Performability Modeling of Messaging Services in Distributed Systems[*]

Srinivasan Ramani, Katerina Goševa-Popstojanova, Kishor S. Trivedi
Center for Adv. Computing and Communication, Department of ECE,
Duke University, Durham, NC 27708-0291, USA
{sramani, katerina, kst}@ee.duke.edu

Abstract

Messaging services in distributed systems act as an intermediary between suppliers and consumers, relieving the burden on the supplier. Detailed performance models for messaging services have been developed only recently. However, these models do not include the effect of failures. In this paper we consider the expected loss rate associated with messaging services as a performability measure and derive approximate closed-form expressions for three different quality of service settings.

1 Introduction

Messaging services are useful components in distributed systems that require scalable dissemination of messages (or events) from suppliers to consumers. Typically, they can be configured to provide point-to-point reliable messaging or publish-subscribe paradigms. Examples include the CORBA Notification Service and the Java Messaging Service (JMS). These services act as an intermediary between suppliers and consumers and take care of client registration and message propagation, relieving the burden on the supplier. Thus, the supplier and consumer are not tied up in a client-server type of interaction, but rather are "decoupled". Literature exists on the performance analysis of client-server [4] and producer-consumer [1] systems. Performability [5] has also been evaluated for client-server systems [2, 7]. Recently performance models for the configurable delivery and discard policies found in messaging services have been developed [9]. However, these models do not include the effect of failures. In a distributed system, supplier, consumer, and messaging services can fail independently leading to different consequences; thus, the effect of partial failures needs to be analyzed as well. In this paper we consider the expected loss rate associated with messaging services as a performability measure and derive approximate closed-form expressions for three different quality of service settings - "best effort", "persistent connections", and "persistent connections and messages" [6].

2 Approximate closed-form solutions

In practical situations failure rates of the supplier, messaging service and consumer are very low in comparison with the repair rates and especially in comparison with the message

[*] This work was supported as a core project in the Center for Advanced Computing and Communication, and by Telcordia Technologies.

arrival and delivery rates. This observation allows us to use a hierarchical approach. Using a hierarchical approach gives the modeling framework flexibility and helps us derive approximate closed-form solutions to complex composite models. Since the failures are rare, the queue within the messaging service can be approximated very well by its steady state queue length distribution. Let $\bar{N} = \sum_{i=0}^{k} i * p_i$ denote the expected number of messages in the queue at steady state, where p_i is the steady state probability that the queue has i messages, and p_k is the probability of the queue being full [10]. The low failure rates also allow us to ignore multiple failures.

The hierarchical approach makes use of the availabilities of the supplier, messaging service, and the consumer denoted by A_S, A_M, and A_C respectively, which have to be evaluated from detailed lower-level availability models. When steady state availability measures are required, it is useful to reduce the models to two-state availability models [10] and derive equivalent failure (γ_{eq}) and repair rates (τ_{eq}). This reduction simplifies the analyses.

The expected loss rates can now be written down for the three quality of service settings as follows:

- Best effort (BE)

$$\begin{aligned} LR_{BE} &= \text{loss rate when the messaging service crashes} \\ &\quad + \text{loss rate when messaging service is down} \\ &\quad + \text{loss rate when the consumer crashes} \\ &\quad + \text{loss rate when the consumer is down } + \text{ loss rate due to discards} \\ &= A_S A_M A_C \gamma_{eq}^{(M)} \bar{N} + A_S(1 - A_M) A_C \lambda + A_S A_M A_C \gamma_{eq}^{(C)} \bar{N} \\ &\quad + A_S A_M (1 - A_C) \lambda + A_S A_M A_C \lambda p_k. \end{aligned} \tag{1}$$

