High-Performance QoS Routing

Progress Report on NSF Career Grant (2000-2001)

Klara Nahrstedt

1 Overview of Past Research Accomplishments

Internet traffic and the volume of Internet multimedia applications are increasing, hence forcing us to investigate new concepts and paradigms in the high-performance Quality of Service(QoS) routing. Especially, the recent development of Internet Integrated (IntServ) and Differentiated Services (DiffServ) aims to satisfy the large heterogeneity of applications going through Internet and to satisfy various degrees of Quality of Service in WAN networks.

Our goal in this proposal is (a) to explore differentiated routing and (b) to investigate parallel algorithms within high-performance QoS routers. This year we have concentrated on (1) parallel algorithms in IP packet forwarding for DiffServ, (2) understanding of relation between guaranteed and non-guaranteed flows in end-to-end bandwidth reservation, which will be beneficial for our exploration of differentiated routing, and (3) implementation of the DiffServ model in OPNET Simulation environment, which will be needed for the validation of future algorithms. I will elaborate on the results and reason what we learned from the research and results.

1.1 Parallel IP Packet Forwarding

We have investigated a novel architecture, called the *High-Performance QoS-capable IP Router (HPQR)* architecture, shown in Figure 1[WN01a]. In our research of HPQR we concentrate on the control plane and QoS issues, as there are many excellent results in the data transmission path, and several innovative router architectures exist [CP00, KS98, McK97, KLS98, IAM00].

Our architecture belongs to the fourth generation of switch routers. It consists of line cards (LC) to receive packets, routing agents (RA) to perform routing table lookups in parallel manner, control agents (CA) to handle routing table computation and QoS control tasks, and high-speed switch fabric. Separating IP header analysis and the routing table lookups, the line cards become light weighted. Furthermore, this separation allowed us to achieve better load-balancing and therefore higher performance.

The major contribution in this work is the *IP Packet Distribution Approach*, performed by LCs and RAs. This approach satisfies two requirements: RAs are working in a load-balanced manner, and packets from the same flow are not reordered. The algorithm distributes packets, when arriving at LCs, using the Enhanced Hash-based Distribution Algorithm (EHDA). When a packet arrives, EHDA generates a hash value for each IP packet, based on its destination address and then EHDA uses indirect hashing (hash values are used as indices to a hash table) to obtain the ID number of each RA. The content of the hash table is dynamically changed according to the instantaneous workload of each RA. Note that we use different entries for TCP packets and UDP packets. In order to differentiate the distribution of TCP and UDP requests, different hash table update policies are used. Figure 2 shows couple of policy points in the hash-table update probability vs

