Designing A New Routing Simulator for DiffServ MPLS Networks [*]

Peng Zhang, Zhansong Ma, Raimo Kantola
Networking Laboratory
Helsinki University of Technology
Otakaari 5A, Espoo, FIN-02015, Finland
Tel: +358 9 4515454   Fax: +358 9 4512474
Email: {pgzhang@tct.hut.fi, zhansong@tct.hut.fi, raimo.kantola@tct.hut.fi}

Keywords
Differentiated Services (DiffServ), Multi Protocol Label Switching (MPLS), Constraint Based Routing, Quality of Service, Link State Update (LSU), Multipath QoS Routing

Abstract

In this paper, we present the design and implementation of a new routing simulator called Extended QoS_based Routing Simulator (EQRS). EQRS is a discrete-event simulator developed for studying constraint based routing in the Internet. It is an upgraded version of QRS – QoS_based Routing Simulator [1]. Our objective in this version is to provide new capabilities that enable simulating DiffServ MPLS networks. EQRS allows users to configure parameters of DiffServ MPLS networks, where the dynamics of constraint based routing algorithms as well as traffic engineering mechanisms can be investigated. Particularly, with the help of EQRS, we investigate the performance and cost of QoS routing under a QoS routing algorithm and a specific link state update (LSU) algorithm. The simulation results prove that: 1) QoS routing acheives better network performance under careful configuration; 2) the simulator is suitable for modeling, designing and evaluating DiffServ MPLS networks.

1 INTRODUCTION

Offering guaranteed or assured services in the Internet is becoming one of the hottest topics in today's Internet. A large amount of previous work have been carried on in this field including various aspects such as traffic engineering, signaling protocols, network management and so on[2-3]. Among them, constraint based routing, as a general term of QoS based routing, is proposed as one of the enhancing mechanisms for deploying quality classes into the IP networks[4]. It selects routes according to not just a single metric (e.g., hop counts) but also additional routing metrics (e.g., bandwidth, delay) and administrative policies (e.g., access authentication). In general, the goal of constraint based routing is to improve the efficient utilization of network resources and to provide flexibility in support for various services.

Meanwhile, constraint based routing is expected to be used in DiffServ (DS) MPLS networks. DiffServ is proposed as the QoS architecture in the Internet [5]. MPLS is a key switching technology for implementing DiffServ [6]. In MPLS networks, data flows are forwarded along label switched paths (LSPs) on which appropriate traffic engineering mechanisms might be applied. Constraint based routing may help finding feasible routes for label switched paths in DS MPLS networks, and it may support explicit routes and multi paths for traffic engineering over MPLS [7].

Unfortunately, study on constraint based routing in DS MPLS networks is still a missing part. Many issues related to constraint based routing still remain open. For example, the dynamics of constraint based routing in DS MPLS networks, that is, how constraint based routing algorithms impacts traffic behavior and network performance is unclear yet. Furthermore, we need to study the strategy of implementing constraint based routing and the potential routing algorithms in DS MPLS networks. Additionally, simulation tools for studying constraint based routing in DS MPLS networks are not available.

In this paper, we introduce a new routing simulator – EQRS to remedy the above vacancy. Based on the previous version - QRS, we develop new capabilities in EQRS for simulating DS MPLS networks, e.g., label switching, DS traffic engineering, multipath QoS routing, network management, etc. EQRS allows users to arbitrarily configure the network topologies and network parameters, log and save the selected parameters. EQRS is suitable for modeling and evaluating DS MPLS networks.

The rest of the paper is organized as follows. In section 2, we present the structure of EQRS. In section 3, we give the design and implementation of EQRS. We examine the

---

[*] This work is supported by IMELIO project carried out in Networking Laboratory, Helsinki University of Technology.

performance and cost of EQRS under a specific LSU algorithm by simulating a typical DS MPLS ISP network in section 4. The conclusions and our future works are shown in the final section.

2 STRUCTURE OF EQRS

EQRS has the same structure as QRS: a simulation engine; a user interface; and a set of components that accommodates a variety of target systems and performance measures.

