ABSTRACT

PROVIDING GUARANTEED QOS IN THE HOSE-MODELED VPN

by
Dong Wei

With the development of the Internet, Internet service providers (ISPs) are required to offer revenue-generating and value-added services instead of only providing bandwidth and access services. Virtual Private Network (VPN) is one of the most important value-added services for ISPs.

The "classical" VPN service is provided by implementing layer 2 technologies, either Frame Relay (FR) or Asynchronous Transfer Mode (ATM). With FR or ATM, virtual circuits are created before data delivery. Since the bandwidth and buffers are reserved, the QoS requirements can be naturally guaranteed. In the past few years, layer 3 VPN technologies are widely deployed due to the desirable performance in terms of flexibility, scalability and simplicity. Layer 3 VPNs are built upon IP tunnels, *e.g.*, by using PPTP, L2TP or IPSec. Since IP is "best-of-effort" in nature, the QoS requirement cannot be guaranteed in layer 3 VPNs. Actually, layer 3 VPN service can only provide secure connectivity, *i.e.*, protecting and authenticating IP packets between gateways or hosts in a VPN. Without doubt, with more applications on voice, audio and video being used in the Internet, the provision of QoS is one of the most important parts of the emerging services provided by ISPs. An intriguing question is: "Is it possible to obtain the best of both layer 2 and 3 VPN? Is it possible to provide guaranteed or predictable QoS, as in layer 2 VPNs, while maintaining the flexibility and simplicity in layer 3 VPN?" This question is the starting point of this study.

The recently proposed hose model for VPN possesses desirable properties in terms of flexibility, scalability and multiplexing gain. However, the "classic" fair bandwidth allocation schemes and weighted fair queuing schemes raise the issue of low

overall utilization in this model. A new fluid model for provider-provisioned virtual private network (PPVPN) is proposed in this dissertation. Based on the proposed model, an idealized fluid bandwidth allocation scheme is developed. This scheme is proven, analytically, to have the following properties: 1) maximize the overall throughput of the VPN without compromising fairness; 2) provide a mechanism that enables the VPN customers to allocate the bandwidth according to their requirements by assigning different weights to different hose flows, and thus obtain the predictable QoS performance; and 3) improve the overall throughput of the ISPs' network. To approximate the idealized fluid scheme in the real world, the 2-dimensional deficit round robin (2-D DRR and 2-D DRR+) schemes are proposed. The integration of the proposed schemes with the best-effort traffic within the framework of virtual-router-based VPN is also investigated. The 2-D DRR and 2-D DRR+ schemes can be extended to multi-dimensional schemes to be employed in those applications which require a hierarchical scheduling architecture. To enhance the scalability, a more scalable non-per-flow-based scheme for output queued switches is developed as well, and the integration of this scheme within the framework of the MPLS VPN and applications for multicasting traffics is discussed. The performance and properties of these schemes are analyzed.

PROVIDING GUARANTEED QOS IN THE HOSE-MODELED VPN

by
Dong Wei

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

May 2004

APPROVAL PAGE

PROVIDING GUARANTEED QOS IN THE HOSE-MODELED VPN

Dong Wei

Dr. Nirwan Ansari, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Jianguo Chen, Committee Member — Date
MTS, Agere Systems

Dr. Symeon Papavassiliou, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Teunis Ott, Committee Member — Date
Professor of Computer and Information Science, NJIT

Dr. Lev Zakrevski, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Dong Wei

Degree: Doctor of Philosophy

Date: May 2004

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2004
- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2001
- Bachelor of Engineering in Electrical Engineering,
  Tsinghua University, Beijing, P.R. China, 1991

Major: Electrical Engineering

Presentations and Publications:

D. Wei and N. Ansari,
"Approximating H-GPS with Multi-Dimensional Deficit Round Robin Scheme,"
to be submitted.

D. Wei and N. Ansari,
"A Novel Modified Secant Method for Computing the Fair Share Rate,"
IEE Proceedings on Communications, submitted.

D. Wei and N. Ansari,
"Implementing a Tiered Deficit Round Robin Scheme in Packet Networks,"
IEEE Globecom 2004, submitted.

D. Wei and N. Ansari,
"Implementing Fair Bandwidth Allocation Schemes in the Hose-modeled VPN,"
IEE Proceedings on Communications, accepted.

D. Wei, J. Yang, N. Ansari and S. Papavassiliou,
"Cell-based Schedulers with Dual-rate Grouping,"
IEICE Transactions on Communications, Vol. E86-B, No. 2. pp 637-645, February 2003.

D. Wei, J. Yang, N. Ansari and S. Papavassiliou,
"Guaranteeing Service Rates for Cell-based Schedulers with a Grouping Architecture,"
IEE Proceedings on Communications, Vol. 150, Issue 1. pp 1-5, February 2003.

D. Wei and N. Ansari,
"On IP Traffic Monitoring,"
Book Chapter. Intelligent Virtual World: Technologies and Applications in Distributed Virtual Environments, WSP, 2004.

D. Wei, J. Yang, N. Ansari and S. Papavassiliou,
"Implementing the Dual-rate Grouping Scheme in Cell-based Schedulers,"
Proceedings of IEEE Global Telecommunications Conference 2002, Vol. 3. pp 2410-2414, 2002.

D. Wei and N. Ansari,
"Implementing IP Traceback in the Internet - An ISP Perspective,"
Proceedings of the 2002 IEEE Workshop on Information Assurance, pp 326-332, June 2002.

J. Yang, D. Wei, S. Papavassiliou and N. Ansari,
"Improving Service Rate Granularity by Dual-rate Session Grouping in Cell-based Schedulers,"
Proceedings of IEEE Global Telecommunications Conference 2001, Vol. 4. pp 2425-2429, 2001.

D. Wei, N. Ansari and J. Chen,
"A Compressed and Dynamic-range-based Expression of Timestamp and Period for Timestamp-based Schedulers,"
Proceedings of IEEE Global Telecommunications Conference 2001, Vol. 4. pp 2353-2357, 2001.

D. Wei, N. Ansari and J. Chen,
"An Efficient Expression of Timestamp and Period in Packet-based and Cell-based Schedulers,"
Proceedings of IEEE International Conference on Communications 2001, Vol. 1. pp 95-99, 2001.

D. Wei and N. Ansari,
"IP Traffic Monitoring: An Overview and Future Considerations,"
Proceedings of the second IEEE Pacific-Rim Conference on Multimedia, October 2001

To my parents and my wife, without whom this would not have been possible

ACKNOWLEDGMENT

I should thank many people. Without their guidance, support and help, this dissertation could not be completed.

First, I would like to thank my dissertation advisor, Prof. Nirwan Ansari. Prof. Ansari always gave me guidance, help and support whenever I needed. From him, I have learned how to do research work. I appreciated his patience in revising my papers. Without that, those papers might not be published. I also appreciated that, in the past two years, I had enough freedom to study this interesting topic.

I would like to thank my dissertation committee members. Prof. Papavassiliou gave me many insightful comments on several published papers. Dr. Chen also gave me constructive suggestions on my research. Prof. Ott taught me on Internet QoS, from which I received many good ideas. Prof. Zakrevski took time out of his busy schedule to provide great help. Their participation made a great difference in this dissertation.

I would also like to thank the faculty of the ECE department. They taught me, advised me, and helped me to develop my knowledge and skills.

Most especially, from the bottom of my heart, I would like to thank my wife, Zheng Lin, for her constant support and understanding. Without her steadfast confidence in me, I would not be able to complete this dissertation. My parents were always emotionally supportive while encouraging me to get things done. My deepest appreciation goes to them for their support and encouragement.

Last but not the least, I would like to thank all my friends, especially Jie Yang and Gang Chen, and other colleagues in the Advanced Networking Lab (ANL) for their friendship, great help and support. I was able to receive much useful information from them in our ANL group seminars. It will always be good memories.

This work has been supported in part by the New Jersey Commission on Higher Education via the NJ I-TOWER project, and I am grateful for the support.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

With the development of the Internet, the Internet service providers (ISPs) are required to offer revenue-generating and value-added services instead of only bandwidth and access services [1]. VPN is one of the most important value-added services which can be provided by ISPs.

"Virtual Private Network" (VPN) facilitates the communication among a set of sites and provides customers with predictable and secure network connections over a shared network infrastructure. Multiple sites of a private network may therefore communicate via the public infrastructure, mimicking the operation of the private network. The logical structure of VPN, such as topology, addressing, connectivity, reachability, and access control, is equivalent to part of or all of a conventional private network using private facilities.

As shown in Fig. 1.1, a virtual private network has multiple sites, such as headquarters, branch offices, manufactories, home offices, suppliers, and remote users. Those sites are connected over a public IPS network. From the VPN customers' perspective, they seem to have their own private network. Since the Internet Assigned Numbers Authority (IANA) [2] has reserved the following three blocks of the IP address space for private networks [3]: 1) 10.0.0.0 - 10.255.255.255, 2) 172.16.0.0 - 172.31.255.255 and 3) 192.168.0.0 - 192.168.255.255. An enterprise or an organization can assign internal (unofficial) IP addresses to their network components, such as computers, printers, routers, and servers, and manage the access control by themselves.

To compare with "the dedicated line" in the old world, VPNs have two main advantages:

- VPNs can reduce the total cost of ownership of a real private network.

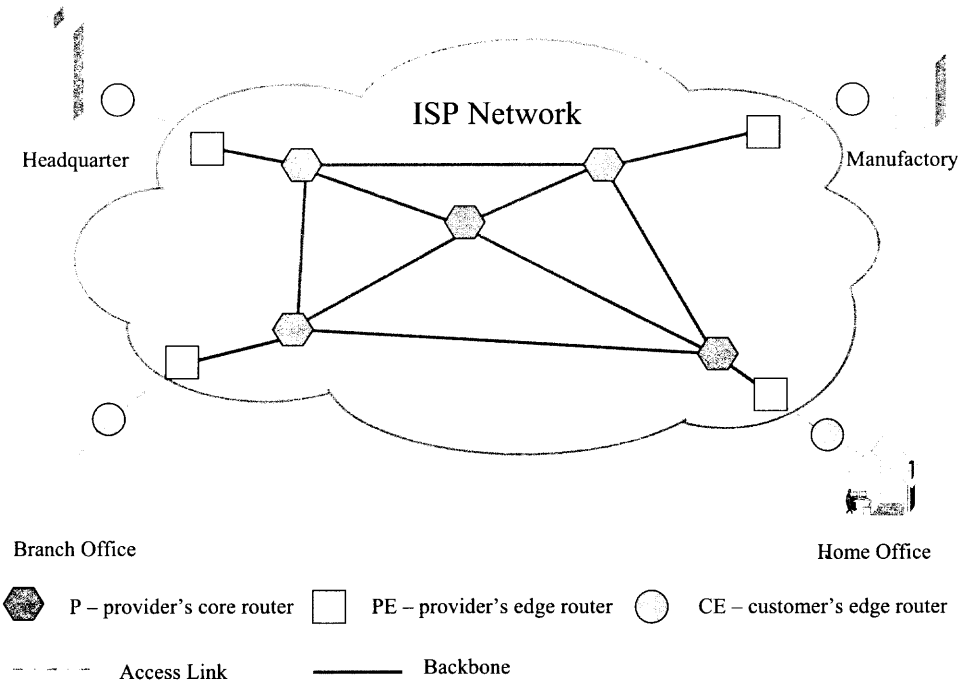

Figure 1.1 An example of virtual private network

- VPNs can be provisioned, dynamically and more easily, according to customers' requirements.

Provider-provisioned VPN (PPVPN) service is the service provided by the Internet service provider, via network components, such as ISP backbones, provider edge routers and provider core routers, in the ISPs' cloud, as shown in Fig. 1.1.

The "classical" VPN service is provided by implementing layer 2 technologies, either Frame Relay (FR) [4] or Asynchronous Transfer Mode (ATM) [5]. With FR or ATM, virtual circuits are created before traffic delivery. Since the bandwidth and buffers are reserved, the QoS requirements can be naturally guaranteed. In the past few years, layer 3 VPN technologies are widely deployed due to the desirable performance in terms of flexibility, scalability and simplicity. Layer 3 VPNs are built upon IP tunnels, *e.g.*, by using PPTP and IPSec. Since IP is "best-effort" in nature, the QoS requirement cannot be guaranteed by layer 3 VPNs. In fact, layer 3 VPN

service can only provide secure connectivity, *i.e.*, protecting and authenticating IP packets between gateways or hosts in the VPN. Without doubt, with more applications on voice, audio and video being used in the Internet, the provision of QoS is one of the most important parts of the emerging services provided by ISPs. An intriguing question is: "Is it possible to obtain the best of both layer 2 and layer 3 VPNs? Is it possible to provide guaranteed QoS, as in layer 2 VPNs, while maintaining the flexibility and simplicity in layer 3 VPN?" This question is the starting point of this dissertation.

Note that this research is conducted from the service providers' perspective, and thus the scope of this research work is within the ISPs' network cloud. Since the initial and ending network component of a packet could be a host in the VPN customers' sub network, which is behind the customer edge router, this part is out of the ISPs' reach. Therefore, when the end-to-end QoS is mentioned in this dissertation, it means customer-edge-to-customer-edge, not the conventional host-to-host.

To partition resources, there are two service models for VPN [6]: the pipe model and the hose model. The hose model was proposed by Duffield, *et al*, in [6], for its scalability, flexibility and multiplexing gain. It is believed that the hose-modeled VPN is going to be one of the most important VPN services for ISPs.

1.1 Motivations

Although the hose-modeled VPN possesses desirable properties in terms of scalability, flexibility and multiplexing gain, it cannot provide guaranteed QoS itself. As known, the guaranteed QoS is what the VPN customers expect. Furthermore, there is quite a need for a mechanism that enables the VPN customers to manage their VPN and allocate their bandwidth according to their requirements. To meet these requirements, the objectives of this dissertation are:

- Providing guaranteed and predictable QoS for the VPN customers.

- Creating a mechanism that enables the VPN customers to allocate their VPN resources according to their own requirements.
- Maximizing the overall throughput of the VPN.
- Improving the overall throughput of the ISP's network.

1.2 The Scope of this Dissertation

This dissertation focuses on how to provide VPN services from the perspective of ISP. From this point, a service model, based on hose model, is developed to provide guaranteed QoS and enable the VPN customers manage their VPNs by themselves. At the same time, the overall throughput of each VPN and the overall throughput of the ISP's network should be maximized. Furthermore, implementation of the proposed schemes with the following desirable properties are considered:

- Implementation complexity
- Scalability
- Flexibility
- Compatibility with the current VPN techniques, such as MPLS VPN [7, 8] and virtual router VPN [9]

Note that, in this dissertation, flow (or hose-flow) means a stream of data originated from one customer edge router (CE) and destined to another CE, instead of the normal definition of a stream of data which has the same tuple of source address, destination address, source port, destination port and protocol number.

1.3 Contributions

The main contributions of this dissertation are summarized as follows:

- The fluid model for provider-provisioned virtual private network (PPVPN) is proposed. Based on the proposed model, an idealized fluid bandwidth allocation scheme is developed, which is proven, analytically, to have the following properties: 1) maximize the overall throughput of the VPN without compromising fairness; 2) provide a mechanism that enables the VPN customers to allocate the bandwidth according to their requirements by assigning different weights to different hose flows, thus achieving the predictable QoS performance; and 3) improve the overall throughput of the ISP's network.

- To approximate the idealized fluid scheme, the 2-dimensional deficit round robin (2-D DRR and 2-D DRR+) schemes are proposed. Integration of the proposed schemes with the best-effort traffic, within the framework of virtual-router-based VPN, is presented.

- To enhance scalability, a more scalable non-per-flow-based scheme for output queued switches is proposed. Integration of this scheme within the framework of the MPLS VPN and applications for multicast traffics is investigated.

- To compute the fair share rate more accurately, a novel, modified secant method is proposed. The proposed method is demonstrated to achieve better performance in terms of convergence and accuracy than that proposed in [10].

- Although the 2-D DRR and 2-D DRR+ schemes are proposed to approximate the idealized fluid scheme in a hose-modeled VPN and provide guaranteed bandwidth to both individual flows and aggregation of flows, they can also be deployed when a "tiered" scheduling scheme is required. These schemes can be extended to multi-dimensional.

1.4 Organization

The rest of the dissertation is organized as follows:

Chapter 2 provides background information. In the first part, it reviews the definition and today's requirements of virtual private network, introduces the current VPN technologies and two service models for VPN, and reviews the existing approach to manage QoS in the hose-modeled VPN. In the second part, it presents the basics of QoS. Finally, the max-min fair bandwidth allocation scheme is reviewed.

Chapter 3, first, introduces a fluid VPN model and discusses its properties. In the second part, a fluid hose-modeled VPN is proposed and its properties are presented, upon which an idealized fluid bandwidth allocation scheme is proposed. It is proved, analytically, that the proposed scheme is able to maximize the overall through-put of the VPN while enabling the VPN customers to manage their network resources in terms of bandwidth according to their own QoS requirements.

Chapter 4, first, describes the DRR scheme and demonstrates the issue due to the "flat" structure. The second part introduces the ERR scheme and discusses its limitations. Then, it presents the 2-D DRR scheme and 2-D DRR+ scheme, and proposes how to integrate the best-effort traffic with the proposed schemes. It also discourses the implementation of the proposed schemes within the framework of the current virtual-router-based VPN and the applications of the proposed schemes in other scenarios. To provide the guaranteed QoS, the admission control issue is also discussed.

Chapter 5, first, reviews the core-stateless fair queueing (CSFQ). Based on the CSFQ scheme, a modified method, which is much more scalable, to approximate the idealized fluid bandwidth allocation scheme, is presented. Then, a new modified secant approach is presented to compute the fair share rate. The geometrical reasoning and numerical results demonstrate that the proposed, modified secant method achieves better performance in terms of convergence and accuracy than that of the original

method proposed in [10]. Applying the proposed scheme for multicast traffic and implementing it within the framework of MPLS VPN are investigated. In order to provide the guaranteed QoS, the admission control issue is discoursed.

Finally, Chapter 6 concludes the dissertation, presents the limitations of this work, and discusses the directions for future work.

CHAPTER 2

BACKGROUND

In this chapter, various service models, techniques of facilitating VPN are reviewed, and the corresponding performance is compared. QoS and the idealized fluid bandwidth allocation scheme are discussed, briefly.

2.1 Virtual Private Network

Private networks are used to connect multiple users and enable them to communicate. The network users are required to build up a dedicated network. Virtual private networks are employed to reduce the total cost of ownership of a corporate network.

Virtual private network is defined in [11] as follows:

"VPN is a generic term that covers the use of a public or private networks to create groups of users that is separated from other network users and may communicate among them as if they were on a private network."

From the business perspective, a VPN can be classified as [12, 13]: 1) intra-organizational communication (intranet); 2) communication with other organizations (extranet); and 3) communication with mobile users, home workers, remote offices, and so on, through cheap dial-up media. Although these three classes of VPN solutions differ greatly in the level of security in their implementation, they cover most of the topologies and technologies provided by ISPs[12].

With the increasing need of the use of VPNs as a more cost effective means of building and deploying private communication networks for multi-site communication than the dedicated private networks, ISPs have an opportunity to provide such value-added service other than bandwidth and access services [1]. This VPN service provided by IPSs is called provider-provisioned VPN (PPVPN) [14, 15]. Generally, there are two basic requirements for the VPN service: 1) the customer's IP network may use non-unique, unofficial IP addressing, and 2) traffic in the VPN should

be protected, *i.e.*, isolated from other traffics. With the rapid transformation of the Internet into a commercial infrastructure and more real-time applications are employed, demands for QoS have rapidly emerged [16]. Therefore, QoS, measured by bandwidth, delay, jitter and packet loss rate, is an important requirement too [17]. Besides, an ideal PPVPN is also required to possess the following properties[18]:

1. Fairness - the ISP's network resource should be partitioned fairly among the VPN users and other customers.

2. Availability - the ISP's network resource is available to the VPN customers.

3. Restoration - in the event that one network component fails, the substitute picks up immediately.

4. Reporting - the ISP is able to collect the VPN traffic profile and report the events of traffic.

5. Manageability - the VPN customers are able to configure the topology of their VPN and allocate the reserved network resources according to their own requirements.

6. Immunity capability to flooding-based Denial-of-Service (DoS) [19] and distributed Denial-of-Service (DDoS) attacks.

Until today, no ISP is able to provide PPVPN services which include all the above properties. It is believed that, in the next few years, the study of PPVPN services will make a great progress.

Another important issue of VPN is the topology. An ideal VPN topology is supposed to be: 1) easily provisioned, managed, and restored, and 2) cost effective, efficient and scalable. There is always a trade-off between property 1 and 2. A full-mesh topology and a tree topology represent the two extreme points of this spectrum.

In the real world, the hybrid topology with partial-mesh, spoke-and-hub (partial-tree) is mostly employed in large VPN networks [12, 13].

2.1.1 VPN Technologies

PPVPN technologies can be classified, according to their implementations [15], as in Table 2.1.

<table>
<tr><td rowspan="5">VPN
Technologies</td><td rowspan="3">Layer 2 VPN</td><td>Point-to-point VPN</td><td>VPWS</td></tr>
<tr><td rowspan="2">Point-to-multipoint VPN</td><td>VPLS</td></tr>
<tr><td>IPLS</td></tr>
<tr><td rowspan="3">Layer 3 VPN</td><td rowspan="2">PE-based VPN</td><td>BGP/MPLS VPN</td></tr>
<tr><td>Virtual router VPN</td></tr>
<tr><td>CE-based VPN</td><td>IPSec VPN</td></tr>
</table>

Table 2.1 Categories of VPN technologies

ATM and Frame Relay provider networks are commonly used to provide layer 2 VPN services to customers. Traditionally, layer 2 VPN is implemented by creating "virtual circuits" or "virtual paths" in the provider networks [5, 4, 20]. With Virtual Private Wire Service (VPWS), also known as Virtual Leased Line Service (VLLS), a pseudo wire is an emulated point-to-point connectivity over a packet switched network that gives the possibility to interconnect two customer edge nodes with any L2 technology. Virtual Private LAN Service (VPLS), which is also known as Transparent LAN Service (TLS), is a provider service that emulates the full functionality of a traditional Local Area Network. A VPLS makes it possible to interconnect several LAN segments over a packet switched network (PSN) and makes the remote LAN segments behave as one single LAN. In a VPLS, the provider network emulates a learning bridge and forwarding decisions are taken based on MAC addresses or MAC addresses and VLAN tags. An IP-only LAN-like Service (IPLS) is very similar to a

VPLS, except that: 1) it is assumed that the CEs are hosts or routers, not switches; and 2) it is assumed that the service will only need to carry IP packets, and to support packets such as ICMP and ARP; otherwise layer 2 packets which do not contain IP are not supported. Since the resources in terms of bandwidth and buffer are partitioned into individual "virtual paths" or "virtual circuits", layer 2 VPN can naturally meet the QoS requirements, and also possesses the immunity capability to flooding-based DoS and DDoS which attempt to degrade the service by consuming the bandwidth and buffer.

A layer 3 VPN interconnects several sets of hosts and routers, and allows them to communicate based on layer 3 (network layer) addresses [1]. It has drawn much attention in the past few years due to its desirable properties in terms of flexibility, scalability and manageability. In a CE-based (Customer Edge router or gateway) VPN, the service provider network shared by multiple customers does not have any knowledge of the customer VPN. This information is limited to CE nodes. The CE-based VPNs may use tunnel-mode IPSec (or may perform transport-mode IPSec on IP-in-IP encapsulated packets) with ESP (Encapsulating Security Payload) to encrypt the customer's packets before sending them to the PE (service Provider Edge router or gateway) devices [21]. This means that the information in the customer's IP packet (IP header and payload) is unusable by the PE device. An important consequence is that the ISP's network cannot use this information for QoS-related tasks. Some of the IP packet's IP header information might be copied or translated though into equivalent information in the visible outer IP header, at the CE devices. The intermediate routers in the ISP's network treat these packets as normal IP packets, since the Internet itself can only provide best-effort service, and thus IPSec VPN itself cannot provide guaranteed QoS without a resource reservation mechanism. Most commercial products employ layer 3 CE-based VPN technologies. In a layer 3 PE-based (service Provider Edge router or gateway) VPN, a service provider network

is used to interconnect customer sites using shared resources. Specifically, the PE devices maintain the VPN state, isolating the traffic of customers of one VPN from those of other VPNs. Since the PE devices maintain all required VPN state, the CE devices may behave as if they were connected to a private network. Specifically, the CE devices in a PE-based VPN must not require any changes or additional functionality to be connected to a PPVPN instead of a private network. MPLS (Multi-Protocol Labeling Switch) [22, 23] is the IETF standardized technology that combines the deterministic traffic engineering control of layer 2 ATM switching along with the flexible topology management of IP routing. MPLS accomplishes this by combining label switching with IP routing protocols. In a BGP/MPLS VPN, each PE router maintains a separate forwarding environment for each VPN and a separate forwarding table for each VPN [8, 7, 12, 13]. In order to maintain multiple forwarding table instances while running only a single routing protocol instance, BGP/MPLS VPNs mark route advertisements with attributes that identify their VPN context. The BGP/MPLS VPNs are based on the approach described in [7] and [8]. In a Virtual Router VPN, each router in the ISP's network maintains a complete logical router for each VPN that it supports, *i.e.*, each router in the ISP's network behaves as a virtual router for each VPN [9]. Each logical router maintains a unique forwarding table and executes a unique instance of the routing protocols. The virtual router VPNs possess two main desirable properties: 1) the VPN customers are able to manage their VPN topology by themselves; and 2) it allows a complete separation of the VPN customers' routing from ISPs' backbone routing, *i.e.*, the VPN customers retain control over their network routing and can run their own IGPs (Interior Gateway Protocols), rather than being required to learn and run BGP (Border Gateway Protocol) [24] to connect to and distribute routes into the VPN services. The virtual router VPNs are described in [9].

In layer 3 VPNs, one of the most important issues is QoS. Since the routers in the ISPs' network treat each encapsulated packet as a regular IP packet and forward it with their "best-effort", the QoS requirements cannot be guaranteed. Another issue is its vulnerability to flooding-based DoS and DDoS, when attackers attempt to send huge volume of traffic to a VPN site or another node via a specific intermediate router, the bandwidth to this VPN site or the buffer of the intermediate router is depleted. Then, the packets from other sites of VPN to this site will be discarded. Therefore, the service to this VPN site can be degraded and even denied.

2.1.2 The Service Models of VPN

In order to provide guaranteed or predictable QoS, two PPVPN service models have been proposed [6]: 1) the pipe model, where the QoS specifications are given on a per pair of endpoints, and 2) the hose model, where the QoS specifications are given for each demarcation point between the user and the network.

The difference between the hose model and the pipe model is illustrated by means of the following example.

Suppose one company has four offices, in Boston, New York, San Francisco and Los Angles, respectively, as shown in Fig. 2.1. An ISP provides VPN service for this company to connect these sites. The average traffic load between any two sites is 10 Mbps.

