Performance of Broadcast and Unknown Server (BUS) in ATM LAN Emulation[*]

Hairong Sun, Xinyu Zang and Kishor S. Trivedi
{hairong, xzang, kst@ee.duke.edu}
Center for Advanced Computing and Communications
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708

Abstract

In this paper, we develop a performance model of the Broadcast and Unknown Server (BUS) in the LANE. The traffic on the BUS is divided into two classess: the broadcast and multicast traffic, and the *unicast relay flow*. The broadcast and multicast traffic is assumed to form a Markov Modulated Poisson Process (MMPP). The traffic for a particular *unicast relay flow* is an MMPP as well. However, the number of active *unicast relay flows* sojourning on the BUS is determinated by a tandem queueing system, where the flow arrival process is Poisson, the address resolution delay is exponentially distributed, and the connection setup delay is 3-stage Erlang distributed. The size of data frames in traffic flows is a stochastic variable which has three possibilities: short, medium, and large. In order to deal with the intractability (i.e., largeness and stiffness) of the underlying Markov chain, a hierarchical model is used to decompose the system. With the help of $SPNP$ (Stochastic Petri Net Package), a software package for the automated genaration and solution of Markovian stochastic systems, and the decomposition method, the performance of the BUS affected by the stochastic processes at different time scales is studied. We also investigate the effect of address resolution delay and conection setup delay on the performance of the BUS.

Index Terms: ATM, LAN Emulation, Broadcast and Unknown Server, Stochastic Reward Nets, Stochastic Petri Net Package (SPNP)

[*] This research was supported in part by the National Science Foundation under Grant No. EEC9418765, and by the North Carolina Advanced Network Program as an enhancement Project to the Center for Advanced Computing and Communications in Duke University.

1 Introduction

Over the last decade, ATM (Asynchronous Transfer Mode) technology has emerged as a key component of next generation networks. It offers unprecedented scalability and performance /cost ratio, as well as the ability to reserve network resources for real-time traffic and support multimedia and multipoint communications. Obviously, in the future information infrastucture, ATM will play an important role. However, the current information infrastructure, e.g., the vast installed base of LANs (Local Area Networks) and WANs (Wide Area Networks), uses internetwork layer protocols such as IP, IPX and AppleTalk to internetwork the subnets. Therefore, a key to ATM's success and Internet's further success will be the ability to allow for interoperation between the existing network technologies and ATM. It is thus necessary to be able to use network layer protocols such as IP and IPX on both existing networks and ATM networks, since it is the function of the network layer to provide a uniform network view to higher level protocols and applications [1, 2, 3].

LANE (LAN Emulation) [3, 4], standardized by ATM Forum, is one of the several solutions to integrate IP and ATM and has been implemented by almost all the vendors in their local area ATM switches. As the name suggests, the function of the LANE protocol is to emulate a LAN on top of an ATM network, and enable existing applications to access the ATM network via protocol stacks such as NetBIOS and IPX, as if they were running over traditional LANs. The LANE protocol defines a service interface for the network layer protocols, which is identical to that of existing LANs. Data sent across the ATM network are encapsulated in the appropriate MAC (Medium Access Control) frame format. LANE works at the MAC layer and enables legacy Ethernet, Token Ring, or FDDI traffic to run over ATM with no modifications to applications, network operating systems, or desktop adapters. Legacy end stations can use LANE to connect to other legacy systems, as well as to ATM-attached servers, routers, hubs, and other networking devices.

Each ELAN (Emulated LAN) is composed of a group of LAN Emulation Clients (referred to as LECs) and a LAN Emulation Service (referred to as LE Service). LECs may be bridges/routers acting as a proxy for end-systems or ATM hosts. An LE Service consists of three different functional entities: the LE Configuration Server (LECS), the LE Server (LES) and the Broadcast and Unknown Server (BUS). The LE Service may reside in ATM switches, routers, bridges or workstations. The three servers comprising LE Service may exist independently but are typically colocated in a single device. The LECS is responsible for dynamically assigning different LECs to different emulated LANs. The LES handles address resolution and control information, and its primary job is to register and resolve MAC addresses to ATM addresses. The BUS is designed for carrying broadcast and multicast traffic, and the initial unicast frames for the LECs before point-to-point direct Virtual Channel Connection (called *Data Direct VCC*) is established between the source/destination pairs (for the ease of description, we call this kind of traffic *unicast relay flow*[1])

The LANE specification was standardized by ATM Forum in May 1995, and widely implemented in the existing ATM switches. However, the performance analysis for the LANE is rare in the references, and only some measurement-based study can be found [8, 9]. Because the BUS must forward not only all the broadcast and multicast frames, but also the frames for the LECs before they establish the *Data Direct VCC* to the destinations,

[1] In the previous studies *flow* was defined as a sequence of IP datagrams containing the same values for the following IP and transport fields in one period: IP version, source and destination address, header length, type-of-service, and time-to-live (TTL). The *flow* concept may be used to implement flow-based switching and develop the efficiency of IP packet forwarding. If we adopt virtual channel multiplexing in ATM networks to transfer IP packets, the frames belonging to the same *flow* will be transfered via the same *Data Direct VCC*.

some researchers conjectured that the BUS is the potential bottleneck in the process of forwarding frames [5].

In this paper, we develop a performance model of the BUS in the LANE. The traffic to the BUS is divided into two classes: the broadcast and multicast traffic, and the *unicast relay flow.* The broadcast and multicast traffic is assumed to form a Markov Modulated Poisson Process (MMPP). The traffic for an individual unicast flow is an MMPP as well. However, the number of active *unicast relay flows* sojourning on the BUS is determined by a tandem $M/M/1/-E_3/1$ queueing system where the flow arrival process is Poisson, the address resolution delay is exponentially distributed, and the VCC setup delay is 3-stage Erlang distributed. The size of frames is a stochastic variable which has three possibilities as observed by the previous studies [14]. From intuition, the performance of the BUS is mainly affected by the probabilistic behavior of the *unicast relay flow* and broadcast traffic. While the effect of broadcast traffic on the BUS is easier to understand, the influence of *unicast relay flow* on the BUS is very complicated. Various processes at connection level, bursty level and cell level together contribute to the complexity. The arrival process of the flow, the address resolution delay and connection setup delay are connection level processes and determine the number of active *unicast relay flows* on the BUS. The frame arrival process during the lifetime of a particular *unicast relay flow* belongs to bursty level. The frames whose size may be a stochastic variable must be segmented to fixed-size cells and then transmitted at cell level. These three levels work at widely different time scales, from several microseconds to tens of minutes. It is a challenge to solve the underlying Markov chain because of the stiffness, which is caused by the large differences in the rates of occurrence of cell-level events and the rates of the rare, connection-level or burstiness-level events. Stiffness leads to difficulty in the solution of the model and numerical instability. Methods of dealing with stiffness are discussed in [16, 17]. In this paper, we adopt a hierarchical model to solve the stiffness existing in the performance analysis of LANE. The performance of the BUS affected by the stochastic processes at different time scales is studied. The underlying Markov models are generated automatically via stochastic reward net models and solved by $SPNP$ (Stochastical Petri Net Package), a package developed by us [20]. We also investigate the effect of address resolution delay and conection setup time on the performance of the BUS.

The organization of the paper is as follows. In Section 2, a brief description of LANE is given. In Section 3, the stochastic reward net models for connection setup delay, address resolution delay, broadcast traffic, *unicast relay flow* and the BUS system are presented. Numerical results obtained by using $SPNP$ are discussed in Section 4. Conclusions are drawn in Section 5.

2 LAN Emulation

The LANE specification is based on a client-server implementation model. An ELAN consists of one LAN Emulation service and multiple LECs communicating through the LUNI (LAN Emulation UNI). After a series of procedures (i.e., Initialization and Registration [3]), the LEC gets the ATM address of the LECS, the LES, and the BUS, and establishes Virtual Channel Connections (VCCs) to them.

Fig. 1 shows the complete procedure of a host on legacy LAN communicating with an ATM host. The source LEC resides in a LAN/ATM bridge and is the proxy of the legacy LAN hosts connected to the bridge. The desination LEC resides in the ATM host.

