Distrib. Syst. Engng **3** (1996) 53–67. Printed in the UK

Design and evaluation of a connection management mechanism for an ATM-based connectionless service

Geert Heijenk† and Boudewijn R Haverkort‡

Centre for Telematics and Information Technology, University of Twente, PO Box 217, 7500 AE, Enschede, The Netherlands

Abstract. The Asynchronous Transfer Mode (ATM) has been developed as a connection-oriented technique for the transfer of fixed-size cells over high-speed networks. Many applications, however, require a connectionless network service. In order to provide such a technique, one can built a connectionless service on top of the connection-oriented service. In doing so, the issue of connection management comes into play. In this paper we propose a new connection management mechanism that provides for low bandwidth usage (as compared to a permanent connection) and low delays (as compared to a connection-per-packet approach). We model the new mechanism under two workload scenarios: an ordinary Poisson process and an interrupted Poisson process. We use Markovian techniques as well as matrix-geometric methods to evaluate the new connection management mechanism. From the evaluations it turns out that the proposed mechanism is superior to older approaches (which can be seen as limiting cases).

1. Introduction

The Asynchronous Transfer Mode (ATM) is becoming increasingly important in telecommunication networks. ATM has been developed as a *connection-oriented* technique, which implies that a connection should be established prior to any information transfer. In order to be able to support applications with a *connectionless* nature such as electronic mail and information retrieval, there is a need to provide a connectionless service with networks based on ATM [1]. Furthermore, a connectionless service would be very suitable for the interconnection of the installed base of basically connectionless Local Area Networks (LANs).

One of the key functions in providing a connectionless service using ATM is connection management. This function is responsible for establishing and releasing ATM connections in such a way that (connectionless) packets can be transferred from their source to the proper destination using these connections. A problem appears in the fact that the connection management function has no advance knowledge about the offered traffic that has to be transferred, because of the connectionless nature of this traffic. It should, however, make an agreement with the signalling system of the ATM network about the characteristics of the reserved bandwidth for the established connections (e.g. mean and peak bandwidth).

In this paper we present and evaluate the performance of a connection management mechanism called On-demand Connection with Delayed Release (OCDR). The OCDR mechanism establishes a connection between two nodes in an ATM network as soon as a packet needs to be transferred between these nodes. After the packet has been transferred, the connection is not immediately released. It can be used for subsequent packets for the same destination as well. Only if no packet is transferred for a certain period of time (the holding time) is the connection released again.

We present a number of different Markov models for the OCDR mechanism. The first one is a simple model, where the packets are assumed to arrive according to a Poisson process. Furthermore, the time needed for establishing a connection and the holding time of the mechanism are assumed to have an exponential distribution. For this model, we present an analytical solution. The second model is a refinement of the first one, where packets arrive according to an interrupted Poisson process (IPP). This model is further refined, so that the connection establishment time and the holding time can be modelled with an Erlang distribution. The second and third model are solved numerically using the matrix geometric solution method developed by Neuts [2], and supported by the package Xmgm [3].

This paper is organized as follows. First, in section 2,

† Present address: Ericsson Business Mobile Networks BV, PO Box 645, 7500 AP, Enschede, The Netherlands (e-mail address: gehe@ensae.ericsson.se).
‡ Present address: RWTH-Aachen, LFG Informatik/Verteilte Systeme, D-52056, Germany (e-mail address: haverkort@informatik.rwth-aachen.de).

0967-1846/96/010053+15$19.50

we define the connection management function, present alternatives to implement this function, and present the OCDR mechanism in more detail. In section 3, we model and analyse the performance of the OCDR mechanism under various traffic characteristics. In section 4, we evaluate the performance of the mechanism, and compare it to the performance of some conventional strategies. Finally, in section 5, we give some concluding remarks.

2. Connection management

In order to avoid confusion concerning the meaning of the term connection management, we explain what this function comprises in section 2.1. After that, in section 2.2, we propose a candidate mechanism for connection management, and relate it to other, more conventional mechanisms. Finally, in section 2.3, a number of performance measures are identified, which will be used to evaluate and compare the mechanisms.

2.1. Definition

Connection management is the function that interacts with the signalling system of the ATM network to ensure that an ATM connection to the proper destination, and with sufficient bandwidth, is available for the transfer of a packet. The connection management function only interacts with the signalling system locally. It can ask the signalling system to establish a connection to a certain destination, to modify the bandwidth assigned to a connection, or to release a connection. The actual establishment, maintenance, and release of ATM connections is performed by the signalling system. For this purpose, the signalling system must interact with ATM layer entities in all systems along the route of the connection. In the case of an establishment or a request for additional bandwidth, negotiation is done between the requesting protocol entity, the destination protocol entity, and the signalling system (on behalf of the ATM network). The signalling system confirms the success or failure of the request to the connection management function. No negotiation is needed for the release of a connection.

Basically, a connectionless service over ATM can be provided in two different ways: depending on the application either an end-to-end protocol or a node-by-node protocol can be used. The end-to-end protocol corresponds to the case with connectionless protocol entities only in the end systems. In order to transfer a packet from one end system to another end system, use should be made of an ATM connection between these end systems. This way of providing a connectionless service is referred to as the indirect method within the ATM recommendations of the International Telecommunication Union [4]. The node-by-node protocol corresponds to the presence of one or more intermediate connectionless protocol entities called Connectionless Servers (CLSs). If an end system has to transfer a packet, it has to use an ATM connection to a CLS. In this CLS, the packet is routed to a connection with the appropriate next node, and so on. Finally, in the destination node, the packet is delivered to the receiving service user. The ITU refers to this way of providing a connectionless service as the direct method.

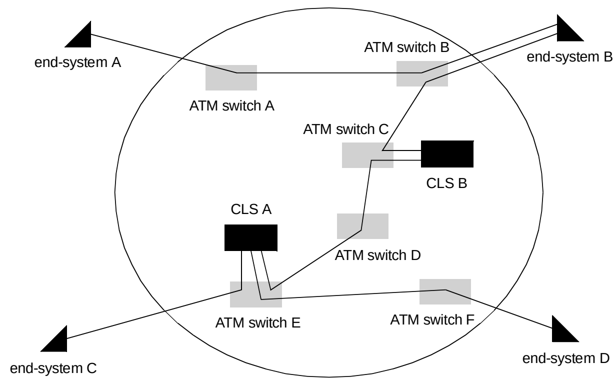

Figure 1. Use of ATM connections to provide a connectionless service.

Figure 1 illustrates the different types of ATM connections that can be used in both methods. A line in the figure represents a set of two connections, one for each direction. Here, the indirect method is used between end-systems A and B. End-system B also employs the direct method, and therefore has a connection with CLS B. End-systems C and D are connected to CLS A, and both CLSs are also connected.

In order to maintain the overlay network of connections properly, the connection management function must analyse the need for the transfer of packets. Here a problem arises: in principle, the protocol entity does not have advance knowledge about the destination and the required bandwidth of the ATM connections it will need. This is because of the connectionless nature of the communication, where only individual packets are transferred, which are routed independently through the network.

In practice, a certain correlation, in the destination as well as in the arrival time, can be expected due to the behaviour of (application) protocols in the higher layers. An application for which a single packet is transferred is likely to generate more data, so that more packets to the same destination will follow within a limited period of time. This expected correlation can be exploited by the connection management function, i.e. it can be used in the establishment, modification and release of ATM connections. For example, the expectation that there is a large amount of traffic between two LANs which have to be interconnected can lead to the establishment of a permanent connection between these LANs. The new mechanism we present in the following section exploits this correlation to a large extent.

2.2. Candidate mechanisms

Several mechanisms for setting up ATM connections for the transfer of packets can be envisaged. The objective of any such mechanism is to minimize the load that is put on the ATM network for maintaining connections. Bandwidth that is reserved for these connections cannot be used for other (connection-oriented) applications. Furthermore, the load on the signalling system for establishing, modifying, and releasing connections must not be too high. Too many signalling operations could result in an overloaded signalling system, and hence in a degraded signalling performance, e.g. in a high connection setup delay. Finally, the delay experienced by packets in an end-system or CLS must be acceptable.

Let us now discuss the candidate connection management mechanisms. First, we describe two conventional mechanisms.

2.2.1. Connection per packet (CpP). A connection, necessary to transfer a packet, can be established as soon as the packet arrives at a protocol entity. The connection is released again immediately after the transfer of the packet. This mechanism does not exploit the expected correlation between packets. All necessary knowledge, i.e. destination and amount of data to be transported, is known at the moment the connection is established. Note that a packet will in general be sent in several ATM-SDUs (cells). A practical extension might therefore be to transfer subsequent packets with the same destination, arriving during the transmission time of the current packet, over the same connection.

2.2.2. Permanent connection (PC). Alternatively to the previous mechanism, a protocol entity can maintain one or more permanent connections to various possible destination entities. A packet arriving in this protocol entity is transferred on one of these connections. Note that the entity must maintain connections to all entities to which it must be able to transfer packets directly. Other entities may be reached in several steps, via intermediate entities.

