Queuing Model Based End-to-End Performance Evaluation for MPLS Virtual Private Networks

Yanfeng Zhu, Yibo Zhang, Chun Ying, and Wei Lu

IBM China Research Laboratory
Email: zyfeng@cn.ibm.com, zhangyib@cn.ibm.com, yingchun@cn.ibm.com, luw@cn.ibm.com

***Abstract*—Monitoring the end-to-end Quality-of-Service (QoS) is an important work for service providers' Operation Support System (OSS), because it is the fundamental requirement for QoS provisioning. However, it is in fact a challenging work, and there are few efficient approaches to address it. In this paper, for Multi-Protocol Label Switching (MPLS) Virtual Private Networks (VPNs), we propose a queuing model based end-to-end performance evaluation scheme for OSS to monitor the end-to-end delay, which is one of the important QoS metrics. By means of a queuing model, we deduce the relationship between the end-to-end delay and the information available in the Management Information Base (MIB) of routers, and then we present the evaluation scheme which avoids the costly per-packet measurement. The complexity of the proposed scheme is much lower than existing schemes. Extensive simulation results show that the proposed scheme can efficiently evaluate the end-to-end performance metrics (the estimation error is nearly 10%).**

I. Introduction

Multi-Protocol Label Switching (MPLS) Virtual Private Network (VPN) service [1], which allows customers to connect several sites of their network over a provider owned network cloud, has emerged as an important source of revenue (15 billion dollars in 2007 [2]) for the service providers (SPs) of IP network. From the statistical data in [2], the market of MPLS VPN increases 38% per year, and a number of SPs have emerged in order to make profits from the developing trend. To better serve and attract customers, the SPs imminently hope to develop a method to monitor the end-to-end Quality-of-Service (QoS) that customers are experiencing, which is much important for Operation Support System (OSS) in service provisioning [8] and admission control [9].

In this paper, the definition of end-to-end is that from the enter point of the core network to the exit point, rather than the concept of host-to-host. SPs only provide the network services of connecting hosts, just like provide a pipeline between hosts, and thus SPs are responsible for providing the QoS guarantee of the pipeline only. Moreover, hosts are in fact out of the management scope of OSS, OSS does not have the right to monitor their performance without the admission of hosts. Generally speaking, the QoS is defined by a set of performance metrics, which include throughput, packet loss rate and delay etc. The throughput can be measured directly in the router, and it has been a basic function of many network management software, for example IBM Tivoli/Netcool product [18] and open source software Cacti [4]. Some academic work about the optimization of the throughput monitoring can be found in [7][10][16]. Per-router packet loss rate can be obtained by analyzing the MIB information in each router: the number of discard packets and total packets, and then the end-to-end packet loss rate can be inferred with help of path information [7], which can be obtained from VPN route forwarding tables (VRFs) and Label Switching Path (LSP) [1]. However, for the measurement of end-to-end delay, which is much important to evaluate the quality of delay-sensitive service, there are no cost-efficient and accurate approaches still in both products and literature.

Currently, there are two types of approaches to estimate the end-to-end delay: one is to enable NetFlow [19] for per-packet monitoring; the other is to use management packets for end-to-end detecting, for example, ping command based on ICMP. The former is a direct measurement method, which positively detects all packets at the source and destination, and can be get an relatively[1] accurate end-to-end delay. However, much practical operating experience shows that NetFlow is a costly tool and usually occupies up to 40% system resource of routers in CPU and memory utilization. Therefore, the SPs enable NetFlow only at setup or test stage of the network, and disable it at runtime stage. A similar costly tool can be found in [6]. Alternatively, ping brings less cost in system resource of routers than Netflow, but the measured end-to-end delay ping is not accurate enough due to the following reasons:

1) Ping packets and practical data packets may go through different paths due to the best effect property of routing protocol.
2) Ping packets are management packets, and distinguish with practical data packets in packet length and priority, which results in different sojourn time in routers.
3) Ping packets are still add-on traffic, and the overhead to the network traffic will influence the end-to-end delay of the measured traffic seriously when lots of periodically end-to-end monitoring operate simultaneously.

To enhance the accuracy of ping, a MPLS ping [5] is proposed to make ping packets share the same path as data packets. MPLS ping packets are encapsulated in a data packet format, and as a result they are operated as practical data packets in routers. Therefore, the MPLS ping alleviates the inaccuracy due to the first two reasons list above. We made a

[1]The accurate delay is built on the assumption that the clock of all routers are synchronized.