- To determine the destination's MAC address, the source broadcasts an ARP request containing the destination's IP address. This is a standard procedure in any IP

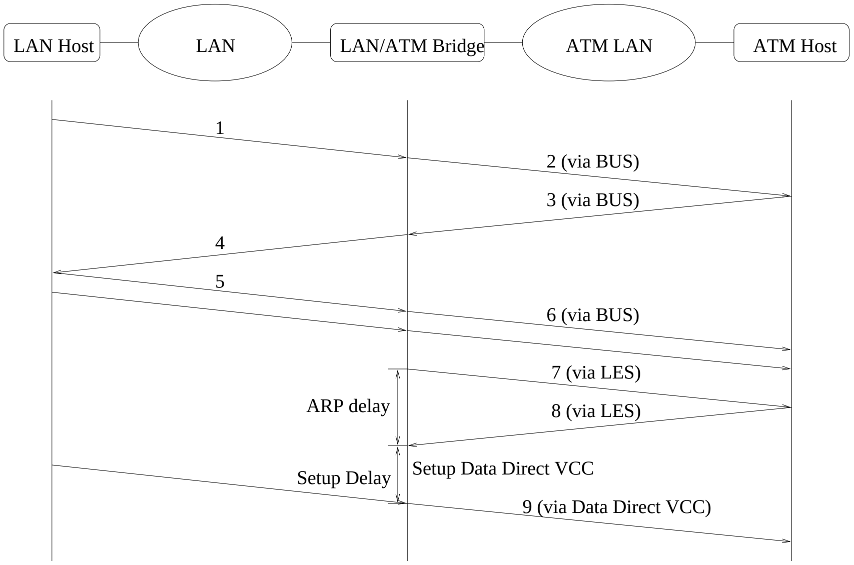


Figure 1: Procedure of a Host on Legacy LAN Communicating with an ATM Host

network. The *ARP* request reaches the LAN/ATM bridge via the legacy LAN (see the arc labelled 1 in Fig. 1).

- The LEC in the LAN/ATM bridge forwards the broadcast packet to the BUS. The BUS sends the *ARP* request to all members of the emulated LAN (see the arc labelled 2 in Fig. 1).
- The destination receives the *ARP* request and recognizes its own IP address. In response, it puts its own MAC address into an *ARP* reply. Because there is no direct VCC to the LAN/ATM bridge yet, the destination LEC sends the *ARP* reply to the BUS. The BUS forwards the *ARP* reply to the LAN/ATM bridge (see the arc labelled 3 in Fig. 1).
- The LAN/ATM bridge relays the *ARP* reply to the source over the legacy LAN (see the arc labelled 4 in Fig. 1).

- The source now has the MAC address of the destination and begins sending packets via the LAN (see the arc labelled 5 in Fig. 1).
- The bridge relays the packets to the BUS and the BUS forwards the packets to the destination (see the arc labelled 6 in Fig. 1).
- Meanwhile, the *LEC* in LAN/ATM bridge sends an *LE-ARP* request to the LES, asking for the ATM address that corresponds to the destination's MAC address. If the LES does not know the mapping, the LES multicasts the *LE-ARP* request to all LECs. The destination LEC receives the *LE-ARP* request and puts its own ATM address into an *LE-ARP* reply and sends it back to the LES, which then forwards it to the source LEC (see the arc labelled 7 in Fig. 1).
- The source LEC receives the *LE-ARP* reply from the LES (see the arc labelled 8 in Fig. 1), extracts the ATM address and sets up a *Data Direct VCC* between the destination and source. (The details regarding the connection setup procedure will be discussed in a later section.)
- After the *Data Direct VCC* is set up, the packets from the bridge will be transferred via the *Data Direct VCC* instead of the BUS (see the arc labelled 9 in Fig. 1).

Steps 1 to 5 above are analogous to the legacy IP networks, while steps 6 to 9 are ATM specific. One of the complexity of LANE lies in the protocol of the BUS, the BUS forwards not only all the broadcast and multicast frames, but also the unicast frames for the LECs before they establish the *Data Direct VCC* to the destinations. For a particular flow of frames from a source LEC to a destination LEC, it is transmitted via two different VCCs during two periods, i.e., the BUS and the *Data Direct VCC*. This multipath property opens up the possibility for frames to become out of order, since the first frames sent on the *Data Direct VCC* could arrive before the earlier frames sent via the BUS. In order to guarantee the order of the frames in the procedure of channel switch, a flush message protocol is used. When switching from the old path (BUS) to the new path (*Data Direct VCC*), the source LEC first transmits a flush message down the old path, and any further frames to the destination LEC are held or discarded at the source LEC. After the flush message returns from the destination LEC, the source LEC can start using the new path. The channel switch and flush message protocol is an overhead of LANE, however, the BUS forwarding function can reduce the delay significantly, especially when the address resolution delay and connection setup delay are large.

3 The Analytic Model

3.1 Address Resolution and Connection Setup

For a particular flow of frames from a source LEC to a destination LEC, the time it sojourns on the BUS consists of two parts: ATM address resolution delay, and connection setup delay.

3.1.1 ATM address resolution delay

During the procedure of ATM address resolution, if the source LEC knows the ATM address that corresponds to the destination's MAC address, i.e., it has an entry for the mapping in its cache, then it is unnecessary to shake hand with the LES. If the source LEC does not know the mapping, then it sends a request to the LES, asking for the ATM address that corresponds to the destination's MAC address. If the LES has an entry for the mapping in its cache, then it replies, and it needs to exchange two messages in the network for the source LEC to get the mapping. If the LES does not have an entry for the mapping in its cache, then it needs to exchange four messages in the network for the source LEC to get the

```
if (there is no entry for destination's MAC-ATM address mapping in the source LEC cache) then
{
    send ATM-ARP Request to the LES;
    if ( there is an entry for the destination's MAC-ATM address mapping in LES's cache) then
            send ATM-ARP Reply to the source LEC;
    else{
          forward ATM-ARP Request to all the LECs;
          forward ATM-ARP Reply to the source LEC after getting the reply from the destination LEC;
    }
}
establish connection;
```

Figure 2: Address Resolution Procedure for the LANE Version 1.0

mapping. If the source LEC and the destination LEC belong to different ELANs, then it is required to exchange messages between LESs [4]. Fig. 2 shows the procedure of address resolution for LANE.

Therefore, there are several factors determining the address resolution delay: the location of the source and destination, the aging time of the LEC (i.e., the maximum time that the LEC will maintain an entry in its LE-ARP cache in the absence of a verification of that relationship, whose default value is 300 seconds in [3, 4]), and the frequency of the particular source/destination contacts. Up to now, there is no data on the ATM address resolution delay in the references. We assume it to have an exponential distribution with mean $1/\mu$.

3.1.2 ATM connection setup delay

According to the UNI3.1 defined by ATM Forum [6], a set of messages must be exchanged between the source (destination) and ATM switch to establish a VCC. Fig. 3 shows the set of messages exchanged to establish a connection between two ATM endpoints [7].

- The source sends a $SETUP$ message to the network, which contains information that identifies the source and the destination.
- The network node locates the destination and selects a path to the destination using its

topology database. The first switch directly connected to the source sends a *SETUP* message to the next switch on the path to the destination. The latter switch may send a *CALL-PROCEEDING* message to the source asking for more time to process the *SETUP* message. When the *SETUP* message reaches the destination switch, it is delivered to the destination.

- After the destination gets the *SETUP* message, it either sends a *CALL- PROCEEDING* message to the network asking for more time to process the *SETUP* message, or replies with a *CONNECT* message if it accepts the connection.
- When the network gets the *CONNECT* message from the destination, it sends a *CONNECT- ACK* message to the destination, and propagates the *CONNECT* message back until it gets to the source.
- The network sends a *CONNECT* message to the source.
- The source sends a *CONNECT- ACK* message to the network when it gets the *CONNECT* message and decides to accept the connection.

The ATM connection setup delay is the time that elapses from the instant the source sends the *SETUP* message until the instant that it receives the *CONNECT* message. This delay is a complex function of hardware, software, configuration, implementation details, network size, among others. In [7], the test results of signalling in several ATM testbeds (e.g., MAGIC, ATDnet, AII WANs and ATM LAN at Naval Research Laboratory) are reported. Some important characteristics of the setup delay were revealed. The call sequence test, one of the test categories undertaken in [7], which presented a series of calls to the ATM networks after the previous call is accepted or rejected, unveiled the connection setup delay from one point of view. It was found that the different combinations of hardware and software displayed significantly different performance, but isolated ATM switches exhibited

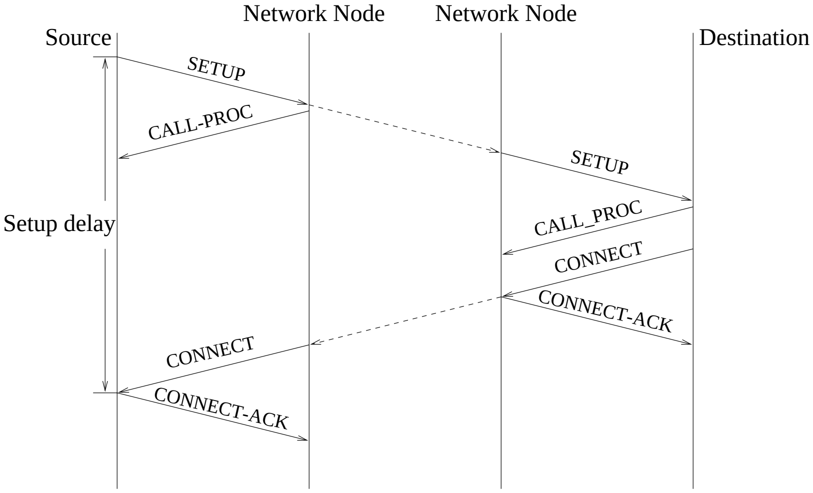


Figure 3: Procedure of Establishing a Connection Between Two ATM Endpoints

generally constant setup delay (tens of milliseconds).

