Providing Packet-Loss Guarantees in Differentiated Services Architectures

by

Haiqing Chen

A thesis submitted to the

Department of Computing and Information Science

in conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

April 2001

Abstract

The Integrated Services (IntServ) and Differentiated Services (DiffServ) models have been introduced by Internet Engineering Task Force (IETF) to meet the increasing demand for delivering Quality of Service (QoS) over the Internet. The IntServ model has a scalability problem as it is based on individual flows. The DiffServ approach alleviates this problem by providing QoS based on flow aggregates.

Two services, Premium and Assured, have been introduced in the DiffServ model. Premium service provides low delay and low jitter service by reserving the peak rate of the user flows. Thus it is more expensive and is only used for high-demand applications. Assured service provides customers with a relatively reliable service, but without any quantitative guarantee.

In this thesis, we propose a new type of service, namely the *Loss Guaranteed* (LG) service, for the Differentiated Services architecture. The Loss Guaranteed service can provide a quantitative QoS guarantee in terms of loss rate without per-flow based resource reservation. A signaling protocol conforming to the DiffServ model, along with measurement-based admission control are designed to implement the Loss Guaranteed service. Because this service does not allocate resources according to the peak traffic requirement of each flow, it can achieve a high utilization level and, at the same time, provide a loss bound to the flows requesting this service. An extensive simulation model has been developed to study the performance and viability of our proposed service model. The results show that the Loss Guaranteed service can achieve a high level of utilization while still reliably keeping packet loss within the desired target. Indeed, we are able to provide packet-loss guarantees within the DiffServ architecture without the need for explicit resource reservation.

Keywords: DiffServ, Premium Service, Assured Service, Call Admission Control, Token Bucket, Buffer Management, Simulation.

Acknowledgements

I would like to express my deep gratitude to my advisors, Dr. Hossam Hassanein and Dr. Hussein T. Mouftah, for their patience, encouragements and assistance that guided me toward the final accomplishment of the work.

I would also like to thank the Department of Computing and Information Science at Queen's University and Communication and Information Technology Ontario (CITO) for their financial supports.

I am greatly indebted to my parents for their loves, supports and encouragements that accompanied me all the way here.

Finally, I thank all my friends and fellow students for their kindly assistance and helpful discussions.

List of Acronyms

AF	Assured Forwarding
BB	Bandwidth Broker
BE	Best Effort
BF	Burstiness Factor
CBQ	Class-Based Queueing
DiffServ	Differentiated Services
DSCP	DiffServ Code Point
EF	Expedite Forwarding
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
ISP	Internet Service Provider
Kb	Kilobit

LG	Loss Guaranteed
Mb	Megabit
MBAC	Measurement-Based Admission Control
MF	Multi-Field
PHB	Per Hop Behavior
QoS	Quality of Service
RED	Random Early Detection
RIO	RED with In and Out
RSVP	Resource reSerVation Protocol
SLA	Service Level Agreement
TCA	Traffic Conditioning Agreement
TCP	Transmission Control Protocol
TOS	Type Of Service
Tspec	Traffic specification
UDP	User Datagram Protocol
VPN	Virtual Private Networks
WFQ	Weighted Fair Queueing

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The current Internet delivers one type of service, best effort, to all traffic. Traffic is processed as quickly as possible, but there is no guarantee of Quality of Service (QoS) in terms of performance, throughput and/or latency for individual or aggregated flows.

Nowadays, customers demand a wide range of information types to be transferred over the Internet, such as voice, music, video, graphics, streaming media traffic, Java scripts, etc. Each type of information has different quality of service requirements. At the same time, and with the rapid transformation of the Internet into a commercial infrastructure, the service expectations of the Internet customer became widely variable. For example, the DotCom companies are willing to make an investment to provide more reliable and predictable services to customers. Similarly, the users of IP telephony and videoconferencing are willing to pay a higher price to improve the performance of such

applications. Meanwhile, there are still many more users who only want to pay as little as possible for basic services, such as email exchange or web surfing.

Motivated by the changes in user expectations and Internet applications, there is a growing demand to replace the current best effort service with a model in which users, applications, or individual packets are differentiated based on their service needs. Two broad paradigms for quality of service in next-generation Internet have emerged: Integrated Services (IntServ), and Differentiated Services (DiffServ).

The IntServ model [1] provides Quality of Service to individual packet flows through resource reservation and admission control mechanisms. Resource ReSerVation Protocol (RSVP) [2,3], which is being implemented for IP routers, is used as the signaling protocol in IntServ. The basic concept of the IntServ model is the enhancement of the existing IP routers with resource reservation capabilities, thus giving the Internet a connection-oriented character. Hence, operations like policing, shaping, admission control and QoS management are provided by RSVP routers on a per-flow basis [31]. However, in a large-scale network with millions of connected users, the number of IP sessions handled by a core RSVP router can be very large. Therefore, the execution of the above functions for every active flow in a core IP router leads to poor performance and to non-scalable network architecture.

The Differentiated Services (DiffServ) model [4] was introduced to alleviate the difficulties associated with the IntServ model. This approach focuses primarily on aggregates of flows in the core routers, with the intention of differentiating between service classes rather than providing absolute per-flow QoS measures. Though the access routers (or first hop routers) can still process packets on a per-flow basis, the core routers do not maintain per-flow state and process traffic based on a small number of Per Hop Behaviors (PHB) encoded in the packet header [5]. Therefore, the DiffServ model is more scalable as only a limited number of service classes exist in the network. The amount of state information is proportional to the number of classes rather than the number of flows. Moreover, it is easier to implement because sophisticated classification, marking, policing and shaping operations are only needed at the network boundary.

Premium service and Assured service are the first two types of services other than the best effort service, which have been discussed within the Internet community. Premium service [4,6] provides low-delay and low jitter service by guaranteeing the peak rate of user flows [7]. It is expected that the Premium service traffic would be allocated a small percentage of the total network capacity, but would be priced much higher than other traffic. One use of such a service might be to create "virtual leased lines", saving the cost of building and maintaining a separate network. This service can also be used for commercial applications, such as video broadcasts and voice-over-IP. The Expedited Forwarding PHB [8] defined by the Internet Engineering Task Force (IETF) can be used to build Premium service.

A potential disadvantage of Premium service is its weak support for bursts and the fact that users have to pay even if they are not using the bandwidth. The Assured service model [10] tries to offer a service that cannot guarantee bandwidth but provides assurance that high priority packets receive preferential treatment over lower priority packets [11]. IETF defined the Assured Forwarding PHB [9] which can be used to build Assured service. A queue management mechanism, Random Early Detection (RED) with multiple thresholds [11,12], is introduced for the implementation of this service. During periods of network congestion, packets from the low-profile traffic have higher drop probability than those from the high-profile traffic. This then assures the high-profile packets can always gain a relatively higher throughput under such a situation. However, Assured service cannot provide any quantitative guarantee to the traffic.

In this thesis, we introduce a new type of service within the DiffServ architecture. This service is intended to provide a more predictable Quality of Service than Assured service in terms of loss rate. The service guarantees a loss bound to user flows as long as the flows conform to the traffic specification (Tspec) [13]. Hence we call it the *Loss Guaranteed* (LG) service. The motivation of our research is based on the following two facts:

- The flows generated by many Internet applications are very bursty in nature, but the burstiness tends to be smoothed out with increased traffic aggregation and multiplexing.

- Many applications are tolerant to occasional packet loss and/or delay violation.

By using the first argument above, we can develop a new service, which takes advantage of traffic aggregation by not allocating the bandwidth based on peak rates of individual flows. The second argument allows us to loosen the loss and delay bounds committed to user flows in order to achieve a high multiplexing gain. A measurement-based admission control (MBAC) is designed to implement this service.

The role of any admission control algorithm is to ensure that admitting a new flow into a resource constrained network does not violate service commitments made by the network to existing flows. There are two basic approaches to admission control: *parameter-based* and *measurement-based* admission control. Parameter-based admission control computes the amount of network resources required to support a set of flows given a priori flow characteristics. Measurement-based admission control relies on measurements of observed behavior of the current aggregated traffic. The main criterion used in evaluating any admission control algorithm is how well it fulfills its primary role of ensuring that service commitments are not violated. The simplest way to ensure complete commitment conformance is to allocate enough resources to meet the worst-case requirement of each flow. This is the admission control scheme used in the Premium service model. For bursty sources, this scheme results in low network utilization. Hence, a second evaluation

criterion is the level of network utilization an admission control algorithm can achieve while still meeting its service commitments.

The admission control scheme we designed for providing the Loss Guaranteed service does not allocate resources according to the peak traffic requirement of each flow. Instead, the measured quantities are used to reflect the actual resource requirements of the aggregate flows. The effects of the newly arriving flow on existing flows are calculated from the traffic specification (Tspec) of the new flow. Admission control tries to maximize utilization while still keeping the loss bound commitments to existing flows on aggregate basis.

The signaling protocol designed for the Loss Guaranteed service is different from RSVP in that it does not require per flow resource reservation. Per-flow state information does not need to be maintained within the network. Thus the periodical "reserv" message required in RSVP to refresh/update flow's reservation state is not necessary in our signaling protocol. Similarly, the "teardown" message in RSVP to clear the state information is also not required.

The Loss Guaranteed service is not intended to replace any existing services proposed for the Differentiated Services architecture, but is an additional class of service that can provide a quantitative Quality of Service, which is not available with Assured service. When compared with Premium service, the Loss Guaranteed service does not provide

bandwidth guarantees of users' peak rates. Packets using this service may experience occasional delay and loss within a controlled level. However, the Loss Guaranteed service can achieve a much higher utilization level than Premium service.

The rest of the thesis organized as follows. In Chapter 2, we introduce the Integrated Services (IntServ) and Differentiated Service (DiffServ) architectures. For the IntServ model, we discuss the necessary mechanisms and components with the focus on the signaling protocol, RSVP, and the two service classes, Guaranteed service and Controlled-Load service. Then the problems with the IntServ model are noted. For the DiffServ model, we describe the architecture, resource allocation and the two proposed services, Premium and Assured. Chapter 2 also gives an introduction to measurement-based admission control.

In Chapter 3, we introduce the Loss Guaranteed service proposed for DiffServ. The signaling process and the admission control scheme, which include admission control algorithm, loss estimation process and measurement process, are described in detail.

In Chapter 4, we provide a performance evaluation of the admission control scheme designed for the Loss Guaranteed service using a comprehensive simulation study.

Finally, Chapter 5 provides conclusions and future directions.

Chapter 2

Related Work

In this chapter, we overview the Integrated Services (InServ) and Differentiated Services (DiffServ) architectures. For the IntServ architecture, Resource ReSerVation Protocol (RSVP) is described in detail and the problems with the IntServ model are pointed out. The DiffServ architecture and its two main services, Premium and Assured, are described. Measurement-based admission control schemes are also discussed. Such schemes are essential to our proposed Loss Guaranteed service model.

2.1 Integrated Services

The IntServ approach [1] focuses on individual packet flows, which represent streams of IP packets having the same source and destination address, the same TCP/UDP port number and the same protocol field. In this approach, each flow can request specific

levels of service from the network. The levels of service are quantified as a minimum service rate, a delay bound and/or a maximum loss rate. The network grants or rejects a traffic flow based on availability of resources and the guarantees provided to other flows.

The IntServ architecture includes three major components:

- The admission control unit, which checks if the network can grant the service request.
- The packet forwarding mechanisms, which performs the per-packet operations of flow classification, shaping, scheduling and buffer management in the routers
- Resource ReSerVation Protocol (RSVP), which sets up some flow state, e.g., bandwidth reservations, filters or accounting in the routers a flow goes through.

The IntServ model proposes two service classes in addition to the best effort service. These are:

1) Guaranteed service for applications requiring a fixed delay bound.
2) Controlled-Load service for applications requiring probabilistic delay bounds.

2.1.1 Guaranteed Service

Guaranteed service, defined in [14], provides firm bounds on the queueing delays that a packet will experience in a router. In this service model, a source is characterized by

Tspec[13], which takes the form of a token bucket specification plus a peak rate, a minimum policed unit and a maximum packet size. The token bucket specification [35] has two parameters: the token rate, r, and the token bucket depth, b. Each token represents a single bit; sending a packet consumes as many tokens as there are bits in the packet. A source is said to conform to its token bucket specification if no packet is sent when the token bucket is empty. The requested Guaranteed service is characterized by a transmission rate, R, at which packets will be transmitted. In essence, a session requesting Guaranteed service requires that the bits in its packets be guaranteed a forwarding rate of R bits/sec. Given that traffic is specified using a token bucket specification and a guaranteed rate of R is being requested, it is also possible to bound the maximum queueing delay at the router.

The Guaranteed service traffic must be policed at the network access points to ensure conformance to their Tspecs. This ensures that non-conforming traffic does not interfere with other conforming flows causing them to miss their contracts. Guaranteed service is intended for applications with stringent real-time delivery requirement such as audio and video applications that have fixed "play-out" buffers and are intolerant of any datagram arriving after their playback time. In fact, Guaranteed service provides a service that is similar to traditional telecommunications circuit switching. The flow effectively sees a dedicated wire of bandwidth between the sender and receiver.

2.1.2 Controlled-Load Service

A session receiving Controlled-Load service will receive "a quality of service closely approximating the QoS that same flow would receive from an unloaded network element" [15]. Given a fixed amount of bandwidth allocated to serve the Controlled-Load traffic, admission control must be employed to control admittance of flows using the service. A flow requesting Controlled-Load service sends its Tspec to the routers in the network. If the flow is admitted, each router commits to offer the flow a service equivalent to that seen by a best-effort flow on a lightly loaded network. The important difference from the traditional best effort service is that the performance of a Controlled-Load flow does not noticeably deteriorate as the network load increases. By contrast, a best-effort flow would experience progressively worse service (higher delay and loss) as the network load increases. Controlled-Load service is intended for applications that can tolerate a certain amount of loss and delay provided it is kept to an acceptable level [36].

2.1.3 Resource Reservation Protocol

Resource ReSerVation Protocol (RSVP) [2] has been developed as a signaling protocol for resource reservation and is one of the major components of the IntServ architecture. The signaling process is illustrated in Figure 2.1. The sender sends a PATH Message to the receiver specifying the characteristics of the traffic. Every intermediate router along the path forwards the PATH Message to the next hop determined by the routing protocol. Upon receiving a PATH Message, the receiver responds with a RESV message to request resources for the flow. Every intermediate router along the path can accept or reject the

request of the RESV Message. If the request is rejected, the router will send an error message to the receiver, and the signaling process will terminate. If the request is accepted, link bandwidth and buffer space are allocated for the flow and the related flow state information will be installed in the router.

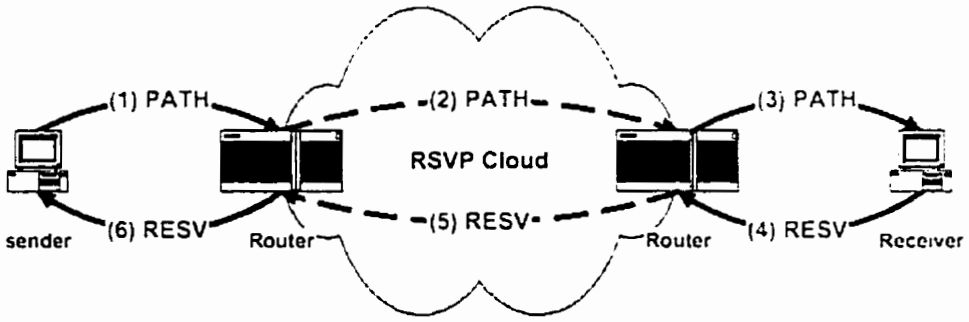


Figure 2.1 RSVP Signaling

RSVP uses a so-called "soft-state" whereby reservations timeout in the absence of refresh RESV messages from end systems within a certain timeout period. In the refreshing state, a RESV message received by a node is not propagated upstream immediately. Instead it is held until the end of some refresh period epoch whereupon it is merged with RESV messages that were received from other downstream nodes during the last refresh period epoch to create a single refresh RESV message to be propagated upstream. This merging mechanism ensures that the number of the refreshing RESV messages received by the sender in the refreshing state is determined by the number of its next hops rather than the number of receivers.

2.1.4 Drawbacks of the IntServ Architecture

The IntServ model provides users with an end-to-end QoS through resource reservation and admission control mechanisms along the data path. It makes it possible for applications with different levels of QoS requirement to share the use of the Internet. However, as this service model is based on individual flows, it suffers from the following drawbacks:

- Scalability: The amount of state information increases proportionally with the number of flows. This places a huge storage and processing overhead on the routers. Therefore, this architecture does not scale well in Internet backbone networks.
- Cost: RSVP, admission control, Multi-Field (MF) classification and packet scheduling have to be supported by all the routers. This places a very high computational requirement on the routers.
- Deployment: Partial and incremental deployment is not possible for Guaranteed service. The current router architecture has to be modified in order to support RSVP in the IntServ architecture.