The simulation engine manages the event list and the user interface. When EQRS is started, the simulation engine initializes variables of the user interface, processes command line options (including reading any files), and then goes into the basic simulator loop.

The components are for modeling the target systems and certain simulation functions. A component in EQRS (and in QRS) consists of a data structure and a set of events, each with an associated routine. A component can be instantiated by appropriately initializing its data structure. Multiple instances of a component are needed to model a target system.

By interconnecting multiple instances of various types of components, a target system of arbitrary topology, routing protocol, and workload can be obtained.

3 DESIGN AND IMPLEMENTATION OF EQRS

We have several objectives in designing EQRS. First of all, we aim to investigate the dynamics of constraint based routing in DS MPLS networks. Second, we intend to develop new constraint based routing algorithms and look for a feasible way to build up a quality guaranteed network. Third, we intend to achieve substantial knowledge in developing such a routing simulator. Especially, we have the rule to avoid complicated policies and traffic management in EQRS. Thus, we make out the general design on the basis of the above objectives and rules.

3.1 DiffServ Architecture

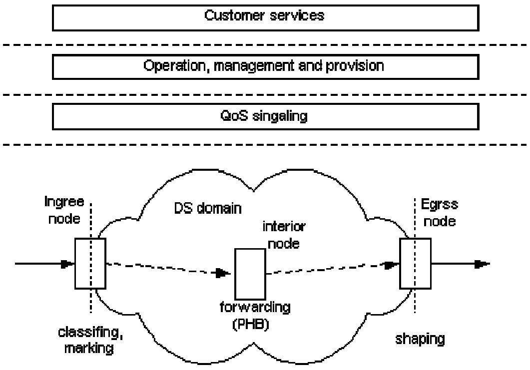

Figure 1. DiffServ QoS architecture

As shown in Figure 1, an DS network should consist of such components as customer services, network operation, management and provision, QoS signaling, intra-domain and inter-domain QoS routing, traffic scheduling and management.

Requirements from customer services should be clarified first using a number of QoS metrics. Then, network provider will provision the network resource for supporting these services. To maintain reliability and usability of the network, the network provider must carry on network operation, management and provision. QoS signaling protocol[†] is needed to broadcast control messages from network manager or to exchange interoperation information between network nodes. Intra-domain & inter-domain QoS routing selects routes for data transfer within and outside DS domains. Data packets are classified and forwarded to the destination by the network nodes.

Furthermore, the traffic scheduling can be implemented as shown in Figure 2. At the ingress node of a DS domain data packets are classified, metered and marked with a DS code point (DSCP). Then, at the interior node data packets are simply forwarded according to its DSCP, which corresponds to a specific Per-Hop Behavior (PHB). Finally, Data packets are possibly shaped at the egress node before they get into the next DS domain. Dropper and queuer[‡] are needed in all nodes.

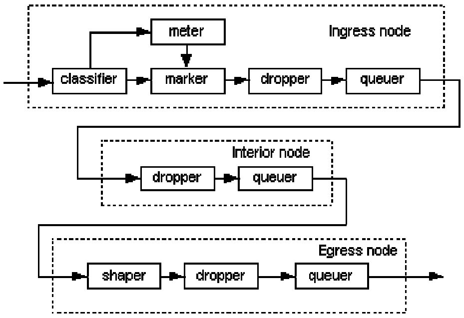

Figure 2. Traffic engineering in a DS domain

3.2 Pre-decisions

Before we begin our design of EQRS, we make some pre-decisions as follows.

First, we design and implement four classes of AF service with three dropping precedence in each kind [10] and one more DS service: EF service [11]. We choose AF and EF because they are both clearly defined by IETF. Moreover, AF services might be implemented to support most current applications, either delay-sensitive or loss-sensitive, while

---

[†] Currently, there are two candidates for signalling protocols in the Internet, i.e., Constraint based Routing – Label Distribution Protocol (CR-LDP) [8] and Extended Resource Reservation (ERSVP) [9].

[‡] We create a new term "QUEUER", which means queuing server.

