A Solution to the NKR Problem in End-to-end Bandwidth Reservation [1]

Jun Wang, Klara Nahrstedt [2]
Department of Computer Science
University of Illinois at Urbana-Champaign
{junwang3, klara}@cs.uiuc.edu

Abstract

With up-coming Quality of Service (QoS) requirements raised by a wide range of communication-intensive, real-time multimedia applications, resource reservation is one of the approaches to satisfy requested QoS. Resource ReSerVation Protocol (RSVP) is a well-known signaling protocol for bandwidth reservation in the Internet. However, the current reservation protocols like RSVP 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, Section II- B). In this paper, we study the NKR problem and answer two questions: (1) under what scenarios does the NKR problem become a serious problem; and (2) how can we solve this problem. We introduce the Marginable Bandwidth Reservation Protocol (MBR) as a solution to the NKR problem. The results show that MBR protocol yields the desired improvement of flow acceptance rate at a low cost.

I. Introduction

In the Internet, Integrated Service (Intserv) and Differentiated Service (Diffserv) are two well-known models which provide QoS to end users ([1], [2], [6]). In the Intserv, Resource reservation plays a key role. Many efforts have been put into the resource reservation research ([4], [5], [7]).

In connection-oriented networks, such as ATM, a reservation is made once for the lifetime of each connection. For connectionless datagram networks, such as the Internet, a reservation is soft and needs to be renewed periodically.

In both network types, a general reservation scheme consists of two steps: (1) resource reservation from source node to destination node, and (2) resource allocation from the destination node to the source node. The main overhead for this scheme comes from the crank-back procedure (in which the pre-reserved resources must be released) due to a failed try, since the reservation request rejection may occur close to the destination when most of the route is already reserved for the connection. We call this problem the New Killer Reservation (NKR) problem and discuss it below.

Multimedia and WWW traffic is now counted as an important part of the overall traffics in the Internet. These applications may reserve bandwidth in high demand within a short period of time and they cause the Internet connections to be bursty ([4]). We will show that in an environment with bursty connections, if bandwidth reservation is used, the NKR problem becomes even worse.

In this work, we present the Marginable Bandwidth Reservation (MBR) protocol as a solution to the NKR problem. Both the NKR problem and the MBR protocol are analyzed theoretically. In addition, simulations are conducted to study the performance of the MBR protocol. The results show that the MBR protocol solves the NKR problem at a low cost with respect to message overhead.

The paper is organized as follows. In section II, we introduce the reservation model and formulate the NKR problem. Section III describes the MBR protocol in detail. In section IV, we do the theoretical analysis on the NKR problem and the MBR protocol. Section V covers the simulations and their results. In the last section, we conclude our work.

II. Problem Formulation

Although reservation is an important issue with respect to different resources, we only target the bandwidth reservation issue in the Internet. We believe that the bandwidth reservation mechanism might be applied to other distributed resources where general resource reservation is required.

Throughout the paper we will use the symbols shown in Figure 1.

Symbol	Descriptions	Section(s)
λ	Reservation request arrival rate (the mean of Poisson distribution)	2, 4, 5
$1/\mu 1$	Average delay it takes the system to accept/reject a request (average RTT)	4, 5
d	Duration of a reservation (exponentially distributed with mean $1/\mu 2$)	2, 4, 5
$1/\mu 2$	Expected value of d ($E(d) = 1/\mu 2$)	4, 5
b	The maximum value of required bandwidth (bandwidth requirement is uniformly distributed within [0, b])	5
p	Probability at which a reservation request gets accepted finally	2, 4
$\rho 1$	$\rho 1 = \lambda / \mu 1$	4
$\rho 2$	$\rho 2 = \lambda / \mu 2$	4
α	The reservation request accept rate	4, 5
ζ	The improvement of accept rate α in the simulations	5
m	The link capacity used in the theoretical model of MBR protocol	4
k	The "margin" bandwidth used in the theoretical model of MBR protocol	4

Figure 1: Symbols and their descriptions

A. Reservation Model

Reservation protocols are divided into two categories: sender-oriented and receiver-oriented ([8]). For example, ST-II is a sender-oriented reservation protocol ([9]), while RSVP is receiver-oriented ([1]). In our study, we generalize these two categories into one simplified model, called the conventional reservation model. In this model, we use reservation initiator (the source node) and initiatee (the destination node) instead of the sender and receiver.

