Transient Behavior of ATM Networks Under Overloads *

Chang-Yu Wang¹, Dimitris Logothetis², Kishor S. Trivedi³ and Ioannis Viniotis⁴

¹Dept. of Computer Science, Center for Adv. Computing & Comm., Duke University, Durham, NC 27708
²GTE Laboratories Inc., Waltham, MA 02254
³Dept. of Electrical Eng., Center for Adv. Computing & Comm., Duke University, Durham, NC 27708
⁴Dept. of Electrical and Computer Eng., Center for Adv. Computing & Comm., North Carolina State University, Raleigh, NC 27695

Abstract

In this paper we characterize the time-dependent behavior of typical queueing systems that arise in ATM networks under the presence of overloads. Transient queue length distribution and transient cell loss probability are obtained numerically and transient characteristics such as maximum overshoot and relaxation time are used to quantify the effects of congestion periods. A new measure, Expected Excess Loss in Overload (EELO) is defined to quantify the effects of overload when compared with the system behavior in the steady-state regime. The basic modeling technology that we use is an extended form of Stochastic Petri nets and a software tool called the Stochastic Petri net package (SPNP).

1 Introduction

Overload conditions in networks represent an undesirable but nevertheless inevitable situation. Unfortunately, steady-state analyses fail to predict such transient phenomena. Time-dependent analysis is required to accurately compute measures such as queue size and overshoots, the length of congestion periods; such measures can be used to better dimension the buffer size. Transient analysis of queueing systems is known to be an analytically non-tractable problem even for simple queueing systems. The queue length distribution of an M/M/1 queue is given as an infinite sum of modified Bessel functions of the second kind [10] which is in general hard to evaluate. In [7], a simple analytical expression of the transient probabilities of an M/M/1 queue has been developed given that the queue is initially empty. However, when the queue is not initially empty, the expression obtained still requires computations of the modified Bessel function of the first kind. More complex queueing systems do not yield analytically tractable results. In particular, queueing networks are very hard to analyze for transient measures since desirable properties such as "product form" and Little's law are valid only for stationary measures. For Markovian queueing systems, numerical computation based on the Chapman-Kolmogorov differential equations provides an alternative solution, which however, suffers from state space explosion. Randomization or uniformization is commonly used to solve the underlying continuous-time Markov chain [6, 14]. Fluid-flow or diffusion approximations have been considered to overcome the state space explosion by approximating discrete-state processes by continuous-state ones and solving partial differential equations. Most of these methods, however, are limited in that they imply heavy traffic assumptions that may actually overestimate the measures of interest.

In this paper, we study the exact discrete-state processes by means of Stochastic Reward Nets (SRNs) [3]. SRNs are a form of stochastic Petri nets and provide a specification language that facilitates the automated generation and solution of large Markovian models like the ones generated when studying the transient behavior of queueing networks. We allow for arrival processes that can capture both correlation and burstiness, two important factors when characterizing arrivals in ATM networks. Our analysis provides valuable insights when dimensioning buffers in ATM networks to control overload peaks and their durations.

The purpose of this paper is to quantify the effects of transients in communication networks. Our approach differs from previous approaches in that we consider actual point processes rather than fluid-flow or diffusion approximations and that we consider more general arrival processes (such as the Markov Modulated Poisson Process (MMP)) and general service time distributions (such as k-stage Erlang). Transient effects are quantified using measures such as relaxation time, maximum overshoot and a new measure, Expected Excess number of Losses in Overload (EELO).

The remainder of the paper is organized as follows: in Section 2 we provide the necessary background for transient measures and the Stochastic reward nets. In Section 3 we present the queueing model of the net-
work while in Section 4, we develop the SRN model that corresponds to the queueing system. In Section 5 we show our numerical results and finally in Section 6 we provide our conclusions.

2 Background

Although there is a significant amount of work for the steady-state analysis of queueing models, there is considerably less work done in the transient analysis of queues. The early works of Takács [15] and Benes [2] represent a rare incursion into the transient behavior of the infinite capacity single server queues. Results for the M/G/1 queue are left in terms of transforms which have to be inverted to provide numerical results. Very few results such as Odoni and Roth [12] provide empirical evaluations of the time-constants involved in transient analysis.

