

# Analysis of Resource Reservation Aggregation in On-Board Networks

Muhammad Ali Malik<sup>\*,†</sup> Lavy Libman<sup>†</sup>  
\*School of Computer Science and Engineering  
University of New South Wales  
Sydney, NSW 2052, Australia  
{mamalik,salilk,mahbub}@cse.unsw.edu.au

Salil S. Kanhere<sup>\*</sup> Mahbub Hassan<sup>\*,†</sup>  
<sup>†</sup>National ICT Australia  
Bay 15, Australian Technology Park  
Eveleigh, NSW 1430, Australia  
Lavy.Libman@nicta.com.au

**Abstract**—The concept of providing mobile Internet connectivity for passengers in public transport vehicles, where users connect to a local network that attaches to the Internet via a mobile router and a wireless link, has become increasingly popular in recent years, as evidenced by the growing amount of commercially available systems and associated research and standardization activities. The challenge of providing wireless connectivity to networks in motion is compounded by the highly dynamic nature of the user population and the strict Quality-of-Service (QoS) requirements of many applications typical of such environments. As a result, several protocols extending Internet QoS support approaches to on-board mobile networks have been proposed in the past. In this paper, we focus on modeling and performance evaluation of periodical aggregation of resource reservation messages, which forms the basis of the On-Board RSVP protocol. We present a model consisting of a discrete-time, multiple-server and finite-capacity queueing system with bulk arrivals and departures, conduct a detailed analysis of the model, and use it to evaluate the performance of the resource reservation aggregation scheme in a practical scenario. The validity of our model is also backed by extensive simulation results.

## I. INTRODUCTION

Recent years have witnessed an explosive growth in the availability of interconnected computing devices (e.g., PDAs, laptops, and 3G mobile phones) and the deployment of more sophisticated wireless communication infrastructure. In order to achieve a truly pervasive computing environment it is imperative for Internet services to be introduced in public transport systems. An on-board communication solution will enable transport operators to deliver value-added communication, information and entertainment services to their passengers.

A typical on-board mobility architecture consists of three main components: a high-speed on-board local area network (OBLAN), a Mobile Router (MR), and a Mobile Wireless Internet Connection (MWIC). The OBLAN provides local high-speed connectivity for on-board passengers and their devices. The MR facilitates communication between the OBLAN and the global communication infrastructure (e.g., Internet). The MWIC connects the MR to the Internet through a time-varying point of attachment, e.g. via a 3G cellular station. Recently, the IETF Network Mobility (NEMO) working group proposed a basic solution [1] for mobility support for networks

in motion; in addition, the on-board network architecture forms the basis for research projects [2] as well as several commercial deployments.

On-board networks must be able to support a variety of user applications, particularly those requiring end-to-end resource provisioning, such as voice and video calls. Among resource management mechanisms, those offering the most fine-grain control operate on a per-flow basis. However, flow-based reservation schemes such as RSVP [3] can cause significant bandwidth, memory and processing overheads for nodes along the flow path. These overheads increase proportionally with the number of QoS-supported sessions, which, in general, presents a scaling problem due to the volume of RSVP refresh messages. Nonetheless, in on-board networks, the number of sessions is limited both by the capacity of the MWIC and by the number of people and devices that can physically be on-board; therefore, for this kind of networks, scalability of RSVP refresh messages is not considered a significant issue.

On the other hand, in many applications requiring strict QoS guarantees (such as voice calls), the session duration can be relatively short. The dynamic of adding or dropping user sessions institutes a large number of *trigger* (or *setup*) resource reservation messages in the network. This can result in a large load on the intermediate routers, since the processing of trigger messages requires various stages such as a new reservation session lookup, setting up new reservation states and setting the internal traffic control. Further, trigger messages introduce bandwidth overheads. Previously proposed schemes for overhead reduction in RSVP [4]–[7] focused on reducing the number of *refresh* messages in the network, but do not necessarily result in any decrease in the *trigger* messages. The problem of trigger messages is also identified in [8]. The scalability problem introduced by trigger messages tends to become more severe in an on-board communication setting, due to the potentially large and dynamic number of users and the scarcity of bandwidth in the wireless link (MWIC).

