The Internetworking of Connectionless Data Networks over Public ATM: Connectionless Server Design and Performance

Duke P. Hong, Brett J. Vickers, Tatsuya Suda and Carlos Oliveira

Technical Report No. 94-41

28 October 1994

Department of Information and Computer Science
University of California, Irvine, CA 92717-3425
(phone) 714-824-5474 (fax) 714-824-4056 (email) suda@ics.uci.edu

Abstract

The transport of connectionless data is necessary for the compatibility of emerging wide area ATM networks with existing computer networks. This paper investigates the design and performance of a virtual connectionless network overlayed on top of cell-based connection-oriented ATM networks. The virtual connectionless network is made possible by utilizing connectionless servers. Two packet forwarding techniques used by connectionless servers are examined. Forwarding connectionless packets in a cell-by-cell manner (streaming mode) is shown to result in lower packet loss and end-to-end delay than packet-by-packet forwarding of connectionless data (reassembly mode). A design and implementation are presented for a streaming mode connectionless server capable of a maximum throughput of 100 Mbps.

This research is supported in part by grants from the University of California MICRO program, Nippon Steel Information and Communication Systems Inc. (ENICOM), Hitachi Ltd., Hitachi America, Tokyo Electric Power Company, and Hewlett Packard (HP).

1 Introduction

The emergence of a high speed, wide area ATM network presents both opportunities and challenges to network service providers. While emerging ATM networks must be capable of supporting future applications such as digital voice and video, many of the design choices made to optimize ATM for voice and video render the transport of traditional connectionless data traffic from LANs and MANs difficult. If ATM is to be widely accepted in the wide area arena, it must efficiently support high speed data applications and be able to interconnect existing LANs and MANs.

There are a variety of approaches to providing connectionless data service in ATM networks. A simple approach is to use the connection oriented service of ATM to interconnect pairs of connectionless data networks. The example in Figure 1(a) illustrates an interworking unit (IWU) in one LAN maintaining a connection to another LAN's interworking unit. Note that in order to send packets to a third LAN, a separate connection is required. This approach is called the *indirect* approach since the network only indirectly achieves connectionless service through the use of connections already provided by ATM.

Within the indirect approach, the type of connections used to support connectionless service may vary. One possibility is to use permanent virtual connections, or PVCs, between each pair of ATM endpoints (*e.g.*, interworking units or end systems) that require communication. PVCs are particularly useful in situations where traffic between the two endpoints is known to have regular characteristics or where connection establishment overhead cannot be tolerated. The drawback of PVCs is that they consume network resources even when idle. An alternative to PVCs is switched virtual connections, or SVCs. An SVC can be established upon the reception of the first packet destined for a given network and terminated after some period of inactivity. The use of SVCs reduces the waste of network resources while at the same time imposing a delay on packet transmission due to connection setup. It is generally agreed that the use of SVCs will outweigh the use of PVCs, mostly due to the fact that they provide more flexibility in determining the topology of a network.

Variants of the indirect approach have been adopted by both the ATM Forum and the Internet Engineering Task Force's IP over ATM working group. The ATM Forum has focused its attention on the provision of connectionless service through LAN emulation [1]. The basic idea of LAN emulation is to make an ATM end system appear to the network as a typical IEEE 802 LAN end system. This is achieved through the implementation of a specialized MAC sublayer, tailored specifically for ATM. Within the ATM MAC sublayer proposed by the ATM Forum exists such functionality as connection management, signalling, address resolution, and MAC-layer encapsulation. By emulating the IEEE 802 interface, the LAN emulation approach allows ATM end stations to interconnect with other networks through bridges or routers in much the same way that systems in Ethernets and token rings do now.

The IETF has taken a slightly different approach from the ATM Forum, opting instead to support connectionless service through the network layer rather than through the MAC sublayer. The IETF solution consists of encapsulating IP frames with IEEE 802.2 LLC and IEEE 802.1a SNAP headers [2]. The encapsulated IP frames are then stored in the payload of an AAL 5 PDU for subsequent segmentation and transmission via the ATM network. Address resolution in the IETF model of connectionless service is achieved through the use of ARP servers located in each IP subnet [3]. Interconnection of isolated ATM networks is realized through the use of IP routers.