The Pipe Model

With the pipe model, a pipe (virtual circuit or virtual path) is created between each pair of endpoints (customer's edge router) with the minimum cost. Usually, the cost is the summation of the required bandwidth of each link on each virtual circuit, *i.e.*, it is computed by the hop count and bandwidth. In this scenario, six pipes are created:

1. San Francisco - Los Angles : PE_a - P_1 - P_2 - PE_b, with a cost of 30 Mbps;

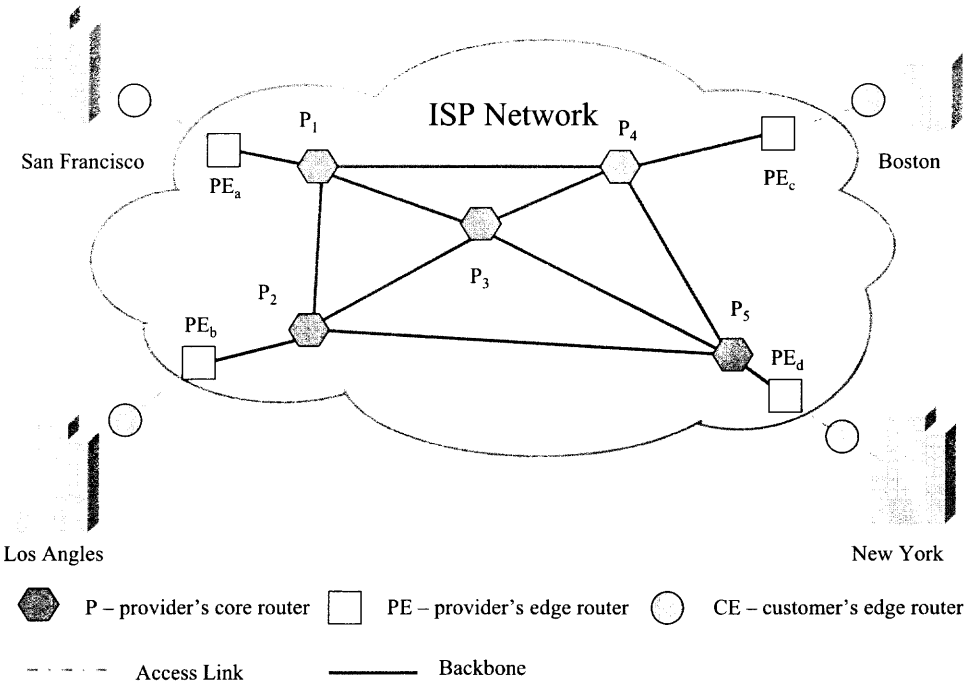

Figure 2.1 An example of the differences between the service models

2. San Francisco - Boston : PE_a - P_1 - P_4 - PE_c, with a cost of 30 Mbps;

3. San Francisco - New York : PE_a - P_1 - P_3 - P_5 - PE_d, with a cost of 40 Mbps;

4. Los Angles - Boston: PE_b - P_2 -P_3 - P_4 - PE_c, with a cost of 40 Mbps;

5. Los Angles - New York : PE_b - P_2 - P_5 - PE_d, with cost of 30 Mbps;

6. Boston - New York : PE_c - P_4 - P_5 - PE_d, with a cost of 30 Mbps.

The overall cost in terms of bandwidth of the ISP network is 200 Mbps. Each access link capacity (between the customer site and the ISP's network) is 30 Mbps. Each endpoint-to-endpoint transmission capacity is 10 Mbps. Certainly, the endpoint-to-endpoint transmission capacity can be increased to 30 Mbps by reserving 30 Mbps bandwidth on each pipe. However, this would lead to the low utilization issue of the

overall VPN. From this example, the features of the pipe model can be observed as follows:

1. Each pipe between any pair of endpoints is created independently, and the bandwidth on this pipe is reserved accordingly, and thus the QoS can be naturally guaranteed.

2. If any pair of endpoints is required to communicate, then the VPN has a fully-meshed topology.

3. To create the pipes, the VPN customers are supposed to specify the traffic of each pipe.

4. To add one more site in the VPN, if there are already N sites in the VPN, the ISP should create N new pipes.

5. No multiplexing gain among the VPN customers. Imagine one scenario that it is 10 AM eastern time on a working day. There may be heavy data traffic between Boston and New York that is greater than the average 10 Mbps, but these two sites can communicate only at a speed of 10 Mbps, although their access link capacity are 30 Mbps and there is very light traffic among other pipes. Note that the links on the 6th pipe can be also used by other customers, and therefore there could be multiplexing gain between this VPN customer and other customers. However, the ideal solution for the customers is to share the reserved bandwidth of one VPN among the users in the same VPN.

6. The VPN customer cannot control or manage their VPN except the specification of the traffic on each pipe.

The pipe model is traditionally implemented with ATM or FR technologies. The best feature of the pipe model is the guaranteed QoS, if and only if the VPN customer

is able to specify the traffic characteristic on each pipe. The main drawbacks are: 1) scalability - a fully-meshed topology is not scalable and it is also, if not impossible, very difficult for the VPN customers to specify the complete traffic matrix, while the number of endpoints of a VPN is constantly increasing and the traffic pattern between each pair of endpoints is becoming increasingly complex; 2) flexibility - the VPN customer cannot manage the VPN after the pipes are created; 3) no multiplexing gain among the users in the same VPN.

The Hose Model

In order to alleviate the shortcomings of the pipe model, the hose model has been proposed by Duffield, *et al* [6]. For a link between nodes u and v in a tree $\mathbb{T}$, denote $\mathbb{T}^u_{u,v}$, $\mathbb{T}^v_{u,v}$ as the connected components of $\mathbb{T}$ containing u and v, respectively, when the link between u and v is deleted. Denote $\mathbb{V}_{CE}$ as the set of endpoints. Denote $\mathbb{V}_{CE}^{u/(u,v)}$, $\mathbb{V}_{CE}^{v/(u,v)}$ as the endpoint set of $\mathbb{T}^u_{u,v}$ and $\mathbb{T}^v_{u,v}$, respectively. The remaining notations in this dissertation are listed in Table 2.2.

$\mathbb{G}\ (\mathbb{V}, \mathbb{E})$	Graph of a VPN with node set $\mathbb{V}$ and edge set $\mathbb{E}$.
$\mathbb{V}$	Nodes in a VPN.
$\mathbb{E}$	Edges in a VPN.
$\mathbb{V}_{PE}$	The set of provider edge routers (service access points).
$\mathbb{V}_P$	The set of provider core routers, where $V = \mathbb{V}_{CE} \cup \mathbb{V}_{PE} \cup \mathbb{V}_P$
N	The number of endpoints of $\mathbb{G}\ (\mathbb{V}, \mathbb{E})$, *i.e.*, $\lvert\mathbb{V}_{CE}\rvert$
M	The number of VPNs in $\mathbb{G}\ (\mathbb{V}, \mathbb{E})$
i, j	Endpoints in a VPN, where $i, j \in \mathbb{V}_{CE}$.
L_i^{In}	The ingress link capacity of endpoint i.
L_i^{Out}	The egress link capacity of endpoint j.
$L_{u,v}$	The link capacity of the directional link from u to v, if u and v, where $u, v \in \mathbb{V}$, are connected.
$h_{i,j}$	Hose flow, a stream of packets originated from endpoint i and destined to endpoint j in $\mathbb{G}\ (\mathbb{V}, \mathbb{E})$, where $i, j \in \mathbb{V}_{CE}$.
$a_{i,j}^u$	The arrival rate of hose flow $h_{i,j}$ at node u.
$d_{i,j}^u$	The departure rate of hose flow $h_{i,j}$ at node u.
$g_{i,j}$	The guaranteed bandwidth for the hose flow $h_{i,j}$ at node j.
f_j^u	The normalized fair share rate of hose flows destined to j at u.

Table 2.2 Notations used in the dissertation.

Definition 2-1: As shown in Fig. 2.2, a virtual private network is called hose-modeled VPN if it has the following features:

1. it has a tree topology $\mathbb{T}$,
2. $\forall u, v \in \mathbb{V},\ L_{u,v} = \min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u.v)}} L_j^{Out})$, when node u and v are connected.

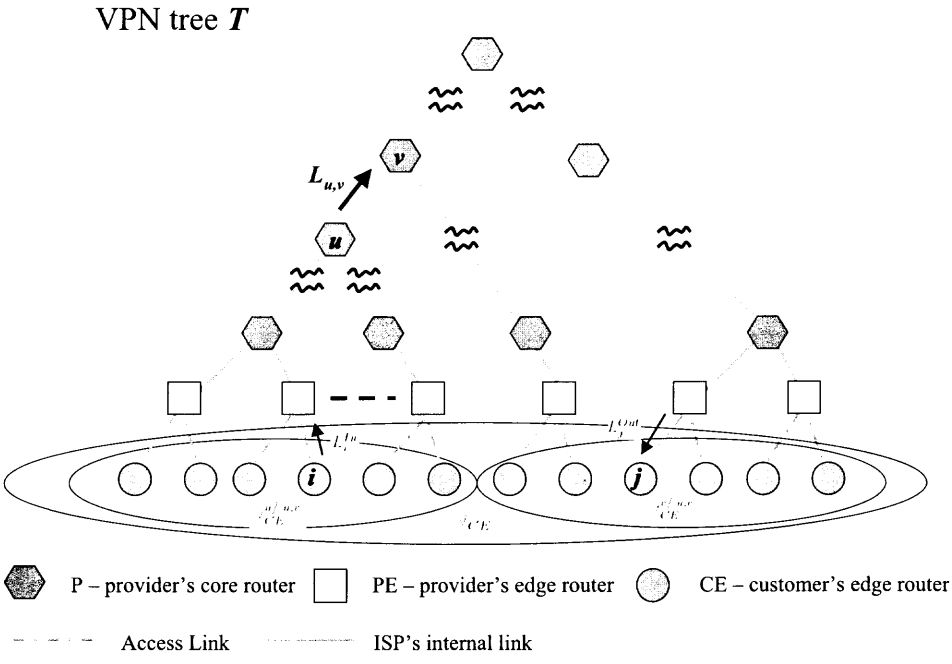

Figure 2.2 VPN tree topology

With the hose model, the logical topology of the above example is shown in Fig. 2.3. The capacity of each logical link of the VPN in the ISP network is 30 Mbps. The overall cost in terms of bandwidth of the ISP network is 240 Mbps. Each access link capacity (between the customer site and the ISP's network) is 30 Mbps. Each endpoint-to-endpoint transmission capacity is 30 Mbps. From this example, the features of the hose model can be observed as follows [6]:

1. Easy specification, the customers only need to specify the capacity of each link between a customers' edge router and the corresponding providers' edge router.

2. Flexibility, traffic from or to an endpoint can be distributed arbitrarily over other endpoints as long as the access link capacity of each endpoint are not violated.

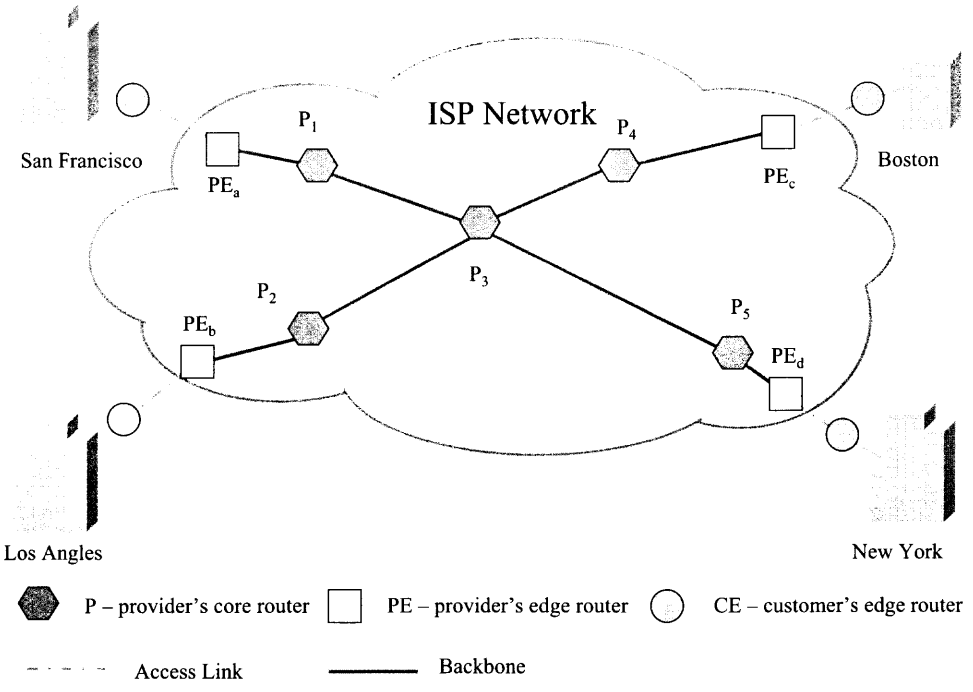

Figure 2.3 The logical topology of the above example with the hose model

3. Scalability, the tree topology is much more scalable than a fully-meshed topology and if one endpoint is added, it is not necessary to create an additional pipe between the new endpoint and each existing endpoint.

4. Simplicity of routing and restoration owing to the tree topology.

5. Multiplexing gain among the same VPN customers; due to the statistical multiplexing gain, the overall throughput of a VPN can be improved.

6. Characterization, the aggregated traffic from or to an endpoint is easier to be characterized than the individual endpoint-to-endpoint traffic since the aggregated traffic smoothes the statistical variation in the individual endpoint-to-endpoint traffic.

It is shown, in [6], that the hose model can achieve the above properties with the cost of a little additional bandwidth reservation under the Waxman model [25]

and the power-law model [26]. Note that, in the above example, the hose model needs to reserve extra 40 Mbps than the pipe model. Although the hose model alleviates the shortcomings in the pipe model, there are two main drawbacks: 1) no guaranteed QoS; and 2) the VPN customer still cannot manage their own VPN.

The provisioning and restoration algorithms of the hose-modeled VPN can be found in [27] and [28], respectively.

2.1.3 Managing QoS in the Hose-modeled VPN

In order to provide guaranteed QoS in the hose-modeled VPN, Duffield, *et al*, proposed an approach to reallocate resource to each hose flow [6]. Assuming that the bandwidth is the only QoS metric, *i.e.*, the QoS is guaranteed as long as the bandwidth requirement is met. With this approach, a hose (link between customer's edge and ISPs' edge) is implemented by a mesh of provider pipes between the ingress and egress provider's edge routers. This approach can be summarized as follows:

1. Measuring. The traffic of each hose flow is measured at the endpoint from which it initiates.

2. Predicting. The traffic rate of each hose flow is predicted according to the corresponding measurement.

3. Resizing. The bandwidth of each pipe between each pair of endpoints is reallocated, dynamically, according to the predicted traffic rate of each hose flow.

Dynamically resized VPN gain, which reflects the multiplexing gain by using the hose model, is the ratio of the maximum offered traffic over the length of the experiment to the time average of the renegotiated hose requirement. It was shown by experiments [6] that greater resizing frequency leads to greater dynamically resized VPN gain. The reason is that more frequent resizing leads to more accurate resizing. Since more frequent resizing needs more traffic sampling, which leads to more accurate

measurement and prediction, more traffic sampling and frequent predicting need more CPU processing power, and frequent resizing needs more bandwidth to transmit signaling packets to reserve bandwidth. As it is known, no traffic prediction scheme is perfect because the internet traffic is bursty and uncertain by nature [29]. Therefore, it is believed that a proper approach without the need of traffic measurement and prediction could potentially further improve the multiplexing gain, and thus could obtain a greater overall throughput. Furthermore, since the bandwidth is reallocated according to traffic measurement and prediction, the VPN customers are not able to manage the VPN according to their requirements, and thus the predictable QoS performance cannot be obtained.

2.2 Providing QoS in the Internet

The concept of QoS did not exist at the infancy of the Internet. Whether packets could arrive to their destinations was the foremost concern. According to "first-come-first-serve" policy, intermediate nodes (routers) forward packets as fast as possible, *i.e.*, the Internet only provided best effort service. With the emergence of new real time applications, such as video conferencing and voice over IP, QoS has become a necessary concern [16, 1]. There are two main driving forces for QoS: 1) from the ISP clients' perspective, companies that do business on the Web need QoS for better delivery of their content and services to attract more customers; 2) from the ISPs' perspective, ISPs need the QoS-based value-added services in their networks to increase their revenue.

Quality of service is a hotly debated topic both in the industry and academia. It is looked from different perspectives by the ISPs and their customers. From the ISP's point of view, QoS refers to the ability to provide different treatments to different traffics of different customers. The primary goal is to increase the overall utility of the network by granting priority to higher-value or more performance-sensitive flows.

"Priority" means either lower drop probability or preferential queuing at congested interfaces. QoS that attempts to increase the priority of some flows above the level given to the default best-effort service class, requires admission control and policing of those flows to prevent the theft of service. These services may provide hard worst-case performance guarantee to certain flows. The remaining traffics are receiving service on a best-effort basis. In either case, it should be noted that QoS does not prevent congestion or generate more bandwidth; it only adds "intelligence" at congested interfaces that allows the network to make intelligent decisions on how to queue or drop packets. From the ISP customers' point of view, QoS is the service quality they experience. For different customers and different applications, QoS means different things. The main metric to measure QoS quantitatively are [16]: 1) delay; 2) delay jitter; 3) bandwidth; and 4) loss rate. Note that, in the research community, the QoS metric is analyzed by means of policing, queuing management, buffer management, and scheduling schemes, *i.e.*, other factors, such as propagation delay and loss on the physical link, are not considered.

There are two classes of guaranteed QoS [17]: 1) hard guarantees - QoS guarantees that are precisely provided to individual end users, regardless of traffic conditions; 2) soft guarantees - QoS guarantees that are provided to aggregates, or classes of users; these guarantees translate to guarantees to the individual users that are not as precise as the hard guarantee.

To provide QoS in a network, resources such as bandwidth and buffer need to be reserved, packets are queued and scheduled at the intermediate routers, and congestion management is required. The resource reservation is performed based on signaled traffic parameters. It is enabled with the current resource reservation technique - RSVP [30]. In order to ensure that a flow does not exceed its resource usage by sending more than it reserves, a traffic policer is required to "police" the traffic of each flow. Therefore, some function blocks that are needed in the

intermediate routers and networks to provide guaranteed QoS are: 1) traffic parameter signaling and corresponding resource reservation schemes, *i.e.*, Call Admission Control (CAC); 2) traffic policing and shaping schemes to ensure that a flow does not exceed its traffic contract; 3) traffic classification schemes in order to associate the traffic flow to the corresponding reserved resources; 4) buffer management and congestion control schemes; and 5) traffic scheduling and queuing schemes.

There are also two models to provide guaranteed QoS - IntServ [31] and DiffServ [32].

With the IntServ model [31], each packet is treated according to the state of the flow it belongs to. With proper admission control, and scheduling and buffer management schemes, hard guaranteed QoS can be obtained. However, it requires maintaining the state of each flow on every intermediate node, which is not scalable in the current high speed network with hundreds of thousands of flows. Due to the difficulty in implementing and deploying Intserv, Diffserv [32] has been introduced. The principle of Diffserv is to divide traffic into multiple classes, and treat them differently accordingly, especially when there is a shortage of network resources. With the DiffServ model, soft guaranteed QoS can be obtained. Compared with IntServ, DiffServ possesses desirable properties in terms of scalability and implementation simplicity. It is also believed that [17], with proper admission control, scheduling, and queuing, hard guaranteed QoS can also be achieved even with the DiffServ model.

2.3 Fair Bandwidth Allocation

Fair bandwidth allocation schemes play a very important, even a necessary, role in congestion control and provisioning guaranteed QoS. A bufferless fluid fair bandwidth allocation scheme is described in [10] as follows; note that the notations in [10] are modified to make them consistent with those in this dissertation:

Definition 2-2: Max-min fair bandwidth allocation of a single link: a set of sessions share a link with link capacity L, the arrival rate of session i at this link is x_i, and the weight of this session is ϕ_i. The bandwidth of this link is allocated such that the departure rate of session i, $y_i \leftarrow \min(f, x_i)$, where the fair share rate of this link, f, can be computed with the following algorithm. Note that, $\mathbb{Z}$ represents the set of flows whose arrival rates are greater than the fair share rate f, and $\overline{\mathbb{Z}}$ represents the set of flows whose arrival rates are not greater than the fair share rate f. Φ is the set of weights ϕ_i, where $\sum_{\forall \phi_i \in \Phi} \phi_i = 1$.

Algorithm 2-1: Fairshare($\mathbb{X}, \Phi, L$) // Computation of f of a single link

if ($\sum_{\forall x_i \in \mathbb{X}} x_i \leq L$) then

$f \leftarrow \max_{\forall x_i \in \mathbb{X}} \frac{x_i}{\phi_i}$;

else

$\mathbb{Z}^{(0)} \leftarrow \mathbb{X}$;

$\overline{\mathbb{Z}}^{(0)} \leftarrow \varnothing$;

for($x_i \in \mathbb{X}$)

$f^{(0)} \leftarrow \phi_i \cdot L$;

$y_i^{(0)} \leftarrow \min(f^{(0)}, x_i)$;

if($x_i > f^{(0)}$) then

$\mathbb{Z}^{(1)} \leftarrow x_i$;

else

$\overline{\mathbb{Z}}^{(1)} \leftarrow x_i$;

$k = 1$;

while($\mathbb{Z}^{(k)} \neq \mathbb{Z}^{(k-1)}$)

for($x_i \in \mathbb{X}$)

$f^{(k)} \leftarrow \frac{\phi_i}{\sum_{\forall x_i \in \mathbb{Z}^{(k)}} \phi_j} \cdot (L - \sum_{\forall x_i \in \overline{\mathbb{Z}}^{(k)}} x_j)$;

$y_i^{(k)} \leftarrow \min(f^{(k)}, x_i)$;

$$\text{if}(x_i > \phi_i^{(k)}) \qquad \text{then}$$

$$\mathbb{Z}^{(k+1)} \leftarrow \mathbb{Z}^{(k+1)} + x_i;$$

$$\text{else}$$

$$\overline{\mathbb{Z}}^{(k+1)} \leftarrow \overline{\mathbb{Z}}^{(k+1)} + x_i;$$

$$k \leftarrow k + 1;$$

$$f \leftarrow \max_{\forall x_i \in \mathbb{X}} \frac{y_i^{(k-1)}}{\phi_i};$$

return f;

The procedure of computing the fair share rate f can be summarized as follows:

1. Compute $f = \frac{L}{|\mathbb{X}|}$;
2. Find the flow with the minimum allocated bandwidth;
3. Subtract this rate at the link and eliminate the corresponding flow;
4. Compute in the reduced set of flows $f = \frac{L - \sum rate\_of\_eliminated\_flow}{|\mathbb{X}| - \sum rate\_of\_eliminated\_flow}$;
5. Repeat steps 2-4 until all flows are eliminated.

Note that, when a flow is bottlenecked at a downstream link, a feedback mechanism is necessary to inform the upstream links to constrain the flow, thus saving more bandwidth and further improving the overall throughput of the network [33]. However, the cost to improve the overall throughput in this scenario is the overhead of the feedback signaling, *i.e.*, the bandwidth and processing power of CPU.

2.4 Summary

In this chapter, various service models and techniques for facilitating VPN have been reviewed, and the corresponding performance has been discussed. The QoS issue in the Internet has been discoursed and the idealized fluid bandwidth allocation scheme has been introduced.

CHAPTER 3

The PROPOSED FLUID HOSE-MODELED VPN

As mentioned in Chapter 1, this research focuses on providing VPN service in the ISP's network. Therefore, from this chapter on, VPN is referred to as provider-provisioned VPN (PPVPN) which includes the network components in the ISP's cloud, plus customer edge routers.

In this chapter, with an idealized fluid VPN model, the bound of the overall VPN throughput and the bound of the endpoint-to-endpoint transmission capacity are analyzed. Based on the fluid VPN model, a fluid hose-modeled VPN, which can be seen as a "Superswitch" by the VPN customers, is proposed. With this model, an idealized fluid bandwidth allocation scheme, which can provide predictable QoS and enable the customer allocate bandwidth according to their requirements, is developed. With a given fluid hose-modeled VPN, it is proved, analytically, that the proposed fluid scheme is able to achieve the maximum overall VPN throughput, enabling the customer to transmit data at the maximum endpoint-to-endpoint transmission rate.

3.1 The Fluid Virtual Private Network

The fluid virtual private network is an idealized model, which is very useful to analyze the properties and characteristics of a VPN.

3.1.1 The Definition of the Fluid Virtual Network

Definition 3-1: A fluid virtual private network is a network which has the following properties:

1. It is composed of network components, such as customer edge routers (CEs), provider edge routers (PEs), provider core routers (Ps), and links among them;

2. The network could have an arbitrary topology $\mathbb{G}(\mathbb{E}, \mathbb{V})$, except that each P has no link connected directly to any CE, and each CE has only one link connected to a PE with ingress link capacity L_i^{In} (from CE to PE) and egress link capacity L_i^{Out} (from PE to CE), respectively;

3. Any traffic flow is initiated at a CE and destined to another CE;

4. Any network node (CE, PE and P) is bufferless;

5. $\forall i \in \mathbb{V}_{CE}$, $\sum_{\forall j \in \mathbb{V}_{CE}} d_{i,j}^i \leq L_i^{In}$ and $\forall j \in \mathbb{V}_{CE}$,$\sum_{\forall i \in \mathbb{V}_{CE}} a_{i,j}^j \leq L_j^{Out}$.

Definition 3-2: The overall throughput of the virtual private network $\mathbb{G}(\mathbb{E}, \mathbb{V})$ is defined as:

$$U = \sum_{\forall j \in \mathbb{V}_{CE}} u_j, \tag{3.1}$$

where u_j is the throughput of the egress link to endpoint j.

According to Eq. 3.1, the overall throughput of the network is the summation of the throughput of each egress link, u_j, where

$$u_j = \sum_{\forall i \in \mathbb{V}_{CE}} a_{i,j}^j. \tag{3.2}$$

3.1.2 Properties of the Fluid Virtual Private Network

Lemma 3-1: Given a fluid VPN with an arbitrary topology $\mathbb{G}(\mathbb{E}, \mathbb{V})$. $\forall j \in \mathbb{V}_{CE}$, the throughput of the egress link to endpoint j,

$$u_j \leq \min(\sum_{\forall i \in \mathbb{V}_{CE}} d_{i,j}^i, L_j^{Out}). \tag{3.3}$$

Proof: By properties 3 and 4 in Definition 3-1, $a_{i,j}^j \leq d_{i,j}^i$, and thus with Eq. 3.2,

$$u_j \leq \sum_{\forall i \in \mathbb{V}_{CE}} a_{i,j}^i. \tag{3.4}$$

By Eq. 3.2 and property 5 in Definition 3-1,

$$u_j \leq L_j^{Out}. \tag{3.5}$$

Combining Ineqs. 3.4 and 3.5, Ineq. 3.3 is obtained.

■

Lemma 3-2: In a fluid VPN, $\forall i, j \in \mathbb{V}_{CE}$, $a_{i,j}^j \leq \min(L_i^{In}, L_j^{Out})$.

Proof: $\because \sum_{\forall j \in \mathbb{V}_{CE}} d_{i,j}^i \leq L_i^{In}$, $\therefore a_{i,j}^i \leq L_i^{In}$. By properties 3 and 4 in Definition 3-1, $a_{i,j}^j \leq d_{i,j}^i$, thus $a_{i,j}^j \leq L_i^{In}$.

By property 5 in Definition 3-1, $\sum_{\forall i \in \mathbb{V}_{CE}} a_{i,j}^j \leq L_j^{Out}$. $\because a_{i,j}^j \leq \sum_{\forall i \in \mathbb{V}_{CE}} a_{i,j}^j$, $\therefore a_{i,j}^j \leq L_j^{Out}$.

$$\therefore a_{i,j}^j \leq \min(L_i^{In}, L_j^{Out}). \tag{3.6}$$

■

Definition 3-3: A single endpoint-to-endpoint transmission capacity $L_{i,j}$, $\forall i, j \in \mathbb{V}_{CE}$, is the maximum rate of traffic from i to j that endpoint j can observe.

By Lemma 3-2, it is easy to obtain the following theorem with Definitions 3-1 and 3-3.

Theorem 3-1: In a fluid virtual private network,

$\forall i, j \in \mathbb{V}_{CE}$, $L_{i,j} \leq \min(L_i^{In}, L_j^{Out})$.

Theorem 3-2: Given a VPN with any arbitrary topology $\mathbb{G}(\mathbb{E}, \mathbb{V})$, the overall throughput of the VPN,

$$U \leq \sum_{\forall j \in \mathbb{V}_{CE}} \min\Big(\sum_{\forall i \in \mathbb{V}_{CE}} d_{i,j}^i, L_j^{Out}\Big). \tag{3.7}$$

Proof: Ineq. 3.7 can be readily derived from Eq. 3.1, along with Eq. 3.2 and Ineq. 3.3.

■

Theorem 3-2 provides the maximum overall throughput of a bufferless, fluid virtual private network with an arbitrary topology. Ineq. 3.6 presents the maximum throughput of a single traffic flow, for given ingress and egress link capacities of each endpoint.

3.2 The Fluid Hose-modeled VPN

In the previous section, the maximum overall throughput of a bufferless, fluid virtual private network with an arbitrary topology is derived. In this section, based on the previous fluid VPN model and the hose-modeled VPN, another fluid VPN model is presented. It is able to maximize the overall throughput and enables the VPN customers to allocate the bandwidth according to their requirements by themselves.

3.2.1 The Definition of the Fluid Hose-modeled VPN

Definition 3-4: A VPN is said to be a fluid hose-modeled VPN, if it has the following properties:

1. the VPN has a tree topology and any traffic is initiated at a leaf node and destined to another leaf node, where a leaf node represents a CE, *i.e.*, endpoint;
2. $\forall u, v \in \mathbb{V}$, if u, v are connected, $L_{u,v} = \min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} L_j^{Out})$;
3. $\forall i \in \mathbb{V}_{CE}$, $\sum_{\forall j \in \mathbb{V}_{CE}} d_{i,j}^i \leq L_i^{In}$ and $\sum_{\forall j \in \mathbb{V}_{CE}} a_{i,j}^j \leq L_j^{Out}$;
4. $\forall j \in \mathbb{V}_{CE}$, the bandwidth of the egress link leading to leaf node j, $a_{i,j}^j$, is allocated according to the max-min fairness criterion with weight $\phi_{i,j}^j$, where link capacity L_j^{Out}, $a_{i,j}^j = \min(f_j^j \cdot \phi_{i,j}^j, a_{i,j}^i)$ and $\sum_{\forall i \in \mathbb{V}_{CE}} \phi_{i,j}^j = 1$.