978-1-4244-3487-9/09/$25.00 © 2009 IEEE

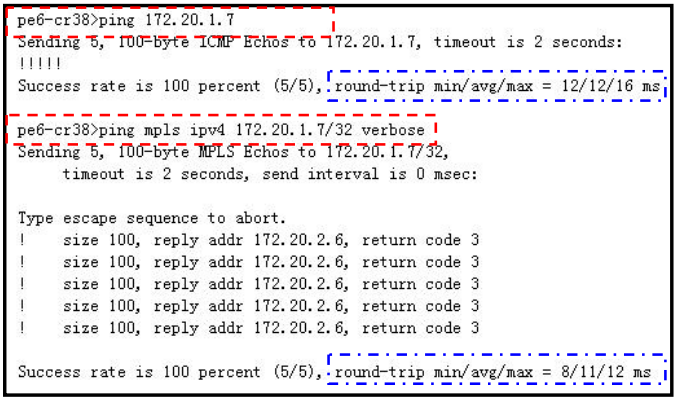


Fig. 1. Comparison of traditional ping and MPLS ping

simple experiment based on the IBM Tivoli/Netcool Southbank testbed in UK to estimate the end-to-end delay with traditional ping and MPLS ping, respectively. Fig.1 provides a comparison result for the traditional ping and the MPLS ping in estimating end-to-end delay. It is observed that they results in different estimations, which infers that the inefficiency of traditional ping. However, the MPLS ping is still a costly tool, especially when lots of periodically end-to-end monitoring operate simultaneously in the core networks. For example, for a core network with 1000 edge routers, there will be C^2_{1000} end-to-end pairs, and C^2_{1000} MPLS packets will be sent per estimation period. Therefore, it is impossible for OSS to achieve a frequent end-to-end delay estimation. In fact, both ping and MPLS ping are designed to evaluate the reachability of the networks, but they are not the efficient tools for estimating end-to-end delay.

In this paper, we consider an indirect approach to estimate the end-to-end delay according to the information available in the management information base (MIB) of routers. Most of management software of OSS, such as IBM Tivoli/Netcool products, are configured to periodically collect the MIB information with probes installed in routers for performance monitoring: payload, throughput and discard etc. We propose a queuing model to build the relationship between the per-hop delay and the MIB information. Then, combining the per-hop results with the LSP information, we can compute the end-to-end delay. Such an indirect approach does not induce an additional network traffic to links and system resource cost in routers. The cost is mainly in the model computation and moved to in OSS management server. We will show that the computation complexity is linear with the number of routers in the core networks. Based on the queuing model, we develop a scheme to estimate the end-to-end delay at run-time. Finally, NS-2 based simulations are given to validate the efficiency of the proposed scheme.

The rest of the paper is organized as follows. Section II gives a brief overview to MPLS VPNs. In Section III, we present a queueing model based analytical model to investigate the QoS metrics, and an evaluation scheme and the complexity analysis are given as well. Section IV shows the simulation results for performance validation. Section V concludes the paper.

II. MPLS VPN Overview

The VPN service, which is connection-oriented, enables customers to connect their sites geographically dispersed but belonging to the same authority over the core network of a VPN SP, and thus creating the appearance of a seamless intranet. A typical application of VPN service is to connect branch offices belonging to an enterprise. Before the appearance of MPLS, customers implement the VPN service by leasing layer 2 links (ATM or FrameRelay), which is considered to be much expensive and not scalable.

MPLS VPNs are created in Layer 3 and are based on the peer model, which makes them more scalable and easier to be built and managed than conventional VPNs. The MPLS VPN is a true peer VPN model that enforces traffic separations by assigning unique VPN route forwarding tables (VRFs) to each customer's VPN. Thus, users in a specific VPN cannot see traffic outside their VPN. The SP's core network is comprised of the provider edge routers and its provider routers. The OSS of the SP can get the routing information about a particular VPN by checking the Label Switching Path (LSP) in provider edge routers that attach to that VPN.

From [2], MPLS VPNs provide the following benefits:

- Privacy and security equal to Layer-2 VPNs by constraining the distribution of a VPN's routes to only those routers that are members of that VPN, and by using MPLS for forwarding;
- Seamless integration with customer intranets;
- Increased scalability with thousands of sites per VPN and hundreds of thousands of VPNs per service provider;
- Easy management of VPN membership and rapid deployment of new VPNs.

A typical MPLS VPN network is shown in Fig.2. The main functionalities of each type of router are summarized below:

- **Provider edge routers (PEs):** which are managed by the SP, serve as the customers' entry and exit points for the VPN. PE's setup and maintain LSP's among PE's by label distribution protocol.
- **Customer edge routers (CEs):** which are associated with customer sites and are usually managed by the customers. CE's are responsible for forwarding the VPN traffic of customers to the associated PE's.
- **Provider routers (Ps):** which are managed by the SP, are mainly responsible for forwarding the VPN traffic encapsulated in MPLS frames by PE's.

When the VPN service is set, an LSP between the PE nodes is automatically established on the IP topology of the P node network. As shown in Fig.2, LSP1 between PE1 and PE2 is established to connect two sites of VPN1, and LSP2 between PE1 and PE3 is established to connect two sites of VPN2. After the setup of LSP is completed, the transmission of the data using label switching within the established LSP starts. Consequently, the VPN service in the provider's network can be managed in a secure and connection-oriented mode.

