A PATH TUNING ALGORITHM FOR SHORT-TERM TRAFFIC ENGINEERING OF QOS DIFFERENTIATED MPLS NETWORKS

Mohammad Ashour, Bassel Kassem, Tho Le-Ngoc, Tallal El-Shabrawy
Department of ECE, McGill University, 3480 University, Montréal, Québec, Canada H3A 2A7

Abstract-*The recent growth in Internet traffic and services has made the provisioning of End-to-End Quality of Service (QoS) in large-scale networks a necessity. Traffic Engineering (TE) emerges as an important constituent of QoS provisioning. This paper presents a distributed Path Tuning Algorithm (PTA) for short-term traffic engineering in a delay differentiated MPLS network. In this network, each node carries a number of MPLS paths passing through a set of delay classes. These paths are established and assigned to certain delay classes at each node using a centralized long-term traffic engineering mechanism. The main objective of the proposed PTA is to fine-tune the long-term initial configuration and respond to short-term variations in order to minimize reserved resources. The PTA is constrained to maintain the end-to-end QoS requirements for all provisioned unicast paths*

1. INTRODUCTION

Traffic on Internet networks has changed dramatically from text-based web pages to video and voice-over-IP applications, which require stringent end-to-end quality of service (QoS) guarantees. As a result, the single-class best effort architecture currently used in the Internet is no longer adequate in delivering predictable and bounded QoS performance. To provide end-to-end QoS assurances, the IETF has proposed a number of architectures to overcome the limitations associated with the single class best-effort service. Integrated Services (IntServ) and Differentiated Services (DiffServ) were proposed for the QoS provision of future Internet, but have been recently challenged by the emerging Multi-Protocol Label Switching (MPLS). MPLS is an approach to achieve connection-oriented forwarding characteristics. At the edge of an MPLS network, traffic is labelled and forwarded through pre-established Label Switched Paths (LSP) [2], allowing traffic to be forwarded as an aggregate. MPLS can, therefore, establish explicit routes between source-destination pairs. This allows the implementation of traffic engineering techniques [10] as a major enabler of QoS provisioning in future Internet.

Internet traffic engineering is an aspect of network engineering concerned with the issue of network performance optimization. It includes the measurement, modelling, characterization and control of Internet traffic [3]. Traffic engineering aims at efficiently utilizing network resources while providing reliable and fast movement of traffic through the network. Traffic engineering can achieve a number of objectives depending on the timescale through which it is deployed. The Internet Engineering Task Force (IETF) has classified traffic engineering based on time-dependency into three time scales [4]. The first time scale is on the order of months, and used to make traffic forecasts as a basis for long-term network configuration, where the design network topology, the choice of different traffic routing configurations and capacity augmentation is performed. The second is on the order of days or hours, and used to improve established connections and manage network capacity to maintain an optimal network configuration. The finest scale is on the order of minutes or less, and is used to react to local short-term traffic congestions by temporarily re-optimizing flows locally until long-time scale re-optimization is performed. The ability of traffic engineering to respond to short-term traffic congestions is the main focus of this paper.

In this paper, we present a Path Tuning Algorithm (PTA) to be used on the finest scale to react to QoS degradations, resulting from short-term traffic congestions. The proposed PTA works in distributed manner to provide short-term traffic engineering in an MPLS network. We assume that MPLS nodes support a limited number of QoS classes at each output link. In the assumed network, LSP are aggregated at each output link according to the QoS they receive. The resulting LSP end-to-end delay is the sum of the delays of its assigned QoS classes at each link it goes through. The initial mapping of LSP to certain QoS class at each output is calculated as part of the off-line long-term QoS. Each node can then change this LSP/QoS mapping as part of its short-term traffic engineering. The proposed PTA controls the change of the LSP QoS mapping in already established paths in order to respond to short-term QoS violation. The PTA is constrained to maintain the end-to-end QoS requirements for all provisioned unicast paths.

2. RELATED WORK

In MPLS, end-user connections from a given origin and to a certain destination are grouped into end-to end MPLS LSPs. These LSPs are pre-established based on long-term traffic measurements. Because of the dynamic nature of a network, these end-user connections between source-destination pairs are continuously established or terminated. As a result, the aggregate LSP traffic rate can vary greatly with time. Each of theses LSP goes though a certain QoS class at each link. The combination of LSP causes an increase in burstiness in the aggregate traffic stream to a given input class. It becomes very difficult to allocate resources to meet the service levels provided by that class, and, therefore, to degrade the delay guarantees for this QoS class.