2.2 Differentiated Services

To alleviate the problems with the IntServ architecture, Differentiated Services has been introduced. The idea behind the Differentiated Services (DiffServ) architecture is based on aggregation of flows. Service discrimination is provided through the differentiation

between service classes rather than providing absolute per-flow QoS management. Core routers do not maintain per-flow states and forward packets based on a small number of Per Hop Behaviors (PHB) encoded in the packet header [5,16]. Per-flow based operations need only be processed at the edge of the network.

2.2.1 Differentiated Services Architecture

The DiffServ architecture 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 (classes of service). Each behavior aggregate is identified by a single DiffServ Code Point (DSCP) [5,16]. Within the core of the network, packets are forwarded according to the per-hop behavior associated with the DSCP.

2.2.1.1 Differentiated Service Domain

A DiffServ domain is a contiguous set of DiffServ nodes, which operate with a common service provisioning policy and set of PHB groups implemented on each node (see Figure 2.2). A DiffServ domain has a well-defined boundary consisting of DiffServ boundary nodes, which classify and possibly condition ingress traffic to ensure that packets, which transit the domain, are appropriately marked to select a PHB from one of the PHB groups supported within the domain. Nodes within the DiffServ domain select the forwarding behavior for packets based on their DSCP, mapping that value to one of the supported PHBs.

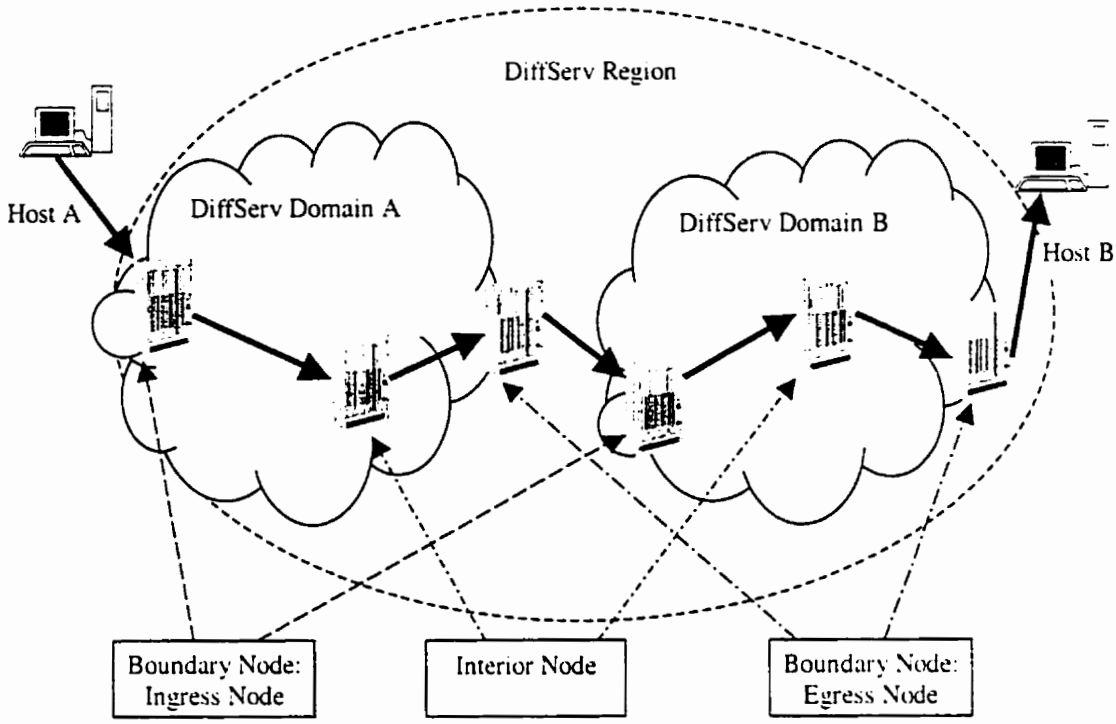


Figure 2.2 DiffServ Architecture

A DiffServ domain consists of DiffServ boundary nodes and DiffServ interior nodes. DiffServ boundary nodes interconnect the DiffServ domain to other DiffServ or non-DiffServ-capable domains, while DiffServ interior nodes only connect to other DiffServ interior or boundary nodes within the same DiffServ domain. Both DiffServ boundary nodes and interior nodes must be able to apply the appropriate PHB to packets based on the DSCP. In addition, DiffServ boundary nodes may be required to perform traffic conditioning functions as defined by a Traffic Conditioning Agreement (TCA) between their DiffServ domain and peering domain, which they connect to. DiffServ boundary nodes act both as an ingress node and an egress node for different directions of traffic. Traffic enters a DiffServ domain at an ingress node and leaves a DiffServ domain at an

egress node. An ingress node is responsible for ensuring that the traffic entering the DiffServ domain conforms to the TCA between the DiffServ domain and the other domain to which the ingress node is connected. An egress node may perform traffic conditioning functions on traffic forwarded to a directly connected peering domain. depending on the details of the TCA between the two domains.

2.2.1.2 Differentiated Service Region

A DiffServ region is a set of one or more contiguous DiffServ domains. A DiffServ region is capable of supporting Differentiated Services along paths which span the domains within the region. The DiffServ domains in a DiffServ region may support different PHB groups internally and different codepoint to PHB mappings. The peering DiffServ domains must each establish a peering Service Level Agreement (SLA), which defines a TCA. The TCA specifies how transit traffic from one DiffServ domain to another is conditioned at the boundary between the two DiffServ domains.

2.2.1.3 Traffic Classification and Conditioning

DiffServ are extended across a DiffServ domain boundary by establishing an SLA between an upstream network and a downstream DiffServ domain. The SLA may specify packet classification and re-marking rules and may also specify traffic profiles and actions to traffic streams, which are in- or out-of-profile. The TCA between the domains is derived from this SLA. The packet classification policy identifies the subset of traffic

which may receive a Differentiated Service by being conditioned and/or mapped to one or more behavior aggregates within the DiffServ domain.

Traffic conditioning performs metering, shaping, policing and/or re-marking to ensure that the traffic entering the DiffServ domain conforms to the rules specified in the TCA, in accordance with the domain's service provisioning policy. The extent of traffic conditioning required is dependent on the specifics of the service offering, and may range from simple codepoint re-marking to complex policing and shaping operations.

Classifiers:

Packet classifiers select packets in a traffic stream based on the content of some portion of the packet header. Two types of classifiers are defined. The BA (Behavior Aggregate) Classifier classifies packets based on the DSCP only. The MF (Multi-Field) classifier selects packets based on the value of a combination of one or more header fields, such as source address, destination address, DSCP field, protocol ID, source port and destination port numbers, and other information such as incoming interface. Classifiers are used to "steer" packets matching some specified rule to an element of a traffic conditioner for further processing. Classifiers must be configured by some management procedure in accordance with the appropriate TCA. The classifier should authenticate the information which it uses to classify packets.

Traffic Profiles:

A traffic profile specifies the temporal properties of a traffic stream selected by a classifier. It provides rules for determining whether a particular packet is in- or out-of-profile. For example, a profile based on a token bucket may look like:

DSCP = X, use token-bucket r, b

The above profile indicates that all packets marked with *DSCP X* should be measured against a token bucket meter with rate *r* and burst size *b*. In this example out-of-profile packets are those packets in the traffic stream which arrive when insufficient tokens are available in the bucket.

Different conditioning actions may be applied to the in-profile packets and out-of-profile packets, or different accounting actions may be triggered. In-profile packets may be allowed to enter the DiffServ domain without further conditioning; or, alternatively, their DSCP may be changed. Out-of- profile packets may be queued until they are in-profile (shaped), discarded (policed), marked with a new codepoint (re-marked), or forwarded unchanged while triggering some accounting procedure. Out-of-profile packets may be mapped to one or more behavior aggregates that are "inferior" in some dimension of forwarding performance to the BA into which in-profile packets are mapped.

Traffic Conditioner:

A traffic conditioner may contain the following elements: meter, marker, shaper, and dropper. A traffic stream is selected by a classifier, which steers the packets to a logical

instance of a traffic conditioner. A meter is used to measure the traffic stream against a traffic profile. The state of the meter with respect to a particular packet (e.g., whether it is in- or out-of-profile) may be used to affect a marking, dropping, or shaping action. When packets exit the traffic conditioner of a DiffServ boundary node the DSCP of each packet must be set to an appropriate value. Figure 2.3 shows the block diagram of a classifier and traffic conditioner. The functions of traffic conditioners include[1]:

- Meters: Traffic meters measure the temporal properties of the stream of packets selected by a classifier according to a traffic profile specified in a TCA. A meter passes state information to other conditioning functions to trigger a particular action for each packet that is either in- or out-of-profile.

- Markers: Packet markers set the DiffServ field of a packet to a particular DSCP, adding the marked packet to a particular behavior aggregate. The marker may be configured to mark all packets, which are steered to it to a single DSCP, or may be configured to mark a packet to one of a set of DSCPs used to select a PHB in a PHB group, according to the state of a meter.
- Shapers: Shapers delay some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. A shaper usually has a finite-size buffer, and packets may be discarded if there is no sufficient buffer space to hold the delayed packets.
- Droppers: Droppers discard some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. This process is known as

"policing" the stream. Note that a dropper can be implemented as a special case of a shaper by setting the shaper buffer size to zero (or a few) packets.

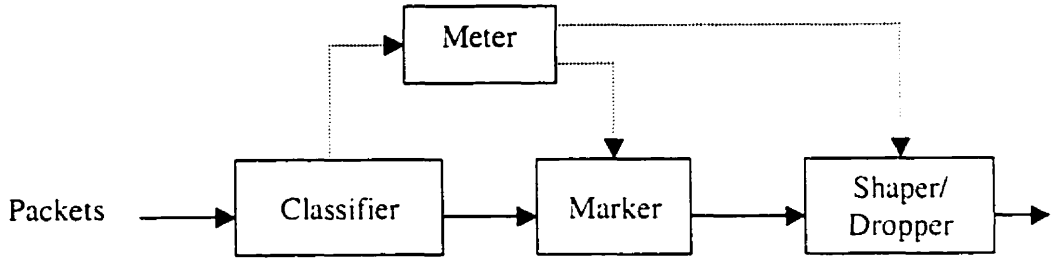


Figure 2.3 Logical View of a Packet Classifier and Traffic Conditioner

Traffic conditioners are usually located within DiffServ ingress and egress boundary nodes, but may also be located in nodes within the interior of a DiffServ domain, or within a non-DiffServ-capable domain.

2.2.1.4 Per-Hop Behavior

A per-hop behavior (PHB) [4,5] is a description of the externally observable forwarding behavior of a DiffServ node applied to a particular DiffServ behavior aggregate. The PHB is the means by which a node allocates resources to behavior aggregates, and it is on top of this basic hop-by-hop resource allocation mechanism used to construct effective services.

PHBs may be specified in terms of their resource (e.g., buffer or bandwidth), priority relative to other PHBs, or in terms of their relative observable traffic characteristics (e.g.,

delay or loss). These PHBs may be used as building blocks to allocate resources and should be specified as a group (PHB group) for consistency. PHB groups will usually share a common constraint applied to each PHB within the group, such as a packet scheduling or buffer management policy. A single PHB defined in isolation is a special case of a PHB group.

The IPv4 Type of Service (ToS) octet [17,34] or the IPv6 Traffic Class octet [18] is used to map the PHB of the packets. Six bits of this byte are used to define the PHB. The codepoint for Expedited Forwarding PHB has been defined in [8] and codepoints for Assured Forwarding PHB [9] have been classified in four classes with three levels of drop precedence for each class. Furthermore, the usage scenarios and encoding rule of the rest of the codepoints are described in [16].

2.2.2 Resource Allocation in Differentiated Services

In order for a customer to receive Differentiated Services from its Internet Service Provider (ISP), it must have a Service Level Agreement (SLA) with its ISP [4,6]. In general, SLAs are complex business-related contracts that cover a wide range of issues, inlcuding network availability guarantees, payment models, and other legal and business necessities. To facilitate QoS specification within the contract, an SLA will contain a service level specification (SLS) that characterizes aggregate traffic profiles and the PHB to be applied to each aggregate. DiffServ can provide end-to-end quality of service by

carefully enforcing the aggregate traffic contracts between domains and ensuring that new sources of marked packets do not cause traffic profiles to be violated.

A SLA can be static or dynamic. If the number of different services offered is small and the contracts relatively static, the SLSs between domains may be manually negotiated and the edge devices configured by the network administrators. Dynamic SLAs require SLS negotiation, admission control and device configuration being processed automatically. A new component called a Bandwidth Broker (BB) is introduced in [6.19] for ensuring that resources within the DiffServ domain and on links connecting adjacent domains are properly provisioned and not oversubscribed. A BB maintains information relating to the SLSs that are defined between a DiffServ domain and its customers. Customers include local users, as well as the adjacent networks that provide connectivity to other parts of the Internet. The BB uses this SLS information to configure the router in the local DiffServ domain, and to make admission control decisions.

Before marked packets from a data source are admitted to a DiffServ domain, the source must signal its local BB to initiate a service reservation. The potential source is authenticated and subjected to local admission control policies. If the service reservation is admitted locally, the BB may initiate an end-to-end reservation request along the chain of BBs in the networks to be traversed by the data flow. When a network-wide admission control decision has been made, the BB will configure the routers in the DiffServ domain to support the requested service profile. The bandwidth broker allows separately

administered DiffServ domains to manage their network resources independently, yet still cooperate with other domains to provide dynamically allocated end-to-end QoS.

2.2.3 Services of the Differentiated Services Approach

In addition to the existing best effort service, the Expedited Forwarding PHB [8] and Assured Forwarding PHB [9] have been defined by IETF to implement Premium service and Assured services, respectively.

2.2.3.1 Premium Service

Premium service provides low-delay and low-jitter service for customers who generate fixed peak bit rate traffic. This service appears to the endpoints like a point-to-point connection or a "virtual leased line". Each customer will have a SLA with its ISP. The SLA specifies a desired peak bit rate for a specific flow or an aggregation of flows. The customer is responsible for not exceeding the peak rate. Otherwise, excess traffic will be dropped. The ISP guarantees that the contracted bandwidth will be available when traffic is sent. Premium service is suitable for Internet telephony, video conferencing or for creating virtual leased lines for Virtual Private Networks (VPN) [20]. The expedited forwarding PHB described in [8] is used to build Premium service.

To implement Premium service, First-hop routers have the task of classifying packets received from end systems, i.e., analyze if Premium service shall be provided to such

packets or not. If yes, the packets are tagged as Premium service (P-bit) and the flow is shaped according to the requested peak rate. After the shaping, the P-bits of all packets are set for the flow that is allowed to use Premium service. The exit routers of the customer domain may need to reshape the traffic to make sure that the traffic does not exceed the peak rate specified by the SLA. When packets are transmitted to the ISP's border router, or ingress router, the policing function will be performed to check whether the user's border router remains below the negotiated or contracted bandwidth. Excess traffic is dropped. There are two queues at the border routers, one for packets with the P-bit set (P-queue) and one for all other traffic. Packets in P-queue have transmission priority over others. Thus the implementation of two queues in every router of the network (ISP and user network) equals to the realization of a virtual network for the Premium service traffic.

2.2.3.2 Assured Service

Assured service provides customers with relatively reliable services even in time of network congestion. Like Premium service, customers will have SLAs with their ISPs. The SLAs will specify the amount of bandwidth allocated for the customers. Customers are responsible for deciding how their applications share the allocated bandwidth.

To implement Assured service, classification and policing are performed at the ingress router of the ISP networks. If the Assured service traffic does not exceed the bit-rate specified by the SLA, they are considered as in-profile. Otherwise, the excess packets are

considered as out-of-profile. Both in-profile and out-of-profile packets are put into an Assured Queue to avoid out of order delivery. The Assured Queue is managed by a queue management scheme call RED with In and Out (RIO)[10,32].

RED (Random Early Detection) is a queue management scheme proposed in [12]. It allows routers to drop packets randomly before the queue is full, which will trigger the TCP flow control mechanisms at different end hosts to reduce their rates. By doing so, RED can prevent the queue at routers from overflowing, hence avoiding the tail-drop behavior described in [21].

RIO basically maintains two RED algorithms, one for in-profile packets and one for out-of-profile packets. There are two thresholds for each queue. When the queue size is below the first threshold, no packets are dropped. When the queue size is between the two thresholds, only out-of-profile packets are randomly dropped. When the queue size exceeds the second threshold, indicating possible network congestion, both in- and out-of profile packets are randomly dropped, but out-of-profile packets have higher drop probability. Therefore, in-profile packets will have a lower loss rate than out-of-profile packets even in cases of congestion. Consequently, customers will receive a predictable service from the network if their traffic conforms to the SLA. When there is no congestion, out-of-profile packets will be also delivered.