EF service might be used for transferring control or signaling messages. Also, Best-Effort (BE) service is supported by default.

Second, we develop QoS signaling protocol - ERSVP on the basis of RSVP. This is mostly because our previous work on RSVP in QRS. Furthermore, RSVP can be extended to support establishment of label switch path tunnel (LSPT) between two nodes in DS MPLS networks [9].

Third, we implement virtual meter and marker in EQRS, that is, each workload is assigned to a DSCP by users through configuration files. Processing time of meter and marker is calculated in EQRS. Therefore, we avoid excess processing time and complexity incurred by meter and marker. On the other hand, users are responsible for assuring consistent Service Level Agreement (SLA) in a DS domain.

3.3 General design

Figure 3 shows the general design and interactions between components of EQRS. As shown in this figure, EQRS consists of six components, i.e., Resource Management (RM), ERSVP, Node, Link, Realtime Traffic, Routing protocol (e.g., QOSPF).

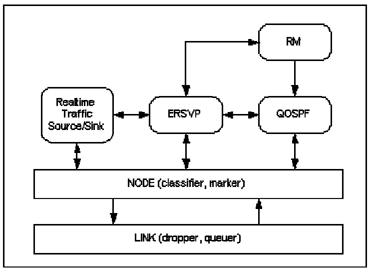

Figure 3. General design and interactions between components of EQRS

ERSVP acts as the signaling protocol for setting up LSPT. QOSPF is the routing component for finding routes within a domain. RM manages the network resources and determines to accept or reject setup requests. Realtime Traffic initializes requests for setting up LSPTs with QoS requirements. Node and Link perform traffic handling such as forwarding, queuing, and so on. In the following subsections, we describe the design and implementation of these components.

3.4 Realtime Traffic

Realtime Traffic has QoS requirements for bandwidth, delay or both. Realtime Traffic Source requests ERSVP to setup a flow connection downstream, and Realtime Traffic Sink replies to reserve resource upstream. If the flow connection is established, the source sends data packets to the sink along the flow connection. The major functions of Realtime Traffic Source/Sink are sending/receiving control messages through ERSVP, producing and consuming data packets.

We prefer to use Realtime Traffic as traffic aggregated from multiple hosts rather than a single host. Moreover, we develop Fractal Brown (fBm) distribution for Realtime Traffic, which is widely used for modeling Internet traffic.

3.5 ERSVP

We upgrade RSVP into ERSVP based on the latest Internet Draft – “Extensions to RSVP for LSP tunnels” [9].

The extensions in ERSVP mainly include two new objects, i.e., LABEL_REQUEST and LABEL, and corresponding procedures for label request and label distribution.

3.6 QoS routing

In EQRS, we enhance QOSPF component by adding two more LSU algorithms, i.e., equal class based (ECB), unequal class based (UCB). The idea of ECB and UCB algorithms is that the scope of a node's update extends to all its incident links, that available bandwidth values for all the interfaces of the node are advertised even when the update is triggered by just one link. It is also in compliance with the behavior of routing protocols such as OSPF that only generate LSUs on all the links attached to a node. In addition, ECB and UCB attempt to trigger an update only when the current available bandwidth of a link differs significantly from the previously advertised value.

- **ECB**: In this algorithm, a constant B is set, which is used to partition the available bandwidth operating region of a link into multiple equal size classes: $(0,B),(B,2B),(2B,3B),\ldots,etc$. An update is triggered when the available bandwidth on an interface changes so that it belongs to a class that is different from the one to which it belonged at the time of the previous update. It has the same degree of accuracy for all ranges of available bandwidth.
- **UCB**: In this algorithm, two constants B and f $(f>1)$ are set, which are used to define unequal size classes: $(0,B)$, $(B,(f+1)B)$, $((f+1)B, (f^2+f+1)B)$, $((f^2+f+1)B, (f^3+f^2+f+1)B),\ldots$, etc. Unlike the equal class based algorithm, the class sizes grow geometrically by the factor f. An update is triggered as before, i.e., when a class boundary is crossed. This policy has fewer and larger classes in the high available bandwidth operating region and more and smaller classes when available bandwidth is low. Consequently, it tends to provide a more detailed and accurate state description for the low bandwidth region.