In this paper, we use a 3-stage Erlang distribution with mean $\frac{1}{3w}$ to model the connection setup delay which is approximately a constant value.

3.2 Traffic Model

We assume that the *unicast relay flow* forms a Poisson process with rate λ. In [11], it is found that the session arrivals of FTP, and TCP are Poisson processes. Considering the main application of LANE is Internet-based, e.g., Web browsing, FTP, etc., our assumption for the arrivals of *unicast relay flow* is reasonable. In reference [7], it is shown that the average rate of flow generation is about tens of flows per second in the LAN context.

Recalling the address resolution and connection setup procedure discussed in the preceeding subsection, $N(t)$, the number of *unicast relay flow* flows sojourning on the BUS at time t, may be described as an $M/M/1 - E_3/1$ tandem queueing system (see Fig. 4).

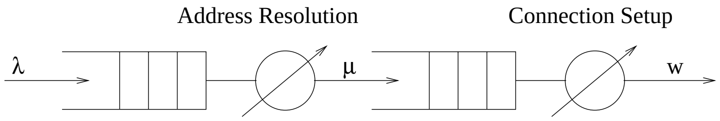


Figure 4: Tandem Queueing System $M/M/1 - E_3/1$

Because the departure process of an $M/M/1$ queueing system is a Poisson process whose rate equals the arrival rate of the system [12], this tandem queueing system consists of two independent queues. Thus, in the steady-state, $N(t)$ can be expressed as the sum of two independent random variables $N_1(t)$, the number of customers in the $M/M/1$ system, and $N_2(t)$, the number of customers in the $M/E_3/1$ system. Therefore, the mean value and generating function of $N(t)$ in steady-state can be easily derived.

Let p_n and $P(z)$ respectively denote the probability mass function and the probability generating function of $N(t)$ in the steady-state, and let $E[N]$ be its mean value, then according to [13] (p.48 and p.167):

$$p_n = Pr[N = n] = \lim_{t\to\infty} Pr[N(t) = n] = \lim_{t\to\infty} Pr[N_1(t) + N_2(t) = n] \tag{1}$$

$$P(z) = \sum_{n=0}^{\infty} p_n z^n = P_1(z)P_2(z) = \frac{(1-\rho_1)(1-\rho_2)(1-z)}{(1-z\rho_1)(1-z(1+r-rz)^3)} \tag{2}$$

$$E[N] = E[\lim_{t\to\infty} N(t)] = \frac{\rho_1}{1-\rho_1} + \frac{\rho_2(3-\rho_2)}{3(1-\rho_2)} \tag{3}$$

where

$$\rho_1 = \frac{\lambda}{\mu}, \rho_2 = \frac{\lambda}{w}, r = \frac{\rho_2}{3} \tag{4}$$

However, in order to get p_n from the generating function, we have to invert the z-transform $P(z)$, whose complexity is dependent on the complexity of the generating function. Moreover, if the tandem queueing system only has finite space to accomodate the customers, the generating function is more complex than Equation (2). In this paper we will circumvent the use of generating function, and construct a stochastic reward net model for the tandem queueing system. With the help of SPNP, a package developed by Duke University researchers, the numerical results on p_n are easily obtained. More details will be discussed in Section 3.3.

For a particular *unicast relay flow*, the traffic characteristics during its lifetime is assumed to form a Markov Modulated Poisson Process (MMPP). MMPP is used extensively to describe the traffic in high-speed networks because of its property of capturing both the time-varying arrival rates and correlations between interarrival times. We only consider the two-state MMPP; our method can be easily extended to multi-state MMPP. The two-state MMPP is illustrated in Fig. 5a, whose frame arrival process is Poisson with rate a_0 while the MMPP is at state 0, and is Poisson with rate a_1 while the MMPP is at state 1. The transitions between these two states form a homogeneous continuous-time Markov chain, with rates α_{01} and α_{10} respectively.

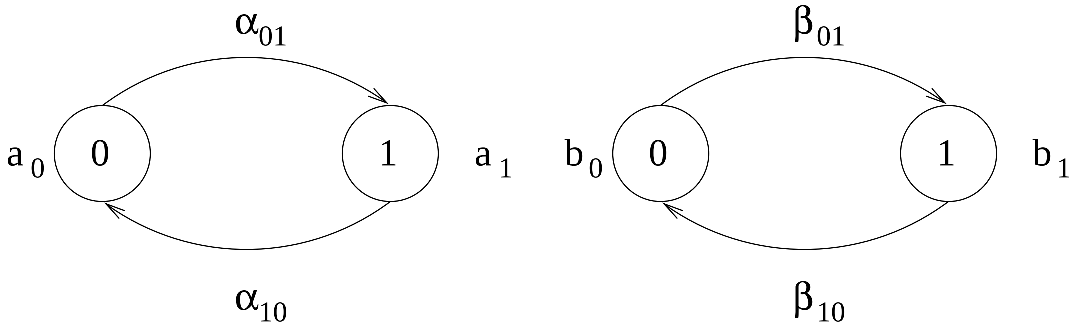
For a particular flow, once the *Data Direct VCC* is set up, the frames of the flow will not enter the BUS, therefore the aggregated *unicast relay flow* is the superposition of $N(t)$ homogeneous MMPPs.

We assume the broadcast and multicast traffic is an MMPP as well, whose four parameters are b_0, b_1, β_{01} and β_{10} as shown in Fig. 5b.

Therefore the total aggregated traffic to the BUS is the superposition of (*N(t)+1*) MMPPs. The mean packet arrival rate to the BUS is then given by:

$$rate_p = E[N]\frac{\alpha_{01}a_1 + \alpha_{10}a_0}{\alpha_{01} + \alpha_{10}} + \frac{\beta_{01}b_1 + \beta_{10}b_0}{\beta_{01} + \beta_{10}} \tag{5}$$

After the frames arrive into the buffer of the BUS, they will be segmented into cells and each cell is serviced with a deterministic service time, i.e., 2.7 microseconds (we assume the BUS is connected to an 155.520 Mb/s port). From the previous measurements of real



(a) Unicast Flow

(b) Broadcast Flow

Figure 5: MMPP Model for the Unicast Flow and Broadcast Flow

networks, it has been observed that the frame sizes of data traffic in Ethernet have three predominant values: 46 bytes (with probability c_1), 144 bytes (with probability c_2) and 1500 bytes (with probability c_3), which corresponds to 2 cells, 4 cells and 32 cells, respectively, after including the LANE overhead (see [3, 14]). It is found that $c_1 = 0.342$, $c_2 = 0.093$, $c_3 = 0.565$.

These three different sizes are the consequence of three different application classes. Short frames are transmitted during terminal-to-host communication, whereas applications based on network-file system protocol (NFS) generate short frames in one direction followed by medium-sized frames in the reverse direction. The maximum frame size in Ethernet traffic is 1512 bytes, used mostly during file transfer applications. In other words, the data frames in the LAN consist of large packets, medium packets and small packets. This observation allows us to analyze and design the buffer more effectively.

The mean cell arrival rate to the BUS is thus given by:

$$rate_c = (E[N]\frac{\alpha_{01}a_1 + \alpha_{10}a_0}{\alpha_{01} + \alpha_{10}} + \frac{\beta_{01}b_1 + \beta_{10}b_0}{\beta_{01} + \beta_{10}})(2c_1 + 4c_2 + 32c_3) \tag{6}$$

3.3 Stochastic Reward Net Models

3.3.1 A Brief Description of Stochastic Reward Net

Stochastic reward nets are extensions of stochastic Petri nets. We give a brief overview of these constructs before presenting the model of the BUS.