The exact specification of the required bandwidth is a problem for this mechanism. In principle, the CL protocol entity has no advance knowledge about the arrival times and the lengths of the packets. It can only use information about subscription, and statistics to predict the required bandwidth. Optionally, the connection management mechanism may modify the bandwidth of the connection during the lifetime of a connection [5].

2.2.3. On-demand connection with delayed release (OCDR). In this mechanism a connection will be established if a packet has arrived, and no connection to the proper destination protocol entity is available. The connection will not be released immediately after transferring the packet. It can be used for consecutive packets to the same destination as well. The connection will be released if it has not been used for a certain period of time, the *holding time*.

This mechanism tries to exploit the expected clustering of arrivals of consecutive packets for the same destination entity. It assumes that the expected time until the next arrival is longer after the holding time has expired than immediately after a departure. Thus, it can reduce the time a connection has to be maintained, compared to the PC mechanism, and at the same time, reduce the mean delay experienced by packets, compared to the CpP mechanism.

The CpP and PC mechanisms are special cases of this one: the OCDR mechanism with zero holding time is equivalent to the CpP mechanism with the mentioned extension, and the PC mechanism is equivalent to the OCDR mechanism with an infinite holding time.

2.2.4. Discussion. The described mechanisms can be applied between two end-systems, between an end-system and a CLS, as well as between two CLSs. The choice for a certain mechanism and the accompanying parameters (e.g. holding time) can be made for each pair of (source and destination) protocol entities† individually. The choice will depend on the expected arrival times and packet lengths of the traffic from the source protocol entity to the destination protocol entity. These depend heavily on the expected applications. Furthermore, the arrival times depend also on the extent to which traffic from different applications or end-systems can be multiplexed onto one connection. If the direct method of providing a connectionless service is used, packets to different end-systems can be multiplexed, e.g. on the connection to the first CLS, or between CLSs. For the indirect method, packets to different end-systems must use different connections, since end-to-end connections are used in this case. Note that an end-system can for instance be a router or bridge to a LAN, so that it does already multiplex the traffic of different LAN stations.

The amount of traffic that must be transferred over connections between CLSs can be expected to be so high that permanent connections must be maintained between these CLSs. Between which CLSs connections must be established, and how much bandwidth must be assigned to the connections is a dimensioning problem that is similar to dimensioning problems that can be found in traditional data networks [6, 7].

In this paper, we focus on mechanisms for the establishment and release of fixed-bandwidth connections. The proposed OCDR mechanism is a mechanism that can provide for this. If the direct method of providing a connectionless service is employed, it can be used to connect end-systems to an Access CLS. It can be applied on the connection from end-system to CLS and from CLS to end-system independently. If the indirect method is employed, i.e. if end-systems need to be connected directly, and no CLSs are used, the OCDR mechanism can be used to control the connection between end-systems.

2.3. Performance criteria

The three connection management mechanisms (CpP, PC, and OCDR) differ in a number of ways. In the following sections, we will investigate what the performance differences are. The following performance measures are important for the evaluation of the mechanisms.

- *Average delay*
  The average delay is the mean time elapsing from the arrival of a packet in the buffer of a CLS or end-system until the departure of (the last cell of) the packet from the buffer. This delay consists of the time a

† The terms ‘source’ and ‘destination’ do not necessarily refer to end-systems. They must be interpreted relative to the connection, and can as such refer to intermediate systems, i.e. CLSs.

packet spends in a buffer plus the time necessary for transmission of the consecutive cells of a packet on an outgoing connection. It can include the connection setup delay, if no connection is readily available.

- *Average reserved bandwidth*
  The average reserved bandwidth is the long-term average bandwidth reserved on a connection between a source/destination pair of CLSs. Periods of time in which no connection is available are taken into account in this average, and are considered as periods during which the reserved bandwidth is zero. The average reserved bandwidth will be strongly related to the costs of the service.
- *Average number of connection setups per second*
  The average number of connection setups per second is the long-term average number of times a connection is established per second. This measure is an indication for the load on the signalling system, which is also a cost factor for the provision of a connectionless service. Note that the number of requests for connection release equals the number of requests for connection establishment.

3. Modelling and analysis

In the previous section, we have presented OCDR as one of the candidate mechanisms for the connection management function. The purpose of this mechanism was to reduce the reserved bandwidth, compared to the PC mechanism, while reducing the delay, compared to the CpP mechanism. The purpose of this section is to investigate this effect quantitatively. Therefore, we model and analyse the mechanism to obtain results for the performance measures identified in section 2.3. The modelling and analysis can also be used for the CpP and PC mechanisms, since these are limiting cases of the OCDR mechanism.

We model the behaviour of the OCDR mechanism as far as a single pair of source and destination nodes is concerned. A connection between these entities may or may not exist. The connection is established if a packet arrives in the source node for the destination node under consideration. After finishing the transfer of a packet, the connection is released if no new packet for the destination arrives before the holding time expires.

We refrain from modelling the detailed behaviour at cell level, since the packet level behaviour is dominant at the time-scale that is relevant for the connection management mechanisms. Furthermore, the models in this section are continuous-time models, since the slotted nature of ATM is not relevant at the considered time-scale.

Table 1 gives an overview of the model parameters for later reference. They will be introduced later on.

In section 3.1, we first discuss the workload of the model, i.e. the stream of packets arriving at the source node that are destined for the destination node. Two different traffic types are defined: Poisson traffic and bursty traffic. In section 3.2, we present the modelling and analysis of the OCDR mechanism assuming Poisson traffic. Next, in section 3.3, we present the modelling and analysis assuming bursty traffic. Finally, in section 3.4, we present an extension to the latter model, which is based on more realistic assumptions for the holding time of the OCDR mechanism and the time needed to establish a connection.

Table 1. Model parameters.

$1/\lambda$	Average packet interarrival time
$1/\mu$	Average packet transmission time
$1/r$	Average connection holding time
$1/c$	Average connection setup time
l	Average packet length
$1/\alpha$	Average burst time
$1/\beta$	Average interburst time
n	Number of Erlang stages for the holding time
m	Number of Erlang stages for the connection setup time

3.1. Workload

We adopt two different workload characterizations for the analysis of the OCDR mechanism. The first one, Poisson traffic, results in an analytically tractable performance model. It gives insight into the behaviour of the protocol. The second one, bursty traffic, gives a more realistic characterization of the expected traffic, and therefore more accurate performance measures.

3.1.1. Poisson traffic. The consecutive interarrival times of packets are assumed to be independent, and exponentially distributed with mean $1/\lambda$.

The operation of the OCDR mechanism is based on the assumption that there will be a correlation in arrival times of subsequently transferred packets. Poisson traffic does not have this property. Therefore, the advantages of the mechanism will not be revealed.

3.1.2. Bursty traffic. Here, we assume that packets arrive in bursts. In [8], it is shown that such a bursty traffic source, which is modelled by a 'Train Model', provides a realistic description of the traffic on a local computer network (Ethernet). In [9], it is shown that a Markov Modulated Poisson Process (MMPP), which can also be used to describe the bursty traffic source is very well suited to represent correlations between subsequent arrivals. We use the simplest MMPP possible, also known as the interrupted Poisson process (IPP), for our evaluations. It should be noted that with IPPs sources with extreme burstiness can be described [10].

Let the stochastic process $\{A(t),\ t > 0,\ A(t) \in \{0, 1\}\}$ describe the arrival mode at time t. If $A(t) = 1$ the arrival process is said to be in a burst, and packets arrive at rate λ, i.e. with exponentially distributed interarrival times with mean $1/\lambda$. If $A(t) = 0$, the arrival process is said to be in an interburst period and no packets arrive. We assume the burst time and the interburst time to be exponentially distributed, with mean $1/\alpha$ and $1/\beta$ respectively (see figure 2). It can easily be seen that the long-term mean of $A(t)$, denoted $E[A]$ can be expressed as

$$E[A] = \frac{1/\alpha}{1/\beta + 1/\alpha} = \frac{\beta}{\alpha + \beta}. \qquad (1)$$

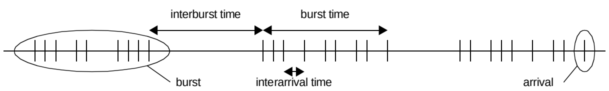

Figure 2. Behaviour of an interrupted Poisson process.

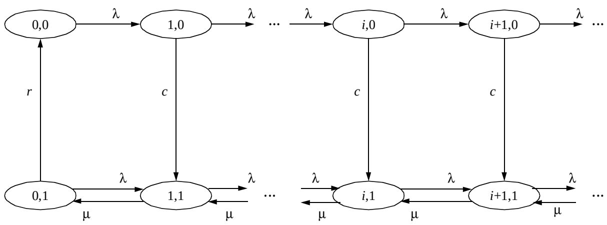

Figure 3. CTMC for OCDR under Poisson traffic.