To address this issue, in an earlier work [9], we proposed On-Board RSVP, which is designed to reduce the frequency of trigger messages over the tunnel between the MR and its home agent (HAoMR) by collecting individual trigger messages in a buffer and aggregating them into a single compressed one. Some of the possible aggregation criteria have subsequently been discussed in [10]; e.g., aggregation

<sup>†</sup>National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

can be time-based, where the accumulated trigger messages are handled periodically with a regular interval, or be based on reaching a threshold value of either the number of requests or the total capacity requested. Our objective in this paper is to develop an analytical model to evaluate the performance of the On-Board RSVP protocol with the time-based aggregation criterion; thus, the compressed trigger messages are sent only at regularly spaced points in time, and the underlying resource reservation mechanism is therefore represented adequately by a discrete-time queueing system [11].

Discrete-time queues received much attention in the literature in recent years due to their direct applicability in the study of many computer and communication systems, where time is divided into fixed-length intervals (slots). Several studies carried out an analysis of single-server, discrete-time  $Geom/G/1$  and  $GI/Geom/1$  systems with both infinite and finite queue capacities [12], [13]. An analysis of a single-server discrete-time  $M^{[x]}/Geom/1$  queue with multiple arrivals per slot was undertaken in [11]. Multi-server discrete-time systems with geometrically distributed inter-batch and service times and infinite buffer capacity ( $Geom^{[x]}/Geom/c$ ) were the subject of [14], [15]. Recently, [16] reported an analysis of a finite-size multi-server discrete queue ( $GI/Geom/m/N$ ), but with a single arrival per slot only. On the other hand, since On-Board RSVP facilitates multiple arrivals of trigger messages per time interval (slot), and since the lifetime of QoS-supported connections can be random, our model must allow for bulk arrivals in a slot as well as multiple parallel servers. Thus, our model for the operation of On-Board RSVP is that of a discrete-time, multi-server, finite-buffer queue with bulk arrivals, i.e. a  $M^{[x]}/Geom/c/N$  system. To the best of our knowledge, no published work has attempted to study queueing systems combining all the above characteristics (the references cited above only tackled different partial combinations thereof).

Accordingly, our contribution in this paper is twofold. First, in Section II, we present in detail the  $M^{[x]}/Geom/c/N$  queueing model and conduct an analysis thereof, which has a generic importance beyond the motivating context of this paper. Then, in Section III, we use the model to study the performance of On-Board RSVP under a practically-inspired high-load scenario, and demonstrate the validity of the model by comparing it to simulation results. Finally, we conclude the paper in Section IV.

## II. DISCRETE-TIME $M^{[x]}/Geom/c/N$ QUEUEING MODEL

We consider the  $M^{[x]}/Geom/c/N$  queue with multiple servers and finite storage capacity, as illustrated in Fig. 1.  $c$  denotes the number of servers and  $N$  is the queue capacity. Thus,  $c + N$  is the maximum number of messages allowed in the system, including those in service. The time is considered to be divided into fixed-length periods, or slots, of duration  $T$ . New messages continuously arrive to the system as a Poisson process with a rate of  $\lambda$  and are assigned to the first available server, possibly waiting for several slots in the queue if necessary. A message that finds the queue to be full upon arrival is lost and immediately discarded from the

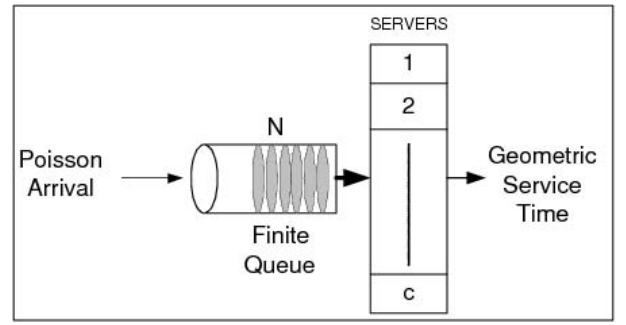


Fig. 1. The  $M^{[x]}/Geom/c/N$  discrete-time queue.

