Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Performance Evaluation 52 (2003) 221–235

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

A model of periodic acknowledgement

Alexandre Brandwajn

Computer Engineering Department, Jack Baskin School of Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064, USA

Received 4 December 2000; received in revised form 7 August 2002

Abstract

We study a problem abstracted from modeling a multicast protocol. In our model, messages generated by a single source are simultaneously forwarded to a set of receivers where they are independently processed. We assume a state-dependent message arrival rate and memoryless service time distributions. The receivers may process messages at different average rates. Messages processed by all receivers are periodically acknowledged and cleared from the system. Due to finite buffer space, the total number of non-acknowledged messages in the system is limited. Our focus in this paper is on the number of messages cleared at acknowledgement time.

The problem under consideration bears resemblance to a fork/join process with heterogeneous servers, used in the study of multiprocessing computer systems. Our model includes the additional features of finite buffer space and delayed periodic departure of completed jobs. Even without these features, the resulting type of queuing model has no known closed-form solution in the general case of more than two servers. Because the arrival processes to the servers are correlated, the model is difficult to decompose. We propose a relatively simple decomposition technique and a fixed-point iteration scheme. In our approach, we consider each receiver (server) in isolation, and we account for the influence of other receivers through the probability that a given number of messages can be cleared at acknowledgement time. We derive elementary differential equations for the number of messages processed by a receiver. These equations involve the conditional probability of the number of messages not yet processed given the number of messages waiting to be cleared. We compute an approximate solution using the conditional probability obtained from a model with exponentially distributed acknowledgement periods. Our numerical results for the average number of messages cleared at acknowledgment time are typically within a few percent of simulation midpoints.
© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Performance evaluation; Fork/join queuing networks; Finite buffers; Computer networks; Multicast protocols

1. Introduction

This paper considers a problem abstracted from modeling a multicast protocol (see e.g. [1]) taking into account finite buffer space. Specifically, messages generated by a single source are simultaneously forwarded to a set of receivers where they are independently processed. We assume a state-dependent message arrival rate and memoryless service time distributions. The receivers may or may not process messages at the same average rate. Messages processed by all receivers are acknowledged and cleared from the

E-mail address: alexb@soe.ucsc.edu (A. Brandwajn).

system within a specified time window. Due to finite buffer space, the total number of non-acknowledged messages in the system is limited. Our focus in this paper is on the number of messages cleared at acknowledgement time.

Multicast communication has received considerable attention in the last decade [2,3] due to the potential development of new applications such as Internet TV, distributed games, e-learning or distributed interactive simulation. Multicast is the ability to send efficiently a flow of packets to a set of (possibly dynamic) receivers, packets being sent once on a given link. Many issues have been studied in this framework such as routing protocols (DVMRP, PIM), mechanisms to enforce reliability (scalable reliable multicast (SRM) [4], RAMP [5], RMTP [6], MTP [7], etc.) or congestion control (PGMCC [8], ECAM [9], TFMCC [10]). In this paper, we focus on the issue of multicast reliability as pioneered by XTP [11]. Reliable multicast protocols perform two major activities that are error detection and error recovery. ARQ (automatic repeat request) solutions have emerged as means to retransmit a packet if it is lost by at least one receiver. Depending on whether error detection is done by the sender or the receivers, positive (ACK) or negative (NACK) acknowledgements are used. Solutions using solely ACK require the sender to retransmit packets until ACK from all receivers are properly received, and, thus, do not scale (ACK implosion problem). NACK-based solutions were preferred and have lead to the introduction of several NACK suppression mechanisms. Moreover, FEC (forward error correction) proposals emerged and were associated with ARQ mechanisms [3,12].

Since no single protocol is able to fit all needs, a large number of reliable multicast protocols were designed according to various objectives. SRM [4] is an NACK-based multicast protocol that uses a variation of the slotting and dumping mechanism introduced by XTP. ALC (asynchronous layered coding [13]) is an SRM protocol using several data streams in a layered fashion. MFTP [14] or RMTP are based on cycles that divide a file into a sequence of fixed-size packets. The file is transmitted to all receivers that reply with a list of missing packets. A new cycle is started while sending again the lost packets until the original file has been successfully received. Finally, another popular solution is to build ACK aggregation trees to improve scalability. A local representative (designated router, group controller or surrogate server) is introduced to perform some specific functions so that receivers are grouped in local domains. A tree hierarchy is constructed, and NACKs are sent to the local representatives, which retransmit the missing packets if they hold them or request them from an upper level in the tree if they do not have them [15–17]. Although a tremendous amount of work was done in the area of group communication and multicast protocols, deployment problems have limited the availability of this technology in commercial networks. Few carriers provide a commercial multicast service nowadays, and alternative solutions have emerged such as CDN (content distribution networks [18]) or application-level multicast [19–21]. Therefore, it is important to provide tools to properly design, manage and configure multicast networks and protocols.

In this paper, we consider the original multicast protocol as introduced in XTP. We now briefly describe the essential mechanism on which we base our analysis. The multicast flow is controlled using a time window. At the beginning of the window, the sender sends a control packet identified by a sequence number. A receiver acknowledges such a control packet by a packet of its own identified by the same sequence number. Therefore, the sender expects to receive acknowledgements from the set of receivers for a given control packet within this time window. Reply packets received within a time window for a given control packet are accumulated and analyzed by the sender in order to extract the most conservative values. The state of every transfer for a given control packet is stored in a bucket, which is a data structure containing fields of interest in the management of the error recovery procedure such as sequence numbers. The system uses a fixed number of buckets to manage the multicast flow. The periodicity at which the control packets are generated is referred to as the control period. The lifetime of a bucket extends from the moment the bucket is created to the moment it has to be recycled to make space available to another control packet. To ensure correct operation, this lifetime should be larger than the round-trip delay between the sender and the receivers. Finally, when a bucket has to be freed, the information it contains allows the sender to determine which packets (if any) need to be retransmitted. The model developed in this paper can also be applied to other protocols where the sender updates periodically its buffer with all the acknowledgements received between two control (update) periods. Moreover, the sender could also be a local representative in an ACK aggregation tree solution.

