Fast Evaluation of Ensemble Transients of M(t)/M(t)/· Networks *

Catalin T. Popescu *A. Udaya Shankar*
cpopescu@cs.umd.edu shankar@cs.umd.edu
Department of Computer Science
University of Maryland, College Park

December 14, 1999

Abstract

We develop a numerical method that approximates the transient solution of time-dependent $M(t)/M(t)/1/\infty$ and M(t)/M(t)/1/K queuing networks. The method generates a set of coupled differential equations, one for each queue in the network. Numerical instability arises under certain conditions, e.g., large bandwidths and buffers, and we present techniques to overcome this problem. We also show how to extend the method to handle TCP/IP networks.

1 Introduction

A large set of performance evaluation and modeling problems can be reduced to the problem of solving a queuing system. Often the steady state solution is not enough: for example, when input rates are time-varying, or if convergence time, maximum transitory value, or some other time-dependent metric is required. In such cases transient evaluation becomes necessary. But for this an adequate technique does not currently exist. Numerical solutions are unmanageable for realistic systems, because the state

*This work is supported partially by ARPA contract number DABT6396C0075 and DoD contract number MDA90497C3015 to the University of Maryland. It should not be interpreted as representing the opinions or views of ARPA, DoD, or the U.S. Government.

space becomes enormous. Discrete-event simulations are too slow for systems with large rates and large buffers, due to the high frequency of simulation events. Flow-level simulation is faster but its accuracy is questionable (because it makes the unrealistic assumption that the interval between successive changes in the network-wide flow pattern is large enough so that steady state holds for most of the interval).

The transient solution of an $M/M/1/\infty$ queue is an ugly expression [7]. For a $M(t)/M(t)/\cdot$ queue, in general case, obtaining the instantaneous probability measures is analytically intractable [9], and numerically very expensive due to the large state space.

The idea of approximating the relation between the transient metrics with the relation between the steady state metrics was introduced in [2]. However, the solution presented there is practically applicable only to single $M/M/1/\infty$ queues. For networks of queues the method would fail due to the huge number of differential equations to be solved ($K_1 \cdot K_2 \cdot ... \cdot K_n$, where K_i is the total buffer space of queue i). According to our knowledge, at this moment, there is no practical solution to generate or properly approximate transient solutions for networks of $M(t)/M(t)/\cdot$. Our method generates a small set of coupled differential equations: one per each queue in the system.

In this paper we present a numerical method, called the Z-iteration, that approximates the transient metrics of $M(t)/M(t)/1/\infty$ and M(t)/M(t)/1/K queues and of open networks created by interconnecting such queues. We also show how to extend this to TCP/IP networks, i.e., datagram networks with end-to-end congestion control.

Z-iteration yields the time evolution of ensemble metrics (e.g. instantaneous queue size distribution) at a cost several orders cheaper than simulation. The first version of the Z-iteration [4, 8] handled *single* multi-class multi-resource queues, adequate for analyzing connection-oriented networks with strict access control and resource reservation. In this paper, we develop a more flexible formulation that also handles networks of time-dependent queues, appropriate for modeling classical datagram networks.

The rest of the paper is organized as follows. Section 2 describes the Z-iteration for single $M(t)/M(t)/\cdot$ queues. Section 3 describes the Z-iteration for for networks of $M(t)/M(t)/\cdot$ queues. Section 4 examines numerical stability issues. Section 5 briefly describes the Z-iteration for TCP/IP networks. We conclude in Section 6.

2 Z-iteration for Single $M(t)/M(t)/\cdot$ Queue

The Z-iteration is an efficient numerical approximation method that computes instantaneous ensemble metrics of time-dependent queuing systems. It is based on functional approximations of relationships between instantaneous metrics by the corresponding steady-state relationships. These approximations allow the evolution of the metrics to be defined by a small number of differential equations, rather than the large number of Chapmann-Kolmogorov equations (which are as many as the maximum queue size).

Table 1 gives the notation we use for a queue. Instantaneous parameters refer to a queue with time-varying arrival and service rates. Steady-state parameters refer to a queue in steady state, with constant arrival and service rates.

$\lambda(t)$	instantaneous arrival rate at time t
$\mu(t)$	instantaneous service rate at time t
$N(t)$	instantantaneous average queue size at time t
$B(t)$	instantaneous blocking probability at time t
$U(t)$	instantaneous utilization at time t
$z(t)$	instantaneous virtual traffic intensity at time t $(\neq \lambda(t)/\mu(t))$
λ	steady-state arrival rate
μ	steady-state service rate
ρ	steady-state traffic intensity $(= \lambda/\mu)$
N	steady-state average number of customers
B	steady-state blocking probability
U	steady-state utilization
$F_{\mathrm{N}}(\cdot)$	function yielding N in terms of ρ
$F_{\mathrm{B}}(\cdot)$	function yielding B in terms of ρ
$F_{\mathrm{U}}(\cdot)$	function yielding U in terms of ρ

Table 1: Notation

Consider a single queue, either M(t)/M(t)/1/∞ or M(t)/M(t)/1/K. We first summarize the **old version** of the Z-iteration [4, 8]. The starting point was the following flow equation, obtainable from the Chapmann-Kolmogorov equations:

$$\frac{dN(t)}{dt} = \lambda(t)[1 - B(t)] - \mu(t)U(t) \tag{1}$$

The idea was to express $B(t)$ and $U(t)$ in terms of $N(t)$, thereby transforming equation (1) into a single scalar differential equation for $N(t)$. It turns out that the relationship between $B(t)$ and $N(t)$ is very well approximated by the relationship between steady-state B and steady-state N. The same holds for the relationship between $U(t)$ and $N(t)$.