$E[A]$ can be interpreted as the fraction of time the arrival process is in burst mode. This implies that the long-term mean arrival rate is given by $E[A]\lambda$.

3.1.3. Packet length distribution. We assume the (application) packet length to be exponentially distributed. The mean packet length is l bits. Note that in a real situation, the packet length will have a discrete distribution. However, the exponential distribution can be seen as a continuous time equivalent of the geometric distribution.

3.2. OCDR under Poisson traffic

Let us now present a model for the OCDR mechanism with Poisson arrivals. The concerned model is a Continuous Time Markov Chain (CTMC). We are able to derive a closed-form solution for the stationary state probabilities of this model [11].

When a packet arrives, and no connection exists, the connection management function invokes the signalling system in order to establish a connection. Setting up a connection is assumed to take an exponentially distributed time with mean $1/c$. When the establishment of the connection has been confirmed to the connection management function, the transmission of packets can be started. We assume that transmission of packets takes place at a rate of μl bits per second, i.e. every packet transmission takes an exponentially distributed time with mean $1/\mu$ seconds. After all packets have been sent, the connection will be released when the system is empty for an exponentially distributed holding time with mean $1/r$.

Let the stochastic process $\{N(t),\, t > 0,\, N(t) \in \mathbb{N}\}$ denote the number of packets in the system at time t, i.e. the number of packets in the source entity, destined for the destination entity under consideration. Furthermore, let the stochastic process $\{V(t),\, t > 0,\, V(t) \in \{0, 1\}\}$ indicate whether or not a connection is available at time t. Then, the process $\{N(t), V(t)\}$ is a CTMC. In figure 3 the resulting state transition diagram is depicted. It should be noted that $N(t)$ and $V(t)$ are not independent.

We are interested in the steady state behaviour of this CTMC. Let us denote the steady state probability distribution of this CTMC as $P(i, j)$:

$$P(i, j) = \lim_{t\to\infty} \{N(t) = i \wedge V(t) = j\}. \tag{2}$$

The following system of balance equations can be obtained (equations (3)–(6)):

$$(c + \lambda)P(i, 0) = \lambda P(i - 1, 0) \quad \text{for } i > 1, \tag{3}$$

by equating the flow into state $(i, 0)$ to the flow out of the same state;

$$\mu P(i, 1) = \lambda(P(i - 1, 1) + P(i - 1, 0)) \quad \text{for } i > 1, \tag{4}$$

by equating the flow across the boundary between the states with $N(t) > i$ and the states with $N(t) < i$;

$$rP(0, 1) = \lambda P(0, 0), \tag{5}$$

by equating the flow into state $(0, 0)$ to the flow out of this state; and

$$\sum_{i=0}^{\infty}(P(i, 0) + P(i, 1)) = 1, \tag{6}$$

for normalization.

We will relate all stationary probabilities to $P(0, 0)$, and use the following notation:

$$\rho = \frac{\lambda}{\mu}, \tag{7}$$

and

$$\sigma = \frac{\lambda}{\lambda + c}. \tag{8}$$

Notice that ρ is the long-term fraction the server is busy, i.e. it is the utilization, expressing the incoming amount of work per unit of time. Consequently, it must hold for stability: $\rho < 1$.

From equation (3) it follows directly that, for all i,

$$P(i, 0) = \sigma^i P(0, 0). \tag{9}$$

From equation (4) and equation (9), we have for the states (i, j) with $j = 1$

$$P(i, 1) = \rho(P(i - 1, 1) + \sigma^{i-1} P(0, 0)). \tag{10}$$

Repeatedly substituting equation (10) into itself yields:

$$P(i, 1) = \rho^i P(0, 1) + \sum_{k=1}^{i} \rho^k \sigma^{i-k} P(0, 0)). \tag{11}$$

Using equation (5), this results, for all i, in

$$P(i, 1) = \left(\frac{\lambda}{r}\rho^i + \sum_{k=1}^{i} \rho^k \sigma^{i-k}\right) P(0, 0). \tag{12}$$

We now realize that the long-term fraction of time a server is busy must be equal to the utilization ρ, i.e. the following equality holds:

$$P\{\text{server actually serving}\} = \sum_{i=1}^{\infty} P(i, 1) = \rho. \tag{13}$$

By substituting equations (13), (5) and (9) in the normalization equation (6), we obtain:

$$P(0,0)\left(\sum_{i=0}^{\infty}\sigma^i + \frac{\lambda}{r}\right) + \rho = 1. \qquad (14)$$

Rewriting the geometric series yields

$$P(0,0)\left(\frac{1}{1-\sigma} + \frac{\lambda}{r}\right) + \rho = 1, \qquad (15)$$

which, after substituting equation (8), results in the following expression for $P(0,0)$:

$$P(0,0) = (1-\rho)\frac{1/\lambda}{1/\lambda + 1/r + 1/c}. \qquad (16)$$

Substitution of this expression in equations (9) and (12) leads to a closed-form solution for the steady state probabilities.

Having deduced this closed-form solution we can derive expressions for a number of performance measures. For some of the measures, we make use of the Cauchy product rule [12], according to which:

$$\sum_{i=0}^{\infty}\sum_{k=0}^{i}\rho^k(i-k)\sigma^{i-k} = \left(\sum_{i=0}^{\infty} i\sigma^i\right)\sum_{i=0}^{\infty}\rho^k \quad \text{for } \rho, \sigma < 1. \qquad (17)$$

The following measures have been derived for the evaluation of the OCDR mechanism:

- average number of packets in the system, $E[N]$:

$$E[N] = \sum_{i=0}^{\infty} i(P(i,0) + P(i,1)) = \frac{\rho}{1-\rho} + \left(\frac{\lambda + c}{c}\right)\frac{1/c}{1/\lambda + 1/r + 1/c}; \qquad (18)$$

- average delay, $E[T]$, using Little's law:

$$E[T] = \frac{1}{\lambda}E[N] = \frac{1}{\lambda}\left(\frac{\rho}{1-\rho} + \left(\frac{\lambda + c}{c}\right)\frac{1/c}{1/\lambda + 1/r + 1/c}\right); \qquad (19)$$

- the fraction of time a connection is available, $E[V]$, using equation (13):

$$E[V] = \sum_{i=0}^{\infty} P(i,1) = \rho + (1-\rho)\frac{1/r}{1/\lambda + 1/r + 1/c}; \qquad (20)$$

- average reserved bandwidth, $E[B]$, in bit/s:

$$E[B] = \mu l E[V] = \mu l\left(\rho + (1-\rho)\frac{1/r}{1/\lambda + 1/r + 1/c}\right); \qquad (21)$$

- average number of connection setups per second, $E[S]$:

$$E[S] = c\sum_{i=1}^{\infty} P(i,0) = (1-\rho)\frac{1}{1/\lambda + 1/r + 1/c}. \qquad (22)$$

In equation (18) and equation (19), the first term equals the M/M/1 result for the average number of packets in the system and the average delay respectively. In equation (20), the first term (ρ) corresponds to the fraction of time that a connection is used for transmission. The second term expresses the time that an existing connection is idle.

3.3. OCDR under IPP traffic

In order to model the behaviour of the OCDR mechanism subject to an IPP as described in section 3.1, the CTMC of the previous section needs to be extended. The system is now modelled by the process $\{N(t), V(t), A(t)\}$, which is again a CTMC. Recall that $A(t)$ denotes the state of the arrival process at time t. The implicit assumption made at this point is that $A(t)$ is independent of $N(t)$ and $V(t)$. As such, recent investigations about the self-similarity of (Ethernet) traffic and the resulting existence of dependencies between these stochastic processes, are not taken into account (see also [13]). By making the above assumption, we will arrive at Markovian models which can be analysed with known techniques. A more detailed investigation of the above mentioned dependencies (their existence in this case, and their influence) goes beyond the scope of the current paper.

We are again interested in the steady-state behaviour of the CTMC. We now define the steady state probabilities, $P(i, j, k)$, as follows:

$$P(i, j, k) = \lim_{t\to\infty} P\{N(t) = i \wedge V(t) = j \wedge A(t) = k\}. \qquad (23)$$

The state transition diagram of this CTMC is depicted in figure 4. It is similar to the one for Poisson arrivals (figure 3). The state space is duplicated, to incorporate the state of the arrival process. The 'front plane' of states (the states labelled $(i, j, 1)$) represents the situation where the arrival process is in a burst. The transitions between the states are identical to those for Poisson arrivals. The 'back plane' of states (the states labelled $(i, j, 0)$) represents the situation where the arrival process is in an interburst period. It is identical to the 'front plane' except for the transitions with rate λ, which have been removed to represent the absence of arrivals. The system transits from the 'front plane' to the 'back plane' and back with rates α and β, respectively.