We also add a number of multipath routing algorithms [13]. Mutlipath routes can be used for balancing the network traffic while keeping the efficient use of the network resource. In EQRS, these algorithms rely on QOSPF to distribute the link state information and compute the routes.

3.7 Resource Management

We have designed and implemented a very simple Resource Management. Till now, we only consider bandwidth request in RM. Its functions can be described as follows.

- When a reserve request from ERSVP is received, RM checks if there is enough bandwidth in the link, and then replies the result and also informs the routing component that the remaining resource has been changed;

- When a tear down message from ERSVP is received, RM updates the remaining bandwidth and informs the routing component that the state of remaining resource has been changed.

Moreover, users can also decide whether a flow of AF service is routed and forwarded with/without resource reservation.

3.8 Node and Link

We expand the functions of Node and Link to accommodate DS and MPLS. We design and implement dropper and queuer for all nodes in a DS domain as shown in Figure 6. The dropper employs the mechanism of WRED with three thresholds for three dropping preferences of AF services. The queuer is derived from the CBQ with priority same as that in QRS. There are six queues instead of four in QRS: one for EF service, four for AF services and one for BE service. There is no dropper with the queues for EF and BE. In a DS domain, a data packet is forwarded according to its DSCP.

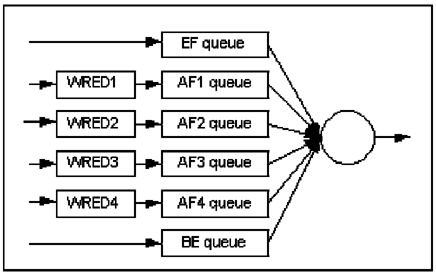

Figure 6. Dropper and Querer in EQRS

Users can arbitrarily configure the parameters of queuer and dropper, e. g., serving rate, buffer size, etc.

4 SIMULATION

In this section, we study the performance and cost of QoS routing with the help of EQRS. We simulate QoS routing in a typical DS MPLS ISP network environments (as shown in Figure 7) in order to:

- examine the basic features of QoS routing, i.e., finding paths that meet QoS requirements; accommodating best-effort traffic; improving the network throughput; indicating disruptions to the current path of a flow and building a new path if one exists;
- investigate how to improve the performance of QoS routing while keeping the cost of QoS routing on an acceptable level;

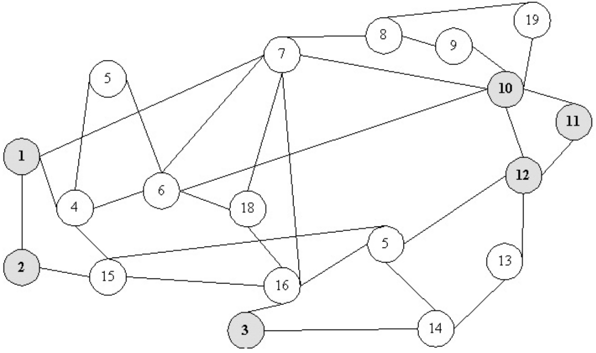

Figure 7. A typical DS MPLS ISP network

4.1 Configuration

Performance

QoS routing performance is usually described in terms of the utilization of network resources or the network throughput achieved by network traffic. It is affected by such parameters as routing algorithms, LSU algorithms, network topology and size, and so on. Here, we use the average network throughput achieved by real-time traffic with bandwidth requirements to represent the network performance. Theoretically, the larger the average network throughput is, the better the network performance should be. To get the average network throughput, we log the number of received packets in real-time traffic sinks during the simulation, then calculate the average throughput: $\Sigma_i(N_i*L_i)/t$, where N is the number of packets which are received by real-time traffic sinks, L is the packets' size, t is the simulation time and i represents the type of a packet.

Cost

QoS routing cost mainly includes path computation cost, LSU cost and storage cost. We use total processing time consumed by QOSPF during the simulation time to represent the cost of QoS routing. The longer the processing time is, the higher the cost will be. To get the cost of QoS routing, we log the time consumed by QOSPF in every node during the simulation time, and then simply calculate the sum.