2.1 Measures of interest

Let S denote the state space of a queueing system, and pi(t) denote the probability that the system is in state i at time t. Note that i is generally a vector. Also, let q(t) denote the queue length distribution of the queueing system at time t, then qj(t) = \sum_{j \in S_j} p_i(t) where Sj is the subset of S such that the queue length is equal to j. The mean queue length at time t is defined as:

\[ E[N(t)] = \sum_{j=0}^{K} j q_j(t) \]  

where K is the size of the queue. The probability that the queueing system is full at time t is simply given by:

\[ p_{full}(t) = q_K(t) \]  

Now, the probability that an incoming customer (cell) is lost \(^1\) is given by [8]:

\[ p_{loss}(t) = P(\text{buffer full at } t | \text{ arrival at time } t) = \frac{\sum_{i \in S} p_i(t) \lambda_i}{\sum_{j \in S} p_j(t) \lambda_j} \]  

where \( \lambda_i \) denotes the arrival rate into the queueing system when the queueing system is in state j at time t. When comparing Equations (2) and (3) we see that these two measures are the same when \( \lambda_i = \lambda \) (i.e., state independent arrivals). Therefore the probability of tagged loss and the probability of the buffer being full are equal only for state independent arrivals i.e., Poisson arrivals. This will not be true however for more general arrival processes such as MMPP. In the following, we will compute all the above mentioned measures.

2.2 Transient characteristics

We now turn our attention to quantities that are commonly used to characterize time-dependent behavior. The discussion is based on the control theory literature, see for example [11]. We assume that the measure of interest as a function of time is denoted by \( m(t) \). We define:

- Maximum Overshoot \( m_{ov} \) the difference between the maximum peak value of \( m(t) \) and the steady state value \( m(\infty) \). Mathematically, it is given as \( m_{ov} = \max_{t} (m(t) - m(\infty)) \) where \( t \) are the (finite) solutions of the equation \( \frac{d^2m(t)}{dt^2} = 0 \) with the additional constraint \( \frac{d^2m(t)}{dt^2} \bigg|_{t=t} < 0 \).

- Relaxation time (Settling time) \( t_r \) is the time required for \( m(t) \) to reach and stay within a range \( \epsilon \) of the steady-state value \( m(\infty) \).

- Expected excess loss in overload (EELO) The previously defined characteristics, overshoot and relaxation time, quantify transient behavior along the y-axis and the time axis, respectively. The next measure captures the transient effects in both axes since it is defined as an integral of the measure in a given time interval. In order to quantify the effect of congestion, especially for the measure of loss, we define the Expected Excess Loss in Overload (EELO) as follows:

\[ EELO = \int_{0}^{\infty} l_r(x)dx - l_r(\infty)t_s \]  

where \( l_r(t) \) is given by:

\[ l_r(t) = \sum_{j \in S} \lambda_j p_j(t) \]  

is the expected loss rate at time t.

The definition of EELO is illustrated in 1. Basically, EELO is the difference of the two shaded areas shown in the figure. EELO is a measure that shows “how bad” is the effect of the overload. It provides us with an estimate of how many more cells we expect to lose during the duration of the overload when compared with standard steady-state-based estimate.

Traffic in broadband networks is expected to be highly bursty. The mathematical measure of burstiness has not been finalized yet. Here, we will simply consider the squared coefficient of variation as our measure of burstiness:

\[ C_A^2 = \frac{Var[A]}{E[A]^2} \]  

where \( A \) denotes the interarrival times. The computation of this coefficient of variation of \( (C_A) \) is not straightforward since the interarrival times are correlated (such a computation is explained in [5]).