For the analysis of this CTMC, we use the matrix geometric solution method developed by Neuts ([2]; see also [3], [14], and [15]). Therefore, the repetitive structure of the CTMC is utilized. Let the states be ordered lexicographically ((0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), etc). Then the generator matrix **Q** of the CTMC

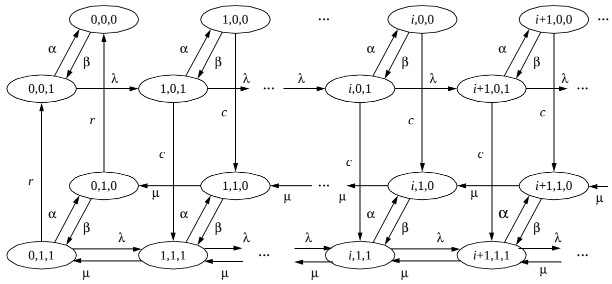

Figure 4. CTMC for OCDR under IPP traffic.

can be written in a block-triagonal form as follows:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_{00} & \mathbf{B}_{01} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B}_{00} & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{24}$$

where all constituent elements of $\mathbf{Q}$ are 4×4 matrices, and $\mathbf{0}$ is a matrix with all zeros. $\mathbf{Q}$ has a tridiagonal form, similar to the generator matrix of an M/M/1 queue. However, here the constituent elements are matrices themselves, unlike in the M/M/1 model, where the constituent elements are scalars. According to the matrix geometric solution technique, the stationary state probability distribution of the Markov process can be easily obtained after numerically solving a matrix quadratic equation with $\mathbf{A}_0$, $\mathbf{A}_1$ and $\mathbf{A}_2$ as coefficients for the 'repeating' behaviour, and a system of linear equations for the 'boundary' behaviour. For this purpose, we have used the software tool Xmgm [3].

Let us now give the constituent $\mathbf{A}$- and $\mathbf{B}$-matrices of the generator matrix $\mathbf{Q}$. The set of states of the CTMC for which the number of customers in the system equals i is called *level* i. Transitions between levels correspond to the arrival or departure of a packet. Transitions within a level correspond to a change in the state of the arrival process, or a change in the presence of the connection. The matrix $\mathbf{A}_0$ gives the transition rates between states, given that the system goes from level i to level $i+1$; its entries correspond to packet arrivals. It is defined as follows:

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix}. \tag{25}$$

$\mathbf{A}_2$ describes the transitions between states, while the system transits from level i to level $i-1$ its entries correspond to packet departures (services):

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}. \tag{26}$$

Finally, $\mathbf{A}_1$ describes the transitions within a level, i.e. the change of arrival process mode and the setup of a connection, and on the diagonal, the zero-row-sum compensation for the complete generator matrix:

$$\mathbf{A}_1 = \begin{bmatrix} -(\beta + c) & 0 & c & 0 \\ \alpha & -(\alpha + c + \lambda) & 0 & c \\ 0 & 0 & -(\beta + \mu) & \beta \\ 0 & 0 & \alpha & -(\alpha + \lambda + \mu) \end{bmatrix}. \tag{27}$$

The $\mathbf{B}$-matrices define the boundary transitions, i.e. the transitions to and/or from level 0. It turns out that for the CTMC described in this subsection, the transitions to and from level 0 are the same as for other levels, i.e.

$$\mathbf{B}_{01} = \mathbf{A}_0, \tag{28}$$

and

$$\mathbf{B}_{10} = \mathbf{A}_2. \tag{29}$$

The transitions within level 0 describe the release of a connection as well as changes in the arrival process mode, and on the diagonal, the usual zero-row-sum compensation:

$$\mathbf{B}_{00} = \begin{bmatrix} -(\beta) & \beta & 0 & 0 \\ \alpha & -(\alpha + \lambda) & 0 & 0 \\ r & 0 & -(\beta + r) & \beta \\ 0 & r & \alpha & -(\alpha + \lambda + r) \end{bmatrix}. \tag{30}$$

We defer the explicit description of the measures of interest to the end of the next section.

3.4. OCDR with Erlang holding and connection setup times

Up to now, we have modelled the holding time of the OCDR mechanism as an exponentially distributed time. In the real system, this time will probably be determined by a fixed timer value, and hence be a deterministic one. In order to model this holding time more accurately, and to check our previous model, we will now model it as an Erlang n distribution, i.e. a distribution with n identical exponentially distributed stages. It is known that the squared coefficient of variation of such a distribution approaches 0 if $n \to \infty$, i.e. the Erlang distribution approaches a deterministic distribution [16]. Similarly, we model the connection setup time with an Erlang m distribution, since this time can also be assumed to have a lower variance than that of an exponential distribution.

Let us now enhance the model that we have developed for OCDR under IPP arrivals, in order to cope with the Erlang distributions for the holding time and the connection setup time. The system is again modelled with the CTMC $(N(t), V(t), A(t))$, but now the stochastic process $V(t)$ is defined to represent the stage of the Erlang distribution for the holding time or connection setup time as follows.

For an empty system, i.e. $N(t) = 0$, $\{V(t), t > 0, V(t) \in \{0, 1, \ldots, n\}\}$ defines the stage of the holding time, which is initially n. Assuming the system stays empty, $V(t)$ decreases with a rate nr. If $V(t) = 0$ the

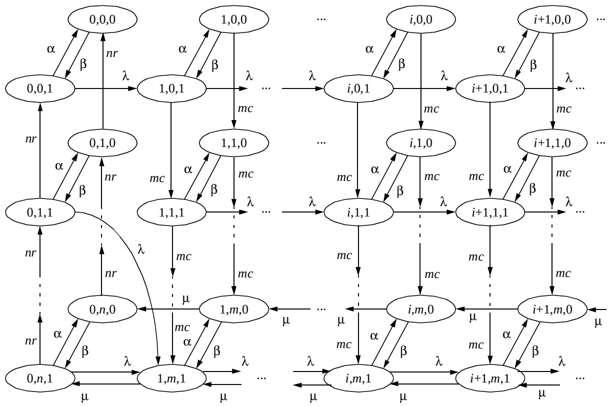

Figure 5. CTMC of OCDR with Erlang holding and connection setup times.

connection has been released, otherwise it is still available. If the system becomes non-empty, i.e. $N(t) > 0$, before the connection has been released, the timer is reset, i.e. $V(t)$ starts at n again the next time the system becomes empty.

For a non-empty system, i.e. $N(t) > 0$, $\{V(t), t > 0, V(t) \in \{0, 1, \ldots, m\}\}$ defines the stage of the connection setup time, which is initially 0. Now, $V(t)$ increases with a rate mc. If $V(t) = m$, the connection has been set up, otherwise it is not yet available. The stationary state probability distribution of the CTMC is still defined as in equation (23), however, note that the domain of $V(t)$ has changed.

The state transition diagram of the CTMC is shown in figure 5. Its generator matrix $\mathbf{Q}$ has the same shape as the one in equation (24). However, the constituent matrices are defined differently. $\mathbf{A}_0$, $\mathbf{A}_1$ and $\mathbf{A}_2$ are square matrices of size $2(m+1) \times 2(m+1)$, with similar semantics as before. They are defined by equations

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & \lambda & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \lambda & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & \lambda \end{bmatrix}, \qquad (31)$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \mu & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & \mu \end{bmatrix}, \qquad (32)$$

and equation (33) shown in figure 6. $\mathbf{B}_{01}$ is a matrix of size $(n+1) \times (m+1)$:

$$\mathbf{B}_{01} = \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & \lambda & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & \lambda \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & \lambda \end{bmatrix}. \qquad (34)$$

$\mathbf{B}_{10}$ is a matrix of size $(m+1) \times (n+1)$, which is defined similarly to $\mathbf{A}_2$:

$$\mathbf{B}_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \mu & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & \mu \end{bmatrix}. \qquad (35)$$

Finally, $\mathbf{B}_{00}$ is a matrix of size $(m+1) \times (m+1)$, which is defined as equation (36) shown in figure 7. Now that we have defined the constituent matrices of the generator matrix, we are able to solve for the stationary state probabilities of the CTMC, again using Xmgm. From the obtained probabilities we can derive a number of performance measures:

- average number of packets in the system, $E[N]$:

$$E[N] = \sum_{i=1}^{\infty} i \sum_{j=0}^{m} (P(i,j,0) + P(i,j,1)); \qquad (37)$$

- average delay, $E[T]$, using Little's law:

$$E[T] = \frac{1}{E[A]\lambda} E[N]; \qquad (38)$$

- the fraction of time a connection is available, $E[V]$:

$$E[V] = \sum_{i=1}^{\infty} (P(i,m,0) + P(i,m,10)) + \sum_{j=1}^{n} (P(0,j,0) + P(0,j,1)); \qquad (39)$$

- average reserved bandwidth, $E[B]$:

$$E[B] = \mu l E[V]; \qquad (40)$$

- average number of connection setups per second, $E[S]$, by recognizing that the establishment of a connection needs to be performed each time a packet arrives while the system is empty and no connection is available:

$$E[S] = \lambda P(0,0,1). \qquad (41)$$

Note that this model is equivalent to the CTMC of the previous section for $n = 1$ and $m = 1$. The expressions for the performance measures (equations (37) to (41)) apply also to that model if n and m are set to 1.