The generalized reservation model works as follows: the reservation initiator sends a REQUEST message with the QoS specifications and requirements included; at each router/switch along the path, the resource reservation manager makes admission decision based on the available resource it has; after the admission decision, the resource manager reserves the resources and updates the corresponding QoS information; if the request is accepted by all the interior nodes in the path and the

[1] This work was supported by the National Science Foundation PACI grant under contract number NSF PACI 1 1 13006, NSF Career Grant under contract number NSF CCR 96-23867 and NSF CISE Grant under contract number NSF EIA 99-72884.

[2] Please address all correspondences to Jun Wang and Klara Nahrstedt at Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, phone: (217) 244-5841, fax: (217) 244-6869.

destination node, a CONFIRM message is sent back from the destination to the initiator; when the connection finishes, a RELEASE message is sent from the initiator to the destination to tear down the reservation.

B. New Killer Reservation Problem

As we know, there are two famous Killer Reservation Problems existing in RSVP: KR-I and KR-II. [3] The reason behind both KR problems is the merging of reservations ([1], [2]). Unlike these KR problems, the New Killer Reservation problem (NKR) can be described as follows:

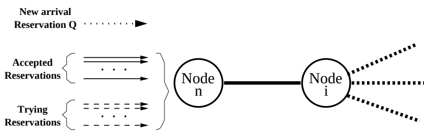


Figure 2: NKR Problem

At node n, a new coming request Q is trying to make a bandwidth reservation b_q. At this time, some portion of bandwidth (denoted as $b_{accepted}$) has been reserved by those accepted reservations, while other portion of bandwidth (denoted as b_{trying}) is occupied by those trying reservations. If the capacity (C) is used up, i.e., $C - b_{accepted} - b_{trying} < b_q$, the request Q will be rejected immediately. But we should notice a fact that some of those trying reservations are being rejected somewhere downstream, for example, at node i. The bandwidth occupied by them is wasted while the new coming request Q has been rejected immediately because of the lack of available bandwidth.

If we make use of this wasted portion of bandwidth, overall acceptance rate and reservation throughput could be improved. Intuitively, the parameters which are expected to have impact on the NKR problem are: reservation arrival rate λ, reservation duration d, probability p at which reservations get accepted downstream, network topology, workflow pattern of reservation requests, and network dynamics (e.g., hot spots in the network). In this paper, we address two questions with respect to the NKR problem:

1. How and to what extend does the NKR problem affect the overall performance such as the overall acceptance rate. In what scenarios, the NKR problem deteriorates and becomes a serious problem.
2. How can we solve this problem, i.e., how can we improve the overall performance in case of the NKR problem.

III. MBR PROTOCOL

This section describes the MBR protocol in detail. We will use this protocol both in our theoretical analysis (Section IV) and our simulations (Section V). Intuitively, the NKR problem is similar to the reservation problem in an air-ticket reservation system. Assuming some customers will cancel their flights, air-tickets are overbooked to improve the occupancy rate.

In the MBR protocol, we assume that one reservation request is originated from the source node s and ended at the destination node t. The bandwidth requirement is b_q. There are four phases in the MBR Protocol: Reserving, Confirming, Rejecting and Releasing, as shown in Figure 3.

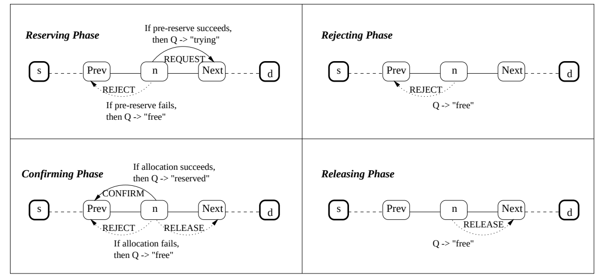


Figure 3: Four phases of the MBR protocol

Phase 1. Reserving When a request comes to node n,it first checks if the residual bandwidth can hold the request or not. If yes, it pre-reserves only a portion of the requested bandwidth (for example, in our simulations we pre-reserve one half of the requested bandwidth in this phase). The request now is considered to be in the "trying" state. Then node n finds the next hop for the request and forwards the request to the next hop. If there is not enough available bandwidth left, a REJECT message will be sent back all the way to the source node, and the reservation goes into phase 3 (rejecting phase).