More recently, IETF defined four classes for Assured Forwarding PHB groups with three levels of drop precedence for each class. A Multiple Threshold RED scheme has been proposed and evaluated in [11,22,23,33,38].

2.3 Admission Control Schemes

The role of any admission control algorithm is to ensure that admittance of a new flow into a resource constrained network does not violate service commitments made by the network to previously admitted flows. In IntServ networks, admission control is incorporated with RSVP and resource reservation mechanisms to achieve end-to-end QoS provisioning. DiffServ requires another type of admission control, which is known as policy-based admission control [24], upon receiving a resource request. Policy-based admission control is used by network managers and service providers to monitor, control and enforce the usage of network resources and services based on policies derived from criteria such as the identity, ingress points, traffic/bandwidth requirements, security considerations, etc. We call the former resource-based admission control to differentiate it from the policy-based admission. In the rest of our thesis, the term admission control will refer to resource-based admission control, unless otherwise specified.

2.3.1 Admission Control Algorithms

There are two basic approaches to admission control: *parameter-based* and *measurement-based*. Parameter-based admission control calculates the amount of network resources

required to support a set of flows based on a priori knowledge of flow characteristics. Measurement-based admission control uses measurement mechanisms to determine the actual current traffic load to make the admission decision. In this section, one parameter-based admission control and three measurement-based admission control schemes are overviewed. In Section 2.3.2, we introduce three measurement mechanisms designed for the measurement-based admission control.

2.3.1.1 Simple Sum Algorithm

The Simple Sum algorithm is a parameter-based admission control algorithm. This admission control algorithm simply ensures that the sum of requested resource does not exceed the link capacity. Let r^{α} be the request rate of an incoming flow α, v the sum of the reserved rate and μ the link bandwidth. This algorithm accepts the new flow if the following check succeeds:

$$v + r^{\alpha} < \mu$$

Instead of using the peak rate that is used in Guaranteed service, this algorithm reserves a bandwidth equivalent to the token rate of the flows. This gives higher link utilization, but flows may suffer from occasional delay and packet loss due to burstiness of traffic flows. To ensure low queueing delays, the weighted fair queueing (WFQ) [25,37] scheduling discipline is implemented with this admission control algorithm. WFQ assigns each flow its own queue served at its own reserved rate, thereby, isolating flows from each other's bursts. Due to its simplicity, this admission control algorithm is the most widely implemented by switch and router vendors.

2.3.1.2 Measured Sum Algorithm

This algorithm uses measurement to estimate the current aggregate rate of existing flows instead of a calculated sum of request rates of existing flows. If $\hat{v}$ is the measured value of the current link rate, the admission condition of this algorithm is:

$$\hat{v} + r^{\alpha} < \upsilon\mu$$

where υ is referred to as the utilization target in [26]. In a simple M/M/1 queue, variance in queue length diverges as the system approaches full utilization. A measurement-based approach is doomed to fail when delay or packet loss rate variations are exceedingly large, which will occur at very high utilization. It is thus necessary to identify a utilization target and require that the admission control algorithm to keep link utilization below this level. In this thesis, we will use this algorithm to check if bandwidth requirements of admitted flows will be met.

2.3.1.3 Equivalent Bandwidth

This measurement-based algorithm computes the equivalent bandwidth for a set of flows using the Hoeffding bounds [27,28]. Hoeffding theorem is one of a family Chernoff-style bounds that considers a sum of random variables, and gives an upper bound on the tail of the distribution. To estimate an upper bound on the equivalent capacity, this algorithm uses a result derived from Hoeffding.

Let X1, X2 ..., X_n be independent, and let $0 \le X_i \le p_i$. Then for $t > 0$,

$$Prob[S \geq \mu_S + nt] \leq e^{-2n^2t^2/\sum_{i=1}^{n}(p_i)^2}$$

where n is the number of samples, μ_S is the average of the sum and S is the value of the sample. This relation quantifies the probability that the sum of the n random variables will be greater than the average of the sum by nt or more. To find t such that

$$Prob[S \geq \hat{v} + nt] \leq \varepsilon .$$

This is satisfied if

$$e^{-2n^2t^2/\sum_{i=1}^{n}(p_i)^2} \leq \varepsilon$$

where ε is a parameter that controls the degree of risk of the admissions control procedure. This is satisfied for

$$nt < \sqrt{\frac{\ln(1/\varepsilon)\sum_{i=1}^{n}(p_i)^2}{2}}$$

Thus the equivalent capacity $\hat{C}_H$ based on peak rate policing is

$$\hat{C}_H(\mu_S, \{p_i\}_{1\leq i\leq n}, \varepsilon) = \hat{v} + \sqrt{\frac{\ln(1/\varepsilon)\sum_{i=1}^{n}(p_i)^2}{2}}$$

where $\hat{v}$ is the measured average arrival rate of existing traffic and ε is the probability that arrival rate exceeds the link capacity. The author of [28] indicates that network administrators should choose a proper value of ε based on an accumulation of experience with the realistic traffic.

When a new flow α requests admission, it will be accepted if

$$\hat{C}_H + p^{\alpha} \leq \mu$$

For flows described by a token bucket filter (r, b) but not peak rate, [28] derives their peak rate ($\hat{p}$) from the token bucket parameters using the equation:

$$\hat{p} = r + b/T$$

where T is a user-defined averaging period.

2.3.1.4 Bounded Delay Algorithm

This algorithm is introduced in [26]. Whereas the previous three algorithms bound bandwidth usage in their admission decisions, this algorithm bounds both bandwidth usage and experienced delay. When a new flow α requests admission to the network, this algorithm uses the measured sum algorithm to check that the bandwidth requirements of admitted flows will be met; then it checks that the delay bound (D) of existing traffic will not be violated by the admittance of the new flow. Presumably the delay bound of a flow is defined as $D = b/r$, where r and b are its token bucket parameters. The flow α is denied admission if it fails the following check:

$$\hat{D} + \frac{b^{\alpha}}{\mu} < D$$

where $\hat{D}$ is the measured delay and $\frac{b^{\alpha}}{\mu}$ is the amount of time to transmit a full token bucket from the new flow. Upon admittance of a new flow, the delay measure is adjusted by adding $\frac{b^{\alpha}}{\mu}$ to the delay estimate.

2.3.2 Measurement Mechanisms

The measurement mechanism is one of the key components of admission control scheme. The admission control decision is based on the result of measurement process. A conservative measurement leads to strict admission control, which results in low utilization. On the other hand, a less conservative measurement results in a more aggressive admission policy that admits more traffic to the network. Consequently, a high utilization level can be achieved in such situation but at the expense of increasing the chance of service commitment violation.

2.3.2.1 Time-window Measurement

In [26], a simple time-window measurement mechanism is used to measure network load with the "Measured Sum" algorithm. As shown in Figure 2.4, an average load is computed every S sampling period. Each measurement window, T, contains several such sampling periods. At the end of a measurement window, the highest average from the last observed window is used as the load estimate for the next time window. When a new flow is admitted to the network, the estimate is increased by the parameters of the new request. If a newly computed average is above the estimate, the estimate is immediately raised to the new average. At the end of every time window, T, the estimate is adjusted to the actual load measured in the previous window. A smaller S makes the measurement mechanism more sensitive to bursts, thus gives higher maximal averages, resulting in a more conservative admission control algorithm. Likewise, a larger T keeps longer

measurement history, resulting in a more conservative admission control algorithm. To get a statistically meaningful number of samples, it is suggested to keep $T / S \geq 10$ [26].

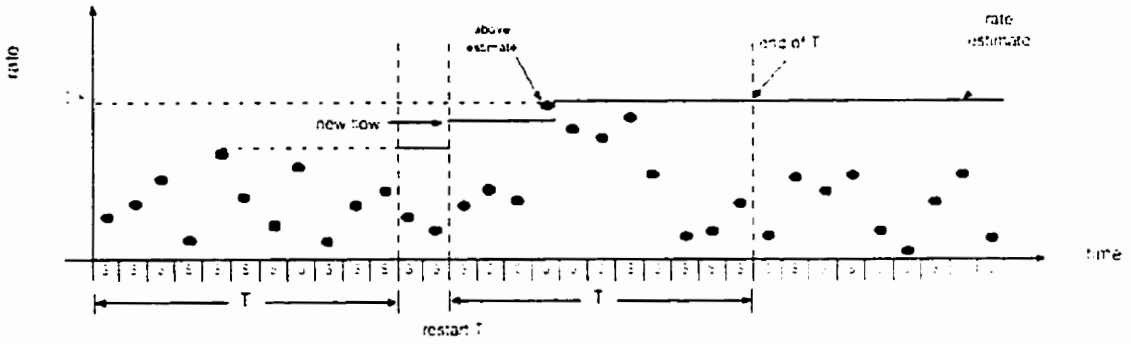


Figure 2.4 Time-window Measurement Mechanism

2.3.2.2 Adaptive-window Measurement

In time-window measurement mechanisms, a too-small window lowers the admission threshold as soon as the traffic thins out, leaving the network dangerously exposed to bursts. On the other hand, a too large measurement window leads to an excessively conservative scheme, which causes low link utilization. Perhaps the most important factor is that without a priori knowledge of the traffic, it is difficult to determine an appropriate length of the measurement window. For these reasons, a measurement window that adapts to network traffic is desirable.

A two-level adaptive-window measurement mechanism is introduced in [29]. The first-level algorithm compares the measured rate with the trigger rate, which is selected by the

second-level algorithm. If the measured rate is lower than trigger rate, the algorithm continuously shrinks the length of the measurement window, resulting a less conservative measurement, until the amount of traffic generated by accepted calls reaches a trigger value. This trigger value is actually a rate value, smaller than the output link capacity, providing an early warning that the system is about to reach a load level. The first-level algorithm then reacts by enlarging the measurement window until the measured rate drops below the trigger, at which point the window can be shrunk again. The second level algorithm lowers and raises the trigger rate in search of a good operating point in response to traffic fluctuations. The instantaneous delay violation percentage and long term delay violation percentage are used in [29] as inputs to adjust the trigger point. If the instantaneous delay percentage exceeds the target value, the trigger value in the first-level algorithm is decreased. Otherwise, the algorithm checks whether the long-term delay violation percentage is currently under the target value. If it is, the window trigger value is raised. If it is not, no action is performed. The algorithm automatically adjusts the length of the measurement window to adapt to different traffic conditions, thus achieving stable link utilization while ensuring QoS requirements.

2.3.2.3 Exponential Averaging

In [28], to compute admission decisions with the equivalent bandwidth approach, an estimate of the average arrival rate is used, instead of instantaneous bandwidth. The average arrival rate $\hat{v}^S$ is measured once every S sampling periods. The average arrival rate is then computed using an infinite impulse response function with weight w:

$$\hat{v}' = (1 - w) * \hat{v} + w * \hat{v}^S$$

The time constant for this is

$$t = -1 / \ln(1 - w) * S$$

which assumes that the traffic rate changes abruptly from 0 to 1, and then remains at the value 1. After t seconds, the estimated average arrival rate will have reached 63% $(1 - 1/e)$ [30] of the new arrival rate value of 1. A larger w makes the averaging process more adaptive to load changes; a smaller w gives a smoother average by keeping a longer history. Similar to the Time-window measurement mechanism, a smaller S makes the measurement mechanism more sensitive to bursts, and a larger S may result in lower averages. On the other hand, a larger S lets the measurement mechanism keep a longer history because the averaging process is invoked less often.

2.4 Summary

Integrated Services (IntServ) and Differentiated Services (DiffServ) are the two service models proposed in order to replace the current best effort service model. The IntServ model can provide end-to-end Quality of Service guarantee through resource reservation for each individual flow. But it does not scale well because it requires that all routers keep the state information for the flows resulting in high complexity in router implementation in the core network. DiffServ solves this problem by differentiating the traffic into several classes. QoS is provided through maintaining service commitments made to each class of service rather than individual flows. Router operations are based on per hop behaviors (PHB) encoded in the packet header.

Two service models have been introduced in the DiffServ architecture. Premium service and Assured service. Premium service provides a service equivalent to a "virtual leased line" service. This is realized through guaranteeing the peak rate of user flows. Assured service assures the in-profile packets have high throughput even during congestion by applying queue management schemes which favor in-profile packets. However, without any admission control mechanism, Assured service only can provide relative Quality of Service to traffic but without any quantitative guarantees.

In this thesis, we introduce the Loss Guaranteed (LG) service for the DiffServ architecture. This service is intended for applications that do not require the absolute bandwidth guarantee of user's desired peak rate. Such applications are developed to be

tolerant to occasional loss bound violation, so users can choose to use the less expensive service, which provides a relatively lower Quality of Service guarantee, instead of using Premium service. The LG service provides a user with a "soft" quantitative guarantee in terms of loss rate as long as the user flow conforms to its Tspec. A signaling protocol, which is in conformance with the DiffServ model, along with a measurement-based admission control, is designed to implement the Loss Guaranteed service. The detailed scheme for the Loss Guaranteed service is described in the following chapter.

Chapter 3

Loss Guaranteed Service

In this chapter, we first discuss the limitations and disadvantages of Premium service and Assured service to certain kind of applications. Then the need, objectives and advantages of the new service we propose are presented. In Section 3.2, we give an overview of the design of the Loss Guaranteed (LG) service. The components necessary for the implementation of the service are introduced. We also describe how the new service fits into the DiffServ architecture. Section 3.3 explains the signaling protocol designed for the LG service. In Section 3.4 the flow admission control scheme is described in detail. This includes three parts: the admission control algorithm, the loss estimate process and the measurement process. We then discuss the effects of several performance tuning knobs in our admission control scheme. Finally, we describe a possible resource allocation scheme for the LG service.

3.1 Design Objectives

Assured service provides users with bandwidth assurances but without strict guarantees that bandwidth will always be available. The RIO packet dropping algorithm works well when resource allocation is fair and all hosts are assumed to implement the TCP congestion control mechanism properly. However, during congestion periods, when the queue length of the in-profile packets exceeds a certain level, packets are dropped aggressively. This will trigger the TCP congestion control mechanism to limit the flow rate, which will cause low throughput for the individual flows. A misbehaving source which does not implement TCP congestion control properly, or has no such control at all, can still gain more bandwidth by injecting excessive traffic onto the network. This may be the case for UDP flows which do not have the rigid congestion control mechanism that is used for TCP. Indeed, the QoS that can be provided by Assured service becomes less predictable under such situations.

Premium service can provide users with low-delay and low-jitter service by guaranteeing a bandwidth level, which is equivalent to the peak rate of the traffic flow. This service is more expensive compared with Assured service, since:

1) only a small percentage of the total network capacity is allocated for Premium traffic, and
2) given the small amount of bandwidth allocated for Premium service, guaranteeing the peak rate further limits the number of flows which can be accepted simultaneously.

Premium service is ideal for smooth traffic when the network allocates bandwidth close to the actual rates of the individual flows. A reasonable utilization level can be achieved in this case. However, when user flows are bursty, the bandwidth allocated for the flows are not efficiently used due to the long idle period of the flows, which results in low network utilization.

In fact, most Internet traffic does not have constant and/or continuous bandwidth requirements. Measurements of Internet traffic show that much of the traffic is very bursty [10]. A service model based on a fixed capacity does not actually meet users' needs very well. On the other hand, evidence shows that in the center of the existing Internet, at the backbone routers of the major ISPs, there is such a high degree of traffic aggregation making the bursty nature of individual traffic flows essentially invisible. This motivates us to introduce a new service model, which is called Loss Guaranteed (LG) service. This service does not allocate resources based on the characteristics of individual flows, but rather on the aggregated flows. To facilitate such a service, an admission control scheme is used. The measured quantities that reflect the flows' multiplexing effect are used to represent the actual resource requirements of the aggregate flows. The effects of the newly arriving flow on existing flows are calculated from the token bucket parameters and peak rate requirement. Compared with Premium service, the LG service can admit more flows simultaneously by taking advantage of the flows' multiplexing and aggregating effects. Thus it can achieve a much higher utilization level than Premium

service. The cost of the service to users, therefore, is expected to be much lower than the cost of Premium service.

Because the admission control scheme relies on measurement, and source behavior is not static in general, the measurement-based admission control can never provide a completely reliable loss bound. However, the applications that are requesting this service are assumed to be tolerant to occasional loss bound violation. Instead of providing an absolute bandwidth guarantee of a user's desired peak rate, this service provides the user with a "soft" guarantee as long as the user flow conforms to its Tspec.

We note that the new service we are proposing is not intended to replace any existing service currently proposed for the DiffServ architecture. The LG service is a complementary service that can provide more predictive QoS guarantees than Assured service in terms of packet loss. Yet, and since the LG service allows resource sharing among user flows, it can be a more economical choice than the peak-rate based Premium service for users who do not demand very high QoS guarantees.