4. Evaluation

Using the models described in subsections 3.2–3.4, we will evaluate the performance of the OCDR mechanism. This will be done by comparing it to the CpP and PC mechanisms, with respect to the performance measures identified in section 2.3.

In section 4.1 we will present the values for the model parameters that have been used. Then, in section 4.2, we

$$\mathbf{A}_1 = \begin{bmatrix} -(\beta+mc) & \beta & mc & 0 & \dots & 0 & 0 \\ \alpha & -(\alpha+mc+\lambda) & 0 & mc & \dots & 0 & 0 \\ 0 & 0 & -(\beta+mc) & \beta & \dots & 0 & 0 \\ 0 & 0 & \alpha & -(\alpha+mc+\lambda) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -(\beta+\mu) & \beta \\ 0 & 0 & 0 & 0 & \dots & \alpha & -(\alpha+\lambda+\mu) \end{bmatrix}. \quad (33)$$

Figure 6. Equation (33).

$$\mathbf{B}_{00} = \begin{bmatrix} -\beta & \beta & 0 & 0 & \dots & 0 & 0 \\ \alpha & -(\alpha+\lambda) & 0 & 0 & \dots & 0 & 0 \\ nr & 0 & -(\beta+nr) & \beta & \dots & 0 & 0 \\ 0 & nr & \alpha & -(\alpha+\lambda+nr) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -(\beta+nr) & \beta \\ 0 & 0 & 0 & 0 & \dots & \alpha & -(\alpha+\lambda+nr) \end{bmatrix}. \quad (36)$$

Figure 7. Equation (36).

evaluate the performance of OCDR with a workload of Poisson traffic. In section 4.3, we evaluate OCDR with a workload of bursty (IPP) traffic. In section 4.4, we do the same, now assuming that the holding time and connection setup time have an Erlang distribution. In the last three subsections, we continue to assume bursty traffic and Erlang holding and connection setup times. In section 4.5, we analyse the behaviour of OCDR under varying holding time, in order to determine the optimal value for this control parameter. Finally, in sections 4.6 and 4.7, we respectively evaluate the performance of OCDR under varying load and burst length.

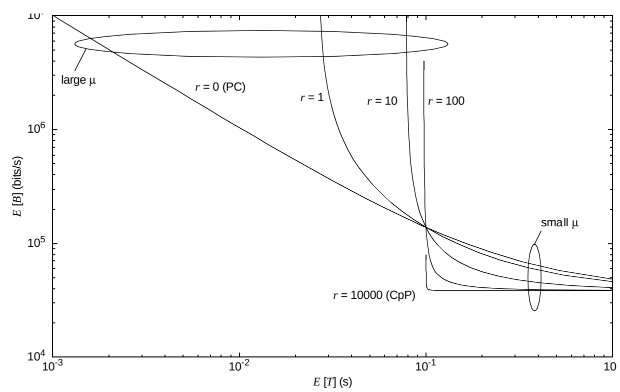

Figure 8. Average reserved bandwidth plotted against average delay (Poisson traffic).

4.1. Parameter values

The values for the model parameters, given in this subsection, are the default parameters used in the experiments. Unless stated differently, these values are used. They are summarized in table 2. Values may be different for different models. Furthermore, sometimes no default value is assumed for a parameter (—), or a parameter is not applicable (n.a.) to a model because it is not defined for that model.

The workload parameters for the IPP traffic are based on measurements in [8], taking into account the high-speed character of future applications. The average interburst time has been taken to be 25 seconds, i.e. $\beta = 0.04$. The average burst time is 1 second, i.e. $\alpha = 1$. The average number of arrivals per burst is 100, i.e. $\lambda = 100$. This is an order of magnitude higher than the value measured in [8], to reflect the expected increase in traffic intensity in the future.

For Poisson traffic, we only have to define the parameter of the exponential distribution of the interarrival time, λ. In order to have the average number of arrivals per second equivalent for IPP traffic and Poisson traffic, we have adopted $\lambda = 100\ E[A] = 100\ \beta/(\alpha+\beta) = 100/26 \approx 3.85$ for Poisson traffic.

The average packet length, l, is assumed to be 10 kbit for both Poisson and bursty traffic.

The average of the connection setup time, $1/c$, is assumed to be 100 ms, i.e. $c = 10$. The default value for the rate at which packets are served if a connection is available has been chosen such that the utilization during a burst is 0.8, i.e. $\mu = 125$. Furthermore, experiments (see, e.g. figure 11) have suggested a holding time of 0.1 second ($r = 10$) as an optimal value.

In the model with Erlang holding and connection setup times, the number of stages for the holding time has been taken to be 30. The connection setup time has an Erlang 5 distribution.

4.2. Poisson traffic

The expected gain of the OCDR mechanism is that it claims less resources from the ATM network than a PC mechanism, because an outgoing connection is released during periods in which no traffic arrives. The average reserved bandwidth ($E[B]$) is a good measure to express this claim, i.e. it is an indication of the costs of the use of the ATM network. A disadvantage of the use of OCDR, compared to PC, is that some packets will experience a higher delay, because a connection must be

Table 2. Default values for model parameters.

	Poisson traffic	IPP traffic (exponential holding and connection setup times)	IPP traffic (Erlang holding and connection setup times)
λ (s^{-1})	100/26	100	100
μ (s^{-1})	—	—	125
r (s^{-1})	—	—	10
c (s^{-1})	10	10	10
l (bit)	10 000	10 000	10 000
α (s^{-1})	n.a.	1	1
β (s^{-1})	n.a.	0.04	0.04
n	n.a.	n.a.	30
m	n.a.	n.a.	5

established explicitly before packets can be transferred. However, by serving packets at a high rate (μ), if the connection is available, i.e. by requesting a high bandwidth for the connection, the average delay ($E[T]$) can be kept acceptable.

The problem we are interested in is the following. Given a certain load (λ), and a certain required average delay ($E[T]$), what holding time ($1/r$) and what connection bandwidth (μ) should be chosen to achieve a minimal average reserved bandwidth ($E[B]$), i.e. minimal costs?

In order to deal with this problem, we investigate how the obtained average reserved bandwidth relates to the obtained average delay, for different values of the connection bandwidth (μ). In figure 8, we display both measures for varying μ, i.e. the curves that are drawn in the figure are parametric curves. μ is increasing from right to left along the curves. As can be observed, the average delay decreases, and the average reserved bandwidth increases with increasing μ. Curves are depicted for different holding times $1/r$. Note that an infinite holding time ($r = 0$) corresponds to the PC mechanism, and that a zero holding time (approached by $r = 10\,000$) corresponds to the CpP mechanism.

Let us discuss the characteristics of the graph in more detail. All the curves with $r > 0$ converge to some vertical asymptote, i.e. they show an asymptotic behaviour to a limiting value of $E[T]$. This is the limiting behaviour for $\mu \to \infty$, for which equation (19) reduces to

$$\lim_{\mu\to\infty} E[T] = (1/c)\frac{1/\lambda + 1/c)}{(1/\lambda + 1/r + 1/c)}. \qquad (42)$$

In this expression, $1/c$ is the expected time a customer has to wait if no connection is available upon arrival. The remaining factor can be considered as the probability that no connection is available for an arriving customer. We directly see that this implies that for the CpP mechanism ($r \to \infty$) the limiting value of the average delay equals $1/c$, the connection setup time, since all packets will find no connection available. For $r = 0$, the limit of $E[T]$ is 0 for $\mu \to \infty$, which implies that for the PC mechanism every delay demand can be guaranteed. However, due to the fact that μ is finite because of the finite capacity of an ATM link, this limit of 0 is never reached.

From equation (19) and (21), it can be derived that all the curves cross at $E[T] = 1/c$. Note that in this point the values for μ differ for the various curves. Only the average reserved bandwidth, which is the product of μ, the average packet length (l), and the fraction of time a connection is available (equation (20)) is constant.

We see from the curves that the optimal value for r for some average delay is either $r = 0$ when the required average delay is smaller than 0.1 s, or $r \to \infty$ when the required average delay is larger than 0.1 s. Since the required average delay can be expected to be in the order of 0.01 s, the PC mechanism ($r = 0$) is the only suitable mechanism, if traffic arrives according to a Poisson process.

Concluding we can state that the OCDR mechanism is not advantageous if packets arrive to a CLS or end-system according to a Poisson arrival process. Either the CpP or PC mechanism needs the least bandwidth to fulfil some delay demand. This is due to the fact that with the Poisson process arrivals do not cluster.