A Petri net is a directed graph with two disjoint types of nodes: *places* and *transitions.* A directed arc connecting a place (transition) to a transition (place) is called an input (output) *arc* of the transition. A positive integer called multiplicity can be associated with each arc. Places connected to a transition by input arcs are called the input places of this transition, and those connected by means of output arcs are called the output places. Each place may contain zero or more tokens. A transition is *enabled* if each of its input places has at least as many tokens as the multiplicity of the corresponding input arc. A transition can *fire* when it is enabled, and upon firing, a number of tokens equal to the multiplicity of the input arc is removed from each of input places, and a number of tokens equal to the multiplicity of the output arc is deposited in each of its output places. The state of a Petri net is characterized by a number of tokens in each place which is called its marking. The initial number of tokens assigned to the places in a Petri net determines the initial state (marking) of the net.

Stochastic Petri nets are Petri nets where exponentially distributed firing time is attached to each transition. In Generalized Stochastic Petri nets (GSPN), transitions are allowed to be either *timed* (exponentially distributed firing time) or *immediate* (zero firing time). A marking of a GSPN is called *vanishing* if at least one immediate transition is enabled in the marking and *tangible* otherwise. GSPN also introduces *inhibitor arc* connecting a place to a transition. A transition with an inhibitor arc can not fire if the input place of the inhibitor arc contains more tokens than the multiplicity of the arc.

Table 1: Rates and Guards for the Transitions in Fig. 6

transition	T_{ARV1}	T_{AR}	T_{CS}	$T_{ARV2}(i)$	T_{broad}

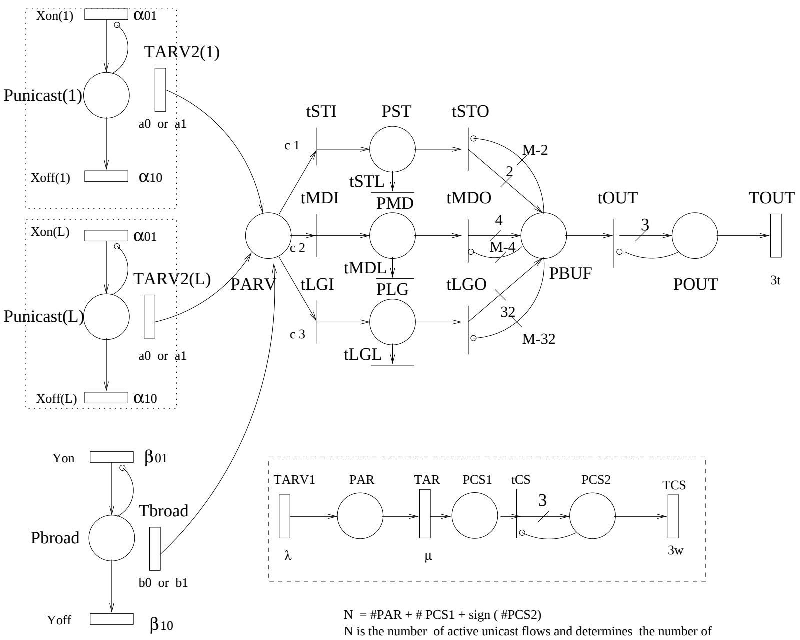
rate	λ	μ	3w	if $(\#P_{unicast}(i) = 1)$ a_1
				else a_0
					if $(\#P_{broad} = 1)$ b_1
					else b_0

transition	T_{OUT}	$X_{on}(i)$	$X_{off}(i)$	Y_{on}	Y_{off}
rate	3t	α_{01}	α_{10}	β_{01}	β_{10}
transition	$T_{ARV2}(i)$	T_{ARV1}	t_{STL}	t_{MDL}	t_{LGL}
guard	$i < \#P_{AR} + \#P_{CS1}$ $+sign(\#P_{CS2})$	$\#\ P_{AR} + \#P_{CS1}$ $+sign(\#P_{CS2}) < L$	$\#\ P_{BUF}$ $> M - 2$	$\#\ P_{BUF}$ $> M - 4$	$\#\ P_{BUF}$ $> M - 32$

Stochastic reward nets (SRNs) are based on GSPN but extend them further [19]. In SRN, every tangible marking can be associated with a reward rate. It can be shown that an SRN can be mapped into a Markov reward model. Therefore, by using software packages [20], the underlying Markov reward model of an SRN can be automatically generated and solved. Thus a variety of performance measures can be specified and calculated using a very convenient formalism. SRN also allows several other features that makes specification convenient: (1) each transition may have an enabling function (also called a guard) so that a transition is enabled only if its (marking dependent) enabling function is true, (2) marking dependent arc multiplicities are allowed, (3) transitions can also have priorities. To represent an 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 a small hollow circle instead of arrows at their terminating ends. The current number of tokens in place p is denoted as $\#p$.

3.3.2 Accurate SRN Model for the BUS

The stochastic reward net in Fig. 6 illustrates the model we described in the preceding subsection. The subnet in the dashed rectangle in Fig. 6 is for the $M/M/1-E_3/1$ tandem



N = #PAR + # PCS1 + sign (#PCS2)
N is the number of active unicast flows and determines the number of components in the dotted rectangle at the left side of the figure.

Figure 6: Accurate SRN Model for the BUS

queueing system. The sum of the number of tokens in P_{AR} and P_{CS1} is limited ($\leq L$), which is implemented by the guard on transition T_{ARV1}. The number of customers in the system, $N(t)(\leq L)$, determinates the characteristics of the traffic to the BUS. We use an array of subnet (1 to L, see the subnet in the dotted rectangle) to describe L number of MMPPs. The rate of transition $T_{ARV2(i)}$ $(i = 1, ..., L)$ is determined by the number of token in place $P_{unicast(i)}$, and is enabled only when there are more than i customers in the $M/M/1 - E_3/1$ tandem queueing system. In other words, we enable transitions $T_{ARV2(i)}$ $(i = 1, ...N)$ when there are N customers in the $M/M/1 - E_3/1$ tandem queueing system.

Besides the unicast traffic, there is a flow of broadcast traffic which is described by an MMPP as well. The frames generated according to the MMPP have three possible sizes, 2 cells (with probability c_1), 4 cells (with probability c_2), and 32 cells (with probability c_3). The split between the large, medium and small frame is represented by the immediate transitions t_{STI}, t_{MDI}, and t_{LGI}. When a packet finds there is enough space in the buffer whose capacity is M cells, the packet enters the buffer, which is represented by immediate transitions t_{STO}, t_{MDO}, and t_{LGO}. If the packet finds that there is not enough space in the buffer to accomodate the entire packet, the immediate transitions t_{STO}, t_{MDO}, and t_{LGO} are inhibited by inhibitor arcs from P_{BUFF} to t_{STO}, t_{MDO}, and t_{LGO}. Then the entire packet will be discarded, which is represented by immediate transitions t_{STL}, t_{MDL}, and t_{LGL}. In this paper, discard is frame-based, not cell-based. Therefore, partial packet loss will not exist in our study, and loss amplification caused by partial packet loss is prevented [23]. The deterministic service time of the BUS is approximated with a 3-stage Erlang distribution.

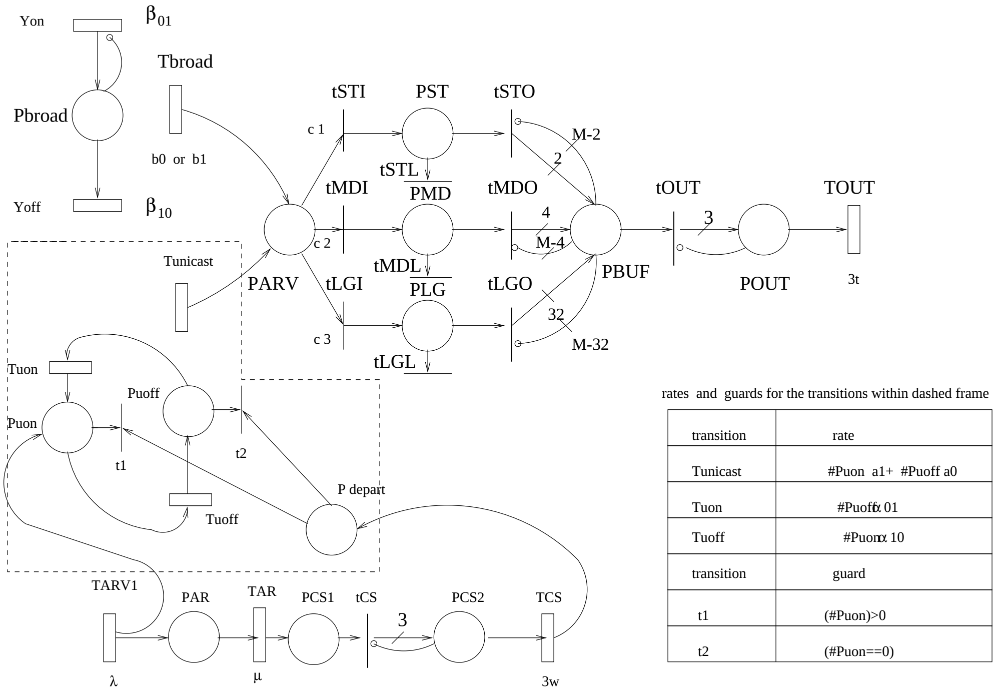
The guards and marking dependent firing rates for the transitions are listed in Table 1.