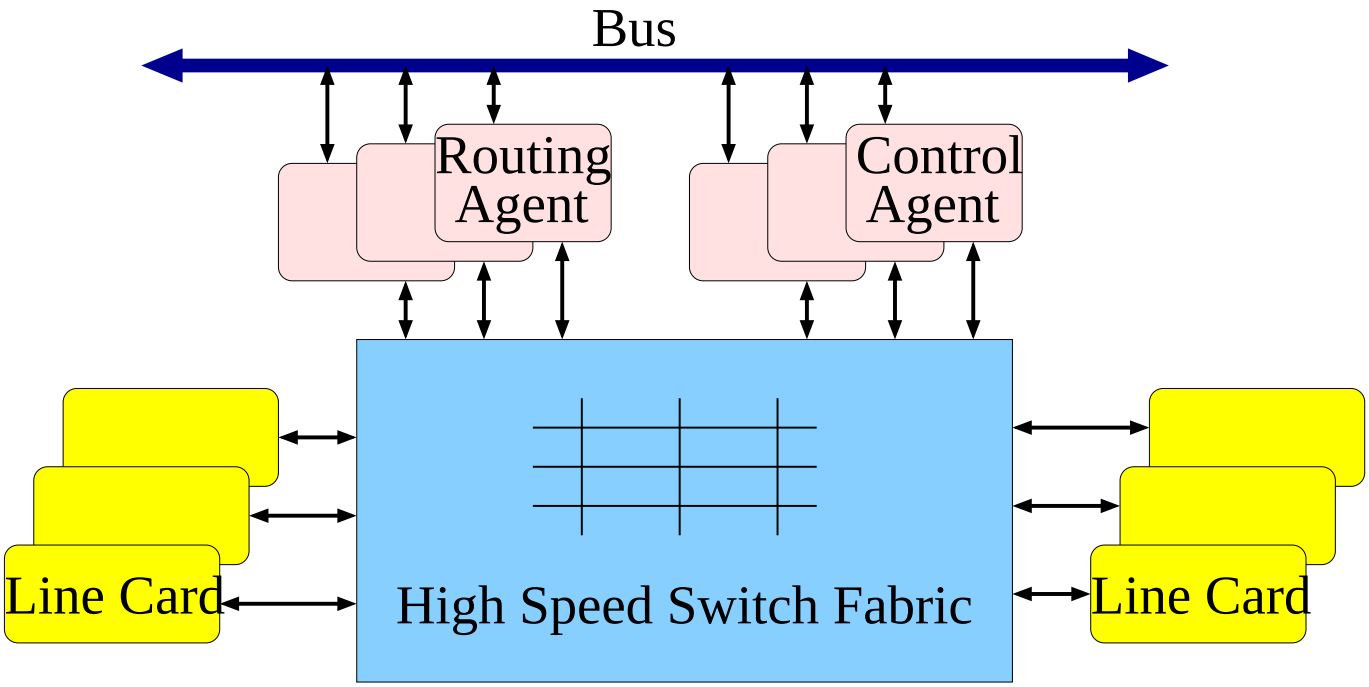

Figure 1: HPQR Architecture

workload graph. For example, if the quantized workload value is 5, then TCP RA ID entries do not change (probability is O), however, the UDP RA ID entries change with probability 0.5.

The simulation results of EHDA under balanced input and unbalanced input show that EHDA can serve all the TCP and UDP packet requests, where Basic HDA looses packets, especially if unbalanced input exists. Figure 3, 4, 5, and 6 show the results.

The initial results above, utilizing parallelism in IP packet routing, are very encouraging and we are further continuing our investigation. Several problems are still open such as possible bottlenecks in the bus architecture, relation of this routing parallelism to scheduling and switching, and others. I will point out some of the problems in the future work that we plan to address next year.

1.2 Marginable Bandwidth Reservation Problem

Part of the high-speed QoS routing infrastructure will be reservation protocols such as RSVP protocol, at least at the edges of the Internet. We looked at the following problem to understand relation between guaranteed and non-guaranteed traffic. The current reservation protocols do not present the maximum flow acceptance rate solution. The problem of finding the maximum reserved flow acceptance rate and minimizing the effects of crank-back procedure is called the *New Killer Reservation Problem (NKR)*. Hence, we asked ourselves the following two questions: (a) under what scenarios does the NKR problem become a serious problem, and (b) how can we solve the problem.

To answer the first question, our analytical solutions showed that in the case of higher arrival rate or larger reservation process delay, the request acceptance rate deteriorates for the conventional system. It is clear that in a high-loaded system, the NRK problem becomes very serious.