The problem under consideration bears resemblance to a fork/join process with heterogeneous servers with the added "wrinkle" of finite buffer space and delayed periodic departure of completed jobs through the acknowledgement process. Even without these features, the resulting models are notoriously difficult to deal with. As stated in [22], because the arrival processes to the servers are correlated, the model is difficult to decompose. The exact analytical solution is only known for the particular case of two homogeneous servers [23,24]. For a larger number of servers, approximation and heuristic methods have been proposed. For example, Dallery et al. [25,26] use a graph method to describe the evolution of complex fork/join queues. Thomasian and Tantawi [27] consider a set of homogeneous servers with infinite capacity and propose a heuristic using values derived from simulation. Frostig and Lehtonen [28] use stochastic ordering and a judiciously chosen state description to study a system with non-homogeneous servers. Lui et al. [22] consider infinite capacity homogeneous servers for parallel processing applications in computer systems, and derive an approximation based on the non-uniform distribution of the state space. Balsamo et al. [29] derive approximations and bounds for a heterogeneous network abstracted from parallel processing. Varki [30] studies the mean response time in closed fork/join networks using an approximate *mean value analysis* technique. None of these papers seems readily applicable to our model with periodic acknowledgement.

We propose a relatively simple decomposition technique and a fixed-point iteration scheme. Our approach, believed to be novel, allows us to consider each receiver (server) in isolation while accounting for the presence of other receivers through the probability distribution of the number of messages that can be cleared at acknowledgement time. We derive elementary differential equations for the number of messages processed by a receiver. These equations involve the conditional probability of the number of messages not yet processed given the number of messages waiting to be cleared. We are able to obtain an approximate solution using the conditional probability computed from a similar model with exponentially distributed acknowledgement periods. We use discrete-event simulation to assess the accuracy of our method. Our numerical results for the average number of messages cleared at acknowledgment time are typically within a few percent of simulation midpoints.

In the next section, we consider the model of a single receiver. Numerical results in this section illustrate the accuracy of our approximation for constant acknowledgment periods. Section 3 is devoted to multiple receivers. It includes numerical results for both homogeneous and non-homogeneous systems. Finally, conclusions are given in Section 4.

2. Single receiver case

We start by considering the system with a single receiver depicted in Fig. 1. Messages arrive from an outside source with a state-dependent rate $\lambda(n)$ where n is the current total number of messages in the

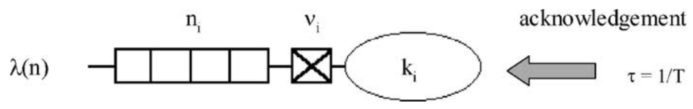


Fig. 1. Acknowledgement model with a single receiver.

system. We assume that the maximum number of messages that can be present in the system at any time is given by N (corresponding, for instance, to buffer space limitations), so that $n \le N$ at all times.

Incoming messages queue for service at the receiver. We denote by n_i the current number of messages at the receiver, and by ν_i the rate at which the receiver processes messages. We let k_i denote the current number of messages waiting to be cleared, i.e., processed by the receiver but not yet cleared by the sender upon bucket processing. Processed messages are acknowledged by the receiver and cleared by the sender periodically every T time units. Following the acknowledgement, the number of messages ready to be freed, k_i, drops instantaneously to zero. It then starts building up again as the receiver processes incoming messages. Clearly, we have $n_i + k_i = n$ at all times.

In our model, we view the messages awaiting acknowledgement as attached to the receiver. In a multicast situation, these messages would correspond to a copy of messages sent kept in a sender's buffer. Our goal is to assess the number of messages acknowledged in each acknowledgement period as a function of system parameters. We analyze the model of Fig. 1 under the assumption of memoryless arrivals, i.e., we assume that the probability of a new message arriving during the time interval $(t, t+\delta t]$ is $\lambda(n)\delta t + o(\delta t)$ given that there are n messages at time t. We also assume that the service time at the receiver is exponentially distributed.

We focus on the number of messages awaiting acknowledgement, and we denote by $p(k_i, t)$ the probability that there are k_i such messages at time t, $t \ge 0$. The differential–difference equations for $p(k_i, t)$ can be readily derived as

$$\begin{aligned}
&\frac{d}{dt}p(0,t) = \mu_i(0,t)p(0,t),\\
&\frac{d}{dt}p(k_i,t) = -\mu_i(k_i,t)p(k_i,t) + \mu_i(k_i-1,t)p(k_i-1,t), \quad k_i = 1,\dots,N-1,\\
&\frac{d}{dt}p(N,t) = \mu_i(N-1,t)p(N-1,t),
\end{aligned} \tag{1}$$

where

$$\mu_i(k_i,t) = \nu_i \Pr\{n_i>0|k_i \text{ at time } t\}. \tag{2}$$

In order to simplify the solution of the system of equations (1), we make two assumptions. First, we assume that the time-dependent conditional probability in (2) can be approximated by the corresponding steady-state conditional probability so that we have

$$\mu_i(k_i,t) \approx \mu_i(k_i) = \nu_i \Pr\{n_i > 0|k_i\}, \quad k_i = 1,\dots,N-1. \tag{3}$$

Note that using (3) it is straightforward to solve the equations for $p(k_i, t)$. However, because the values of $\mu_i(k_i)$ for neighboring values of k_i tend to be close, the solution can be difficult to evaluate numerically. Therefore, as a further simplification, we approximate the $\mu_i(k_i)$ by the "average" value

$$\mu_i(k_i) \approx \mu_i = \nu_i \Pr\{n_i > 0|k_i < N\}. \tag{4}$$

With these assumptions, the solution of (1) can be written as

$$p(k_i,t)=\sum_{j=0}^{k_i} C_j \frac{(\mu_i t)^{k_i-j}}{(k_i-j)!}\,\mathrm{e}^{-\mu_i t},\quad k_i=0,\ldots,N-1, \tag{5}$$

where the C_j are determined from the initial conditions. Equating the time $t=0$ with the beginning of an acknowledgement period, we have $p(k=0,t=0)=1$ and $p(k,t=0)=0$, for $k>0$. This yields for the probability that there are k_i messages awaiting acknowledgement at time t

$$p(k_i,t)=\frac{(\mu_i t)^{k_i}}{k_i!}\,\mathrm{e}^{-\mu_i t},\quad k_i=0,\ldots,N-1 \quad\text{and}\quad p(N,t)=1-\sum_{k_i=0}^{N-1} p(k_i,t),\quad k_i=N. \tag{6}$$