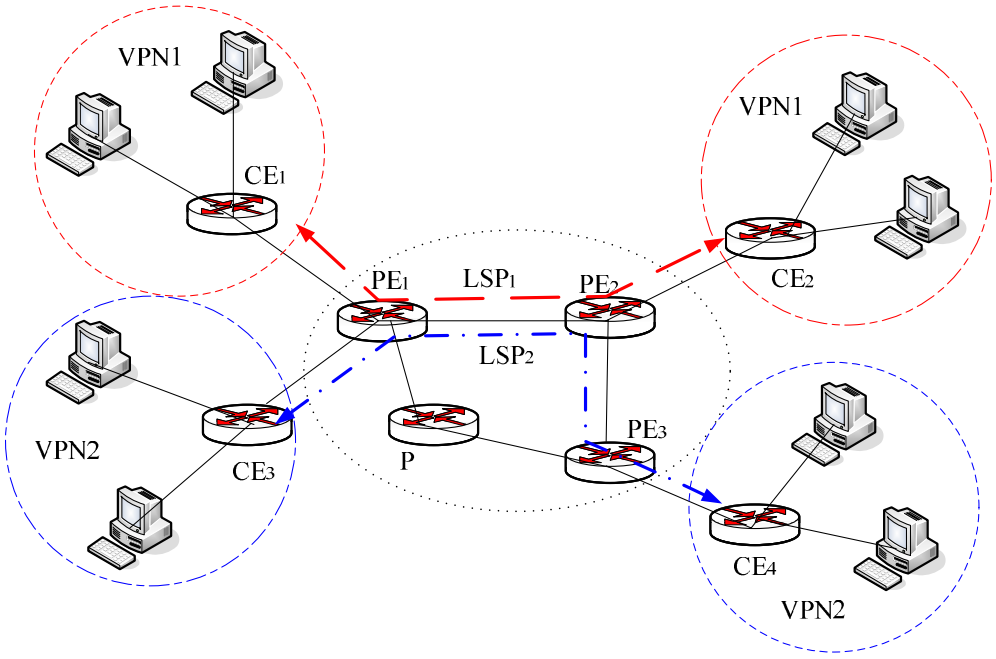


Fig. 2. A typical MPLS VPN network

In MPLS VPN, the management domain of the OSS is often restricted in PEs and Ps, and thus the OSS need to evaluate the performance users experiencing by monitoring PEs and Ps only. Most of OSS products [18] can positively collect the MIB information of routers by deploying probes in PEs and Ps: load, queue size, CPU and memory utilization etc. In this paper, we concentrate on the method of estimating the end-to-end delay of MPLS VPN with the information collected from MIB of PEs and Ps.

III. Queueing Model Based Analysis and Evaluation Scheme

In this section, we develop a simple queuing model to analyze the end-to-end performance in packet loss rate and delay, and then a performance evaluation scheme is presented for the runtime operation of OSS. Herein, the concept of "end-to-end" is from the point that customers' traffic enters the core network to the point that leaving the core network (i.e., from PE's to PE's) in layer 2 or layer 3, rather than that in layer 4 (transport layer). This is because the protocol stack of layer 4 is out of the management scope of OSS.

A. Analytical Model

Each router has multiple independent ports, and each port is connected to a forwarding link. From [19], output queuing is widely employed in current products. In output queuing system, shown as Fig.3 each port maintains a buffer, and all packets routed to the port will be stored in the buffer of the port before forwarding. Correspondingly, we can treat each port as an independent queuing system. Although weighted-fair-queuing (WFQ)[13], which provides independent buffers for data packets with different quality labels, is widely supported

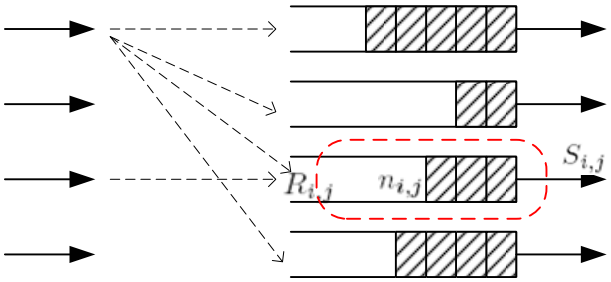


Fig. 3. Queuing model within a router

by various routers, we only consider first-in-first-out (FIFO) queuing in this paper due to two reasons. The first reason is about the complexity of the analytical model. Much research work has proved that it is very difficult to analyze WFQ, and there is no systematic theory about WFQ. Differently, much work about FIFO has provided lots of mature analytical method. The other reason is that most of traffic in current network is configured as the same quality (best-effort), i.e., in most cases, only one buffer is active. Therefore, we model the serving process as a simple FIFO serving process with a single queue.