The IETF and ATM Forum are focusing (at least initially) on providing ATM connectionless service in the local area environment with bridges and routers serving as interconnection devices. However, in order to efficiently support connectionless service in a *public* wide area ATM network setting, special attention must be paid to scalability. One method that addresses scalability is the *direct* approach to connectionless service [4]. This approach is the focus of the remainder of this paper. It is based on the deployment of connectionless servers (CLS) throughout the network, forming a virtual connectionless network overlayed on the connection-oriented ATM network. Connectionless servers act as packet routers, routing packets to other connectionless servers until the destination is reached. These connectionless servers are attached strategically to a subset of switches within the network. The topology of the virtual connectionless network may take any number of forms including ring, mesh, and star depending on the desired efficiency and robustness [5]. An example configuration is shown in Figure 1(b). Compared to the indirect approach in Figure 1(a), a LAN needs only one open connection to a connectionless server in order to send packets to one or more destination LANs. This method is called the direct approach because the ATM network provides the connectionless service directly to the user; the user is responsible only for sending encapsulated connectionless data into the virtual connectionless network. The direct approach has been suggested as a means of supporting connectionless services similar to SMDS [6].

The direct approach has several advantages over the indirect approach in a wide area environment. First, it is inherently scalable. Fewer connections, and thereby fewer network resources, are needed to interconnect LANs in the direct approach. Network resources such as bandwidth are concentrated on the connections between connectionless servers as opposed to being dispersed throughout the network as in the case of the indirect approach.

Second, the direct approach relieves the local area network customers of responsibility for making routing decisions by forcing that responsibility onto the ATM network. This simplifies the use of connectionless service for the user since little or no special hardware or software support on the user's side is required. Furthermore, the user of connectionless service requires only one connection into the virtual connectionless network, as opposed to the numerous connections required by the indirect approach. This may aid the user in terms of resource usage and connection setup overhead.

Finally, the connections between connectionless servers in the direct approach have better traffic characteristics than connections between LAN pairs since they aggregate data from a number of data sources. Because of this aggregation, the resulting cell streams are less bursty and more predictable, yielding less cell loss at ATM switches.

The direct approach, however, does have its disadvantages. For one, the use of connectionless servers increases the time required to transmit connectionless data from source to destination. They also increase the likelihood of network congestion between servers. This problem has been studied by a number of researchers, and the general conclusion is that some form of bandwidth management should be applied on inter-CLS connections. For instance, the virtual circuits may reserve some amount of bandwidth and adjust it slowly as demand changes [7]. Alternatively, bandwidth may be borrowed from other unused connections on a more flexible basis whenever traffic flows are heavy [8].

Another disadvantage is that, depending on the routing protocol used, packets may take different paths and arrive at their destinations out of sequence. This is particularly true in the case where the routing protocols used to provide direct connectionless service are dynamic (*e.g.*, OSPF). Nevertheless, out-of-sequence packets are part and parcel of any connectionless service offering, and higher layers are adept in responding to it.

To summarize, the indirect approach is the simplest to implement but has problems with scalability to public, wide area environments. It is the most appropriate approach for small ATM networks (*e.g.*, virtual LANs) where connection cost is not as important a factor. The direct approach requires the most effort on the part of the network service provider, but it is the most scalable of the approaches. Furthermore, it provides the ATM endpoint with a single connection onto which it can send all connectionless traffic, and it shapes traffic in such a way that it is more easily accommodated by an ATM network. In a public ATM setting where a large number of interconnected LANs will exist, the direct approach is the most appropriate.

Previous work by the authors has focused on buffering techniques and a preliminary architecture for connectionless service [9]. This paper focuses specifically on the design, implementation and performance of connectionless servers for the direct approach. In section 2, issues involved in connectionless server design are discussed. In section 3, the two connectionless server forwarding modes described in section 2 are evaluated via simulation. In section 4, a design for one of the connectionless servers based on one of the forwarding modes is specified. Finally, observations are summarized in section 5.

2 Connectionless Servers

Connectionless servers are network devices attached to or incorporated into ATM switches as shown in Figure 2. Cells requiring connectionless service are switched to the connectionless server where they are processed. After being processed, the cells are transmitted on a connection back to the switch where they are switched to the next connectionless server or interworking unit. Interworking units, which reside at the perimeter of the ATM network, encapsulate connectionless packets from LANs and MANs using a framing protocol such as CLNAP [10]. The packets are then segmented according to a given ATM adaptation layer (AAL) protocol and sent to connectionless servers via virtual connections. After being routed through a series of connectionless servers, the cells of each packet are received at the destination interworking unit where the packets are reassembled, stripped of their CLNAP headers, and forwarded on the connectionless data network using the appropriate MAC protocol. This section discusses the framing of connectionless packets, the forwarding of segmented packets, and the management of traffic flow used for connectionless service. Other issues such as routing algorithms and connectionless network topology are beyond the scope of this paper.

