucdavis.edu/.

[7] W. Almesberger, "Linux Network Traffic Control – Implementation Overview," ftp://lrcftp.epfl.ch/pub/people/almesber/pub/tcio-current.ps, April 1999.

[8] W. Almesberger, J. H. Salim, and A. Kuznetsov, "Differentiated Services on Linux," IETF Internet-Draft, draft-almesberger-wajhak-diffserv-linux-01.txt, June 1999.

[9] J. Heinanen, "Use of IPv4 TOS Octet to Support Differential Services," IETF Internet-Draft, draft-heinanen-diff-tos-octet-01.txt, November 1997.

[10] K. Nochols et al., "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF RFC 2474, December 1998.

[11] S. Blake et al., "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.

[12] Y. Bernet et al., "A Framework for Differentiated Services," IETF Internet-Draft, draft-ietf-diffserv-framework-02.txt, February 1999.

[13] J. Heinanen et al., "Assured Forwarding PHB Group," IETF RFC 2597, June 1999.

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

[15] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," IETF RFC 2638, July 1999.

[16] Y. Bernet et al., "Requirements of Diff-serv Boundary Routers," IETF Internet-Draft, draft-bernet-diffedge-01.txt, November 1998.

[17] M. Daniele et al., "Internet Endpoint MIB," IETF Internet-Draft, draft-ops-endpoint-mib-02.txt, November 1999.

[18] S. Brim, B. Carpenter, and F. Le Faucheur, "Per Hop Behavior Identification Codes," IETF Internet-Draft, draft-ietf-diffserv-phbid-00.txt, October 1999.

[19] IETF DiffServ Working Group Homepage, http://www.ietf.org/html.charters/diffserv-charter.html.

[20] Y. Bernet, A. Smith, and S. Blake, "A Conceptual Model for Diffserv Routers," IETF Internet-Draft, draft-ietf-diffserv-model-01.txt, October 1999.

[21] D. Grossman, "New Terminology for Diffserv," IETF Internet-Draft, draft-ietf-diffserv-new-terms-00.txt, October 1999.

[22] R. Braden et al., "ReSerVation Protocol (RSVP) Version 1 Functional Specification," IETF RFC 2205, September 1997.

[23] J. Boyle et al., "The COPS (Common Open Policy Service) Protocol," IETF Internet-Draft, draft-ietf-cops-07.txt, August 1999.

[24] R. Yavatkar et al., "COPS Usage for Differentiated Services," IETF Internet-Draft, draft-ietf-rap-cops-pr-00.txt, December 1998.
[25] R. Rajan et al., "Schema for Differentiated Services and Integrated Services in Networks," IETF Internet-Draft, draft-rajan-policy-qosschema-01.txt, April 1999.
[26] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," IETF RFC 1633, June 1994.
[27] W. Almesberger, Differentiated Services on Linux, Internet Web site, http://lrcwww.epfl.ch/linux-diffserv/.
[28] B. Carpenter and D. Kandlur, "Diversifying Internet Delivery," IEEE Spectrum, Vol. 36, No. 11, November 1999, pp.57-61.
[29] J. W. Hong, J. Y. Kong, T. H. Yun, J. S. Kim, J. T. Park and J. W. Baek, "Web-based Intranet Services and Network Management," IEEE Communications Magazine, Vol. 35, No. 10, October 1997, pp. 100-110.
[30] S. Radhakrishnan, "Linux – Advanced Networking Overview – Version 1," a technical paper of Department of Electrical Engineering and Computer Science, University of Kansas, August 22, 1999.
[31] G. Dhandapani and A. Sundaresan, "Netlink Sockets – Overview," a technical paper of Department of Electrical Engineering and Computer Science, University of Kansas, September 13, 1999.
[32] ITU-T Recommendation M.3010, "Principles for a Telecommunications Management Network," 1996.
[33] ITU-T Recommendation M.3200, "Maintenance: Telecommunications Management Network, TMN Management Services: Overview," 1992.
[34] ITU-T Recommendation M.3400, "TMN Management Functions," 1997.