4.3. IPP traffic with exponential holding and connection setup times

Results for the OCDR model assuming arrivals according to an IPP and exponential holding and connection setup times are obtained using Xmgm with equations (25–30) as input. Analogously to the previous subsection, we plot the average reserved bandwidth versus the average delay (see figure 9). Curves have been drawn for the same values of r as in section 4.2. Again, the parameter μ varies along the curves.

Contrary to the OCDR mechanism under Poisson traffic, the mechanism is now advantageous if packets arrive to the CLS or end-system according to an IPP. Depending on the required average node delay, one of the values for r yields the lowest average reserved bandwidth. It can roughly be said that a PC mechanism ($r = 0$) is the optimal solution for a required average delay of less than 0.01, and a CpP mechanism ($r \to \infty$) for a required average delay of more than 0.1. For average delay requirements between 0.01 and 0.1, other values for r are optimal.

4.4. IPP traffic with Erlang holding and connection setup times

In the graph of figure 9, we have still assumed that the holding time and the connection setup time are

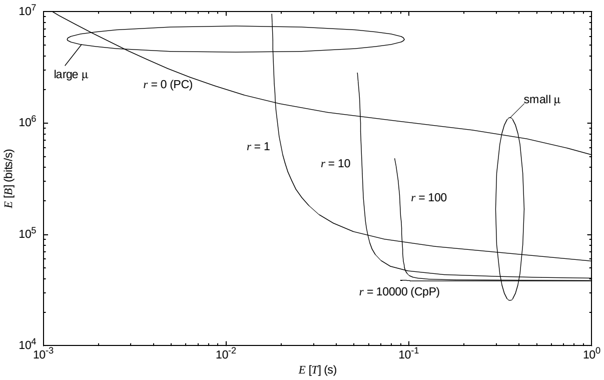

Figure 9. Average reserved bandwidth plotted against average delay (IPP, n = 1, m = 1).

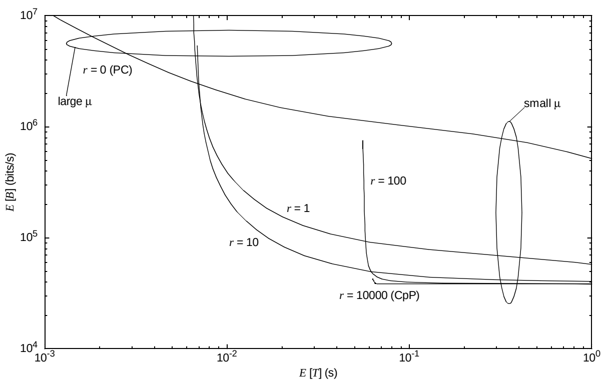

Figure 10. Average reserved bandwidth plotted against average delay (IPP, n = 30, m = 5).

exponentially distributed. In a real system this will definitely not be true. The holding time will be a deterministic one, and the connection setup time will also have a lower variance than an exponential distribution.

In order to model the real system more accurately, we assume that both the holding time and the connection setup time are distributed according to an Erlang distribution. To reflect the fact that the holding time will be deterministic, we model it with an Erlang 30 distribution, i.e. $n = 30$ in figure 5. For the connection setup time, we assume an Erlang 5 distribution, i.e. $m = 5$. In [14], it is shown that the results from the model are not very sensitive to the exact values of m and n, in the range of system parameters that has been used. The mentioned distributions are used throughout the rest of the section.

Let us again give the same type of graph as in the previous two subsections. Figure 10 gives curves of the average reserved bandwidth versus the average delay. The most conspicuous difference between this graph and the previous one appears in the curve for $r = 10$. This curve has shifted to the left, i.e. a lower average delay can be provided with the same average reserved bandwidth. As a result, OCDR, with a holding time of 0.1 s, is the most optimal mechanism for a wide range of delay requirements.

Why is a holding time of 0.1 s the most optimal one? If the holding time is (an order of magnitude) lower (e.g. $r = 100$), it is frequently shorter than the packet interarrival

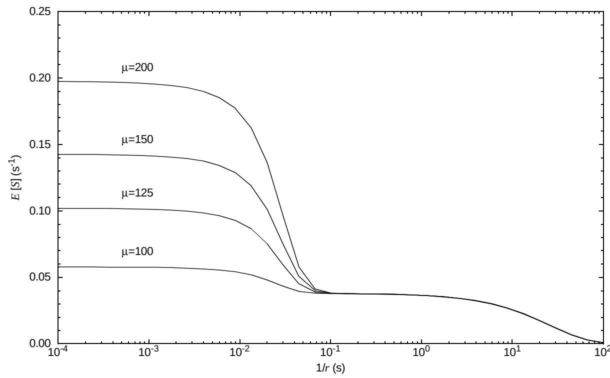

Figure 11. Average number of connection setups per second for varying holding time.

time during a burst. Consequently, the connection is released for a very short time, because the next packet of the burst will arrive shortly, and hence, only little bandwidth is saved at the cost of an increased average delay. If the holding time is (an order of magnitude) higher (e.g. $r = 1$), it is of the same order as the length of the entire burst, and bandwidth is wasted because the connection is not released fast enough, i.e. not even between bursts. Now, the difference between the results for exponential and Erlang holding time also become clear. If the holding time were taken from an exponential distribution instead of an Erlang (or deterministic) one, its actual value would often be far different from the mean of the distribution. As a result, the OCDR mechanism would perform worse, despite the optimal mean holding time.

For a required average delay in the range between 0.007 s and 0.06 s, the OCDR mechanism with a holding time of 0.1 s can fulfil the delay requirements at the lowest cost, i.e. with the lowest average reserved bandwidth. This range can expected to be the operational region of the mechanism. Compared to the PC mechanism a significant reduction of the average reserved bandwidth can be obtained, i.e. up to 95% (for $E[T] = 0.06$).

4.5. Optimal holding time

From now on, we assume IPP traffic and Erlang holding and connection setup times. In order to obtain more insight in the optimal value for the connection holding time, we do an experiment where we vary this parameter, keeping the other parameters constant (see table 2 for the default values). We perform the experiment for different values of μ, so that the utilization during a burst is between 0.5 ($\mu = 200$) and 1.0 ($\mu = 100$). Here we only show a graph for the average number of connection setups per second ($E[S]$) as a function of the holding time ($1/r$), in figure 11.

The influence of the holding time on the behaviour of the OCDR mechanism becomes very clear from the graph. If the holding time is larger than 0.1 second, the rate at which new connections are established is very close to the rate at which new bursts start, i.e. once per 26 seconds. This indicates the desired behaviour of OCDR: the connection is released between bursts, and held during bursts. If the

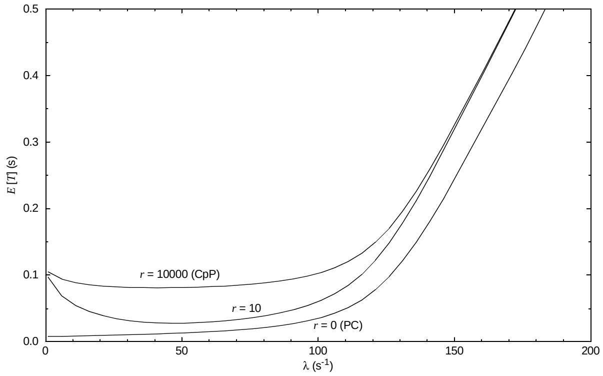

Figure 12. Average delay under varying load.

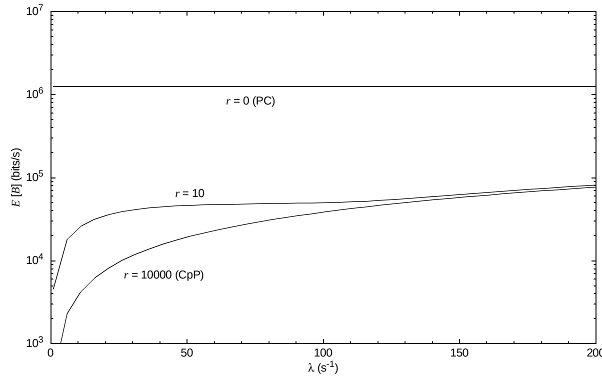

Figure 13. Average reserved bandwidth under varying load.

holding time becomes much larger, the average number of connection setups per second decreases, because the connection will no longer be released between bursts. The mechanism behaves like a PC mechanism. For small holding times (smaller than 0.01 second), the rate at which connections are set up depends on the utilization during a burst. In general, this rate is high, because the connection is often released during a burst, especially for low utilizations (i.e. high μ). If the utilization is high ($\mu = 100$), this is not the case because the system will only rarely become empty during a burst. In the range of $1/r$ between 0.01 s and 0.1 s, the average number of connection setups per second sharply decreases with increasing holding time. Concluding, we can state that for the given parameters, a holding time of 0.1 s is optimal for the proper operation of the OCDR mechanism.