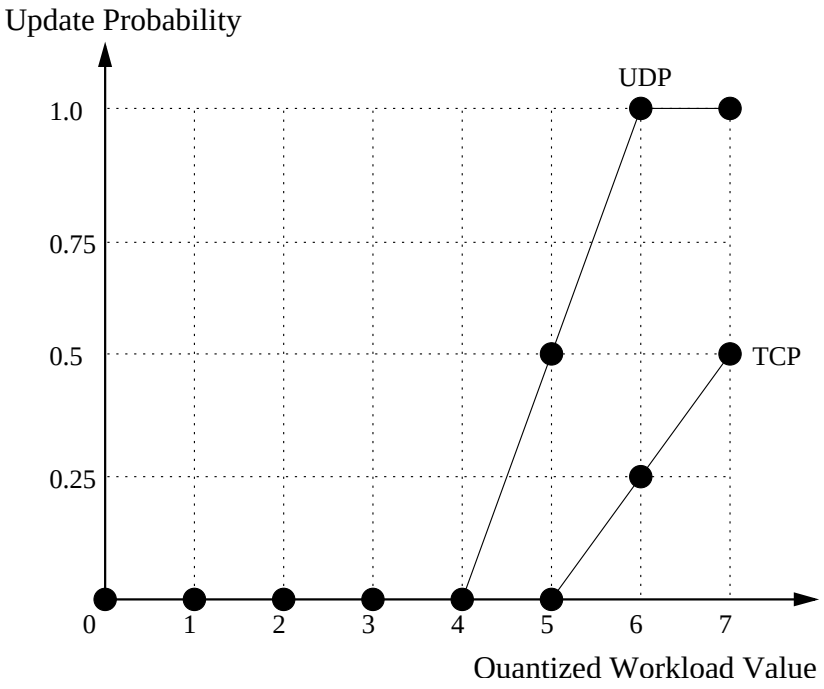

Figure 2: Update Policies for TCP and UDP Traffic

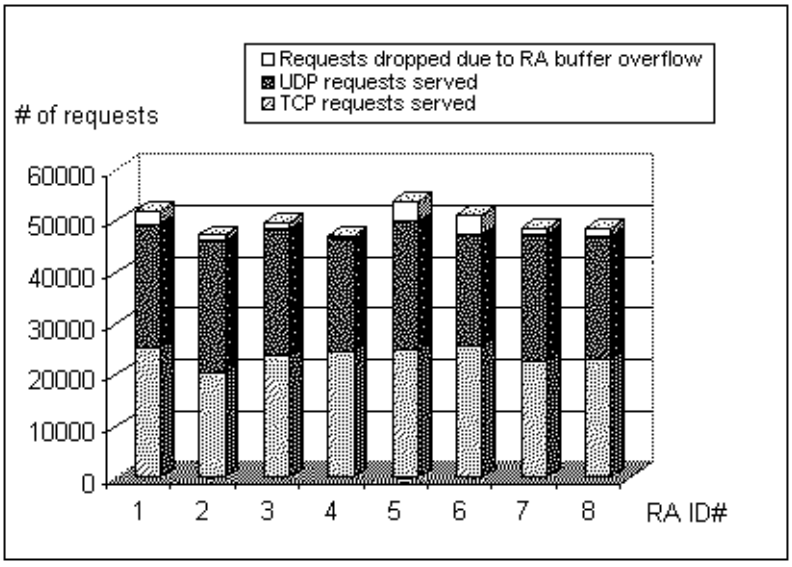

Figure 3: Distribution of Lookup Requests with Basic HDA under Balanced Input

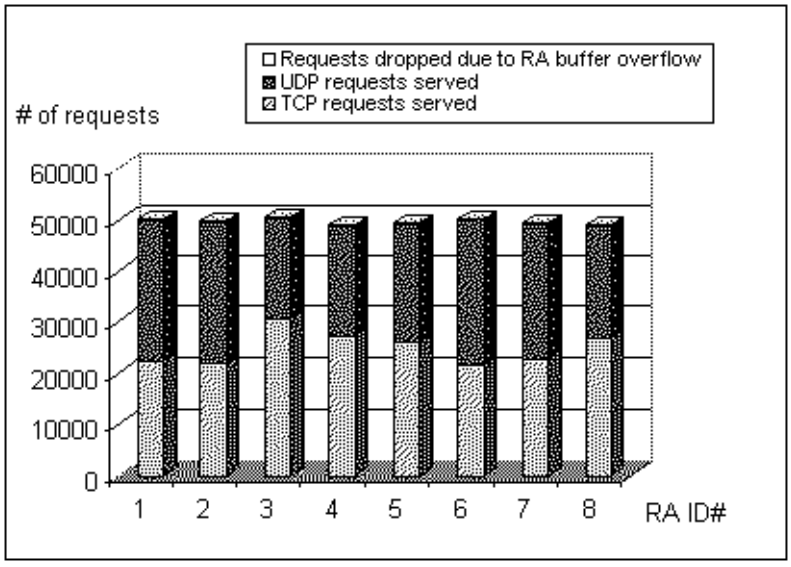

Figure 4: Distribution of Lookup Requests with Enhanced HDA under Balanced Input

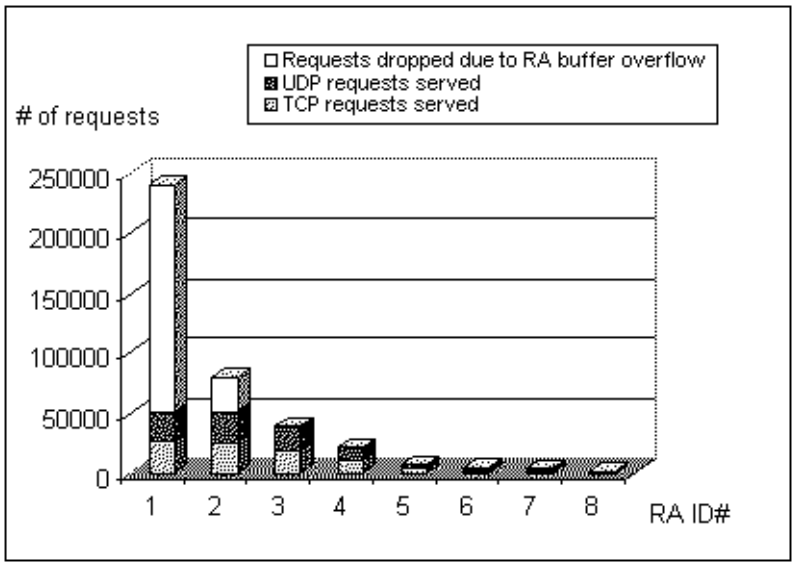

Figure 5: Distribution of Lookup Requests with Basic HDA under Unbalanced Input

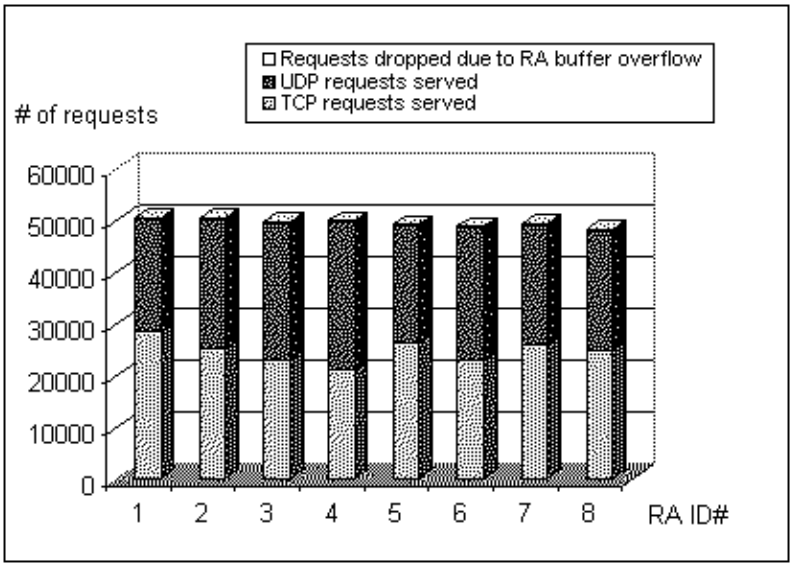

Figure 6: Distribution of Lookup Requests with Enhanced HDA under Unbalanced Input

Our solution to the second question, hence to the NRK problem, is the *Marginable Bandwidth Reservation Protocol (MBR)* [WN01b]. In the MBR protocol, we assume that one reservation request is originated from the source node and ended at the destination node. There are four phases during this protocol: Reserving, Confirming, Rejecting and Releasing of banbwidth. During the Reserving phase, the protocol reserves only a portion of the requested bandwidth (e.g., just one half of the requested bandwidth). The request is considered in the "trying state". Then the node finds the next hop with requested bandwidth. If not enough bandwidth is discovered along the path, reject message is sent back. If the request is successful, on the way back the Confirming request tries to allocate the rest of the bandwidth. If successful, then the state is changed from "trying" to "reserved" and continues to the next hop towards the source node. Otherwise, the pre-reserved bandwidth is released, and release and reject messages are sent to source and destination.

Figure 7 shows the acceptance rate improvement under different reservation durations and different bandwidth distributions.