As shown in Fig.3, we concentrate on the queuing system in a port of a router. We index all routers within the core network by $R_i, i = 1, 2, ...M$, where M is the total number of routers, and the jth port of R_i is represented by $R_{i,j}$. From [19], output queuing, which implies that the arrived packets are first switched to output port and buffered there for transmissions, is widely employed in practical devices. Therefore, the number of buffers of one router is equal to

the number of ports. Let $N_{i,j}$ denote the buffer size of $R_{i,j}$, where the unit is packet. At time t, the number of packets buffered in $R_{i,j}$ is denoted variable $n_{i,j}$. Let $S_{i,j}$ denote the transmission rate of port $R_{i,j}$, which is also the data rate of the link connected to the port. Then, if we assume the length of all packets transmitted in the core network is independently and identically distributed (i.i.d), given the packet length l, the sojourn time of a data packet in $R_{i,j}$ can be expressed as a function $d(n_{i,j}, l, S_{i,j})$, which consists of queuing delay, processing delay[2] and transmission delay.

Given a pair of CEs within the same VPN, the route between them can be obtained by checking the VRF table and the related LSPs. In the following analytical model, we take one pair of CEs for example to compute the QoS metrics, and the route between them is indicated by the corresponding LSP. The considered LSP corresponds to a set of router ports that the data pacekts of the selected CEs will pass through. Let Φ_{LSP} denote the set of router ports corresponding to the considered LSP. Then, the end-to-end delay in the core network is given by

$$d_{\text{LSP}} = \sum_{R_{i,j} \in \Phi_{\text{LSP}}} d(n_{i,j}, l, S_{i,j}) + \theta_{\text{LSP}} \tag{1}$$

where θ_{LSP} is the propagation delay (the physical link distance divided by the electronic speed).

In the queuing theory, it is a classic problem to compute the sojourn time with the arrival process and serving process given. Unfortunately, the traffic in the network is usually self-similar [14]. The authors in [15] showed that it is very difficult to compute the queuing model with self-similar arrival process. In this paper, we concentrate on how to use the statistical data stored in the MIB of routers (can be obtained directly by Netcool) to estimate the QoS metrics. In most of routers and switches, the periodic statistics to packet loss rate ($p_{i,j}$), mean queue length ($E[n_{i,j}]$), and utilization of a port $\rho_{i,j}$ are available in the MIB. It is straightforward to get the end-to-end packet loss rate by $p = 1 - \prod_{R_{i,j} \in \Phi_{\text{LSP}}} (1 - p_{i,j})$.

From the queuing theory, the packet loss rate at $R_{i,j}$, which is equal to the probability that there are $N_{i,j}$ packets in the buffer, is given by

$$p_{i,j} = P(n_{i,j} = N_{i,j}) \tag{2}$$

and the mean queue length is given by

$$\begin{aligned} E[n_{i,j}] &= \sum_{k=1}^{N_{i,j}} kP(n_{i,j} = k) \\ &= N_{i,j}p_{i,j} + \sum_{k=1}^{N_{i,j}-1} kP(n_{i,j} = k) \end{aligned} \tag{3}$$

Therefore, under the condition that the new packet can enter the buffer of $R_{i,j}$, the observed mean buffer length should be given by

$$E^*[n_{i,j}] = E[n_{i,j}] - N_{i,j}p_{i,j} \tag{4}$$

In addition, the utilization of the port is given by

$$\rho_{i,j} = \frac{\lambda_{i,j} S_{i,j}}{E[l]} \tag{5}$$

As a consequence, we have

$$\lambda_{i,j} = \frac{\rho_{i,j} E[l]}{S_{i,j}} \tag{6}$$

From the Little's formula, given the arrival rate $\lambda_{i,j}$ and $E^*[n_{i,j}]$, the mean sojourn time is given by $\frac{E^*[n_{i,j}]}{\lambda_{i,j}}$, i.e.,

$$d(n_{i,j}, l, S_{i,j}) = \frac{S_{i,j}(E[n_{i,j}] - N_{i,j}p_{i,j})}{\rho_{i,j} E[l]} \tag{7}$$

In the expression above, all parameters can be retrieved from the MIB of routers directly, thus it is measurable on-line. Substituting (7) into (1), we can compute the end-to-end delay with

$$d_{\text{LSP}} = \sum_{R_{i,j} \in \Phi_{\text{LSP}}} \frac{S_{i,j}(E[n_{i,j}] - N_{i,j}p_{i,j})}{\rho_{i,j} E[l]} + \theta_{\text{LSP}} \tag{8}$$