The fair share rate f_j^j can be calculated recursively by Algorithm 2-1, Fairshare($\mathbb{X}$, Φ, L), where $\mathbb{X} = \{a_{1,j}^1, a_{2,j}^2, ...a_{i,j}^i..\}$, $\Phi = \{\phi_{1,j}^j, \phi_{2,j}^j, ...\phi_{i,j}^j..\}$, and $L = L_j^{Out}$.

3.2.2 Properties of the Fluid Hose-modeled VPN

Theorem 3-3: In a fluid hose-modeled VPN, the overall throughput is maximized, *i.e.*,

$$U = \sum_{\forall j \in \mathbb{V}_{CE}} \min(\sum_{\forall i \in \mathbb{V}_{CE}} d_{i,j}^i, L_j^{Out}). \tag{3.8}$$

Proof: By property 4 in Definition 3-4, the bandwidth of the egress link leading to leaf node j is allocated according to the max-min fairness criterion among hose flows destined to j.

According to the max-min fairness criterion, when $\sum_{\forall i \in \mathbb{V}_{CE}} d_{i,j}^i \leq L_j^{Out}$, $u_j = \sum_{\forall i \in \mathbb{V}_{CE}} a_{i,j}^j$; when $\sum_{\forall i \in \mathbb{V}_{CE}} d_{i,j}^i > L_j^{Out}$, $u_j = L_j^{Out}$. Thus, $u_j = \min(\sum_{\forall i \in \mathbb{V}_{CE}} d_{i,j}^i, L_j^{Out})$. Furthermore, Eq. 3.8 is obtained. By Ineq. 3.7, the overall throughput is maximized when Eq. 3.8 holds.

■

Theorem 3-3 demonstrates that, with the fluid hose-modeled VPN, the overall throughput of the VPN can be maximized, naturally. By Property 4 of Definition 3-4, this model enables the VPN customers to allocate the bandwidth according to their requirements. From the perspective of the VPN customers, the ISPs' cloud, which includes PE, P and links among them, can be considered as a "Superswitch", as shown in Fig. 3.1. At endpoint j, the VPN customer is able to allocate the bandwidth on the link destined to j, by assigning weight $\phi_{i,j}^j$ to hose flow $h_{i,j}$. If $\sum_{\forall i \in \mathbb{V}_{CE}} \phi_{i,j}^j \leq 1$, then $a_{i,j}^j$, the bandwidth allocated to $h_{i,j}$, can be guaranteed no less than $\phi_{i,j}^j \cdot L_j^{Out}$, when there is enough traffic volume in this hose flow. However, a centralized weight management mechanism is needed to assign weights to all hose flows. It can be implemented by a destination-based weight control approach, *i.e.*,

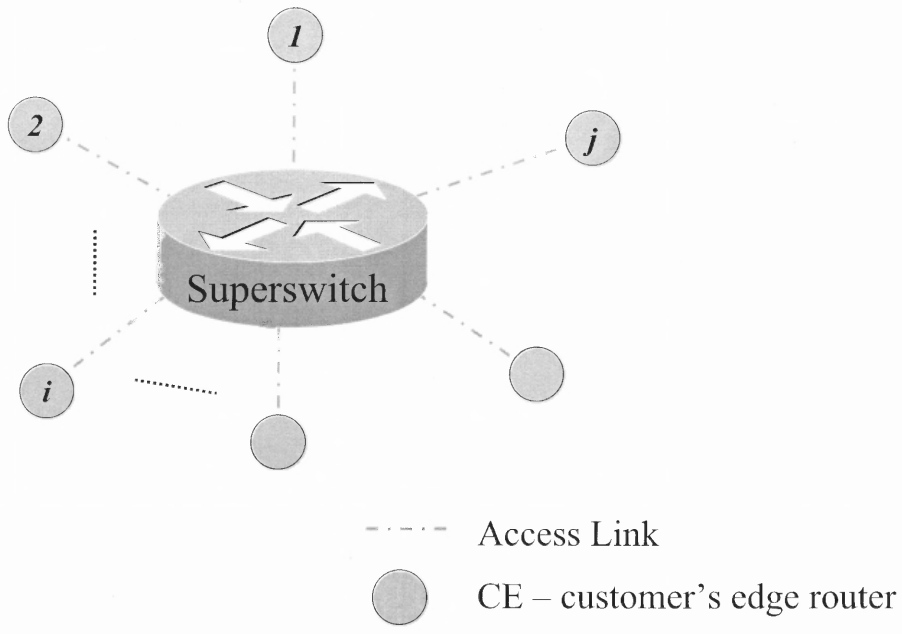

Figure 3.1 The "Superswitch" Model

if one endpoint tries to change its assigned weight, it first sends a weight request message to the destination node; the destination node has to perform the admission control scheme, and sends a reply message to accept or deny the weight request. The following theorem provides the maximum bandwidth allocated to hose flow $h_{i,j}$.

Theorem 3-4: In a fluid hose-modeled VPN, $\forall i, j \in \mathbb{V}_{CE}$, the throughput of a single hose flow can be maximized with given ingress and egress link capacities of each endpoint, *i.e.*, $a^j_{i,j}$ could be equal to $\min(L^{In}_i, L^{Out}_j)$.

Proof: Let $\phi^j_{i,j} = 1$, and $\phi^j_{k,j} = 0$ for $\forall k \in \mathbb{V}_{CE}$ and $k \neq i$.

When $d^i_{i,j} = L^{In}_i$, by property 4 in Definition 3-4, if $d^i_{i,j} \leq L^{Out}_i$, $f^j_{i,j} = a^i_{i,j}$, and thus $a^j_{i,j} = \min(f^j_{i,j}, a^i_{i,j}) = a^i_{i,j} = L^{In}_i$; if $d^i_{i,j} > L^{Out}_i$, by properties 3 and 4 in Definition 3-4, $a^j_{i,j} = L^{Out}_i$.

Thus, $a^j_{i,j} = \min(L^{In}_i, L^{Out}_j)$.

■

Theorem 3-1, in the previous section, demonstrates that the maximum endpoint-to-endpoint transmission capacity of hose flow $h_{i,j}$, in any PPVPN, is $\min(L_i^{In}, L_j^{Out})$. Theorem 3-4 shows that the maximum endpoint-to-endpoint transmission capacity can be achieved in the proposed reference fluid model. The following theorem on packet delay can be readily proved.

Theorem 3-5: In a fluid hose-modeled VPN, if a packet is sent from i to j at moment t_0, where $\forall i, j \in \mathbb{V}_{CE}$. Without considering propagation delay, the packet arrives node j at moment $t_0 + \frac{Size\_of\_packet}{a_{i,j}^j}$, *i.e.*, the packet delay is $\frac{Size\_of\_packet}{a_{i,j}^j}$.

Theorem 3-6: Given a network $\mathbb{G}(\mathbb{E}, \mathbb{V})$, any CE has only one link connected to a PE with ingress link capacity L_i^{In} (from CE to PE) and egress link capacity L_i^{Out} (from PE to CE), respectively; any traffic flow is initiated at a CE and destined to another CE; every network node (CE, PE and P) is bufferless; and $\forall i \in \mathbb{V}_{CE}$, $\sum_{\forall j \in \mathbb{V}_{CE}} d_{i,j}^i \leq L_i^{In}$ and $\forall j \in \mathbb{V}_{CE}$, $\sum_{\forall i \in \mathbb{V}_{CE}} a_{i,j}^j \leq L_j^{Out}$. $\forall j \in \mathbb{V}_{CE}$, assume v is connected to j, as shown in Fig. 3.2. On the directional link from v to j, the bandwidth for each flow is allocated by $a_{i,j}^j = \min(f_j^j \cdot \phi_{i,j}^j, a_{i,j}^v)$, where $\sum_{\forall i \in \mathbb{V}_{CE}} \phi_{i,j}^j = 1$. Denote $\mathbb{S}$ and $\mathbb{D}$ as the sets of endpoints, where $\forall i \in \mathbb{S}$, $a_{i,j}^v \leq f_j^j \cdot \phi_{i,j}^j$; $\forall k \in \mathbb{D}$, $a_{k,j}^v > f_j^j \cdot \phi_{k,j}^j$; and $\mathbb{V}_{CE} = \mathbb{S} \cup \mathbb{D} \cup \{j\}$. The fair share rate f_j^j on the link from v to j can be computed by Algorithm 2-1 in Chapter 2. Let us consider two scenarios. In Scenario 1, $\forall i \in \mathbb{V}_{CE}$, $a_{i,j}^{v(1)} = d_{i,j}^i$; denote $a_{i,j}^{j(1)}$ as the arrival rate of flow from i to j, at endpoint j; in Scenario 2, $\forall i \in \mathbb{V}_{CE}$, $d_{i,j}^i \geq a_{i,j}^{v(2)} \geq a_{i,j}^{j(1)}$; denote $a_{i,j}^{j(2)}$ as the arrival rate of flow from i to j, at endpoint j. Then, $\forall i \in \mathbb{V}_{CE}$ and $i \neq j$, $a_{i,j}^{j(2)} = a_{i,j}^{j(1)}$.

Proof: In Scenario 1, $\forall i \in \mathbb{S}^{(1)}$, where $a_{i,j}^{v(1)} \leq f_j^{j(1)} \cdot \phi_{i,j}^j$, then $a_{i,j}^{j(1)} = \min(f_j^{j(1)} \cdot \phi_{i,j}^j, a_{i,j}^{v(1)}) = a_{i,j}^{v(1)} = d_{i,j}^i$;

$\forall k \in \mathbb{D}^{(1)}$, where $a_{k,j}^{v(1)} > f_j^{j(1)} \cdot \phi_{k,j}^j$, then $a_{k,j}^{j(1)} = \min(f_j^{j(1)} \cdot \phi_{k,j}^j, a_{i,j}^{v(1)}) = f_j^{j(1)} \cdot \phi_{k,j}^j$, where

$$f_j^{j(1)} = \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j}. \tag{3.9}$$

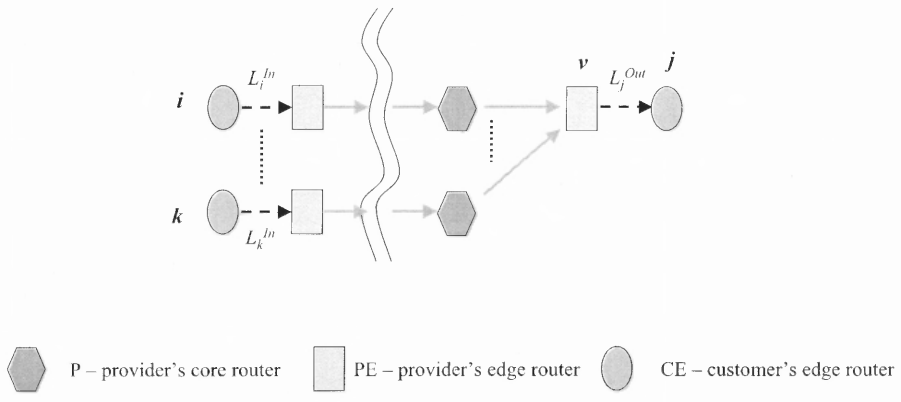

Figure 3.2 Endpoint j in the network $G(V,E)$

In Scenario 2, $\forall i \in \mathbb{S}^{(1)}$, $a_{i,j}^{j(1)} = a_{i,j}^{i}$.

$\because \forall i \in \mathbb{V}_{CE}$, $a_{i,j}^{i} \geq a_{i,j}^{v(2)} \geq a_{i,j}^{j(1)}$,

$\therefore a_{i,j}^{j(2)} = a_{i,j}^{j(1)} = a_{i,j}^{i}$, $\mathbb{S}^{(1)} \subseteq \mathbb{S}^{(2)}$, $\mathbb{D}^{(2)} \subseteq \mathbb{D}^{(1)}$.

$$f_j^{j(2)} = \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(2)}} d_{i,j}^{i}}{\sum_{\forall k \in \mathbb{D}^{(2)}} \phi_{k,j}^{j}}. \tag{3.10}$$

$\sum_{\forall i \in \mathbb{S}^{(2)}} d_{i,j}^{i} \geq \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^{i}$, and $\sum_{\forall k \in \mathbb{D}^{(2)}} \phi_{k,j}^{j} \leq \sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^{j}$. $\therefore f_j^{j(1)} \geq f_j^{j(2)}$.

Hypothesis: suppose $f_j^{j(1)} > f_j^{j(2)}$, and $\mathbb{D}^{(2)} \neq \mathbb{D}^{(1)}$, $\mathbb{S}^{(1)} \neq \mathbb{S}^{(2)}$.

Thus, there must exist some endpoint m, as shown in Fig. 3.3, where $m \in \mathbb{M}$, and $\mathbb{M} \subset \mathbb{D}^{(1)}$ and $\mathbb{M} \subset \mathbb{S}^{(2)}$. Thus, $\mathbb{V}_{CE} = \mathbb{M} \cup \mathbb{S}^{(1)} \cup \mathbb{D}^{(2)} \cup \{j\}$. $\mathbb{M}$ is a set of endpoints.

$\because m \in \mathbb{M}$, $\mathbb{M} \subset \mathbb{S}^{(2)}$,

$\therefore a_{m,j}^{j(2)} = a_{m,j}^{v(2)} \leq f_j^{j(2)} \cdot \phi_{m,j}^{j}$.

$\because m \in \mathbb{M}$, $\mathbb{M} \subset \mathbb{D}^{(1)}$,

$\therefore a_{m,j}^{j(1)} = f_j^{j(1)} \cdot \phi_{m,j}^{j} < a_{m,j}^{v(1)} = a_{m,j}^{m}$.

$\therefore a_{m,j}^{v(2)} \leq f_j^{j(2)} \cdot \phi_{m,j}^{j} < f_j^{j(1)} \cdot \phi_{m,j}^{j} = a_{m,j}^{j(1)}$.

$\therefore a_{m,j}^{v(2)} < a_{m,j}^{j(1)}$. This is contradictory to the statement that $\forall i \in \mathbb{V}_{CE}$, $a_{i,j}^{i} \geq a_{i,j}^{v(2)} \geq a_{i,j}^{j(1)}$.

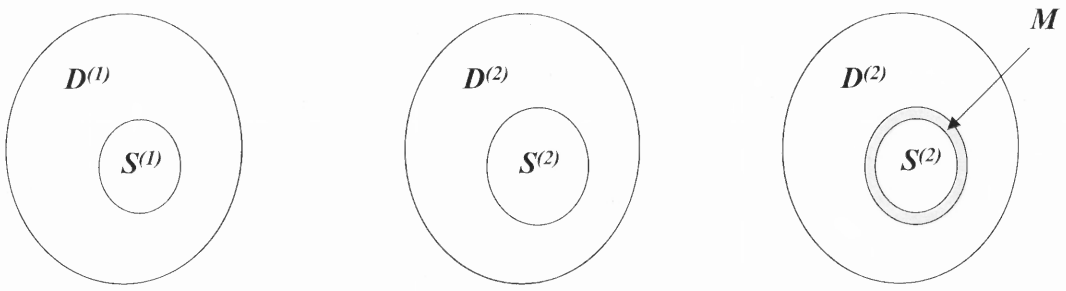

Figure 3.3 Illustration of the proof of Theorem 3-6

Thus, the hypothesis cannot hold and $f_j^{j(1)} = f_j^{j(2)}$, $\mathbb{M} = \varnothing$, and $\mathbb{S}^{(2)} = \mathbb{S}^{(1)}$, $\mathbb{D}^{(2)} = \mathbb{D}^{(1)}$.

$\forall k \in \mathbb{D}^{(1)}$, $\because a_{k,j}^{j(1)} = f_j^{j(1)} \cdot \phi_{k,j}^j$, and $a_{k,j}^{v(2)} \geq a_{k,j}^{j(1)}$,

$\therefore a_{k,j}^{v(2)} \geq f_j^{j(2)} \cdot \phi_{k,j}^j$

$\therefore a_{k,j}^{j(2)} = \min(f_j^{j(2)} \cdot \phi_{k,j}^j, a_{i,j}^{v(2)}) = f_j^{j(2)} \cdot \phi_{k,j}^j$.

and $\because f_j^{j(1)} = f_j^{j(2)}$,

$\therefore \forall k \in \mathbb{D}^{(2)}, a_{k,j}^{j(2)} = \min(f_j^{j(1)} \cdot \phi_{k,j}^j, a_{i,j}^{v(2)}) = f_j^{j(1)} \cdot \phi_{k,j}^j = a_{k,j}^{j(1)}$.

Therefore, $\forall i \in \mathbb{V}_{CE}$ and $i \neq j$, $a_{i,j}^{j(2)} = a_{i,j}^{j(1)}$.

■

Theorem 3-7: Given a network $\mathbb{G}(\mathbb{E}, \mathbb{V})$, $\forall u, v \in \mathbb{V}$, u and v are connected, and $\mathbb{V}_{CE}^{u(1)/(u,v)} \subseteq \mathbb{V}_{CE}^{u(2)/(u,v)} \subseteq \mathbb{V}_{CE}^{u/(u,v)}$. Let us consider two scenarios. In Scenario 1, only those endpoints belonging to set $\mathbb{V}_{CE}^{u(1)/(u,v)}$ are connected to u; $\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}$, $a_{i,j}^{u(1)} = d_{i,j}^{i(1)}$; denote $a_{i,j}^{v(1)}$ as the arrival rate of flow from i to j, at v; in Scenario 2, those endpoints belonging to set $\mathbb{V}_{CE}^{u(2)/(u,v)}$ are connected to u. Denote $a_{i,j}^{j(2)}$ as the arrival rate of flow from i to j, at u, and $\forall k \in \mathbb{V}_{CE}^{u(2)/(u,v)}$, $a_{i,j}^{u(2)} = d_{k,j}^{k(2)}$, where $\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)} \subseteq \mathbb{V}_{CE}^{u(2)/(u,v)}$, $d_{i,j}^{i(1)} = d_{i,j}^{i(2)}$. If the max-min fair bandwidth allocation scheme is deployed, subject to link capacity L_j^{Out}, in both scenarios, then, $\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}$, $a_{i,j}^{v(1)} \geq a_{i,j}^{v(2)}$.

Proof: Denote $\mathbb{S}$ and $\mathbb{D}$ as the sets of endpoints, where $\forall i \in \mathbb{S}$, $a_{i,j}^v \leq f_j^u \cdot \phi_{i,j}^j$, $\forall k \in \mathbb{D}$,

$a_{k,j}^{v} > f_j^u \cdot \phi_{k,j}^j$.

1) When $\sum_{\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}} a_{i,j}^{u(1)} \leq L_j^{Out}$,

then in Scenario 1, $\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}$, $a_{i,j}^{v(1)} = a_{i,j}^{u(1)} = d_{i,j}^{i(1)}$.

In Scenario 2, $\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}$, $a_{i,j}^{v(2)} \leq d_{i,j}^{i(2)} = d_{i,j}^{i(1)}$.

$\therefore a_{i,j}^{v(1)} \geq a_{i,j}^{v(2)}$.

2) When $\sum_{\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}} a_{i,j}^{u(1)} > L_j^{Out}$,

$\because \mathbb{V}_{CE}^{u(1)/(u,v)} \subseteq \mathbb{V}_{CE}^{u(2)/(u,v)}$ and $\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)} \subseteq \mathbb{V}_{CE}^{u(2)/(u,v)}$, $d_{i,j}^{i(1)} = d_{i,j}^{i(2)}$,

$\therefore \sum_{\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}} a_{i,j}^{u(1)} \geq \sum_{\forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}} a_{i,j}^{u(1)} > L_j^{Out}$

Denote $f_j^{u(1)}$ and $f_j^{u(2)}$ as the fair share rate at the directional link from u to v in Scenarios 1 and 2, respectively.

Then, $\forall i \in \mathbb{S}^{(1)}$, $a_{i,j}^{u(1)} \leq f_j^{u(1)} \cdot \phi_{i,j}^j$; $\forall k \in \mathbb{D}^{(1)}$, $a_{k,j}^{u(1)} > f_j^{u(1)} \cdot \phi_{k,j}^j$, and $\mathbb{V}_{CE}^{u(1)/(u,v)} = \mathbb{S}^{(1)} \cup \mathbb{D}^{(1)}$.

$\forall i \in \mathbb{S}^{(2)}$, $a_{i,j}^{u(2)} \leq f_j^{u(2)} \cdot \phi_{i,j}^j$; $\forall k \in \mathbb{D}^{(2)}$, $a_{k,j}^{u(2)} > f_j^{u(2)} \cdot \phi_{k,j}^j$, and $\mathbb{V}_{CE}^{u(2)/(u,v)} = \mathbb{S}^{(2)} \cup \mathbb{D}^{(2)}$.

Assume that $f_j^{u(1)} < f_j^{u(2)}$, then $\mathbb{S}^{(1)} \subseteq \mathbb{S}^{(2)}$.

Let $\mathbb{M} \subseteq \mathbb{S}^{(2)}$ and $\mathbb{M} \nsubseteq \mathbb{S}^{(1)}$, then $\mathbb{S}^{(2)} = \mathbb{M} \cup \mathbb{S}^{(1)}$, and $\mathbb{D}^{(1)} = \mathbb{M} \cup \mathbb{D}^{(2)}$.

Thus, $f_j^{u(1)} = \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(1)}} a_{i,j}^{u(1)}}{\sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j} = \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j}$, and

$f_j^{u(2)} = \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(2)}} a_{i,j}^{u(2)}}{\sum_{\forall k \in \mathbb{D}^{(2)}} \phi_{k,j}^j} = \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(2)}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{D}^{(2)}} \phi_{k,j}^j} = \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^i - \sum_{\forall i \in \mathbb{M}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j - \sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j}$.

$\because f_j^{u(1)} < f_j^{u(2)}$,

$\therefore \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j} < \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^i - \sum_{\forall i \in \mathbb{M}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j - \sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j}$.

$\therefore L_j^{Out} \cdot \sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j > \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^i \cdot \sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j + \sum_{\forall i \in \mathbb{M}} d_{i,j}^i \cdot \sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j$.

$\therefore \frac{\sum_{\forall i \in \mathbb{M}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j} < \frac{L_j^{Out} - \sum_{\forall i \in \mathbb{S}^{(1)}} d_{i,j}^i}{\sum_{\forall k \in \mathbb{D}^{(1)}} \phi_{k,j}^j} = f_j^{u(1)}$.

$\therefore \sum_{\forall i \in \mathbb{M}} d_{i,j}^i < f_j^{u(1)} \cdot \sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j$.

$\because \mathbb{M} \nsubseteq \mathbb{S}^{(1)}$, then $\mathbb{M} \subseteq \mathbb{D}^{(1)}$,

And $\forall k \in \mathbb{D}^{(1)}$, $d_{k,j}^k = a_{k,j}^{u(1)} > f_j^{u(1)} \cdot \phi_{k,j}^j$, then $\sum_{\forall i \in \mathbb{M}} d_{i,j}^i > f_j^{u(1)} \cdot \sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j$.

Thus, this is contradictory to $\sum_{\forall i \in \mathbb{M}} d_{i,j}^i < f_j^{u(1)} \cdot \sum_{\forall k \in \mathbb{M}} \phi_{k,j}^j$.

Therefore, the assumption $f_j^{u(1)} < f_j^{u(2)}$ cannot hold.

$\therefore f_j^{u(1)} < f_j^{u(2)}$.

And $a_{k,j}^{v(1)} = \min(f_j^{u(1)} \cdot \phi_{k,j}^j, a_{i,j}^{u(2)}) = \min(f_j^{u(1)} \cdot \phi_{k,j}^j, d_{i,j}^i)$

$a_{k,j}^{v(2)} = \min(f_j^{u(2)} \cdot \phi_{k,j}^j, a_{i,j}^{u(2)}) = \min(f_j^{u(2)} \cdot \phi_{k,j}^j, d_{i,j}^i)$,

$\therefore \forall i \in \mathbb{V}_{CE}^{u(1)/(u,v)}$, $a_{k,j}^{v(2)} \leq a_{k,j}^{v(1)}$.

■

3.3 The Idealized Fluid Bandwidth Allocation Scheme

In order to realize the features of the "Superswitch" model, an idealized fair bandwidth allocation scheme is developed as follows:

Algorithm 3-1: $\forall u, v \in \mathbb{V}$, and u and v are connected.

$\forall i \in \mathbb{V}_{CE}^{u/(u,v)}$, $j \in \mathbb{V}_{CE}^{v/(u,v)}$, $a_{i,j}^v = d_{i,j}^u = \min(f_j^u \cdot \phi_{i,j}, a_{i,j}^u)$, where the fair share rate f_j^u can be computed by Algorithm 2-1 in Chapter 2, Fairshare($\{a_{i,j}^v\}, \{\phi_{i,j}\}, L_j^{Out}$).

By Theorems 3-6 and 3-7, the following theorem can be readily obtained.

Theorem 3-8: In a fluid VPN model, which has the same properties as those of hose-modeled VPN. If Algorithm 3-1 is implemented in each intermediate router, then Property 4 of Definition 3-4 can be met and thus this model is a fluid hose-modeled VPN.

It is also necessary to prove that this scheme can be employed without violating the link capacity of each link in the hose-modeled VPN.

Theorem 3-9: In a fluid hose-modeled VPN, $\forall u, v \in \mathbb{V}$, and u and v are connected. If Algorithm 3-1 is employed, then

$$\sum_{j \in \mathbb{V}_{CE}^{v/(u,v)}} \sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} d_{i,j}^u \leq L_{u,v}. \tag{3.11}$$

Proof: In a hose-modeled VPN, $L_{u,v} = \min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u.v)}} L_j^{Out})$.

1) When $L_{u,v} = \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u.v)}} L_j^{Out}$,

if the Algorithm 3-1 is used, $\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} a_{i,j}^v \leq L_j^{Out}$, then

$\sum_{j\in\mathbb{V}_{CE}^{v/(u,v)}}\sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} a_{i,j}^{v} \le \sum_{j\in\mathbb{V}_{CE}^{v/(u,v)}} L_j^{Out} = L_{u,v}$.

$\therefore \sum_{j\in\mathbb{V}_{CE}^{v/(u,v)}}\sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} a_{i,j}^{v} = L_{u,v}$.

2) When $L_{u,v} = \sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}$.

$\because \sum_{\forall j\in\mathbb{V}_{CE}^{v/(u,v)}} d_{i,j}^{i} \le L_i^{In}$ and $d_{i,j}^{i} \ge d_{i,j}^{u}$,

$\therefore \sum_{\forall j\in\mathbb{V}_{CE}^{v/(u,v)}} d_{i,j}^{u} \le \sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} d_{i,j}^{i} \le L_i^{In}$,

$\therefore \sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}}\sum_{\forall j\in\mathbb{V}_{CE}^{v/(u,v)}} d_{i,j}^{u} \le \sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}}\sum_{\forall j\in\mathbb{V}_{CE}^{v/(u,v)}} d_{i,j}^{i}$

$\le \sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}$, and $a_{i,j}^{v} = d_{i,j}^{u}$

$\therefore \sum_{j\in\mathbb{V}_{CE}^{v/(u,v)}}\sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} a_{i,j}^{v} \le \sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}$.

$\therefore \sum_{j\in\mathbb{V}_{CE}^{v/(u,v)}}\sum_{\forall i\in\mathbb{V}_{CE}^{u/(u,v)}} a_{i,j}^{v} = L_{u,v}$.

■

It is clear that if a packet has to be discarded before it arrives its destination, it should be discarded as early as possible, thus avoiding congestion and resource waste in the network. In the proposed fluid bandwidth allocation scheme, knowing the egress link capacity of an endpoint, the intermediate routers are able to drop those packets at the earliest stage, without the knowledge of the downstream traffic pattern. Since ISPs may carry many customers' traffic on a single link, then if some bandwidth from a VPN customer is saved, it can be used to carry best-effort traffic of other customers, because the VPN customers would observe the same result even if those packets are not discarded earlier. In this case, the ISP is able to "steal" some bandwidth from the VPN link without noticeable performance degradation from the VPN customers' perspective. In this sense, the overall throughput of the ISP's network can be improved with the proposed scheme.

3.4 Provisioning and Managing the Hose-modeled VPN

The hose-modeled VPN is suitable for those organizations which require to communicate among headquarters, branch offices, and suppliers. From the perspective of the ISP, this VPN needs a long term connection (weeks, months), which is not highly

dynamic. The provisioning of VPN can be formed as a minimal cost problem for a given network topology, which is discussed in [27]. It can be performed by the ISP's network operation center (NOC).