From the former description, the stochastic reward net in Fig. 6 captures the probabilistic behavior of the BUS over all three time scales. The $M/M/1 - E_3/1$ tandem queueing system reflects the connection level property of the ELAN. The frame arrival is described by an MMPP, and represents the bursty level property. The transmission of cells is approximated by a 3-stage Erlang distribution, and belongs to cell level. These processes operate on widely different time scales, and the transition rates of the Markov chain differ widely, e.g., the cells may be transmitted within 3 microseconds if the BUS is connected to an 155Mb/s ATM switching port, but the flow interarrival interval is about tens of milliseconds. The wide difference in the transition rates will cause numerical difficulties while solving the Markov chain, which is called stiffness [16, 17].

Besides the stiffness, the model in Fig. 6 is plagued by the largeness of state space, e.g., if the $M/M/1-E_3/1$ tandem queueing system can accomodate L customers, and the queue in the BUS can buffer M cells, then the number of states in the model illustrated in Fig. 6 will be $O(2^L L^2 M)$. It is practically impossible to give numerical solution to this huge Markov chain when L and M are large. In this paper, we present three approximate solutions to the SRN in Fig. 6.

3.3.3 Approximate Solution 1 : Birth-Death Process

Because we assume L flows of unicast traffic are homogeneous, their aggregation may be approximated by a birth-death process, which is represented by the subnet in the dashed frame in Fig. 7. When a customer enters the $M/M/1-E_3/1$ tandem queueing system, one token will be deposited in place P_{uon}. The rate of transitions T_{uon} and T_{uoff} depends on the number of tokens in P_{uoff} and P_{uon} respectively. When a customer leaves the $M/M/1-E_3/1$ tandem queueing system, transition T_{CS} fires and deposits one token in place P_{depart}. The guards of immediate transitions t_1 and t_2 guarantee one token is removed from either P_{uoff} or P_{uon}. The number of states in the model illustrated in Fig. 7 is $O(L^4 M)$, which is rather



rates and guards for the transitions within dashed frame

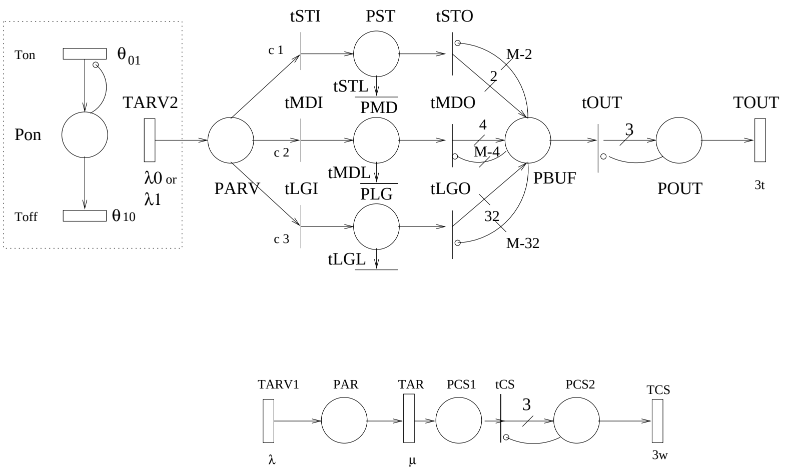
transition	rate
Tunicast	#Puon a1+ #Puoff a0
Tuon	#Puoffα 01
Tuoff	#Puonα 10
transition	guard
t1	(#Puon)>0
t2	(#Puon==0)

Figure 7: An Approximate SRN Model for the BUS: Birth-Death Process

less than that of the accurate model. But its underlying Markov chain is still huge.

3.3.4 Approximate Solution 2: MMPP

According to [14, 15], the superposition of multiple MMPPs is also an MMPP. However, as the number of component processes increases, the number of states of the superposed process increases exponentially, moreover, in this paper the number of superposed MMPP (i.e.,*N(t)+1*) is random. To reduce the complexity of solving queues with a large number of arrival streams, the superposed process may be approximated by a simpler process that captures the important characteristics of the original process as closely as possible. The simplest model that has the potential to approximate the superposition of multiple MMPPs



N = #PAR + # PCS1 + sign (#PCS2) is the number of active unicast flows

N determines the parameters of the subnet in the dotted rectangle

Figure 8: An Approximate SRN Model for the BUS: MMPP

is a two-state MMPP. Assume the parameters of the MMPP approximating the superposition of multiple MMPPs are λ_0, λ_1, θ_{01}, and θ_{10}. In [14, 15], authors give a method to obtain these four parameters from the metrics of the superposed processes. The method is given in Appendix. The number of *unicast relay flow* is $\#PAR + \#PCS1 + sign(\#PCS2)$. Substituting this value to the S in the Appendix, we can obtain the four parameters of the MMPP approximating the aggregated flow. This approach is illustrated in Fig. 8, and still plagued by the largeness of state space. The number of states in the system illustrated in Fig. 8 will be $O(L^2M)$ which is less than that of the Birth-Death process approximation. But for a practical ATM switch whose buffer capacity of one port is about 2000 cells, and whose users can tolerate tens of seconds delay for connection setup (i.e., the $M/M/1-E_3/1$ tandem queueing system can accomodate thousands of customers), the number of states will still exceed one billion.

3.3.5 Approximate Solution 3: Hierachical Decompsition

The third approximate solution is based on hierarchical decomposition [18]. We notice that the events of flow arrival and departure to the $M/M/1-E_3/1$ tandem queueing system are rather rare compared with the occurrences of cell transmission. Consequently, we can assume that the buffer for cell transmission attains a (quasi-)steady state with respect to $M/M/1-E_3/1$ tandem queueing system between successive occurrences of flow arrival and departure events. Thus, we can compute the performance measures for the buffer in each of these (quasi-)steady states. The overall system can then be characterized by weighting these quasi-steady state performance measures by the structure state probabilities.

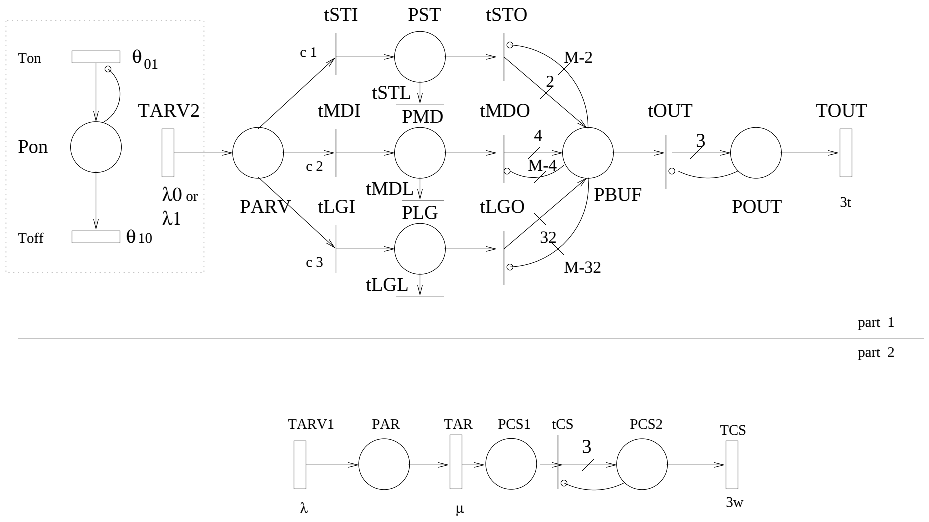
We are interested in two performance indices: the packet loss ratio P_{loss}, and buffer occupancy $P_B(i)$ (i.e., the probability that there are i cells in the buffer). Then, according to the principle of hierarchical decomposition, we have

$$P_{loss} = \sum_{n=0}^{\infty} P_{loss}(n) p_n \tag{7}$$

$$P_B(i) = \sum_{n=0}^{\infty} P_B(n,i) p_n \tag{8}$$

where $P_{loss}(n)$ and $P_B(n,i)$ are the packet loss ratio and buffer occupancy under the condition that there are n customers in the $M/M/1-E_3/1$ tandem queueing system. The SRN in Fig. 9 can be decomposed into two parts (i.e., part 1 and part 2) and solved separately. Through solving part 1, we can obtain P_n. Through solving part 2, we can obtain $P_{loss}(n)$ and $P_B(n,i)$. Then the results can be composed using Equations (7) and (8). The number of states of the separate stochastic reward nets are $O(L^2)$ and $O(M)$ respectively after decomposition. The hierarchical model can at once avoid the stiffness and largeness.