To overcome degradation in QoS levels, traffic conditioning and load balancing can be performed at the edge of the network. Load balancing has been the focus of

CCECE 2003 - CCGEI 2003, Montreal, May/mai 2003
0-7803-7781-8/03/$17.00 © 2003 IEEE

a number of studies [6,7]. Load balancing distributes network resources between routes according to the average amount of incoming traffic. Various architectures, such as the centralized bandwidth broker architecture [8], dynamically provision network resources and use admission control to minimize network congestions and increase QoS guarantees. These are based on a global knowledge of the network traffic. The knowledge can be either based on historical data or on measurement. The traffic data is usually inaccurate as historical data provide a long-term rough estimate of the traffic (which does not provide details about the short-term traffic variation) and the network-wide traffic measurement takes a long time and may have problem in information synchronization. Although load-balancing techniques can improve the overall network performance, their inability to provide a fast response to traffic bursts cause major short-term QoS degradations for established LSPs. In addition, due to excessive computational complexity, most of the existing techniques do not address the interaction among traffic service classes when performing load balancing [1].

The other approach towards guaranteeing QoS is traffic conditioning and rescheduling. Traffic conditioning shapes traffic according to specific characteristics and Service Level Agreements (SLA). Shaping controls the amount of traffic entering a network, therefore, reducing the occurrence of traffic bursts. Unfortunately, traffic conditioning is performed only at the edge of a network. Therefore, due to the continuous aggregation of connections at core routers, traffic conditioning at the edge of the network might be inaccurate, and insufficient to avoid traffic bursts and congestions. An example of traffic rescheduling approach was proposed in [5], which uses an adaptive weighted re-scheduling algorithm to overcome increase in delay. The algorithm dynamically changes the weights assigned to each service class in the Weighted Fair Queuing (WFQ) system. WFQ is usually used to provide differentiation between service classes in a network. Using the technique in [5], traffic bursts can be absorbed and QoS guarantees provisioned for high-priority QoS is continuously satisfied. Nevertheless, by continuously changing class weights, the initial QoS guarantees used to establish end-to-end connections no longer reflect the actual state of the network. Continuous change in weights might cause network instability, hence, affecting the overall network resource utilization and optimality. Furthermore, by increasing the weight of a class in a router, the performance of other classes might be greatly degraded. LSPs going through the degraded classes have no means to enhance their QoS in order to maintain the end-to-end QoS. In [5], QoS bounds are provided only for priority traffic and do not account for the effect of changing weights on the rest of service classes. As a result, the overall gain in QoS assurances might not justify the use of dynamic weight rescheduling. Furthermore, dynamic weight rescheduling requires a large amount of processing to achieve an overall improvement in QoS guarantees, making it an inefficient approach. With the increase of capacity that can be provided over network links, there were proposals for an intermediate solution of allocating extra resources to each service class at core routers. In [9], each service class is allocated bandwidth exceeding the required amount by a certain fraction. This gives service classes the ability to absorb transient traffic bursts in order to continue satisfying their provisioned QoS guarantees. The repaid increase in end-to-end traffic demands proved this solution impractical. It also results in poor utilization of network resources, and hence, increases the cost of QoS provisioning. Due to the dynamic nature of traffic, the amount of extra bandwidth reserved might be too complex to determine.

3. PROPOSED ALGORITHM

To overcome dynamic traffic congestions resulting from flow aggregations, we propose the Path Tuning Algorithm (PTA). This algorithm is implemented at every node of the network, such that each node reacts to traffic variations independent of the rest of nodes in the network.

The PTA adapts the WFQ system to provide service differentiation between service classes. To provide network stability, service-class weights are always kept constant, hence, providing fixed partitioning between classes. As a result, the capacity of the output link is divided between behavior aggregates according to their weight assignments, which sets a guaranteed lower bound for service rate on each queue.

Each service class in a core router is assigned a fixed maximum delay bound, to provide an upper threshold on the packet delay experienced at that class. This bound is used by the central mechanism to configure connections to satisfy end-to-end QoS. By setting an upper bound on packet delay and service rate, the maximum bandwidth reserved for each queue can be established. Exceeding this maximum bandwidth will cause a violation of QoS guarantees, and trigger the need for congestion response using the PTA.

The PTA algorithm performs as follows. It monitors packet delay at each class in the router, and reacts to consistent delay violation by triggering connection redistribution as soon as the delay of a number of packets has exceeded the delay bound provisioned for that class. When a request is initiated, the router picks a connection that passes through the queue (service class) where the QoS is violated, and assigns it to another service class either by upgrading it to a higher-priority class, or degrading it to a lower-priority class.

