Resource Reservation Mechanisms for Distributed Multi–path Quality of Service Routing

Xin Yuan Hui Ding Yuan Zhong Jie Zhang
Department of Computer Science, Florida State University, Tallahassee, FL 32306
{xyuan,ding,zhong,zhang}@cs.fsu.edu

Abstract

This paper studies resource reservation mechanisms that can be incorporated into distributed multi–path Quality–of–Service (QoS) routing algorithms. Three resource reservation schemes, the forward reservation scheme, the backward reservation scheme and the hybrid reservation scheme, are investigated. Simulations are carried out to compare the performance of the protocols and to study the impact of system parameters on the performance of the protocols. Our results show that, in most of the cases, backward reservation protocols are more efficient than their forward reservation and hybrid reservation counterparts.

1 Introduction

Quality of Service (QoS) routing and resource reservation [12] are two closely related network components. Traditionally, these two tasks are separated into two steps. First, a route is selected, then the route is set–up and the resources are reserved along the route. Separating routing and resource reservation simplifies the protocol design. However, in B–ISDN, resource availability may change rapidly and the route information may be outdated. In such environment, a route that was computed in the first step may lack resources in the second step. Combining the two steps was suggested to overcome this problem [3, 6, 7, 8, 13].

When combining resource reservation with the QoS routing schemes, resource reservation affects the performance of the multi–path QoS routing algorithms[1] that use global network state information more than it affects the performance of the single–path QoS routing algorithms [4, 5, 9] or that of the flooding based algorithms [3, 7, 8]. Combining resource reservation into a multi–path routing scheme may require reserving resources along multiple paths for a connection request, which result in the *over reservation problem*. Furthermore, reserving resources on multiple paths can greatly change the resource availability characteristics in the network system and decrease the precision of the global network state information, which in turn, can affect the effectiveness of the routing algorithm. Thus, incorporating resource reservation into multi–path QoS routing algorithms not only requires the design of new efficient protocols that combine resource reservation and QoS routing, but also requires the re–study of the performance issues for the routing algorithms.

Our previous work [13] developed a protocol that combines *forward* reservation with a ticket based distributed multi–path QoS routing scheme [1]. In forward reservation, resources are reserved from the source to the destination when the routing algorithm searches the QoS paths. The main limitation of this scheme is that the protocol tends to reserve more resources than needed due to the imprecise global network state information. In this paper, we study the *backward* reservation scheme and the *hybrid* reservation scheme. Backward reservation is similar to combining RSVP[12] with multi–path routing. In the backward reservation scheme, the sender sends probe packets in search of the paths that satisfy the QoS constraints without resource reservation. When the probe packet reachs the destination, the destination starts resource reservation towards the source. Hence, the imprecise global network information is only used to make routing decisions for the probe packets, more up–to–date network state information along the paths that are probed on demand is used to make reservation decisions. The hybrid reservation scheme is the combination of the forward and backward scheme. It allows a limited number of forward reservations while probing other paths. If the forward reservations failed, the hybrid scheme resorts to backward reservation. We compare the effectiveness of the three schemes and study the impact of system parameters on the resource reservation schemes. Our results show that, in most of the cases, backward reservation protocols are more efficient than their forward reservation and hybrid reservation counterparts, although backward reservation is slightly more complicated than forward reservation.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 introduces the ticket based QoS routing scheme with forward reservation.

Section 4 presents the backward reservation protocol. Section 5 describes hybrid reservation. Section 6 reports our performance study and Section 7 concludes the paper.

2 Related work

QoS routing has attracted much attention recently. An extensive survey can be found in [2]. Most existing QoS routing schemes decouple the routing issues from the resource reservation issues [1, 4, 5, 9]. Few schemes [3, 6, 7, 8, 13] combine routing with resource reservation. The routing schemes in [3, 7, 8] are flooding based where the global network information does not affect the routing performance. The scheme in [6] is a single–path routing scheme where the over reservation problem is not severe. Zhong [13] describes a protocol that combines the forward reservation scheme with a multi–path QoS routing algorithm. All the protocols [3, 6, 7, 8, 13] employ the forward reservation scheme to reserve resources. The idea of the backward reservation scheme was proposed for path establishment in multiplexed all optical networks [10]. Routing and resource reservation in all optical networks is different from routing and resource reservation for establishing a path that satisfies QoS constraints. The backward reservation scheme and the hybrid reservation scheme for QoS routing have not been studied before. In [1], the authors proposed a ticket–based routing algorithm to solve the delay–constrained least–cost routing without considering resource reservation. We use a variant of this algorithm as the base QoS routing algorithm and study resource reservation mechanisms.