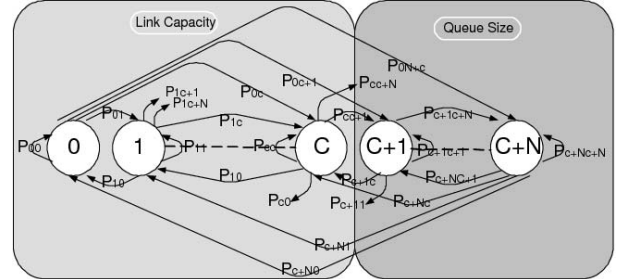


Fig. 2. Markov chain diagram of the queueing model

system. The queue service discipline is first-come-first-served (FCFS). The service of a message always starts and ends at slot boundaries, and its duration (in slot units) is independently and geometrically distributed with a parameter of  $\mu$ .

Consider the system population (i.e. total number of messages in the queue and the servers) at the start of some time slot. At the end of that slot, the population will increase by the number of newly arrived messages (excluding those discarded due to a full queue), minus the number of messages whose service is completed in that slot. Denote by  $a_n$  the probability of  $n$  arrivals in a slot, and  $s_n^m$  the probability of  $n$  service completions in a slot, given that  $m$  messages were serviced. Thus,  $a_n$  is Poisson-distributed:

$$a_n = \frac{(\lambda T)^n e^{-\lambda T}}{n!}, \quad n = 0, 1, 2, \dots, \quad (1)$$

and, due to the independence of service times,  $s_n^m$  is distributed binomially:

$$s_n^m = \binom{m}{n} \mu^n (1-\mu)^{m-n}, \quad 0 \leq m \leq c, \quad 0 \leq n \leq m. \quad (2)$$

Fig. 2 shows the Markov chain representation of the system with finite state space  $\{0, 1, 2, \dots, c + N\}$ . The one-step transition probabilities  $p_{ij}$  from state  $i$  to state  $j$  are given by the following:

$$p_{ij} = \begin{cases} \sum_{l=0}^j a_l s_{i-j+l}^i & 0 \leq i \leq c, 0 \leq j < i; \\ \sum_{d=0}^i a_{j-i+l} s_d^i & 0 \leq i \leq c, i \leq j < c+N; \\ \sum_{d=0}^i \left( \sum_{\nu=j-i+d}^{\infty} a_{\nu} \right) s_d^i & 0 \leq i \leq c, j = c+N; \\ \sum_{l=0}^{j-i+c} a_l s_{i-j+l}^c & c < i \leq c+N, \\ & i-c \leq j < i; \\ \sum_{d=0}^c a_{j-i+l} s_d^c & c < i \leq j < c+N; \\ \sum_{d=0}^c \left( \sum_{\nu=j-i+d}^{\infty} a_{\nu} \right) s_d^c & c < i \leq c+N, \\ & j = c+N; \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

It is easily seen that the Markov chain is irreducible and positive recurrent. Hence, the steady state probabilities  $\pi_j$ ,  $j = 0, 1, \dots, c+N$ , are given by the unique solution of the following linear equations:

$$\pi_j = \sum_{i=0}^{c+N} p_{ij} \pi_i = \begin{cases} \sum_{l=0}^j a_l \sum_{d=0}^{\min(c, c+N-j+l)} s_d^{\min(c, j+h-l)} \pi_{j+h-l} & 0 \leq j < c+N; \\ \sum_{l=0}^j \left( \sum_{\nu=l}^{\infty} a_{\nu} \right) \sum_{d=0}^{\min(c, c+N-j+l)} s_d^{\min(c, j+h-l)} \pi_{j+h-l} & j = c+N; \end{cases} \quad (4)$$

and

$$\sum_{j=0}^{c+N} \pi_j = 1. \quad (5)$$

This system of linear equations can be solved numerically; note that, for practical purposes, the infinite sum  $\sum_{\nu=l}^{\infty} a_{\nu}$  can obviously be replaced by  $1 - \sum_{\nu=0}^{l-1} a_{\nu}$ .