Thus we want functions expressing B and U in terms of N. What is available, however, are functions expressing N, B, and U in terms of the steady-state traffic intensity

ρ $(= \lambda/\mu)$. We denote these functions by $F_{\mathrm{N}}(\cdot)$, $F_{\mathrm{B}}(\cdot)$, and $F_{\mathrm{U}}(\cdot)$, respectively. For example, for a M/M/1/∞ queue we have [6, 7]:

$$
\begin{aligned}
N &= F_{\mathrm{N}}(\rho) = \frac{\rho}{1-\rho} \\
B &= F_{\mathrm{B}}(\rho) = 0 \\
U &= F_{\mathrm{U}}(\rho) = \rho
\end{aligned}
\tag{2}
$$

For a M/M/1/K queue, we have [6, 7]:

$$
\begin{aligned}
N &= F_{\mathrm{N}}(\rho) = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{(K+1)}}{1-\rho^{(K+1)}} \\
B &= F_{\mathrm{B}}(\rho) = \frac{1-\rho}{1-\rho^{(K+1)}}\, \rho^{K} \\
U &= F_{\mathrm{U}}(\rho) = \frac{1-\rho^{K}}{1-\rho^{(K+1)}}\, \rho
\end{aligned}
\tag{3}
$$

For M/M/1/∞, we can invert $F_{\mathrm{N}}(\rho)$ and so obtain B and U in terms of N, specifically, $B = 0$ and $U = N/(N+1)$. But in general, including the case of blocking queues, we cannot invert $F_{\mathrm{N}}(\rho)$ analytically. So instead the inversion was done numerically, using another approximation as follows:

- U is computed from N *assuming a non-blocking system.*
- B is computed as the fixed point of $B = F_{\mathrm{B}}(\rho)$ and $\rho = U/(1-B)$ (obtained by equating the inflow $\lambda(1-B)$ to the outflow μU). The resulting value of ρ is simply the steady-state traffic intensity value consistent with B and N.

This approach works very well for M(t)/M(t)/1/∞ and M(t)/M(t)/K/K queues, and for M(t)/M(t)/1/K queues when $\lambda(t) < \mu(t)$. But it does not work for networks of these queues or for M(t)/M(t)/1/K queues when $\lambda(t) > \mu(t)$.

We now present the **new version** of the Z-iteration. This version eliminates the non-blocking assumption used in the numerical inversion operation above, and works for open networks of M(t)/M(t)/1/K and M(t)/M(t)/1/∞ queues.

As mentioned above, formulas for N, B and U are usually in terms of ρ. This suggests that we introduce an instantaneous version of ρ, which we refer to as the *instantaneous virtual traffic intensity*, denoted by $z(t)$, and develop a differential equation for $z(t)$ rather than for $N(t)$. Then $N(t)$, $U(t)$, and $B(t)$ can be approximated by

$$N(t) = F_{\mathrm{N}}(z(t))$$

$$B(t) = F_{\mathrm{B}}(z(t))$$
$$U(t) = F_{\mathrm{U}}(z(t)) \tag{4}$$

Although $z(t)$ is fictitious, it has a natural interpretation: at any time t, it is the amount of traffic intensity that if applied constantly would result in steady-state N, B, and U equal to $N(t)$, $B(t)$, and $U(t)$, respectively. In fact, $z(t)$ is just a more accurate version of the iterate ρ that appears in the numerical fixed-point inversion in the old version. *Note that $z(t)$ is not equal to $\lambda(t)/\mu(t)$.*

To obtain a differential equation for $z(t)$, we start with the differential equation for $N(t)$

$$\frac{dN(t)}{dt} = \lambda(t)[1 - B(t)] - \mu(t)\ U(t);$$

Replacing $dN(t)/dt$ by $(dN(t)/dz(t))(dz(t)/dt)$, $dN(t)/dz(t)$ by $dF_{\mathrm{N}}(z)/dz$, $B(t)$ by $F_{\mathrm{B}}(z(t))$, and $U(t)$ by $F_{\mathrm{U}}(z(t))$, we obtain

$$\frac{dz(t)}{dt} = \frac{1}{dF_{\mathrm{N}}(z)/dz}[\lambda(t)(1 - F_{\mathrm{B}}(z(t))) - \mu(t)F_{\mathrm{U}}(z(t))] \tag{5}$$

Thus we have a scalar differential equation whose solution yields the evolution of $z(t)$. Plugging $z(t)$ into equation (4) yields evolutions of $N(t)$, $U(t)$, and $B(t)$.

Equation (5) can be instantiated for any type of M(t)/M(t)/$\cdot$ queue. For a M(t)/M(t)/1/∞ queue, we obtain

$$\frac{dz(t)}{dt} = (1 - z(t))^2(\lambda(t) - \mu(t)z(t)) \tag{6}$$

For a M(t)/M(t)/1/K queue, we obtain

$$\frac{dz(t)}{dt} = \frac{(1 - z(t)^{(K+1)})(1 - z(t)^K)(1 - z(t))^2}{(1 - z(t)^{(K+1)})^2 - (K+1)^2 z(t)^K (1 - z(t))^2}(\lambda(t) - \mu(t)z(t)) \tag{7}$$

Example: Consider a M/M/1/K queue with $\lambda = 1.5$, $\mu = 2.0$, $K = 7$, and initially empty. Figure 1 shows plots of $N(t)$ and $U(t)$ obtained from the Z-iteration and also obtained from solving the Chapman-Kolmogorov equations. Figure 2 is the same for $\lambda = 2.0$, $\mu = 1.5$ and $K = 10$.