3 Ticket–based distributed QoS routing with forward reservation

In this section, we will first briefly describe the ticket–based distributed multi–path routing algorithm [1]. We will then introduce the protocol [13] that combines the ticket–based routing algorithm with forward reservation. In this paper, we assume that a link state algorithm is used to maintain the global state information. We also assume that bandwidth is the QoS metric. Note that the protocols can easily be modified to deal with other QoS metrics.

The ticket–based distributed routing algorithm works as follows. When a connection request arrives at the source node, a certain number (t) of tickets are generated and a probe packet with the t tickets is sent to the destination in search of paths that satisfy the QoS constraints. Each probe packet carries one or more tickets. When an intermediate node receives a probe packet, it determines the next hops that can potentially establish connections for the request, calculates the number of tickets to be distributed for each of the next hops, and sends a probe packet with its share of the tickets to each of the next hops. In our algorithm, we use the routing and ticket distribution algorithm in [13] to determine the next hops and to distribute tickets to each of the next hops. A probe fails if there is not outgoing link that can satisfy the QoS requirement. When a probe packet reachs the destination, a path that can satisfy the QoS requirement is found. The algorithm controls the messaging overhead by manipulating t. Since intermediate nodes only distribute the tickets but not generate any new tickets, the maximum number of probe packets at any time is bounded by t. Since each probe packet probes a path, the maximum number of paths probed is also bounded by t.

Incorporating forward reservation with the ticket–based routing scheme is straight forward. When a connection request arrives at the source, the source node sends a reservation packet with t ticket to the destination. The reservation packet goes through the network, separates into multiple packets if necessary and reserves resources along the path. A reservation packet fails when there is no available resource to support the QoS requirement of the connection. In this case, the resources reserved by the reservation packet in the partial path are released if no other reservation packets for the same connection request share the resources. When the reservation packet reaches the destination, the destination sends an accept packet to inform the source and the intermediate nodes that this connection has been established. When the accept packet reaches the source, the source can start sending data. Once the source finishes sending data, it sends a release packet to release the resources along the path for the connection.

4 Backward Reservation

The forward reservation protocol relies on the global network state information maintained by the link state algorithm to make both routing and resource reservation decisions. Due to the imprecise global network state information, the forward reservation protocol tends to overly reserve resources for a connection request. The backward reservation scheme overcomes this limitation. In the backward reservation protocol, the network is probed before any reservation is made. The imprecise global network information maintained by the link state algorithm is only used in making routing decisions, more accurate network state information along the paths which are probed on demand for each connection request is used to make reservation decisions.

The detail about the backward reservation protocol can be found in [11]. Here, we will describe the protocol in an abstract manner. The protocol works as follows. When a connection request arrives at the source, a probe packet with the t tickets is sent to the destination in search of paths that satisfy the QoS constraints. The probe packets will follow the multi–path routing algorithm to search the paths without any resource reservation. Once a probe packet reaches the destination, the destination knows that resources on the path that the probe packet traveled can satisfy the QoS requirement and sends a reservation packet to the source following the path in the opposite direction. The reservation packet reserves resources along the path towards the source. Once the source node receives the reservation packet, it can start the data transmission. Note that the reservation packet may be rejected due to the lack of resources since the resource availability may change. However, this is a low probability event since the network state information carried in the probe packet is quite precise. Note also that since the protocol explores multiple paths, the over reservation problem can still exist in the backward reservation scheme. In our protocol, we use a protocol parameter, bn, to control the number of outstanding backward reservations for each request. At a given time, there are at most bn reservation packets sent from the destination for a given request.