Phase 2. Confirming Suppose the request arrives at t finally and gets admitted at t, a CONFIRM message will be sent back to the previous hop and all the way to the source node s along the same path. When a node n receives a CONFIRM message for a connection request, it tries to reserve and allocate the rest of the requested bandwidth (in our simulations, it is the other half of the requested bandwidth) and change the request from "trying" state to "reserved" state. If the reservation and allocation succeed, the CONFIRM message is sent back to the previous hop of the request at node n. Otherwise, the pre-reserved bandwidth is released. Meanwhile a REJECT message is sent back to n's previous hop and a RELEASE message is sent to n's next hop simultaneously. In this case, some nodes of the path will be in phase 3 (rejecting phase) and other nodes will be in phase 4 (releasing phase).

Phase 3. Rejecting When a node receives a REJECT message for a connection request, it is in the rejecting phase. It first releases the pre-reserved bandwidth, then it forwards the REJECT message backward to its previous hop.

Phase 4. Releasing When a node receives a RELEASE message or the reservation finishes at the source node, it is in the releasing phase. The node releases the fully reserved bandwidth and then forwards the RELEASE message to its next hop.

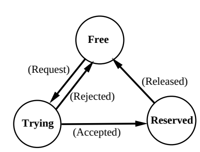


Figure 4: MBR state transit graph

The NKR problem is solved by the MBR system as follows: (1) Reservation requests in a conventional system contains two states: "free" and "reserved". Our MBR system adds a new "trying" state. When a reservation request comes to a node, it gets into the "trying" state at that node first. After a confirmation is received, it is changed from "trying" state to "reserved" state (Figure 4). (2) The bandwidth required by a "trying" request is marginable, which means an arriving request can be admitted into the "trying" state by pre-reserving only a portion of the required bandwidth. This is similar to a marginable account in the stock market (the account holder pays only a portion of the cost for stocks and the other portion is covered by margin). As a result, more reservation requests can be admitted into the system with "trying" state, which creats the effect that the capacity increases VIRTUALLY at each node. (3) The basic idea behind the "trying" state is to assume that a portion of reserved but not allocated bandwidth is free and this free portion is entered into the general resource pool of available bandwidth during

[3] "The KR problem is a denial of quality-of-service that can result from merging two different flowspecs in RSVP. If the path towards the source has sufficient resources for the smaller of the reservations but not for the merged reservation request, then the effect of merging can be to deny reservations to both." [2]

the admission process. Therefore, more requests can be admitted into "trying" state. For example, in a conventional system, suppose at most m requests can be injected into the system, while in the MBR system, k more requests can be admitted due to the usage of margin. Since some portion of those m requests may be rejected somewhere in the downstream, we hope the extra k admitted requests can take the "wasted" bandwidth so that the overall acceptance rate is improved. This is similar to the ticket overbooking in an air-ticket reservation system.

IV. Theoretical Analysis

We give theoretical analysis on both the NKR problem and the MBR protocol in this section.

A. System Model

Based on the network model we have introduced in Section II, we make several assumptions to simplify our analysis. The target we will analyze consists of (1) node n, (2) one outgoing link attached to the node (link l) and (3) all reservations going through the node and link, as shown in Figure 5.

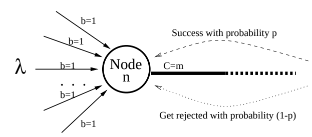


Figure 5: System Model

Here are our assumptions:

- Reservation requests arrive at Poisson distribution with arrival rate λ.
- Reservation process delay is the interval from the time node n initiates a reservation until the time node n receives an acceptance confirmation or reject message. The delay is exponentially distributed with mean $1/\mu_1$.
- Reservation duration d is exponentially distributed with mean $1/\mu_2$.
- The probability at which reservations get accepted downstream is p, i.e, each reservation will be rejected later in downstream at probability $1 - p$.
- The capacity of the link is fixed to m, i.e., $C(l) = m$.
- All reservations request the same unit bandwidth ($b_q = 1$), which means the link can hold as many as m distinct reservations simultaneously, including accepted and trying reservations.