3 Z-iteration for M(t)/M(t)/$\cdot$ Networks

We extend the Z-iteration to networks of M(t)/M(t)/1/∞ and M(t)/M(t)/1/K queues. Here, a departure from queue i is routed to queue j with a time-dependent probability $r_{ij}(t)$, and leaves the network with probability $1 - \sum_j r_{ij}(t)$. The arrivals to queue i consist of external arrivals $\lambda_i(t)$ (from outside the network) and departures from queues

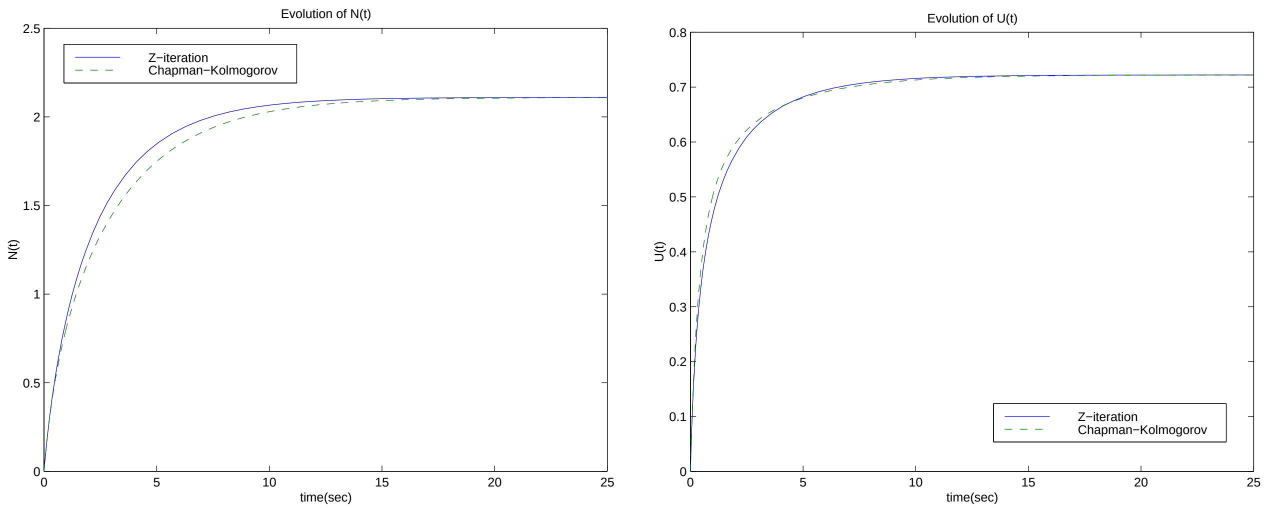

Figure 1: Results for M/M/1/K queue ($\lambda = 1.5$, $\mu = 2.0$, $K = 7$)

in the network routed to queue i. The total arrival rate of queue i, denoted $\lambda_i^*(t)$, is given by

$$\lambda_i^*(t) = \lambda_i(t) + \sum_{j=1}^{n} r_{ji}(t)\ \mu_j(t)\ U_j(t). \tag{8}$$

Any arriving customer can be blocked, except, of course, customers feeding back from queue i. The differential equation for $z_i(t)$, the instantaneous virtual traffic intensity of queue i, is obtained by appropriately modifying equation (6) or (7). If queue i is a M/M/1/∞ queue, we have

$$\frac{dz_i(t)}{dt} = (1 - z_i(t))^2[\lambda_i(t) + \sum_{j=1;j\neq i}^{n} r_{ji}(t)\mu_j(t)F_{\mathrm{U}_j}(z_j(t)) - \mu_i(t)(1 - r_{ii}(t))z_i(t)] \tag{9}$$

If queue i is a M/M/1/K queue, we have

$$\begin{aligned} \frac{dz_i}{dt} &= \frac{(1 - z_i^{(K_i+1)})(1 - z_i^{K_i})(1 - z_i)^2}{(1 - z_i^{(K_i+1)})^2 - (K_i + 1)^2 z_i^{K_i}(1 - z_i)^2} \\ &\quad \times [\lambda_i(t) + (\sum_{j=1;j\neq i}^{n} r_{ji}\mu_j(t)F_{\mathrm{U}_j}(z_j)) - \mu_i(t)(1 - r_{ii})z_i] \end{aligned} \tag{10}$$

Examples: We present three examples. The first example is the small network shown in Figure 3(a). The arrival rates are varied as shown in part (b) of the figure. The evolution of $N_1(t)$ and $N_2(t)$ are plotted in Figure 3(c) and (d).

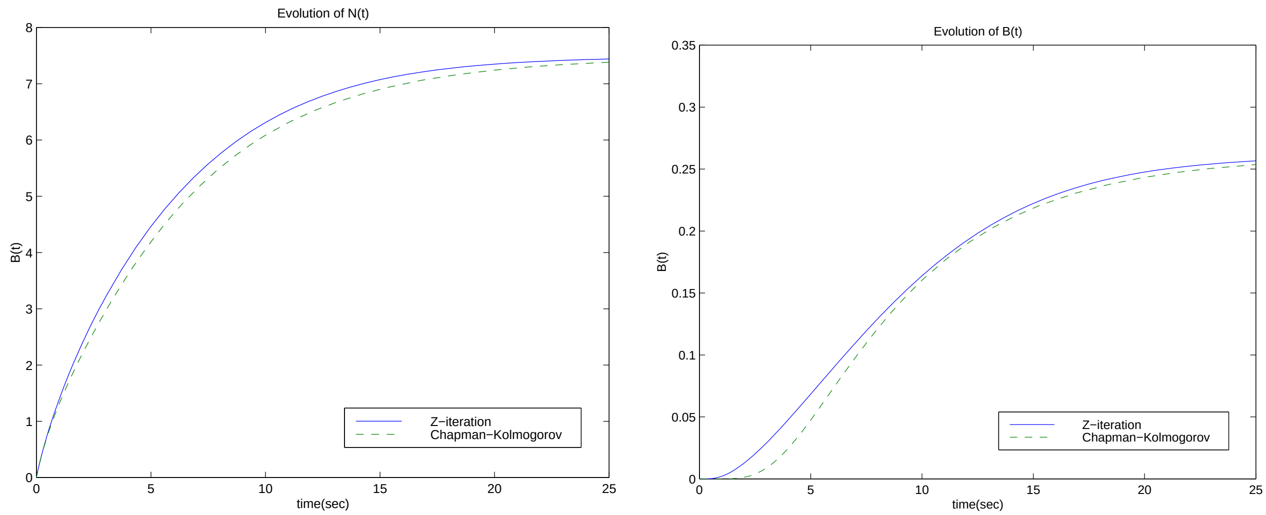

Figure 2: Results for a M/M/1/K queue ($\lambda = 2.0$, $\mu = 1.5$, $K = 10$)

The second example is a tandem network with feedback, presented in Figure 4. The corresponding plots are in Figure 5. We compare our method against the average obtained after 10,000 runs by the simulator.