Once the steady state probabilities are known, they can then be used to evaluate, in particular, the loss rate of incoming messages and the average waiting time in the queue, both of which are important metrics for the On-Board RSVP protocol performance. The mean message loss rate (i.e. the expected number of messages that are discarded per time slot) is calculated by considering, for each state, the probability that the number of arrivals minus service completions exceeds the queue capacity:

$$A_{lost} = \sum_{i=0}^{c+N} \sum_{l=1}^{\infty} \sum_{d=0}^{\min(i, c)} \pi_i a_l s_d^{\min(i, c)} \max\{[i+l-d-(c+N)], 0\}. \quad (6)$$

The percentage of messages lost,  $P_{lost}$ , can therefore be found by  $P_{lost} = \frac{A_{lost}}{\lambda T}$ , and the 'effective' arrival rate seen by the system (excluding the discarded messages) is  $\lambda_A = \lambda - \frac{A_{lost}}{T}$ .

By Little's formula, the average time spent in the system can now be found by

$$W_S = \frac{\sum_{i=0}^{c+N} i \cdot \pi_i}{\lambda_A}; \quad (7)$$

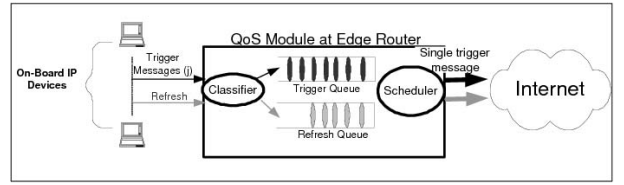


Fig. 3. Handling of 'trigger' and 'refresh' resource reservation messages at the MR.

subtracting the average time that it takes a message to be served ( $\frac{T}{\mu}$ ), we finally obtain the average waiting time in the queue as  $W_D = W_S - \frac{T}{\mu}$ .

### III. SIMULATION STUDY

The On-Board RSVP protocol, proposed in [9], specifies that the on-board MR includes a mechanism to classify resource reservation messages into two queues, as shown in Fig. 3. Incoming RSVP messages are checked by the classifier against an existing reservation state in order to determine whether they are 'trigger' or 'refresh' messages. After every period of  $T$ , the scheduler scans the queue of trigger messages and compresses some of them (depending on the available capacity) into a single trigger message, which is then sent over the MWIC to the MR's home agent (HAoMR). The HAoMR decompresses this message back into multiple trigger messages and sends them on to their respective endpoints. Any trigger messages that find the queue full upon arrival are not admitted. Note that this mechanism is entirely orthogonal to any possible optimization of *refresh* messages, such as the one proposed in [17]. We refer the reader to [9] for a more detailed description of the On-Board RSVP protocol.

We now describe the scenario used for our simulation study. We assume an on-board network deployment on a public transport vehicle using a 3G cellular connection with a maximum data rate of 144Kbps. The network is utilized solely by VoIP calls, using a G.729a 8Kbps codec, RTP header compression and voice activity detection (silence suppression), and thus requiring 12Kbps per call on average. Therefore, the maximum number of simultaneous calls that can be supported is  $c = 12$ . We assume an average call duration is 120sec, and set the arrival rate of new calls to be 0.1calls/sec. Thus, demand is equal to 100% of capacity, which represents a high but not unreasonable load on the network.<sup>1</sup> Our goal is to investigate the effect of the aggregation period  $T$  and the queue size  $N$  on the protocol performance metrics,  $P_{lost}$  (percentage of lost messages) and  $W_D$  (mean queue waiting time). To that end, we evaluate these metrics using the  $M^{[x]}/Geom/c/N$  model and compare them with actual results obtained by simulation.

Figures 4 and 5 show  $P_{lost}$  and  $W_D$ , respectively, as a function of  $N$  for  $T = 30$ sec, and as a function of  $T$  for  $N = 12$ . We observe, as expected, that the percentage of lost

<sup>1</sup>Indeed, e.g., for a suburban train with 150 passengers, a demand of 12 simultaneous calls at any time imply that each user has an active duty cycle of up to 8% on average.