From the above equation, we get the probability that there are k_i messages ready to be freed at acknowledgement time as a simple truncated Poisson distribution with parameter $\mu_i T$, where T is the acknowledgment period. For this solution to be of practical value, we must be able to assess the conditional probability $\Pr\{n_i>0|k_i<N\}$ appearing in (4). The exact computation of this probability under the assumption of a constant acknowledgment period is likely to be as difficult as the solution of the initial problem so it may seem that we have reached an impasse. It is, however, not very difficult to solve the model of Fig. 1 for the joint probability $p(n_i,k_i)$ under the assumption of exponentially distributed acknowledgment period. Details of a fixed-point solution can be found in Appendix A.

We make the assumption that the conditional probability in (4) is not strongly influenced by the distribution of the acknowledgement period so that we have

$$\Pr\{n_i>0|k_i<N\}\approx 1-\frac{\sum_{k_i=0}^{N-1} p(n_i=0,k_i)}{\sum_{k_i=0}^{N-1}\sum_{n_i=0}^{N-k_i} p(n_i,k_i)}. \tag{7}$$

In the above equation, $p(n_i,k_i)$ is the probability that there are n_i messages to be processed at the receiver and k_i messages ready to be freed assuming that the acknowledgement period is exponentially distributed.

The approximation expressed in (4) yields results that differ only slightly from those obtained using state-dependent values for $\mu_i(k_i)$ (3) but are much easier to evaluate numerically. Fig. 2 shows examples of numerical results obtained using our approach for a set of values of the acknowledgment period T. The maximum number of messages N is limited to 7. The message arrival rate function used in our examples is $\lambda(n)=\alpha$, for $n=0,\ldots,N-1$, and $\lambda(N)=0$, where α is a constant. The mean service time for a message $1/\nu_i$ is taken to be 5 ms. We have represented in Fig. 2 the expected number of messages freed each acknowledgment period given by

$$\bar{k}_i=\sum_{k_i=1}^{N} k_i\, p(k_i,T). \tag{8}$$

For comparison, we have included the mean number of messages freed estimated from discrete-event simulation runs with seven independent replication of 10,000 acknowledgment periods each. We have also included the value for the mean number of messages freed obtained assuming exponentially distributed acknowledgment periods (labeled *Expon* in the figures). The message arrival rates in the figures correspond to the values of α in the assumed form of $\lambda(n)$. Note the effect of finite buffer space as the acknowledgement period and the message arrival rate increase, e.g., for α of 100 messages/s and T of 40 ms the average

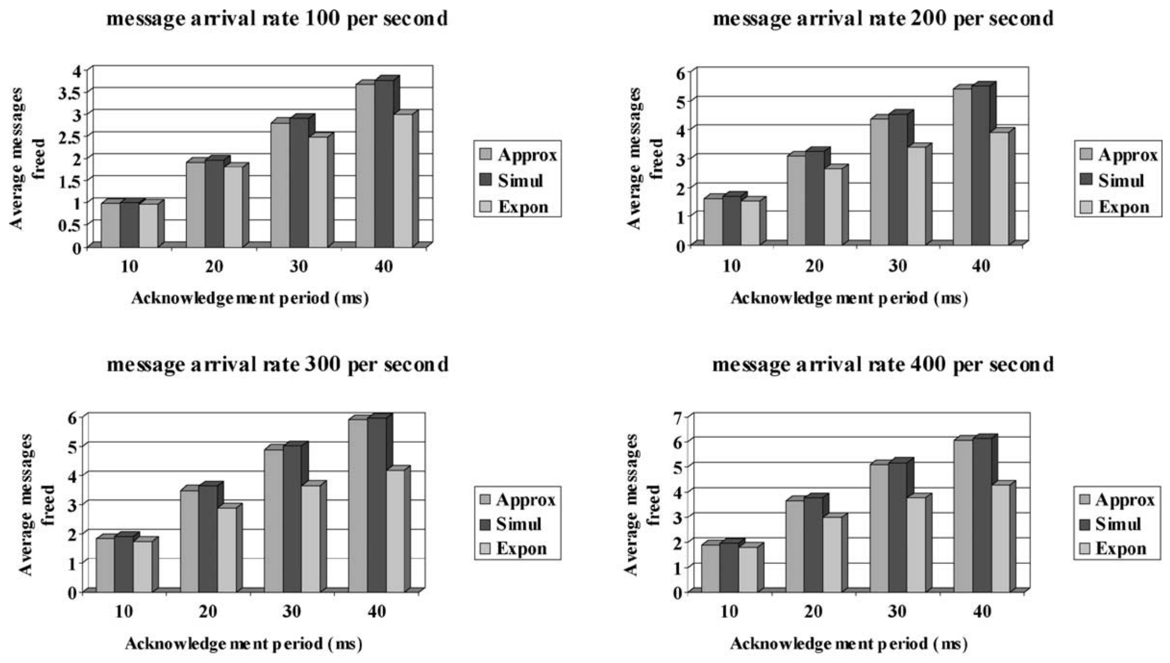


Fig. 2. Results with buffer size $N = 7$.