B. Evaluation Scheme for Runtime Operation

In the actual routers, the value of packet loss rate ($p_{i,j}$), mean queue length ($E[n_{i,j}]$) and the utilization of a port $\rho_{i,j}$ are collected at intervals, and the interval varies with the setting of routers. In our work, the interval is 15 minutes, which is a typical collection interval in OSS product (Tivoli/Netcool product[18]). During the kth interval, the statistics of routers, which are represented by $\{p_{i,j}^{(k)}, E^{(k)}[n_{i,j}], \rho_{i,j}^{(k)}\}$, are obtained directly by checking the MIB of routers. Then, the parameters employed to compute the end-to-end delay are updated with

$$\begin{cases} p_{i,j} = \gamma p_{i,j} + (1-\gamma) p_{i,j}^{(k)} \\ E[n_{i,j}] = \gamma E[n_{i,j}] + (1-\gamma) E^{(k)}[n_{i,j}] \\ \rho_{i,j} = \gamma \rho_{i,j} + (1-\gamma) \rho_{i,j}^{(k)} \end{cases} \tag{9}$$

where γ is a smooth factor, which is widely adopted in network protocols to obtain reliable estimates. The selection of γ need to consider the compromise between accuracy and promptness. The detailed selection scheme and impact evaluation is out of the scope of this paper due to the limited space, and γ is set to 0.5 in our work.

After getting $\{p_{i,j}, E[n_{i,j}], \rho_{i,j}\}$, we can compute the end-to-end packet loss rate and the end-to-end delay by

$$\begin{cases} p = 1 - \prod_{R_{i,j} \in \Phi_{\text{LSP}}} (1 - p_{i,j}) \\ d_{\text{LSP}} = \sum_{R_{i,j} \in \Phi_{\text{LSP}}} \frac{S_{i,j}(E[n_{i,j}] - N_{i,j}p_{i,j})}{\rho_{i,j} E[l]} + \theta_{\text{LSP}} \end{cases} \tag{10}$$

Here, Φ_{LSP} is retrieved from the LSP table of PE directly, and θ_{LSP} is estimated according to the geographic distance of VPN sites[3].

[2]The processing delay is due to checking routing table and internal polling, and it is usually a fixed delay and much smaller than the queuing delay and the transmission delay. Therefore, in the analytical model, we ignore the processing delay directly.

[3]In the simulation work, we ignore θ_{LSP}, because it is a fixed value and nothing to do with our evaluation scheme.

C. Complexity Analysis

From the description above, we can find that the complexity of the proposed scheme in information collection is $O(M)$ in a statistical interval, because the OSS needs to collect $\{p_{i,j}, E[n_{i,j}], \rho_{i,j}\}$ from every port of all routers in the core networks. The complexity of computation is in proportion to the number of LSPs, i.e., $O(M_C^2)$ where M_C is the number of CEs. For a typical performance evaluation scheme proposed in [7], the complexity of information collection is $O(M^2)$, which is in proportion to the number of links, and the complexity of computation is $O(M^2 M_C^2)$. Therefore, the proposed scheme has lower complexity in both information collection and computation, and the price is a little estimation error which will be indicated in the next section.

In addition, it is worth noting the complexity of MPLS ping. Intuitively, MPLS ping does not bring any computation complexity, but the MPLS ping packets bring too much overhead to the transmission links. As indicated in Fig.1, the size of a MPLS ping packet is 100 bytes, and 5 MPLS ping packets are sent in each ping. In a round of MPLS ping, the total number of MPLS pings is $C_{M_C}^2$. As a result, the overhead to the transmission links should be $5 \times 100 \times 8 \times C_{M_C}^2$. If we want to employ the same evaluation interval (15 minutes), the overhead brought by MPLS pings approaches to $4.4C_{M_C}^2$. When the number of CEs is large, the MPLS pings will be so hug overhead to the transmission links that the utilization and packet loss rate of each port will be influenced seriously. Correspondingly, the estimation results by MPLS pings are much different from the situation without MPLS pings.

IV. Performance Validation

In this section, we take a simple MPLS VPN for example to simulate the evaluation effect of the proposed scheme. The simulation topology is the same to that shown in Fig.2. Because all routers are duplex, we only configure uni-directional traffic in both VPN's (from CE1 to CE2 in VPN1 and from CE3 to CE4 in VPN2). Additional background traffics are configured to links those from PE2 to CE2 and from PE3 to CE4 so as to increase the traffic load in both PE2 and PE3. The data rate of all link among PE's is 100 Mbps.

In each link, a stable background traffic is configured with the load shown in Table I. The traffic loads for both VPN1 and VPN2 are configured in "on-off" mode, and the traffic load in the "on" state is twice that in "off" state. The loads in both "on" and "off" modes are exponentially distributed, and the mean traffic load in "on" mode are listed in Table I. The intervals of the "on" and "off" modes are also exponentially distributed with the same mean interval 10 minutes.

To investigate the impact of the service time on the proposed scheme, we consider fixed packet length and varied packet length, respectively. In the scenario with fixed packet length, the packet length is fixed to 80 kb, and in the scenario with varied packet length, the packet length is exponentially distributed with the mean length as 80 kb. The buffer size is set to 100 packets.