Backward reservation improves the resource reservation efficiency and reduces the over–reservation problem comparing to forward reservation. In addition, it also improves the precision of the global network state information since resources are less likely to be in the transient state. The backward reservation protocol also has limitations. First, since resource probing and resource reservation are done at different times, there are chances that resource availability changes and resource reservation fails after the path is probed. Second, backward reservation requires more message types and incurs more messaging overheads. Third, on average, it takes the backward reservation protocol longer time to reject a request than it takes the forward reservation protocol, especially when the network is under heavy load.

5 Hybrid scheme

The idea of hybrid reservation is to perform resource reservation in the forward pass along a limited number (ft) of paths that have the highest successful reservation probability and to probe the rest ($t - ft$) paths, where t is the number of tickets for each request and ft is the number of tickets for forward reservation. If the forward reservation successfully establishes the path, the destination node can just accept the connection and ignore the results of the probing. In this case, both the problems in the backward scheme and the over reservation problem in the forward scheme are under control. If the forward reservation fails, the hybrid scheme resorts to the results of the probing and tries to reserve resources following the backward reservation protocol. In this case, the hybrid scheme behaves similar to the backward scheme. Thus, the hybrid scheme uses backward reservation as its base and tries to explore the advantages of the forward reservation scheme. Obviously, the parameter ft is important for this reservation scheme.

6 Experiments

To evaluate and compare the performance of the protocols that combine different reservation schemes with the ticket–based QoS routing scheme, we develop a cycle–by–cycle network simulator that simulates all three protocols. We use bandwidth as the QoS metric for the experiments. The simulator parameterizes the following protocol, network and traffic parameters:

• *network load*. The network load is specified using three parameters, the connection request generation rate (r), the connection duration (d) and the bandwidth requirement (b) of each connection. The connection requests arrive at each node following a Poisson distribution with an average of r messages generated every time unit. When a request is generated at a node, the destination of the request is generated randomly among the other nodes in the system. A request is discarded when it is blocked.

• *link state update frequency* (uf). This parameter determines the precision of the global network state information used by the routing algorithm.

• *number of tickets* (t) for each request.

• *forward reservation tickets* (ft).

• *the number of outstanding backward reservations* (bn).

We use two performance metrics, the blocking probability and the average number of connections granted, to evaluate the protocols. The blocking probability is the ratio of the number of connection requests blocked and the number of connection requests processed. The simulation time is measured in time cycles, which is the basic unit of the network activities. All other network activities, such as the control packet processing time and packet propagation time over a link, take multiples of the time cycle to complete. In the simulations, we assume that the propagation delay in each link 1 cycle. All the simulation results are obtained with 95% confidence level and 5% confidence interval.

We use two representative traffic patterns to compare the performance of the protocols, long connection duration with small bandwidth requirement and short connection duration with large bandwidth requirement. The long duration with small bandwidth requirement traffic pattern is generated by letting $d = 512$ and $b = 2$. This pattern is denoted as $(512, 2)$. Under this traffic pattern, the global network state information is relatively accurate while the over reservation problem is not very severe. The short connection duration with large bandwidth requirement traffic pattern represents the cases where the global network state is extremely imprecise and the over–reservation problem is very severe. We generate such traffic pattern by letting $d = 8$ and $b = 32$ and denote such pattern as $(8, 32)$.