For a conventional reservation protocol, consider a source node n connected by a link l with capacity m, it can support m connections at a time (Figure 5). The arrival process of connection requests is Poisson distributed with mean λ. When a connection request arrives at node n, it begins to setup the reservation. At first, the node tries to find the path for the request using a QoS routing algorithm (see [3]). Suppose the link l is selected to get to the next hop. Then the resource manager at node n examines the resource availability (residual bandwidth for link l) at this time. If there is enough bandwidth left, the request goes through this node and is forwarded to the next hop. Meanwhile, the residual bandwidth for link l is updated (deducted by 1 in our case). The request is forwarded all the way to its destination hop-by-hop, trying to reserve bandwidth at each hop, just like at this node. If the request arrives at the destination finally and is accepted, a confirmation is sent back to n. Otherwise, a reject message is sent back. The period from when the request is initiated at n until a confirmation or reject message is received by n is called *reservation process delay* and is exponentially distributed with mean $1/\mu_1$. If the reservation is accepted, it will last for duration d which is exponentially distributed with mean $1/\mu_2$. $1/\mu_1 \approx 1/\mu_2$ means that the connection is short and its duration is comparable to the reservation process delay. There are some advanced reservation protocols which, upon receiving a reject message, make additional attempts. We do not consider these cases individually in this paper [4].

In the MBR system, only a portion of requested bandwidth is reserved at the "reserving" phase. As a result, requests are easier to be admitted into the "reserving" phase and more requests can be injected into the system with the "trying" state. It has the same effect as increasing link capacities virtually. For simplicity, the capacity increases by k at each link in the MBR system, which can be considered as the "margin". Suppose at certain time when the system contains j connections in "reserved" state and i connections in "trying" state, a new request Q arrives. As long as $i + j < m + k$ (recall m is the real available bandwidth) and $j < m$, Q will be accepted into "trying" state and i increases by 1. If Q arrives at its destination and gets accepted, then a confirmation is sent backward along the same path. At each node along the path, the state for Q is transferred from "trying" to "reserved" if $j < m$ upon receiving the confirmation. The values of i and j are updated accordingly. Otherwise, a reject message is sent to both directions and the state of Q is transferred from "trying" to "free". i decreases by 1.

In order to study the NKR problem, we introduce an ideal system for comparison. In this system, all assumptions hold except that the reservation process delay is zero. When a reservation request arrives, it will get accepted or rejected immediately without any delay. By comparing the ideal system and the conventional system, we are able to analyze how the NKR problem affects the overall system performance.

B. Analysis of the System Model

Based on the assumptions and system descriptions above, we use the Markov chains to analyze and compare three systems: MBR, ideal and conventional. The reason why we analyze the systems by the Markov Chains is that, from the Markov Chains and their balance equations, we can derive some QoS parameters such as the acceptance rate in the steady state.

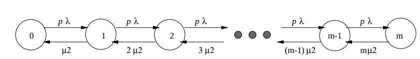


Figure 6: Markov chain for the ideal reservation system

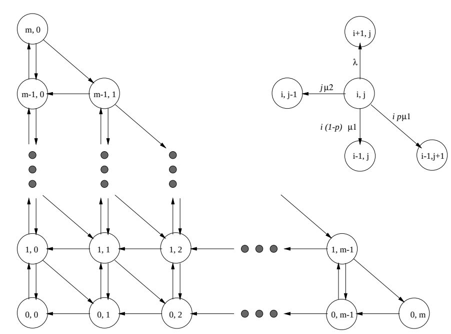


Figure 7: Markov chain for the conventional reservation system

In Figure 6, the Markov chain for the ideal reservation system is shown. In this case, state i denotes a state in which i connections have been accepted (in "reserved" state). Since in the ideal system the reservation process delay is zero, we do not need the third state - "trying". Figure 7 and Figure 8 illustrate the Markov chains for the conventional reservation system and our MBR reservation system respectively. In the figures, (i, j) denotes a state in which there are i connection requests in "trying" state and j connections in "reserved" state. The balance

[4] Another way to think about these cases is that we can consider the re-trials as new requests. The result is that the arrival rate λ increases.

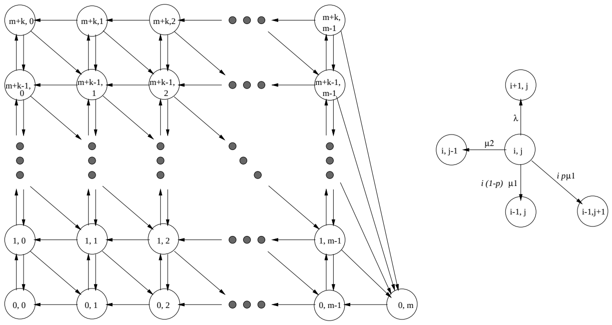


Figure 8: Markov chain for the MBR reservation system

equations for these three Markov chains are given in [10].