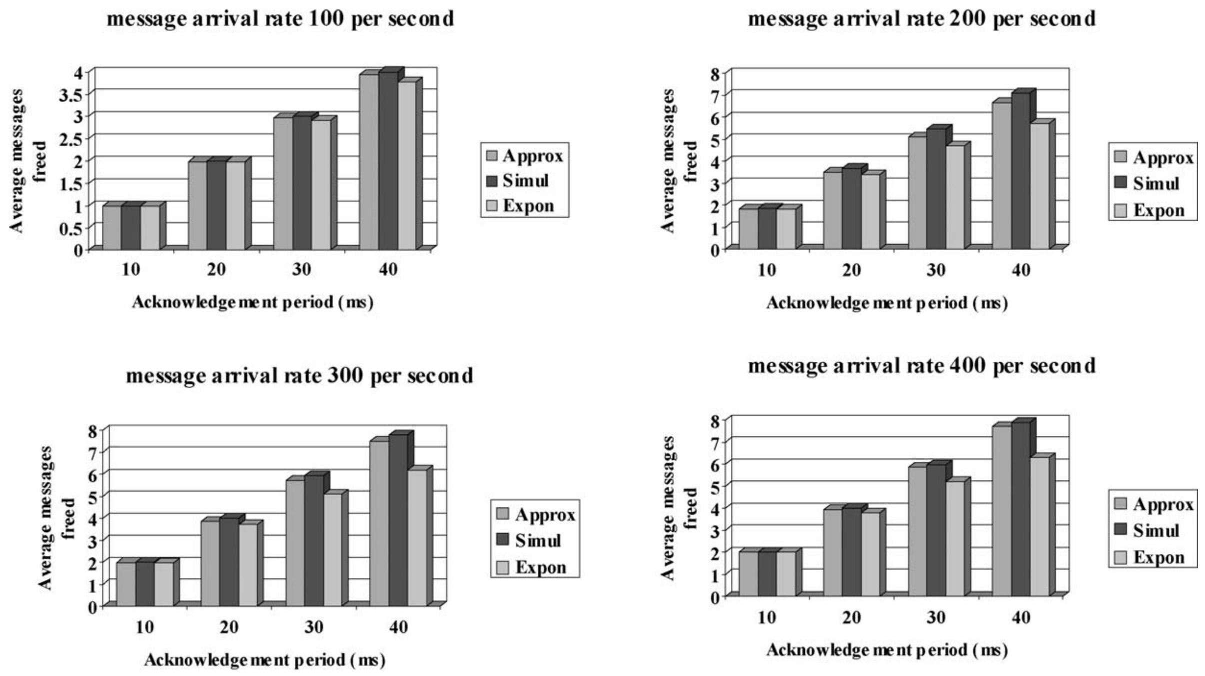


Fig. 3. Results with buffer size $N = 14$.

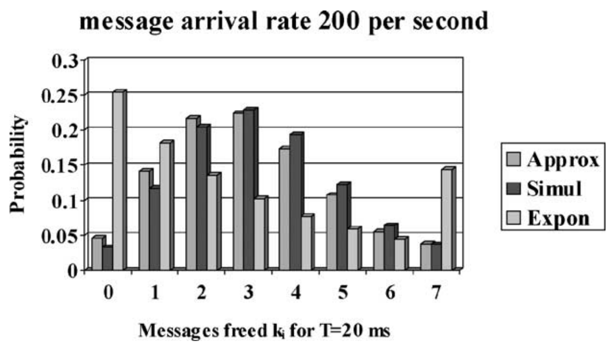


Fig. 4. Shape of the distribution of the number of messages cleared.

number of messages freed is clearly below 4. As the message arrival rate goes up, this effect becomes more apparent.

We observe that our method produces results that deviate from the simulation midpoints by typically less than 5%. By contrast, the relative errors for exponentially distributed acknowledgment grow as the duration of the acknowledgment period, T, increases, and attain some 30% in our examples. Fig. 3 shows analogous results for the case when the maximum total number of messages N is doubled to 14.

It is interesting to note that even when the exponential assumption for the acknowledgment period produces a value for $\bar{k}_i$ not very different from the simulation value, the distributions of the number of messages freed have a markedly different shape. This is illustrated in Fig. 4, where we compare the distribution obtained under the exponential assumption with the one estimated from discrete-event simulation, as well as from our approach. It is apparent that our approach more correctly captures the shape of the distribution for the number of messages freed.

Since our approach to the solution of the model with a constant acknowledgment period relies on the use of the conditional probability $\Pr\{n_i > 0|k_i < N\}$ obtained from a memoryless model, we compared the latter with the values estimated in the simulation. For the parameter values investigated, the relative difference between the two quantities was typically below 5%.

In the next section, we propose an extension of our approach to the case of multiple receivers.

3. Model with multiple receivers

We now consider the system depicted in Fig. 5. The messages arriving from the source are sent simultaneously to the set of r receivers. As before, we denote by n_i the current number of messages at receiver i, and by k_i the number of messages processed by this receiver but not yet cleared. The receivers are not synchronized, and may process messages at different rates ν_i, $i = 1, \dots, r$. This takes into account heterogeneous environments, various acknowledgement strategies or different round-trip times.

At acknowledgement time (i.e., when the time window associated with the bucket expires), only messages processed by all the receivers may be freed, so that the number of messages acknowledged is given by $m = \min(k_1, \dots, k_r)$. Note that for all receivers we have $n_i + k_i = n$, where n is the current total number of messages in the system.

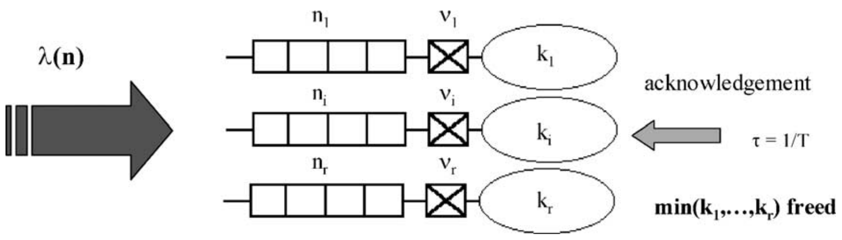


Fig. 5. Acknowledgement model with multiple receivers.