It is well known that the inter-arrival time of a Poisson process has a squared coefficient equal to 1 and therefore cannot capture burstiness in arrivals. For this reason, alternative processes should be considered. One such process that captures both burstiness and correlation in successive interarrivals and yet is analytically tractable is the Markov Modulated Poisson Process (MMPP). For the purposes of this paper we will consider a two state MMPP but our models are easily extended for any arbitrary size MMPP.
but extend it further. In SRN, every tangible marking can be associated with a reward rate. It can be shown that a SRN can be mapped into a Markov reward model. Therefore, by using software tools [4], the underlying Markov reward model of a SRN is automatically generated and solved. Thus a variety of performance measures can be specified and evaluated using a very convenient formalism. SRN also allows some other features that makes specification convenient: each transition may have an enabling function such that a transition is enabled only if its enabling function is true. Marking dependent enabling function and arc multiplicities are introduced in SRN. Transitions can also have priorities. If two transitions are enabled and ready to fire simultaneously, then 1) they have the same priorities in which case the transition to fire is determined probabilistically, or 2) the one with higher priority is picked to fire.

To represent a SRN as a graph, places are represented by circles and tokens are black dots (or integers) inside places. Immediate transitions are drawn as thin bars and timed transitions as white boxes. Inhibitor arcs have small hollow circles instead of arrows at their terminating ends. The current number of tokens in a place $p$ is denoted as $\#p$.

### 3 The queueing model

In this section, we describe the queueing model and define the notations used in this paper. Figure 2 (a) shows three network nodes, namely, $N_1$, $N_3$ and $N_2$ and two established connections $(N_1, N_3)$ and $(N_2, N_3)$. These connections can be either direct link connections or virtual path connections defined by a series of links. At the occurrence of a failure in the connection $(N_1, N_3)$ traffic destined for $N_3$ is chosen to be rerouted through $N_2$. Before the global routing table is updated, the rerouting decision is made locally; cells destined to node $N_3$ still arrive at node $N_1$.
first without knowing that the outgoing link of node $N_1$ is broken. Figure 2(b) shows the output queue of node $N_1$ that contains packets or cells destined for $N_2$. This queue is feeding the input queue of node $N_2$. Redirected cells from node $N_1$ are rejected and lost if the input queue of node $N_2$ is full. Right after the rerouting starts, all the cells leftover in the queue of node $N_1$ are sent to node $N_2$ one by one; this may cause overload in node $N_2$. The queue length and the cell loss probability of node $N_2$ will overshoot during this transient period, and they then reach a new steady state.

We assume that the cell transmission time of node $N_i$ has Erlang distribution with mean $\mu_i$ and $r_i$ number of phases. To model the burstiness of the ATM traffic, we also assume that the cell arrival process of node $N_i$ is a two-state MMPP with infinitesimal generator $Q_i$ and arrival rates $(\lambda_1, \lambda_2)$. Our model can be very easily extended to MMPPs with larger numbers of states.

The notation is summarized in the following:

- $K_i$: queue size of node $N_i$.
- $\mu_i$: mean cell transmission time at node $N_i$.
- $Q_i = \begin{bmatrix} -a_i & a_i \\ b_i & -b_i \end{bmatrix}$: the infinitesimal generator of the auxiliary CTMC of the two-state MMPP arrival process to node $N_i$.
- $(\lambda_1, \lambda_2)$: arrival rate vector of the two-state MMPP process to node $N_i$.

The state of the system described above at time $t$ is characterized by a vector $Z(t) = (X_1(t), J_1(t), B_1(t), X_2(t), J_2(t), B_2(t))$, where $X_i(t)$ denotes the number of cells in node $N_i$ at time $t$, $J_i(t)$ denotes the state of the MMPP arrival process to node $N_i$ at time $t$, and $B_i(t)$ denotes the phase of the Erlangian service time of node $N_i$ at time $t$. The stochastic process $\{Z(t), t \geq 0\}$ is a Continuous-Time Markov Chain (CTMC), and its transient behavior can be derived by solving a system of differential equations $dP(t) = P(t)Q$, where $P(t)$ is the matrix of probabilities $P_{ij} = P\{Z(t) = j | Z(0) = i\}$ and $Q$ is the infinitesimal generator of the CTMC. However, the CTMC has number of states equal to $(2 + 2r_1K_1) \times (2 + 2r_2K_2)$; for a case that $r_1 = r_2 = 5$ and $K_1 = K_2 = 16$, the CTMC will have 26,244 number of states. To generate the CTMC manually is out of the question, not to mention to solve the system of 26,244 differential equations. Therefore, we use the tool SPNP to automatically generate and solve the CTMC in transient state.