In order to enable the VPN customers to manage their VPN resources by themselves, a centralized VPN "manager-client" architecture is proposed. The VPN manager works as the VPN operation center, which processes the requirements of the VPN clients and allocates bandwidth to hose flows by assigning different weights to them. Since, from the VPN customers' perspective, the hose-modeled VPN works like a "Superswitch", the customers themselves do not know the VPN's topology and connections in detail. Therefore, it is required to create a connection, such as IPSec connection, between the VPN manager and the ISP's NOC. If the VPN manager changes the weight of a hose flow, it informs the ISP's NOC, and the ISP's NOC informs those corresponding routers. This operation should be performed by hours or days.

3.5 Summary

In this chapter, based on the fluid VPN model and the hose-modeled VPN, a new bufferless, fluid hose-modeled VPN is proposed. It can be viewed by the VPN customers as a "Superswitch". With this model, the VPN customers are able to allocate the bandwidth according to their own requirements by assigning different weights. To meet their requirements, an idealized fluid bandwidth allocation scheme, which is implemented in each intermediate router in the ISP's network, is also proposed. It is proved, analytically, that the proposed scheme is able to 1) allocate bandwidth fairly according to the customers' requirements; 2) achieve the maximum overall VPN throughput; 3) achieve the maximum endpoint-to-endpoint transmission capacity; and 4) improve the overall throughput of the ISP's network.

QoS is a set of measurable metrics which reflect the service quality that the ISPs provide to the customers. The QoS metric include: 1) bandwidth; 2) delay; 3) delay jitter; 4) loss rate. It is clear that the bandwidth is the only concern in a fluid VPN model. With the proposed idealized fluid bandwidth allocation scheme, regardless of the traffic patterns, the bandwidth of hose flow $h_{i,j}$ can be guaranteed, *i.e.*, $g_{i,j} = \phi_{i,j} \cdot L_j^{Out}$, where $g_{i,j} \leq L_i^{In}$. The VPN customers do not have to know the exact traffic pattern of each hose flow; they can just estimate it and assign a corresponding weight to this hose flow. Then, they can obtain a guaranteed QoS in terms of bandwidth. However, there would be contention if all customers increase their weights when they want to have more bandwidth. Therefore, a centralized management mechanism is needed to assign the weight to each source endpoint. Finally, a centralized "server-client" architecture to perform the VPN management is proposed.

The merits of the proposed fluid model can be summarized as follows:

- From the ISP's perspective, they are able to:
  - Meet the requirements of the VPN customers;
  - Possess the desirable flexibility in terms of multiplexing gain;
  - Possess the desirable properties in terms of scalability, routing and restoration due to its tree topology;
  - Improve the overall throughput of the ISP's network;
  - Cost less since the hose-modeled VPN topology is constructed with the least cost in terms of the total reserved bandwidth;
  - Possess the immunity capability to flooding-based DoS or DDoS, since the resources in terms of bandwidth and buffer are partitioned.
- From the VPN customers' perspective, with the simplified "Superswitch" model, they are able to:

- Achieve the maximum overall VPN throughput;
- Manage their VPN resource in terms of bandwidth according to their own requirements without knowing the exact traffic pattern;
- Obtain guaranteed bandwidth of each hose flow regardless of the traffic patterns;
- Maximize the single endpoint-to-endpoint transmission capacity.

Future Work:

Note that, unicast traffics are assumed in this chapter. However, there is an increasing need for applications requiring traffic multicast in the Internet, such as video conferencing and online seminars, Property 2 in Definition 3-4 may not hold, and thus it is necessary to conduct further study on the fluid fair bandwidth allocation scheme for multicast traffics.

CHAPTER 4

IMPLEMENTING THE SCHEDULING SCHEME IN THE HOSE-MODELED VPN

High-speed, service-integrated routers (Ps, PEs) are required to support a large number of sessions with diverse service rate requirements. When multiplexed at the same output of a scheduler, different sessions interact with each other, and therefore scheduling algorithms are used to control the interactions among them. A scheduler decides which session is served, and which packet is transmitted at each moment. Scheduling is a critical task for providing QoS in term of service rate (bandwidth), delay, jitter and loss rate [34, 35].

4.1 Deficit Round Robin Scheme

For the cell-based network, Nagle [36] proposed a scheduling scheme, which is called round robin (RR), by enabling each router to discriminate flows and then providing service to each flow in turn. Flows are identified by their source-destination addresses. Each flow has a separate queue, and each queue is served in a round-robin fashion. This scheme possesses the desirable implementation simplicity, which is $O(1)$. Weighted round robin (WRR) [37] is a variant of the round robin scheme, when each session has a different bandwidth requirement. Deficit round robin scheme (DRR) [38] is a variant of WRR, which is employed in the packet-based network.

4.1.1 The DRR Scheme

DRR still has the same implementation simplicity as that of the RR scheme. The DRR scheme is presented as follows (the notations in [38] have been modified to be consistent with those in this dissertation):

Algorithm 4-1: The Deficit Round Robin Scheme (on the directional link from **u** to **v**)

Initialization Procedure:

for$(i = 0; i < |\mathbb{V}_{CE}^{u/(u,v)}|; i++)$

$DC_{i,j} = 0;$

Enqueuing Procedure: // on arrival of packet p

Hose_Flow(i, j) = $ExtractFlow(p)$

if (ExistsInActiveList(i, j) ==FALSE) **then**

AppendToActiveList(i, j);

$DC_{i,j} = 0;$

if (no_free_buffer_left==TRUE) **then**

Discard_Packet(p);

else

Enqueue(p, Hose_Flow(i, j));

Dequeuing Procedure:

while (ActiveListIsEmpty==FALSE)

$DC_{i,j} = DC_{i,j} + Q_{i,j};$

while $((DC_{i,j} != 0) and (FlowIsIdle(i, j) == FALSE))$

if $(PacketSize(p_{i,j}^{Head}) <= DC_{i,j})$ **then**

Send$(p_{i,j}^{Head})$;

$DC_{i,j} = DC_{i,j}$−PacketSize$(p_{i,j}^{Head})$;

else

Rotate_Hose_Flow(i, j); //skip to the next flow

break; //skip while loop

if $(FlowIsIdle(i, j) == TRUE)$

Dequeue_Flow(i, j);

$DC_{i,j} = 0;$

if $(DC_{i,j} == 0)$

Rotate_Hose_Flow(i, j); //skip to the next flow

The DRR scheme can be summarized as follows:

- When the scheduler is initialized, the quantum for each flow $Q_{i,j}$ is computed according to the required bandwidth. The deficit counter of each flow, $DC_{i,j}$, is set to 0.

- When a new packet arrives, the Enqueueing Procedure is initiated. First, it checks if the corresponding flow is in the active flow list. If not, this flow is appended to the active flow list, which is a linked list. Then, if there are enough buffers, this packet is placed in the corresponding queue; otherwise it is discarded.

- The Dequeuing Procedure always operates at the head of the active flow list (*i.e.*, the first flow in the linked list). When a flow is placed at the head of the active flow list, the flow deficit counter $DC_{i,j}$ is increased by the flow quantum $Q_{i,j}$. When this flow is not idle and $DC_{i,j}$ is not 0, if the size of the first packet in this flow, PacketSize$(p_{i,j}^{Head})$, is no greater than $DC_{i,j}$, this packet is transmitted and $DC_{i,j}$ is decreased by PacketSize$(p_{i,j}^{Head})$; otherwise, the procedure Rotate_Hose_Flow(i, j) dequeues this flow and appends it at the tail of the active flow list, places the next flow at the head, and then goes back to the starting point of the Dequeuing Procedure. Second, if the head flow is idle, the head flow is dequeued from the active flow list, and then the deficit counter $DC_{i,j}$ is set to 0. Finally, if $DC_{i,j}$ is 0, the procedure Rotate_Hose_Flow(i, j) dequeues this flow, appends it at the tail of the active flow list, and places the next flow at the head.

4.1.2 The Computation of the Quantum Assigned to Each Flow

It is known that, in the DRR scheme, the allocated bandwidth of a flow is proportional to its assigned weight. That is, with the notation in this dissertation, at the directional link from node u to v,

$$\forall i,k \in \mathbb{V}_{CE}^{u/(u,v)} \text{ and } \forall j,l \in \mathbb{V}_{CE}^{v/(u,v)}, \frac{\phi_{i,j}^{u}}{\phi_{k,l}^{u}} = \frac{Q_{i,j}^{u}}{Q_{k,l}^{u}}. \tag{4.1}$$

Therefore, the quantum of each flow is defined as proportional to its guaranteed rate on this link, *i.e.*,

$$\forall i,k \in \mathbb{V}_{CE}^{u/(u,v)} \text{ and } \forall j,l \in \mathbb{V}_{CE}^{v/(u,v)}, \frac{g_{i,j}^{u}}{g_{k,l}^{u}} = \frac{Q_{i,j}^{u}}{Q_{k,l}^{u}}. \tag{4.2}$$

It is also known that, to maintain the implementation simplicity of $O(1)$, each flow must be served at least once when it is visited by the scheduler, and thus the following inequality must be met:

$$\forall i \in \mathbb{V}_{CE}^{u/(u,v)} \text{ and } \forall j \in \mathbb{V}_{CE}^{v/(u,v)}, Q_{i,j}^{u} \geq \max(PacketSize). \tag{4.3}$$

In order to minimize the delay bound,

$$\min_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}, \forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q_{i,j}^{u}) = \max(PacketSize). \tag{4.4}$$

The quantum of each flow can be computed by Eqs. 4.2 and 4.4.

4.1.3 The Low-network-utilization Issue Induced by Deploying the DRR Scheme Directly

Although DRR is not the best scheme to approximate the ideal fluid fair bandwidth allocation scheme (some schemes, such as PGPS [39], WFQ [40], WF2Q [41], and

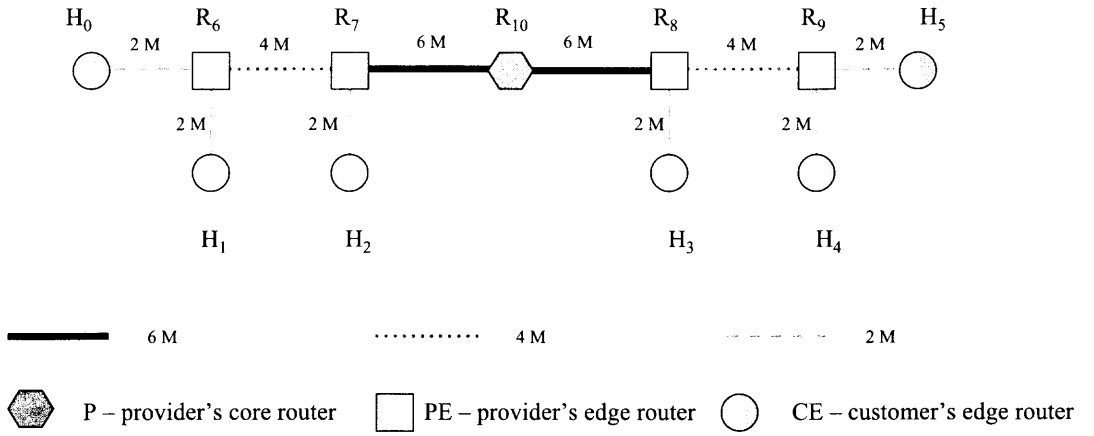

Figure 4.1 A hose-modeled virtual private network

WF2Q+ [42], can approximate it better), presented in Chapter 2, it possesses the desirable implementation simplicity, and thus it is more scalable than other schemes. However, it raises the low utilization issue if the DRR scheme is employed directly in the hose-modeled VPN. This issue is demonstrated by means of the following example.

Fig. 4.1 shows a typical symmetric VPN, in which each endpoint has the hose link capacity of 2 Mbps. This experiment is conducted by NS-2 [43]. Note all loads are CBR traffic.

H_0, H_1, H_2, H_3 and H_4 start to transmit packets to H_5 at time 0, 2, 4, 6, 8, with a constant rate of 0.2 Mbps, 1 Mbps, 1 Mbps, 1 Mbps and 1 Mbps, respectively (*i.e.*, flow 0, 1, 2, 3, 4, respectively). H_0, H_1, H_2, and H_3 start to transmit packets to H_4 at time 0, 2, 4, 6, with a load of 0.2 Mbps, 1 Mbps, 0.4 Mbps and 0.4 Mbps, respectively (*i.e.*, flow 5, 6, 7, 8, respectively). The experiment results are shown in Figs. 4.2 and 4.3. From Fig. 4.3, it is seen that, at H_4, the totally received traffic, from time 6, cannot achieve 2 Mbps, although the link capacity of H_4 is 2 Mbps and there is enough traffic to H_4 (from time 6, the total traffic to H_4 is 2 Mbps). This is attributed to the following: at link $R_8 - R_9$ in the time interval (6,

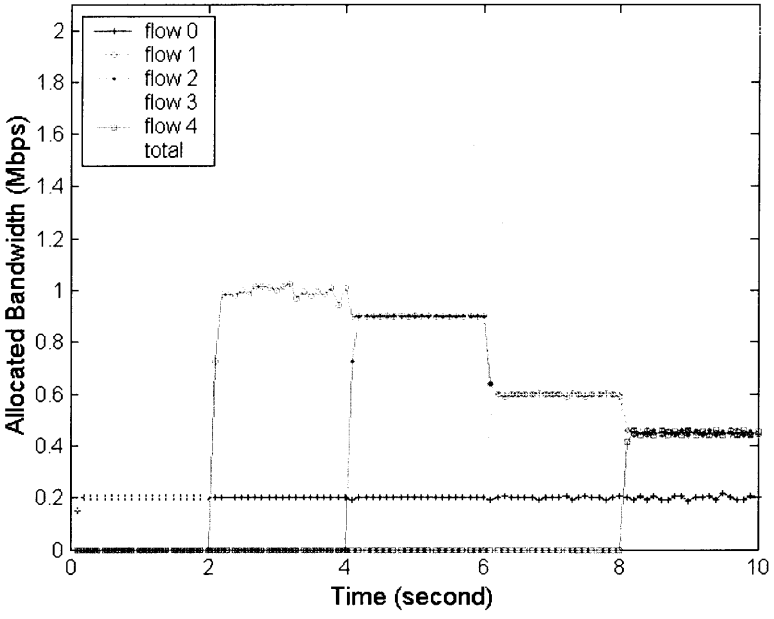

Figure 4.2 The arrival patterns of hose flows at H_5 with DRR

8), there are 8 active flows (flow 0, 1, 2, 3, and 5, 6, 7, 8), the fair share of each flow is (4-0.2-0.2-0.4-0.4)/4= 0.7 Mbps, and so flow 1, 2, 3 and 5 will receive 0.7 Mbps bandwidth. However, since flow 0, 1, 2, 3 are destined to H_5, their total traffic is 2.3 Mbps at link $R_8 - R_9$ because the link to H_5 is 2 Mbps, and therefore the excessive 0.3 Mbps traffic is dropped at link $R_8 - R_9$. At the same time, the totally allocated bandwidth destined to H_4 is only 0.2+0.7+0.4+0.4 = 1.7 Mbps, and again there is enough traffic (2 Mbps) to H_4 and H_4 has enough link capacity (2 Mbps). This happens because the fair bandwidth allocation of a single link could adversely affect the overall throughput. In this experiment, the fair bandwidth allocation of link $R_8 - R_9$ decreases the throughput on link leading to H_4.

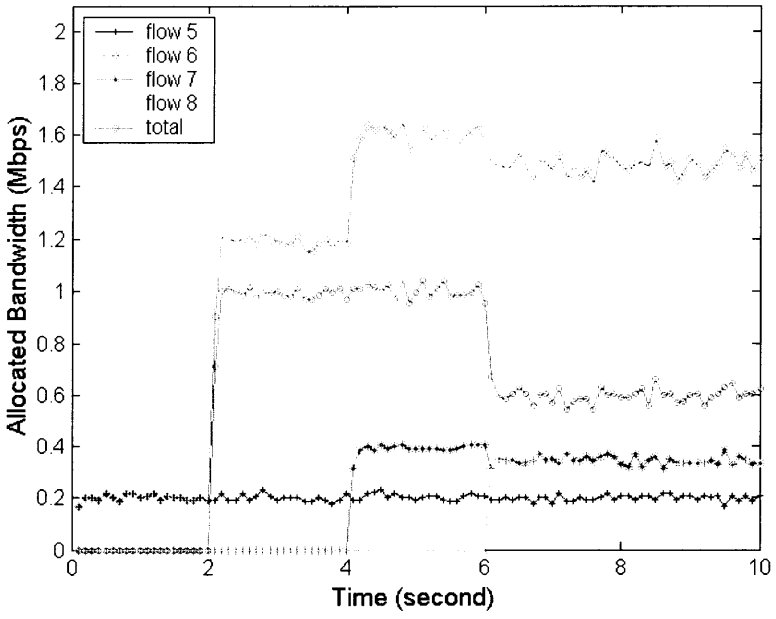

Figure 4.3 The arrival patterns of hose flows at H_4 with DRR

4.1.4 Fair Bandwidth Allocation Scheme with the Feedback Mechanism

Another approach is the max-min bandwidth reallocation scheme with consideration of bottleneck on other links, as described in [33]. Fair share rate can be computed as in the previous chapter, but in Step 2, the minimum allocated bandwidth could be the one bottlenecked at the downstream links. This approach requires a feedback mechanism to inform a switch that if any flow is bottlenecked at its downstream links, *e.g.*, at moment 6s, R_9 should inform other upstream routers, with RM cells [5], that flow 1, 2 and 3 should be constrained, thus saving bandwidth for flow 6. This mechanism requires, at least, traffic measurement and a signaling protocol for dynamically reallocating bandwidth. As it is known, more accurate bandwidth reservation needs more accurate traffic measurement and more signaling overhead, thus consuming CPU processing power and more bandwidth. Another issue is the convergence time. Since the convergence time to reach the desired, stable fair bandwidth allocation is

an increasing function of the round trip time and the hop count, it could take several seconds, which are too slow for the Internet. Furthermore, the Internet traffic is bursty and highly dynamic in nature, the slow convergence time does not allow this approach to track the traffic pattern responsively. Therefore, the effectiveness of the fair bandwidth allocation scheme with the feedback mechanism is questionable.

As compared to the fair bandwidth allocation scheme with the feedback mechanism, the fair scheduling scheme operated at the smallest timescale is able to allocate bandwidth fairly in the smallest time slot.

4.2 Enhanced Round Robin scheme

Having noticed that the "flat" structure of the DRR scheme cannot meet the requirement for flexible bandwidth management, Francini, *et al*, proposed the Enhanced Round Robin scheme (ERR) [44]. The ERR scheme has a hierarchical structure, with two classes: 1) bundle - a subset of flows (aggregation of certain flows), and 2) flow. In order to provide guaranteed bandwidth not only to individual flows, but also to aggregations of flows, the ERR scheme first allocates bandwidth to each bundle, then allocates bandwidth to each flow in each bundle.

4.2.1 The ERR Scheme

The ERR scheme works almost the same as the DRR scheme, which is described in the previous section. From the implementation perspective, there are four major distinctions between the ERR scheme and the DRR scheme. First, the instant quantum of each flow $Q'_{i,j}$ is employed in ERR instead of the quantum of each flow $Q_{i,j}$. The quantum of each flow $Q_{i,j}$ can be computed by Eqs. 4.2 and 4.4. At the beginning of each service frame, the instant quantum of each flow $Q'_{i,j}$ is computed based on the backlogged flows which belong to the same bundle as follows:

$$Q'^{u}_{j} = Q^{u}_{j} \cdot \frac{\sum_{\forall j \in \mathbb{V}^{v/(u,v)}_{CE}} Q^{u}_{j}}{\sum_{\forall j \in \mathbb{V}^{v/(u,v)}_{CE} and\ bundle(j) \in B^{u}} Q^{u}_{j}}, \tag{4.5}$$

where B^u is the set of backlogged bundles at node u, and Q^u_j is the aggregate quantum of bundle destined to node j, *i.e.*,

$$Q^{u}_{j} = \sum_{\forall i \in \mathbb{V}^{u/(u,v)}_{CE}} Q^{u}_{i,j}. \tag{4.6}$$

$$Q'^{u}_{i,j} = Q'^{u}_{i,j} \cdot \frac{\sum_{\forall i \in \mathbb{V}^{u/(u,v)}_{CE}} Q^{u}_{i,j}}{\sum_{\forall i \in \mathbb{V}^{u/(u,v)}_{CE} and\ flow(i,j) \in B^{u}_{j}} Q^{u}_{i,j}}, \tag{4.7}$$

where $B^u(j)$ is the set of backlogged flows, destined to node j, at node u.

Note that the original ERR scheme computes the finish timestamp of each flow and then decides how much service each flow should receive in a single service frame. In order to compare the ERR scheme with the proposed scheme, the newly introduced instant quantum is computed, instead. Actually, the ERR scheme described here is equivalent to the original one.

4.2.2 Issues in the ERR Scheme

The basic principle is that the unused bandwidth of one flow is shared by those backlogged flows in the same bundle. With this scheme, the bandwidth segregation among bundles of flows can be achieved, too. Secondly, the ERR scheme was developed based on GPS and PGPS [39]; it requires, for each flow, to compute the number of bytes (or packets) to be transmitted in a single service frame. Furthermore, since the ERR scheme requires to maintain the information of all flows and bundles - which are backlogged, it is a stateful scheduling scheme. Although, as compared to the implementation of the DRR scheme, the additional cost is the memory space needed to store the state information of all flows and bundles of flows, and some operations that

maintain that information, since it is a stateful scheme, the overall implementation complexity is increased to $O(M \cdot N^2)$, in the PPVPN cases. Therefore, the desirable implementation simplicity of the DRR scheme is degraded when it is modified to the ERR scheme. Thirdly, based on the Surplus Round Robin (SRR) scheme [45], the ERR scheme uses condition $(DC_{i,j} > 0)$, instead of $(PacketSize(p_{i,j}^{Head}) <= DC_{i,j})$, to decide if the packet $p_{i,j}^{Head}$ should be sent. One advantage of the SRR scheme over DRR is that it does not require to know the length of the head-of-the-queue packet to determine if it should be sent. Finally, the other undesirable property is the additional delay of one service frame. With the DRR scheme, a newly backlogged flow can be appended at the tail of the active flow linked list. Stiliadis [46] proved that the delay bound is $\frac{3 \cdot F - 2 \cdot Q_{i,j}^{u}}{L_{u,v}}$, where the service frame size $F = \sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)},\ \forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q_{i,j}^{u})$. With the ERR scheme, the computation of instant quantum is performed at the beginning of each service frame, and a newly backlogged flow can only be appended to the tail of the linked list at the end of each service frame. Thus, if a flow becomes backlogged just at the beginning of a service frame, it has to wait one more service frame to be appended to the linked list. Therefore, the delay bound is increased by $\frac{F}{L_{u,v}}$.

4.3 2-Dimensional Deficit Round Robin

The fair bandwidth allocation scheme in the fluid hosed-modeled VPN is an ideal scheme, which cannot be implemented in the real world. To approximate that scheme, a modified deficit round robin scheme is proposed. It is called 2-dimensional deficit round robin (2-D DRR). The principle of the proposed scheme is, the same as that of the ERR scheme: the unused bandwidth of one flow is shared among those flows destined to the same endpoint, and thus the throughput of the egress link to this destination can be maximized.

4.3.1 The 2-D DRR Scheme

Algorithm 4-2: 2-Dimensional Deficit Round Robin (on the directional link from u to v)

Initialization Procedure:

for $(j = 0; j < |\mathbb{V}_{CE}^{v/(u,v)}|; j++)$

$DC_j = 0;$

for$(i = 0; i < |\mathbb{V}_{CE}^{u/(u,v)}|; i++)$

$DC_{i,j} = 0;$

Enqueuing Procedure: // on arrival of packet p

Hose_Flow$(i, j) = ExtractFlow(p)$

if (ExistsInActiveGroupList(j) ==FALSE) **then**

AppendToActiveGroupList(j);

$DC_j = 0;$

if (ExistsInActiveList(i, j) ==FALSE) **then**

AppendToActiveList(i, j);

$DC_{i,j} = 0;$

if (no_free_buffer_left==TRUE) **then**

Discard_Packet(p);

else

Enqueue(p, Hose_Flow(i, j));

Dequeuing Procedure:

while (ActiveListIsEmpty==FALSE)

while $(DC_j \; != 0)$

if (ExistsInActiveGroupList(j) ==TRUE) **then**

if $(DC_j >= Q_{i,j})$

$DC_{i,j} = DC_{i,j} + Q_{i,j};$

$DC_j = DC_j \; - Q_{i,j};$

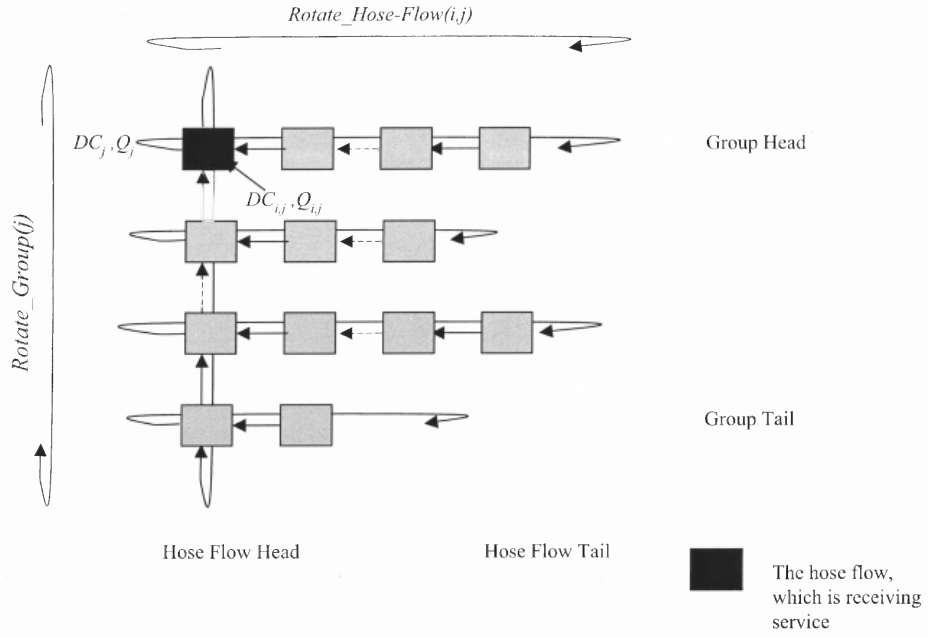

Figure 4.4 The architecture of the 2-D DRR scheme

else

$DC_{i,j} = DC_{i,j} + DC_j$;

$DC_j = 0$;

while (PacketSize($p_{i,j}^{Head}$) <= $DC_{i,j}$)

if(HoseFlowIsNotIdle(i, j)) **then**

Send($p_{i,j}^{Head}$);

$DC_{i,j} = DC_{i,j}$−PacketSize($p_{i,j}^{Head}$);

Rotate_Hose_Flow(i, j);

else

$DC_j = 0$;

Rotate_Group(j);

$DC_j = Q_j$;

The proposed 2-D DRR scheme can be explained by Fig. 4.4. Considering the constraint of the link capacity of all endpoints, different flows are placed into different groups according to their destined endpoints, *i.e.*, flows with the same destination address are placed into the same group. Each group has its own deficit counter and a group quantum. The proposed scheme can be summarized as follows:

- When the VPN is set up, the quantum for each group Q_j in each router is computed according to the link capacity of the corresponding destination, and the quantum of each hose flow, $Q_{i,j}$, is also computed according to the weight of the hose flow. Each group deficit counter DC_j and hose flow deficit counter $DC_{i,j}$ are set to 0.

- When a new packet arrives, the Enqueueing Procedure is initiated. First, it checks if the group, to which this flow belongs, is in the active group list. If not, this group is appended to the tail of the group list. Second, it checks if the corresponding flow is in the active flow list. If not, this flow is appended to the active flow list. Finally, if there are enough buffers, this packet is placed in the corresponding queue; otherwise it is discarded.

- The Dequeuing Procedure always operates at the head flow (*i.e.*, the first flow of the group) in the head group, *e.g.*, flow (i, j). When a group is rotated to the head, the group counter DC_j is set to the group quantum Q_j. First, if the group counter DC_j is no less than the flow quantum $Q_{i,j}$, the flow deficit counter $DC_{i,j}$ is increased by the flow quantum $Q_{i,j}$, and the group deficit counter DC_j is decreased by $Q_{i,j}$; otherwise, the $DC_{i,j}$ is increased by DC_j, and DC_j is set to 0. Second, if the size of the first packet in this flow, PacketSize($p_{i,j}^{Head}$), is no greater than $DC_{i,j}$, this packet is transmitted and $DC_{i,j}$ is decreased by PacketSize($p_{i,j}^{Head}$). This step is repeated until the queue for this hose flow is empty or the deficit counter is smaller than the size of the first packet in this

queue. Third, procedure Rotate_Hose_Flow(i, j) dequeues this flow and appends it to the tail of the group, and then places the next flow at the head. Finally, if the group deficit counter DC_j is not 0, the procedure goes back to the first step; otherwise, procedure Rotate_Group(j) takes this group from the head and appends it to the tail of the group list, and then places the next group at the head. Note, when a flow or a group becomes idle, it is dequeued from the corresponding active list.