Consider receiver i, $i = 1, \dots, r$ as represented in Fig. 6. The main difference compared with the situation in Fig. 1 is that now, if there are k_i messages ready to be freed at acknowledgment time, the number of messages actually freed, denoted by m, can be 0 through k_i. Let $p_m(n_i, k_i)$ be the corresponding conditional probability when there are n_i messages in processing

$$p_m(n_i, k_i) = \Pr\{m \text{ messages freed} | n_i, k_i\}. \tag{9}$$

As before, the equations for $p(k_i, t)$ are given by (1) so that using approximations (3) and (4) we obtain again

$$p(k_i, t) = \sum_{j=0}^{k_i} C_j \frac{(\mu_i t)^{k_i - j}}{(k_i - j)!} \mathrm{e}^{-\mu_i t}, \quad k_i = 0, \dots, N-1.$$

This time, not all ready messages are freed at acknowledgement time, and we can express the initial conditions, needed to determine the constants C_j, as

$$p(k_i, t = 0) = \sum_{m=0}^{N-k_i} p(k_i + m, t = T) p_{m, k_i + m}, \quad k_i = 0, \dots, N, \tag{10}$$

where p_{m,k_i} is the conditional probability that m messages are freed given that k_i are waiting to be freed

$$p_{m,k_i} = \Pr\{m \text{ messages freed} | k_i \text{ waiting to be freed}\}, \quad m = 0, \dots, k_i. \tag{11}$$

As in the case of a single receiver, we need the conditional probability $\Pr\{n_i > 0 | k_i < N\}$ to make use of this solution. Again, we assume that solving the model for an exponentially distributed acknowledgement period will give us a reasonable approximation for the conditional probability $p(n_i|k_i)$ that there are n_i messages queued and in service at receiver i given that k_i messages are waiting to be freed.

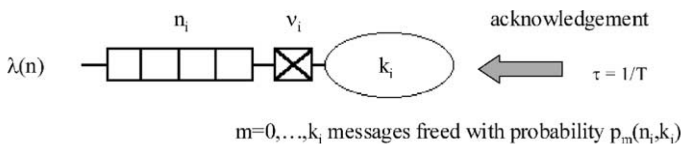


Fig. 6. Model of a receiver in the case of multiple receivers.

We need also the probability p_{m,k_i} that m messages are freed given that k_i are waiting to be freed. Assume that $p_m(n_i, k_i)$ is known. Then we can obtain p_{m,k_i} as

$$p_{m,k_i} = \sum_{n_i=0}^{N-k_i} p_m(n_i, k_i) p(n_i|k_i), \quad m = 0, \dots, k_i. \tag{12}$$

Denote by $p(k_1, \dots, k_r|n, k_i)$ the probability that there are $k_1, \dots, k_r$ messages waiting to be freed at the respective receivers given that there are k_i messages at receiver i and a total of n messages in the system. Note that knowing (n, k_i) is equivalent to knowing (n_i, k_i) so that we can express $p_m(n_i, k_i)$ as

$$\begin{aligned} &\sum_{j\neq i}\sum_{k_j=m}^{n} p(k_1, \dots, k_r|n, k_i) \quad \text{if } m = k_i, \\ &\sum_{j\neq i}\sum_{k_j=m}^{n} p(k_1, \dots, k_r|n, k_i) - \sum_{j\neq i}\sum_{k_j=m+1}^{n} p(k_1, \dots, k_r|n, k_i) \quad \text{if } m < k_i. \end{aligned} \tag{13}$$

Our solution of the model of Fig. 6 allows us to obtain $p(k_i|n)$, the probability that there are k_i messages waiting to be freed given the total number of messages, as follows:

$$p(k_i|n) \approx \frac{p(k_i, t = T) p(n_i = n - k_i|k_i)}{\sum_{k_i=0}^{n} p(k_i, t = T) p(n_i = n - k_i|k_i)}, \quad i = 1, \dots, r. \tag{14}$$

The numerator in (14) corresponds to the probability that there are k_i messages waiting to be freed and a total of n messages in the system, while the denominator expresses the probability of n messages in the system. We use $p(k_i|n)$ from (14) to approximate $p(k_1, \dots, k_r|n, k_i)$

$$p(k_1, \dots, k_r|n, k_i) \approx \prod_{j=1, j\neq i}^{r} p(k_j|n). \tag{15}$$

This allows us to consider receivers separately, accounting for the presence of other receivers through the probability p_{m,k_i} computed from (12) together with (13)–(15). Since we need p_{m,k_i} to obtain $p(k_i, t)$ from (10), and we need $p(k_i, t)$ for p_{m,k_i} in (14), we use a fixed-point iteration. We use the "trivial" distribution $p_{m,k_i} = 1$ if $m = k_i$ as the starting point. As detailed in Appendix B, we use an analogous approach to obtain the conditional probabilities $p(n_i|k_i)$ from our model under the assumption of an exponentially distributed acknowledgment period.

Note that the approximation in (15) amounts to disregarding the knowledge of the number of requests waiting to be freed in favor of the total number of messages present in the system. Despite the product of probabilities in (15), our approximation does not amount to assuming that the receivers are independent since (15) is a product of conditional probabilities. In practice, in the cases investigated, the fixed-point iterations tend to converge rapidly, and the approach produces reasonably accurate results.

Fig. 7 shows, as an example, the influence of the number of receivers r on the expected number of messages freed. The message arrival rate in Fig. 7 is $200\,\mathrm{s}^{-1}$, and the acknowledgment period is set to 20 ms. All receivers are assumed to be statistically identical, and other model parameters have the same values as in Fig. 2. Simulation results, as well as results obtained assuming exponentially distributed acknowledgment period are included for comparison.

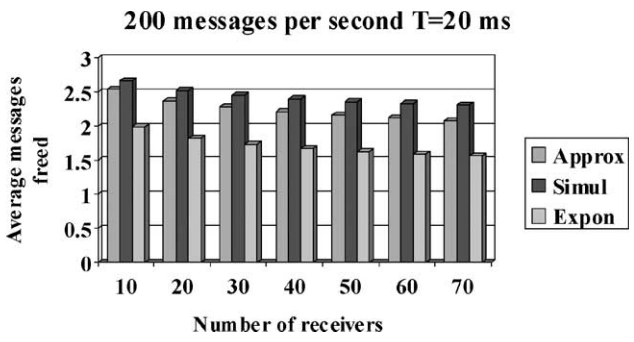


Fig. 7. Effect of the number of receivers.