The third example is a larger queueing network, presented in Figure 6. Some plots are presented in Figure 7. The simulator spent 160 seconds for averaging 30,000 runs, while the Z-iteration equations were solved in less than one second by Matlab. This discrepancy increases for higher input rates and/or smaller service times, because this increase the number of events to be processed but has no effect on the equations to be solved.

From experimentation we find that the method works well for open queuing networks but fails for closed queuing networks. To work well for closed queuing networks, we need the standard normalization constant (for computing B, U and N).

4 Numerical Issues

For M(t)/M(t)/1/K queues the $z(t)$ differential equation becomes *extremely stiff* for values of K larger than 10^3. This gives rise to problems regarding numerical stability and convergence. Applying regular solution methods to the differential equations stated above will produce an incorrect answer or no answer at all. Applying stiff equation solvers will produce a correct answer but *extremely slowly* (for a good coverage of differential equation solvers, see [5]).

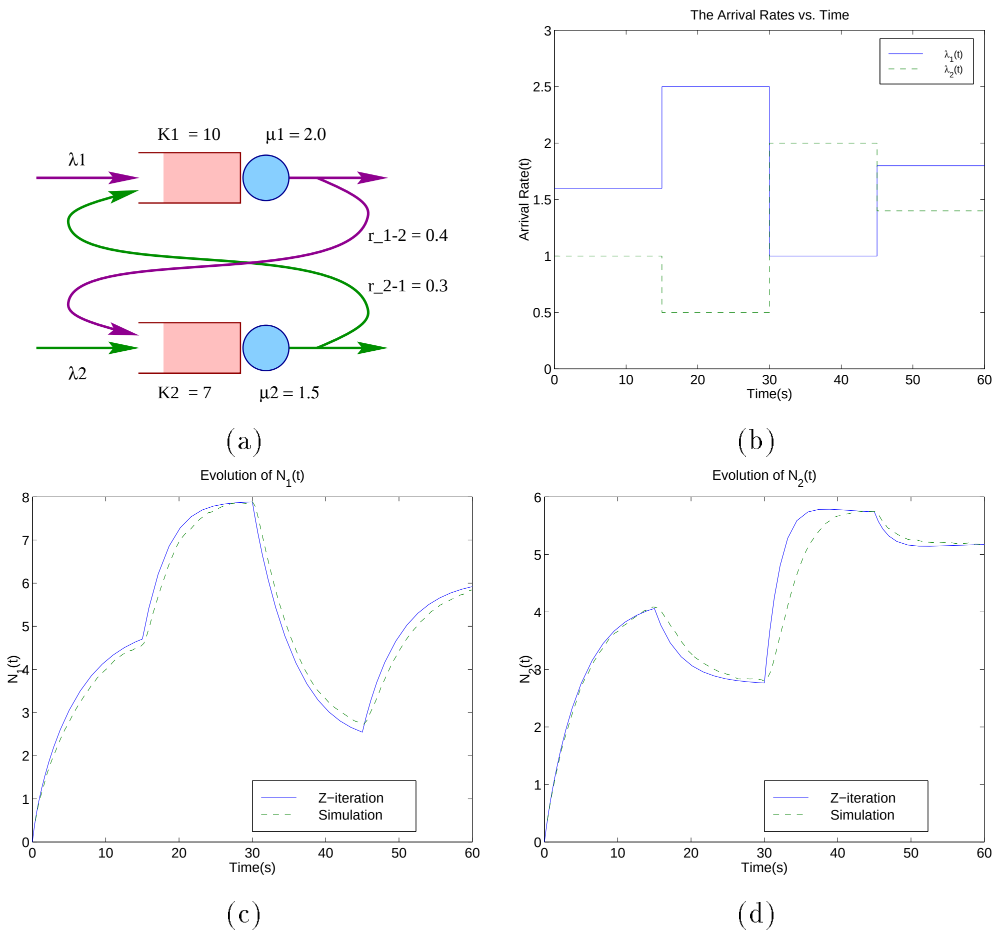

Figure 3: A simple network and its evolutions.

In this section, we describe how to overcome these problems for networks of M(t)/M(t)/1/K queues. Consider the differential equation (10). Observe that for high K ($\approx 10^3$), we have $z^K \approx 0$ for $z < 1 - \epsilon$ and $z^K \approx \infty$ for $z > 1 + \epsilon$, for some $\epsilon > 0$. Using this approximation in equation (10) and simplifying, we get

$$\frac{dz_i}{dt} = \begin{cases} (1-z_i)^2[\ \lambda_i(t) + (\ \sum_{j=1;j\neq i}^{n} r_{ji}\mu_j \min(z_j,1)) - \mu_i(t)(1-r_{ii})z_i\ ] & \text{for } z_i < 1-\epsilon_i \\ ((1-z_i)^2/z_i)[\ \lambda_i(t) + (\sum_{j=1;j\neq i}^{n} r_{ji}\mu_j \min(z_j,1)) - \mu_i(t)(1-r_{ii})z_i\ ] & \text{for } z_i > 1+\epsilon_i \\ \text{as in equation (10)} & \text{otherwise} \end{cases} \tag{11}$$