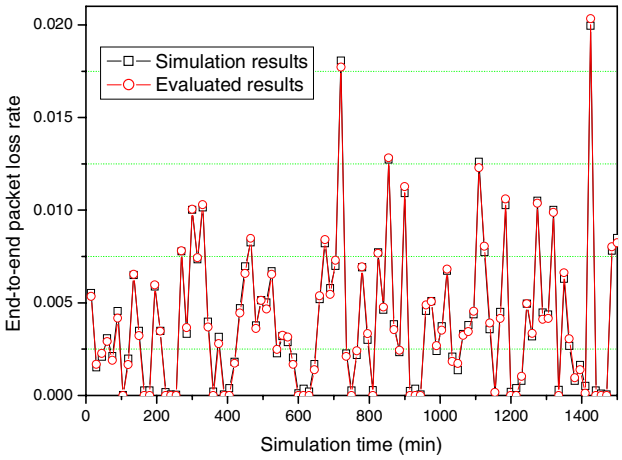


Fig. 4. End-to-end packet loss rate comparison for high traffic load

TABLE I
Traffic Configuration

Scenario	VPN1 (Mbps)	VPN2 (Mbps)	Background (Mbps)
medium load	22	22	22
high load	35	35	35

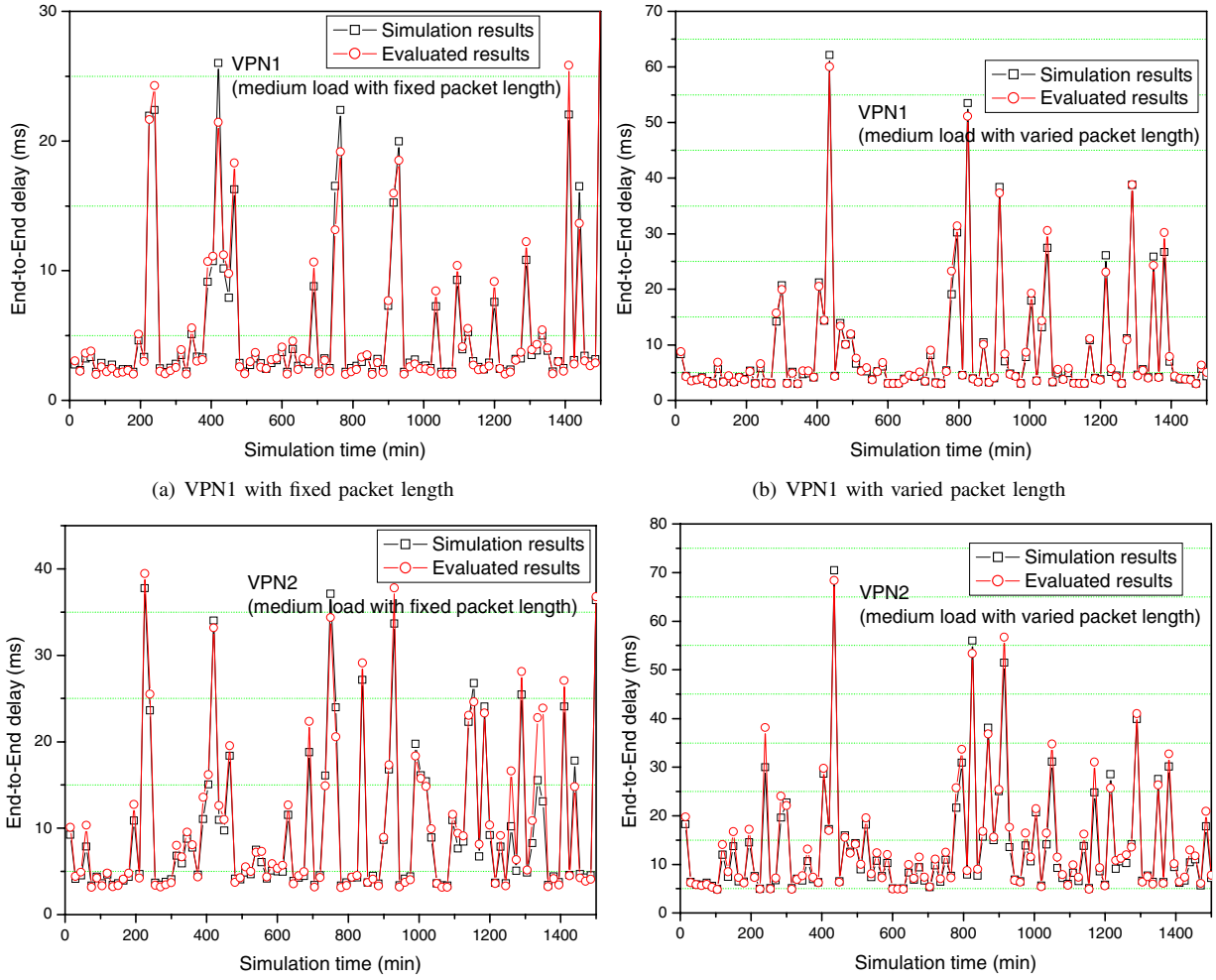
We first investigate the evaluation effect of packet loss rate. The results are shown in Fig.4. Here, the simulation results refer to the results got by end-to-end monitoring in the simulations, and the evaluated results refer to those got by the evaluation scheme developed in the last section. We only give the results based on high traffic load because the packet loss rate approaches to zero in the scenario with medium traffic load. From the figure, it is observed that the evaluated results approach to the simulation results very well. This is mainly because the end-to-end packet loss rate is estimated with the per-router packet loss rate, which can be obtained directly by checking the MIB.