### 4 SRN model

In this section, we present the SRN model of the queuing network. The SRN model is shown in Figure 3. The place $mmpp_\text{i}$ and transitions $t1_\text{i}$, $t2_\text{i}$, $tar_\text{i}$ model the two-stage MMPP arrival process to the $i$th queue. When the place $mmpp_\text{i}$ is not empty, transition $t1_\text{i}$ will fire with rate $a_i$, otherwise transition $t2_\text{i}$ will fire with rate $b_i$. This models the auxiliary CTMC of the MMPP source. Transition $tar_\text{i}$ has marking dependent firing rate. If the place $mmpp_\text{i}$ is not empty, the firing rate of transition $tar_\text{i}$ is $\lambda_i$, otherwise it is $0$. Place $buf_i$ represents the buffer of the $i$th queue, and submodels $rerouting time$ and $service time$ model the Erlang distributed cell transmission times. Firing transition $Erj_\text{i}$ will deposit $r_i$ tokens in the place $Er_token_\text{i}$; this models the start of the service. Transition $Ertrans_\text{i}$ will then fire $r_i$ times which models the $r_i$ sequential exponential phases of the Erlang distributed service time. When the place $Erstage_\text{i}$ has collected $r_i$ tokens, it indicates that a job has finished its service and is ready to depart the system; transition $Erput_\text{i}$ fires and take all the $r_i$ tokens. Note that the server can process only one cell at any time, that is, the total number of tokens in places $ErToken_\text{i}$ and $ErStage_\text{i}$ must be zero before the next cell can be admitted. This is modeled by the inhibitor arcs from places $ErToken_\text{i}$ and $ErStage_\text{i}$ to transition $ErTrans_\text{i}$.

To model the limited sizes of queues, transitions $t_1$ and $t_2$ are controlled by the enabling functions such that $t_1$ and $t_2$ can fire only if the corresponding queue is not full. Rejection of cells from the first queue to the second queue is modeled by the variable cardinality arc from submodel $rerouting time$ to the place $buf_2$. If the second queue is full, no token is deposited in the place $buf_2$ which models the loss of cells.

To compute the cell loss probability and the queue length of the second queue, we use reward rates. The reward rate function $QLen$ returns the total number of tokens in place $buf_2$ plus the total number of tokens in places $ErToken_\text{i}$ and $ErStage_\text{i}$ divided by the number of phases $r_1$ which represents the total number of cells in the $i$th queue. The reward rate function $PFull$ returns one if the second queue is full (when rejection occurs), otherwise it returns zero. The function $ELR$ computes the expected loss rate at node $N_2$, and the function $Arrival$ computes the expected arrival rate to node $N_2$. The cell loss probability is simply obtained by computing the ratio of $ELR$ to $Arrival$. Once we obtain the relaxation time of the expected loss rate (ELR), the value of EELO is equal to the expected accumulated value of ELR till the relaxation time minus the product of the steady state value of ELR and the relaxation time.

### 5 Numerical Results

In this section we present numerical results that were obtained using the Stochastic Reward Net model presented in the previous section. Our goal is to characterize transient overload conditions and in particular maximum overshoot and relaxation times. We also wish to compare transient and steady-state values of the EELO measure. We focus on the transient mean queue length and transient loss probability as the measures of interest. For both nodes, we assume a 5-stage Erlangian service time with mean 2.73 $\mu$sec per byte long ATM cells on a 155 Mbit/s link. Also, the size of each queue is assumed to be 16.