These results show us that if we are going to use reservation schemes in the next generation Internet, we can further enhance the performance and increase the reservation acceptance rate if considering only partial reservation of bandwidth during the signaling request. Our theoretical studies and simulations confirm this statement for various reservation request rates, reservation durations and bandwidth requests which is encouraging. We will use some of these results when considering differentianted routing and bandwidth reservation for premium and assured classes in DiffServ and the relation among the individual traffic classes in a high-performance QoS router.

1.3 OPNET Simulation of DiffServ Router

Last summer we have implemented many functionalities of the DiffServ Router in the OPNET simulation package [WN00]. This work was the basis for our further validation and it was worth-while putting effort to learn OPNET and do extensive simulations of DiffServ functionalities. This implementation now serves as

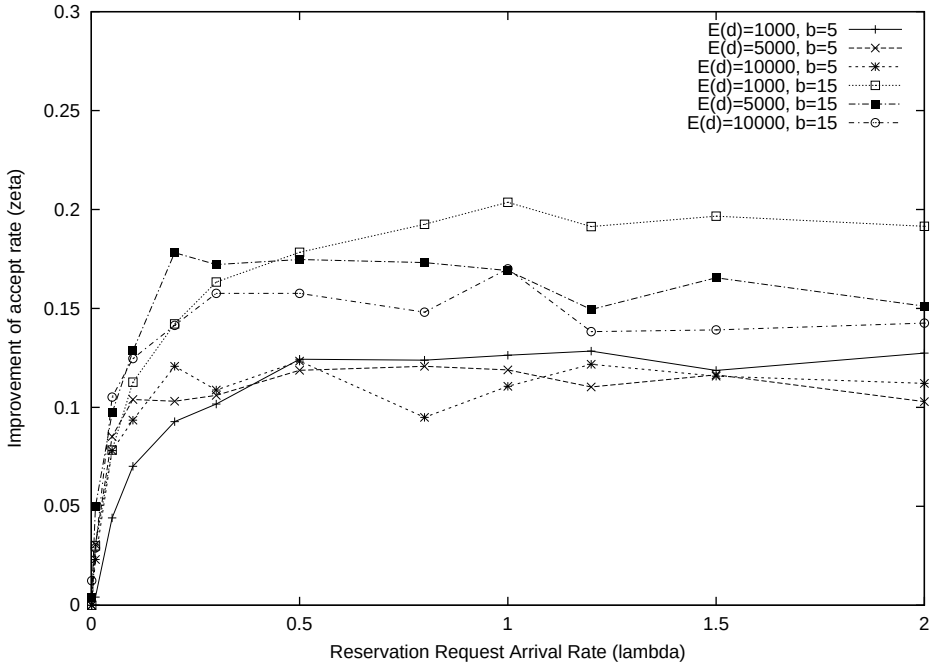

Figure 7: MBR Protocol Simulation Results

our testbed to investigate the parallelism in DiffServ routing and control. We have had many requests after publishing these simulations in OPNETWORK 2000 to provide the source code to the simulation.

1.4 Results Summary

The result have been published in the IEEE Workshop on High Performance Routing and Switching 2001, IEEE International Conference on Communications 2001 and OPNETWORK'2000. The feedback from the community was positive, and encouraging, as well as pointing out weak points in our first approaches. Designing high-performance QoS-aware routers will take couple of iterations as the complexity is large and relations are not always simple and explicit. Currently, the biggest response from the community was about the OPNET DiffServ Model. The HPQR was presented last week at the High-Performance Routing Workshop, and we are currently evaluating the feedback from this community.

Our project web page is http://cairo.cs.uiuc.edu/qosrouting/index.html

1.5 Impact on Education

Several of the results were introduced during the graduate seminar on multimedia communication, which I taught in Fall 2000 because a strong part of the seminar is QoS routing and bandwidth reservation. Furthermore, I have trained two PhD students who work with me on this project. One student started immediately last summer, when the grant was awarded, and the other PhD student started with me this Spring 2001. Both students will continue with me until completion of the project.