The formulas for N_i, B_i and U_i are similarly modified. For example, for N_i we have:

$$N_i(z_i) = \begin{cases} (z_i/1 - z_i) & \text{for } z_i < 1-\epsilon_i \\ K_i + (z_i/1 - z_i) & \text{for } z_i > 1+\epsilon_i \\ \text{as in equation (3)} & \text{otherwise} \end{cases} \tag{12}$$

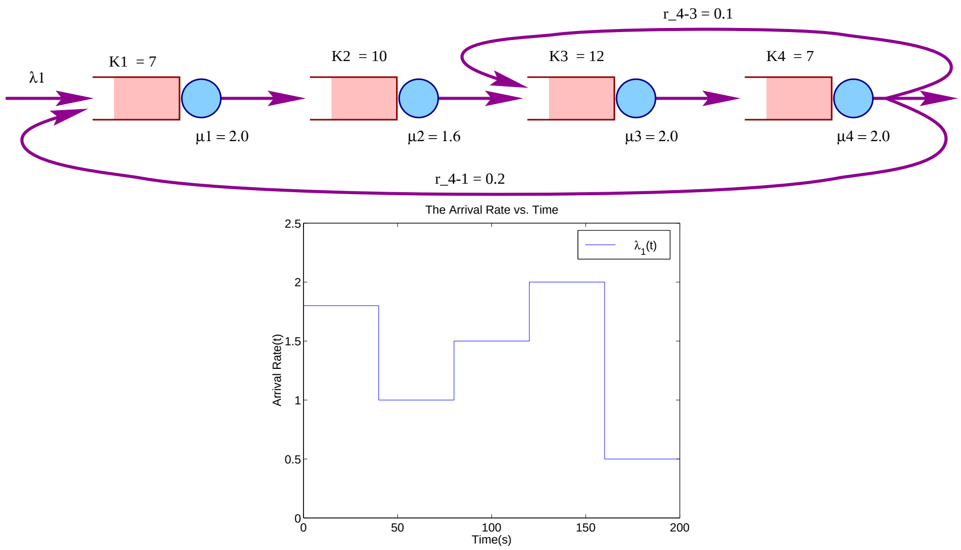

Figure 4: A tandem network with feedback

These modifications are not sufficient. We need to stop using equation (10) around $z_i = 1.00$, the point of instability. To do so, we find an $\epsilon_i > 0$ such that $N(1-\epsilon_i) = N(1+\epsilon_i)$. That is,

$$K_i + \frac{1+\epsilon_i}{1-(1+\epsilon_i)} = \frac{1-\epsilon_i}{1-(1-\epsilon_i)} \tag{13}$$

which yields $\epsilon_i = 2/K_i$. So we make the computation jump over the interval $[1-\epsilon_i,\ 1+\epsilon_i]$ as follows: whenever $1-2/K_i < z_i < 1$ would hold, we set $z_i = 1+2/K_i$; whenever $1 < z_i < 1+2/K_i$ would hold, we set $z_i = 1-2/K_i$. Outside this interval, we continue to use the first two lines of equation (11).

This is still not good enough. For large $\lambda_i^*(t)$ and $\mu_i(t)$, we get a lot of large fluctuations when $z_i(t)$ comes close to the value $\lambda_i^*(t)/\mu_i(t)$. This is because $dz_i(t)/dt$ becomes highly negative (positive) when $z_i(t)$ is slightly higher (lower) than $\lambda_i^*(t)/\mu_i(t)$. To overcome this problem, we exploit the following monotonicity property in the evolution of $z_i(t)$:

> At any moment t, $z_i(t)$ tends to evolve monotonically (increasing or decreasing) towards $\lambda_i^*(t)/\mu_i(t)$ from its current value.

We use this property by allowing a change in the sign of $dz_i(t)/dt$ to occur, during a computational step of the solver ($[t-\delta t, t]$), only if the current value of $\lambda_i^*(t)/\mu_i(t)$ has been changed from the previous step such that

$$\frac{z_i(t-\delta t)}{\delta t} \in \left[\min\left(\frac{\lambda_i^*(t)}{\mu_i(t)}, \frac{\lambda_i^*(t-\delta t)}{\mu_i(t-\delta t)}\right),\ \max\left(\frac{\lambda_i^*(t)}{\mu_i(t)}, \frac{\lambda_i^*(t-\delta t)}{\mu_i(t-\delta t)}\right)\right] \tag{14}$$