4.6. Behaviour under varying load

In section 4.4, we have compared the OCDR mechanism to the PC and CpP mechanisms. It turned out that the OCDR mechanism can reduce the average reserved bandwidth by up to 95%, depending on the required average delay. Of course, the obtained results depend on the parameters of the arrival process. In order to obtain insight in this dependency, the effect of the load and the burst length on the results will be investigated below.

In order to vary the load on the connectionless protocol, λ is varied. Increasing λ means that the interarrival time in a burst decreases, and the number of arrivals per burst increases. μ is kept constant at 125, which implies that the bandwidth reserved for a connection is 1.25 Mbit/s (μl), if the connection is established. We give curves for three values of r. A holding time of 0.1 s ($r = 10$) turned out to be most optimal for OCDR in the previous subsection. For comparison, we also display curves for $r = 0$ (PC mechanism) and $r = 10\,000$ (approximating the CpP mechanism). Graphs are given for the average delay, $E[T]$ (figure 12), the average reserved bandwidth, $E[B]$ (figure 13), and the average number of connection setups per second, $E[S]$ (figure 14).

From figure 12, it can be observed that for the same μ, the average delay for OCDR is about twice the delay for PC if λ is between 50 and 125. This range corresponds to a utilization of the connection during a burst between 0.4 and 1.0. Recall that λ is the arrival rate during a burst. The CpP mechanism has a much higher delay here, because the connection is often released during a burst. The same can be observed for the OCDR mechanism if λ decreases below 50. If the load approaches zero, the mean delay for both the CpP and the OCDR mechanism approaches the sum of the packet transmission time and the connection setup time ($1/\mu + 1/c = 0.108$ s), because a new connection needs to be set up for every packet. In that case, the mean delay of the PC mechanism becomes the packet transmission time ($1/\mu = 0.008$ s), because a packet will not experience waiting time any more.

For $\lambda > \mu$ the average delays for OCDR and CpP converge, because the system will no longer become empty during a burst, due to temporary overload. Consequently, the CpP mechanism will not release the connection during the burst. From $\lambda = 150$, the difference in delay between the mechanisms will be constant with increasing load. The delay of both mechanisms will increase almost linearly. If the load is so high that the system cannot transmit the excess traffic of a burst in an interburst period any more, the delay will increase more rapidly with increasing load.

In figure 13, it can be seen that the average reserved bandwidth of the OCDR mechanism is almost constant for $\lambda > 50$. This indicates that the OCDR mechanism is stable with these parameters: the connection is available during bursts, and released between bursts, regardless of the traffic intensity during the burst. Of course, the average reserved bandwidth in the case of a permanent connection (PC) is constant at 1.25 Mbit/s. For the CpP mechanism, only the bandwidth that is really used is reserved, i.e. $E[B] = E[A]\lambda l = \lambda/2600\,\text{Mbit/s}$ (see equation (1)). The mechanism does not waste any bandwidth. As a result, the difference between the curve for CpP and the curves for the other mechanisms can be interpreted as the bandwidth that is wasted by the concerned mechanisms. If λ is larger than μ, i.e. $\lambda > 125$, OCDR wastes only very little bandwidth. The PC mechanism on the other hand uses about 20 times as much.

For high loads, a small difference between the average reserved bandwidth of the CpP and OCDR mechanisms remains, because of the bandwidth reserved during the holding times, after each burst. When the system becomes

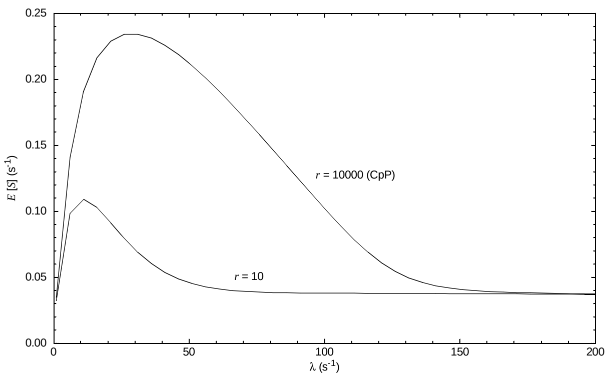

Figure 14. Average number of connection setups per second under varying load.

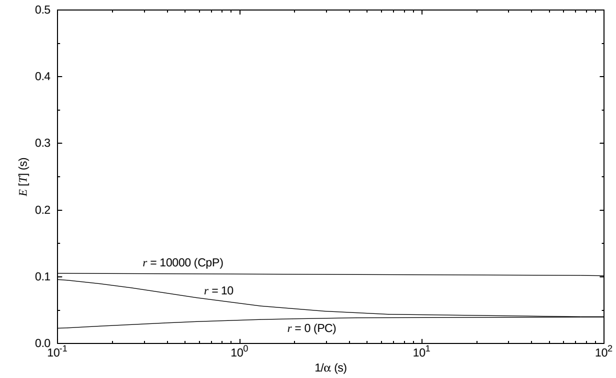

Figure 15. Average delay under varying burst length.

overloaded, i.e. $\lambda E[A] \rightarrow \mu$, the average reserved bandwidth for both mechanisms converges to the one for the PC mechanism, because the connection will not be released any more. If the load approaches zero, the reserved bandwidth for the CpP mechanism goes to zero. The average reserved bandwidth for the OCDR mechanism goes to zero as well, but much slower, because a connection, established for the transfer of a single packet, will only be released after the holding time expires.

Figure 14 shows the average number of connection setups per second ($E[S]$) as a function of the load (λ). Clearly, $E[S] = 0$ for the PC mechanism. The shape of the curves for OCDR and CpP is explained as follows. If the load is close to zero, $E[S]$ equals $\lambda/26$, since a new connection is established for each packet. If the load increases, more packets can be served by the same connection, because they arrive before the connection is released. Consequently, the average number of connection setups per second increases less than proportionally with the load. For certain λ, the increase of the number of arriving packets is cancelled by the increase of the number of packets served per connection, so that the curve has its peak. From this point onwards, $E[S]$ decreases for increasing λ. If the load is such that all packets of a burst are served by a single connection, $E[S]$ will not decrease any more. It will be close to the rate at which new bursts are started ($1/(1/\alpha + 1/\beta) = 1/26$). When the system becomes overloaded, i.e. $\lambda E[A] \rightarrow \mu$, the average number of connection setups per second approaches zero, because the connection will not be released any more (outside the scale of figure 14).

The load from which $E[S]$ remains constant differs for the CpP and OCDR mechanisms. For CpP, it remains constant if λ increases beyond 150, i.e. if the load during the burst exceeds the capacity. For OCDR, $E[S]$ remains constant if λ increases beyond 50, i.e. if the interarrival time during a burst exceeds the holding time most of the time. This confirms the indication that OCDR (with $r = 10$) operates properly if $\lambda > 50$, since in this range, a single connection is used to transmit an entire burst.

Concluding, we can state that the OCDR mechanism performs well with the given parameters if the utilization of a connection during a burst is between 0.4 and 1.0. For higher utilizations, the average delay becomes too high for all mechanisms. For lower utilizations, OCDR does not perform well, resulting in a relatively high average delay, and a high load on the signalling system.

4.7. Behaviour under varying burst length

Finally, we want to examine the behaviour of the OCDR mechanism under varying burst length ($1/\alpha$). Note that the burst length is expressed in time, not in the number of packets. Since the arrival rate during a burst (λ) is kept constant at 100, the average number of packets per burst equals $100/\alpha$. In [8], burst lengths of tens of packets have been measured, but the increasing volumes of data that need to be transported for many applications may lead to larger bursts. A burst length of $1/\alpha = 1$ seems to be a good indication of a realistic value. The interburst time ($1/\beta$) is varied proportionally to the burst length, so that the fraction of time the arrival process is in the burst state ($E[A]$), and hence the mean arrival rate is constant. Again, graphs for the average node delay $E[T]$ (figure 15), the average reserved bandwidth $E[B]$ (figure 16), and the average number of connection setups per second $E[S]$ (figure 17) are given.

From figure 15, it can be observed that the average delay of the CpP mechanism is almost constant under varying burst length. It is constantly close to the sum of the average setup time and the average transmission time ($1/c + 1/\mu = 0.108$ s). The average delay of the PC mechanism increases only slightly with the burst length. For very short burst lengths (out of the range of the graph), the average delay will approach the average transmission time ($1/\mu$). For increasing burst lengths, the average delay will convert to the average delay in a regular M/M/1 queue ($(1/\mu)/(1-\lambda/\mu) = 0.04$), since the interburst periods will not contribute to the measure any more, i.e. within a burst, 'steady-state' will be reached.

The average delay of the OCDR mechanism converges to the average delay of the PC mechanism if the burst length increases. The effect of the extra delay for establishing a connection decreases, because this is needed for a decreasing fraction of the customers. For decreasing burst lengths, the average delay of OCDR approaches the delay of the CpP mechanism since the clustering of subsequent arrivals disappears.