Many loss sensitive applications can benefit from the LG service. For example, data services are well known to be loss sensitive. A TCP flow is very sensitive to packet loss, which will slow down the flow rate in responding to packet loss. Guaranteeing the loss rate can ensure a TCP user transmitting data at a relatively stable rate. Meanwhile, the TCP and UDP flows are treated more fairly within this service than in best effort service. This is because the LG service uses admission control to avoid congestion rather than

relying on the TCP congestion control mechanism. Finally, in video distribution systems, such as broadcast TV or HDTV, a lower packet loss rate brings higher quality of the image and audio to the service.

3.2 Design Overview

The LG service is intended for users that need a more reliable and predictable service than Assured service from their service providers. The network guarantees that the aggregate flows subscribed to this service experience low loss rate within a certain adjustable level. The LG service has two components. The first one is the signaling protocol for the resource request between the user and network, and between routers within the DiffServ region. The second is the measurement-based admission control, which controls the loss rate of the aggregate traffic subscribed to this service.

The admission control scheme consists of three major functions: the admission control algorithm, the loss estimate process and the measurement process. The admission control algorithm grants a flow admission only if both of the following two conditions are satisfied: 1) the link should not be overloaded 2) the loss rate should be within a certain controlled level. The loss estimate process estimates the effects of a new requesting flow based on its traffic specification (Tspec) and the measured quantities from the measurement process. Finally, in the measurement process, a time window measurement

mechanism is used to measure the rate, packet loss and queue length of the aggregate traffic.

As mentioned earlier, a Bandwidth Broker (BB) [19] allocates and controls the bandwidth within a DiffServ domain. BBs are also responsible for managing the messages that are sent across domain boundaries to adjacent domains' BBs. In the LG service, BBs have two major responsibilities. One is related to resource reservation and admission control, such as keeping track of network resources, temporary resource reservation, making the admission control decision and cooperating with the routers during the measurement process. The other is to pass the admission request to the adjacent domains' BBs.

Routers are only required to execute the measurement process to implement the LG service. A time-window measurement mechanism is used to measure the utilization, loss rate and queue length. The measurement results are reported to local domain's BB after each measurement window. In fact, routers are unaware of requests made by individual flows, neither do they keep flow states. Packets are forwarded based on the Per Hop Behavior (PHB) defined for the LG service. However, after a flow is accepted, BBs send notifications to routers. In response, routers restart the measurement window to allow the next measurement results to reflect the new accepted flow. This also allows the temporary parameter-based resource reservation to last at least for the duration of the measurement time window.

3.3 Signaling Protocol

Figure 3.1 illustrates the signaling process for the LG service. User Domains A and B are connected with an ISP domain. Host A in User Domain A wants to use the LG service to send data to Host B in User Domain B. The admission control mechanism is deployed at the links between R1-R2, R2-R3, R3-core router, R4-R5 and R5-R6. The service delivery process is described below. The numbers inside the circles are the setup numbers in the service delivery process.

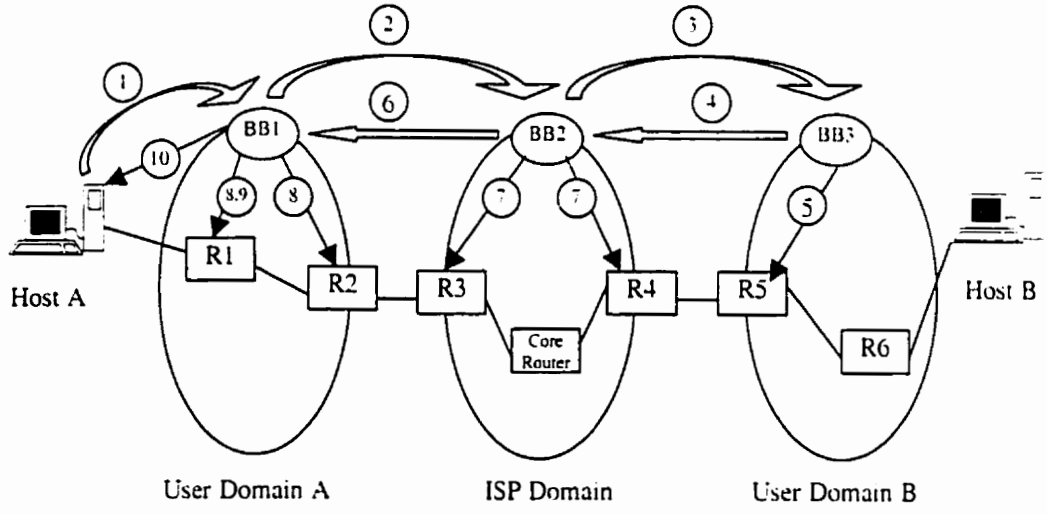


Figure 3.1 Signaling Process for LG Service

Signaling process:

1. Host A sends an admission request with its Tspec to the local Bandwidth Broker BB1.

2. BB1 makes the admission control decision. If the request is denied, an error message is sent back to host A. Otherwise, BB1 passes the request to the ISP domain's BB, which is BB2.
3. BB2 makes the admission control decision as follows:
   - If the request is denied, an error message is sent back to BB1. Host A will be notified through BB1.
   - If the request is accepted, BB2 sends the request to the destination domain's BB, which is BB3.
4. BB3 makes the admission control decision as follows:
   - If the request is denied, an error message is sent back to BB1 through BB2, sender A will be notified.
   - If the request is accepted, BB3 noticing the destination is within its own domain, sends an *accept* message back to BB2.
5. BB3 sends a *restart* message to the edge router R5, notifying the measurement mechanism in R5 to restart the measurement window.
6. BB2 sends an *accept* message back to BB1, indicating the acceptance of the request.
7. BB2 sends a *restart* message to the border routers R3 and R4, notifying the measurement mechanisms in R3 and R4 to restart the measurement windows.
8. BB1 sends a *restart* message to the edge router R2 and leaf router R1, notifying the measurement mechanisms in R1 and R2 to restart the measurement windows.
9. BB1 will also set the classification and shaping rules on leaf router R1, so that if the traffic of the admitted flow does not conform to its Tspec, R1 will shape it.

10. BB1 sends an *accept* message to Host A. Host A can then start transmitting packets.

During the packet forwarding stage, leaf router R1 performs the Multi-Field (MF) classification, which sets the code point for the LG service in the packet DSCP field. It also shapes the traffic such that it conforms to its Tspec. Ingress routers R3 and R5 perform the Behavior Aggregate (BA) classification. Traffic conditioning may be performed at egress routers R2 and R4.

3.4 Admission Control Scheme

Previous work on measurement-based admission control algorithms [26,27,28,29] are mainly concerned with providing a bounded delay service to support real-time applications. Such applications, viz. audio and video conferencing systems, are very sensitive to delay. Measurement-based admission control algorithms use observed delay as one of the inputs to make the admission decision. Since these applications are developed to be loss tolerant, the loss rate is not considered as an important factor in such admission control algorithms. Normally, the buffer size, one of the loss related parameters, is assumed to be infinite or just an experimental setting by the admission control schemes proposed for those applications.

To provide a loss guarantee, the buffer size has to be taken into account while making the admission decision. Packet loss is directly related to the buffer size of an output link. The

buffer space of the output link provides a cushion to packet bursts during periods of congestion. Such packets are later retransmitted during less congested periods. A larger output buffer can accommodate longer packet bursts and, therefore, results in lower packet loss for the aggregate flows. On the other hand, a larger buffer size may result in excessive packet delays. Hence the choice of buffer size has a significant impact on the system's performance. The admission control scheme we propose uses the buffer size and observed queue length as inputs to estimate the loss when considering a flow's request. The admission control scheme tries to ensure the admittance of the new flow does not violate loss bound commitments made to other flows using this service.

When a flow is accepted by a local domain while a global admission decision is yet unknown, to avoid traffic oversubscribing, the local BB will have to reserve the resources for the flow based on an estimated value. We call such a flow "in-processing flow". The resource reservation for an in-processing flow will be cancelled after the global admission decision for this flow is known. If the flow is accepted, BBs perform a temporary resource reservation by artificially increasing the measured value to reflect the worst-case expectations. In our admission control scheme, the admission control and loss estimate process are performed by BBs while the measurement process is implemented at routers.

3.4.1 Admission Control Algorithm

An incoming flow α is granted admission if the following two conditions hold:

1. The sum of the flow's requested rate, r^{α} , the bandwidth reserved for the in-processing flows, v_r , and current usage, $\hat{v}$, measured would not exceed the targeted link utilization level:

$$\upsilon\mu > r^{\alpha} + v_r + \hat{v} \tag{3.1}$$

   where μ is the bandwidth allocated for the LG service, υ is the utilization target.

2. The estimated loss rate as a result of accepting the new flow, $\hat{\rho}$, which is calculated by Equation 3.6 (explained in the next section), would not exceed the targeted loss rate ρ:

$$\rho > \hat{\rho} \tag{3.2}$$

3.4.2 Loss Estimate Process

In this section, we describe the process to estimate the loss rate. The average queue length is an index of link congestion. As the queue length is a dynamic value, an exponentially weighted moving average [12] is used to calculate the average queue length, hence the short-term increases in the queue size that result from bursty traffic or from transient congestion do not result in a significant increase in the average queue length. However, by measuring the average queue length, the dynamic buffer occupancy information is lost. The average queue length does not directly reflect the loss probability. Given the same average queue length, the aggregate traffic can have different

levels of burstiness. When the aggregate traffic is bursty, packets are more likely to be lost. Thus, the loss probability cannot be estimated from the average queue length. In our scheme, to save the dynamic queue occupancy information, we segment the measurement window T into a number of small equivalent time periods. Then the maximum queue length in each period is recorded to represent the queue length in this period. By doing this, the traffic burstiness information is saved and can be used later to estimate the loss rate when making the admission decision.

According to Tspec, a flow is specified by token rate r, token bucket depth b and peak rate p. A flow is said to conform to its specification if no packet arrives when the token bucket is empty. When the flow is idle or is at a lower traffic rate, tokens are accumulated up to b tokens. In our estimation process, we assume in the worst case, the flow requesting the admission will appear to be as bursty as its specification would allow it to be. That is, the flow will be idle for just enough periods to refill an empty bucket. The minimum value of this period can be calculated as $T_{off} = b/r$. Then it goes into the burst period until it exhausts all the tokens. The maximum value of the burst period can be calculated as $T_{on} = b/(p-r)$. In our admission control scheme, a requesting flow is assumed to have this burst pattern during the loss estimate process.

During a measurement window, T, the maximum number of bursts, M_b, a flow can generate is given by

$$M_b = \frac{T}{T_{on} + T_{off}} \tag{3.3}$$

The token depth b is used to represent the maximum queue built up during a burst period. Thus the loss, L_i, caused by the ith burst of flow α can be estimated as follows

$$L_i = \begin{cases} \hat{q}_i + q_r + b^\alpha - B, & \hat{q}_i + q_r + b^\alpha - B > 0, \\ 0, & \hat{q}_i + q_r + b^\alpha - B \le 0, \end{cases} \tag{3.4}$$

where,

$\hat{q}_i$ is the maximum queue length measured during the ith $T_{on} + T_{off}$ period.

q_r is the reserved buffer space for the in-processing flows.

B is the output buffer size of the link.

b^α is the token depth of flow α.

As mentioned earlier, the queue occupancy information is sampled by recording the maximum queue length during every small time segment t. t should be less than the minimum value of $T_{on} + T_{off}$ of flows. Thus, we can obtain $\hat{q}_i$ by selecting the maximum queue length from the relative samples during the ith $T_{on} + T_{off}$ period.

The total loss L^α caused by the flow α is the sum of the estimated loss caused by each burst

$$L^\alpha = \sum_{i=0}^{M_b} L_i \tag{3.5}$$

The estimated loss rate is calculated by

$$\hat{\rho} = \frac{\hat{L} + L_r + L^{\alpha}}{\hat{P} + P_r + (T \times r^{\alpha} + b^{\alpha})} \tag{3.6}$$

where,

$\hat{P}$ is the number of packets that arrived during T. This is a measured value.

P_r is the estimated number of packets that could arrive during T from the in-processing flows.

L_r is the estimated loss that could be caused by the in-processing flows.

3.4.3 Measurement Process

The measurement process takes an important role in the measurement-based admission control scheme because the admission decision is based on measured values in order to estimate the effect of accepting a new flow. Consequently, a conservative measurement results in a strict admission control. In our admission control algorithm, we need two estimate values: utilization and loss rate. To estimate the utilization, we sample the usage rate, $\hat{v}$, over a sampling period of length S packet transmission units. To estimate the loss rate, we sample the maximum queue length of every sampling period of length t packet transmission units and we also measure the actual packet loss during the measurement window period.

3.4.3.1 Measurement Values Adjustment

Whenever a new flow is admitted, we artificially increase the measured values to reflect the effects of the new flow. Thus the calculated effects of a new flow rather than the

measured effects are used until it completes an entire measurement window period without new flow arrival. We also restart the measurement window after admitting a new flow to allow the next measurement results to reflect the actual effect of this accepted flow and other possibly newly accepted flows. Figure 3.2 presents pseudo description of the algorithm for adjusting the measurement values after admitting a new flow. Basically, it adds the requested rate to the link utilization estimate and computes an estimate for loss rate according to equations 3.5 and 3.6.

```
adjust( r^α, b^α, v̂ , (q̂_0, q̂_1, ... q̂_{m-1}), L̂ ) {after admitting a new flow α}
        v̂ = v̂ + r^α ;
        // increase measured rate.
        for every i < m do
           q̂_i = q̂_i + b^α ;
        end for
        // increase measured queue lengths.
        L̂ = L̂ + L^α ;
        // increase measured loss.
        P̂ = P̂ + T × r^α + b^α
        // increase the measured number of arrival packets.
        rate_measure ( v̂ );
        queue_measure ( q̂_0, q̂_1, ... q̂_{m-1} );
        loss_measure ( L̂ );
        // restart the measurement window.
end adjust
```

Figure 3.2 Algorithm for Adjusting Measurement Values

The estimated effects of a new admitted flow are based on worst-case expectation. Therefore, when the effects of a new flow are unknown, the measurement process is rather conservative, which in turn makes the admission control scheme strict. A

temporary parameter-based resource reservation is actually performed after admitting a new flow. Such resource may be later relinquished during the following time window.

In the LG service, routers are unaware of the termination of flows. Instead of explicitly adjusting the measured values upon the termination of a flow, we allow the measurement mechanism to adapt automatically to the observed traffic. That is, the effect of a flow leaving the network will be reflected by later updates of measured values.

3.4.3.2 Rate Measurement

The measurement variable $\hat{v}$ tracks the highest sampled aggregate rate of flows. The value of $\hat{v}$ is updated on three occasions. At the end of the measurement window, we update $\hat{v}$ to reflect the maximal sampled utilization seen in the previous window. Whenever an individual utilization measurement exceeds $\hat{v}$, we immediately update $\hat{v}$ with the new sampled value. Finally, whenever a new flow is admitted, we update $\hat{v}$ by artificially adding the token rate of the new flow. Figure 3.3 is the pseudo code description of the algorithm for rate measurement. It shows the first two instances of $\hat{v}$ update. The $\hat{v}$ update after admitting a new flow is shown in Figure 3.2.

```
rate_measure ( v̂ ) { When a new measurement window is started. }
    v^S_max = 0;
    for every number of sample i < n do
        v̂^S_i = average rate measured in ith sample period
        if ( v̂^S_i > v̂^S_max ) then v̂^S_max = v̂^S_i ;
            // v̂ tracks the highest sampled aggregate rate of flows.
            if v̂^S_i > v̂ then v̂ = v̂^S_i ;
            // individual utilization measurement exceeds v̂ , immediately update
              v̂ with the new sampled value.
           end if
        end if
    end for
    v̂ = v̂^S_max ;
    //update measured rate with the highest sampled rate at the end of the
      measurement window.
end rate_measure
```

Figure 3.3 Algorithm for Rate Measurement

```
queue_measure ( q̂_0, q̂_1 .... q̂_{m-1} ) { When a new measurement window is started.}
    for every sampling period j<m do
        q̂_j^t = maximum queue length of jth t period;
        j = j + 1;
    end for
    for every j < m do q̂_j = q̂_j^t ;
    // update measured queue lengths at the end of the measurement window.
    end for
end queue_measure
```

Figure 3.4 Algorithm for Queue Length Measurement

3.4.3.3 Queue Length Measurement

The measurement variable $\hat{q}_j$ tracks the estimated maximum queue length for period j. The value of $\hat{q}_j$ is updated on two occasions. At the end of the measurement window, we update $\hat{q}_j$ to reflect the queue lengths seen in the previous window. A pseudo code description of the algorithm for queue length measurement is shown in Figure 3.4. Whenever a new flow is admitted, $\hat{q}_j$ is updated by adding a parameter value, which is equivalent to the token depth of the new flow (see Figure 3.2).