Figures 4 shows the transient mean queue length for different initial conditions of the MMPP sources.
The parameters of both MMPP sources are chosen to be $a = 0.0269163$, $b = 0.00672908$, $\lambda_1 = 1.5658$, and $\lambda_2 = 0.00301161$ which yields the squared coefficient of variation equal to 50 and utilization (before failure) equal to 0.85. Note that state 1 of the MMPP source is considered as the “busy” state; cells arrive much faster than in state 2. When both MMPP sources start at state 1, which is the worst case, the queue length overshoots and stays high for a while. In contrast, if both sources start from state 2, we see only a small overshoot and then the queue length drops. This indicates that the overshoot at the second queue actually comes from two factors: the burst of the incoming traffic, and the traffic rerouted from the first queue. Compare the case that the second queue starts at state 1 and the first queue starts at state 2 with the vice versa, the burstiness of the traffic to the second queue has more effect to the overshoot phenomenon.

Figures 5 and 6, show the transient loss probability and the transient mean queue length for moderate utilization before failure ($\rho = 0.5$) as a function of burstiness. The initial condition is assumed to be steady-state queue length in both queues prior to the occurrence of the overload (i.e., switch in position (1) in Figure 2). Note that in all cases a bursty input traffic such that the squared coefficient of variation value greater than or equal to 10 will produce an overshoot. Note also that the transient effects are more “severe” for higher coefficient of variation values ($\sigma_A$) since in many cases long-run values of the measure are smaller for higher coefficient of variation values. (Compare the value of the mean queue length at $t = 200$ for $\sigma_A$ of 10 and 20).

The effect of coefficient of variation on relaxation time are shown on Figures 7 and 8. For all measures, if an overshoot occurs ($C_A^2$ greater than or equal to 10), the relaxation time increases as $C_A^2$ increases, otherwise the relaxation time declines with increasing $C_A^2$. Also, the relaxation time increases as the utilization decreases (see Figure 8).

Similarly, the effect of coefficient of variation on the maximum overshoot of transient loss probability and transient mean queue length are shown in Figures 9 and 10, respectively. Figure 9 shows that the maximum overshoot of the transient loss probability first increases and then decreases as $C_A^2$ increases for medium and high utilization cases. This is because the steady-state value of the loss probability increases as the $C_A^2$ value increases which makes the difference between the maximum and the steady-state value (i.e. the maximum overshoot) smaller. The maximum overshoot of the transient mean queue length increases as the $C_A^2$ value increases.

The effect of $C_A^2$ value on EELO are shown on Figure 11 and Table 1. Figure 11 shows that the value of EELO increases as the $C_A^2$ value increases and also as the utilization increases. This indicates that for burstier traffic and higher utilization, systems, dimensioning based on the steady-state analysis will underestimate the overload condition. For example, in this paper, we show the values of EELO and the relaxation time of the expected cell loss rate for the high utilization case ($\rho = 0.85$), for input traffic with $C_A^2$ value 40, we expect to lose 125.12 more cells when compared with steady-state analysis before the overload condition is considered relaxed (in a time interval of 68.46 μsec).

In Table 1 we also compare the values of EELO with different initial conditions on the first queue: initially full or initially the steady state queue length. Assuming the first queue is initially full provides us with a worst case analysis, since the overload condition becomes more severe if the first queue reroutes more cells to the second queue. From Table 1, we find that changing the initial condition has more impact on the less bursty cases; the value of EELO increases and the relaxation time is longer. However, for burstier traffic (burstiness more than 50), the values of EELO and relaxation time are almost the same with different initial conditions. This is because that for burstier traffic, an arrival of a large burst of cells to the second queue is the dominant cause of the overload on the second queue.

6 Conclusion

In this paper we characterized the congestion periods and their effects using transient analysis and stochastic Petri Nets. In order to compare steady-state and transient measures we define a new measure Expected Excess Loss in Overload (EELO). Our results suggest that as traffic burstiness increases the larger is the error from underestimating expected number of lost cells and mean queue length when compared with steady-state values. Not only can our model predict the transient effects, it can also quantify the overload condition. For example, based on our experiment, around 125 more cells will be lost during the overload period in the worst case compared with steady-state analysis. It is left to the network designer to judge if the anticipated network traffic is bursty enough and decide to dimension network buffers based on transient measures rather than steady-state ones.