We note that the relative difference between simulation results and those obtained from our approach tends to be confined to the 5–10% range. As the number of receivers grows, the relative error grows slightly. In addition, with larger number of receivers, our approach reproduces less accurately the shape of the distribution of the number of messages freed. This is illustrated in Fig. 8. Intuitively, one expects the average number of messages freed to decrease as the number of receivers increases. It is interesting to observe that, for a set of statistically identical receivers, this decrease is quite slow. This is similar to the observation in [30]. Note that a more rapid decrease can be expected in a real network due to ACK implosion.

In a recent paper, Chaintreau et al. [31] consider the throughput degradation with multiple receivers resulting from the variability of network delays. They show that, for light tailed random delays, the throughput decreases proportionally to the inverse of the logarithm of the number of receivers.

Fig. 9 shows an example of our results with two non-homogeneous receivers. The message arrival rate is 200 s^{-1} and the acknowledgement period is set to 20 ms. The mean service time of receiver 1 is kept at 5 ms while we increase the mean service time for receiver 2. With non-homogeneous receivers, our fixed-point iteration tends to converge to slightly different results in terms of the distribution of number of messages freed for each receiver class. The differences are typically only a few percent, and we report the average over receiver classes as the analytical result for the model with a constant acknowledgement period. We also show the shape of the distribution of the number of messages cleared when the mean service time of receiver 2 is 7 ms. Again, we observe that our approximation produces values within a few percent of simulation results.

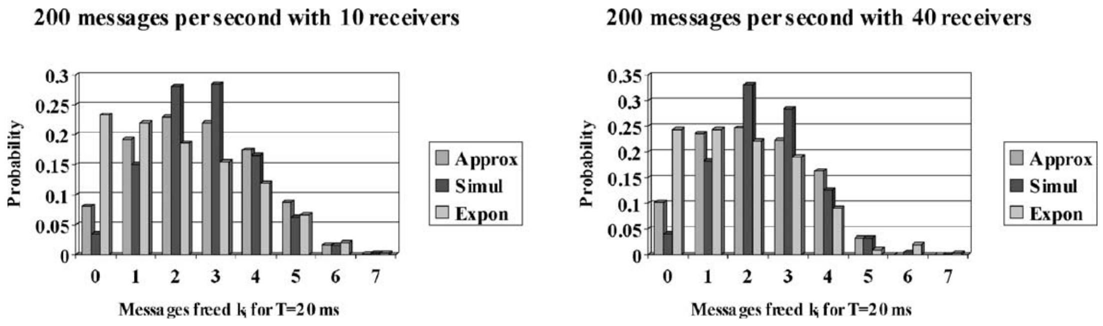
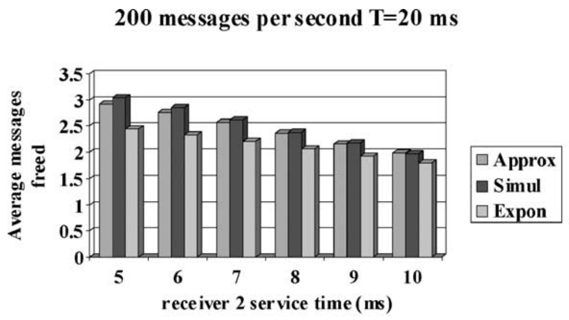


Fig. 8. Distribution of the number of messages cleared with multiple receivers.



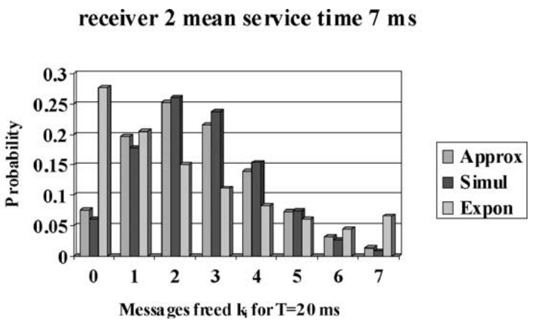


Fig. 9. Results with non-homogeneous receivers.

4. Conclusions

We have considered a model of a fork–join system with possibly non-homogeneous servers where messages processed by all the servers are periodically acknowledged and cleared. The model arises in the study of a multicast protocol. Even without the delayed clearing of processed jobs, no closed-form solution exists for the general case. We have presented a method that allows us to consider each server (receiver) separately while accounting for the presence of other receivers through partial clearing of messages at acknowledgement time. We are able to obtain a simple approximate solution for the number of messages cleared each period. The approximation uses the results of the solution of a similar model with exponentially distributed acknowledgments in the differential equations for the number of messages processed by a receiver. Comparisons with simulation results illustrate the accuracy of the proposed approach. Our analysis is based on a specific multicast protocol, and we recognize that other protocols are likely to require a different analysis and different optimization criteria. It is our hope that the approach presented can be used as a good starting point for analytical models of multicast communication.

Acknowledgements

I would like to thank Dr. Katia Obraczka of the University of California at Santa Cruz for providing several multicast references, and express my indebtedness to Professor Serge Fdida of the University of Paris, who was instrumental in revising the paper.

Appendix A. Solution of the exponential model for a single receiver

Let $x(n_i) = 1$ if $n_i > 0$ and 0 otherwise. The balance equations for the probability $p(n_i, k_i)$ that there are n_i messages to be processed at the receiver and k_i messages awaiting to be freed are given by

$$p(n_i, k_i)[\lambda(n) + \tau + x(n_i)\nu_i] = x(n_i)p(n_i - 1, k_i)\lambda(n - 1) + p(n_i + 1, k_i - 1)\nu_i,$$
$$k_i = 1, \dots, N,\ n_i = 0, \dots, N - k_i,$$
$$p(n_i, 0)[\lambda(n)+x(n_i)\nu_i] = x(n_i)p(n_i - 1, 0)\lambda(n - 1) + \sum_{k_i=1}^{N} p(n_i, k_i)\tau,\quad n_i = 0, \dots, N - 1, \quad (A.1)$$

where $n = n_i + k_i$ and $\tau = 1/T$. Consider the marginal probability $p(n_i)$ and let $\beta(n_i)$