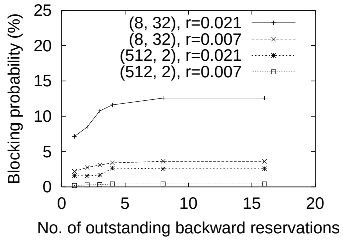

Figure 1. The impact of bn

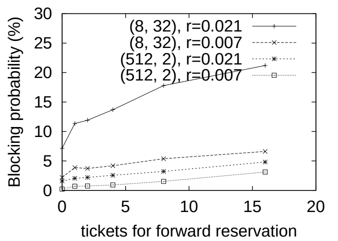

Figure 2. The impact of ft

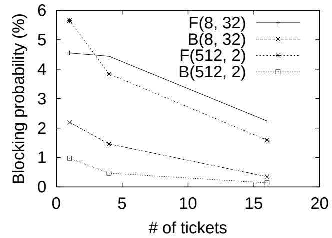

Figure 3. The performance of the protocols when the network is under low load

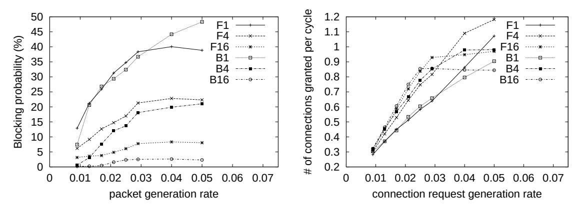

Figure 4. Performance of the forward and backward protocols for $(d = 512, b = 2)$

Figure 1 studies the impact of bn. This experiment is done on 6×6 meshes and assumes $uf = 500$ and $t = 16$. The figure shows the results for the two traffic patterns $(512, 2)$ and $(8, 32)$ with different network load (r). We have also studied other traffic patterns and other network load, the results exhibit similar trend. From the figure, we can see that for all the cases, allowing more outstanding backward reservations increases the blocking probability. There are two reasons for this. First, the probability that the first backward reservation will be successful is very high since the chance that the resource availability changes during the round trip delay is small. Thus, allowing more outstanding backward reservations creates the over-reservation problem without improving the successful reservation probability. Second, limiting $bn = 1$ reduces the amount of resources reserved but not used, which improves the precision of the global network state information. Since $bn = 1$ always results in the best performance, we will set $bn = 1$ for the rest of the evaluations.

Figure 2 shows the performance of the hybrid protocols with different ft. This experiment is done on 6×6 mesh and assumes $uf = 500$ and $t = 16$. The figure shows the results for the two traffic patterns $(512, 2)$ and $(8, 32)$ with different network load. Other traffic patterns and other network loads result in similar trend. For all cases, the backward reservation protocol achieves the best performance, which indicates that the hybrid reservation scheme cannot really combine the advantages of the forward and backward schemes. Although the hybrid reservation scheme overcomes some problems in the backward reservation scheme, it also introduces other inefficiencies. For example, due to the imprecise global network state, the likelihood that the forward reservations successfully establish the path is small. The failed forward reservations result in the over reservation problem and decrease the precision of the global network state information. As shown in Figure 2, the overall effect is that the hybrid schemes perform worse than the pure backward reservation scheme.

The study of the protocol parameters establishes that the backward schemes are more efficient than the hybrid schemes with the same number of tickets and that the backward scheme achieves the best performance when $bn = 1$. In the rest of the section, we will compare the two representative resource reservation schemes, the forward reservation scheme and the backward reservation scheme. We will use the letters B and F to represent the backward reservation protocol and the forward reservation protocol respectively.

Figure 3 compares the performance when the network is under light load. This experiment is done on 6×6 mesh and assumes $uf = 500$ and $r = 0.001$. For all the four traffic patterns, backward reservation schemes result in significantly better blocking probability than their forward reservation counterparts. Reducing the over–reservation problem and improving the precision of the global network state information greatly improve the communication performance when the network is under light load.

Figures 4 and 5 show the cases when the network load is higher. These experiments are done on 6×6 mesh and assumes $uf = 500$. For different traffic patterns, the backward and forward protocols behave differently. Figure 4 shows the case for pattern $(512, 2)$. In this case, the global

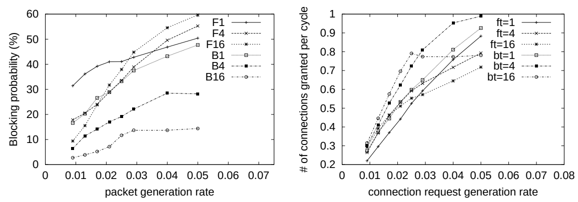

Figure 5. Performance of the forward and backward protocols for $(d = 8, b = 32)$

network state is relatively precise. The behave of the backward reservation protocol is similar to that of the forward reservation protocol. When the network is not saturated, the backward protocols offer lower blocking probability and higher throughput since they avoid the over-reservation problem and have more precise global network state information. When the network is approaching saturation, the forward protocols grant more connections than the corresponding backward protocols do, even though they suffer from the over–reservation problem. This is because the forward reservation scheme rejects requests quicker at high-load and thus has chance to establish short connections.