$Q_{i,j}$, the quantum of hose flow $H_{i,j}$, is proportional to its guaranteed bandwidth $g_{i,j}$, where $g_{i,j} = \phi^j_{i,j} \cdot L^{Out}_j$. It can be computed by Eqs. 4.4 and 4.2. The group quantum Q_j can be computed by Eq. 4.6.

4.3.2 Properties of the 2-D DRR Scheme

The 2-D DRR scheme possesses the following advantages over the ERR scheme:

1. Desirable implementation simplicity $O(1)$ for scheduling, the same as that of the DRR scheme, since there is no need to maintain information of each flow.

2. Less overhead for CPU since it only computes the quantum of each flow when the flow is initiated. There is no need to compute the instant quantum at the beginning of each service frame.

3. It requires no space overhead to store the information of which flows or aggregations of flows are backlogged.

4. Delay bound $\frac{3 \cdot F - 2 \cdot Q^u_{i,j}}{L_{u,v}}$, which is the same as that of the DRR scheme, since a newly backlogged flow can be appended at the tail of the linked list without waiting till the end of the service frame.

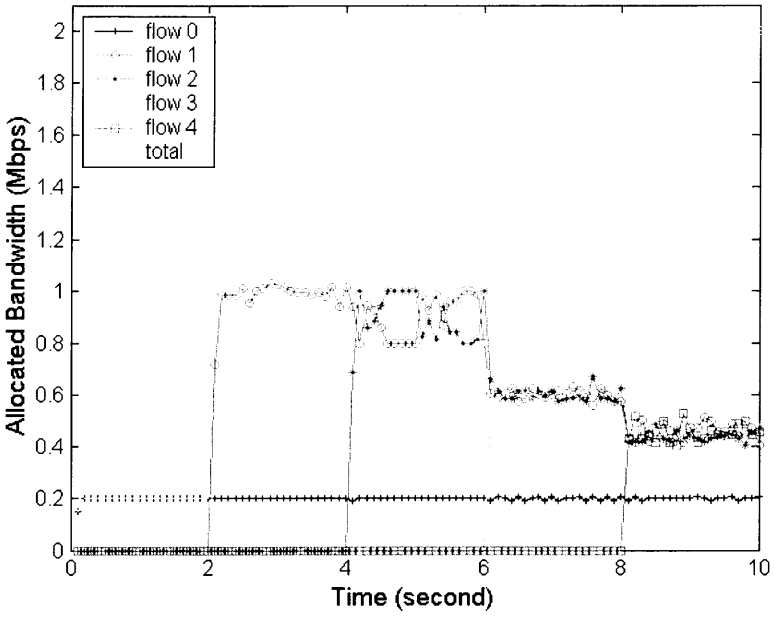

Figure 4.5 The arrival patterns of hose flows at H_5 with 2-D DRR

4.3.3 Simulation Results

The 2-D DRR scheme is implemented by NS-2 [43] for the VPN shown in Fig. 4.1. With the same traffic pattern from sources described in the previous section, it is defined that each flow with a weight proportional to its ingress link capacity, *i.e.*, each flow has the same weight. The arrival traffic patterns of flows 0-4 at H_5, shown in Fig. 4.5, are almost the same as those in Fig. 4.2, and the patterns of flows 5-8 at H_4 are shown in Fig. 4.6. Note that the arrival rates of flows 5, 7 and 8 are almost the same as those in Fig. 4.3, whereas the arrival rate of flow 6 is 1 Mbps from time 2s, which is desirable. Therefore, the overall throughput of VPN is improved from 3.7 Mbps to 4.0 Mbps from time 6.

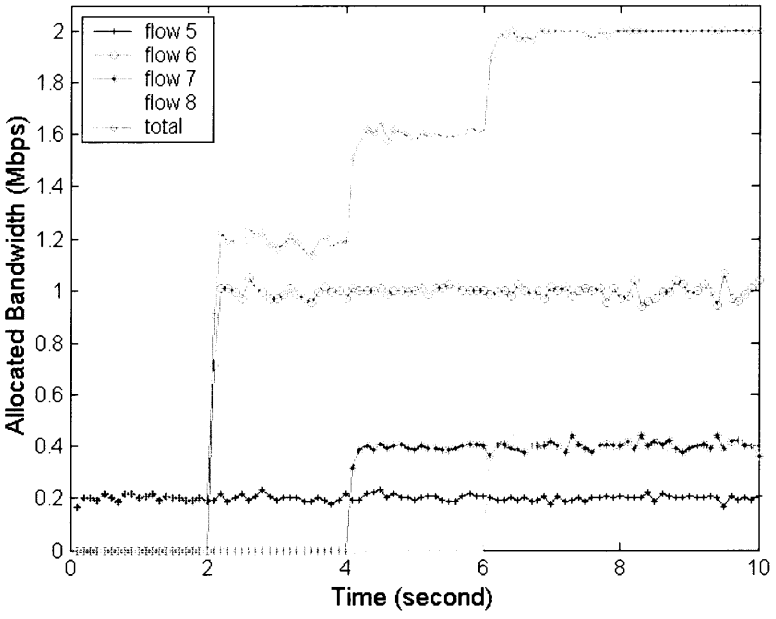

Figure 4.6 The arrival patterns of hose flows at H_4 with 2-D DRR

4.4 Improving the Performance of the 2-D DRR Scheme

The prototype 2-D DRR scheme could raise the issue of burstiness, which is proven to be harmful for those connection-oriented traffics with a feedback mechanism, such as TCP traffics, thus leading to adverse impact on the goodput of these traffics [47, 48, 49].

4.4.1 The Burstiness Issue of the 2-D DRR Scheme

By means of the following example, the burstiness issue introduced by the 2-D DRR scheme is demonstrated.

As shown in Fig. 4.7, suppose each hose link capacity is 4 Mbps, and the capacity of the intermediate links is 20 Mbps. Assume, starting at time 0, all traffics are TCP packets with 1500 bytes, and the arrival rates are as follows: $a_{0,5} = 4Mbps$, $a_{i,j} = 1Mbps$, where $i = 1, 2, 3, 4$, $j = 6, 7, 8, 9$ and other hose flows are idle. By

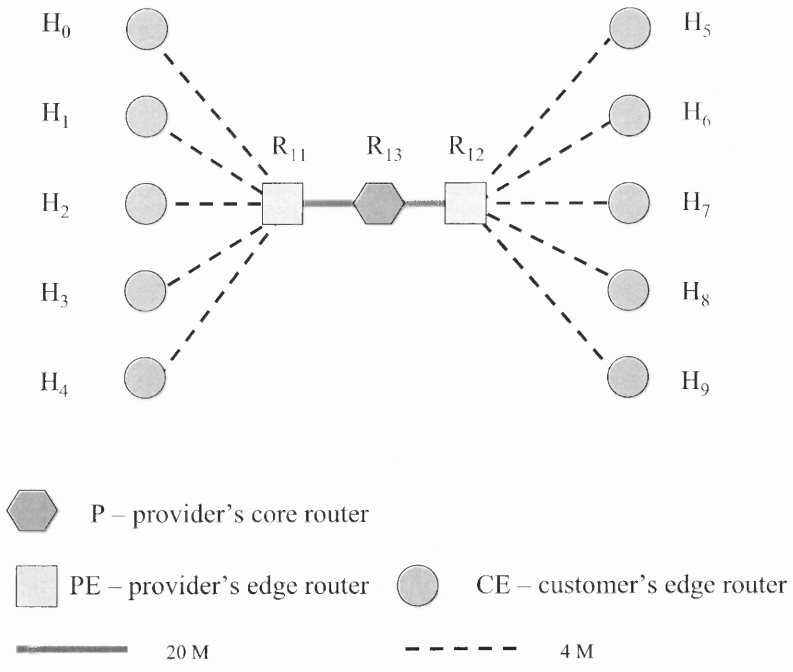

Figure 4.7 An example of a simple hose-modeled VPN

Eq. 4.4, $\min_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)},\ \forall j \in \mathbb{V}_{CE}^{v/(u,v)}}(Q_{i,j}^{u}) = 1500$ bytes. By Eqs. 4.2, $Q_{i,j} = 1500$ bytes. As shown in Fig. 4.8, each flow is served in a round robin fashion (the "flat" DRR structure). The service order is shown in Fig. 4.9.

When the 2-D DRR scheme is employed, then by Eq. 4.6, $Q_j = 7500$ bytes, where $j = 5, 6, 7, 8, 9$. The "tiered" structure of the 2-D DRR scheme employed in the example of the hose-modeled VPN is shown in Fig. 4.10. According to the 2-D DRR scheme described in the previous section, the service order is shown in Fig. 4.11.

Hose flow $h_{0,5}$ receives service at the beginning, and five packets are transmitted. After waiting for 20 packets of other hose flows being sent, hose flow $h_{0,5}$ can transmit five packets again. Obviously, it is a periodic process. The burstiness of $h_{0,5}$ can be observed. As discussed in [48, 47], an intensive burstiness of TCP traffic would make the sender to adjust the TCP congestion window size, thus making the window size oscillate greatly and reducing the goodput of this TCP connection.

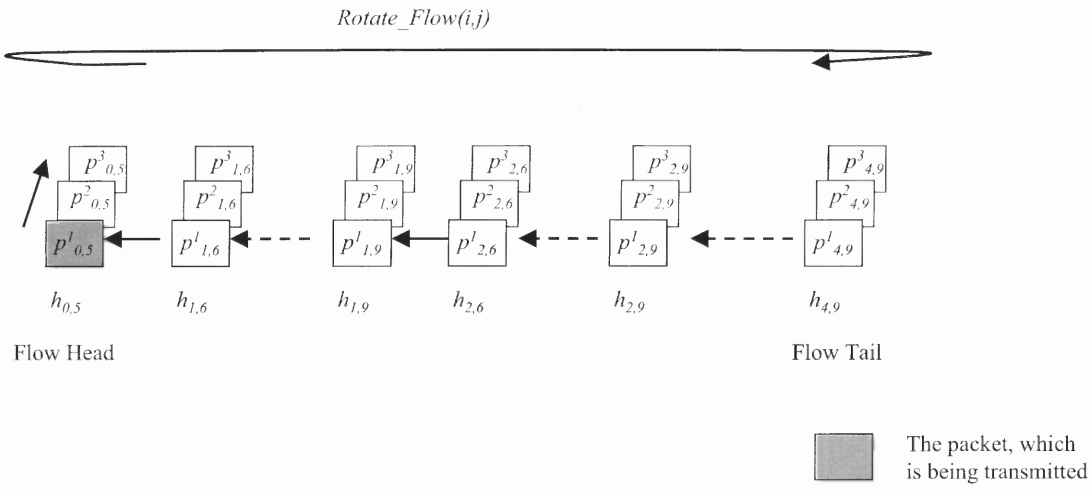

Figure 4.8 The DRR scheme when employed in the example of the hose-modeled VPN

4.4.2 The Computation of the Group Quantum

To alleviate the burstiness in the previous scenario, each group quantum should be minimized, thus reducing the "batch" size of traffic destined to the same endpoint. By Eqs. 4.4, 4.2 and 4.6, the group quantum Q_j can be computed. Now, select

$$\min_{\forall k \in \mathbb{V}_{CE}^{v/(u,v)}} Q_k^{'} = \max(PacketSize). \tag{4.8}$$

The minimized group quantum $Q_j^{'}$ can be computed as:

$$Q_j^{'} = Q_j \cdot \frac{\max(PacketSize)}{\min_{\forall k \in \mathbb{V}_{CE}^{v/(u,v)}} Q_k^{'}}. \tag{4.9}$$

Note that, in a hose-modeled VPN, with the idealized fluid bandwidth allocation scheme proposed in Chapter 3, $\forall j \in \mathbb{V}_{CE}^{v/(u,v)}, \sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} d_{i,j}^{u} \leq \min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, L_j^{Out})$. Therefore, the group quantum can be defined as:

$$\forall k, j \in \mathbb{V}_{CE}^{v/(u,v)}, \frac{Q_j^{'}}{Q_k^{'}} = \frac{\min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, L_j^{Out})}{\min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, L_k^{Out})}. \tag{4.10}$$

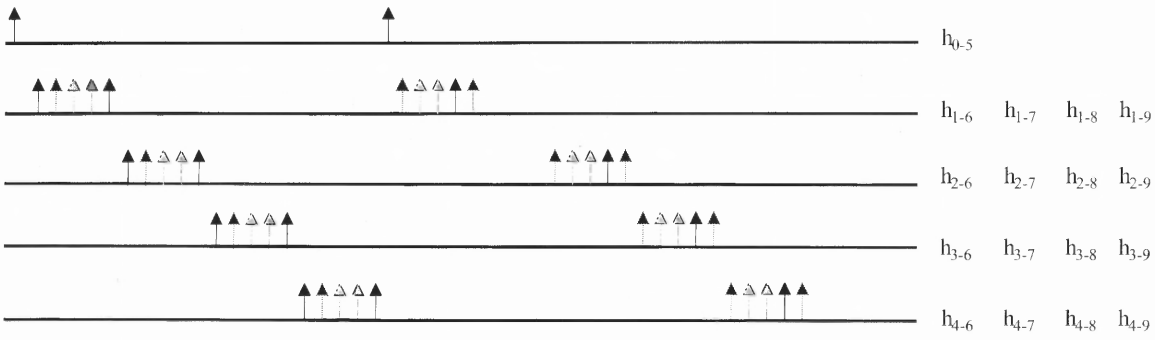

Figure 4.9 Service order of all hose flows in R_{12} when the DRR scheme is employed

As it is known, the quantum assigned to a group is proportional to the bandwidth allocated to the aggregated hose flows to the corresponding endpoint. Eq. 4.10 does not change the allocated bandwidth relationship among all groups. In the proposed scheme, Eq. 4.1 also holds, and therefore the allocated bandwidth relationship among all flows in one group remains the same as in the DRR scheme. Thus, this approach provides guaranteed bandwidth not only to aggregations of flows, but also to individual flows. Since the group quantum is reduced, the cycle time of group rotating will be decreased. However, in one cycle of group rotating, some backlogged hose flows may not receive service, contrary to the one when Eq. 4.2 is employed.

4.4.3 The 2-D DRR+ Scheme

With the newly computed group quantum, the modified 2-D DRR, which is called 2-D DRR+, is shown as follows:

Algorithm 4-3: 2-Dimensional Deficit Round Robin Plus (on the directional link from **u** to **v**)

Initialization Procedure:

for$(j = 0; j < |\mathbb{V}_{CE}^{v/(u,v)}|; j++)$

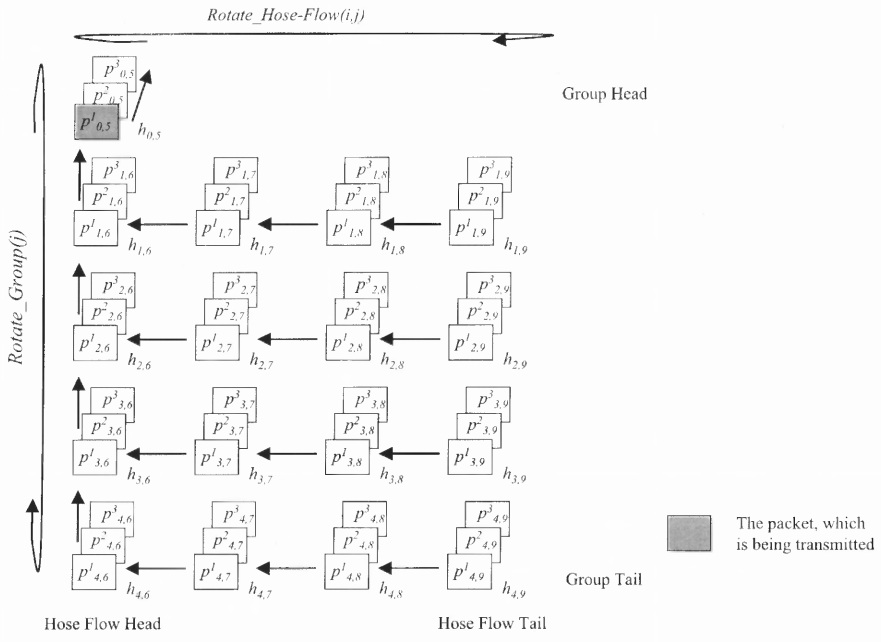

Figure 4.10 The 2-D DRR scheme when employed in the example of the hose-modeled VPN

$DC_j = 0;$

for$(i = 0; i < |\mathbb{V}_{CE}^{u/(u,v)}|; i++)$

$DC_{i,j} = 0;$

Enqueuing Procedure: // on arrival of packet p

Hose_Flow(i, j) = $ExtractFlow(p)$

if (ExistsInActiveGroupList(j) ==FALSE) **then**

AppendToActiveGroupList(j);

$DC_j = 0;$

if (ExistsInActiveList(i, j) ==FALSE) **then**

AppendToActiveList(i, j);

$DC_{i,j} = 0;$

if (no_free_buffer_left==TRUE) **then**

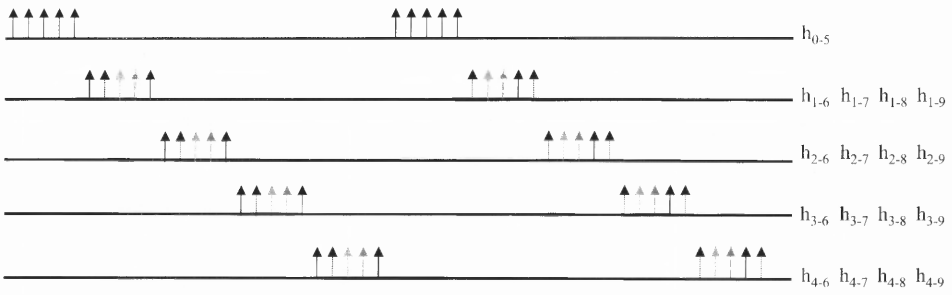

Figure 4.11 Service order of all hose flows in router R_{12}, when the 2-D DRR scheme is employed

Discard_Packet(p);

else

Enqueue(p, Hose_Flow(i, j));

Dequeuing Procedure:

while (ActiveListIsEmpty== $FALSE$)

while (ExistsInActiveGroupList(j) == $TRUE$)

while (($p_{i,j}^{Head} <= DC_{i,j}$)$and$($p_{i,j}^{Head} <= DC_j$)

and(($HoseFlowIsNotIdle(i,j)$) == $TRUE$))

Send($p_{i,j}^{Head}$);

$DC_{i,j} = DC_{i,j}$−PacketSize($p_{i,j}^{Head}$);

$DC_j = DC_j$−PacketSize($p_{i,j}^{Head}$);

if ($DC_j < p_{i,j}^{Head}$) **then**

Rotate_Group(j);

$DC_j = DC_j + Q_j^{'}$;

else

if ($HoseFlowIsNotIdle(i,j)$ == $FALSE$) **then**

$DC_{i,j} = 0$;

$Rotate\_Hose\_Flow(i,j)$;

$$DC_{i,j} = DC_{i,j} + Q_{i,j};$$

The Initialization and Enqueueing Procedures in 2-D DRR+ work the same as those in the 2-D DRR scheme. The distinction between 2-D DRR and 2-D DRR+ comes from the Dequeueing Procedure. The Dequeuing Procedure also always operates at the head flow (*i.e.*, the first flow of the group) in the head group, *e.g.*, flow (i, j). It decides to serve a flow according to both the group deficit counter DC_j and the hose flow counter $DC_{i,j}$, instead of only the hose flow counter $DC_{i,j}$ in the 2-D DRR scheme. When the group deficit counter DC_j and the hose flow counter $DC_{i,j}$ are both greater than the size of the packet at the head, this packet is transmitted, and at the same time both counters DC_j and $DC_{i,j}$ are deducted by the size of this packet; if the group deficit counter DC_j is less than the size of the packet at the head, this group is placed at the end of the linked list of the active group, and the next group is placed at the head of this linked list and its group counter DC_j is increased by its minimized group quantum Q'_j; otherwise, if this flow is idle, reset the deficit counter of this flow to 0 and place the next flow at the head of the linked list of the active flows and at the same time increase its deficit counter $DC_{i,j}$ by its quantum $Q_{i,j}$.

Note that, in a single service frame, each flow is served at least once by the scheduler in both 2-D DRR and 2-D DRR+ schemes; the major difference between 2-D DRR and 2-D DRR+ is as follows:

- In 2-D DRR, in a single service frame, each group is visited by the scheduler only once. Therefore, the unused bandwidth shared by the flows in the same group could lead to burstiness.

- In 2-D DRR+, in a single service, each group may be visited by the scheduler more than once, depending on its group quantum. Thus, the burstiness caused by the 2-D DRR scheme could be reduced.

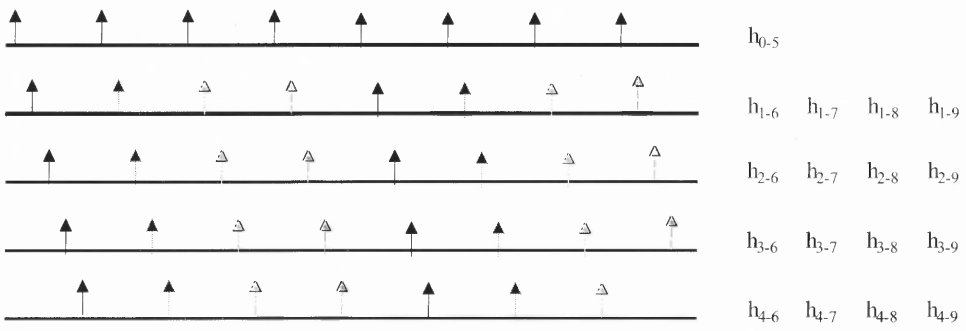

Figure 4.12 Service order of all hose flows at router R_{12}, when the 2-D DRR+ scheme is employed

For the example shown in Fig. 4.7, $Q'_j = 1500$ bytes for $j = 5, 6, 7, 8, 9$. The "tiered" structure of the 2-D DRR remains the same as in Fig. 4.10. According to the 2-D DRR+ scheme, the service order shown in Fig. 4.12 is the ideal service order that is desirable.

Note that, 2-D DRR+ scheme can only alleviate the bustiness of the traffic profile, but it cannot guarantee to provide a service order in which each hose-flow is distributed evenly for any hose-flow bandwidth requirement and egress link capacity.

4.4.4 Latency Analysis

In the previous section, it is demonstrated that the latency in the 2-D DRR scheme is the same as that of the DRR scheme, which is $\frac{3 \cdot F - 2 \cdot Q^u_{i,j}}{L_{u,v}}$, where the service frame size $F = \sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)},\ \forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q^u_{i,j})$ and in each service frame, any backlogged flow must be served at least once. In the 2-D DRR+ scheme, the scheduler decides to serve a flow according to both flow deficit counter and group deficit counter, which are determined by the flow quantum and the minimized group quantum.

Let $\theta = \frac{Q_k}{Q'_k}$.

$\because \forall j \in \mathbb{V}_{CE}^{v/(u,v)}$ and $\forall i \in \mathbb{V}_{CE}^{u/(u,v)}, Q_{i,k} \geq \max(PacketSize)$

$\therefore Q_k \geq \max(PacketSize)$.

$\because \min_{\forall k \in \mathbb{V}_{CE}^{v/(u,v)}} Q_k^{'} = \max(PacketSize)$,

$\therefore \theta \geq 1$.

When a group is visited once, its deficit counter is increased by $Q_k^{'}$. If each flow is served at least once, its deficit counter must be increased by at least its corresponding quantum $Q_{i,k}$, and thus the corresponding group is visited by $\lceil \theta \rceil$ times. Therefore, the overall service provided by the scheduler, in $\lceil \theta \rceil$ times of visit of each group, is $\lceil \theta \rceil \cdot \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} Q_j^{'}$, *i.e.*,

$\lceil \theta \rceil \cdot \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} Q_j^{'} \geq \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q_j^u) = \sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)},\ \forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q_{i,j}^u)$.

$\because \theta \geq 1, \therefore \lceil \theta \rceil < 2\theta$.

$\therefore \lceil \theta \rceil \cdot \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} Q_j^{'} < 2 \cdot \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q_j^u) = 2 \cdot \sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)},\ \forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q_{i,j}^u)$.

Thus, to guarantee that each flow must be served at least once, no more than one additional frame size of service is needed, where the frame size $F = \sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)},\ \forall j \in \mathbb{V}_{CE}^{v/(u,v)}} (Q_{i,j}^u)$, *i.e.*, which is the same as that of the ERR scheme. Therefore, the latency of the 2-D DRR scheme is less than that of the ERR scheme, which is $\frac{4 \cdot F - 2 \cdot Q_{i,j}^u}{L_{u,v}}$. When $\theta = 1$, the 2-D DRR+ scheme is completely the same as the 2-D DRR scheme, and thus the latency is the same as that of the 2-D DRR scheme, which is $\frac{3 \cdot F - 2 \cdot Q_{i,j}^u}{L_{u,v}}$. In fact, the latency is a function of θ, and its value is between $\frac{3 \cdot F - 2 \cdot Q_{i,j}^u}{L_{u,v}}$ and $\frac{4 \cdot F - 2 \cdot Q_{i,j}^u}{L_{u,v}}$.

4.5 Integrating with the Best-effort Traffics

ISPs provide access services as well as VPN services. Those traffics requiring access services are best-effort traffics, which do not require guaranteed or predictable QoS, or even traffic isolation. Thus, the following assumptions are made:

- There are two classes of traffics: VPN traffics and best-effort traffics.
- In VPN traffics, there are two subclasses: guaranteed traffics and non-guaranteed traffics.

- The VPN structures are long term connections, *i.e.*, for weeks or months, among headquarters, branch offices, and partners.

4.5.1 The Computation of Slot Quantum

Denote the physical link capacity of directional link from u to v as $\pounds_{u,v}$, and the bandwidth reserved for the best effort traffic as $L_{u,v}^{BE}$. As it is known, the bandwidth reserved for the VPN is $L_{u,v} = \min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} L_j^{Out})$. Thus:

$$\pounds_{u,v} = L_{u,v} + L_{u,v}^{BE}. \tag{4.11}$$

Therefore, the quantum for the best effort traffic, $Q_{BE}^{'}$, on the directional link from u to v, can be computed as follows:

$$\forall j \in \mathbb{V}_{CE}^{v/(u,v)}, \frac{Q_{BE}^{'}}{Q_j^{'}} = \frac{L_{u,v}^{BE}}{\min(\sum_{\forall i \in \mathbb{V}_{CE}^{u/(u,v)}} L_i^{In}, L_j^{Out})}. \tag{4.12}$$

In order of reduce the burstiness introduced by the "batch" of group quantum, the following approach is proposed:

1. Divide one single service frame into K quantum slots, where $K = \left\lfloor \frac{Q_{BE}^{'}}{\max(PacketSize)} \right\rfloor + \sum_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} \left\lfloor \frac{Q_j^{'}}{\max(PacketSize)} \right\rfloor$. Denote the quantum of slot k as $Q(k)$, and let $\lambda_j = Q_j^{'}$, and $\lambda_{N+1} = \lambda_{BE} = Q_{BE}^{'}$;

2. start at slot 1;

3. assign this slot to group l, if $\lambda_l = \max(\max_{\forall j \in \mathbb{V}_{CE}^{v/(u,v)}} \lambda_j, \lambda_{N+1})$;

4. let the quantum of this slot $Q(k) = \lambda_l$, if $\lambda_l < 2 \cdot \max(PacketSize)$; $Q(k) = \max(PacketSize)$, otherwise;

5. $\lambda_l = \lambda_l - Q(k)$;

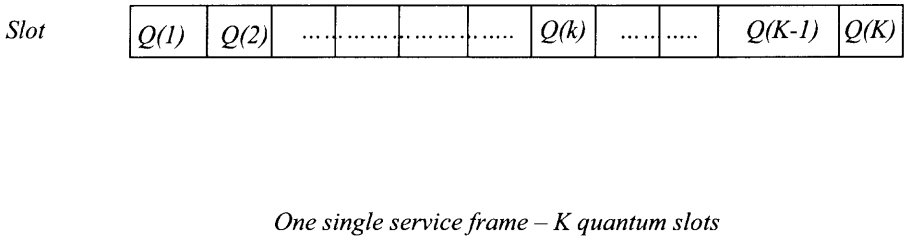

Figure 4.13 The architecture of the 2-D DRR+ scheme integrating with the best effort traffics

6. go to the next quantum slot, repeat Step 3 until all K slots are filled.

4.5.2 The Proposed Scheme Integrating with the Best Effort Traffics

The architecture of the proposed scheme is shown in Fig. 4.13. Each flow (note that all best effort traffics are treated as a single flow) is served in a 2-dimensional round robin fashion.