3.4.3.4 Loss Measurement

The process of measuring loss is rather straightforward. We use a variable $\hat{L}$ to track the measured loss of the previous window. The value of $\hat{L}$ is updated on two occasions. At the end of the measurement window, $\hat{L}$ is updated to reflect the packet loss that might have occurred in the previous window T (see Figure 3.5). Whenever a new flow is admitted, $\hat{L}$ is updated by adding an estimated loss quantity calculated from the loss estimate process (see Figure 3.2). A variable $\hat{P}$ is used to count the arrival number of packets of previous window. $\hat{P}$ is also increased by $T \times r^{\alpha} + b^{\alpha}$ when a new flow α is admitted (see Figure 3.2).

```
loss_measure ( L̂ ) { When a new measurement window is started. }

  while ( packet arrives and measurement window did not expire) do
      P̂^S = P̂^S + 1; // count the arrival packets.
      if (queue is full ) then L̂^S = L̂^S + 1;
      end if
  end while
  P̂ = P̂^S;
  L̂ = L̂^S;
  //update measured number of arrival packets and measured loss at the end of the
    measurement window.
end loss_measure
```

Figure 3.5 Algorithm for Loss Measurement

3.4.4 Discussion on the Performance Tuning Knobs

There are several variables used in our admission control scheme that act as the tuning knobs of the admission control performance. These are:

Utilization target υ

When the bandwidth approaches full utilization, the loss is exceedingly unpredictable. It is thus necessary to identify a utilization target and require the admission control algorithm to keep the link utilization below this level. The appropriate utilization target depends on the characteristics of the traffic. If each source's rate is small compared to the link capacity and bursts are short, the link's utilization target can be set higher. Bursty sources with large bursts will require a lower link utilization target.

Rate sampling period S

The averaging period S in the algorithm for rate measurement described in Figure 3.2 controls the sensitivity of our rate measurement. A smaller averaging period makes it easy to capture individual bursts, which results in a higher measured rate. The traffic appears to be smoother when a larger averaging period is used. Consequently, a smaller averaging period makes the admission control scheme more conservative.

Measurement Window T

In our measurement process, once the values of $\hat{v}$ or $\hat{q}$ is increased, they remain high until the end of the measurement window T. The size of T controls the adaptability of the measurement mechanism to reduced traffic load. Smaller T means more adaptability, but larger T results in greater stability. Varying T has two inter-related effects on the admission control algorithm. First, since T is the length of measurement block used to determine how long we keep the previous maximal sampled rate, increasing T makes the rate estimate more conservative, which in turn makes the admission control algorithm itself more conservative. Thus, larger T means fewer loss violations and lower link utilization. Second, T also determines how long the calculated estimate values for loss and utilization induced by a newly admitted flow are used. Furthermore, flows depart from the network during the period T. A large T makes the measurement less adaptable to the changes of the traffic.

3.5 Resource Allocation for Loss Guaranteed Service

As was introduced in Chapter 2, resource allocation in Differentiated Services architectures can be static or dynamic. Problems can arise if a static resource allocation is used for the LG service. Unlike Assured service, the LG service provides users with a quantitative QoS guaranteed through flow admission control. When the allocated resources reach a high utilization level, users will experience a high call rejection ratio. Allocating enough resources to the service will improve the call rejection ratio but it will also cause a low resource utilization level when the LG traffic is reduced. In the worst case when the traffic is unevenly distributed, flows may be consistently rejected due to one congested link along the data path. For example, in Figure 3.1, if the resource utilization of the LG service is kept high at link between R2-R3, flow requests from Host A with destination to Host B will have a high rejection ratio even though the resource utilization at other links may be quite low.

To solve such problems, we propose to use a dynamic resource allocation scheme for the LG service. Under such scheme, a local domain BB tracks the call rejection ratio of the LG service on each links. Whenever the call rejection ratio exceeds a certain threshold level on a given link, and if the downstream node is within the local domain, the local domain BB will allocate more bandwidth for the service. If the downstream node is in an adjacent domain, the local BB will use a signaling protocol, e.g., RSVP, to negotiate more bandwidth for the LG service based on the Service Level Agreement (SLA) between the two domains. On the other hand, whenever the call rejection ratio drops to a

low level and link utilization remains low, BBs will operate oppositely by decreasing the allocated bandwidth for the service. We note that the decrease of the bandwidth allocation should be done conservatively, to ensure that decrease of the bandwidth allocation does not cause a short term loss bound violation.

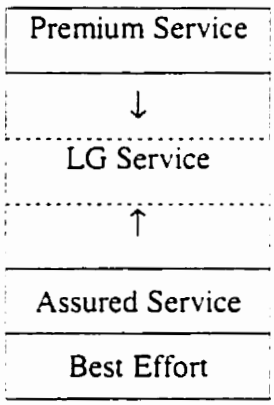


Figure 3.6 Dynamic Bandwidth Allocation for LG Service

BBs are responsible for managing the bandwidth sharing between the LG service and all other services. Figure 3.6 shows one possible way that the dynamic bandwidth allocation for the LG service can be conducted. A maximum amount of bandwidth is designated to the LG service. The LG service tries to yield the bandwidth to Assured service or Premium service as long as the desired loss bound is not violated.

Chapter 4

Performance Evaluation

In this chapter, we study the performance of our admission control scheme designed for the LG service. Section 4.1 describes the simulation model adopted in this study, which includes the network model and the traffic model, followed by a definition of the experimental settings used in the simulation. Simulation results are presented and discussed in Section 4.2. Finally, Section 4.3 gives a summary of the results obtained in this chapter.

4.1 Simulation Model

This section describes the simulation model, which includes the network model and the traffic model, and the experimental setting used in our simulation.

4.1.1 Network Model

To achieve end-to-end delivery in the LG service, the proposed admission control algorithm can be deployed on all routers along the data path. Alternatively, the admission control algorithm can be deployed only at the bottleneck links. In our simulation, the two-node topology that represents the basic unit of the service has been simulated to study our measurement-based admission control scheme designed for the LG service. Figure 4.1 shows the adopted two-node topology. Routers A and B are connected through a 10Mbps link. Sources are connected to Router A through links of infinite bandwidth. Sources have the same destination that is connected to Router B through links of infinite bandwidth. Traffic from all sources is assumed to belong to a single traffic class requesting the LG service. A local domain Bandwidth Broker (BB) is used for resource management. The function of the BB has been described in Chapter 3.

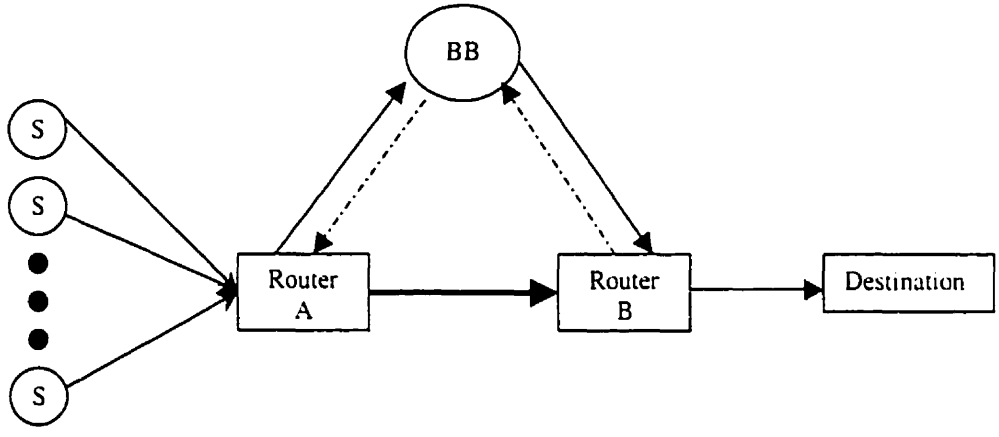


Figure 4.1 Two-node Topology

4.1.2 Traffic Model

The traffic model used in our simulation is the ON/OFF model with exponentially distributed ON and OFF times. Because network traffic generated from many applications including real-time and multimedia applications, can be characterized by the ON/OFF model, it is widely used to simulate network traffic. During each ON period, an exponentially distributed random number of packets with average burst length *BL*, are generated at fixed rate *p* packet/sec. We assume the packet size is 1kb in our simulation. Let the average of the exponentially distributed OFF time be *Idle* millisecond. The average packet generation rate *a* is given by

$$\frac{1}{a} = \frac{Idle}{BL} + \frac{1}{p} \tag{4.1}$$

Each individual flow is generated according to the ON/OFF model. The bursts of flows are independent of each other. The burstiness of the aggregate traffic is affected by several factors, such as the characteristics of the individual flows, the number of flows. The call holding times (also called flow duration) and flow interarrival times also have some effects on the traffic aggregation. In our simulation, these two values are also exponentially distributed.

The traffic generated by the host conforms to a particular token bucket filter as shown in Figure 4.2. Packets are queued when the token bucket is empty until more tokens are available. Applications requesting the LG service are supposed to have such mechanism to regulate their packet flows to conform to its token bucket specification. A source

generated by the ON/OFF model will be shaped, i.e., becomes less bursty, after passing through such token bucket filter, as the burst sizes are limited by the token depth and the token generation rate. Packets from a burst are queued and forced to be transferred at the token rate when the bucket is empty.

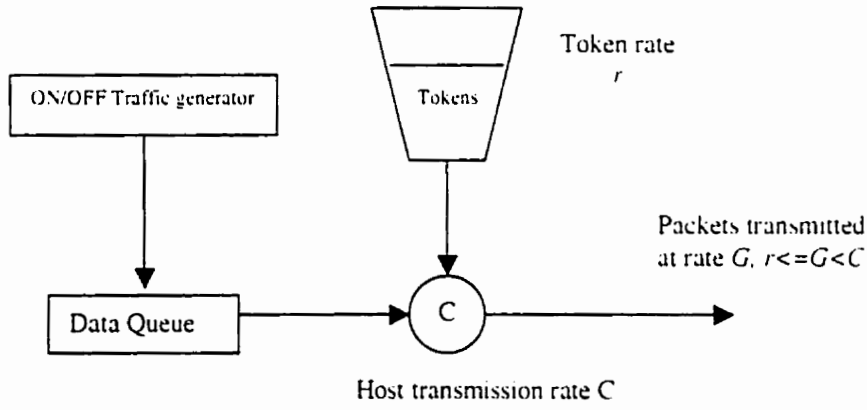


Figure 4.2 ON/OFF Traffic Model with Token-bucket Filter

4.1.3 Experimental Setting

According to the source model in our simulation, during the ON periods, sources generate *BL* packets on average at a peak rate *p*. From Equation 4.1 we see that the average OFF time, *Idle*, is given by

$$Idle = (\frac{1}{a} - \frac{1}{p}) \times BL \qquad (4.2)$$

The values of *a*, *p*, and *BL* are inputs in our simulation model. The complete test results showing the effects of these three factors on the performance of the admission control scheme are provided.

At each host, The generated ON/OFF traffic will pass through a token bucket filter, which forces source traffic to conform to its token bucket specification. The token bucket filter tends to attenuate the burstiness of the traffic. This is because the maximum burst size is limited by

$$BL_{\max} = \frac{b}{p-r} \times p \tag{4.3}$$

where $\frac{b}{p-r}$ is the time to exhaust b tokens. The number of tokens available for a burst is determined by the token rate and the duration of source's last OFF time. When the token rate is not fast enough or the OFF time is not long enough to allow the token bucket to fully refill before the next ON time, the size of the next burst is further limited. A tight token bucket setting characterizes a relatively small bucket depth and slow token rate in accordance to the original source characteristics. On the other hand, a loose token bucket setting characterizes a relatively large bucket depth and fast token rate in accordance to the original source characteristics. A tight token bucket filter will significantly shape the traffic source. A loose token bucket filter may result in unshaped traffic. A source described by such token bucket parameters will request more resources from the network than it actually needs. This causes either increasing the probability of call blocking due to the unavailability of the network resources or excessive unnecessary resources being reserved during flow admission. Both result in reducing the link utilization. In our simulation, we would like to set the token bucket parameters properly, that is not too tight

that could significantly shape the source traffic and not too loose to result in low link utilization.

In our simulation, the token depth b is set to contain enough tokens so that the source can deliver $2BL$ packets at peak rate, p during the burst time without queueing any packets. The ON time is given by $2BL/p$. Thus b can be calculated by

$$b = \frac{2BL}{p} \times (p - r) \tag{4.4}$$

For an extremely bursty source, the token depth b is close to the maximum value $2BL$.

We set the token rate, r, to a value that allows an empty bucket to be refilled within half of the average idle period, thus the token rate can be calculated by

$$r = \frac{b}{Idle/2} \tag{4.5}$$

from Equation 4.3 and Equation 4.4 we can get

$$r = \frac{2BL \times p}{p \times Idle/2 + 2BL} \tag{4.6}$$

and

$$b = \frac{BL \times p \times Idle}{p \times Idle/2 + 2BL} \tag{4.7}$$

Because the token bucket parameters calculated from Equations 4.6 and 4.7 can allow a source burst with a burst size twice as large as its average burst size to pass through without queueing any packet and the token bucket is refilled at a rate twice as fast as it

needs on average, the source will basically maintain the original burst characteristics as generated from the ON/OFF model after passing through such token bucket filter. Therefore, and because the token bucket specification closely describes the source characteristics, the source requests the same amount of the resources from the networks as it actually needs. This makes the admission control and resource reservation more precise and efficient.

The simulations used in this thesis are discrete event driven simulations that are developed in Java. The simulator consists of three main parts:

a) Traffic generator: Responsible for generating traffic patterns with specific parameter settings. It includes two components:
   - Flow generator: Generates independent packet flows characterized by the ON/OFF model and passes them through the token bucket filters.
   - Traffic manager: Schedules the exponentially distributed arrivals of flows. Assigns the value for the traffic model settings and the token bucket filter parameters to achieve the desirable traffic aggregation.

b) Network module: Simulates Layer 3 functions of the DiffServ architecture. It is composed of three components:
   - Node: Responsible for receiving packets and putting them in respective output buffers, conducting the measurement process and cooperating with BBs during admission control.
   - Link: Delivers packets according to its assigned bit rate.

- BB: responsible for the signaling process among source, node and itself, as well as handling admission control requests.

c) Event Handler: Maintains a time-ordered schedule of events in the simulation.

We repeat each simulation experiment five times to achieve a 90% confidence level with 10% confidence intervals (see Appendix A). Each simulation is run for 2000 simulated seconds. The data presented are obtained from the later half of each simulation. By visual inspection, we determined that 1000 simulated seconds are sufficient for the simulation to reach a stable state.

The flow interarrival times are exponentially distributed with an average of 300 milliseconds. We set the average holding time of the sources to 300 seconds. The target utilization υ is set to 95%. The loss bound of the service is set to 0.5%.

4.2 Discussion of Results

A number of simulation experiments were conducted in order to study the effect of the peak rate, average rate and burst size on the performance of the admission control scheme designed for the LG service. We also varied the measurement window and buffer size to study their performance effects under different scenarios.

4.2.1 Effect of Peak Rate

Figures 4.3 and 4.4 plot the link utilization vs. the peak source rate for an output buffer size of 128kb and 256kb, respectively. Different values of average source rate a (50kb/s,

100kb/s and 200kb/s), burst size (10kb and 30kb) and measurement time window (30Mb and 60Mb) were used.

We can see that under a same average rate, the utilization decreases with the increase of the peak rate. This can be explained by noting the source burstiness factor (BF), calculated as the source peak rate divided by source average rate (p/a). BF is an index of the burstiness level of a source. For the same average traffic rate, BF increases with the increase of the peak rate. The greater the BF of individual sources, the more bursty the aggregate traffic tends to be. We can also use Equation 4.2 to explain this phenomenon. When the burst size BL is constant, a decrease in the average rate or an increase in the peak rate will result in an increase of source's idle time. Long idle time of sources results in uneven distribution of the aggregate traffic, which makes the aggregate traffic bursty. With the increase of the traffic burstiness level, and in order to guarantee the loss bound, the admission control scheme will further limit the number of flows admitted into the network, which results in a decrease in the utilization.