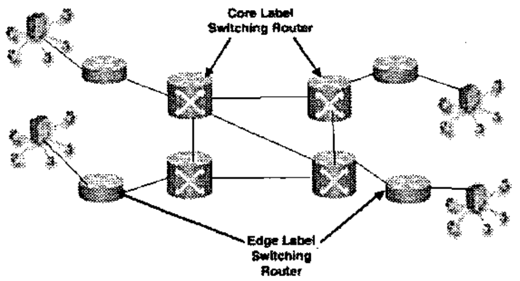

Figure 1: Simulation Network Configuration

If sufficient reserved bandwidth is available at the higher-priority queue, then the connection is upgraded to that class. If the higher class does not have sufficient bandwidth, the connection is downgraded to a lower class. The transfer of connections from the troubled class, contributes to the alleviation of congestions.

In redistributing the connections some may be downgraded to lower service class, which can affect the overall end-to-end QoS of that connection. To overcome this limitation, the PTA algorithm tries to compensate for the loss of QoS guarantees at the node experiencing congestions by upgrading this connection to higher-priority classes at neighbouring nodes along the end-to-end path. Although this process incurs higher signalling overhead, the gain achieved by alleviating congested classes while maintaining end-to-end QoS requirements, justifies this overhead. By redistributing connections along various end-to-end paths, the PTA can achieve better network utilization. Utilization is represented by a cost function, at each service class and represents the amount of reserved bandwidth relative to the maximum reservable bandwidth.

4. PERFORMANCE EVALUATION

Simulations are used to evaluate the proposed PTA mechanism in terms of four main metrics:

(i) *queueing-delay survivor function*: indicating the probability of packet delay exceeding a given value.

(ii) *network cost*: represented by the reserved bandwidth

(iii) *number of re-configuration requests*: representing the signaling overhead, and

(iv) *time to complete a request.*

Figure 1 shows the simple network topology under consideration. Core routers are connected through links with capacities of 155kbps and almost equal traffic loads. The chosen link capacity is a scaled-down version of the 155Mbps OC3 capacity to reduce the number of connections required to provision an 80% network load. The average packet size is exponentially distributed with a mean of 53Bytes. Furthermore, packet inter-arrival times were also exponentially distributed. The variation in packet sizes and inter-arrival times can create a number of traffic bursts with time, which enables us to evaluate the PTA performance. A number of end-to-end connections, with specific average bandwidth and end-to-end delay requirements, originating from various hosts, were provisioned on this network to simulate real traffic.

Weight assignments used at core and edge routers was assumed to be 3, 2 and 1 for the EF, AF and BE classes, respectively. This weight configuration provides the EF with higher priority than the AF and BE classes, AF with higher priority than the BE class.

According to the amount of traffic provisioned over the network, four simulation scenarios with network loads varying between 50% and 80% are considered. With an increase in network load, the number of QoS violations increases and the need for the PTA becomes more evident. The main objective of these simulations is to evaluate the performance of the PTA against static WFQ system.

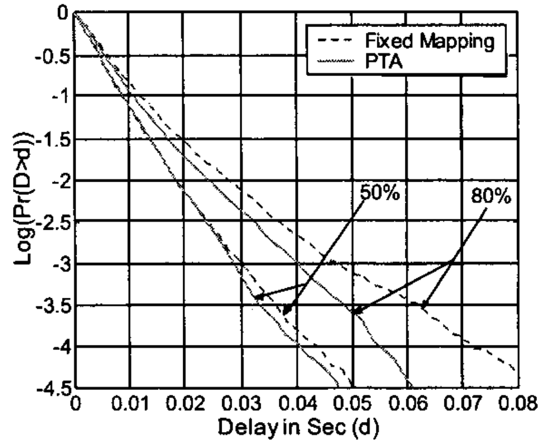

Figure 3: Queue Delay Survivor Function Pr(delay>D) for EF class at 50% network load and 80%

In Figure 2, the *queuing delay survivor function* for the EF class on one of the core routers is presented. A decrease in the graph for the delay survivor function indicates that the probability of violating a certain delay value decreases.