ATM Adaptation Layer

The ATM adaptation layer is responsible for encapsulating and transmitting connectionless packets. As most packets from existing LANs are greater than 48 bytes in length, segmentation of LAN packets must take place at the source interworking units, while reassembly must take place at the destination interworking unit. Additional segmentation and reassembly may also be performed at intermediate connectionless servers. These functions are delegated to the ATM Adaptation Layer.

There are currently two AAL protocols being considered for transporting connectionless data, AAL 3/4 and AAL 5. The primary difference between the two protocols is the presence of a multiplexing identifier (MID) field in the AAL 3/4 cell format. In addition to the MID field, the AAL 3/4 cell also has payload error checksum, sequence number, segment type and length indicator fields. The actual payload size of an AAL 3/4 cell is 44 bytes. AAL 5 cells have none of these fields in their cell format, thus their payload is the full 48 bytes. AAL 5 is more efficient due to its lower overhead and slightly better payload alignment (payload begins at a 4 byte boundary in AAL 5 vs. a 2 byte boundary in AAL 3/4). The primary functional difference between the two schemes is at the protocol level where packet multiplexing takes place.

In AAL 3/4, cells belonging to a packet are identified by their connection (VPI/VCI) and 10-bit multiplexing (MID) identifiers, whereas in AAL 5, cells of a packet can be identified only by their connection identifiers. Thus, in order to cell interleave (*i.e.*, multiplex cells from different packets onto one stream) multiple packets with AAL 3/4, only one connection is needed and packets may be differentiated by their MID values. With AAL 5, cell interleaving is only possible on a virtual path level with packets distinguished by their VCIs. To allow the same level of cell interleaving with AAL 5 as AAL 3/4, 1024 virtual circuits are needed for each hop.

In smaller ATM networks the cost of establishing virtual paths is not prohibitive, so either AAL may be used to implement cell interleaving. For larger ATM networks where the cost of virtual paths is a factor, AAL 5 should not be used with cell interleaving but instead should be used only with packet interleaving (*i.e.*, the unmingled transmission of one packet at a time). Since it includes a MID field, AAL 3/4 can be used with either cell or packet interleaving in either local or wide area ATM environments. Thus, while AAL 5 is slightly more efficient, AAL 3/4 is more flexible. Because of ATM's high transmission rates, the minor bandwidth efficiency gained from AAL 5's smaller overhead may be insignificant. Since the primary focus of the work presented in this paper is large scale public ATM networks directly providing connectionless service, we assume the use of AAL 3/4.

Forwarding Scheme Alternatives

The direct approach to connectionless service requires each connectionless server to make routing decisions and to forward packets to the subsequent server hop. Because packets are segmented and delivered in cell streams, there are two possible methods for forwarding packets at a connectionless server: packet based forwarding and cell based forwarding [11] [12].

In packet based forwarding, incoming cells are reassembled into packets at each connectionless server. The outgoing connection to the next connectionless server is determined for the packet, and the packet is (re)segmented into cells which are delivered to the next hop using the appropriate connection and packet identifiers. This mode of operation shall henceforth be referred to as *reassembly mode* operation.

In cell based forwarding, packets are not reassembled in connectionless servers. Instead, the packet's constituent cells are forwarded one at a time, as they are received, in a cut-through manner to the next hop. When the first cell of a packet arrives, the routing protocol uses the destination interworking unit address stored in the encapsulation header to determine the next hop connectionless server or interworking unit. This routing information is stored in a table so that the connectionless server can simply perform a lookup and determine the next hop for each of the packet's remaining cells. The use of the table allows cells to be delivered before the entire packet has been received and also prevents cells of the same packet from following different routes. This mode of operation shall henceforth be referred to as *streaming mode* operation.

The performance of these two packet forwarding modes is investigated in detail in Section 3.