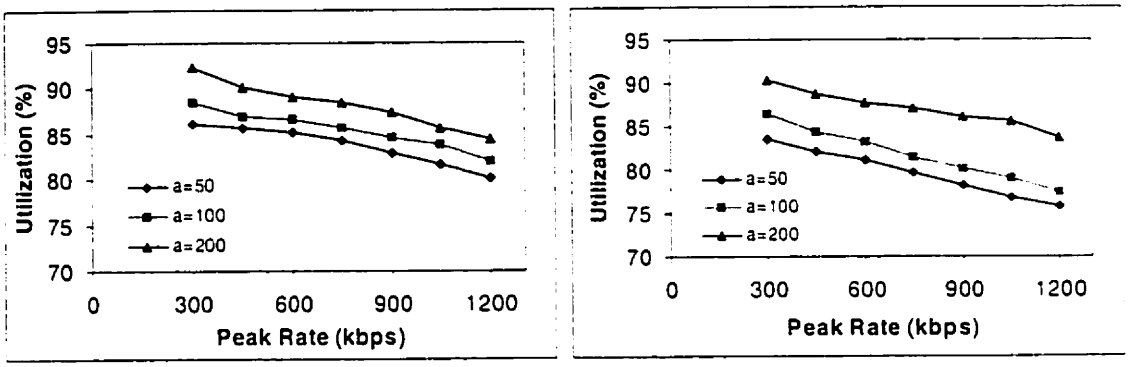
For sources with the same average rate, BF increases with the increase of the peak rate, resulting in lower utilization. Likewise, as we can see from the plots, given the same peak rate, a higher average rate of source traffic results in higher utilization, since for a given same peak rate, a higher average rate results in a smaller BF value. From Figures 4.3 and 4.4, we can see that a smaller burst size of source traffic (10kb) results in a higher utilization. This is because sources with a smaller burst size can achieve a higher

multiplexing gain than sources with larger burst size. The effect of the burst size will be further studied in the section 4.2.3.

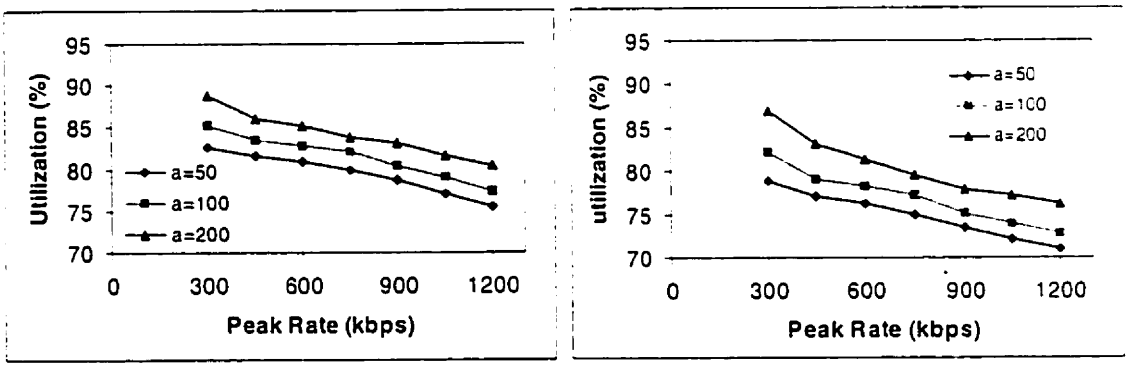
The results also show that a large measurement window results in a relatively low utilization level. This is because increasing T makes the measurement process more conservative by sampling the queue lengths and utilization for a longer history, which in turn makes the admission control algorithm itself more conservative. Meanwhile, a large T depresses the utilization by keeping the artificially increased measured values for a long period.

When a larger buffer size (256kb) is assigned (see Figure 4.4), the utilization increases. This is because a larger buffer can absorb more bursts from the traffic flows. Consequently, the admission control scheme can admit more flows into the network. The increase in utilization is more significant for the bursty traffic than for the smooth traffic (see Figures 4.3 and 4.4).

Burst size = 10 kb **Burst size = 30 kb**



(a) **Measurement window = 30Mb** (b)

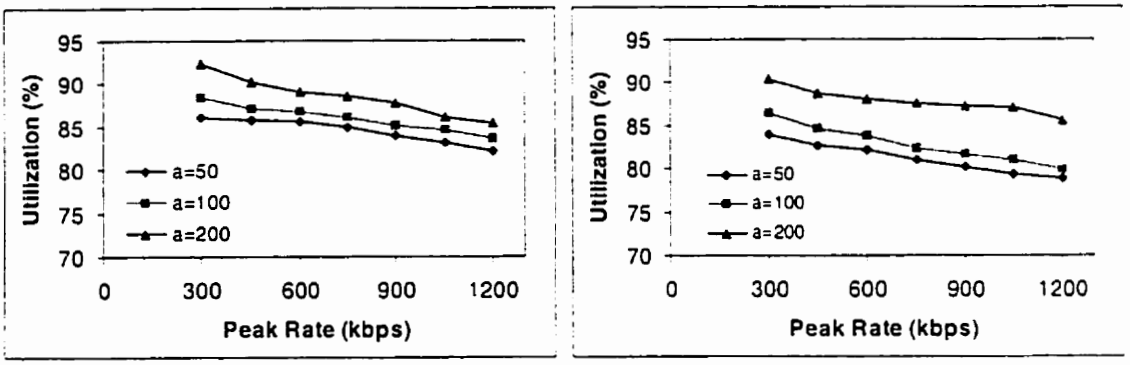


(c) **Measurement window = 60Mb** (d)

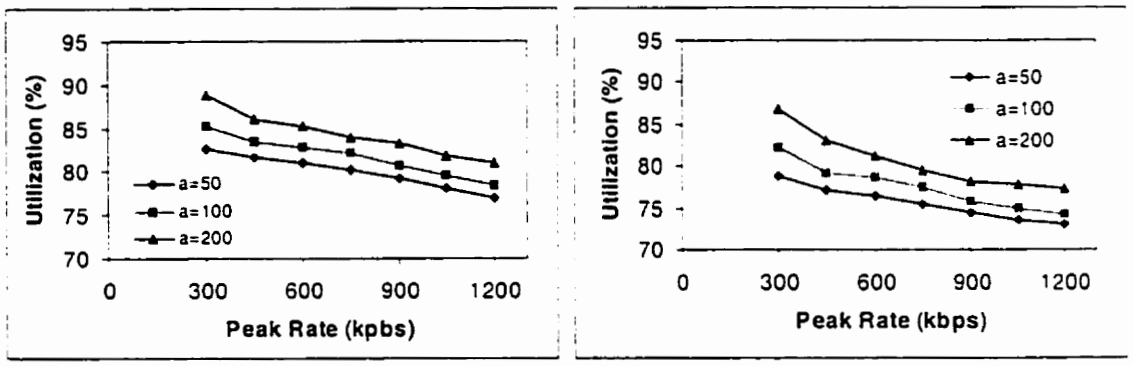
a **: average rate**

Figure 4.3 Peak rate vs. utilization (buffer size = 128kb)

Burst size = 10 kb **Burst size = 30 kb**



(a) **Measurement window = 30Mb** (b)



(c) **Measurement window = 60Mb** (d)

***a* : average rate**

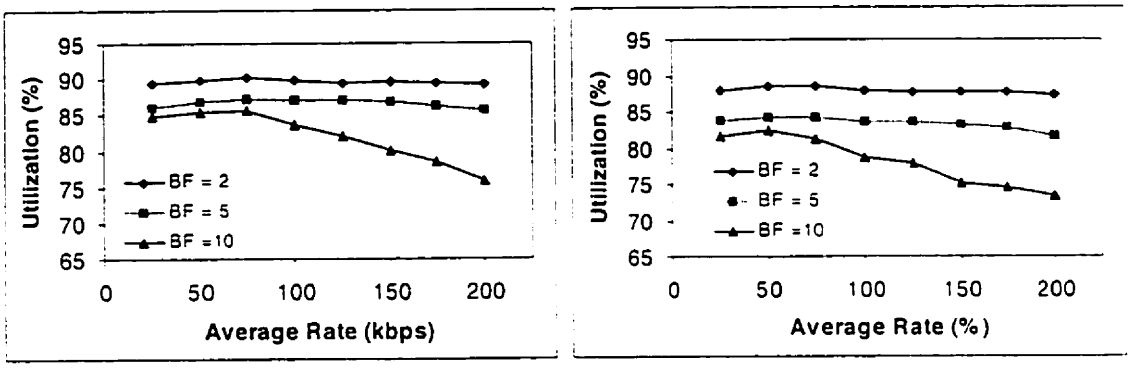
Figure 4.4 Peak rate vs. utilization (buffer size = 256kb)

4.2.2 Effect of Average Rate

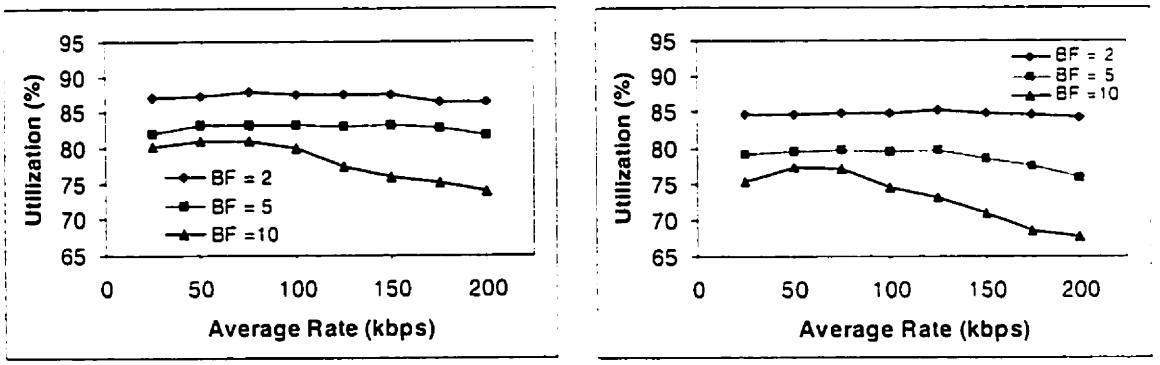
Figures 4.5 and 4.6 plot the link utilization vs. the average source rate for an output buffer size of 128kb and 256kb, respectively. Different values of burstiness factor (2, 5,10), burst size (10kb and 30kb) and measurement time window (30Mb and 60Mb) were used.

We can see that increasing the average rate initially results in an increase of the utilization. Then the utilization decreases when the average rate is further increased. From Equation 4.2 we can see that when BF is fixed, increasing the average rate causes a decrease of the source idle time. This explains the initial increase in utilization. However, further increasing of the average rate, and because the bandwidth assigned for the service is fixed in our simulation, the number of flows that can be admitted decreases. The flow multiplexing effect drops with the lower number of co-existing flows, which leads to the decrease in utilization. For higher values of BF, the second effect has a more significant impact and utilization drops faster as the average rate increases.

Burst size = 10 kb **Burst size = 30 kb**



(a) **Measurement window = 30Mb** (b)

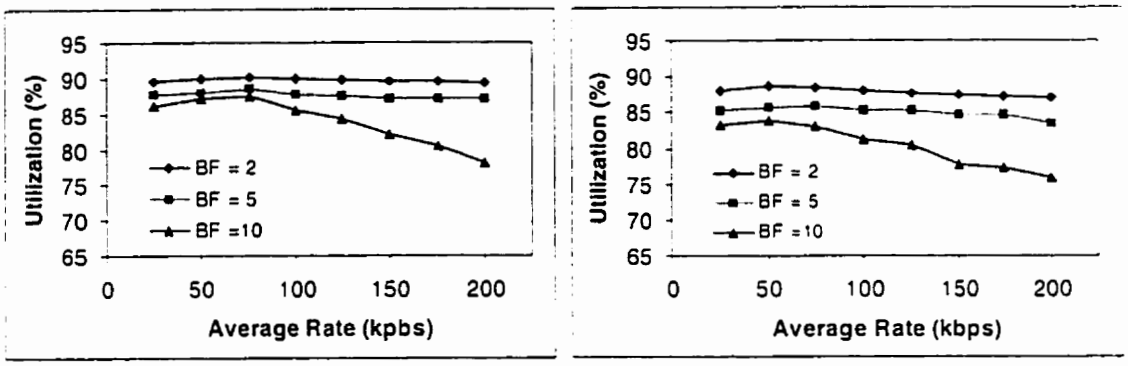


(c) **Measurement window = 60Mb** (d)

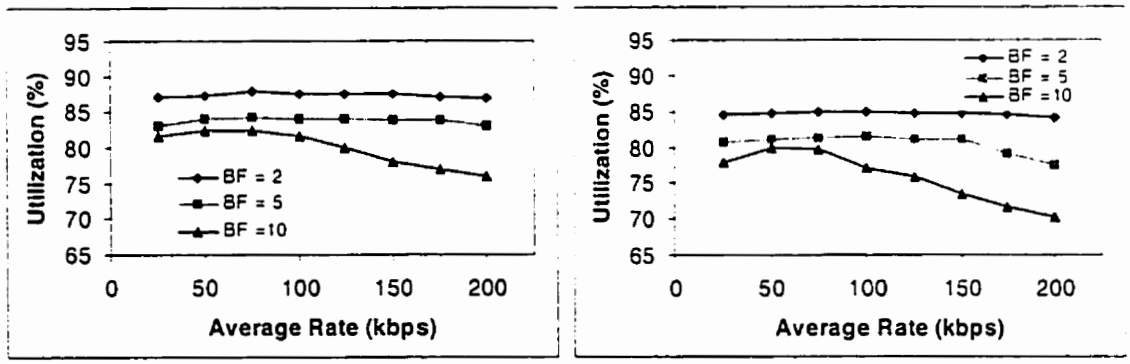
BF: Burstiness Factor

Figure 4.5 Average rate vs. utilization (buffer size = 128kb)

Burst size = 10 kb **Burst size = 30 kb**



(a) **Measurement window = 30Mb** (b)



(c) **Measurement window = 60Mb** (d)

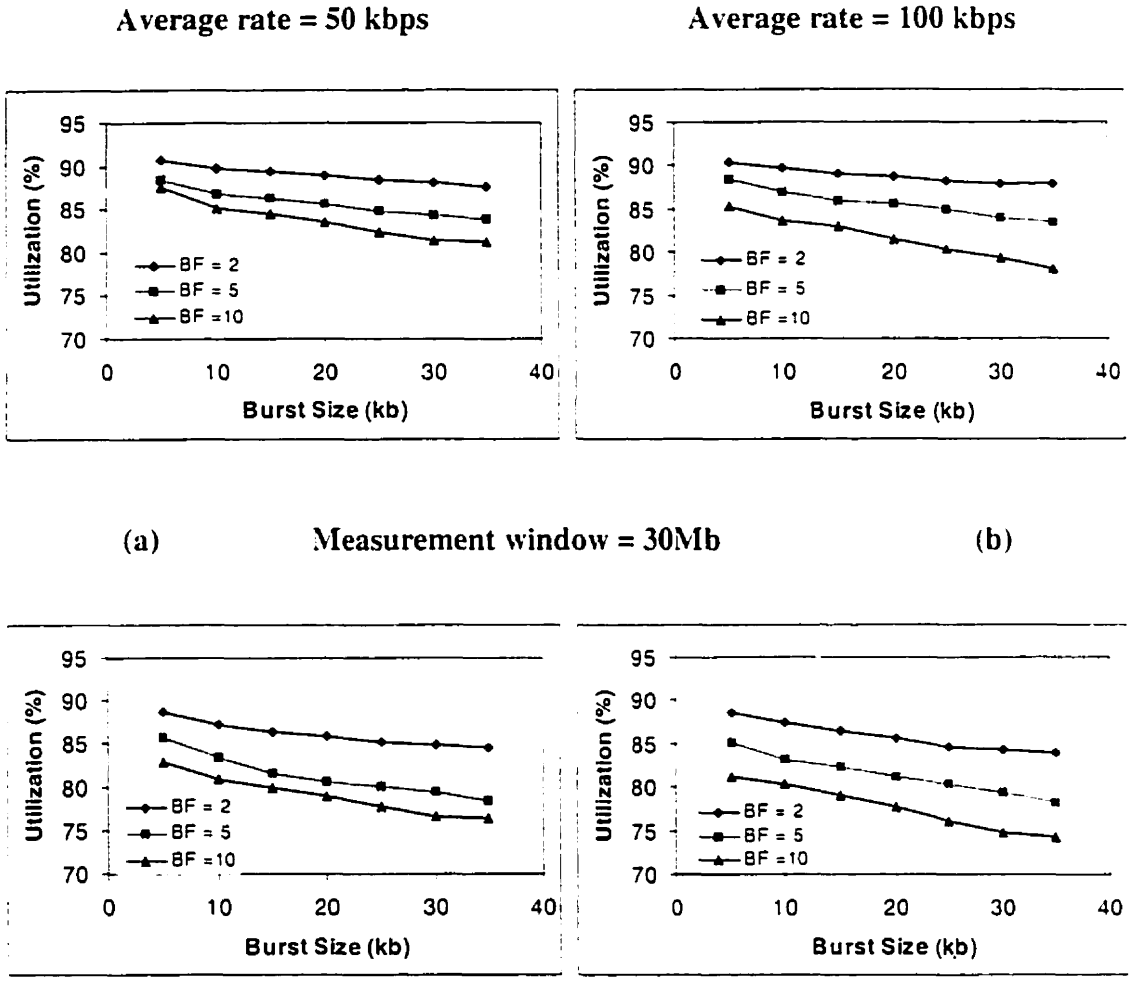
BF: Burstiness Factor

Figure 4.6 Average rate vs. utilization (buffer size = 256kb)