Note that the size of each quantum, $\forall 1 \leq k \leq K, \max(PacketSize) \leq Q(k) < 2 \cdot \max(PacketSize)$. There are two reasons to select $Q(k)$ as above: 1) to maintain the implementation complexity of scheduling as $O(1)$; when a flow is visited by the scheduler, it must be served if it is backlogged, and thus $\max(PacketSize) \leq Q(k)$; 2) to reduce the potential burstiness of a flow; the "batch" size of a quantum should be reduced, and thus let $Q(k) < 2 \cdot \max(PacketSize)$. Since Eqs. 4.10 and 4.1 still hold, the proposed scheme provides guaranteed bandwidth for the aggregation of flows (group) and individual flows as well.

Algorithm 4-4: 2-Dimensional Deficit Round Robin Plus Scheme Integrating with the Best Effort Traffic (on the directional link from **u** to **v**)

Initialization Procedure:

for$(j = 0; j < |\mathbb{V}_{CE}^{v/(u,v)}|; j++)$

$DC_j = 0;$

for$(i = 0; i < |\mathbb{V}_{CE}^{u/(u,v)}|; i++)$

$DC_{i,j} = 0;$

$DC_{N+1} = 0;$

Enqueuing Procedure: // on arrival of packet p

if $(p \subseteq BestEffortTraffic)$ **then**

AppendToActiveGroupList$(N + 1)$;

else Hose_Flow$(i, j) = ExtractFlow(p)$

if (ExistsInActiveGroupList(j) ==FALSE) **then**

AppendToActiveGroupList(j);

$DC_j = 0;$

if (ExistsInActiveList(i, j) ==FALSE) **then**

AppendToActiveList(i, j);

$DC_{i,j} = 0;$

if (no_free_buffer_left==TRUE) **then**

Discard_Packet(p);

else

Enqueue(p, Hose_Flow(i, j));

Dequeuing Procedure:

while (ActiveListIsEmpty== $FALSE$)

for $(k = 1; k < K + 1; k++)$

if$((Slot(k)\_IsAssignedTo\_VPNTraffic == TRUE)$

and(ExistsInActiveGroupList$(j) == TRUE)$) **then**

$DC_j = DC_j + Q(k);$

while $((p_{i,j}^{Head} <= DC_{i,j}) and (p_{i,j}^{Head} <= DC_j)$

$and((HoseFlowIsNotIdle(i,j)) == TRUE))$

Send($p_{i,j}^{Head}$);

$DC_{i,j} = DC_{i,j}$−PacketSize($p_{i,j}^{Head}$);

$DC_j = DC_j$−PacketSize($p_{i,j}^{Head}$);

if $(DC_j < p_{i,j}^{Head})$ **then**

Rotate_Group(j);

else

if $(HoseFlowIsNotIdle(i,j) == FALSE)$ **then**

$DC_{i,j} = 0$;

$Rotate\_Hose\_Flow(i,j)$;

$DC_{i,j} = DC_{i,j} + Q_{i,j}$;

else

$DC_{N+1} = DC_{N+1} + Q(k)$;

while $((p_{N+1}^{Head} <= DC_{N+1})$

$and((HoseFlowIsNotIdle(N+1)) == TRUE))$

Send(p_{N+1}^{Head});

$DC_{N+1} = DC_{N+1}$−PacketSize(p_{N+1}^{Head});

if $(HoseFlowIsNotIdle(N+1)) == FALSE)$

$DC_{N+1} = 0$;

The Initialization Procedure works almost the same as that of the 2-D DRR scheme, with the exception of resetting the deficit counter, DC_{N+1}, for the best effort traffic to 0. The Enqueueing Procedure also works the same as that of the 2-D DRR scheme, except that if the newly arrived packet belongs to the best effort traffic, it should be placed at the tail of the FIFO queue for the best effort traffics. The Dequeueing Procedure visits the K quantum slots repeatedly in a round robin fashion. When it visits slot k, group j receives service if this slot is assigned to group j. In each group of the VPN traffic, each flow is served in a round robin fashion; if a

flow becomes idle, the unused quantum will be allocated to other flows in the same group; if this group becomes idle, the unused quantum will be allocated to the flows of the best effort traffic. When the Dequeueing Procedure visits slot k, which is assigned to group $N+1$, *i.e.*, the best effort traffic, or, slot k is assigned to a VPN group which is idle, the best effort traffic will receive service in a "first-come-first-serve" fashion.

The proposed 2-D DRR+ scheme possesses the following properties:

- implementation complexity of scheduling - $O(1)$
- reduced burstiness due to minimized big "batch" size

4.6 Further Discussion

To implement the proposed scheme in the routers in the ISP's network, it is assumed that the ISPs provide two kinds of services, the VPN service and the best-effort service. All packets belonging to the best-effort service are placed in a single FIFO queue; other packets belonging to the VPN service are placed in queues per hose-flow based. In order to improve the overall throughput of the ISPs' network, it is proposed to assign the unused quantum of each group, which represents the bandwidth allocated to an endpoint, to the queue for the best-effort traffic.

4.6.1 Implementing the Proposed Scheme within the Virtual Router VPN Framework

In the architecture of a VPN topology with virtual routers, the virtual routers are the logical atomic element for constructing the VPN topology [9]. Since a virtual router is the functional equivalent to a standalone router device, it can be used in a software-controlled manner to construct separate virtual private networks mapped across an ISP's backbone network. The PE routers encapsulate IP packets, which come from the CE routers, with their VPN information in the new IP header. The ISP's core routers forward packets according to the information containing the new

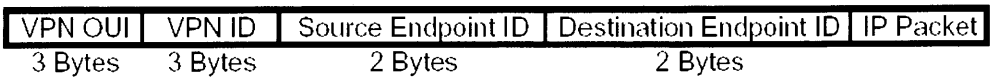

Figure 4.14 The format of packet forwarded in the ISPs' network

header and destination address in the original IP header. When packets arrive at the destination PE routers, the encapsulated IP header is stripped, and forwarded according to the destination address in the original IP header. Note that, in each ISP's core router, a routing table is maintained and updated for each VPN.

With the current IP VPN architecture using virtual routers, based on [50], it is proposed to encapsulate the identification number of the source (starting) endpoint and destination (ending) endpoint into the IP packet, just after the VPN identification number, as shown in Fig. 4.14. Two bytes are used to represent the endpoint ID. Considering the increasing need for multicast applications, define the first bit (MSB) to represent: 1 as point-to-point transmission and 0 as multicast. If the first bit (MSB) is 0, the rest of the bits represents the ID of the multicast group. Therefore, totally 32768 endpoints and 32768 multicast groups can be supported in a single VPN. However, in the Source Endpoint ID segment, only an endpoint ID can be used.

Note that OUI represents IEEE 802-1990 Organizationally Unique Identifier. It is also proposed to create a table for each VPN at the output side of all routers, and based on this table, the classifier at the output side of each router places each packet into its corresponding queue.

Actually, the following addresses are reserved by the Internet Assigned Numbers Authority (IANA) [2] for the private networks [3]: 1) 10.0.0.0 - 10.255.255.255, 2) 172.16.0.0 - 172.31.255.255, and 3) 192.168.0.0 - 192.168.255.255. With proper design and planning, the subnet address, such as 10.0.1.0, 172.17.0.0 and 192.168.2.0, can be used to substitute endpoint ID, thus reducing the corresponding overhead. Therefore,

the proposed scheduling scheme can be integrated seamlessly with the virtual router framework.

For example, in a virtual private network as shown in Fig. 2.3, define the IP addresses of all devices in Boston office as 192.168.3.0/24 and devices in New York office as 172.26.0.0/16. All packets originated from the New York office and destined to the Boston office have the source address 172.26.x.x and destination address 192.168.3.x. Therefore, the group (aggregation) of flows destined to Boston can be discerned with destination address 192.168.3.0/24, and the hose flow from New York to Boston can be discerned with the 2-tuple of (172.16.0.0/16, 192.168.3.0/24) as (source, destination). Therefore, using this approach, the overhead of the Source Endpoint ID and Destination Endpoint ID in Fig. 4.14 can be reduced.

4.6.2 Call Admission Control

Without a proper call admission control scheme, the guaranteed or predictable QoS cannot be provided. In this dissertation, it is assumed that the QoS metrics include two parameters: 1) bandwidth, and 2) latency.

By Eqs. 4.1 and 4.9, the bandwidth of each hose flow and the aggregate bandwidth of each group can be guaranteed in each intermediate node in the ISP's domain. The latency of each hose flow, in each intermediate node, is not greater than $\frac{4 \cdot F - 2 \cdot Q_{i,j}^{u}}{L_{u,v}}$. With these and given propagation delays, the call admission control can be readily performed.

4.6.3 Other Applications

Although the 2-D DRR and 2-D DRR+ schemes are developed to approximate the idealized fluid bandwidth allocation scheme for the hose-modeled VPN, they can be deployed in many other applications, which require a hierarchical scheduling scheme. In these applications, according to different service requirements, all flows are placed

into different sets and served accordingly. Hierarchical general processor sharing (H-GPS) server is proposed in [42]. As extended versions of the DRR scheme, 2-D DRR and 2-D DRR+ can be extended, to approximate P-GPS, to n-D DRR and n-D DRR+, while maintaining the desirable implementation simplicity. Some examples are described as follows:

- Company A and B share the same link connected to the Internet. Company A and B reserve 75% and 25% bandwidth of the shared link, respectively. Therefore, the bandwidth of this link must be guaranteed accordingly, if there is enough traffic in A and B, no matter how many flows originate from A and B.

- ISP C provides network access service to organization D, and C knows the physical link capacity is 100 Mbps. Knowing that the additional traffic, destined to D, more than 100 Mbps, is going to be discarded, the ISP's core routers can allocate bandwidth to the aggregate traffic destined to D with the constraint of no greater than 100 Mbps, thus saving bandwidth to serve other traffics in the ISP's network.

- Company E provides video-on-demand and file storage service via a single link connected to the Internet. To avoid starvation of the best-effort traffic, such as ftp traffic, at least 50% of the bandwidth is allocated to the best-effort traffic. The remaining 50% is allocated to the real-time traffic.

- Carrier's application [8]: ISP X provides VPN service to ISP Y and Z, and Y and Z provide VPN service to other organizations. ISP X must guarantee that the bandwidth in its domain should be allocated, accordingly, among Y, Z and other customers, regardless of the traffic patterns.

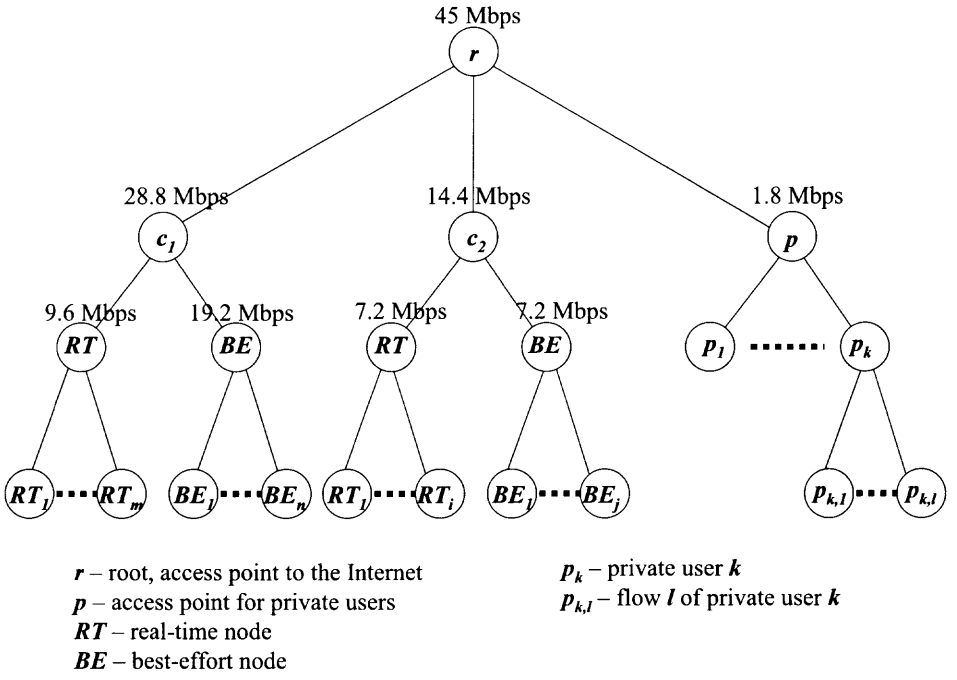

Figure 4.15 One example of a "tiered" scheduling architecture

Note that the proposed schemes can be deployed in these scenarios only when the scheduler has the prior knowledge of the constraints, *i.e.*, which flows or sessions should be placed in the same set.

4.7 Approximating H-GPS by Using Multi-dimensional Deficit Round Robin (M-D DRR) Scheme

Hierarchical Generalized Processor Sharing (H-GPS) [42] possesses desirable properties in terms of flexibility of bandwidth allocation. Hierarchical Packet Fair Queueing (H-PFQ) [42] is deployed to approximate H-GPS. However, P-PFQ requires maintaining the state information of each flow, and thus the implementation cost hinders its application in the real world. The proposed 2-D DRR scheme can be extended to a multi-dimensional scheme, which is able to approximate the H-GPS scheme, while maintaining desirable implementation complexity - $O(1)$.

As shown in Fig. 4.15, node r represents the output link in the Internet of an ISP. c_1 and c_2 are two enterprise customers, with reserved access bandwidth of 28.8 Mbps and 14.4 Mbps, respectively. Node p represents the access point for private users. The unused bandwidth of one flow should be reallocated according to a hierarchical rule. The bandwidth reserved for the private users is shared among all private users equally, regardless of the number of users and flows. For enterprise customer c_1, 1/3 bandwidth is reserved for real-time applications, the rest for best-effort traffics. The bandwidth reserved for real-time applications can be allocated to the real-time flows according to the customers' requirement; the bandwidth reserved for best-effort traffics should be allocated to the best effort flows equally. If there is unused bandwidth of the real-time applications in c_1, this bandwidth should be shared by best-effort traffic in c_1. If there is unused bandwidth of enterprise customer c_1 or c_2, it should be shared by private users.

With the H-PFQ scheme, each non-leaf node works as a regular PFQ server, selecting the flow which should receive service next from its children node. It is a bottom-up process. At the root, the flow which should be served by the H-PFQ scheduler next is selected.

The basic principle of the Multi-Dimensional Deficit Round Robin (M-D DRR) scheduler is that the unused bandwidth of one flow or a set of flows is shared among those flows in the same aggregation. As it is known, in a DRR scheduler, the allocated bandwidth is proportional to the assigned quantum, and thus reassigning the unused quantum of a flow or a set of flows in the same aggregation of flows will not change the allocated bandwidth of the same aggregation. Note that, in the original DRR scheme, the unused quantum of a flow is ignored (discarded) when this flow becomes idle. As described above, the H-PFQ scheme works from the bottom up. On the contrary, the MD DRR scheme works from the top down, *i.e.*, the scheduler decides which children node of the root should be served, first; then from this node, the scheduler decides

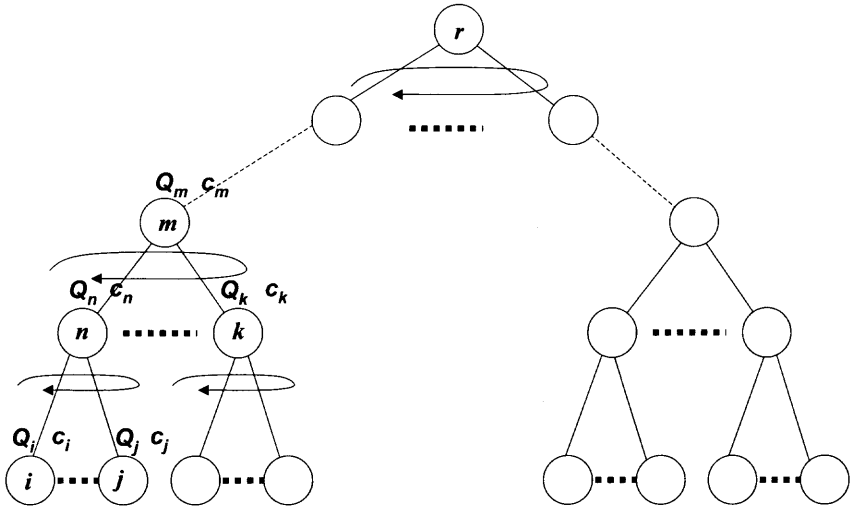

Figure 4.16 The architecture of the M-D DRR scheme

which children node of this node should be served, until the flow which should be served is selected.

Fig. 4.16 shows the architecture of the M-D DRR scheme. Nodes i, j are leaf nodes, which represent flows; m, n and k are non-leaf nodes, which represent the aggregation of a set of flows. Node n is the parent node of nodes i and j. Nodes n and k are children nodes of node m. Denote the first child node of m as $m.head$ (n), and the last child node of m as $m.tail$ (k).

Algorithm 4-5: Multi-Dimensional Deficit Round Robin Scheme (at node **r**)

Initialization Procedure

$createNode(r)$; //create root node

for$(\forall i \in leafNode)$

$computeQuantum(i)$;

$c_i \leftarrow 0$;

for($\forall n \in nonLeafNode$)

$computeQuantum(n)$;

$c_n \leftarrow 0$;

Enqueue Procedure //on packet p arrival

$j \leftarrow extractPacket(p)$;

if($nodeIsIdle(j)$) **then**

$c_j \leftarrow 0$;

$m \leftarrow j.parent$;

if($m == r$) **then**

$appendToAggregation(r.tail)$;

else

$n \leftarrow m.parent$;

while($(n! = r)\&(nodeIsIdle(m))$

$c_m \leftarrow 0$;

$m \leftarrow n$;

$n \leftarrow m.parent$;

$appendToAggregation(n.tail)$;

else

$appendPacketToFlow(j, p)$;

Dequeue Procedure

while($NoActiveChild(r)! =$ **TRUE**)

while($Size(PacketHead) <= c_i$)

$transmit(PacketHead)$;

$c_i \leftarrow c_i - Size(PacketHead)$;

if($NodeIsIdle(i) ==$ **TRUE**)

break;

if($NodeIsIdle(i) ==$ **TRUE**) **then**

$m \leftarrow i;$

$n \leftarrow m.parent;$

while$((m == m.next)\&(n! = r))$

$c_n \leftarrow c_n + c_m;$

$c_m \leftarrow 0;$

$DequeueNode(m);$

$m \leftarrow n;$

$n \leftarrow m.parent;$

if$((n == r)\&(m == m.next))$ **then**

$c_m \leftarrow 0;$

break;

else

$n.head \leftarrow m.next;$

$n.tail \leftarrow m;$

while$(IsLeafNode(m) ==$ **FALSE**$)$

$n \leftarrow m;$

$m \leftarrow n.head;$

$i \leftarrow m;$

else

if$(i! = i.next)$ **then**

$m \leftarrow i.parent;$

$m.head \leftarrow i.next;$

$m.tail \leftarrow i;$

$i \leftarrow m.head;$

$m \leftarrow i;$

$n \leftarrow m.parent;$

while$((Q_m > c_n)\&(n! = r))$

$m \leftarrow i$;

$n \leftarrow m.parent$;

if$(n == r)$ **then**

$n.head \leftarrow m.next$;

$n.tail \leftarrow m$;

$m \leftarrow n.head$;

$c_m \leftarrow c_m + Q_m$;

$n \leftarrow m$;

$m \leftarrow n.head$;

while$(IsLeafNode(m) ==$ **FALSE**$)$

$c_m \leftarrow c_m + Q_m$;

$c_n \leftarrow c_n - Q_m$;

$n \leftarrow m$;

$m \leftarrow n.head$;

$i \leftarrow m$;

The Multi-Dimensional Deficit Round Robin (M-D DRR) scheduler works as follows:

- The M-D DRR scheduler has a tree topology. The root node represents the output link of the scheduler; each leaf node represents a flow, and a non-leaf node represents the aggregation of a set of flows.

- Each non-leaf node works as a deficit round robin scheduler, while accepting quantum from its parent node and assigning quantum to its children nodes.

- For a non-leaf node, 1) if there is unused quantum of its children node, the unused quantum is shared among those nodes who are the children nodes of this non-leaf node; 2) if there is unused quantum when this non-leaf node becomes

idle, it passes the unused quantum to its parent node, unless its parent node is the root.

Regarded as nodes of a linked list, in the DRR scheme, all flows are served in a round robin fashion. The scheduler leaves a flow when it becomes idle or the size of the first packet is greater than the deficit counter of this flow. The M-D DRR scheduler works in a similar way. It leaves a node when 1) it is idle, or 2) the size of the first packet is greater than the deficit counter of this flow, if the node represents a flow, or 3) the deficit counter of this node is less than the quantum of its first children node. Without 3), it is not able to guarantee that each flow must be served at least once when visited by the scheduler.

4.8 Summary

In this chapter, based on the idealized fluid bandwidth allocation scheme proposed in Chapter 3 and the Deficit Round Robin scheme, a novel 2-Dimensional Deficit Round Robin scheme has been proposed. It can be implemented in the real world. The principle of the 2-D DRR scheme is to share the unused bandwidth of one flow to other flows destined to the same endpoint. As compared with the DRR scheme, the 2-D DRR scheme is able to maximize the overall throughput of the VPN, while maintaining the desirable scheduling implementation complexity - $O(1)$. As compared with the related work - the ERR scheme, it possesses the following advantages over the ERR scheme: 1) no need to compute the instant quantum of each flow at the beginning of each service frame; 2) since there is no need to maintain the backlogged information of each flow, it is a stateless scheme; 3) since a newly backlogged flow can be placed at the tail of the linked list of the active flows, the delay bound can be reduced by $\frac{F}{L_{u,v}}$. In order to improve the performance in terms of reduced burstiness, a modified 2-D DRR scheme - 2-D DRR+ is also proposed, the tradeoff is the additional latency which is less than $\frac{F}{L_{u,v}}$. It is demonstrated with an example that the 2-D

DRR+ scheme is able to reduce the burstiness induced by the 2-D DRR scheme. Based on the 2-D DRR+ scheme, a scheme which integrates the best effort traffic has been proposed as well. Deployment of the proposed schemes in the virtual router VPN framework and other applications has been discussed. To approximate H-GPS, the 2-D DRR and 2-D DRR+ schemes can also be extended to M-D DRR and M-D DRR+ schemes, while maintaining the desirable implementation simplicity. They are the extended versions of the DRR scheme. With the proposed 2-D DRR and 2-D DRR+ schemes, since the queueing delay in each core node is bounded by the linear function of the size of service frame F, the overall delay of any packet can be bounded. A hose flow may include many sessions, which have different delay requirements. Thus, the proposed schemes may not meet these requirements.

The performance comparison of the DRR, ERR, 2-D DRR and 2-D DRR+ schemes is listed in Table 4.1.

	DRR	ERR	2-D DRR	2-D DRR+
Guaranteed bandwidth for flow	yes	yes	yes	yes
Guaranteed bandwidth for aggregation	no	yes	yes	yes
Implementation complexity	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Maintain information of each flow	no	yes	no	no
Recompute quantum	no	yes	no	no
Latency	$\frac{3 \cdot F - 2 \cdot Q_{i,j}^{u}}{L_{u,v}}$	$\frac{4 \cdot F - 2 \cdot Q_{i,j}^{u}}{L_{u,v}}$	$\frac{3 \cdot F - 2 \cdot Q_{i,j}^{u}}{L_{u,v}}$	$< \frac{4 \cdot F - 2 \cdot Q_{i,j}^{u}}{L_{u,v}}$

Table 4.1 Performance comparison of various schemes

Usually, a scheduling scheme cannot be described without queueing. Although the scheduling complexity of 2-D DRR+ is $O(1)$, at the output port, it is required to maintain MN^2 queues, where M and N are the number of VPNs that ISP provides

service, and the number of endpoints connected to the ISP's network, respectively. In order to place each incoming packet into its corresponding queue, the queueing complexity is $O(\log MN^2)$. Therefore, the overall implementation complexity at the output port of the router is $O(\log MN^2)$. Clearly, 2-D DRR+ can be implemented, when M and N are not large, to approximate the fluid fair bandwidth allocation scheme for the hose-modeled VPN. However, it dose not scale well, when M and N are large.

Future Work:

1. In order to increase the scalability of the queueing/scheduling scheme, it is necessary to further decrease its implementation complexity. It is believed that hash function could be a good solution, in which some hose-flows share the same queue, and the fairness can be statistically guaranteed. Note that, with hash function, the overall implementation complexity can be reduced to $O(1)$. However, since some aggregations of flows share one queue, the performance in terms of traffic isolation, segregation and hard-guaranteed-QoS could be degraded.

2. With 2-D DRR+, the delay bound of a packet is inversely proportional to the allocated bandwidth to the corresponding flow. However, in the real world, some applications, such as IP telephony, require small bandwidth and small delay, and other applications, such as bulk file transfer, may require large bandwidth, but not sensitive to delay. Therefore, it is necessary to decouple the delay bound and the allocated bandwidth. It is believed that one method is to introduce a priority queue, which buffers all time critical packets. This queue is served until it is empty. However, the additional priority queue could lead to adverse impact on fair bandwidth allocation. Further study on this approach is necessary.

3. Starting from the idealized fluid hose-modeled VPN, it is assumed that all traffics are unicast traffics. However, there is an increasing need for multicast applications in the real world. Thus, it is necessary to develop an implementable and practical scheme which integrates multicast traffics too.

CHAPTER 5

IMPROVING THE SCALABILITY BY APPROXIMATING FAIR BANDWIDTH ALLOCATION

In the previous chapter, 2-D DRR and 2-D DRR+ are proposed to approximate the idealized fluid fair bandwidth allocation scheme in Chapter 3. Although the proposed schemes possess the desirable implementation complexity for scheduling, they require to maintain state, manage buffers and perform packet scheduling on a per hose-flow basis; this complexity may prevent them from being cost-effectively implemented and widely deployed when the number of hose-flows and the link capacity of the core routers are very large. In [51], Stoica, *et al*, proposed a core-stateless fair queueing scheme which can achieve approximately fair bandwidth allocation in the high speed networks. The core-stateless fair queueing scheme, which is also known as non-per-flow-based fair queueing, can significantly reduce the overall implementation complexity. However, this scheme cannot be deployed directly without modification in the hose-modeled VPN environment. In this chapter, the core-stateless fair queueing scheme is reviewed, a modified scheme for the hose-modeled VPN is presented, and its performance is analyzed.

5.1 The Core-stateless Fair Queueing

In a contiguous region of the network, routers are classified as edge routers and core routers. Edge routers maintain per flow state and estimate the incoming rate of each flow; based on this estimation, each packet is labeled when it departs from the edge router. Core routers do not maintain per flow state; they simply use a FIFO queue and a probabilistic dropping algorithm that uses the packet labels and an estimate of the aggregate traffic at the router. Since no per flow state is maintained in the core routers and packets are served based on "first come first serve" policy, the implementation complexity of the core routers is $O(1)$, which is the most desirable.

In Chapter 2, the fluid max-min fair bandwidth allocation scheme of a single link is demonstrated. Denote C as the throughput of this link, where $C = \sum_{i=1}^{N} y_i$, and $y_i = \min(x_i, f \cdot \phi_i)$. Then, each incoming packet of flow i is dropped with probability p_i, where $p_i = \max(0, 1 - \frac{f \cdot \phi_i}{x_i})$. Thus, the throughput of this link C can be written as:

$$C = \sum_{i=1}^{N} \min(x_i, f \cdot \phi_i). \tag{5.1}$$

Note that C is a continuous non-decreasing concave and piecewise-linear function of the fair share rate, f. If the link is congested, *i.e.*, $\sum_{i=1}^{N} x_i > L$, there is a unique solution of f to $\sum_{i=1}^{N} \min(x_i, f \cdot \phi_i) = L$; if the link is not congested, *i.e.*, $\sum_{i=1}^{N} x_i \leq L$, define $f = \max_{i=1}^{N} \frac{x_i}{\phi_i}$. Clearly, if x_i is known, fair share rate f can be computed. However, this requires maintenance of per flow information. To avoid maintaining per flow state, the aggregate measures of C and $\sum_{i=1}^{N} x_i$ are used to compute f.

Simulations in [51] showed that CSFQ approximates DRR reasonably well for both single congested link and multiple congested links. However, as it is pointed out in the previous chapter, if the DRR scheme is employed directly in the hose-modeled VPN, it could lead to low throughput of the whole network, since it does not consider the constraint of the egress link capacity of each endpoint. For the same reason, with a "flat" structure, CSFQ cannot be deployed in the hose-modeled VPN directly. Therefore, a modified scheme to meet property 4 of Definition 3-4 instead of Definition 1-1 will be developed.