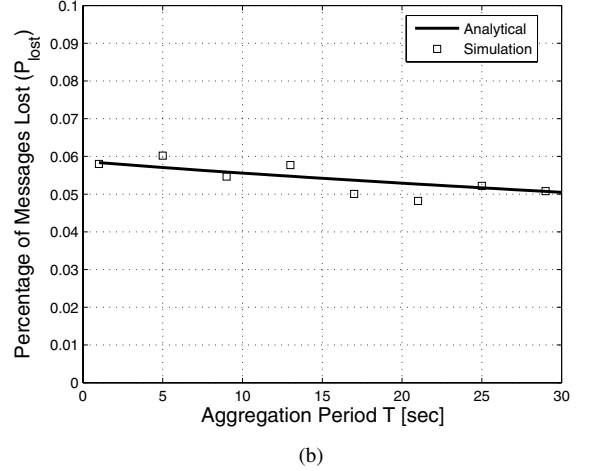
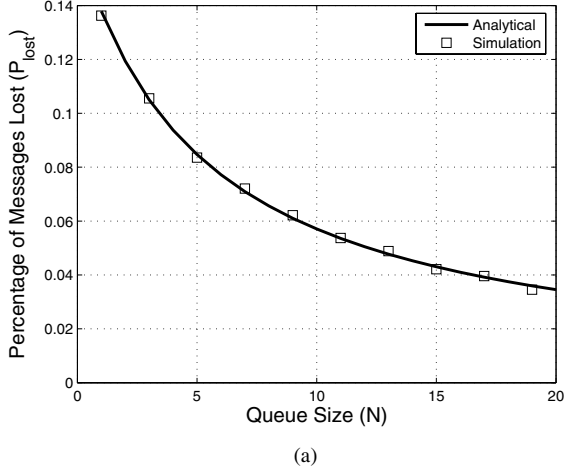


Fig. 4. Percentage of Messages Lost  $P_{lost}$ : (a) as a function of  $N$  for  $T = 30\text{sec}$ ; (b) as a function of  $T$  for  $N = 12$ .

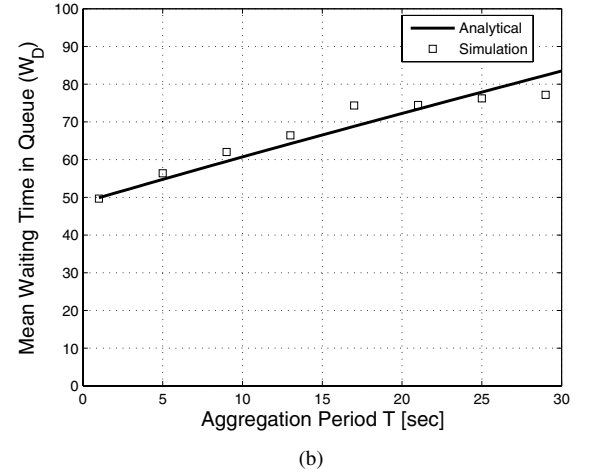
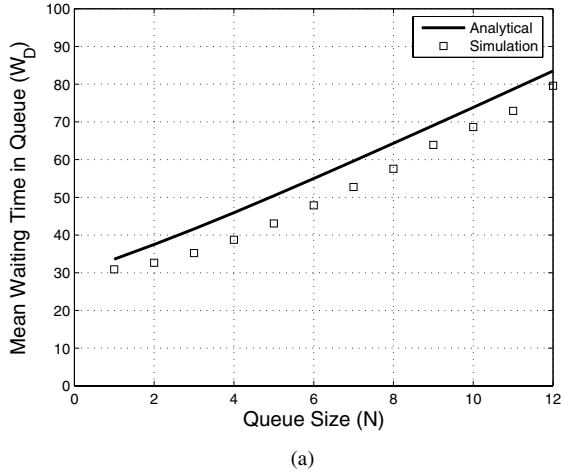