Traffic Model

In the following simulations, a number of Realtime Traffic (RT) pairs and background traffic are installed into the simulated network. We use Realtime Traffic Source/Sink to model real-time traffic in terms of requests for setting up LSPs with specific bandwidth requirements as the QoS traffic. A request is characterized by its source, destination, requested bandwidth, active period (ON) and inactive period

(OFF), etc. We use Simple Traffic (ST), FTP and Telnet to model traffic without resource reservation requirements as the background traffic opposed to RT. Background traffic is installed to fill all incident links of the concerned nodes when there is no RT. Every RT pair has a unique MPLS label, and a DSCP of AFs. The background traffic is set to BE type. The control messages (e.g., RSVP messages and routing messages) are set to EF type.

In ISP network, we configure: 7 RT pairs from node 1 to each node of 10, 11, 12; 7 RT pairs from node 2 to each node of 10, 11, 12; 7 RT pairs from node 3 to each node of 10, 11, 12. Totally there are 63 RT pairs.

All RT pairs' ONs and OFFs are set randomly from 1s to 3s and 0.1 to 0.3 respectively. If a request of connection setup fails, it will re-request after 100ms. We construct two different traffic models, one is called uniform traffic (UT) model in which all RT pairs have the same flow rate, the other one is called non-uniform traffic (NT) model in which not all RT pairs have the same flow rate. For the UT, the workload of every real-time traffic flow is set to be 0.5Mbps. The average of total workload is about 27Mb/s. For the NT, the workload of each real-time traffic flow is distributed randomly from 0.1Mbps to 3Mpbs. The average of total workload is about 28Mb/s. All RT pairs start requiring LSP setup randomly. From the ISP network, we can see that the minimal cut (7-8, 7-10, 6-10, 5-12 and 13-12) has five links with total capacity 30Mb/s(5×6Mb/s). Obviously, the total network throughput achieved by RT should be at most 30Mb/s.

Each simulation runs for 100 seconds.

Routing Algorithm

Either lowest cost algorithm (LCA) or widest bandwidth algorithm (WBA) can be configured in the simulator [1]. The link cost can be defined in different ways. Here, we only use lowest cost algorithm with the hop-normalized delay cost function.

LSU algorithm

There are four LSU algorithms available in EQRS, i.e., Period Based (PB), Threshold Based (TB) [1], ECB and UCB. Here, we investigate how UCB influence the performance and cost of QoS routing.

4.2 Simulation results under different UCB values

In this subsection, we illustrate the simulation results of performance and cost under LCA and UCB LSU algorithm with different B/f values.

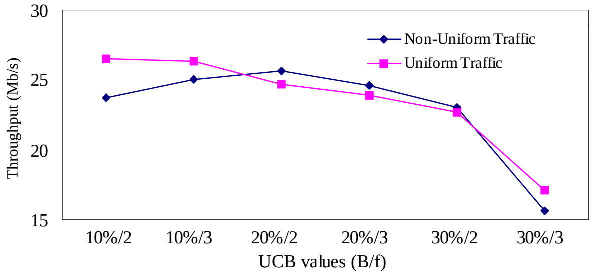

Figure 8. Performance Under LCA and NT /UT with Different UCB Values

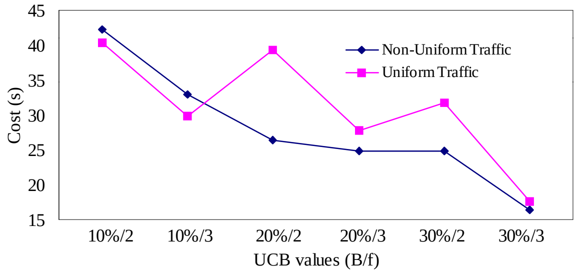

Figure 9. Cost under LCA and NT /UT with Different UCB Values

Figure 8 shows that the network achieves very good performance when B is lower than 30% and f is lower than 3. When B is 30% and f is 3, the performance degrades rapidly. Figure 9 shows that the cost generally decreases with the increment of B and f. The irregular change of the cost at B/f=20%/2 is because the complexity of the UCB algorithm with two controlling parameters. Considering both figures, the best performance/cost ratio can be achieved at B/f=20%/3.