For the performance evaluation on the end-to-end delay, we simulate the scenarios with medium traffic load and high traffic load, respectively. The end-to-end delays of both VPN1 and VPN2, which have different path length, are collected.

Fig.5 shows the results based on medium traffic load. Firstly, it is observed that the estimation results based on the proposed schemes can approach to the simulation results very well in both VPN1 and VPN2. Secondly, we get an insight that the distribution of packet length influences the end-to-end delay seriously. Therefore, it is efficient that shaping the packet length by some encapsulation technologies in CE's before entering the core network.

For the scenario with high traffic load, we can find the similar results (Fig.6) as those got under the scenario with medium traffic load. Although the higher traffic load increases the end-to-end delay obviously, our scheme can get a proper evaluation still.

To better understand the evaluation effect, we define an



(a) VPN1 with fixed packet length

(b) VPN1 with varied packet length

(c) VPN2 with fixed packet length

(d) VPN2 with varied packet length

Fig. 5. End-to-end delay comparison for medium traffic load

estimation error as follow:

$$\text{Estimation error} = \text{E}\left[\frac{\text{simulation result} - \text{evaluated result}}{\text{simulation result}}\right] \tag{11}$$

The estimation errors of all scenarios are listed in Table.II. From the table, it is observed that the error is nearly 10%, which is acceptable for the OSS.

TABLE II
ESTIMATION ERROR

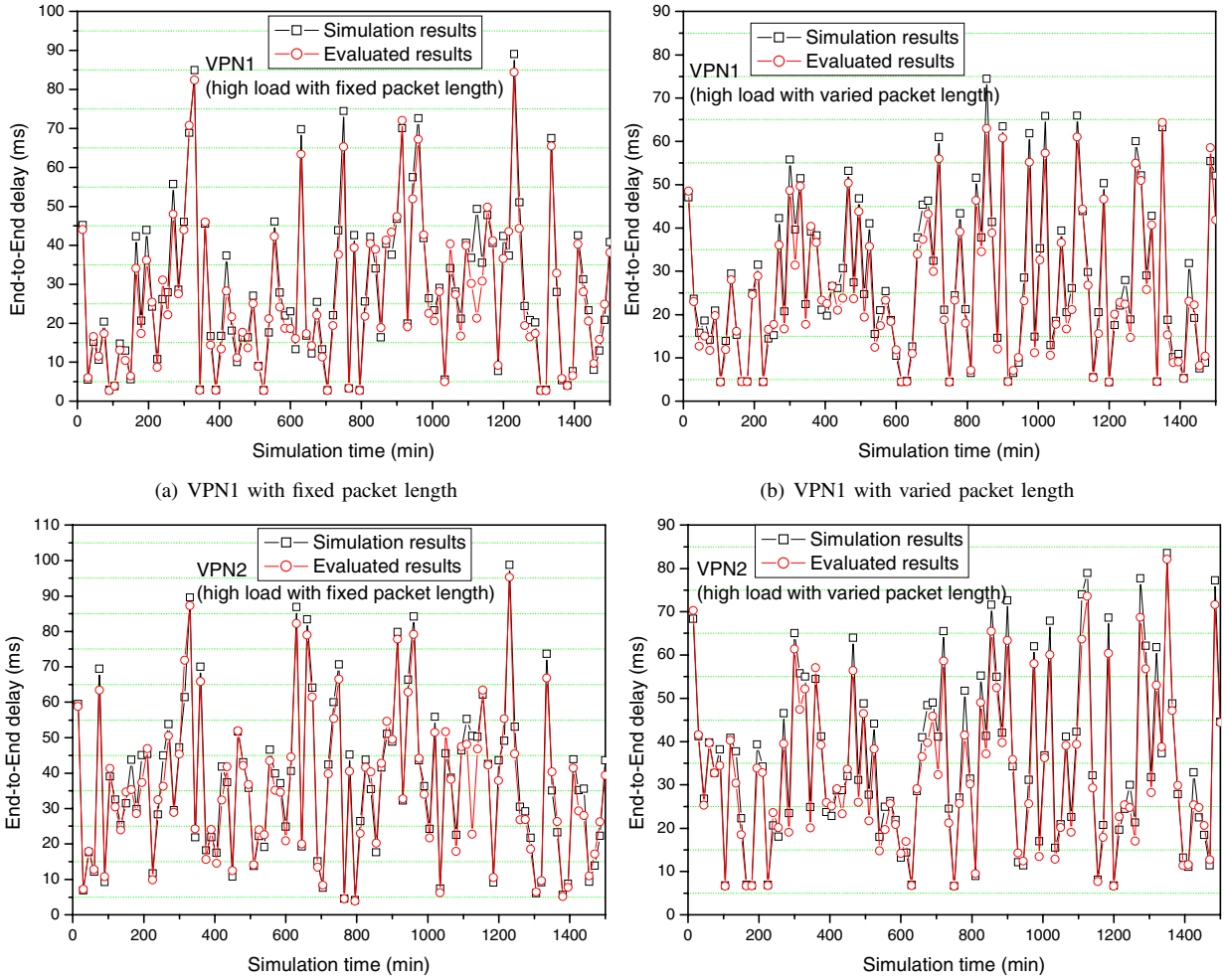
Scenario	Error (%)
VPN1, medium load with fixed packet length	9.4
VPN2, medium load with fixed packet length	10.7
VPN1, medium load with varied packet length	5.4
VPN2, medium load with varied packet length	9.14
VPN1, high load with fixed packet length	9.6
VPN2, high load with fixed packet length	11.4
VPN1, high load with varied packet length	9.6
VPN2, high load with varied packet length	10.6