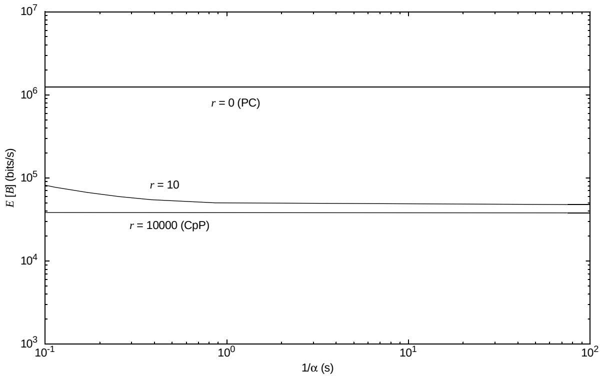

Figure 16. Average reserved bandwidth under varying burst length.

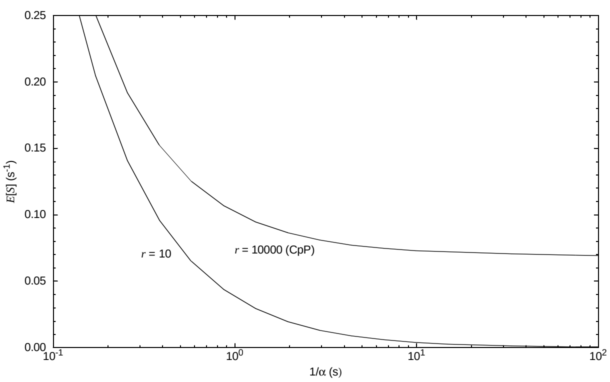

Figure 17. Average number of connection setups per second under varying burst length.

The average reserved bandwidth for OCDR is not very dependent on the burst length. In figure 16, it can be seen that it only increases slightly if the burst length decreases. The increase is caused by the fact that the connection is established and released more often. Before the connection is released, it has not been used for the holding time, i.e. some bandwidth is wasted. Again, the reserved bandwidth for the PC mechanism is fixed at 1.25 Mbit/s. The average reserved bandwidth for the CpP mechanism equals $E[A]\lambda l = 1/26\,\text{Mbit/s}$, i.e. the number of bits per second that are offered to the system.

Figure 17 indicates that the average number of connection setups per second is very close to the average number of bursts that start per second ($\alpha/26$). It is slightly higher, because the connection is sometimes accidentally released during a burst. For the CpP mechanism, $E[S]$ is much higher, since the connection is often released during a burst. Both curves converge if the burst length decreases, because having a connection per burst is the same as having a connection per packet if the burst consists of only a single packet. Clearly, the average connection setup rate equals zero for the PC mechanism.

From the graphs that we have considered, we can conclude that the average delay and the average reserved bandwidth of the CpP and PC mechanisms is hardly sensitive to the burst length. For each measure, the OCDR yields a value between those for the other two mechanisms.

5. Conclusions

In this paper, the connection management function has been defined as the function in a CLS or end-system that interacts with the signalling system to ensure that ATM connections for the transfer of packets are available when needed. A number of mechanisms have been identified that can be used to implement this function. The 'On-demand connection with delayed release' mechanism is one that maintains a connection only during periods in which packets arrive regularly. If no packets have arrived after the last transmission for a period of time, called 'holding time', the connection is released.

In order to evaluate the performance of the OCDR mechanism, a number of models have been constructed and analysed. The new mechanism has been compared with a 'Permanent connection' and a 'Connection per packet' mechanism. From the evaluation of a model that assumed a Poisson arrival process, some characteristics of the mechanism have been identified. However, this model did not reveal any advantages for the OCDR mechanism, because the underlying assumption of clustered arrivals was not captured in the arrival process. Therefore, a second model was used that does capture this aspect. It assumed an IPP as the arrival process. The evaluation of this model showed that OCDR can support the same average delay with significantly less reserved bandwidth than the PC mechanism.

In order to model the deterministic holding time more accurately, a third model has been evaluated, assuming an Erlang-30 distribution for the holding time. In this model, it is also assumed that the connection setup delay is distributed according to an Erlang-5 distribution. The evaluation of this model reveals the same bandwidth reductions as the previous model. However, here the most optimal value for the holding time can be identified more clearly.

Compared to the PC mechanism, the average delay increases only slightly for a given bandwidth of the outgoing connection, if the connection is released after each burst. However, the reduction of the average reserved bandwidth is significant (about 95% for the parameters used). This behaviour can be obtained at the cost of extra signalling traffic, at the beginning and end of every burst. The evaluation shows that the advantage is most pronounced if the burst size is large. Furthermore, it is shown that the proper operation of the mechanism depends on the arrival rate during a burst. For the given parameters, the OCDR mechanism operates well if the packet arrival rate in a burst is between 40 and 100% of the packet transmission rate. For lower loads, the connection is released too often, also within bursts, and hence the average number of connection setups per second and the average delay increase. For higher loads, none of the mechanisms operates well, because the system becomes overloaded during a burst, and the average delay becomes too high.

The major conclusion of the evaluation is that the OCDR mechanism is advantageous if the offered traffic has a bursty nature. The mechanism only works if interburst periods are long enough. This is probably not the case on the connections between CLSs, since traffic from many applications will have to be sent over the same connection. As a result, there will probably not be long interburst periods, where no packets have to be transmitted. The OCDR mechanism is most suitable for the indirect method of operation, i.e. for connections between end-systems, and for the connections between end-systems and CLSs. In these cases, only packets from or to a single or a few application process have to be supported by a single connection, so that the burst–silence cycle is most pronounced.

To evaluate the models, we have employed matrix-geometric methods using the tool Xmgm. The specification of the models, however, had to be performed at the Markov chain state level. We recently developed a class of stochastic Petri nets of which the underlying Markov chain exhibits a matrix-geometric solution [17]. Tool support for this class of Petri nets is under development, which, once finished, even more eases the modelling and evaluation process.

References

[1] Heijenk G 1995 Connectionless communications using the asynchronous transfer mode *PhD Thesis* University of Twente

[2] Neuts M F 1981 *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach* (Baltimore, MD: John Hopkins University Press)

[3] Haverkort B R, van Moorsel A P A and Speelman D-J 1994 Xmgm: performance modeling using matrix geometric techniques *Proc. 2nd Int. Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '94) (Durham, NC)* (Los Alamitos, CA: IEEE Computer Society Press) pp 152–7

[4] ITU-T Draft Recommendation I.364 1993 *Integrated Services Digital Network (ISDN), Overall Network Aspects and Functions–Support of Broadband Connectionless Data Service on B-ISDN* (International Telecommunication Union)

[5] Heijenk G and Niemegeers I G 1992 Variable bandwidth connections for a connectionless service on ATM–Performance modelling and evaluation *IFIP Trans. C-4, Proc. IFIP TC6 Workshop on Broadband Communications (Estoril)* ed A Casaca (Amsterdam: North-Holland) pp 361–72

[6] Kershenbaum A, Kermani P and Grover G A 1991 MENTOR: An algorithm for mesh topological optimization and routing *IEEE Trans. Commun.* **39** 503–13

[7] Kershenbaum A 1993 *Telecommunications Network Design Algorithms* (New York: McGraw-Hill)

[8] Jain R and Routhier S A 1986 Packet trains–measurements and a new model for computer network traffic *IEEE J. Select. Areas Commun.* **4** 986–95

[9] Gusella R 1991 Characterizing the variability of arrival processes with indexes of dispersion *IEEE J. Select. Areas Commun.* **9** 203–11

[10] Wong J W 1993 Performance modelling of ATM-based networks *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssysteme* ed B Walke and O Spaniol (Berlin: Springer) pp 15–26

[11] Heijenk G J, van Moorsel A P A and Niemegeers I G 1992 Performance of a connectionless protocol over ATM *Proc. Int. Workshop on Advanced Communications and Applications for High Speed Networks (IWACA '92)* (Munich: Siemens) pp 123–30

[12] Devinatz A 1968 *Advanced Calculus* (New York: Holt, Rinehart and Winston)

[13] Leland W E, Taqqu M S, Willinger W and Wilson D V 1994 On the self-similar nature of Ethernet traffic *IEEE/ACM Trans. Networking* **2** 1–15

[14] van Dieten W J 1994 Performance evaluation of a connectionless protocol over ATM using matrix geometric methods *Technical Report* University of Twente

[15] Nelson R 1991 Matrix geometric solutions in Markov models: a mathematical tutorial *IBM Research Report* RC 16777

[16] Jain R 1991 *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling* (New York: Wiley)

[17] Haverkort B R 1995 Matrix-geometric solution of infinite stochastic Petri nets *Proc. Int. Computer Performance and Dependability Symp.* (Los Alamitos, CA: IEEE Computer Society Press) pp 72–81