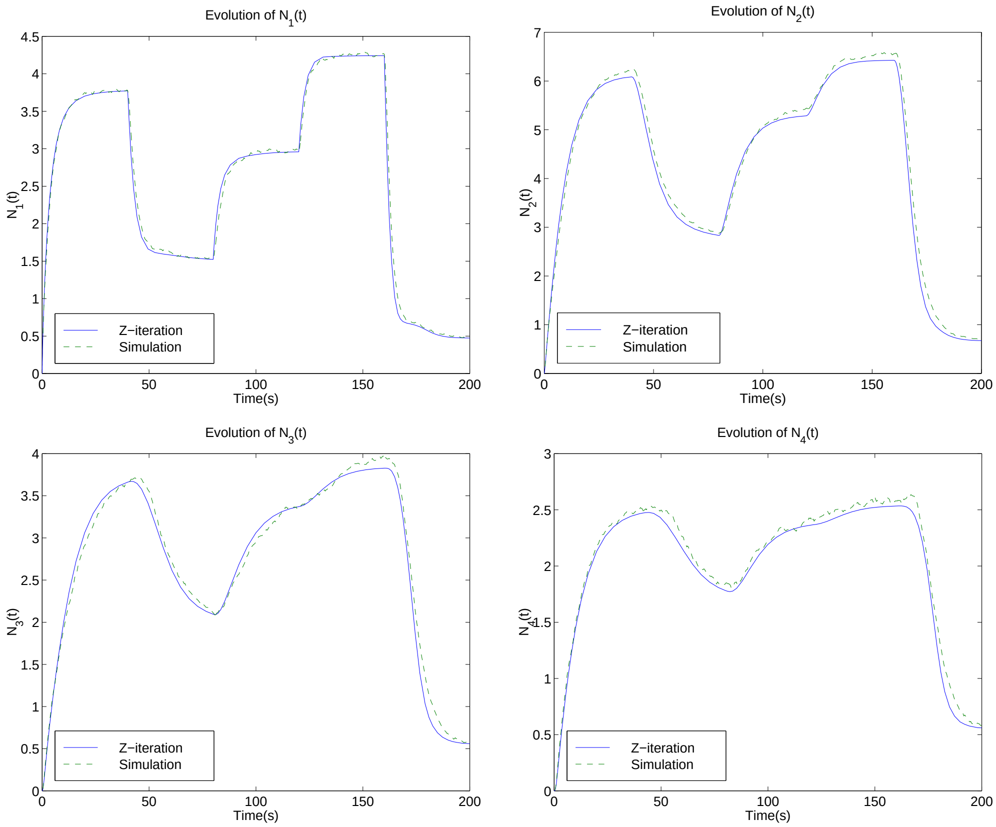

Figure 5: Evolutions for the tandem network with feedback

5 Z-iteration for TCP/IP Networks

We have extended the Z-iteration to handle TCP/IP networks. Consider an IP network with multiple TCP connections routed over some paths. The links have fixed bandwidths and the routers allocate fixed-size buffers for each outgoing link. The traffic is generated by end-to-end bulk TCP connections, each defined by source node, destination node, and connection start and end times. We assume fixed-size packets.

There are three aspects to the extension. First, the TCP/IP network is transformed to a network of queues in the usual way: Each outgoing link of a node is modeled by a queue with service rate equal to the link bandwidth and max queue size equal to the link buffer space. The routing between the queues is determined by the network topology, the connections, and routing tables. To illustrate, suppose node A has an outgoing link A1 to node B, and node B has outgoing links, B1, B2, and B3. Then in the queuing model, the output of A1 can go to B1, B2, B3, or be absorbed within node B, as shown in Figure 8. The probability $r_{\mathrm{A1,B1}}$ of going to B1 is given by the

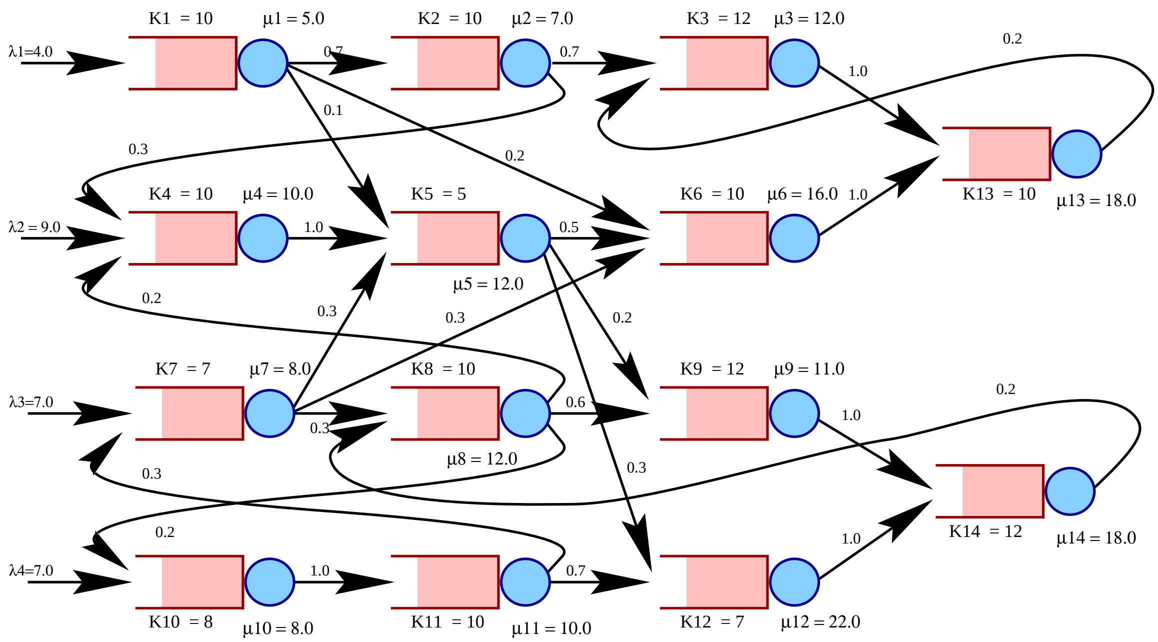

Figure 6: The 14 nodes network

fraction of packets in queue A1 that are forwarded to B1

Second, each TCP source generates packets (customer arrivals) stochastically, with a time-varying rate that accounts for the effects of congestion control. Specifically, at any moment in the Z-iteration computation, the current rate of a TCP source is given by plugging in the current roundtrip time and current loss probability (both available from the current queue sizes and blocking probabilities along the path) into a "TCP profile". The TCP profile is an empirically-obtained function that expresses the instantaneous send rate of a TCP source in terms of the instantaneous roundtrip time and instantaneous loss probability experienced by the source; please refer to [1] for details.