Now we show how to get $P_{loss}(n)$ and $P_B(n,i)$. If we denote packet loss ratio of large packet, medium packet and small packet when there are n customers in the $M/M/1-E_3/1$



N = #PAR + # PCS1 + sign (#PCS2) is the number of active unicast flows , determines the parameters of MMPP in the dotted rectangle

divide the SRN into part 1 and part 2, solve part 1 and part 2 seperately , then use Equation (7) and (8),

Figure 9: An Approximate SRN Model for the BUS: Hierachical Decompsition

tandem queueing system as $P_{lloss}(n)$, P_{mloss}(n) and $P_{sloss}(n)$ respectively. Then

$$P_{loss}(n) = c_3 P_{lloss}(n) + c_2 P_{mloss}(n) + c_1 P_{sloss}(n) \tag{9}$$

We can define *reward function* , which is a powerful function in SPNP [20] to get $P_{lloss}(n)$, $P_{mloss}(n)$, $P_{sloss}(n)$ and $P_B(n, i)$. For example, for the $P_{lloss}(n)$, we define

```
double lloss() {
        if (mark("PBUF") > M - 32){
            if(mark("Pon"))  return($\lambda$1);
            else    return($\lambda$0);
        }
        else    return(0);
}
```

According to [24], we have

$$\begin{aligned} P_{lloss}(n) &= Pr[\ queue\ length\ exceeds\ M-32 \mid a\ large\ packet\ arrives] \\ &= \frac{lloss()(\theta_{01}+\theta_{10})}{c_3(\theta_{01}\lambda_1+\theta_{10}\lambda_0)} \end{aligned} \tag{10}$$

Note that θ_{01}, θ_{10}, λ_1 and λ_0 are functions of n (see Appendix).

4 Numerical Results and Discussions

In this paper, we assume the data output rate of the BUS is 155 Mb/s, and select the time unit (TU) as 10ms (millisecond). Then for the cell transmission, the rate of an Erlangian stage is about $10000/TU$.

In the specification of LANE [3, 4], the average rate of the unicast traffic flow to the BUS is regulated via two parameters, C_{10} and C_{11}, i.e., for a particular flow, the BUS does not transmit more than C_{10} frames during a period of duration C_{11}. The minimal, default, and maximal values of C_{10} and C_{11} are (1,1,10) frames and (1,1,60) seconds in LANE Version 1, and (1,10,20) frames and (1,1,60) seconds in LANE Version 2. Therefore the average rate of MMPP of a particular unicast flow is about $0.2/TU$ if we select C_{10} as 20 and C_{11} as 1. C_{10} and C_{11} can regulate the burstiness of the frame as well. The peak rate duration of the unicast flow will not exceed $\frac{C_{10}}{a_1}$ with C_{10}/C_{11} regulation applied.

From the previous observations on the network traffic [10], the flow generation interval is about tens of milliseconds, which corresponds to about 0.1 to $1/TU$. From the measurements reported in [7], the connection setup delay is about tens of milliseconds, which corresponds to the service rate about 0.1 to $1/TU$.

For the address resolution delay, in order to maintain the steady state of the $M/M/1-E_3/1$ tandem queueing system, we assume its service rate is within the same scale of the

Table 2: Computational Complexity Comparison for the Accurate and Approximate Solutions

	Accurate Model	Birth-Death Process	MMPP	Hierarchical Decomposition
Number of State	$16 * 2^{(L+1)} L^2 * M$	$64L^4M$	$32L^2M$	$8M$ for part 1, $4L^2$ for part 2 solve part 1 L times

Table 3: Parameters for Section 4.1

$\alpha_{01} = 0.01$	$\alpha_{10} = 0.2$	$a_0 = 2$	$a_1 = 4$	$\beta_{01} = 0.01$	$\beta_{10} = 0.01$	$b_0 = 10$	$b_1 = 50$
$c_1 = 0.342$	$c_2 = 0.093$	$c_3 = 0.565$	$M = 200$	$\lambda = 0.5$	$\mu = 0.6$	$w = 1$	$L = 10$

flow generation rate, i.e., about 0.1 to $1/TU$.

For the broadcast traffic, we assume the peak packet rate (b_1) is $50/TU$, (i.e., 5000 packets per seconds, and corresponds to 60 Mb/s if the packet size is 1500 bytes), the lower rate (b_0) is 10 (i.e., corresponds to 12 Mb/s), and $\beta_{01} = \beta_{10} = 0.01$.

4.1 The Comparison of Four Models

Table 2 compares the number of states of the accurate approach in Fig. 6 and the three approximate solutions in Fig. 7 to 9. The hierachical decomposition can be used to study the performance of BUS with realistic system parameters, and the other three can only be used to study the toy system.

Next we show the accuracy of these three approximations. The parameters for the system are given in Table 3. The results are given in Fig. 10. It can be seen that the results obtained from approximate solutions 1 and 2 are very close to the accurate model (we can not discriminate them by eyes). The results from hierarchical decomposition is close to the accurate model. Considering the trade-off between the accuracy and tractability of the model, we choose hierarchical decomposition to study the performance of the BUS.

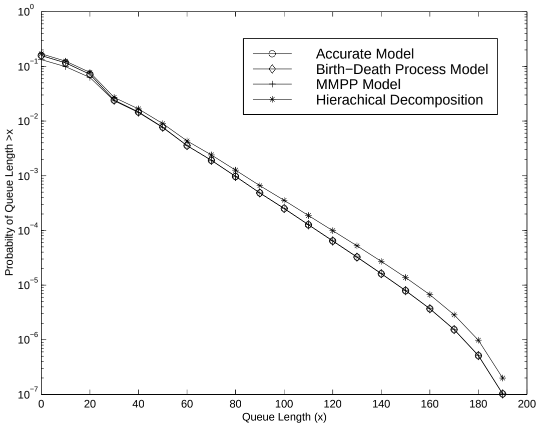


Figure 10: Buffer Occupancy Probability (Comparison of Different Solutions)

Table 4: Parameters for Section 4.2

$\alpha_{01} = 0.01$	$\alpha_{10} = 0.2$	$a_0 = 0$	$a_1 = 4$	$\beta_{01} = 0.01$	$\beta_{10} = 0.01$
$b_0 = 10$	$b_1 = 50$	$c_1 = 0.342$	$c_2 = 0.093$	$c_3 = 0.565$	$M = 2000$

4.2 The Results From The Hierarchical Decomposition

Next we will pay our attention to the hierarchical decompostion method discussed in section 3.3.

Fig. 11 illustrates the buffer occupancy probability with $\lambda = 0.499$, $\mu = 0.5$, $w = 0.5$. The other parameters are given in Table 4. The packet loss ratio is given in Table 5. As we decrease the load to the tandem queueing system, the packet loss ratio will decrease. Fig. 12 plots the buffer occupancy probability as we decrease the flow arrival rate from 0.499 to 0.498, i.e., $\lambda = 0.498$, $\mu = 0.5$, $w = 0.5$ and the other parameters are given in Table 4. The packet loss ratio is also given in Table 5.

In the previous studies, we assumed $c_1 = 0.342$, $c_2 = 0.093$, $c_3 = 0.565$, which were obtained by observations over a long period followed by statistical estimation [14]. However, there may exist some cases where FTP or TELNET dominates the applications during some periods, which will make c_1, c_2, and c_3 change. Fig. 13 plots the buffer occupancy probability with $\lambda = 0.498$, $\mu = 0.5$, $w = 0.5$, $c_1 = 0.142$, $c_2 = 0.093$, $c_3 = 0.765$, which corresponds to FTP-dominated applications. Fig. 14 gives the buffer occupancy probability with $\lambda = 0.498$, $\mu = 0.5$, $w = 0.5$, $c_1 = 0.742$, $c_2 = 0.093$, $c_3 = 0.165$, which corresponds to TELNET-dominated applications. Obviously, the Quality of Service changes with workload. The other parameters for Fig. 13 and Fig. 14 are the same as those in Table 4 excluding c_1, c_2 and c_3.