$$\beta(n_i) = \sum_{k_i=0}^{N-n_i-1} p(k_i|n_i)\lambda(n_i + k_i), \tag{A.2}$$

$p(n_i)$ can be expressed as

$$p(n_i) = \frac{1}{G}\prod_{j=1}^{n_i}\frac{\beta(j-1)}{\nu_i}. \tag{A.3}$$

Using the fact that $p(n_i, k_i) = p(n_i)p(k_i|n_i)$ together with (A.3) we readily transform the balance equations (A.1) into equations for the conditional probability $p(k_i|n_i)$. Noting that we must have

$$\sum_{k_i=0}^{N-n_i} p(k_i|n_i) = 1 \quad \forall n_i = 0, \dots, N, \tag{A.4}$$

we can solve the equations for $p(k_i|n_i)$ through fixed-point iteration. We use the superscript j to refer to values at the jth iteration. Considering the equations in the order of increasing n_i, we have

$$p(0|0) = \frac{\tau}{\lambda(0) + \tau}, \tag{A.5}$$

$$p^j(k_i|0)[\lambda(k_i) + \tau] = p^{j-1}(k_i - 1|1)\beta^j(0), \quad k_i = 1, \dots, N, \tag{A.6}$$

$$p^j(0|n_i)[\lambda(n_i) + \nu_i] = [1 - p^j(0|n_i)]\tau + \frac{p^j(0|n_i - 1)\lambda(n_i - 1)\nu_i}{\beta^j(n_i - 1)}, \quad n_i = 1, \dots, N - 1, \tag{A.7}$$

$$p^j(k_i|n_i)[\lambda(n_i + k_i) + \tau + \nu_i] = \frac{p^j(k_i|n_i - 1)\lambda(n_i + k_i - 1)\nu_i}{\beta^j(n_i - 1)} + p^{j-1}(k_i - 1|n_i + 1)\beta^j(n_i),$$
$$n_i = 1, \dots, N - 1,\ k_i = 1, \dots, N - n_i. \tag{A.8}$$

Although we do not have a proof of convergence, in practice, the proposed iterative scheme has converged for all the examples considered within a small number (low tens) of iterations.

Appendix B. Solution of the exponential model with multiple receivers

We now consider the system of Fig. 6 under the assumption that the acknowledgment period is exponentially distributed with average time $T = 1/\tau$. The balance equations for $p(n_i, k_i)$ are given by

$$\begin{aligned}
&p(n_i,k_i)\left[\lambda(n)+x(n_i)\nu_i+t\sum_{m=1}^{k_i}p_m(n_i,k_i)\right]\\
&\quad=x(n_i)p(n_i-1,k_i)\lambda(n-1)+p(n_i+1,k_i-1)\nu_i\\
&\qquad+\tau\sum_{m=1}^{N-(n_i+k_i)}p(n_i,k_i+m)p_m(n_i,k_i+m),\quad k_i=1,\dots,N,\ n_i=0,\dots,N-k_i,\\
&p(n_i,0)[\lambda(n)+x(n_i)\nu_i]=x(n_i)p(n_i-1,0)\lambda(n-1)\\
&\qquad\qquad+\tau\sum_{m=1}^{N-n_i}p(n_i,m)p_m(n_i,m),\quad n_i=0,\dots,N-1.
\end{aligned}\tag{B.1}$$

As in the case of a single receiver considered in the preceding section, the marginal probability $p(n_i)$ is given by (A.3) with (A.2). This allows us to transform (B.1) into equations for $p(k_i|n_i)$, the conditional probability that there are k_i messages waiting to be freed given n_i. The resulting equations can be solved though an iterative scheme as follows (the superscript refers to the iteration number)

$$\begin{aligned}
&p^j(k_i|n_i)\left[\lambda(n)+\nu_i+x(k_i)\tau\sum_{m=1}^{k_i}p_m(n_i,k_i)\right]\\
&\quad=x(n_i)p^j(k_i|n_i-1)\lambda(n-1)\frac{\nu_i}{\beta^j(n_i-1)}+x(k_i)p^{j-1}(k_i-1|n_i+1)\beta^j(n_i)\\
&\qquad+\tau\sum_{m=1}^{N-(n_i+k_i)}p^j(k_i+m|n_i)p_m(n_i,k_i+m),\quad n_i=0,\dots,N-1,\ k_i=0,\dots,N-n_i.
\end{aligned}\tag{B.2}$$

We have $\sum_{k_i=0}^{N-n_i}p(k_i|n_i)=1\ \forall n_i=0,\dots,N-1$. In our iterative solution, we consider the above equations in the order $k_i=N-n_i,\dots,0$ for $n_i=0,\dots,N-1$. Having solved for $p(k_i|n_i)$ and $p(n_i)$, it is straightforward to obtain $p(k_i|n)$, the conditional probability that k_i messages are waiting to be freed given that there are n messages in the system. We then use (13) together with (15) to compute approximately $p_m(n_i,k_i)$, the probability that m messages are freed given n_i and k_i.

Since we need $p_m(n_i,k_i)$ to compute $p(k_i|n_i)$, we use a fixed-point iteration, starting with a "trivial" setting for $p_m(n_i,k_i)$, viz. $p_m(n_i,k_i)=1$ if $m=k_i$, and 0 otherwise. In all the examples considered, this fixed-point scheme has converged within a low to moderate number of iterations.

References

[1] S. Fdida, Multimedia transport protocol and multicast communication, in: W. Effelsberg, O. Spaniol, A. Danthine, D. Ferrari (Eds.), High-speed Networking for Multimedia Applications, Kluwer Academic Publishers, Boston, 1996.

[2] V. Roca, L. Costa, R. Vida, A. Dracinschi, S. Fdida, A survey of multicast technologies, in: Lecture Notes in Computer Science, vol. 2236, Cooperative Environments for Distributed System Engineering, Also in Technical Report, September 2000. http://www.lip6.fr/∼sf/publicationsWeb.htm#_2000.

[3] S. Paul, Multicasting on the Internet and its Applications, Kluwer Academic Publishers, Dordrecht, 1998.