Fig. 5. Mean Waiting Time in Queue  $W_D$ : (a) as a function of  $N$  for  $T = 30\text{sec}$ ; (b) as a function of  $T$  for  $N = 12$ .

messages decreases considerably with increasing queue size, at the expense of a proportional increase in the waiting time. More importantly, both metrics are clearly seen to have a very benign dependence on  $T$ . This suggests that the additional wait due to aggregation is marginal compared to the queue waiting time due to the high load; furthermore, we curiously note that  $P_{lost}$  may even slightly *decrease* with  $T$ . These observations lead to the conclusion that the cost of reduced performance due to the aggregation process of On-Board RSVP is small compared to the benefit of reducing the signalling overhead.

To quantify the latter benefit with precision, we define the *sending rate*  $S_R$  as the average number of times per second that a compressed trigger message is sent. Since such a message is sent whenever there is at least one unit of capacity available at the end of the slot and there are messages waiting

in the queue (or arriving during that slot), we have

$$S_R = \frac{1}{T} \left[ \left( \sum_{i=0}^{c-1} \pi_i \right) \cdot (1 - a_0) + \pi_c \cdot (1 - a_0) (1 - s_0^c) + \left( \sum_{i=c+1}^{c+N} \pi_i \right) (1 - s_0^c) \right]. \quad (8)$$

The sending rate is plotted in Fig. 6 as a function of  $N$  for  $T = 30\text{sec}$  and of  $T$  for  $N = 12$ . We see that the queue size has virtually no impact on  $S_R$ ; however, the sending rate decreases considerably as  $T$  increases. Thus, On-Board RSVP significantly improves the scalability in terms of signalling and processing overheads with very little sacrifice of performance metrics, especially in high-load scenarios where the total on-board rate demand is close to the MWIC capacity.

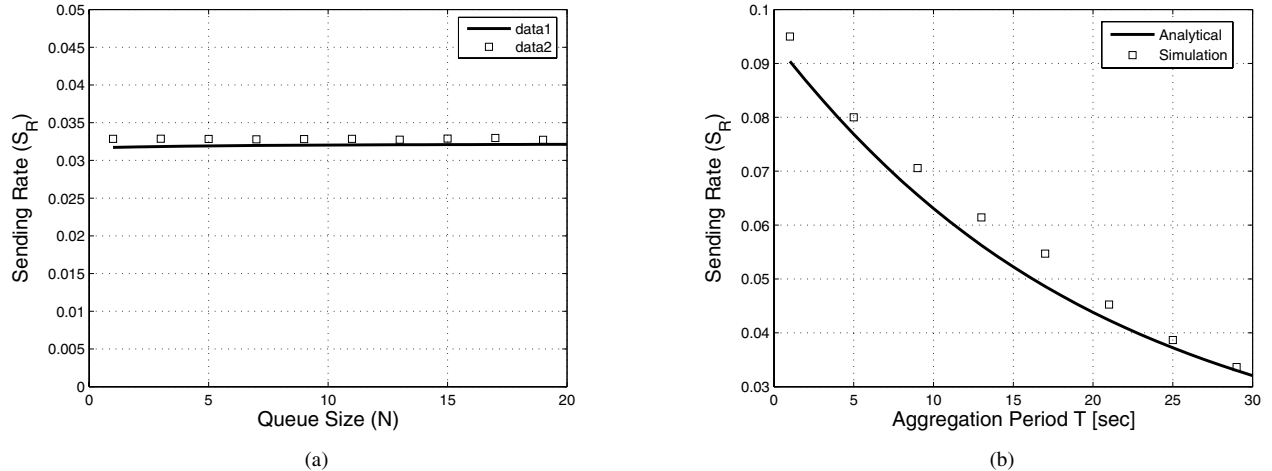


Fig. 6. Sending Rate  $S_R$ : (a) as a function of  $N$  for  $T = 30$ sec; (b) as a function of  $T$  for  $N = 12$ .