2 Overview of Future Work

Over the next year we plan to concentrate on two major problems to achieve our proposed goal: (a) *DiffRouting - A Multi-Class Routing Scheme for DiffServ Networks* and (b) mapping of HPQR onto cluster-based architectures and investigation of parallel algorithms in this environment.

DiffRouting: Different classes of traffic have different requirements of QoS, thus it is natural to look for different paths between a source-destination pair of nodes according to the QoS requirements. Secondly, since the premium class traffic has the highest priority over other classes, it has significant effects to other traffic. When the volume of premium traffic fluctuates, QoS measures of other class traffic deteriorates significantly in terms of bandwidth, delays and delay jitters. In order to minimize the effect of premium class traffic to other classes, we restrict premium class traffic on a spanning tree if possible. As a tree is a minimal structure that connects a network, our algorithms packs smaller bandwidth premium requests together in a way that the number of links used for premium class traffic is minimal. We then try to route assured class traffic using non-tree links that are not used for premium class in the network. If the network is still connected using the non-tree links, totally disjoint paths for premium class and assured class are found and there is no inter-class effect between these two classes. Finally, best-effort routes are found according to the residue bandwidth in the network. In order to maximize the available residue bandwidth for best-effort traffic, wide links are avoided for forwarding premium class traffics.

There are two assumptions behind our method: 1) the network is not too simple so that multiple paths exist between each pair of source- destination nodes; and 2) the best-effort traffic accounts for large portion of total traffic so that by saving the wide paths for the best-effort traffic, the overall throughput of the network will be improved, while the requirements of premium and assured traffic are still met. We will design and validate the routing mechanisms for different classes and embed them into our HPQR router design.

Cluster-based QoS Router: We have started with parallel routing agents in the HPQR architecture. However, there are still open questions such as the bottleneck via bus connecting the RAs and control agents to the switch, scalability, impact of parallel routing and QoS control on switching and scheduling. To answer these questions, we set ourselves goal to design a high-performance QoS-based router in a scalable manner with low cost and reliably. Therefore, we will investigate possible mapping of the HPQR onto cluster architectures. This design will require a careful analysis of possible topology structures for clusters which would be applicable for our HPQR as well as the study of parallel partitioning of the functionality and router data onto the cluster architecture. This research will be accompanied with the analysis of protocols which are needed for cluster-router setup, routing itself, scheduling during transmission and adaptation in case of low or high demand onto the router's bandwidth. Further goal will be to understand the reliability of this design and possibilities for fault tolerance.

Education: I plan to include the results of the HPQR and NKR into my graduate seminar in Fall 2001 and expose students to new results in end-to-end QoS routing. Furthermore, I will continue to train new generation of PhD students and master students who will work on this project.

References

[CP00] T. Chiueh and P. Pradhan. Suez: A cluster-based scalable real-time packet router. In *20th International Conference on Distributed Computing Systems*, pages 136–144, Taipei, Taiwan, April 2000.

[IAM00] S. Iyer, A. Awadallah, and N. McKeown. Analysis of a packet switch with meories running slower than the line rate. In *IEEE Infocom*, Tel-Aviv, Israel, March 2000.

[KLS98] V.P. Kumar, T.V. Lakshman, and D. Stiliadis. Beyond best effort: Router architectures for the differentiated services of tomorrow's internet. In *IEEE Communication Magazine*, pages 152–164, May 1998.

[KS98] S. Keshav and R. Sharma. Issues and trends in router design. *IEEE Communication Magazine*, pages 144–151, May 1998.

[McK97] Nick McKeown. A fast switched backlane for a gigabit switched router. In *Business Communications Review*, March/April 1997.

[WN00] Jun Wang and Klara Nahrstedt. Design and implementation of diffserv routers in opnet. In *OPNETWORK, electronic proceedings*, Washington, DC, August 200.

[WN01a] Jun Wang and Klara Nahrstedt. Parallel ip packet forwarding for tomorrow's ip routers. In *IEEE Workshop on High Performance Switching and Routing*, pages 353–357, Dallas, TX, May 2001.

[WN01b] Jun Wang and Klara Nahrstedt. A solution to the nkr problem in end-to-end bandwidth reservation. In *IEEE International Conference on Communication*, Helsinki, Finnland, June 2001.