Flow Control

In a wide area environment, some form of rate control is necessary to prevent short-term congestion and to enforce flow control between connectionless servers. Suppose a connectionless server is capable of processing p cells per second and there are N incoming connections to the server. In order to avoid overloading the server, the generation rate of each source should not, on average, exceed p/N cells per second. This restriction implies that a rate control permitting the transmission of only one cell per N/p seconds be implemented for the virtual circuits leading to the connectionless server. Since all N sources may not always have cells to transmit, it is possible to allow a few of the N sources to send at a rate greater than p/N for short durations without overloading the connectionless server. However, since it is difficult to coordinate the traffic flows of all N sources, and the goal is to minimize cell loss, we have assumed that traffic sources and connectionless servers are limited to transmitting p/N cells per second. The simulations presented in Section 3 assume a rate control device to implement such a flow control. However, the design presented in Section 4 does not consider the implementation of a flow control device since it can be attached externally.

3 CLS Performance

Three design factors have a particularly important impact on the performance of reassembly and streaming mode connectionless servers. The use of buffer space in connectionless servers is one such factor. Since streaming mode servers forward cells as they come in, buffer space is needed only for output rate control. Reassembly mode servers, on the other hand, require buffer space both for packet reassembly and for output rate control. Therefore, for the same amount of buffer space, buffer utilization is greater for reassembly mode servers than for streaming mode servers, increasing their cell loss rate due to buffer overflow. However, reassembly mode servers have better discarding characteristics than streaming mode. When a cell is dropped in a reassembly mode connectionless server, the entire frame to which the cell belonged can be discarded. On the other hand, streaming mode servers can only discard the cells arriving after the loss. Thus, useless connectionless traffic needlessly fills connectionless server buffers downstream. For this reason, reassembly mode may show lower overall packet loss under certain circumstances, while streaming mode may show lower packet loss in others.

Another factor influencing the overall performance of connectionless service is the nature of streaming and reassembly mode operation. Because reassembly mode requires the reassembly of packets at each CLS hop before transmission can begin, it imposes longer end-to-end packet delays. Since these reassembly delays are experienced at each server hop, routes with a greater number of hops experience proportionally longer end-to-end delays. On the other hand, the delay experienced with streaming mode servers is also a function of the number of hops traversed. Since streaming mode servers must interleave cells, interfering traffic may delay their transmission, thereby increasing the departure time between the first and last cells of a packet. This effect not only increases the end-to-end delay, but it also requires larger reassembly buffers in the destination interworking units. Thus, depending on the state of the network, reassembly mode may experience better end-to-end delay than streaming mode and vice versa.

Finally, the interleaving method used on connectionless server output ports is a design factor that may affect the relative performance of streaming and reassembly modes. For instance, the type of interleaving influences packet loss within the ATM network. For instance, cell interleaving may reduce burstiness in the network. Consider a stream consisting of cell interleaved packets, each of which is ultimately destined for different connectionless servers. As the stream is demultiplexed, the resulting streams become less bursty, thereby reducing the overall cell loss experienced in the network. Lower cell loss translates directly into lower packet loss. On the other hand, packet interleaving fares better under the influence of the consecutive cell loss resulting from buffer overflow, either in switches or in connectionless servers [13]. If consecutive cell loss occurs, fewer packets are affected in the case of packet interleaving than in the case of cell interleaving. Since the loss of a single cell renders a packet useless, isolating cell losses to fewer packets is more advantageous than distributing cell losses across several packets. Thus, while cell interleaved traffic benefits from lower burstiness, packet interleaved traffic benefits from better resilience in the face of consecutive cell loss. The better forwarding mode, streaming or reassembly, will depend partly on which interleaving method results in lower overall loss. Reassembly mode connectionless servers may be used with either interleaving method. However, because streaming mode can only use cell interleaving, it must attain better packet loss and delay performance from reduced burstiness or reduced buffer occupancy.

Simulation Model

The network simulation model is composed of two connectionless servers as shown in Figure 3. The first server, *CLS-1*, is fed by N interworking units, each of which is assumed to maintain a PVC (or a long-lived SVC) to the server. In other words, connection setup is not considered in the simulations. Of the N output PVCs from *CLS-1*, one is fed through the network where it experiences network cell loss and delay before reaching *CLS-2*. The remaining $N - 1$ outgoing connections are assumed to be destined for other CLSs or interworking units not shown in the figure. In addition to the connectionless traffic from *CLS-1*, *CLS-2* receives traffic from $N - 1$ additional interworking units. Using this model, the packet delays and loss probabilities have been studied for both a single hop (*IWU* to *CLS-1*) and a multi-hop (*IWU* to *CLS-1* to *CLS-2* to *IWU*) scenario. Both streaming and reassembly mode[1] connectionless servers have been simulated.