Third, we have to account for the deterministic service times of IP routers. The earlier equations for the M(t)/M(t)/1/K queues do not work well at all. We found that reasonably satisfactory accuracy is achieved by using "deterministic" versions of the equations for dN/dt and U, and retaining the M/M/1/K-derived equations for B and N, as follows:

$$\frac{dN(t)}{dt} = \begin{cases} \lambda(t) - \mu(t)U(t) & \text{if} \quad [\ 0 < N(t) < K\ ] \text{ or} \\ & \qquad [\ N(t) = 0 \text{ and } \lambda(t) > \mu(t)\ )] \text{ or} \\ & \qquad [(\ N(t) = K \text{ and } \lambda(t) < \mu(t)\ )] \\ 0 & \text{otherwise} \end{cases}$$

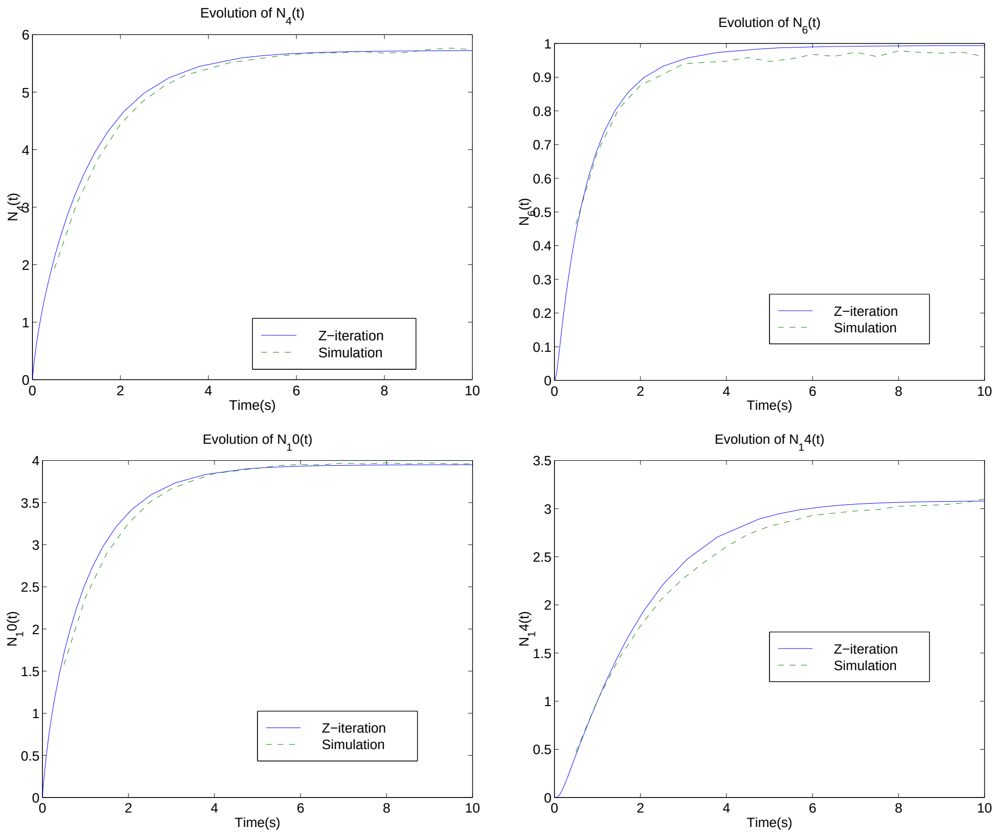

Figure 7: Evolutions of some queues for the 14 nodes network

$$
\begin{aligned}
U(t) &= \min(\lambda(t)/\mu(t), 1) \\
N(t) &= \frac{z(t)}{1 - z(t)} - \frac{(K+1)z(t)^{(K+1)}}{1 - z(t)^{(K+1)}} \\
B(t) &= \frac{1 - z(t)}{1 - z(t)^{(K+1)}}\; z(t)^K
\end{aligned}
\tag{15}
$$

These will generate the following differential equation in $z(t)$:

$$
\frac{dz(t)}{dt} = \begin{cases} \frac{(1-z(t)^{(K+1)})^2(1-z(t))^2}{(1-z(t)^{(K+1)})^2-(K+1)^2 z(t)^K(1-z(t))^2} & \text{for values of } z(t) \text{ such that} \\ \quad \times[\lambda(t) - \mu(t)\min(\lambda(t)/\mu(t), 1)] & \quad [\ 0 < N(t) < K] \text{ or} \\ & \quad [\ N(t) = 0 \text{ and } \lambda(t) > \mu(t)\ )] \text{ or} \\ & \quad [(\ N(t) = K \text{ and } \lambda(t) < \mu(t)\ )\ ] \\ 0 & \text{otherwise} \end{cases}
\tag{16}
$$

Of course, we extend this method to open networks of queues as described in Sec-

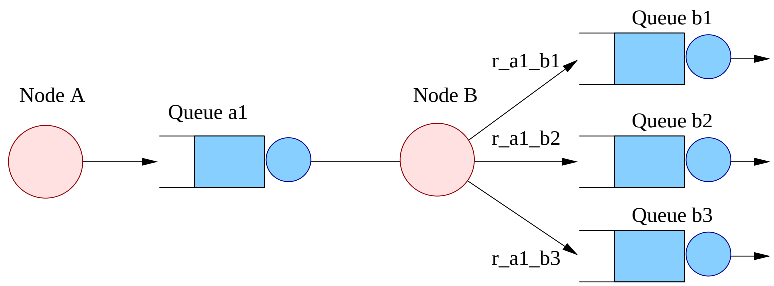

Figure 8: The queuing model for a link from node A to node B.

tion 3, and we employ the techniques presented in Section 4 to deal with numerical instability for large Ks.

Example: We present an example of a network with 18 nodes (Figure 9) and multiple TCP connections. Some connections start at time $t = 0$ and continue until the end of the evaluation; some other connections start at time $t = 30$ and end at time $t = 70$.

We computed the buffer occupancy at the outgoing queues using the Z-iteration and we compare the results against one *ns* simulation run [3]. Our method took 1 second to run, dumping all metrics, while *ns* spent 450 seconds to do the same thing, on the same machine. Without *any* disk access (i.e. no dumping of metrics) *ns* spent 20 seconds while our program runs in fractions of a second.

The plots for some queue sizes are displayed in Figure 10.