Figure 5 shows the case for pattern $(8, 32)$. In this case, the reduction of the over–reservation problem by the backward reservation schemes has great impact on the performance. As can be seen from the figure, the backward schemes achieve both lower blocking probability and higher system throughput comparing to their forward reservation counterparts. In this experiment, forward reservation protocols with a larger number of tickets have lower blocking probability when the request generation rate is low and higher blocking probability when the request generation rate is high. The over–reservation problem in the forward protocols with large numbers of tickets manifests itself. In contrast, backward reservation protocols with a larger number of tickets always have lower blocking probability. Since we set $bn = 1$, over–reservation is not a problem in the backward reservation schemes. Increasing number of tickets only increases the load in the control network and the time to process each request. It does not incur the over–reservation problem. We have also studied the impact of link state update frequency and the impact of network size [11]. We found that these paramenters have similar effects on all the three types of the protocols.

7 Conclusion

In this paper, we develop multi–path QoS routing protocols that combine the ticket–based routing algorithm with the backward reservation scheme and with the hybrid reservation scheme, evaluate the protocols through extensive simulation, and compare these newly developed protocols with a more established forward resource reservation scheme. Our major conclusions are the followings. First, the backward reservation protocols achieve the best performance when the maximum number of outstanding backward reservations is equal to 1. Second, the backward reservation schemes perform better than the hybrid reservation schemes with the same number of tickets. Third, while the backward reservation schemes behave similar to their forward reservation counterparts for different network settings, the backward schemes consistently offer lower blocking probability when the network is not saturated.

References

[1] S. Chen and K. Nahrstedt, "Distributed QoS routing with Imprecise State Information." *ICCCN'98*, 1998.

[2] S. Chen and K. Nahrstedt, "An Overview of Quality–of–Service Routing for the Next Generation High–Speed Networks: Problems and Solutions", *IEEE Network*, Special Issue on Transmission and Distribution of Digital Video, Nov./Dec., 1998.

[3] I. Cidon, R. Rom and Y. Shavitt, "Multi–Path Routing Combined with Resource Reservation", In *IEEE INFOCOM'97*, 1997.

[4] R. Guerin and A. Orda, "QoS–based Routing in Networks with Inaccurate Information: Theory and Algorithms." *IEEE INFOCOM'97*, April 1997.

[5] Q. Ma and P. Steenkiste, "Quality–of–Service Routing with Performance Guarantees." *4th International IFIP Workshop on Quality of Service*, May 1997.

[6] *ATM Forum, Private network network interface (PNNI) v1.0 specifications*, June 1996.

[7] H.K. Pung, J. Song and L. Jacob, "Fast and Effi cient Flooding Based QoS Routing Algorithm", *IEEE International Conference on Computer Communications and Networks (IC3N'99)*, October, 1999.

[8] K. G. Shin and C.C. Chou, "A Distributed QoS Routing Using Bounded Flooding", *University of Michigan CSE technical report*, CSE–TR–388–99.

[9] Z. Wang and J. Crowcroft, "QoS Routing for Supporting Resource Reservation." *IEEE Journal on Selected Areas in Communications*, September, 1996.

[10] X. Yuan, R. Melhem, R. Gupta "Distributed Path Reservation for Multiplexed All-Optical Networks", *IEEE Transaction on Computers*, Volume 48, No. 12, Dec. 1999.

[11] X. Yuan, H. Ding, Y. Zhong and J. Zhang, "Resource Reservation Mechanisms for Distributed Multi-path Quality of Service Routing." *Technical Report TR-000701*, Department of Computer Science, Florida State University, July 2000.

[12] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP, A New Resource ReSerVation Protocol", *IEEE Network*, September 1993.

[13] Y. Zhong and X. Yuan, "Impact of Resource Reservation on the Distributed Multi–path Quality of Service Routing Scheme", *The Eighth International Workshop on Quality of Service (IWQoS2000)*, Pittsburgh, PA, June 2000.