- Persistent connections (PC)

$$\begin{aligned} LR_{PC} &= \text{loss rate when the messaging service crashes} \\ &\quad + \text{loss rate when messaging service is down} \\ &\quad + \text{loss rate when the consumer is down} + \text{ loss rate due to discards} \\ &= A_S A_M A_C \gamma_{eq}^{(M)} \bar{N} + A_S(1 - A_M) A_C \lambda \\ &\quad + A_S A_M (1 - A_C) \lambda \sum_{i=0}^{k} \left(\frac{\lambda}{\lambda + \tau_{eq}^{(C)}} \right)^{k-i} p_i + A_S A_M A_C \lambda p_k. \end{aligned} \tag{2}$$

- Persistent connections and messages (PCM)

$$\begin{aligned} LR_{PCM} &= \text{loss rate when messaging service is down} \\ &\quad + \text{loss rate when the consumer is down} + \text{ loss rate due to discards} \\ &= A_S(1 - A_M) A_C \lambda + A_S A_M (1 - A_C) \lambda \sum_{i=0}^{k} \left(\frac{\lambda}{\lambda + \tau_{eq}^{(C)}} \right)^{k-i} p_i \\ &\quad + A_S A_M A_C \lambda p_k. \end{aligned} \tag{3}$$

We have considered that when the messaging service is down, since the supplier cannot pass on messages to the messaging service, the loss rate is λ. With the BE quality of service setting, the messaging service will not queue messages if the consumer is down. Hence the loss rate is again λ. However, if the consumer is down and the PC or PCM quality of service

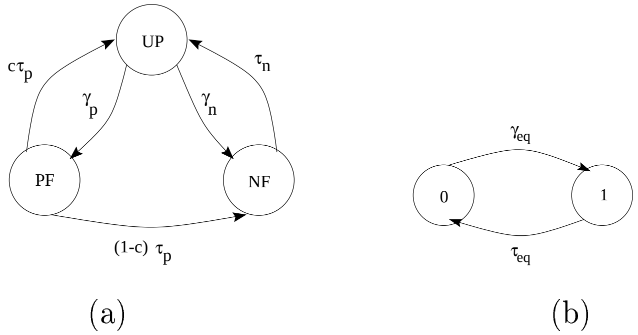


Figure 1: CTMC model of failure behavior - (a) Full model, (b) 2-state equivalent

setting is used, the messaging service continues to queue messages for the consumer. In this case messages are lost only if the queue fills up before the consumer recovers. This happens with the probability $\left(\frac{\lambda}{\lambda+\tau_{eq}^{(C)}}\right)^{k-i}$, given that the queue had i messages when the consumer failed. When the supplier, messaging service, and consumer are all up, with rate $\gamma_{eq}^{(M)}$ the messaging service can fail, leading to the loss of the entire queue ($\bar{N}$ messages on average) when BE or PC quality of service setting is used. In addition, with the BE quality of service setting, the entire queue is also lost when the consumer fails (at rate $\gamma_{eq}^{(C)}$). In all three quality of service settings, messages are also lost when an incoming message sees a full queue (with probability p_k).

3 Numerical illustration

To illustrate our approach, let us consider that the supplier, messaging service, and consumer are running on seperate nodes and that their availabilities are modeled by the continuous time Markov chain (CTMC) shown in Figure 1(a). As before, we should use the superscript (S), (M), and (C) when referring to the rates of the supplier, messaging service, and consumer, respectively. The availability model in Figure 1(a) considers two types of failures; the process crashes with rate γ_p and the node hosting the process crashes with rate γ_n. If the process crashes, an attempt is made to restart the process. Restart completes at rate τ_p and succeeds with probability c. If the restart is unsuccessful (with probability $1-c$), node repair that completes with rate τ_n is attempted. The steady state probabilities can be derived as:

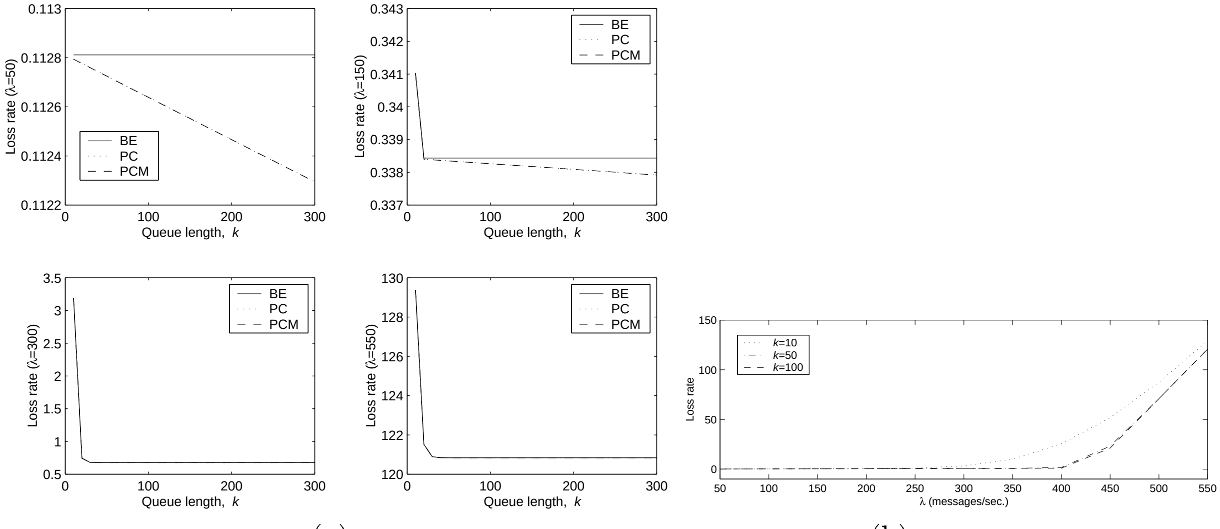
$$\pi_{UP} = \left[1 + \frac{\gamma_p}{\tau_p} + \frac{1}{\tau_n}(\gamma_n + (1-c)\gamma_p)\right]^{-1}, \quad \pi_{PF} = \frac{\gamma_p}{\tau_p}\pi_{UP}, \quad \pi_{NF} = \frac{1}{\tau_n}(\gamma_n + (1-c)\gamma_p)\pi_{UP}. \tag{4}$$

The availability is given by $A = \pi_{UP}$.

The two-state equivalent availability model for Figure 1(a) is shown in Figure 1(b). The equivalent failure and repair rates, γ_{eq} and τ_{eq} and hence availability A can be derived as [10]:

$$\gamma_{eq} = \frac{\pi_{UP}(\gamma_p + \gamma_n)}{\pi_{UP}} = \gamma_p + \gamma_n, \qquad \tau_{eq} = \frac{\pi_{PF} c\tau_p + \pi_{NF}\tau_n}{\pi_{PF} + \pi_{NF}}, \qquad A = \frac{\tau_{eq}}{\gamma_{eq} + \tau_{eq}}. \tag{5}$$

For illustration we take the failure and repair rates to be the same for the supplier, messaging service and the consumer. We assume that on the average processes fail once in 10 days, and nodes fail once in 20 days. Average process restart time is 1 minute and node repair



(a) (b)

Figure 2: Effect of varying - (a) queue length k, (b) input rate λ

time is 30 minutes. The probability $c = 0.99$. This gives $\gamma_{eq} = 1.7361e-06, \tau_{eq} = 0.0015$, and $A = 0.9988$.