Based on the Markov chains for these system models, we define reservation request acceptance rates as the metrics to be evaluated and compared:

$$\begin{cases} \alpha_o = \frac{\sum_{i=0}^{m}(p\lambda P_i)}{\lambda} = p\sum_{i=0}^{m} P_i \\ \alpha_c = \frac{\sum_{j=0}^{m}\sum_{i=0}^{m-j}(ip\mu_1 P_{i,j})}{\lambda} = \frac{p}{\rho_1}\sum_{j=0}^{m}\sum_{i=0}^{m-j}(iP_{i,j}) \\ \alpha_m = \frac{\sum_{j=0}^{m}\sum_{i=0}^{m+k}(ip\mu_1 P_{i,j})}{\lambda} = \frac{p}{\rho_1}\sum_{j=0}^{m}\sum_{i=0}^{m+k}(iP_{i,j}) \end{cases} \quad (1)$$

where α_o, α_c, α_m are expected reservation request acceptance rates for the ideal, conventional, and MBR system respectively.

C. Analysis of a Special Case

In order to calculate the acceptance rates (α_o, α_c, α_m), we have to solve the stationary distribution $P_{(i,j)}$ (where $P_{(i,j)}$ means the probability of the system being in the (i,j) state steadily) for the three different systems. For the Markov chains of the conventional reservation system and the MBR system (Figure (7) (8)) it is hard to obtain closed-form solutions for $P_{(i,j)}$. But we can still get some knowledge on those systems through the analysis of a special case (the simplest case). Simulations will be used to study the general cases.

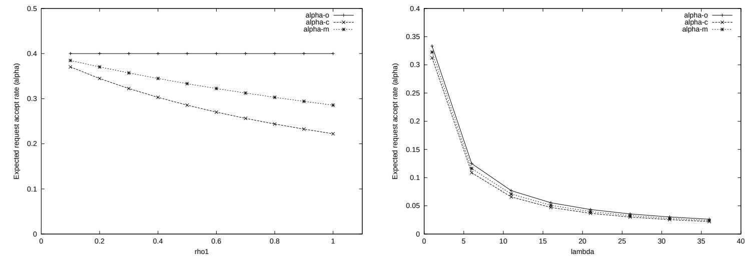
We consider the simplest case where $m = 1$, $k = 1$. Apply $m = 1$ and $k = 1$ to the general model introduced in Section B, we have

$$\begin{cases} \alpha_o = \frac{p}{1+p\rho_2} \\ \alpha_c = \frac{p}{1+\rho_1+p\rho_2} \\ \alpha_m = \frac{2p(1+\rho_1)}{2(1-p)+2p(1+\rho_1)(1+\rho_2)+\rho_1(2+\rho_1)} \end{cases} \quad (2)$$

where $\rho_1 = \lambda/\mu_1$ and $\rho_2 = \lambda/\mu_2$. Equation (2) illustrates the relationship between the request acceptance rates and different parameters, such as the reservation request arrival rate λ, reservation process delay $1/\mu_1$, connection duration $1/\mu_2$, and acceptance probability p. We give numerical results in the following graphs (more details are in [10]).

Figure 9(a) and 9(b) illustrate the relationship between the acceptance rate α and two parameters: ρ_1 and λ respectively. From Figure 9, we can conclude that:

1. In the case of higher arrival rate (larger λ), or larger reservation process delay (smaller μ_1), the request acceptance rate deteriorates for the conventional system. It is clear that, in a high-loaded system (larger λ and larger reservation process delay $1/\mu_1$), the NKR problem becomes more serious.
2. Not surprisingly, our MBR system does improve the acceptance rate, because in Figure 9, the MBR curves (alpha-m curves) are always above the conventional model's curves (alpha-c curves). But the MBR system still can not reach the ideal system, because the MBR curves (alpha-m curves) are always below the ideal model's curves (alpha-o curves).