4.2.3 Effect of Burst Size

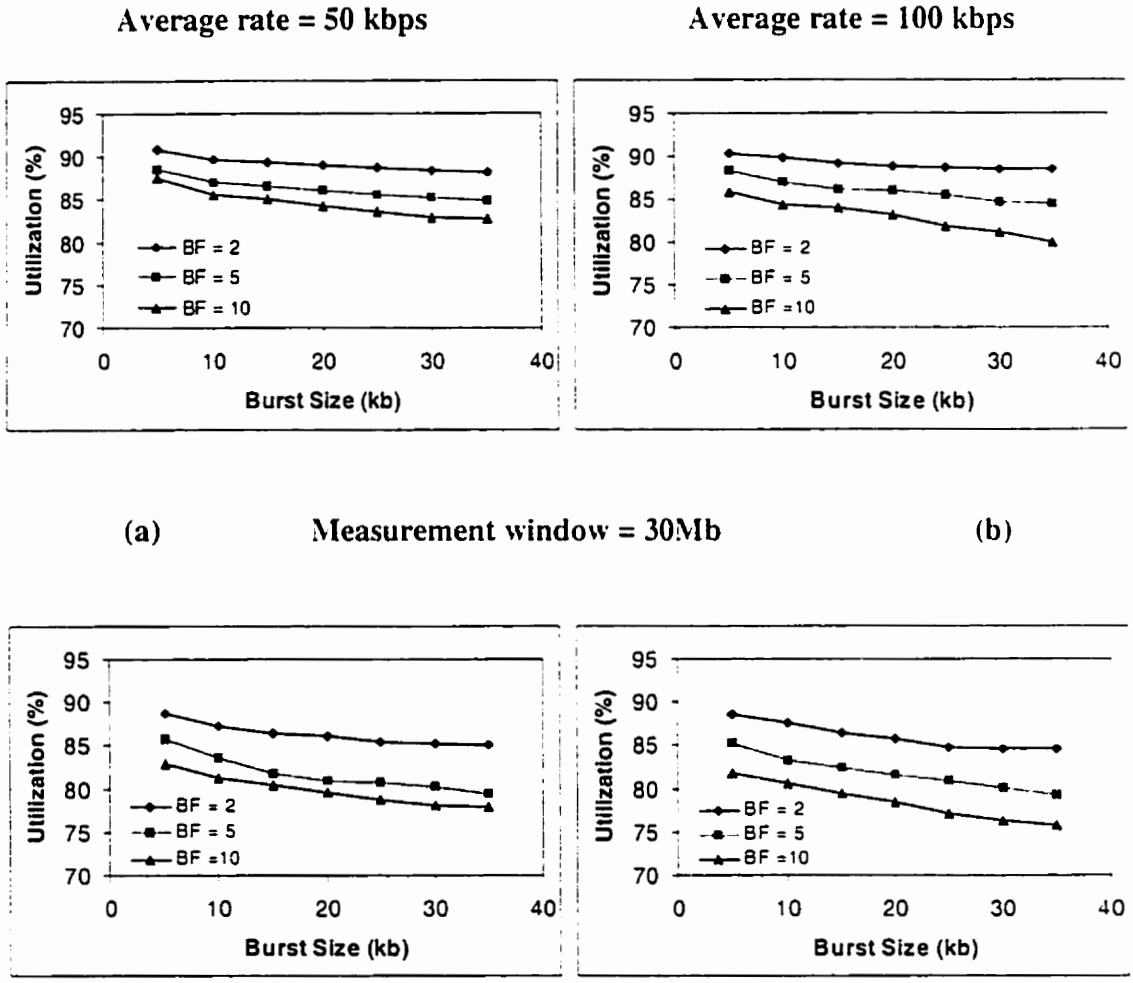
Figures 4.7 and 4.8 plot the link utilization vs. the burst size for an output buffer size of 128kb and 256 kb, respectively. Different values of burstiness factor (2, 5, 10), average source rate (50kb and 100 kb) and measurement time window (30Mb and 60Mb) were used.

The results show that the utilization decreases with increasing the burst size. This is because large bursts are harder to smooth out through natural multiplexing than small bursts. We can also use Equation 4.2 to explain this effect. When the average rate, a, and peak rate, p, are fixed, the source idle time *Idle* increases proportionally with the burst size *BL*. This causes lower utilization.



BF: Burstiness Factor

Figure 4.7 Burst size vs. utilization (buffer size = 128kb)



BF: Burstiness Factor

Figure 4.8 Burst size vs. utilization (buffer size = 256kb)

4.2.4 Effect of Buffer Size

The experiments in this section study the effect of buffer size on the system performance. Unlike the experiments in previous sections, the traffic sources used here exhibit variations in their average and peak rates. The characteristics of the traffic used are described in Table 4.1. The range of average rate and burstiness factor characterize a traffic type. The average value of BF for each traffic type is also shown in Table 4.1. Traffic type A can be characterized as light load, bursty traffic, traffic B can be characterized as medium load, medium bursty traffic and traffic C can be characterized as heavy load, smooth traffic.

Traffic Name	Range of Average Rate *a*	BF
A	*a* in [15kbps, 30kbps]	10
B	*a* in [100kbps, 150kbps]	5
C	*a* in [500kbps, 750kbps]	2

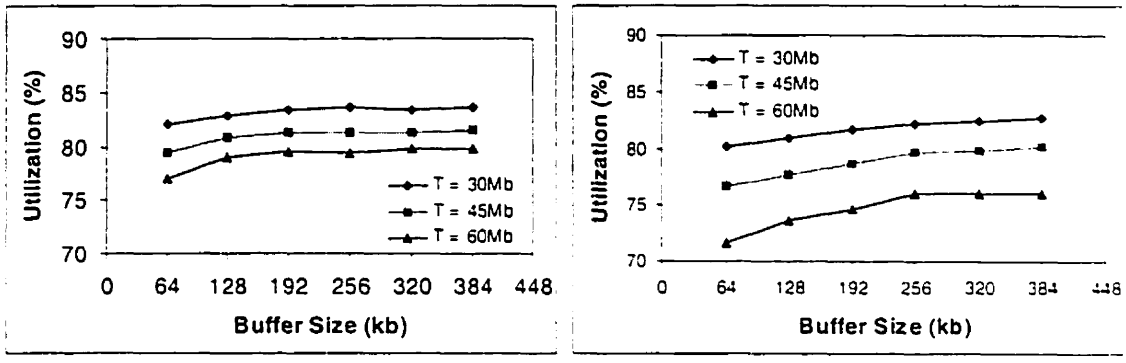
Table 4.1 Traffic Characteristics

Three settings of measurement window (T = 30Mb, 45Mb and 60Mb) are used in the experiments (see Figure 4.9). For traffic type A, which is bursty, the utilization increases quickly with the increase of the buffer size. For traffic type B, which is less bursty, the utilization increases slowly with the increase of the buffer size. There is no significant increase in utilization observed for traffic type C while larger buffers are assigned. This is because bursty traffic has more tendency to build up a longer queue than smooth traffic at the output buffer. As we mentioned before, the buffer size and observed queue length are

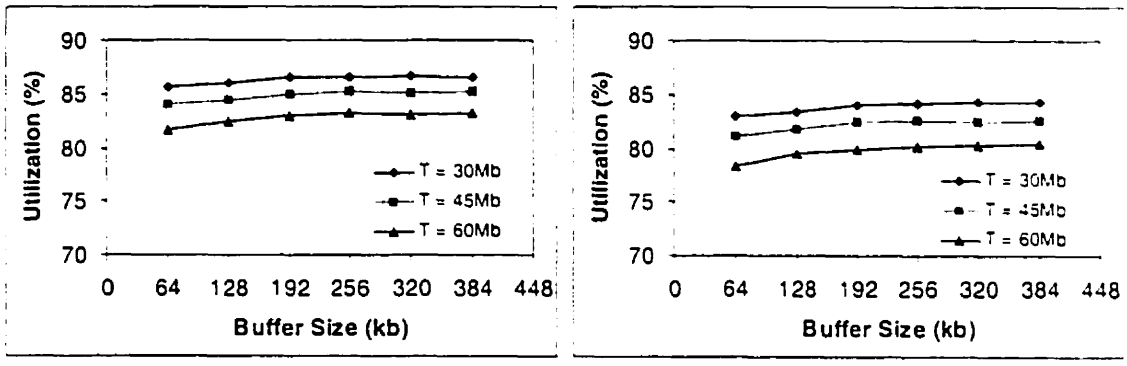
used as inputs in our admission control scheme to estimate the loss when considering a flow's request. When the traffic is bursty, increasing the buffer size will significantly increase the number of admitted flows, which in turn causes an increase in utilization. When the traffic is smooth, a relatively small queue could be built up at the output buffer. Thus the increase of the buffer size will only have little effect on the number of admitted flows.

Again, the large measurement window T leads to a conservative admission control, which results in a low utilization level. The results also give us an idea of how significantly the choice of T affects the system performance. For example, for Traffic A with burst size of 30kb, when a small T (30Mb) is used, the system can achieve a 9% increase in utilization than with a large T (60Mb).

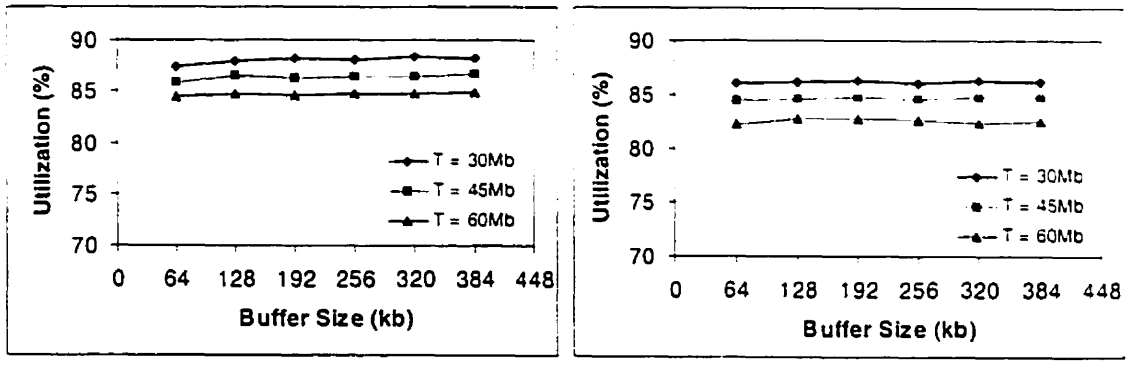
Burst size = 10 kb **Burst size = 30 kb**



(a) Traffic A, *a* in [15kbps – 30kbps] BF= 10 (b)



(c) Traffic B, *a* in [100kbps – 150kbps] BF= 5 (d)



(e) Traffic C, *a* in [500kbps – 750kbps] BF = 2 (f)

Figure 4.9 Buffer size vs. utilization

4.2.5 Comparison with Premium Service

Table 4.2 compares the results of the LG service to an equivalent Premium service counterpart. The results for Premium service are calculated by assuming the peak rate is guaranteed. The results for the LG service are collected through the experiments with buffer size 128kb and source burst size 10kb. We can see that the LG service consistently allows the network to achieve a higher utilization level than Premium service. The utilization gain is not as significant when source traffic is smooth. For instance, Traffic C has a peak rate only twice as its average rate (BF=2). Consequently, the results only show an increase in utilization from 50% to 88%. In contrast, for bursty sources, the gain of using the LG service is more apparent as it achieves a much higher utilization compared to that achievable under Premium service. Traffic A, for example, is very bursty. Under Premium service, only 44 flows on average can be admitted resulting in a low utilization of 10%. Under the LG service, 368 flows are served on the average, resulting in an actual utilization of 83%.

Traffic Name	Premium Service		Loss Guaranteed Service		
	Util	Active flows (N)	Util	Active flows (N)	Loss Rate
Traffic A	10%	44	83%	368	3.0×10^{-3}
Traffic B	20%	16	86%	69	1.6×10^{-3}
Traffic C	50%	8	88%	14	1.0×10^{-4}

Table 4.2 Premium Service vs. Loss Guaranteed Service

Figure 4.10 shows the utilization level for the LG service and Premium service for traffic with different burstiness factors. We can see that with the increase of the burstiness factor, the utilization level of both type of services decrease consistently. However, the utilization level for Premium service drops at a much higher rate than that of the LG service. For example, when BF reaches 7, under the LG service, the system can still achieve a high utilization of 80%, while under Premium service, the utilization drops to as little as 14%.

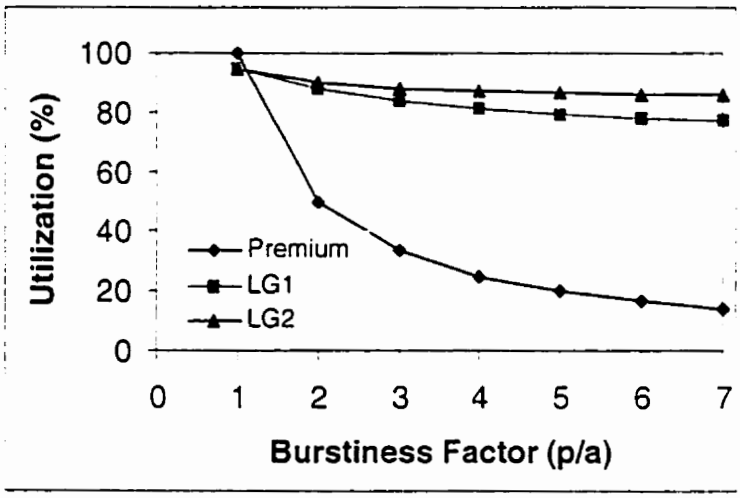
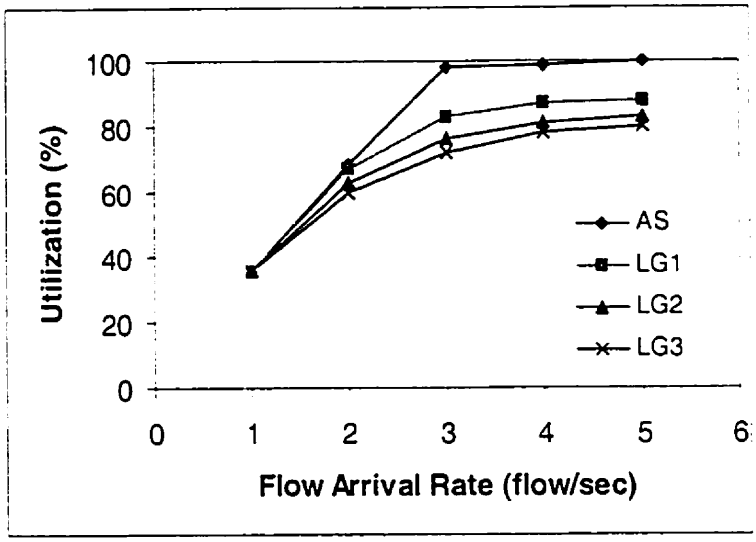


Figure 4.10 Utilization of LG Service vs. Premium Service (Average rate = 50kbps, burst size = 15kb, LG1: buffer size = 64kb, LG2: buffer size = 128kb)

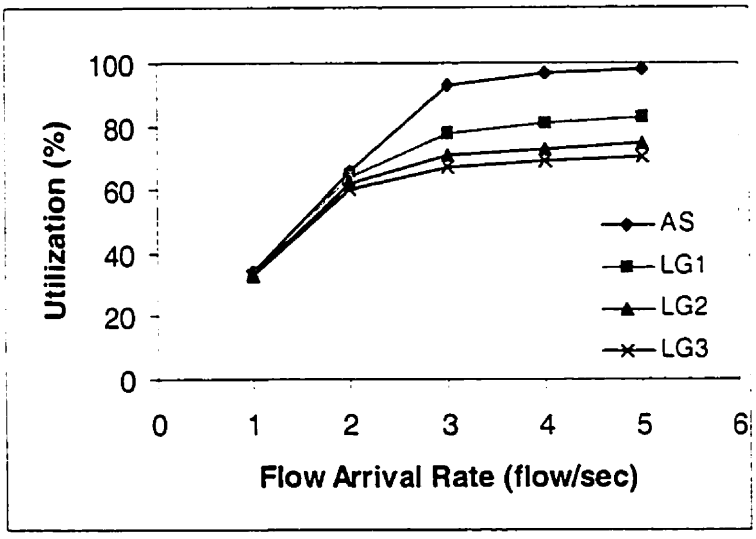
4.2.6 Comparison with Assured Service

The experiments in this section compare the results of LG service vs. Assured service in terms of utilization and loss rate. To simplify our experiments, we compared the LG service with the highest of the four classes of Assured service and assumed that all the Assured packets are in-profile. We also assumed that this class of Assured service has the same bandwidth and buffer assignment as the LG service. Figure 4.11 plots the utilization of Assured service and LG service with a measurement window of 10Mb, 20Mb and 30Mb. We use sources with an average rate of 50kbps and burst size of 15kb in the experiments. Figure 4.11(a) shows the results when a smooth traffic (BF=2) is used, while Figure 4.11(b) shows the results when a bursty traffic (BF=10) is used. In the experiments, we increased the traffic load by increasing the flow arrival rate.