Figure 2(a) shows the effect of varying the queue length k for different input rates, taking the delivery rate to the consumer, μ to be 430 messages/sec. In all plots the loss rates for PC and PCM quality of service settings are practically identical. This is because the relative contribution of the term $A_S A_M A_C \gamma_{eq}^{(M)} \bar{N}$ to the loss rate is negligible. With a low input rate ($\lambda = 50$), the probability of the queue being full at steady state is very small. Hence the main reason for loss in this case is when the consumer is down. With PC and PCM quality of service settings, as the queue size is increased the probability of the queue being able to absorb incoming messages when the consumer is unavailable increases. Hence the loss rates for PC and PCM are lower than for the BE quality of service setting (although not very significantly). As the input rate is increased, the losses due to a full queue start to dominate ($\lambda = 150$). Eventually for high input rates ($\lambda = 300, 550$) the losses are almost entirely due to discards when messages arrive at a full queue, and all three quality of service settings have the same loss rate. It is also interesting to note that for a given consumer delivery rate (430 messages/sec. in this case) it is not useful to increase the queue size beyond a point (around 25 in this case) since the probability of a queue being full does not drop appreciably.

Figure 2(b) shows the effect of input rate for different queue sizes. Only the BE loss rate is shown; the other two are almost identical. It can be seen that when the queue size is larger than the optimal value (around 25 in this case), the loss rate starts to shoot up at around $\lambda = 400$ messages/sec. irrespective of the queue size. But when the queue size is less than 25, the input rate should be lower to avoid a spurt in loss rate. This optimal queue size, of course, also depends on the distribution (and hence variance) of message arrival and delivery times. Markov Modulated Poisson Process (MMPP) can be used to model variances higher than and r-stage Erlang approximations to model variances lower than those associated with exponential distributions [8].

4 Conclusions

We have provided closed-form expressions for the expected loss rates associated with different quality of service settings in messaging services. We have also provided a framework to incorporate details such as application software and node failures into the availability models of the components involved, without affecting the upper level closed-form loss rate equations. The approach thus also illustrates an extension of software architecture-based evaluation techniques [3] to the analysis of distributed systems.

References

[1] G. Ciardo, J. K. Muppala, and K. S. Trivedi. Analyzing concurrent and fault-tolerant software using stochastic reward nets. *Journal of Parallel and Distributed Computing*, 15:255–269, 1992.

[2] S. Garg, Y. Huang, C. M. R. Kintala, K. S. Trivedi, and S. Yajnik. Performance and reliability evaluation of passive replication schemes in application level fault tolerance. In *Proc. of the 29th International Symposium on Fault-Tolerant Computing (FTCS '99)*, pages 322–329, Madison, WI, USA, June 15-18 1999.

[3] K. Goševa-Popstojanova and K. S. Trivedi. Architecture based approach to reliability assessment of software systems. *Performance Evaluation*, 45(2-3):179–204, 2001.

[4] O. C. Ibe, H. C. Choi, and K. S. Trivedi. Performance evaluation of client-server systems. *IEEE Transactions on Parallel and Distributed Systems*, 4(11):1217–1229, November 1993.

[5] J. F. Meyer. On evaluating the performability of degradable computer systems. *IEEE Transactions on Computers*, 29(8):720–731, 1980.

[6] Object Management Group. *Notification Service Specification*, June 2000. Version 1.0, *ftp://ftp.omg.org/pub/docs/formal/00-06-20.pdf.*

[7] A. Puliafito, S. Riccobene, and M. Scarpa. Modelling of client-server systems. In *Proc. of the Third International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '95)*, pages 340–344, Durham, NC, USA, January 18-20 1995.

[8] S. Ramani, K. S. Trivedi, and B. Dasarathy. Performance analysis of the CORBA Event service using stochastic reward nets. In *Proc. of 19th IEEE Symposium on Reliable Distributed Systems (SRDS '00)*, pages 238–247, Nürnberg, Germany, October 16-18 2000.

[9] S. Ramani, K. S. Trivedi, and B. Dasarathy. Performance analysis of the CORBA Notification service. In *Proc. of 20th IEEE Symposium on Reliable Distributed Systems (SRDS '01)*, New Orleans, USA, October 28-31 2001. (to appear).

[10] K. S. Trivedi. *Probability & Statistics with Reliability, Queuing and Computer Science Applications.* John Wiley & Sons, second edition, 2001. (to appear).