There are several ways to decrease the packet loss ratio of the BUS. We may increase the service rate of the tandem queueing system which reduces the address resolution delay and connection setup delay. Fig. 15 plots the buffer occupancy probability with $\lambda = 0.5$, $\mu = 0.6$, $w = 1$, $c_1 = 0.142$, $c_2 = 0.093$, $c_3 = 0.765$. The other parameters are given in Table 4 excluding c_1, c_2 and c_3. Obviously, after decreasing the address resolution delay and connection setup delay, the BUS performs well even in FTP applications. From[7], the Fore Systems' switch ASX-200BX running signalling software version 4.1.0 will introduce 10 milliseconds connection setup delay, and ASX-200WG running signalling software version 4.0.1 will introduce 20 milliseconds connection setup delay. From our studies, the former switch will perform well in the traffic enviornment presented in the paper, while the packet loss ratio of the latter one is unacceptably high especially for large packet as the flow generating rate approaches 0.5 (i.e., 50 flow/second, a real enviornment as discussed in [10]). For the switches with connection setup delay exceeding 20 milliseconds, one possible solution to decrease packet loss ratio is to regulate the load of *unicast relay flows* to the BUS, via C_{10} and C_{11}. We may set $C_{10} = 1$ and $C_{11} = 1$, which means only one packet is permitted to transmit via the BUS during one second for a particular flow. At this time, the mean packet rate is 0.01, and the packet arrival is less bursty and can be approximated with a Poisson process. The parameters of the *unicast relay flows* are changed to $a_0 = 0.01$ and $a_1 = 0.01$. Fig. 16 plots the buffer occupancy probability with $c_1 = 0.142$, $c_2 = 0.093$, $c_3 = 0.765$, $C_{10} = 1$ and $C_{11} = 1$, and 1000 customers persisting in the tandem queueing system. The other parameters are given in Table 4 excluding c_1, c_2 and c_3. It can be seen that regulating the parameters C_{10} and C_{11} can decrease the packet loss ratio effectively even under extreme enviornment (i.e., 1000 persistent flows and FTP applications). However, we do not recommend this method unless the connection setup delay for a switch is unacceptably large. The rationale behind the recommendation is that LANE developed an deliberate protocol to use the BUS relaying unicast traffic in order to reduce the packet transfer delay before establishes the *Data Direct VCC*, we would like to relay the unicast packets as more as possible if congestion does not occur. Of course, there is a tradeoff between the congestion and packet transfer delay. Once the statistics of the address resolution delay and connection setup delay for an ELAN are obtained through test and analysis, the appropriate choice for the parameters C_{10} and C_{11} may be made using the analysis method proposed in this paper.

The analysis method proposed in this paper can be extended to study the performance of MPOA (Multi Protocol Over ATM) and IP Switching, which forwards the packets of a flow via a default path at first, and then adopts a cut-through path after it recognizes that it can benefit from a cut-through path [21, 22]. Although MPOA and IP Switching are rather different from LANE in protocol infrastructure, path switching from the old shared VCC to the new private VCC is one of their common features, which can be analyzed with the method in this paper.

Table 5: Packet Loss Ratio for Fig.11-16

	loss ratio of short packets	loss ratio of medium packets	loss ratio of large packets
Fig.11	1.17732477370066e-06	4.12154398684745e-06	2.30070042753832e-04
Fig.12	5.09700199511697e-13	1.80481827141803e-12	1.17974946529332e-10
Fig.13	1.20074629322922e-06	4.211601590878e-06	2.37020261944003e-04
Fig.14	1.26792546745493e-33	4.19793881001496e-33	1.74705985795902e-31
Fig.15	1.55587916533906e-33	5.02721843671009e-33	2.00009415749799e-31
Fig.16	1.48915379344e-33	5.34859088397e-33	2.17129337889e-31

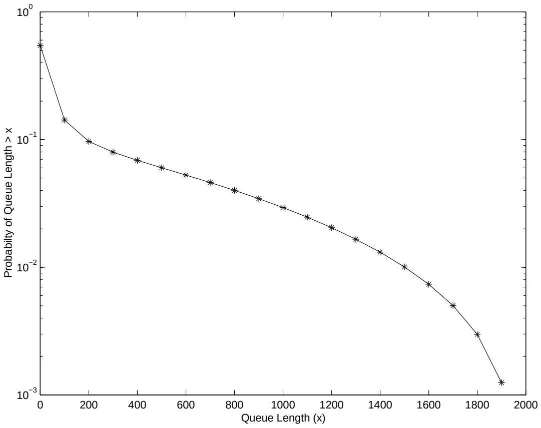


Figure 11: Buffer Occupancy Probability, $\lambda = 0.499$, $\mu = 0.5$, $w = 0.5$

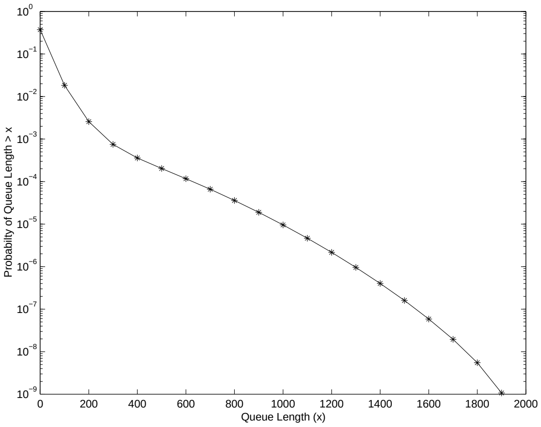


Figure 12: Buffer Occupancy Probability, $\lambda = 0.498$, $\mu = 0.5$, $w = 0.5$

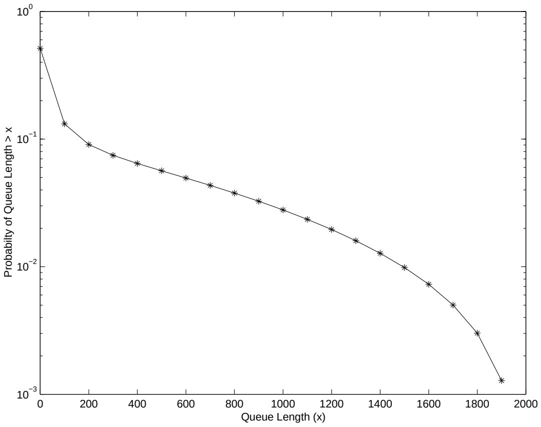


Figure 13: Buffer Occupancy Probability, $\lambda = 0.498$, $\mu = 0.5$, $w = 0.5$, $c_1 = 0.142$, $c_2 = 0.093$, $c_3 = 0.765$

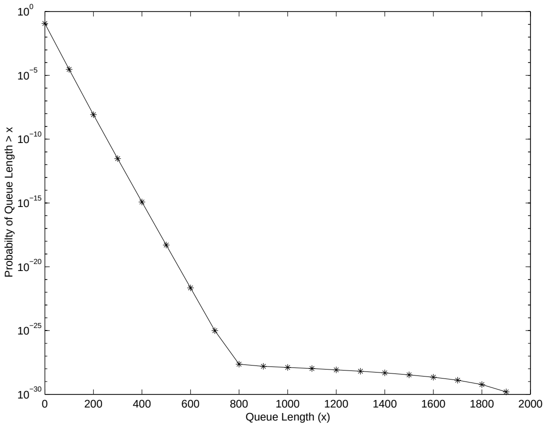


Figure 14: Buffer Occupancy Probability, $\lambda = 0.498$, $\mu = 0.5$, $w = 0.5$, $c_1 = 0.742$, $c_2 = 0.093$, $c_3 = 0.165$

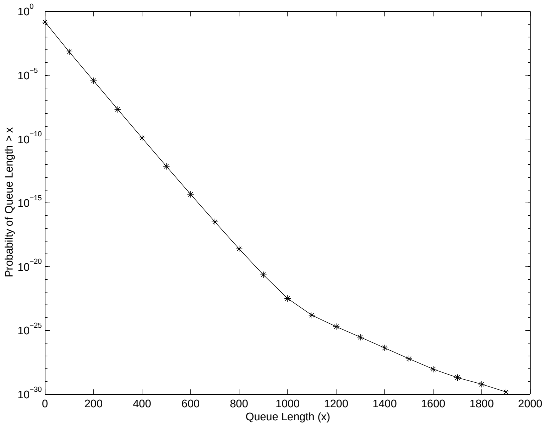


Figure 15: Buffer Occupancy Probability, $\lambda = 0.5$, $\mu = 0.6$, $w = 1$, $c_1 = 0.142$, $c_2 = 0.093$, $c_3 = 0.765$

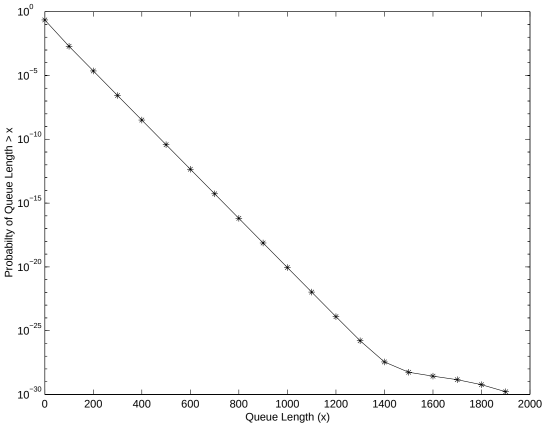


Figure 16: Buffer Occupancy Probability, $c_1 = 0.142$, $c_2 = 0.093$, $c_3 = 0.765$, $C_{10} = 1$ and $C_{11} = 1$, and 1000 Persistent Customers