Acknowledgment

The authors would like to thank Dr. Raymond Marie for conversations that were helpful in the preparation of this paper.

References


Figure 7: Relaxation time vs. $C_A^2$ (util = 0.5)

Figure 8: Queue length relaxation time vs. utilization

Figure 9: Prob. of loss overshoot vs. $C_A^2$

Figure 10: Queue length overshoot vs. $C_A^2$

Figure 11: EELO vs. $C_A^2$

<table>
<thead>
<tr>
<th>Burstiness</th>
<th>EELO Mean</th>
<th>EELO Full</th>
<th>ELR Rel. time (μsec) Mean</th>
<th>ELR Rel. time (μsec) Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.11</td>
<td>-2.17</td>
<td>126.00</td>
<td>348.06</td>
</tr>
<tr>
<td>2</td>
<td>-9.41</td>
<td>-2.52</td>
<td>143.50</td>
<td>267.50</td>
</tr>
<tr>
<td>5</td>
<td>-3.71</td>
<td>-0.19</td>
<td>188.50</td>
<td>201.58</td>
</tr>
<tr>
<td>10</td>
<td>3.73</td>
<td>3.76</td>
<td>166.00</td>
<td>187.00</td>
</tr>
<tr>
<td>20</td>
<td>8.08</td>
<td>10.86</td>
<td>173.00</td>
<td>188.00</td>
</tr>
<tr>
<td>30</td>
<td>19.70</td>
<td>20.83</td>
<td>194.00</td>
<td>197.50</td>
</tr>
<tr>
<td>50</td>
<td>33.39</td>
<td>34.21</td>
<td>231.17</td>
<td>234.76</td>
</tr>
<tr>
<td>60</td>
<td>51.13</td>
<td>51.76</td>
<td>305.69</td>
<td>305.69</td>
</tr>
<tr>
<td>100</td>
<td>66.34</td>
<td>66.88</td>
<td>378.73</td>
<td>378.73</td>
</tr>
<tr>
<td>150</td>
<td>85.54</td>
<td>86.00</td>
<td>475.61</td>
<td>475.61</td>
</tr>
<tr>
<td>100</td>
<td>125.12</td>
<td>125.35</td>
<td>684.98</td>
<td>684.98</td>
</tr>
</tbody>
</table>

Table 1: EELO and relaxation time of ELR for $\rho = 0.85$
Transition | Rate Function | Enabling Function
--- | --- | ---
\( tar_1 \) | if ( \#mmpp.1 \( \lambda_1^1 \) else \( \lambda_1^2 \)) \( Qlen_1 < K1 \) |
\( tar_2 \) | if ( \#mmpp.2 \( \lambda_2^1 \) else \( \lambda_2^2 \)) \( Qlen_2 < K2 \)

Measure | Reward Rate Function
--- | ---
\( QLen_1 \) | return \( \frac{\text{\#buf}_1 + (\text{\#Er\_token}_1 + \text{\#Er\_stage}_1)}{r_1} \); |
\( QLen_2 \) | return \( \frac{\text{\#buf}_2 + (\text{\#Er\_token}_2 + \text{\#Er\_stage}_2)}{r_2} \); |
PFull | if \( Qlen_2 == K2 \) return(1.0); |
\( ELLR \) | if \( QLen_2 == K2 \) \{ |
\( \text{ret\_val} = \#mmpp.2 \lambda_2^2 \lambda_2^2 \); |
\( \text{if} (\#Er\_token_1 == 1) \text{ret\_val}+ = r_1 / \mu_1; \) |
\( \text{return(ret\_val);} \) |
\( \text{else return(0.0);} \) |
Earrival | ret\_val = \#mmpp.2 \( \lambda_2^2 \); |
\( \text{if} (\#Er\_token_1 == 1) \text{ret\_val}+ = r_1 / \mu_1; \) |
\( \text{return(ret\_val);} \)

Figure 3: SRN model for two-state MMPP arrivals and Erlang transmission time