(a) α vs ρ_1: $\rho_2 = 0.5, p = 0.5$

(b) α vs λ: $duration = 1.0, delay = 0.1, p = 0.5$

Figure 9: Acceptance rate for special case (m=1,k=1)

In order to examine exactly how much we can benefit from adopting the MBR model, we define another metric ζ, the improvement of the acceptance rate.

$$\zeta \triangleq \frac{\alpha_m - \alpha_c}{\alpha_c} \quad (3)$$

where α_m and α_c are acceptance rate for the MBR system and acceptance rate for the conventional system respectively.

By studying Equation 2 and 3 we can obtain that (1) The percentage of improvement is strongly affected by ρ_1. Recall that $\rho_1 = \lambda/\mu_1$, therefore, the percentage of improvement is strongly affected by arrival rate λ and processing delay $1/\mu_1$; (2) the acceptance rate deteriorates with a smaller acceptance probablility p; and (3) p has a big influence on the improvement of acceptance rate ζ, while ρ_2 has little.

Theoretically, therefore, we could expect a significant improvement by using MBR in case of high workload (large λ), or in case of large reservation process delay (small μ_1), or in the case that there are some "hot spots" in the network (small p).

V. Simulation and Results

In our simulations, we focus on two real performance measures: the reservation request acceptance rate, and the message overhead. The message overhead means how many extra messages we should use to achieve the improvement of acceptance rate.

A. Simulation Configuration

We conduct our simulations on the topology shown in Figure 10. In

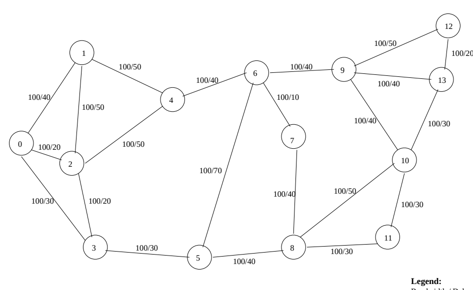


Figure 10: Simulation Topology

the topology, all links have the same bandwidth capacity (we normalize the value as 100 units), while each of them has its individual delay value. At each node, reservation request arrivals are Poisson distributed with arrival rate λ. The bandwidth requirement b_q for each request is uniformly distributed between $[1, b]$. The destination for a request is uniformly distributed among all the nodes except the request initiator (which means the destination could be any node with the same probability except the initiator). The duration of a reservation d is exponentially distributed with the parameter $E(d) = 1/\mu_2$ (recall that we have introduced d and μ_2 in the theoretical analysis). In Figure 10, we can see that there are totally 14 nodes numbered from 0 to 13. The delay of a link is ranged from $10ms$ to $70ms$, normally between $20ms$ and $40ms$. In a real system, a delay includes a queueing delay at a node, plus the service time (time for routing, making admission decision, updating the state or topology database etc.) and the propagation delay of a link. To simplify our simulations, we sum them up as a single delay value and assign it to each link. Furthermore, we consider all nodes to be identical.

B. Analysis of Simulation Results

First, we observe the overall performance between the acceptance rate α and the request arrival rate λ. Figure 11 shows that the overall acceptance rate decreases when λ increases, and this is consistent with the theoretical result (Figure 9(b)). The curve of the MBR system is always above the curve of the conventional system, which means MBR does improve the overall performance in term of acceptance rate.

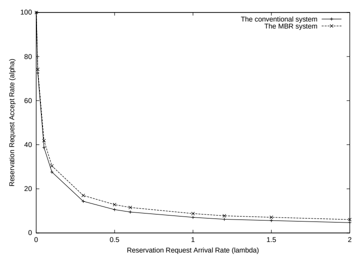
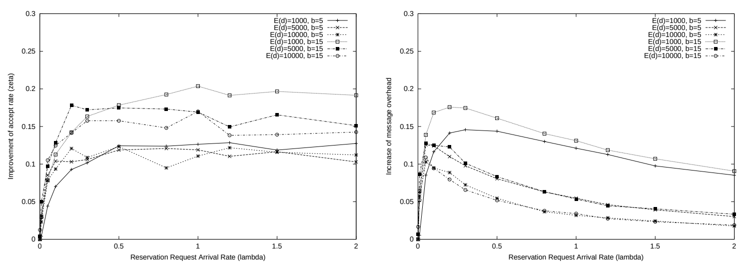


Figure 11: α vs λ



(a) Accept rate improvement ζ vs request arrival rate λ

(b) Increase of message overhead vs request arrival rate λ

Figure 12: Simulation results w.r.t. arrival rate (λ)