#### IV. CONCLUSION

In this paper we have modelled the resource reservation process for an on-board network using a discrete-time  $M^{[x]}/Geom/c/N$  queue. The reservation process is based on our proposed protocol known as On-Board RSVP [9]. We have performed a generic analysis of this queue and obtained the equations describing its steady-state behavior, which are easy to solve numerically due to their linearity. We then used this model to evaluate the performance of the On-Board RSVP protocol, and showed that it decreases the processing and signalling overheads considerably while incurring only an insignificant performance cost in terms of waiting time for connections with QoS requirements, particularly for high load scenarios where the total rate demand in the on-board network is comparable to the capacity of the wireless link connecting it to the Internet.

The tradeoff introduced by the On-Board RSVP aggregation can be captured quantitatively by defining a cost function that accounts for, on one hand, the waiting time and loss rate of connections, and on the other, the sending rate overhead of compressed RSVP trigger messages. This cost function can then be used to find optimal settings of the On-Board RSVP operating parameters, namely, the queue capacity and the aggregation period. A detailed study of the resulting two-dimensional optimization problem is the subject of ongoing work.

#### REFERENCES

- [1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. RFC 3963: Network mobility (NEMO) basic support protocol, January 2005.
- [2] Nautilus 6 working group: <http://www.nautilus6.org>.
- [3] Ed. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205: resource ReSerVation Protocol (RSVP) – version 1 functional specification, September 1997.
- [4] F. Baker, C. Iturralde, F. Le Faucheur, and B. Davie. RFC 3175: Aggregation of RSVP for IPv4 and IPv6 reservations, September 2001.
- [5] A. Terzis, J. Krawczyk, J. Wroclawski, and L. Zhang. RFC 2746: RSVP operation over IP tunnels, January 2000.

- [6] P.P. Pan, E.L. Hahne, and H. Schulzrinne. BGRP: Sink-tree-based aggregation for inter-domain reservations. *Journal of Communications and Networks*, 2(2):157–167, June 2000.
- [7] O. Schelén and S. Pink. Aggregating resource reservations over multiple routing domains. In *Proc. International Workshop on Quality of Service (IWQoS)*, pages 29–32, Napa, CA, May 1998.
- [8] J. Manner and X. Fu. RFC 4094: Analysis of existing quality-of-service signaling protocols, May 2005.
- [9] M.A. Malik, S.S. Kanhere, M. Hassan, and B. Benatallah. On-board RSVP: An extension of RSVP to support real-time services in on-board IP networks. *Springer Lecture Notes in Computer Science (LNCS)*, 3326/2004:264–275.
- [10] M.A. Malik, S.S. Kanhere, and M. Hassan. Aggregation policies over RSVP tunnels. In *Proc. IEEE Vehicular Technology Conference (VTC)*, Dallas, TX, September 2005.
- [11] M.E. Woodward. *Communication and Computer Networks: Modelling with Discrete-Time Queues*. IEEE Computer Society Press, Los Alamitos, CA, 1993.
- [12] M.L. Chaudhry, U.C. Gupta, and J.G.C. Templeton. On the relations among the distributions at different epochs for discrete-time  $GI/Geom/1$  queues. *Operational Research Letters*, 18(5):247–255, March 1996.
- [13] M.L. Chaudhry and U.C. Gupta. Performance analysis of the discrete-time  $GI/Geom/1/N$  queue. *Journal of Applied Probability*, 33(1):239–255, March 1996.
- [14] S. Wittevrongel, H. Bruneel, and B. Vinck. Analysis of the discrete-time  $G^{[G]}/Geom/c$  queueing model. *Springer Lecture Notes in Computer Science (LNCS)*, 2345/2002:757–768.
- [15] P. Gao, S. Wittevrongel, and H. Bruneel. Discrete-time multiserver queues with geometric service times. *Computers and Operations Research*, 31(1):81–99, January 2004.
- [16] M.L. Chaudhry, U.C. Gupta, and V. Goswami. On discrete-time multiserver queue with finite buffer:  $GI/Geom/m/N$ . *Computers and Operations Research*, 31(13):2137–2150, November 2004.
- [17] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini. RFC 2961: RSVP refresh overhead reduction extensions, April 2001.