5 Conclusions

In this paper, we developed a stochastic reward net model to analyze the performance of the BUS in ATM LANE. The number of active *unicast relay flows* sojourning on the BUS is determinated by a tandem queueing system $M/M/1/-E_3/1$, where the flow arrival process is Poisson, the address resolution delay is exponentially distributed, and the connection setup delay is 3-stage Erlang distributed. The broadcast and multicast traffic is assumed to form a Markov Modulated Poisson Process (MMPP). The traffic for a particular *unicast relay flow* is an MMPP as well. The size of data frames in traffic flows is a random variable which has three possibilities, i.e., short, medium, and long. In order to solve the large and stiff Markov chain, a hierarchical decomposition method is adopted. The numerical results obtained by using Stochastic Petri Net Package show the influence of the connection-level statistics on the performance of the BUS and reducing the address resolution delay and connection setup delay, or controlling the packet rate entering the BUS via regulating the parameters C_{10} and C_{11}, will decrease the packet loss ratio effectively. From intuition and the analysis in this paper, the BUS in an ELAN is the double-edged sword, which can reduce the packet transfer delay before the direct VCC connection is established between the source and destination. On the other hand, the overuse of the BUS to relay unicast flows will eventually result in congestion. The analysis method in this paper may be used to utilize and control the BUS effectively.

Appendix

Given a unicast MMPP with parameters a_0, a_1, α_{01}, α_{10}, the mean m_1, variance v_1, third moment u_1 and third central moment u_1' of its arrival rate are given by:

$$m_1 = \frac{\alpha_{01}a_1 + \alpha_{10}a_0}{\alpha_{01} + \alpha_{10}} \tag{11}$$

$$v_1 = \frac{(a_1 - a_0)^2\alpha_{01}\alpha_{10}}{(\alpha_{01} + \alpha_{10})^2} \tag{12}$$

$$u_1 = \frac{a_1^3\alpha_{01} + a_0^3\alpha_{10}}{\alpha_{01} + \alpha_{10}} \tag{13}$$

$$u_1' = u_1 - 3m_1v_1 - m_1^3 \tag{14}$$

respectively, and the time constant τ_1 for the correlation in the arrival stream is given by

$$\tau_1 = \frac{1}{\alpha_{01} + \alpha_{10}} \tag{15}$$

For a given broadcast MMPP with parameters α_0, α_1, β_{01}, β_{10}, the mean, variance, third moment and third central moment of its arrival rate, and the time constant for the correlation of the arrival process are

$$m_2 = \frac{\beta_{01}b_1 + \beta_{10}b_0}{\beta_{01} + \beta_{10}} \tag{16}$$

$$v_2 = \frac{(b_1 - b_0)^2\beta_{01}\beta_{10}}{(\beta_{01} + \beta_{10})^2} \tag{17}$$

$$u_2 = \frac{b_1^3\beta_{01} + b_0^3\beta_{10}}{\beta_{01} + \beta_{10}} \tag{18}$$

$$u_2' = u_2 - 3m_2v_2 - m_2^3 \tag{19}$$

$$\tau_2 = \frac{1}{\beta_{01} + \beta_{10}} \tag{20}$$

Then for the superpostion of S unicast MMPPs and one broadcast flow, its mean, variance, and third central moment of the arrival rate, and the time constant for the correlation of the arrival stream are

$$m = Sm_1 + m_2 \tag{21}$$

$$v = Sv_1 + v_2 \tag{22}$$

$$u' = Su'_1 + u'_2 \tag{23}$$

$$\tau = \frac{Sv_1\tau_1 + v_2\tau_2}{N(t)v_1 + v_2} \tag{24}$$

respectively.

The four parameters describing the superposition of the MMPPs, λ_0, λ_1, θ_{01}, and θ_{10} are given by

$$\lambda_0 = m + \sqrt{\frac{v}{\eta}} \tag{25}$$

$$\lambda_1 = m - \sqrt{v\eta} \tag{26}$$

$$\theta_{01} = \frac{1}{\tau(1+\eta)} \tag{27}$$

$$\theta_{10} = \frac{\eta}{\tau(1+\eta)} \tag{28}$$

where

$$\eta = 1 + \frac{\delta(\delta - \sqrt{(4+\delta^2)})}{2} \tag{29}$$

$$\delta = \frac{u'}{v^{3/2}} \tag{30}$$

References

[1] Anthony Alles, “ATM Internetworking,” *Cisco Systems*, Inc. May 1995.

[2] Hairong Sun, et al., “Supporting IP on the ATM Networks : An Overview,” *Computer Communications*, p1020-1029, Aug. 1998, Elsevier Science.

[3] “LAN Emulation over ATM Specification Version 1.0,” *ATM Forum, af-lane-0021.000*, Jan. 1995.

[4] “LANE V2.0 LUNI Interface,” *ATM Forum, af-lane-0084.000* , July 1997.

[5] “ATM Update II: An Anixter Technology White Paper,” *http://www.anixter.com /techlib /whiteppr /network /1757750.htm.*

[6] “ATM User-Network Interface Specification V3.1,”*ATM Forum, af-uni-0010.002.*

[7] Douglas Niehaus, et al., “Performance Benchmarking of Signalling in ATM Networks,” *IEEE Communication Magazine*, p134-14, Aug.1997.

[8] Riku Kreula, Harri Haapasalo, “Transfer Delay at ATM LAN Emulation and Classical IP over ATM,” p478-482, *ICC'1997.*

[9] T. Driscoll, Daniel, et al., “Performance Comparison Between ATM LAN Emulation, Classical IP over ATM, and Native ATM in a Multiplatform Multioperating System Environment,” p434-438, *1997 MILCOM Conference.*

[10] Stevin Lin, Nick McKeown, “A Simulation Study of IP Switching,” *SIGCOM'1997.*

[11] V. Paxson, S. Floyd, “Wide Area Traffic: The Failure of Poisson Modeling,” *IEEE/ACM Transactions on Networking*, 3(3), p226-244, 1995.

[12] L. Kleinrock, “Queueing Systems, Vol.I: Theory,” Wiley, NY, 1976.

[13] Donald Gross, Carl M. Harris, “Fundamentals of Queueing Theory,” John Wiley & Sons, 1974.

[14] Raif O. Onvural, “Asynchronous Transfer Mode Networks: Performance Issues,” Artech House, 1994.

[15] H. Heffes, “A Class of Data Traffic Processes - Covariance Function Characterization and Related Queueing Results,”*BSTJ*, 59(6), p897-929, 1980.

[16] A. Bobbio, K. S. Trivedi, “ An Aggregation Technique for the Transient Analysis of Stiff Markov Chain,” *IEEE Trans. Comput.*, C-35(9),P803-814,1986.

[17] M. Malhotra, J. K. Muppala, K. S. Trivedi, “Stiffness-tolerant Methods for Transient Analysis of Stiff Markov Chains,”*Microelectron.Reliab.*, 34(11), p1825-1841, 1994.

[18] J. K. Muppala, K. S. Trivedi, “Composite Performance and Availability Analysis using a Hierarchy of Stochastic Reward Nets,” *Proc. of the Fifth International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Torino, 1991.

[19] G. Ciardo, J. Muppala, K. S. Trivedi, ”Analyzing Concurrent and Fault-tolerant Software Using Stochastic Reward Nets,” *Journal of Parallel and Distributed Computing*, 15, p255-269, 1992.

[20] G. Ciardo, J. Muppala and K. Trivedi. “SPNP: Stochastic Petri Net Package,” *International Conference on Petri Nets and Performance Models*, Kyoto, Japan, December 1989.

[21] “Multi-Protocol Over ATM Specification v1.0,” *ATM Forum, af-mpoa-0087.000*, July, 1997.

[22] P. Newman, T. Loyn, G. Minshall, “Flow Labelled IP: A Connectionless Approach to ATM,” p1251-1260, *INFOCOM'96*.

[23] A. Romanow, S. Floyd, ” Dynamic of TCP Traffic over ATM Networks,” *IEEE Journal on Selected Areas in Communications*,13(4), p 633-641, May 1995.

[24] C. Y. Wang,D. Logothetis,K. S. Trivedi,I. Viniotis, “Transient Behavior of ATM Networks Under Overloads”,*INFOCOM'96*,p978-985.