Figure 12 illustrates the improvement of acceptance rate ζ (Equation 3) and the increase of message overhead versus request arrival rate λ. Figure 12(a) shows that ζ increases with a larger λ, especially during the interval from 0 to 0.5. Meanwhile, Figure 12(b) shows that the message overhead increases too, but the increase is in line with the acceptance rate improvement. We notice that for a smaller reservation duration d or a larger maximum required bandwidth b we can get larger ζ (the curve of $(E(d) = 1000, b = 15)$ is above the curve of $(E(d) = 10000, b = 15)$, and the curve of $(E(d) = 1000, b = 15)$ is above the curve of $(E(d) = 1000, b = 5)$).

More details in the simulation and the results are shown in [10]. The simulation results clearly show that:

1. The MBR protocol does improve the overall request acceptance rate at the expense of limited extra message overhead.
2. The performance improvement in the simulations does not reach the theoretical analysis results. The possible reason is that, in our theoretical analysis, we considered only one node, while in the simulations, we considered a much more complicated system which consists of 14 nodes. And the traffic pattern in the simulations is also much more complicated than those assumptions we made in the theoretical analysis.

In our simulations we did not take the "hot spots" in the Internet into consideration. We project that with some "hot spots" in the topology we can get even better results. Another issue we did not address is the influence from different reservation schemes, for example, multi-path reservation, multi-re-trial reservation etc. But we can also expect a better performance in these cases since multi-path or multi-re-trial reservations can be considered as reservations with a larger arrival rate λ.

VI. CONCLUSION

In this paper, we formulated the NKR problem and presented the MBR protocol as a solution to it. Both the NKR problem and the MBR protocol were analyzed theoretically by using Markov chains. Simulations were also carried out. The theoretical analysis and the simulation results show clearly that the MBR protocol improves the overall request acceptance rate especially in a high-loaded reservation system where the NKR problem is more serious. The simulation results also show that the improvement is achieved at a low cost with respect to the message overhead. Hence, the NKR problem can be solved by using the MBR model.

This paper is only the first step in studying the MBR protocol. The future work includes: (1) The pre-reservation ratio is an important parameter for the MBR protocol. Its role will be addressed in the future; (2) Poisson distribution is used for the request arrival rate. The effects of other distributions will be studied by using simulations; and (3) Some QoS metrics in end-to-end bandwidth reservations, such as the possible extra delay incurred by the MBR protocol and the possible memory overhead for keeping states in nodes etc., are not covered in this paper. They will be studied as a future work.

VII. REFERENCES

[1] R. Braden and L. Zhang. Resource ReSerVation Protocol (RSVP) - Verion 1 Functional Specification. *RFC 2205*, September 1997.

[2] Robert Braden, Deborah Estrin, Steven Berson, Shai Herzog, and Daniel Zappala. The Design of the RSVP Protocol. *RSVP Project: Final Report*, June 1995.

[3] 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.

[4] Israel Cidon, Raphael Rom, and Yuval Shavitt. Analysis of Multi-Path Routing. *IEEE/ACM Transactions on Networking*, 7(6):885–896, December 1999.

[5] Israel Cidon, Raphael Rom, and Yuval Shavitt. Bandwidth Reservation for Bursty Traffic in the Presence of Resource Availability Uncertainty. *Computer Communications*, 22(10):919–929, June 25th 1999.

[6] K.Nichols, V.Jacobson, and L.Zhang. A Two-bit Differentiated Services Architecture for the Internet. *RFC 2638*, July 1999.

[7] Nageswara S.V. Rao and Stephen G. Batsell. QoS Routing Via Multiple Paths Using Bandwidth Reservation. In *IEEE INFOCOM'98, San-Francisco, CA*, March 29 - April 2 1998.

[8] Ralf Steinmetz and Klara Nahrstedt. *Multimedia: Computing, Communications and Applications*. Prentice Hall PTR, Upper Saddle River, NJ 07458, 1995.

[9] C. Topolcic. Experimental Internet Stream Protocol, Version 2 (ST-II). *RFC 1190*, October 1990.

[10] Jun Wang and Klara Nahrstedt. Marginable Bandwidth Reservation. *Technical Report UIUCDCS-2000-2178, Department of Computer Science, University of Illinois at Urbana-Champaign*, 2000.