V. CONCLUSION

In this paper, we concentrated on the problem of end-to-end performance evaluation in MPLS VPN, which is a key problem for the OSS. By means of a queuing model, we deduced the relationship between the end-to-end performance metrics (packet loss rate and delay) and the information available in the MIB of routers, and then proposed an evaluation scheme for the end-to-end performance metrics. Extensive simulation results showed that the proposed scheme can efficiently evaluate the end-to-end performance metrics without per-packet measurement.

REFERENCES

[1] RFC 4364, BGP/IP Virtual Private Networks (VPNs), Feb. 2006.
[2] Infonetics research report, Ehernet & IP MPLS VPN Services: Annual Market Size & Forecasts, May 2, 2007.
[3] N. G. Duffield, Pawan Goyal, Albert Greenberg, etc, "Resource Management With Hoses: Point-to-Cloud Services for Virtual Private Networks", IEEE/ACM Transactions on Networking, vol. 10, no. 5, pp. 679-692, Oct. 2002.
[4] http://www.cacti.net.
[5] http://www.nanog.org/mtg-0501/pdf/moizuddin.pdf.



(a) VPN1 with fixed packet length

(b) VPN1 with varied packet length

(c) VPN2 with fixed packet length

(d) VPN2 with varied packet length

Fig. 6. End-to-end delay comparison for high traffic load

[6] http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a0080519a11.html#wp1051265.

[7] Satish Raghunath, K. K. Ramakrishnan, and Shivkumar Kalyanaraman, "Measurement-Based Characterization of IP VPNs", IEEE/ACM Transactions on Networking, vol. 15, no. 6, pp. 1428-1441, Dec. 2007.

[8] Dongli Zhang and Dan Ionescu, "Measurement and Control of Packet Loss Probability for MPLS VPN Services", IEEE Transactions on Instrumentation and Measurement, vol. 55, no. 5, Oct. 2006.

[9] Eleni Mykoniati, *et.al.*, "Admission Control for Providing QoS in Diffserv IP Networks: The TEQUILA Approach", IEEE Communications Magazine, January 2003.

[10] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation", in Proc. ACM SIGCOMM, pp. 301-312, 2003.

[11] S. Raghunath, K. Chandrayana, and S. Kalyanaraman, "Edge-based QoS provisioning for point-to-set assured services", in Proc. ICC 2002, vol. 2, pp. 1128C1134.

[12] S. Raghunath and S. Kalyanaraman, "Statistical point-to-set edge-based quality of service provisioning," in Proc. QoFIS 2003, vol. 2, Springer Verlag, LNCS 2811, pp. 132C141.

[13] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin", IEEE/ACM Transactions on Networking, vol. 4, no. 3, pp. 375-385, Feb. 1996.

[14] W.E. Leland, M.S. Taqqu, W. willinger, and D.V. Wilson, "On the self-similar nature of Ethernet traffic", IEEE/ACM Transactions on Networking, vol. 2, no. 1, pp. 1-15, Feb. 1994.

[15] Kihong Park and Walter Willinger, "Self-Similar Network Traffic and Performance Evaluation", John Wiley & Sons. Inc, 2002.

[16] Satish Raghunath, K. K. Ramakrishnan, and Shivkumar Kalyanaraman, "Trade-offs in Resource Management for Virtual Private Networks", ACM journal.

[17] Satish Raghunath and K. K. Ramakrishnan, "Resource Management for Virtual Private Networks", INFOCOM 2005.

[18] www.ibm.com/software/tivoli/products/netcool.

[19] www.cisco.com/web/go/netflow.