From Figure 8, we also find the best performance of QoS routing achieved with the UCB LSU algorithm is above 25Mb/s. Since the capacity of each link is 6Mb/s, so this result means that at least 5 paths are used to transport real-time traffic simultaneously during the simulations.

If we use a best effort routing scheme, for example SPF, the shortest path from node 1 to nodes 10, 11, 12 is on links 1-7 and 7-10; the shortest path from node 2 to nodes 10, 11, 12 is either on links 2-1, 1-7, 7-10 or on links 2-15, 15-5, 5-12; the shortest path from node 2 to nodes 10, 11, 12 is either on links 13-16, 16-5, 5-12 or on links 3-14, 14-13, 13-12. Totally, there are only three paths that can be used simultaneously, and the total best throughput can be only 18Mb/s, far below 25Mb/s QoS routing achieved in our simulations.

By analyzing the ISP network, we are convinced that this result can not be achieved by a best effort routing scheme. Therefore, this result also proves the benefit of QoS routing as well as the usability of EQRS.

5 CONCLUSIONS

Constraint based routing plays an important role in DiffServ MPLS networks for supporting real time traffic transmission. It has two objectives, that is, finding routes that satisfy the constraints and making efficient use of network resources. However, there are still many open issues for deploying constraint based routing into the networks. One such problem is in which way constraint based routing is feasibly introduced. One other problem is how constraint based routing efficiently computes routes without losing stability and scalability. In this paper, we present EQRS used for studying these problems. By using EQRS, we investigate the performance and cost of LCA and UCB LSU algorithms of constraint based routing in a typical DS MPLS networks. The result proves that QoS routing algorithms achieves better performance than shortest path algorithms and also EQRS is flexible and suitable for modeling, designing and evaluating DiffServ MPLS networks. Our future work will focus on using EQRS for more simulation study.

References

[1] P. Zhang, R. Kantola, Z. Ma. "*Design and Implementation of A New Routing Simulator*". 2000 SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'2K). July 16-20, 2000, Vancouver, Canada.

[2] P. Ferguson, G. Huston. "*Quality of Service - Delivering QoS on the Internet and in Corporate Networks*". John Wiley & Sons, 1998.

[3] S. Floyd, V. Jacobson. "*Link-Sharing and Resource Management Models for Packet Networks*". IEEE/ACM Transactions on Networking, Vol. 3, No. 4, August 1995.

[4] S. Chen and K. Nahrstedt. "*An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions*". IEEE Networks. November/December 1998.

[5] S. Blake, et al. *"An Architecture for Differentiated Services*". RFC2475. December 1998.

[6] F. L. Faucheur, et al. *"MPLS Support of Differentiated Services*". Draft-ietf-mpls-diff-ext-02.txt. October 1999.

[7] D. O. Awduche, et al. *"Requirement for Traffic Engineering over MPLS*". RFC 2702. September 1999.

[8] B. Jamoussi. "*Constraint-Based LSP Setup using LDP*". Draft-ietf-mpls-cr-ldp-03.txt. September 1999.

[9] D. O. Awduche, et al. *"Extensions to RSVP for LSP Tunnels*". Draft-ietf-mpls-rsvp-lsp-tunnel-04.txt. September 1999.

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

[11] V. Jacobson, et al. "*An Expedited Forwarding PHB*". RFC 2598. June 1999. September 1997.

[12] G. Apoatolopouloa, R. Guerin, S. Kamat. "*Implementation and Performance Measurements of QoS Routing Extensions to OSPF*". Proceedings of INFOCOM'99, New York, NY, March 21-25, 1999.

[13] P. Zhang. "*Multipath QoS Routing Algorithms for DiffServ MPLS Networks*". Work in Progress. Helsinki University of Technology.