6 Conclusions

Queuing systems are a natural way of modeling computer networks and many other systems. One usually obtains steady-state metrics of queuing models, because this is

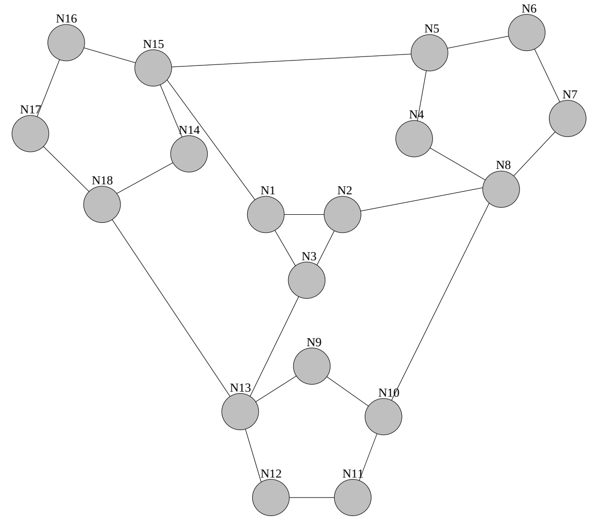

Figure 9: The TCP example network.

relatively tractable for many kinds of queuing systems. But steady-state metrics do not offer answers to many interesting questions. Transient evolutions, on the other hand, can provide answers to most questions that one would like to ask in evaluating a system. They also allow for realistic modeling of critical events such as network routing updates, unlike steady-state models.

Transient metrics are usually very hard to obtain, unmanageable by analytical methods and time-consuming by simulation. But the Z-iteration changes this premise, allowing very fast computation of some very useful transient metrics. It translates a queuing network with N nodes into a system of N coupled differential equations. Looking at the results and the run time, we conclude that it offers the power of simulation at a fraction of cost for these transient metrics.

Regarding future work, a theoretical justification for the Z-iteration is very much needed.

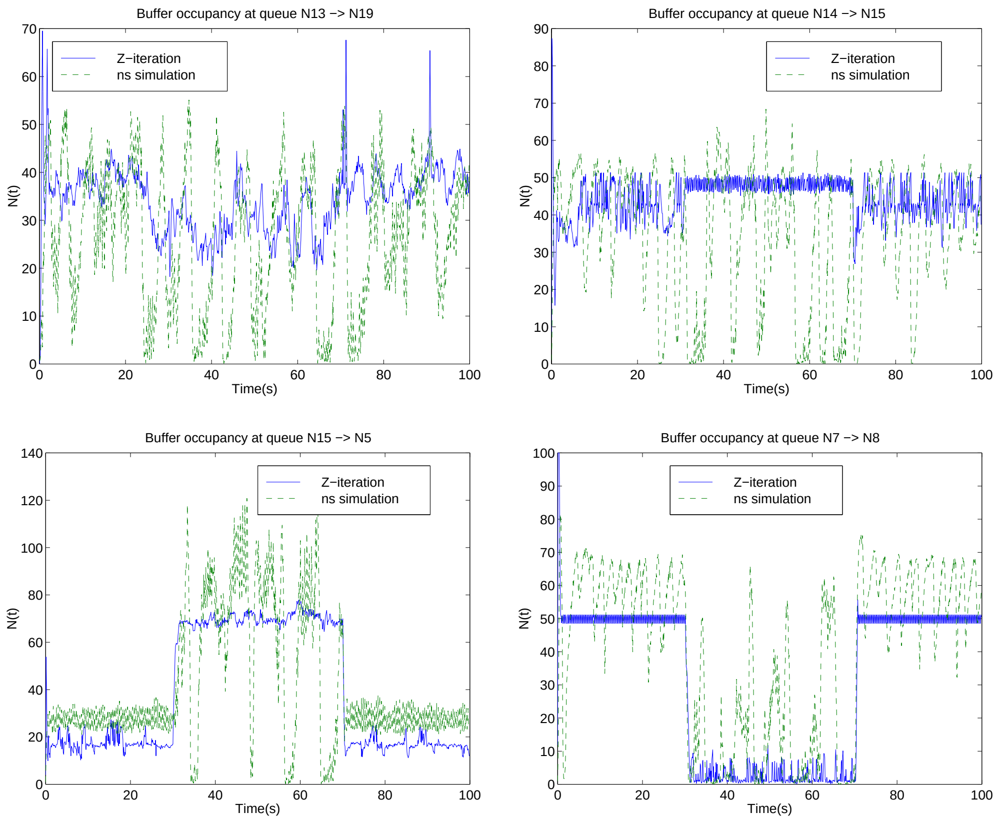

Figure 10: The TCP example network - Results.

References

[1] C.T. Popescu and A.U. Shankar. Empirical TCP Profiles and Application. In *IEEE International Conference on Network Protocols '99*, November 1999.

[2] D. Tipper and M.K. Sundareshan. Numerical Methods for Modeling Computer Networks Under Nonstationary Conditions. In *IEEE J. Select Area Commun., 8(9):1682-1695*, 1990.

[3] K. Fall and K. Varadhan. *ns Notes and Documentation.*

[4] I. Matta and A.U. Shankar. An Iterative Approach to Comprehensive Performance Evaluation of Integrated Services Networks. In *IEEE International Conference on Network Protocols '94*, Boston, Massachusetts, October 1994.

[5] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag,

Second edition, 1993.

[6] K.S. Trivedi. *Probability & Statistics with Reliability, Queuing and Computer Science Applications*. Prentice Hall, 1982.

[7] L. Kleinrock. *Queueing Systems*, volume I and II. New York: Wiley, 1976.

[8] I. Matta and A.U. Shankar. Z-iteration: A simple method for throughput estimation in time-dependent multi-class systems. In *ACM SIGMETRICS/PERFORMANCE '95*, pages 126–135, Ottawa, Canada, May 1995.

[9] S. Tripathi and A. Duda. Time-Dependent Analysis of Queuing Systems. In *INFOR, 24(3):199-219*, 1986.