[4] S. Floyd, V. Jacobson, S. McCanne, C. Liu, L. Zhang, A reliable multicast framework for light-weight sessions and application level framing, in: Proceedings of the ACM SIGCOM'95, August 1995.

[5] A. Koifman, S. Zabele, RAMP: a reliable adaptive multicast protocol, in: Proceedings of the IEEE INFOCOM'96, San Francisco, CA, March 1996, pp. 1442–1451.

[6] S. Paul, K.K. Sabnani, J.C. Lin, S. Bhattacharyya, Reliable multicast transport protocol (RMTP), IEEE J. Select. Areas Commun. 15 (3) (1997) 407–421.

[7] S. Amstrong, A. Frieier, K. Marzullo, Multicast transport protocol (MTP), Internet RFC1301, February 1992.

[8] L. Rizzo, A TCP-friendly single-rate multi-cast congestion control scheme, in: Proceedings of the ACM SIGCOM'2000, August 2000.

[9] A. Dracinschi, S. Fdida, Efficient congestion avoidance mechanism, in: Proceedings of the 25th IEEE Conference on Local Computer Networks (LCN), November 2000.

[10] J. Widmer, M. Handley, Extending equation-based congestion control to multicast applications, in: Proceedings of the ACM SIGCOM'2001, August 2001.

[11] H. Santoso, S. Fdida, Transport layer multicast: an enhancement for XTP bucket algorithm, in: Proceedings of the Fourth IFIP High Performance Networking (HPN'92), Liege, Belgium, December 1992.

[12] C. Huitema, The case for packet level FEC, in: Proceedings of the Protocols for High-speed Networks (Pfhsn'96), October 1996.

[13] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, Asynchronous layered coding: a massively scalable reliably content delivery protocol, October 18, 2001. INTERNET-DRAFT, draft-ietf-rmt-pi-alc-03.txt, draft-ietf-rmt-pi-alc-03.ps (expires April 2002).

[14] K. Miller, K. Robertson, A. Tweedly, M. White, Starbust multicast file transfer protocol (mftp) specification, Internet Engineering Task Force, January 1997. draft-miller-mftp-sepc-02.txt.

[15] S. Kasera, J. Kurose, D. Towsley, Scalable reliable multicast using multicast groups, in: Proceedings of the ACM SIGMETRICS, June 1997.

[16] D. DeLucia, K. Obraczka, Multicast feedback suppression using representatives, in: Proceedings of the IEEE INFOCOM'97, April 1997.

[17] R.G. Kermode, Scoped hybrid automatic repeat request with forward error correction (sharqfec), in: Proceedings of the ACM SIGCOM'98.

[18] K.L. Johnson, J.F. Carr, M.S. Day, M. Frans Kaashoek, The measured performance of content distribution networks, in: Proceedings of the International Web Caching and Content Delivery Workshop, June 2000.

[19] B.N. Levine, J. Crowcroft, C. Diot, J.J. Garcia-Luna Aceves, Consideration of receiver interest for IP multicast delivery, in: Proceedings of the IEEE INFOCOM'2000, March 2000.

[20] Y. Chawathe, S. McCanne, E.A. Brewer, RMX: reliable multicast for heterogeneous networks, in: Proceedings of the IEEE INFOCOM'2000, March 2000.

[21] P. Francis, Yoid: extending the Internet multicast architecture, Technical Report, ACIRI, April 2000. http://www.aciri.org/yoid.

[22] J.C. Lui, D. Muntz, D. Towsley, Computing performance bounds of fork–join parallel programs under a multiprocessing environment, IEEE Trans. Parallel Distribut. Syst. 9 (3) (1998) 295–311.

[23] L. Flatto, S. Hahn, Two parallel queues created by arrivals with two demands, I, SIAM J. Appl. Math. 44 (5) (1984) 1041–1053; Erratum 45 (1985) 168.

[24] L. Flatto, Two parallel queues created by arrivals with two demands, II, SIAM J. Appl. Math. 45 (5) (1985) 861–878.

[25] Y. Dallery, Z. Liu, D. Towsley, Equivalence, reversibility, symmetry and concavity properties in fork–join queueing networks with blocking, J. ACM 41 (5) (1994) 903–942.

[26] Y. Dallery, Z. Liu, D. Towsley, Properties of fork/join queueing networks with blocking under various operating mechanisms, IEEE Trans. Robot. Automat. 13 (4) (1997) 503–518.

[27] A. Thomasian, A.N. Tantawi, Approximate solutions for M/G/1 fork/join synchronization, in: Proceedings of the 1994 Winter Simulation Conference, Orlando, FL, December 1994, pp. 361–368.

[28] E. Frostig, T. Lehtonen, Stochastic comparisons for fork–join queues with exponential processing times, J. Appl. Probab. 34 (1997) 487–497.

[29] S. Balsamo, L. Donatiello, N.M. Van Dijk, Bound performance models of heterogeneous parallel processing systems, IEEE Trans. Parallel Distribut. Syst. 9 (10) (1998) 1041–1056.

[30] E. Varki, Mean value technique for closed fork–join networks, Perform. Eval. Rev. 27 (1) (1999) 103–112.

[31] A. Chaintreau, C. Diot, F. Baccelli, Impact of delay variation on multicast session performance with TCP-like congestion control, in: Proceedings of the IEEE INFOCOM’2001, April 2001.

Alexandre Brandwajn holds an Ingénieur Civil des Télécommunications degree from the Ecole Nationale Supérieure des Télécommunications in Paris, and a Docteur d’Etat degree in Computer Science from the University of Paris VI. He worked as researcher at the Institut de Recherche en Informatique et Automatique (IRIA), France, then he was on the Faculty of the Ecole Nationale Supérieure des Télécommunications in Paris, where he directed a project in adaptive computer architecture. In 1979, he joined Amdahl Corporation in Sunnyvale, California, where he was a Senior Computer Architect, and later Manager of Systems Analysis group. Since 1985, he is a Professor of Computer Engineering at the University of California at Santa Cruz. His current research interest include efficient solution of systems with large state space, multimedia/networking problems, and applications of priority systems with finite capacity.