5.2 The Modified Non-per-hose-flow-based Fair Bandwidth Allocation Scheme

Same as the mechanism proposed by Stoica in [51], each hose-flow is measured at the customer edge router, *i.e.*, CE, each packet is labeled with the measured rate. In core

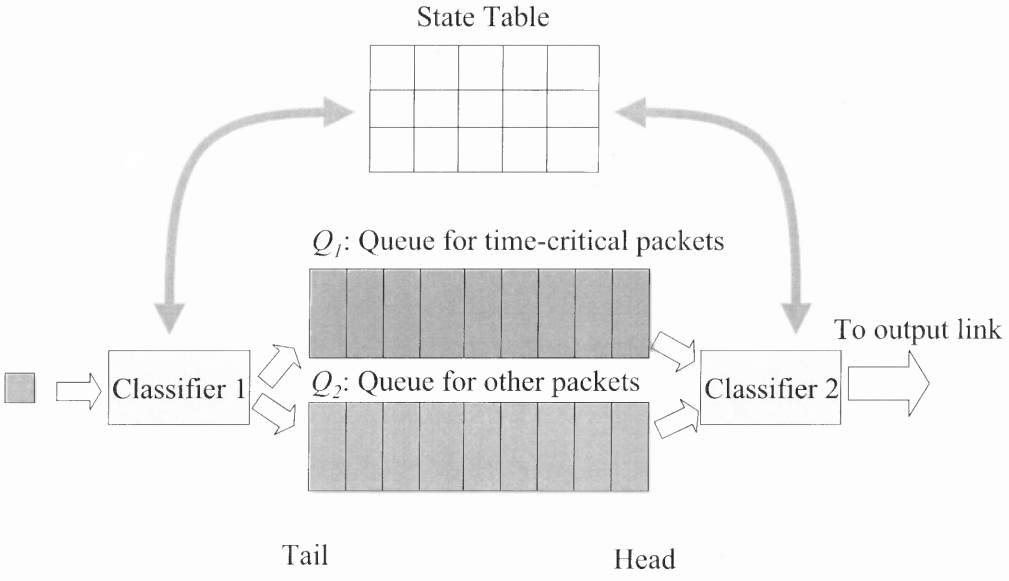

Figure 5.1 The architecture of the output queue in the ISP's router

routers, *i.e.*, PE and P, at the output port, it is necessary to maintain the state of the aggregate of a set of hose-flows to each endpoint. Based on this information, each packet is placed in a FIFO queue or dropped. In order to decouple the delay and allocated bandwidth, two FIFO queues should be used, one for time-critical packets, the other for the remaining packets. The architecture of the output queue is shown in Fig. 5.1.

The state information, such as the fair share rate and aggregate rate to a single endpoint, is maintained in the state table. The classifier before the FIFO queue reads the header of each packet, computes the aggregate rate and the fair share rate for each aggregation of flows destined to the same endpoint, decides if it is accepted or dropped according to its corresponding fair share rate and aggregate rate, and finally updates the state information in the state table. Note, comparing with $M \cdot N^2$ queues in 2-D DRR+, the proposed mechanism requires only two FIFO queues, and thus the

implementation complexity of queueing is $O(1)$. There are two differences between the proposed scheme and the CSFQ scheme: 1) the proposed scheme decides to accept or drop an incoming packet according to its arriving rate and the aggregate rate of its corresponding group, whereas the CSFQ performs according to the arriving rate and the aggregate rate of all flows on this link; 2) in order to provide guaranteed service to some flows, packets in these flows are placed into $Queue_1$. With the CSFQ scheme, the aggregate rate and fair share rate are computed once when a packet arrives in the intermediate node; this could be processing intensive when the output link capacity is large. In the proposed scheme, the time-sliding window (TSW) approach proposed in [52] should be used. The aggregate bandwidth allocated to the hose flows destined to endpoint j, d_j^u, is updated as Eq. 5.2, when the time interval is greater than the predefined time-sliding window, *i.e.*, $T^u - T_0^u \geq TSW$.

$$New\_d_j^u = (1 - rate\_factor) \cdot Old\_d_j^u + rate\_factor \cdot \frac{ctr_j^u}{T^u - T_0^u}, \tag{5.2}$$

where $0 < rate\_factor \leq 1$. ctr_j^u is the aggregate traffic volume, in the current time interval, destined to endpoint j.

Label	aggregate traffic volume	starting time	aggregate bandwidth		fair share rate		egress link capacity
j	ctr_j^u	t_j^u	$Old\_d_j^u$	$New\_d_j^u$	$Old\_f_j^u$	$New\_f_j^u$	L_j^{Out}

Table 5.1 The state table of the aggregate hose-flows

At the output port of each intermediate router, all packets destined to the same endpoint have the same label. The state table is shown in Table 5.1.

The computation of the fair share rate will be discussed in the next section.

Algorithm 5-1: The proposed non-per-flow-based fair bandwidth allocation scheme at intermediate node **u** (link to node v)

Initialization Procedure:

$rate\_factor = constant$;

$TSW = constant$;

$T_0^u = T^u$;

for$(j = 0; j < |\mathbb{V}_{CE}^{v/(u,v)}|; j++)$

$New\_f_j^u = 1$;

$Old\_f_j^u = 1$;

$New\_d_j^u = 0$;

$Old\_d_j^u = 0$;

$ctr_j^u = 0$;

Enqueuing Procedure: (Run in Classifier) // on arrival of packet p

Aggregate_Hose_Flow$(j) = ExtractFlow(p)$

if (PacketIsTimeCritical(p) ==TRUE) **then**

Enqueue($Queue_1$);

$ctr_j^u = ctr_j^u + PacketSize(p)$;

else

if$(random(0,1) >= \max(0, 1 - \frac{New\_f_j^u}{label(p)}))$ **then**

Enqueue($Queue_2$);

$ctr_j^u = ctr_j^u + PacketSize(p)$;

else

Discard_Packet(p);

Dequeuing Procedure: (Run in Scheduler)

while ((QueueIsEmpty($Queue_1$) ==FALSE)

OR(QueueIsEmpty($Queue_2$) ==FALSE))

if (QueueIsEmpty($Queue_1$) ==FALSE) **then**

Packet(p)=HeadPacket($Queue_1$);

else

Packet(p)=HeadPacket($Queue_2$);

Send(p);

State Updating Procedure: (Run in Classifier) //Time driven procedure

while $(T^u - T_0^u >= TSW)$

for$(j = 0; j < |\mathbb{V}_{CE}^{v/(u,v)}|; j++)$

Update_Aggregate_Rate($Old\_d_j^u, New\_d_j^u$);

Update_Fair_Share_Rate($Old\_f_j^u, New\_f_j^u$);

$ctr_j^u = 0$;

$T_0^u = T^u$;

In the Initialization Procedure, the time sliding window size and the rate factor are defined, the time stamp T^u is assigned the value of the system time T^u, and all state parameters of each aggregation of flows are initialized. In the Enqueueing Procedure, on arrival of a new incoming packet, if this packet is time critical, then the classifier places it at the tail of $Queue_1$; otherwise, according to the label in the packet header, the classifier looks up the state table, decides whether this packet should be accepted or discarded with a randomly generated number. If this is accepted, then the classifier places it at the tail of $Queue_2$. The Dequeueing Procedure runs until both queues are empty, and $Queue_1$ is served unless it is empty. An additional procedure, the State Updating Procedure, runs in a time-driven fashion to set the time stamp, to reset the traffic counter, and to re-compute the aggregate rate and the fair share rate of each aggregation of flows.

Note that the classification implementation complexity at the output queue is $O(logM \cdot N)$, although there are only two queues. The feasibility of implementing the proposed scheme is discussed in the next section.

5.3 The Computation of the Fair Share Rate

The fair share rate estimation scheme is one key element for the core-stateless fair queueing scheme. The accuracy and effectiveness of the scheme for computing the fair share rate affects the effectiveness of the core-stateless fair queueing scheme.

As shown in Eq. 5.1, the throughput of a single link, C, is a continuous, non-decreasing, concave, and piecewise-linear function of the fair share rate, f. If the link is congested, *i.e.*, $\sum_{i=1}^{N} x_i > L$, there is a unique solution of f to $\min_{i=1}^{N}(x_i, f \cdot \phi_i) = L$; if the link is not congested, *i.e.*, $\sum_{i=1}^{N} x_i \leq L$, define $f = \max_{i=1}^{N} \frac{x_i}{\phi_i}$. Since x_i is unknown, f is estimated, based on both measurement and the properties of the function of C. Therefore, the estimation of the fair share rate can be formulated as a root finding problem for a nonlinear equation as follows:

$$\Psi(f) = \sum_{i=1}^{N} \min(x_i, f \cdot \phi_i) - L = C(f) - L = 0. \tag{5.3}$$

To allocate bandwidth accurately, and obtain a stable queue size, a good estimation should approach (converge to) the exact f reasonably fast.

A modified secant method is proposed in this section. It is demonstrated that the proposed method converges faster than the original method proposed in [10].

5.3.1 The Original Fair Share Rate Estimation Algorithm

Stoica proposed an iterative method [10]. Denote the kth iteration of the fair share rate estimation and the measured throughput by $f^{(k)}$ and $C^{(k)}$, respectively. Denote the exact fair share rate by f^*. The original fair share rate estimation algorithm proposed in [10], which is referred to as the FSR_{CSFQ} method in this dissertation, is written as follows:

$$f^{(k+1)} = f^{(k)} \cdot \frac{L}{C^{(k)}}. \tag{5.4}$$

Eq. 5.4 can be written as:

$$f^{(k+1)} = f^{(k)} - \frac{f^{(k)}}{C^{(k)}} \cdot (C^{(k)} - L). \tag{5.5}$$

5.3.2 The Regula Falsi Method, the Newton-Raphson Method, and the Secant Method

For a nonlinear equation 5.3, the root must be in the interval $[0, L]$. The regula falsi method, a closed domain method, can be expressed as follows:

$$f^{(k+1)} = \begin{cases} f^{(k)} - \frac{1}{\frac{C^{(k)} - C^{(k-1)}}{f^{(k)} - f^{(k-1)}}} \cdot (C^{(k)} - L), & when\ (C^{(k)} - L) \cdot (C^{(k-1)} - L) < 0 \\ f^{(k)} - \frac{1}{\frac{C^{(k)} - C^{(k-2)}}{f^{(k)} - f^{(k-2)}}} \cdot (C^{(k)} - L), & otherwise \end{cases} \tag{5.6}$$

whereas $f^{(0)} = 1, f^{(1)} = L$, and $f^{(2)} = f^{(1)} - \frac{1}{\frac{C^{(1)} - C^{(0)}}{f^{(1)} - f^{(0)}}} \cdot (C^{(1)} - L)$.

The Newton-Raphson method for solving nonlinear equations is one of the most well-known and powerful procedures in numerical analysis [53]. It always converges if the initial approximation is sufficiently close to the root, and the rate of convergence is quadratic. In this scenario, it can be written as:

$$f^{(k+1)} = f^{(k)} - \frac{1}{\frac{\partial C^{(k)}}{\partial f^{(k)}}} \cdot (C^{(k)} - L). \tag{5.7}$$

The disadvantage of the Newton-Raphson method is that the derivative $\frac{\partial C^{(k)}}{\partial f^{(k)}}$ must be evaluated. When the derivative $\frac{\partial C^{(k)}}{\partial f^{(k)}}$ is unavailable, an alternative is required. The preferred alternative is the secant method, which is written as:

$$f^{(k+1)} = f^{(k)} - \frac{1}{\frac{C^{(k)} - C^{(k-1)}}{f^{(k)} - f^{(k-1)}}} \cdot (C^{(k)} - L). \tag{5.8}$$

A secant to a curve is a straight line which passes through two points on the curve. The secant method also converges to the root as the Newton-Raphson method. Note that the slope of a secant $\frac{C^{(k)}-C^{(k-1)}}{f^{(k)}-f^{(k-1)}}$ in Eq. 5.8 is used to approximate the derivative $\frac{\partial C^{(k)}}{\partial f^{(k)}}$ in Eq. 5.7. Two initial approximations $f^{(0)}$ and $f^{(1)}$ are required.

The regula falsi is quite robust, but converges slowly. The secant method is preferred since it converges much more rapidly. It is demonstrated, in the next subsection, that the proposed method converges faster than the original method proposed in [10].

5.3.3 The Proposed, Modified Secant Method

In this section, the proposed, modified secant method is presented. The secant method, described in the previous section, is modified for the following two reasons: 1) The fair share rate may change when the traffic pattern changes, and so it is dynamic. For example, after a period of stable status, *i.e.*, when $f^{(k)} = f^{(k-1)}$, if there is a change of the throughput, the fair share rate should be adjusted accordingly. However, Eq. 5.8 cannot be used, since $f^{(k)} - f^{(k-1)} = 0$. However, the origin must be on the curve of Eq. 5.1. Therefore, in this case, $f^{(k-1)} = 0, C^{(k-1)} = 0$ is used to compute $f^{(k+1)}$. 2) As shown in the next part, when the fair share rate decreases, by the secant method, $f^{(k+1)}$ may be less than 0, which has no meaning. Therefore, let $f^{(k+1)} = 0$ in this case.

When the scheduler starts to transmit packets, to avoid buffer overflow, a fair share rate $f^{(1)} = \delta \cdot L$, which is small enough that the throughput is not greater than L, is first attempted. With the measurement of $C^{(1)}$, the secant method can be started. The proposed scheme can be written as follows:

Algorithm 4-2: The modified secant algorithm

Initialization Procedure

$f^{(0)} = 0;$

$C^{(0)} = 0;$

$f^{(1)} = \delta \cdot L;$

$C^{(1)} = Measure\_Throughput;$

Update Fair Share Rate Procedure

$k = k + 1;$

$C^{(k)} = Measure\_Throughput;$

if $((f^{(k)} - f^{(k-1)}) \cdot (C^{(k)} - C^{(k-1)}) == 0)$

$f^{(k-1)} = 0;$

$C^{(k-1)} = 0;$

$f^{(k+1)} = f^{(k)} - \frac{1}{\frac{C^{(k)} - C^{(k-1)}}{f^{(k)} - f^{(k-1)}}} \cdot (C^{(k)} - L);$

if $(f^{(k+1)} < 0)$

$f^{(k+1)} = f^{(k)} \cdot \frac{L}{C^{(k)}};$

5.3.4 Performance Analysis of the Proposed, Modified Secant Method

In this chapter, the performance in terms of convergence rate is compared between the method proposed in [10] and the modified secant method. It is demonstrated that, with the proposed secant method, the exact fair share rate can be reached; and, with the FSR_{CSFQ} method, the exact fair share rate can be approximated, but may not be reached. Consider the following two scenarios: 1) when the fair share rate increases, it is demonstrated that the proposed method approximates the exact fair share rate better than the FSR_{CSFQ} method; 2) when the fair share rate decreases, the FSR_{CSFQ} method is equivalent to the regula falsi method, which converges much slower than the secant method.

As shown in Eq. 5.1, the throughput of a single link, C, is a continuous, non-decreasing, concave, and piecewise-linear function of the fair share rate, f. Therefore, the following inequalities must hold.

$$\forall\ 0 < f^{(a)} < f^{(b)} < L,\ C(f^{(a)}) \leq C(f^{(b)}). \tag{5.9}$$

$$\forall\ 0 < f^{(a)} < f^{(b)} < L,\ \frac{\partial C(f^{(b)})}{\partial f} \leq \frac{\partial C(f^{(a)})}{\partial f}. \tag{5.10}$$

$$\forall\ 0 < f^{(a)} < f^{(b)} < L,\ \frac{C(f^{(b)})}{f^{(b)}} \leq \frac{C(f^{(a)})}{f^{(a)}}. \tag{5.11}$$

$$\forall\ f^{(b)},\ \frac{\partial C(f^{(b)})}{\partial f} \leq \frac{C(f^{(b)})}{f^{(b)}}. \tag{5.12}$$

Ineq. 5.9 demonstrates the non-decreasing property, and Ineqs. 5.10, 5.11 and 5.12 demonstrate the concave property. By the above inequalities and the fact that the origin is on the curve of $C(f)$, Ineq. 5.13 can be readily derived.

$$\forall\ 0 < f^{(a)} < f^{(b)} < L, \frac{\partial C(f^{(b)})}{\partial f} \leq \frac{C(f^{(b)}) - C(f^{(a)})}{f^{(b)} - f^{(a)}} \leq \frac{\partial C(f^{(a)})}{\partial f}. \tag{5.13}$$

Case 1: The Fair Share Rate Increases

Let $0 < f_{Secant}^{(k-1)} < f_{Secant}^{(k)} < f^*$, and $f_{CSFQ}^{(k)} = f_{Secant}^{(k)}$.

By Ineq. 5.11,

$\frac{C(f_{Secant}^{(k)})}{f_{Secant}^{(k)}} \leq \frac{C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k-1)}} \Rightarrow C(f_{Secant}^{(k)}) \cdot f_{Secant}^{(k-1)} \leq C(f_{Secant}^{(k-1)}) \cdot f_{Secant}^{(k)}$

$\Rightarrow C(f_{Secant}^{(k)}) \cdot f_{Secant}^{(k)} - C(f_{Secant}^{(k)}) \cdot f_{Secant}^{(k-1)} \geq C(f_{Secant}^{(k)}) \cdot f_{Secant}^{(k)} - C(f_{Secant}^{(k-1)}) \cdot f_{Secant}^{(k)}$

$\Rightarrow C(f_{Secant}^{(k)}) \frac{}{f_{Secant}^{(k)} \geq \frac{C(f_{Secant}^{(k)}) - C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k)} - f_{Secant}^{(k-1)}}}$

$\because f_{Secant}^{(k)} < f^*$, by Ineq. 5.9, $C(f_{Secant}^{(k)}) \leq L$.

$\therefore f_{Secant}^{(k)} - \frac{1}{\frac{C(f_{Secant}^{(k)})}{f_{Secant}^{(k)}}} \cdot (C(f_{Secant}^{(k)}) - L) \leq f_{Secant}^{(k)} - \frac{1}{\frac{C(f_{Secant}^{(k)}) - C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k)} - f_{Secant}^{(k-1)}}} \cdot (C(f_{Secant}^{(k)}) - L)$

$\because f_{CSFQ}^{(k)} = f_{Secant}^{(k)}$

$$\therefore f_{CSFQ}^{(k)} - \frac{1}{\frac{C(f_{CSFQ}^{(k)})}{f_{CSFQ}^{(k)}}} \cdot (C(f_{CSFQ}^{(k)}) - L) \leq f_{Secant}^{(k)} - \frac{1}{\frac{C(f_{Secant}^{(k)}) - C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k)} - f_{Secant}^{(k-1)}}} \cdot (C(f_{Secant}^{(k)}) - L)$$

By Eqs. 5.5 and 5.8,

$$f_{CSFQ}^{(k+1)} \leq f_{Secant}^{(k+1)}.$$

By Ineq. 5.13,

$$\frac{C(f_{Secant}^{(k)}) - C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k)} - f_{Secant}^{(k-1)}} \leq \frac{\partial C(f^{(k)})}{\partial f} \leq \frac{C(f^*) - C(f_{Secant}^{(k)})}{f^* - f_{Secant}^{(k)}}.$$

$$\because C(f_{Secant}^{(k)}) \leq L$$

$$\therefore f_{Secant}^{(k)} - \frac{1}{\frac{C(f_{Secant}^{(k)}) - C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k)} - f_{Secant}^{(k-1)}}} \cdot (C(f_{Secant}^{(k)}) - L) \leq f_{Secant}^{(k)} - \frac{1}{\frac{C(f^*) - C(f_{Secant}^{(k)})}{f^* - f_{Secant}^{(k)}}} \cdot (C(f_{Secant}^{(k)}) - L)$$

$$\because C(f^*) = L$$

$$\therefore f_{Secant}^{(k)} - \frac{1}{\frac{C(f_{Secant}^{(k)}) - C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k)} - f_{Secant}^{(k-1)}}} \cdot (C(f_{Secant}^{(k)}) - L) \leq f^*$$

By Eq. 5.8, $f_{Secant}^{(k+1)} \leq f^*$.

$$\therefore f_{CSFQ}^{(k+1)} \leq f_{Secant}^{(k+1)} \leq f^*. \tag{5.14}$$

Ineq. 5.14 demonstrates that the secant method can approximate the exact value of the fair share rate better than the CSFQ method, when the fair share rate increases.

As shown in Fig. 5.2, the function of C vs f can be depicted as the solid line, which passes through the origin. a, b, c, d, and e are points on the curve with Cartesian coordinates $(f_{Secant}^{(k-1)}, C(f_{Secant}^{(k-1)}))$, $(f_{Secant}^{(k)}, C(f_{Secant}^{(k)}))$, (f^*, L), $(f_{CSFQ}^{(k+1)}, C(f_{CSFQ}^{(k+1)}))$, and $(f_{Secant}^{(k+1)}, C(f_{Secant}^{(k+1)}))$, respectively. d, b and the origin are on the same line, which is used to compute $f_{CSFQ}^{(k+1)}$ by the FSR_{CSFQ} method. a, b and e are on the same line, which is used to computed $f_{Secant}^{(k+1)}$ by the proposed method. Since $0 < f_{Secant}^{(k-1)} < f_{Secant}^{(k)} < f^*$, and $f_{CSFQ}^{(k)} = f_{Secant}^{(k)}$, the slope of the dotted line $\bar{ab}$, $\frac{C(f_{Secant}^{(k)}) - C(f_{Secant}^{(k-1)})}{f_{Secant}^{(k)} - f_{Secant}^{(k-1)}}$, must be greater than the slope of the dotted line $\bar{bd}$, $\frac{C(f_{CSFQ}^{(k)})}{f_{CSFQ}^{(k)}}$. The ideal slope is the slope of the dotted line $\bar{bc}$, which passes the point with the exact

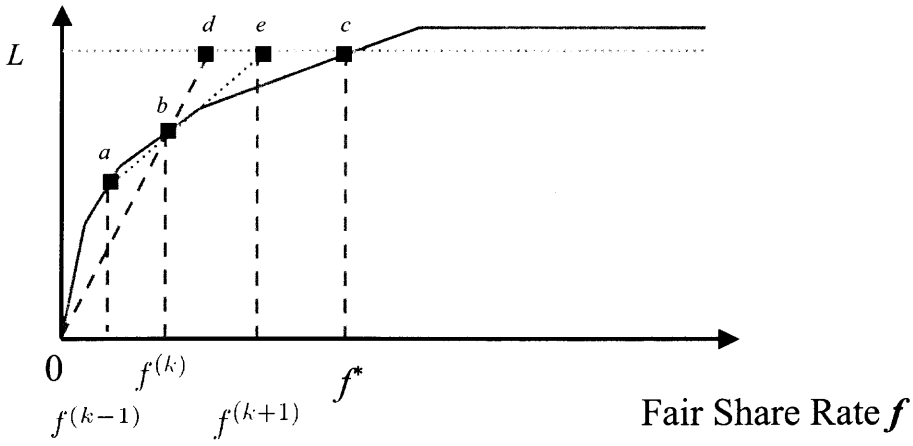

Figure 5.2 The geometric explanation of the fair share rate computation when it increases

fair share rate. Therefore, the slope used by the proposed secant method approximates point c better than that by the FSR_{CSFQ} method.

Note that 1) if there is only one line segment between point c and the origin, both methods are able to compute the exact fair share rate; 2) if there are more than one line segment between point c and the origin, the FSR_{CSFQ} method can approximate point c infinitesimally, but can never reach it, mathematically. However, the proposed method can still reach point c, if, with enough iterations, point b is on the line which passes through point c; 3) theoretically, when the fair share rate increases, the computed fair share rate of each iteration is not greater than the exact value, implying that the queue size cannot grow infinitely.

Case 2: The Fair Share Rate Decreases

In this section, first, it is shown that, when the fair share rate decreases, the FSR_{CSFQ} method is equivalent to the regula falsi method. Since it is claimed, in [53], that the secant method converges much more rapidly than the regula falsi method, the proposed method possesses better performance in terms of convergence than the

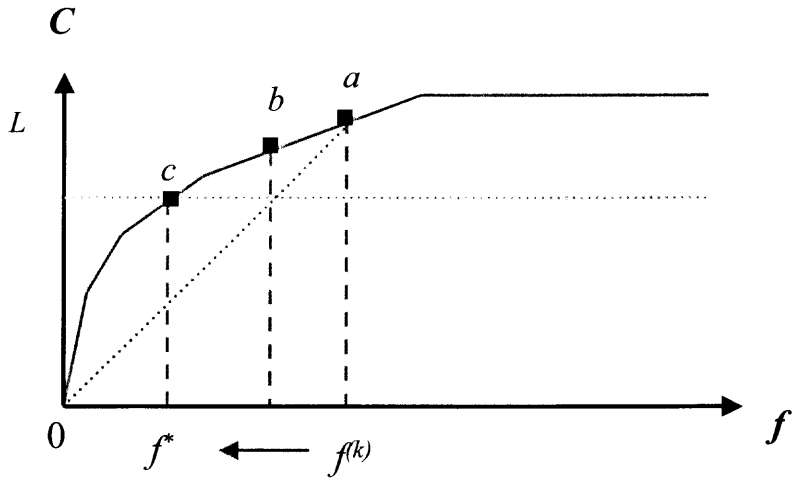

Figure 5.3 The geometric explanation of the fair share rate computation when it decreases, using the FSR_{CSFQ} method

FSR_{CSFQ} method. Finally, the reason to modify the secant method, due to some computation of $f^{(k+1)}$, is demonstrated.

Let $f^* < f^{(k)}_{CSFQ} < f^{(k-1)}_{CSFQ}$. Thus, by Eq. 5.9, $L < C(f^{(k)}_{CSFQ})$.

Then, by Eq. 5.4, $0 \leq f^{(k+1)}_{CSFQ} < f^{(k)}_{CSFQ}$, *i.e.*, the computed fair share rate approximates the exact rate from the right hand side, as shown in Fig. 5.3. a, b, and c are points on the curve with Cartesian coordinates $(f^{(k-1)}_{CSFQ}, C(f^{(k-1)}_{CSFQ}))$, $(f^{(k)}_{CSFQ}, C(f^{(k)}_{CSFQ}))$, and (f^*, L), respectively. The FSR_{CSFQ} method always uses the origin and $f^{(k)}_{CSFQ}$ to compute $f^{(k+1)}_{CSFQ}$. It is, mathematically, equivalent to find the root of Eq. 5.3, by Eq. 5.6, whereas $f^{(k-2)} = 0$. Again, if there is only one line segment between point c and the origin, the FSR_{CSFQ} method is able to compute the exact fair share rate; otherwise, the FSR_{CSFQ} method can approximate point c infinitesimally, but can never reach it.

When the secant method is deployed, $\frac{C^{(k)}-L}{C^{(k-1)}-L}$ may be less than $\frac{f^{(k)}}{f^{(k-1)}}$, and thus by Eq. 5.8, $f^{(k+1)} < 0$, as shown in Fig. 5.4. a, b, c, d, and e are points on the curve with Cartesian coordinates $(f^{(k-1)}_{Secant}, C(f^{(k-1)}_{Secant}))$, $(f^{(k)}_{Secant}, C(f^{(k)}_{Secant}))$, (f^*, L),

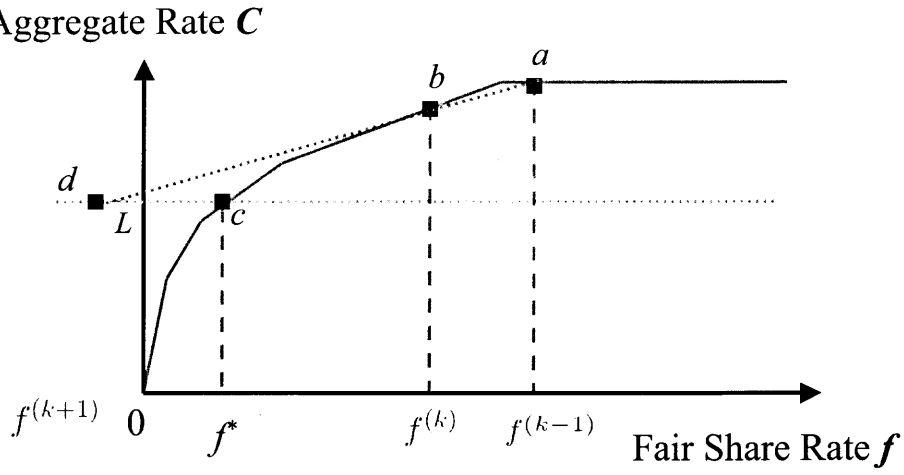

Figure 5.4 The geometric explanation of the fair share rate computation when it decreases, using the proposed modified secant method

and $(f_{Secant}^{(k+1)}, C(f_{Secant}^{(k+1)}))$, respectively. The fair share rate cannot be set to a value which is less than 0. Thus, in this sense, the original secant method cannot be used without modification. Therefore, when $f^{(k+1)} < 0$, $f^{(k+1)}$ is computed by Eq. 5.4. That is, if $f^{(k+1)} < 0$, the FSR_{CSFQ} method is used again in this iteration.