The plots show that the utilization level of Assured service increases almost linearly with the increase of the flow arrival rate until it approaches full utilization. Under the LG service, when the traffic load is light, the utilization also increases with the increase of the flow arrival rate. When the traffic load approaches link capacity, the utilization increase is slower and stabilizes at a certain threshold point. This is because when the traffic load is light, a faster flow arrival rate leads to a larger number of admitted flows, which results in a higher utilization. With the increase of the utilization, the admission control scheme tends to reject more flow requests and will reject all the extra traffic requests when the utilization reaches a certain threshold. For bursty traffic (Figure 4.11(b)), the utilization level achievable under the LG service is lower.



(a) BF = 2



(b) BF = 10

Figure 4.11 Utilization vs. arrival rate of LG Service and Assured Service (LG1: window size = 10Mb, LG2: window size = 20Mb, LG3: window size = 30Mb. average rate = 50kbps, burst size = 15kb)

The measured packet loss results for the experiment in Figure 4.11 are shown in Table 4.3. We can see that when the traffic is light and smooth, Assured service provides results which are as good as the LG service. When the traffic is heavy and/or bursty, the lack of admission control in the Assured service model results in high loss rate. Under the LG service, the loss rate is controlled within the target loss rate. For instance, for BF=10 and an arrival rate of 5 flows per second, 57% of packets are dropped under Assured service as opposed to less than 0.2% for the Loss Guaranteed service. A larger measurement window for the LG service makes the admission control more conservative, resulting in a lower loss rate.

Service Name	Arrival Rate (flow/sec)				
	1	2	3	4	5
AS	0	0	5.8×10^{-2}	3.4×10^{-1}	5.5×10^{-1}
LG1 (T=10Mb)	0	0	1.9×10^{-7}	6.1×10^{-5}	2.1×10^{-4}
LG2 (T=20Mb)	0	0	0	4.3×10^{-6}	1.7×10^{-5}
LG3 (T=30Mb)	0	0	0	9.8×10^{-7}	5.5×10^{-6}

(a) BF = 2

Service Name	Arrival Rate (flow/sec)				
	1	2	3	4	5
AS	0	5.7×10^{-5}	7.4×10^{-2}	3.8×10^{-1}	5.7×10^{-1}
LG1 (T=10Mb)	0	0	1.8×10^{-4}	7.6×10^{-4}	2.3×10^{-3}
LG2 (T=20Mb)	0	0	4.9×10^{-6}	3.3×10^{-5}	8.9×10^{-5}
LG3 (T=30Mb)	0	0	2.5×10^{-7}	1.2×10^{-5}	1.6×10^{-5}

(b) BF = 10

Table 4.3 Loss Rate of Loss Guaranteed Service vs. Assured Service

4.3 Summary

This chapter evaluated the performance of our measurement-based admission control scheme designed for the LG service, through a comprehensive simulation model. The effects of the source peak rate, average rate and burst size on the system performance were investigated. Experiments comparing the LG service with Premium service and Assured service were also conducted. It was shown that:

- With proper parameter setting of admission control tuning knobs, the LG service can achieve a high utilization level while still guaranteeing the loss bound.
- The multiplexing gain of traffic bursts is affected by the burst size, burstiness factor and the number of flows involved. A high multiplexing gain allows the LG service to achieve a high utilization level.
- Higher utilization levels can be achieved through applying a smaller measurement window T as long as the loss bound is not violated.
- Sources with a high burstiness factor (BF) result in a low utilization level both under the LG service and Premium service. However, by taking advantage of the multiplexing effect of traffic bursts, given the same BF, the LG service can achieve a higher utilization level than Premium service. This utilization gain is more significant when the BF is higher.

Chapter 5

Conclusion

5.1 Concluding Remarks

With the transformation of the Internet into a commercial infrastructure, customers expect differentiated QoS in terms of performance, throughput and/or latency. To meet such requirements, IntServ and DiffServ have been introduced to replace the current simple one-service model provided by the Internet. The IntServ model can provide users with an end-to-end QoS guarantee through resource reservation and admission control mechanisms along the data path. However, as the IntServ model uses per-flow signaling and per-flow traffic management, it introduces significant overhead and scalability problems. The DiffServ approach alleviates these problems by providing QoS based on flow aggregates. Service discrimination is provided through differentiation among service classes. Thus it is scalable and easy to implement.

Two services, Premium and Assured have been introduced in the DiffServ model, each intended for certain applications with different QoS requirements, but each having its own limitations. Premium service guarantees the peak rate of user flows, but results in a low bandwidth utilization. Assured service provides the assurance that high priority packets can always gain a relatively higher throughput than low priority packets during periods of network congestion. But Assured service cannot provide any quantitative QoS guarantee to the traffic flows.

The Loss Guaranteed (LG) service we proposed for the DiffServ architecture can provide a quantitative QoS guarantee in terms of loss rate without per-flow based resource reservation. A signaling protocol which is in conformance with the DiffServ model, along with a measurement-based admission control are designed to implement the LG service. Because this service does not allocate resources according to the peak traffic requirement of each flow, it can achieve a much higher utilization level than Premium service and at the same time, provides a loss bound to the flows requesting the service.

An extensive simulation model has been developed to study the performance and viability of the LG service. We have tested a variety of traffic conditions and admission control parameter settings in our simulation. The results show that with proper settings of admission control parameters, the LG service can achieve a high level of utilization while still reliably keeping packet loss within the desired target. Experiments comparing the LG service with Premium service and Assured service have also been conducted. Comparing

Comparing with Premium service, the results show that a significant utilization gain can be achieved under the LG service. Comparing with Assured service, the results show the LG service can guarantee a loss bound even during congestion periods.

In conclusion, the LG service is able to provide packet-loss guarantees within the DiffServ architecture without the need for explicit resource reservation.

5.2 Future Work

This thesis proposed a signaling protocol and admission control scheme which are necessary for implementation of the LG service. Our simulation results show that the loss bound can be provided and a high utilization level can be achieved by this service. However, several aspects still need further investigation.

The performance of the admission control scheme we proposed is highly dependent on the values of the performance tuning knobs discussed in Section 3.4.4. In a short time scale, we can assume the traffic is stable in terms of load and burst characteristics. However, the traffic situation fluctuates in a larger time scale. Moreover, if a dynamic bandwidth allocation scheme for the LG service is used, the allocated bandwidth and buffer space can also change. An ideal admission control scheme for the LG service should have a mechanism that automatically tunes the admission control settings according to changes in traffic conditions. For instance, the adaptive-window measurement mechanism introduced in Section 2.3.2.2 can be applied to the LG service.

The adaptive-window measurement mechanism automatically adapts to traffic fluctuations by adjusting the measurement window size. In such mechanism, a first-level algorithm is used to adjust the window size according to the current link load. Whenever the link load is lower than a trigger value, the algorithm continuously shrinks the measurement window size, resulting a less conservative measurement, until the link load reaches a trigger value. Afterwards it starts to enlarge to measurement window size until the link load drops below the trigger value. The trigger value is selected by a second-level algorithm, which takes some measured values as inputs. In our case, the short term and long term loss violations can be used as inputs to adjust the trigger value.

We proposed a possible dynamic resource allocation scheme for the LG service in this thesis. Under such scheme, the LG service can allow Premium traffic and Assured traffic to share part of its bandwidth when its traffic is light. This allows efficient usage of the bandwidth among different services within the DiffServ architecture. The increase or decrease of the resource allocation for the LG service should also be based on real-time measurement. The decrease of the bandwidth allocation should be done in a more conservative way than the increase of the bandwidth allocation, to ensure that decrease of the bandwidth allocation does not cause a short term loss bound violation. A criterion used to evaluate such a scheme should be based on how efficiently it manages the bandwidth sharing among the LG service and other services in terms of the overall link utilization, while the loss bound is still reliably provided.

Bibliography

[1] R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: and Overview", *RFC 1633*, Jun. 1994, *http://ietf.org/rfc/rfc1633.txt*.

[2] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", *RFC 2205*, Sept. 1997, *http://ietf.org/rfc/rfc2205.txt*

[3] Mankin et al., "RSVP Version 1: Applicability Statement, Some Guidelines on Deployment", *RFC 2208*, Sept. 1997, *http://ietf.org/rfc/rfc2208.txt*.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, " An Architecture for Differentiated Services", *RFC 2475*, December 1998, *http://ietf.org/rfc/rfc2475.txt*.

[5] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers" *RFC 2474*, Dec. 1998, *http://ietf.org/rfc/rfc2474.txt*.

[6] K. Nichols, V. Jacobson and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", *Internet Draft, <draft-nichols-diff-svc-arch-00.txt>*, Nov. 1997.

[7] M. Hou and H. Mouftah, "Performance Evaluation of Premium Service", *Internet draft <draft-hou-Diffserv-premium-eval-00.txt>*, June 1999.

[8] V. Jacobson, K. Nichols and K. Poduri, "An Expedited Forwarding PHB" *RFC2598*, June 1999, *http://ietf.org/rfc/rfc2598.txt.*

[9] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assured Forwarding PHB Group", *Internet Draft, <draft-ietf-DiffServ-af-03.txt>*, June 1999.

[10] D. Clark, J. Wroclawski, "An approach to Service Allocation in the Internet", *Internet Draft, <draft-clark-diff-svc-alloc-00.txt>*, Aug. 1997.

[11] J. F. Rezende, "Assured Service Evaluation", *Proceedings of the IEEE Global Telecommunications Conference - GLOBECOM'99, pp. 100-104*, Rio De Janero, December 1999.

[12] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance" *IEEE/ACM Transactions on Networking, vol.1, no.4, pp. 397-413.* August 1993.

[13] S. Shenker and J. Wroclawski, "Network Element Service Specification Template" *RFC2216*, Sept. 1997, *http://ietf.org/rfc/rfc2216.txt*.

[14] S. Shenker, C. Partridge and R. Guerin, "Specification of Guaranteed Quality of Service", *RFC 2212*, Sept. 1997, *http://ietf.org/rfc/rfc2212.txt*.

[15] J. Wroclawski, "Specification of the Controlled-Load Network Element Service", *RFC2211*, Sept. 1997, *http://ietf.org/rfc/rfc2211.txt*.

[16] S. Brim, B. Carpenter and F. Le Faucheur, May 2000 "Per Hop Behavior Identification Codes", *RFC2836*, May 2000, *http://ietf.org/rfc/rfc2836.txt*.

[17] J. Postel, "Service Mappings", *RFC 795*, 1981, *http://ietf.org/rfc/rfc795.txt*.

[18] S. Bradner and A. Mankin, "IPng: Internet Protocol Next Generation", *Reading MA: Addison-Wesley*, 1996.

[19] Internet2 QoS Working Group Draft, "Draft Qbone Architecture", May, 1999.

[20] T. Li, "CPE based VPNs using MPLS", *Internet draft*, *<draft-li-MPLS-vpn-00.txt>*, Oct. 1998.

[21] J. Wroclawski, and L. Zhang. "Recommendations on Queue Management and Congestion Avoidance in the Internet." *RFC 2309*, April 1998, *http://ietf.org/rfc/rfc2309.txt*.

[22] J. Ibanez and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service", *Internet Draft*, *<draft-ibanez-diffserv-assured-eval-00.txt>*, August 1998.

[23] M. Goyal, A. Durresi and R. Jain, "Effect of Number of Drop Precedences in Assured Forwarding", *Internet Draft*, *<draft-goyal-dpstdy-diffserv-01.txt>*, Dec. 1999.

[24] R. Yavatkar, D. Pendarakis and R. Guerin, "A Framework for Policy-based Admission Control", *RFC2753* Jan. 2000, *http://ietf.org/rfc/rfc2753.txt*

[25] I. Stoica, H. Zhang and T. S. Eugene Ng, "A hierarchical fair service curve algorithm for link-sharing, real-time, and priority services", *ACM/IEEE Transactions on Networking, vol. no.2, pp.185-199,* 2000.

[26] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang. "A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks (Extended Version)". *ACM/IEEE Transactions on Networking, vol.5, no.1, pp.56-70,* Feb. 1997.

[27] R. Guerin, H. Ahmadi and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High Speed Networks", *IEEE Journal of Selected Areas in Communication, vol.9, no.7, pp.968-981,* Sept. 1991.

[28] S. Floyd. "Comments on Measurement-based Admissions Control for Controlled-Load Service". *Technical Report,* 1996.

[29] C. Casetti, J. Kurouse and D. Towsley, "An Adaptive Algorithm for Measureent-based Admission Control in Integrated Services Packet Networks", *Technical Report TR 96-76,* Oct. 1996.

[30] Young, P., "Recursive Estimation and Time-Series Analysis", *Springer-Verlag, New York,* 1984.

[31] P. Ferguson and G. Huston, "Quality of Service", *John Wiley & Sons, New York,* 1998.

[32] D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service. IEEE/ACM Transactions on Networking", *IEEE/ACM Transactions on Networking, vol. 6, no. 4, pp. 362-373*, August 1998.

[33] N. Seddigh, B. Nandy and P. Pieda, "Study of TCP and UDP Interactions for the AF PHB", *Internet Draft, <draft-nsbnpp- DiffServ-tcpudpaf-00.txt>*, June 1999.

[34] J. Postel, Editor "Internet Protocol", *RFC791*, September 1981. *http://ietf.org/rfc/rfc791.txt.*

[35] C. Partridge, "Gigabit Networking", *Addison Wesley, Reading, MA*, 1994.

[36] D. Clark, S. Schenker and L. Zhang, "Supporting Real-Time Applications in an Integrated Packet Services Network", *SIGCOMM'92*, Aug. 1992.

[37] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case", *IEEE/ACM Transactions on Networking, vol.1, no.3, pp.344-357*, June 1993.

[38] M. Goyal, A. Durresi, R. Jain, C. Liu, "Performance Analysis of Assured Forwarding", *Internet Draft, <draft-goyal-diffserv-afstdy-00>*, Feb. 2000.

Appendix

Confidence Interval

The accuracy of simulation results is normally described in terms of confidence intervals placed on the mean values of these results. The procedure to calculate the confidence intervals is described in this appendix.

Let $Y_1, Y_2, Y_3, \ldots, Y_N$ be the simulation results of the same experiment resulting from N different runs. Furthermore, these results are assumed to be statistically independent. The sample mean, $\bar{Y}$, of these results is given by

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

and the variance of the distribution of the sample values, S_Y^2, is defined as follows:

$$S_Y^2 = \frac{1}{N-1}\sum_{i=1}^{N} (Y_i - \bar{Y})^2$$

the standard deviation of the sample mean is given by:

$$\frac{S_Y}{\sqrt{N}}$$

The Mean of simulation runs may fall in the interval ±ε within the actual mean with a certain probability drawn from the *t*-distribution.

$$\varepsilon = \frac{S_Y t_{\alpha/2,N-1}}{\sqrt{N}}$$

where $t_{\alpha/2,N-1}$ is the value of the *t*-distribution with *N*-1 degrees of freedom with probability $\alpha/2$.

The confidence intervals with respect to the simulation results have upper and lower limits, which are defined as follows:

$$LowerLimit = \overline{Y}_N - \varepsilon$$

$$UpperLimit = \overline{Y}_N + \varepsilon$$

In this thesis, 90% confidence interval for each data point was obtained based on 5 independent runs per simulation experiment. The simulation running times have been chosen long enough to ensure stability and tight confidence intervals. In all simulation results presented in this thesis, the confidence intervals rarely exceeded 10% of their corresponding mean values.

Vita

Name: Haiqing Chen

Education: Queen's University, 1999 – 2001
M.Sc. in Computing and Information Science

Zhejiang University, China, 1988 - 1992
B.Sc. in Electrical Engineering

Place and Year of Birth: China, 1969

Experience: Research and Teaching Assistant, Department of Computing and Information Science, Queen's University, 1999 - 2001

Software Engineer, Blue Streak Electronics, 1998 - 1999

Software Engineer, Heuristics Inc, 1995 - 1996

Software Engineer, Inventec Electronics, 1992 – 1994

Awards: R. S. McLaughlin Fellowship, Queen's University, 1999
Research Assistant Award, Queen's University, 1999 – 2001
Teaching Assistant Award, Queen's University, 1999 – 2000