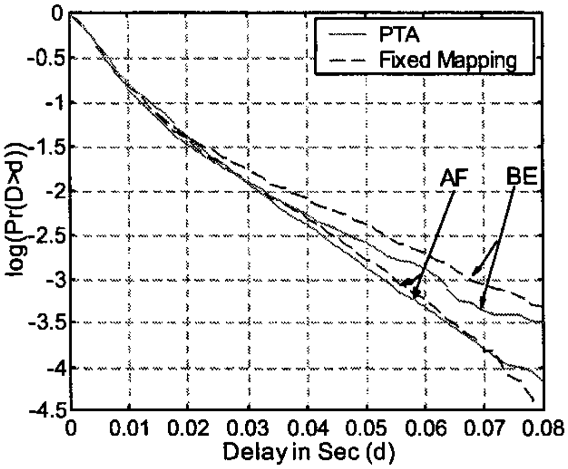

Figure 2: Queue Delay Survivor Function for BE and AF class at 80% network load

Compared to the static WFQ, the PTA mechanism significantly reduces the probability of violating QoS bounds on queues due to the fact that connections are removed from troubled classes into other classes with relatively lower loads. However, the move of connections from one class to another might affect the class accepting these connections. Figure 3, shows the delay survivor function of the AF and BE classes at the same node. It is evident that the effect of moving connections to a different queue does not cause a great degradation on its performance, compared to the degradation caused by a dynamic weight re-assignment for example. By looking at the graphs presented above, the gain achieved in QoS guarantees by utilizing the PTA algorithm is justifiable. However, it is important to show the effect of locally re-distributed connections on the overall network resource reservation. Figure 4, shows the cost for a network using the PTA as opposed to using fixed QoS mapping in four simulation scenarios. The horizontal lines represent the fixed QoS mapping since the cost does not change with time. It is evident that the PTA reduces the amount of reserved bandwidth relative to the maximum available bandwidth and improves the optimality of the network. This improvement in network cost comes as a result of the flexibility provided by locally upgrading or downgrading of individual connections at each node.

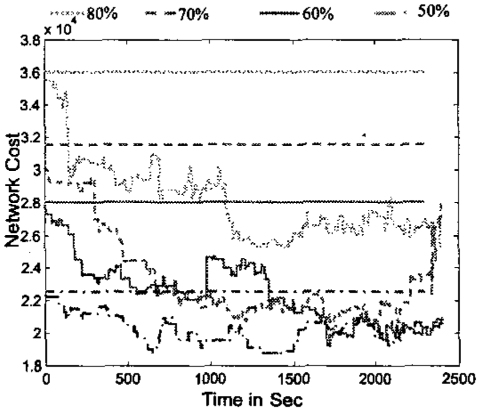

Figure 4: Network Cost at 50%-80% network load

It is important to observe that the improvement increases with the provisioned network load. As the load increases, the network cost decreases much faster. This fact indicates that, at low network loads, there is a small need for re-distributing connections since QoS requirements are satisfied most of the time. However, as the network load increases, the amount of QoS violations increases, causing a need for network re-optimization.

Table 1 shows a low number of configurations requests, indicating a small signaling overhead. Table 1 also shows the average time required to complete a request on the order of ms. This is a relatively small time with respect to the links utilized in the network (155kbps). Short congestion response times indicate an improvement in the overall performance of the network. The PTA mechanism allows core routers to respond to traffic congestions locally, independent of the state of the network.

Table 1: number of re-distribution requests

Loading	50%	60%	70%	80%
Request/sec	0.0325	0.0525	0.1275	0.316
Average response time (ms)	1.553	1.654	1.784	2.1

5. CONCLUSION

In this paper we have introduced the PTA algorithm, which responds to local traffic congestions by upgrading/downgrading connections locally at core routers. It was observed that the PTA mechanism absorbs traffic bursts and improves QoS guarantees of the network, while improving the overall network cost. Furthermore, it was shown that the signaling and processing overhead incurred by this algorithm is relatively very low and is justified by the amount of gain in QoS guarantees and network cost.

6. REFERENCES

[1] Jeonghwa Song *et al.* "Dynamic Load Distribution over Multipath in MPLS Networks," *ICOIN*, Feb. 2003

[2] D. Awduche *et al.*, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999

[3] D. O. Awduche, "MPLS and traffic engineering in IP networks", *IEEE Communications Magazine,* December 1999.

[4] W. Lai *et al*, "A Framework for Internet traffic Engineering Measurement", *IETF Internet draft*, March 2002

[5] H. Wang, C. Shen and K. G. Shin, "Adaptive-Weighted Packet Scheduling for Premium Service", *IEEE ICC'01*, Helsinki.

[6] W.T. Zaumen and J.J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations," in *IEEE INFOCOM'98*, San Francisco, 1998, vol. 3, pp. 1408–1417.

[7] C. Villamizar, "OSPF Optimized Multipath," Internet Draft <draft-ietf-ospf-omp-00.txt>, Mar. 1998.

[8] Z. Zhang, Z. Duan, L. Gao, and Y. T. Hou, "Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services" *ACM SIGCOMM'2000*,September, 2000

[9] C.C. Cheng and R. Izmailov, "The Notion of Overbooking and its Application to IP/MPLS Traffic Engineering," *Internet Draft*, Nov. 2001.

[10] XiPeng Xiao, *et al.*, "Traffic Engineering with MPLS in the Internet", *IEEE Network magazine*, pp. 28-33, March 2000.