Note that 1) if there is only one line segment between point a and the origin, both methods are able to compute the exact fair share rate; 2) if there are more than one line segment between point a and the origin, the FSR_{CSFQ} method can approximate point c infinitesimally, but can never reach it, mathematically. However, the proposed method can still reach point c, if, with enough iterations, point b is on the line which passes through point c; 3) theoretically, with the method proposed in [10], when the fair share rate decreases, the computed fair share rate is always greater than the exact value, and then the rate of the accepted traffic is greater than the link capacity, thus making the queue size grow infinitely. With the proposed modified secant method, the computed fair share rate of each iteration may be greater or less

than the exact value before it reaches the exact value, and thus, it performs better in terms of controlling the queue size.

5.3.5 Simulation Results

In the previous section, it is demonstrated, geometrically, that the secant method converges faster than the FSR_{CSFQ} method, when the fair share rate increases; it is also demonstrated that, when the fair share rate decreases, the FSR_{CSFQ} method performs the same as the regula falsi method, which converges slower than the secant method. With the following examples, it is shown that the proposed scheme converges faster than the original fair share rate estimation algorithm, thus possessing better performance in terms of approximating the exact fair share rate.

Example 1: The fair share rate increases

Suppose 20 flows share one link whose link capacity is 10. Each flow has the same weight, *i.e.*, $\forall i = 1, 2, ..20$, $\phi_i = 0.05$. Assume starting from time 0, the arrival rate of the first five flows are 1 , 2, 3, 4, and 5, respectively, and the remaining 15 flows are 1. At t_{K-2}, the fair share rate reaches the exact rate and remains stable till t_{K-1}, *i.e.*, $f^{(K-1)} = f^{(K-2)} = 0.02$. Starting from t_K, only the first five flows remain the same and the remaining 15 flows become idle. Therefore, the function of aggregate rate vs the fair share rate can be written as:

$$C = \begin{cases} 5 \cdot f\ , & when\ 0 \leq f < 1 \\ 4 \cdot f + 1, & when\ 1 \leq f < 2 \\ 3 \cdot f + 3, & when\ 2 \leq f < 3 \\ 2 \cdot f + 6, & when\ 3 \leq f < 4 \\ f + 10, & when\ 4 \leq f < 5 \\ 15, & when\ 5 \leq f \end{cases}, \tag{5.15}$$

whereas the root of the equation, $\Psi(f) = C(f) - L = 0$, is $f^* = 2.33333333$. The relatively error is defined as:

$$e^{(k)} = \left| \frac{f^{(k)} - f^*}{f^*} \right| \cdot 100\%. \tag{5.16}$$

k	$K-1$	K	$K+1$	$K+2$	$K+3$	$K+4$	$K+5$	$K+6$	$K+7$
$f^{(k)}_{CSPF}$	0.5000	2.0000	2.2222	2.2989	2.3229	2.3302	2.3324	2.3331	2.3332
$f^{(k)}_{Secant}$	0.5000	2.0000	2.2308	2.3333	2.3333	2.3333	2.3333	2.3333	2.3333
$e^{(k)}_{CSPF}$	–	25.00	4.76	1.47	0.45	0.13	0.04	0.01	0.00
$e^{(k)}_{Secant}$	–	25.00	4.39	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.2 The comparison of different computation methods in Example 1

Denote the fair share rate computed by the method in [10] and the proposed, modified secant method by $f^{(k)}_{CSPF}$ and $f^{(k)}_{Secant}$, respectively. Denote the relative error by using the FSR_{CSFQ} method and the proposed, modified secant method by $e^{(k)}_{CSPF}$ and $e^{(k)}_{Secant}$, respectively. Note that, in the computation, only four digits after the decimal point are kept. Table 5.2 lists the comparison of the iterative computation of the FSR_{CSFQ} method and the proposed, modified secant method. With 3 iterations, by using the proposed method, the exact fair share rate can be reached, whereas with the FSR_{CSFQ} method, it needs 9 iterations.

Example 2: The fair share rate decreases

Suppose 5 flows share the same link in Example 1. Each flow has the same weight, *i.e.*, $\forall i = 1, 2, , 3, 4, 5$, $\phi_i = 0.2$. Assume starting from time 0, the arrival rate of the first four flows are 1 , 2, 3, and 4, respectively, and the last flow is idle. At t_{k-2}, the fair share rate reaches the exact rate and remains stable till t_{K-1}, *i.e.*, $f^{(K-1)} = f^{(K-2)} = 4$. Starting from t_K, only the last flow becomes active and

the arrival rate is 5, while the rest of the flows remain the same as before. Then, Eq. 5.15 still holds, and $f^* = 2.3333$. Table 5.3 lists the comparison of the iterative computation of the original method and the proposed, modified secant method. With 3 iterations by using the proposed method, the exact fair share rate can be reached, whereas the FSR_{CSFQ} method needs 9 iterations. This result is very similar to that in Example 1.

k	$K-1$	K	$K+1$	$K+2$	$K+3$	$K+4$	$K+5$	$K+6$	$K+7$
$f_{CSFQ}^{(k)}$	4.0000	2.8571	2.4691	2.3725	2.3449	2.3368	2.3344	2.3336	2.3334
$f_{Secant}^{(k)}$	4.0000	2.8571	2.1176	2.3333	2.3333	2.3333	2.3333	2.3333	2.3333
$e_{CSFQ}^{(k)}$	–	22.45	5.82	1.68	0.50	0.15	0.05	0.01	0.00
$e_{Secant}^{(k)}$	–	22.45	9.24	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.3 The comparison of different computation methods in Example 2

5.4 Implementing the Proposed Scheme within the Framework of MPLS VPN

5.4.1 Multi-Protocol Label Switching (MPLS)

Multi-Protocol Label Switching (MPLS) is an emerging technology that aims to address many of the existing issues which are associated with packet forwarding in today's Internetworking environment. As described in [23], the primary goal of the MPLS working group of IETF is to standardize a base technology that integrates the label swapping forwarding paradigm with network layer routing. The label swapping technology is expected to improve the price/performance of network layer routing, enhance the scalability of the network layer, and provide greater flexibility in the delivery of new routing services. It is able to support, with a single network, different kinds of network protocols, such as IP, ATM, frame relay, and so on.

A router is called a label switch router (LSR) if it supports MPLS. There are two kinds of LSR: 1) LSR, which forwards label packets; 2) edge-LSR, which receives an IP packets, performs Layer 3 lookups and imposes a label stack before forwarding the packet into the LSR domain, or receives a labeled packet, removes labels, performs Layer 3 lookups, and forwards the IP packet.

5.4.2 BGP/MPLS VPN

In [7, 8], a framework of BGP/MPLS VPN is proposed.

P and PE routers share a common routing protocol with the ISP's core network. These routers use this routing information to build label-switched paths between PE routers and use two levels of labels to forward packets. Virtual routing and forwarding (VRF) instances are defined on the PE routers, and each VRF instance represents an endpoint of VPN, a separate routing information base (RIB), and a set of interfaces to which VPNs are attached. To identify VPN routes on a PE router, the VRF needs to define a route distinguisher (RD) which is pre-appended to each VPN route. The route information is passed via a multi-protocol border gateway protocol (MP-BGP), which is an extension of BGP, to peer PE routers, *i.e.*, VPN routing tables are propagated between PE routers by using MP-BGP. Therefore, within the framework of BGP/MPLS VPN, BGP is used to exchange VPN routing information and MPLS is used to build label-switched path and forward packets.

At the ingress endpoint of BGP/MPLS VPN, the PE router adds two labels to the IP packets: the inner label is for the destination VPN route, which carries the VPN identifier; the outer label is used to select the label-switched path and the next peer PE router (or the BGP next hop). Note that, the outer label is used in the ISP's domain for packet forwarding. The inner label is used only at a PE router to identify to which VPN the packets belong.

5.4.3 Integrating the Proposed Scheme within the Framework of MPLS VPN

In order to integrate the proposed scheme within the framework of MPLS VPN, it is necessary to do some modifications.

First, the forwarding tables in P routers have to be modified. Table 5.4 shows an example of the forwarding table in a P router. According to its incoming interface and incoming label, the P router swaps a label and then forwards the incoming packet to the outgoing interface. In this example, packets from interface eth0 with label 200 and packets from interface eth1 with label 100 are assigned a new label before they are forwarded to interface eth2.

Note that, In. Int represents incoming interface, In. Lbl represents incoming label, and O. Lbl and O. Int represent outgoing label and outgoing interface, respectively. Time Stmp. represents the time the counter is reset, and Tra. Vol. is the counter to record how many bytes have been accepted after the counter is reset. Bdwidth records the old and new aggregate bandwidth, F. Share records the old and new fair share rate. Cpy represents the egress link capacity connected to the endpoint, *i.e.*, the destined CE router.

Incoming Interface	Incoming Label	Outgoing Label	Outgoing Interface
eth0	200	300	eth2
eth1	100		

Table 5.4 An example of the forwarding table in a P router

It is necessary to combine Table 5.1 and Table 5.4 together to generate a modified forwarding table. Table 5.5 shows an example of the modified forwarding table in a P router.

With the modified forwarding table, the P router has to read not only the outer label (incoming label) to assign a new label to the packet, but also the inner label

(the VPN ID) and the arrival rate of the incoming packet to decide if this packet should be accepted according to the fair share rate. If this packet is accepted, the counter of traffic volume should be updated.

In. Int	In. Lbl	VPN ID	O. Lbl	O. Int.	Tra. Vol.	Time Stmp.	Bdwidth	F.Share	Cpy
eth0	200	1024	300	eth2	2702	19:02:14:001	1.7/1.8	0.4/0.5	2
eth1	100	1024							

Table 5.5 An example of the modified forwarding table in a P router

5.5 Integrating Multicast Traffics

Multicast applications are becoming popular today. These applications include multimedia conferencing, data distribution, software upgrades, real-time data multicast, and bootstrap server services [1]. Multicast communication helps to reduce the amount of traffic in the ISPs' backbone, and hence to make better use of the network resources in terms of bandwidth and buffer. The operation of IP multicast is demonstrated in [48] and [47]. To realize IP multicast communication in MPLS environment, Ooms, *et al*, proposed [54] to integrate multicast communication in the framework of MPLS.

It is believed that the proposed, modified non-per-hose-flow-based fair bandwidth allocation scheme can be readily integrated in the framework of MPLS. The following modifications are proposed:

- distinguish the multicast label from the unicast label by setting the most significant bit to 1;
- create the forwarding table in each P or PE router with consideration of multicast traffic, *i.e.*, a separate forwarding table for multicast traffics; in this table, there are pointers pointing to the records of the unicast forwarding table;

- if an incoming packet belongs to multicast traffic, the corresponding traffic volume counters should be updated.

The proposed method is demonstrated via the sample example shown in Fig. 4.1 in Chapter 4. Assume H_0 sends multicast packets to H_3, H_4 and H_5. At R_8, the interfaces connected to R_{10}, R_9 and H_3 are $s0$, $s1$ and $s2$, respectively. The part of the forwarding table for the unicast traffic (with destinations to H_3, H_4 and H_5, respectively) is shown in Table 5.6.

Des.	In. Int	In. Lbl	O. Lbl	O. Int.	Tra. Vol.
H_3	s0 s2	200 -	-	s2	TC_4
H_4	s0 s2	300 -	400	s1	TC_5
H_5	s0 s2	400 -	500	s1	TC_6

Table 5.6 The part of the forwarding table for unicast traffic in router R_8

TC_i represents the traffic volume counter for traffic destined to H_i. Table 5.7 shows a part of the forwarding table for the multicast traffic (with destinations to H_3, H_4 and H_5, respectively).

When R_8 finds out that an incoming packet is a unicast packet, *i.e.*, the most significant bit of the label is 0, it performs as described in the previous section. If an incoming packet is a multicast packet to H_3, H_4 and H_5, *i.e.*, the label of the incoming packet is 32780, R_8 makes a copy and changes the label to 34001 and sends it to R_9 via interface $s1$; makes another copy and removes the label and sends it to H_3 via interface s2. Note that, one more operation is needed to update the traffic volume counter pointed by the last column in Table 5-7, *i.e.*, TC_3, TC_4 and TC_5 in Table 5.6 are incremented by the size of the incoming packet.

In. Int	In. Lbl	O. Lbl	O. Int.	Des.
s0	524300	524400	$s1$	H_3, H_4
		–	$s2$	H_5

Table 5.7 One part of the forwarding table for multicast traffic in router R_8

Fig. 4.1 shows a logic configuration of a hose-modeled VPN. In the ISP's architecture, the physical link capacity on each link is greater than its value shown in Fig. 4.1. From the ISP's perspective, the proposed method is able to reduce the amount of traffic and hence makes better use of the ISP's bandwidth resource. For example, assume H_0 sends multicast packets to H_3, H_4 and H_5 at a constant rate of 1 Mbps; this is time critical traffic, which should be placed in $Queue1$ in Fig. 5.1. H_0 also sends unicast packets to H_4 and H_5 at a constant rate of 0.5 Mbps, respectively. H_1 sends unicast packets to H_4 and H_5 at a constant rate of 1 Mbps, and H_2 does the same as H_1. There is no other traffic. With the proposed scheme, at R_7, the bandwidth for traffic destined to H_4 and H_5 can be limited to 2 Mbps, respectively. Therefore, the remaining 2 Mbps bandwidth on links $R_7 - R_{10}$ and $R_{10} - R_8$ can be saved from this VPN and allocated to other traffic. Note that, without the proposed method, the bandwidth resource cannot be saved from this VPN.

5.6 Call Admission Control

As discussed in Chapter 4, without a proper call admission control scheme, the guaranteed or predictable QoS cannot be provided. Again, in this dissertation, it is assumed that the QoS metrics include two parameters: 1) bandwidth, and 2) latency.

With the computed fair share rate, the predictable bandwidth allocated to each hose flow can be provided, statistically. The latency of each hose flow, in each intermediate node, is determined by the size of $Queue1$ and the output link capacity of this node. With these and given propagation delays, the call admission control can

be performed. Note that, using the modified non-per-hose-flow-based scheme, the predictable bandwidth, but not the hard guaranteed bandwidth, can be provided. Therefore, those packets which require hard guaranteed service rate and latency are marked and placed in $Queue1$, provided that the aggregate arrival rate of these traffics is not greater than the output link capacity of this node. Other traffics should be placed in $Queue2$.

5.7 Summary

In this chapter, based on the non-per-flow-based packet fair queueing scheme [10], a much more scalable, non-per-hose-flow-based fair bandwidth allocation scheme has been proposed to approximate the idealized fluid bandwidth allocation scheme proposed in Chapter 3. The principle of the proposed scheme is to limit, on each intermediate node, the aggregate traffic of those flows to the same endpoint, according to the egress link capacity of this endpoint. As compared with the 2-D DRR and 2-D DRR+ schemes, the proposed scheme does not need to perform per-hose-flow-based queueing, and thus it is much more scalable than the 2-D DRR and 2-D DRR+ schemes. A modified secant method has also been proposed to compute the fair share rate. As compared with the method proposed in [10], the proposed, modified method can 1) approximate the exact fair share rate much more rapidly; 2) eventually reach the exact fair share rate, and 3) possess better performance in terms of controlling the queue size. This chapter also discusses how to integrate the proposed scheme within the framework of MPLS/VPN. Extension of the proposed scheme for multicast traffic has been investigated, and the proposed scheme has been demonstrated to be able to make better use of the ISP's network in terms of bandwidth and memory. Furthermore, to provide guaranteed QoS, call admission control issue and hierarchical VPN issue are briefly discussed.

Although the proposed scheme provides better performance, it requires the ISP to read the inner label, which is not required in the current framework of BGP/MPLS VPN. Besides, the forwarding table in each intermediate router in the ISP's domain requires some modifications. Based on the MP-BGP extension, it is also necessary to develop an approach to exchange information on the egress link capacities.

Finally, since the non-per-hose-flow-based scheme only provides soft-guaranteed QoS, it is necessary to study the statistical characteristics of allocated bandwidth of each hose flow and aggregation of hose flows, thus being able to predict the corresponding QoS performance.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter concludes the dissertation by 1) summarizing the contributions, 2) illustrating the basic limitations of the current solutions, and 3) discussing the directions for future work.

6.1 Contributions

With the development of the Internet, the current Internet service providers (ISPs) are required to offer revenue-generating and value-added services instead of only bandwidth and access services. VPN is one of the most important value-added services which can be provided by ISPs. Provider-provisioned VPN (PPVPN) service is provided by the Internet service provider, via network components, such as ISP backbones, provider edge routers and provider core routers, in the ISPs' cloud.

With the "classical" layer 2 VPN technologies, virtual circuits are created before traffic delivery. Since the bandwidth and buffers are partitioned accordingly, the QoS requirements can be naturally guaranteed. In the past few years, the layer 3 VPN technologies are widely deployed due to the desirable performance in terms of flexibility, scalability and simplicity. Since Layer 3 VPNs are built upon IP tunnels and IP is "best-effort" in nature, the QoS requirements cannot be guaranteed by the layer 3 VPNs. Thus, the layer 3 VPN services can only provide secure connectivity between gateways or hosts over public shared networks. This dissertation tries to shed some lights on how to provide guaranteed or predictable QoS, as with the layer 2 VPN technologies, while maintaining the flexibility and simplicity of the layer 3 VPN technologies. It also attempts to propose a mechanism which enables the VPN customers to manage their VPN resource. The major contributions of the dissertation are summarized as follows:

- The fluid hose-modeled VPN has been proposed. Based on the proposed model, an idealized fluid bandwidth allocation scheme has been developed. It is proven, analytically, the proposed bandwidth allocation scheme possesses the following desirable properties: 1) maximize the overall throughput of the VPN without compromising fairness; 2) provide a mechanism that enables the VPN customers to allocate the bandwidth according to their requirements by assigning different weights to different flows, thus obtaining the predictable QoS performance; and 3) improve the overall throughput of the ISPs' network.

- To approximate the proposed, idealized fluid fair bandwidth allocation scheme, the 2-dimensional deficit round robin (2-D DRR and 2-D DRR+) schemes have been proposed. Integration of the proposed schemes with the best-effort traffic within the framework of virtual-router-based VPN has been presented.

- To enhance the scalability, a framework with a more scalable non-per-flow-based scheme has been proposed. Integration of the proposed non-per-flow-based scheme within the framework of the MPLS VPN and applications for multicast traffics has been investigated.

- To compute the fair share rate more accurately, a modified secant method has been proposed. It has been demonstrated that the proposed method possesses better performance in terms of convergence and accuracy than the current method proposed in [10].

- Although the 2-D DRR and 2-D DRR+ schemes have been proposed to approximate the idealized fluid scheme in a hose-modeled VPN and provide guaranteed bandwidth to both individual flows and aggregation of flows, they can also be deployed when a "tiered" scheduling scheme is required. These schemes can be extended to multi-dimensional schemes and used to approximate H-GPS scheme.

6.2 Limitations

Although the hose-modeled VPN possesses the desirable properties in terms of scalability, flexibility, and multiplexing gain, it requires the prior knowledge of the traffic specification and needs a centralized mechanism to assign weights to different flows. Each intermediate router in the ISPs' domain has to maintain the information of each flow as well. Since provisioning in the hose-modeled VPN requires the centralized ISP management, it is not suitable for those short-term, highly dynamic VPN services, such as a VPN connection between a telecommuter or mobile user and the enterprise VPN.

In the fluid hose-modeled VPN, unicast traffics are assumed. However, there is an increasing need for the multicast VPN service. Thus, it is necessary to analyze the properties of this model and develop a general scheduling scheme with consideration of both unicast and multicast traffics.

With the proposed 2-D DRR and 2-D DRR+ schemes, since the queueing delay in each core node is bounded by the linear function of the size of service frame, the overall delay of any packet can be bounded. However, the delay bound for all packets belonging to the same hose flow is the same, although these packets may have different delay requirements. Thus, the schemes proposed in Chapter 4 may not meet these different delay requirements. Furthermore, although the scheduling implementation complexity of the 2-D DRR and 2-D DRR+ schemes is $O(1)$, they require to manage the queues on a per-hose-flow basis, which cannot be cost-effectively implemented when the number of hose flows is very large. Again, when the proposed 2-D DRR and 2-D DRR+ schemes are deployed in other applications, the scheduler requires the prior knowledge of the scheduling constraints, *i.e.*, which flows belong to the same aggregate.

When the non-per-hose-flow-based scheme is used, the property of the queueing delay cannot meet the different delay requirements either. Another issue is that, when

this scheme is deployed, it is assumed that an incoming packet is not dropped before it arrives the output queue. However, in a switching fabric, a packet could be dropped due to contention before it arrives its output queue. Finally, since the scheduler drops incoming packets with certain probability, the non-per-hose-flow-based scheme can only provides predictable QoS performance instead of the hard guaranteed QoS performance.

6.3 Future Work

Since, currently, most enterprises use the hub-and-spoke VPN topology instead of the hose-model topology, it is necessary to develop a scalable scheduling and admission control scheme for the hub-and-spoke-based VPN. With the increasing need for building connections between telecommuters or mobile users and enterprise headquarters, new approaches are required to provision short-term, highly dynamic VPN connections, and the corresponding scheduling scheme, which does not require prior knowledge of the scheduling constraints, should be developed. Finally, since the non-per-hose-flow-based scheme only provides soft-guaranteed QoS, it is necessary to study the statistical characteristics of the allocated bandwidth of each hose flow and aggregation of hose flows, in order to be able to predict the corresponding QoS performance.

REFERENCES

[1] M. Morrow and K. Vijayananda, *Developing IP-Based Services - Solutions for Service Providers and Vendors.* Morgan Kaufmann, 1st ed., 2002.

[2] IANA, *http://www.iana.org.* IANA, 1996.

[3] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, *Address Allocation for Private Internets.* http://www.ietf.org/rfc/rfc1918.txt: IETF, 1996.

[4] J. Buckwalter, *Frame Relay: Technology and Practice.* Addison-Wesley Pub Co., 1st ed., 1999.

[5] U. Black, *ATM, Volume I: Foundation for Broadband Networks.* Prentice Hall PTR, 2nd ed., 1999.

[6] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. van der Merwe, "Resource management with hoses: Point-to-cloud services for virtual private networks," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 679 –692, 2002.

[7] E. Rosen and Y. Rekhter, *BGP/MPLS VPNs.* http://www.ietf.org/rfc/rfc2547.txt: IETF, 1999.

[8] E. Rosen, Y. Rekhter, T. Bogovic, R. Vaidyanathan, S. Brannon, M. Carugi, C. Chase, L. Fang, T. Chung, J. De Clercq, E. Dean, P. Hitchen, A. Smith, M. Leelanivas, D. Marshall, L. Martini, M. Morrow, V. Srinivasan, and A. Vedrenne, *BGP/MPLS VPNs.* IETF, 2002.

[9] P. Knight, H. Ould-Brahim, G. Wright, B. Gleeson, T. Sloane, R. Bubenik, C. Sargor, I. Negusse, and J. Yu, *Network Based IP VPN Architecture Using Virtual Routers.* IETF, 2003.

[10] I. Stoica, *Stateless Core: A Scalable Approach for Quality of Service in the Internet.* Pittsburg, PA: Ph.D dissertation, Carnegie Mellon University, 2000.

[11] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis, *A Framework for IP Based Virtual Private Networks.* http://www.ietf.org/rfc/rfc2764.txt: IETF, 2000.

[12] I. Pepelnjak and J. Guichard, *MPLS and VPN architectures, vol. 1.* CISCO Press, 1st ed., 2000.

[13] I. Pepelnjak and J. Guichard, *MPLS and VPN architectures, vol. 2.* CISCO Press, 1st ed., 2003.

[14] IETF, *http://www.ietf.org/html.charters/ppvpn-charter.html.* IETF, 1996.

[15] L. Andersson and T. Madsen, *PPVPN Terminology.* IETF, 2001.

[16] S. Shenker, C. Partridge, and R. Guerin, *Specification of Guaranteed Quality of Service.* http://www.ietf.org/rfc/rfc2212.txt: IETF, 1997.

[17] F. Chiussi, J. De Clercq, S. Ganti, W. Lau, B. Nandy, N. Seddigh, and S. Van den Bosch, *Framework for QoS in Provider-Provisioned VPNs.* IETF, 2001.

[18] A. Nagarajan, *Generic Requirements for Provider Provisioned VPN.* IETF, 2001.

[19] M. Kaeo, *Design Network Security.* CISCO Press, 1st ed., 1999.

[20] W. Stallings, *Data and Computer Communications.* Prentice Hall PTR, 5th ed., 1996.

[21] B. Kent and A. R., *Security Architecture for the Internet Protocol.* http://www.ietf.org/rfc/rfc2401.txt: IETF, 1998.

[22] V. A. Rosen, E. and R. Callon, *Multiprotocol Label Switching Architecture.* http://www.ietf.org/rfc/rfc3031.txt: IETF, 2001.

[23] IETF, *http://www.ietf.org/html.charters/mpls-charter.html.* IETF, 1996.

[24] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (BGP-4).* http://www.ietf.org/rfc/rfc1771.txt: IETF, 1995.

[25] B. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, 1988.

[26] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *Proceedings of ACM SIGCOMM'99*, pp. 251–262, 1999.

[27] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener, "Algorithms for provisioning virtual private networks in the hose model," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 565–578, 2002.

[28] G. Italiano, R. Rastogi, and B. Yener, "Restoration algorithms for virtual private networks in the hose model," *Proceedings of IEEE INFOCOM 2002*, vol. 1, pp. 131 –139, 2002.

[29] P. V. and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, 1995.

[30] R. Braden, L. Zhang, Berson, H. S., S., and S. Jamin, *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification.* http://www.ietf.org/rfc/rfc2205.txt: IETF, 1997.

[31] S. S., R. Braden, and D. Clark, *Integrated Services in the Internet Architecture: An Overview.* http://www.ietf.org/rfc/rfc1633.txt: IETF, 1994.

[32] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Service.* http://www.ietf.org/rfc/rfc2475.txt: IETF, 1998.

[33] A. Arulambalam, X. Chen, and N. Ansari, "Allocating fair rates for available bit rate service in atm networks," *IEEE Communication Magazine*, vol. 34, pp. 92–100, 1996.

[34] S. Keshev, *An Engineering Approach to Computer Networking : ATM Networks, the Internet, and the Telephone Network.* Reading, MA: Addison-Wesley Pub Co., 1st ed., 1997.

[35] J. Chao and X. Guo, *Quality of Service Control in High-Speed Networks.* John Wiley Sons, 1st ed., 2001.

[36] J. Nagle, "On packet switches with infinite storage," *IEEE Transactions on Communication*, vol. 35, pp. 435–438, 1987.

[37] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose atm switch chip," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1265–1279, 1991.

[38] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," *ACM/IEEE Transactions on Networking*, vol. 4, pp. 375–385, 1996.

[39] A. Parekh, *General Processor Sharing Scheme.* Boston, MA: Ph.D dissertation, MIT, 1991.

[40] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queuing algorithm," *Internetworking: Research and Experience*, vol. 1, pp. 3–26, 1990.

[41] J. Bennett and H. Zhang, "wf^2q: Worst-case fair weighted fair queueing," *Proceedings of IEEE INFOCOM'96*, pp. 120–128, 1996.

[42] J. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 675–689, 1997.

[43] *The Network Simulator - ns-2.* 1996.

[44] A. Francini, F. Chiussi, R. Clancy, K. Drucker, and N. Idirene, "Enhanced weighted round robin schedulers for accurate bandwidth distribution in packet networks," *Computer Networks*, vol. 37, pp. 561–578, 1999.

[45] H. Adiseshu, G. Parulkar, and G. Varghese, "A reliable and scalable striping protocol," *Proceedings of ACM SIGCOMM'96*, vol. 26, pp. 131–141, 1996.

[46] D. Stiliadis, *Traffic Scheduling in Packet-switched Networks: Analysis, Design and Implementation.* Santa Cruz, California: Ph.D dissertation. University of California, Santa Cruz, 1996.

[47] A. Tanenbaum, *Computer Networks.* Prentice Hall, 3rd ed., 1996.

[48] W. Stevens, *TCP/IP Illustrated, Volume 1 - The Protocols.* Addison-Wesley Pub Co, 1st ed., 1993.

[49] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, 1993.

[50] B. Fox and B. Gleeson, *Virtual Private Networks Identifier.* http://www.ietf.org/rfc/rfc2685.txt: IETF, 1999.

[51] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks," *Proceedings of ACM SIGCOMM'98*, pp. 118–130, 1998.

[52] D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, 1998.

[53] J. Hoffman, *Numerical Methods for Engineers and Scientists.* Marcel Dekker Inc., 2nd ed., 2001.

[54] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, and F. Ansari, *Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment.* http://www.ietf.org/rfc/rfc3353.txt: IETF, 2002.