The interworking units shown in Figure 3 generate packet interleaved packets with geometrically distributed sizes of minimum size L_{min}, maximum size L_{max}, and average size L. In order to simulate random delays between packets generated by interworking units, the time between the arrival of the last cell of one packet and the first cell of the next packet at *CLS-1* is assumed to

[1]Reassembly mode connectionless servers were simulated using packet interleaving.

Parameter	Value	Description
N	4	Number of input/output ports at CLS
$1/\mu_{Co}$	4 cell slots	Minimum cell interdeparture time at output ports
$1/\lambda_C$	4 cell slots	Average cell interarrival time from IWUs
$1/\lambda_F$	*variable*	Average interpacket delay from IWUs
L	15 cells	Mean packet size
L_{min}	2 cells	Minimum packet size
L_{max}	210 cells	Maximum packet size
B_{CLS}	250/500/1000 cells	Buffer size at CLS
l_{net}	10^{-5}	Network cell loss probability

Table 1: Simulation Parameters

follow a geometric distribution with mean $1/\lambda_F$. Similarly, spacing between cells is drawn from a geometric distribution with mean $1/\lambda_C$.

At connectionless servers, a shared output buffer of size B_{CLS} is employed for both streaming and reassembly mode in order to store incoming cells. Within the connectionless servers, cell loss and subsequent packet loss occurs when B_{CLS} overflows. If a single cell of a packet is lost, the remaining cells of the packet can be discarded. As described earlier, streaming mode servers are only able to discard cells arriving after the lost cell, since preceding cells have already been transmitted. However, reassembly mode servers discard both the successive and preceding cells, since the preceding cells remain in the buffer for packet reassembly.

The destinations of the incoming packets are assumed to be uniformly distributed among the N output ports, each of which is regulated using a simple rate control scheme. The rate control models a general rate control device that restricts the cell interdeparture times to a minimum of $1/\mu_{Co}$. Such rate control prevents traffic overload at the next hop connectionless server.

In addition to cell loss and delay within connectionless servers due to buffering, there can be cell loss and delay within the network. The virtual connection leading to from *CLS-1* to *CLS-2* is affected both by cell loss and delay. The network cell loss is assumed to be random and independent with probability l_{net}. The network delay on the connection between *CLS-1* and *CLS-2* is simulated by a single queue with an interfering Poisson cell stream and a constant service time.

Table 1 summarizes the simulation parameters. The number of ports ($N = 4$) was chosen as a compromise between a realistic value and one that is feasible for simulation. The cell interarrival time and rate control parameters, $1/\lambda_C$ and $1/\mu_{Co}$, were assumed equal to N in order to ensure that cells arriving over N connections would not, on average, overload the connectionless server. The mean packet size L was chosen to be 15 cells. This size is based on the IP MTU of 500 bytes, which has been found to form a significant portion of TCP/IP data traffic [14]. The minimum and maximum packet sizes, L_{min} and L_{max}, are set in accordance with CLNAP encapsulation of TCP/IP packets using AAL 3/4. The buffer size B_{CLS} was chosen to match reasonably sized fast memories of 32 kB and 64 kB. Finally, the high network loss rate ($l_{net} = 10^{-5}$) was chosen to amplify the effect of network cell loss on packet level performance.

In most of the simulation results that follow, the output load on each CLS connection is the parameter against which connectionless server characteristics are measured. The load on each CLS output port is defined as the fraction of time an output port is busy with the transmission or rate control of exiting cells. It therefore equals the number of cells that arrive per unit time (λ) divided by the number of cells that can depart per unit time (μ_{Co}).

$$\rho = \frac{\lambda}{\mu_{Co}} \tag{1}$$

The overall cell arrival rate λ is obtained from the average interpacket delay ($1/\lambda_F$) and average per-packet cell interarrival time ($1/\lambda_C$) as follows:

$$\lambda = \frac{L}{\frac{L-1}{\lambda_C} + \frac{1}{\lambda_F}} \tag{2}$$

Results

In order to capture the impact of different design factors on the performance of streaming and reassembly mode connectionless servers, several performance measures have been obtained. The first such performance measure is packet loss probability for a single CLS hop. See Figure 4. Note that for a buffer size of 1000 cells, the packet loss probabilities are virtually identical for the two forwarding modes. However, as the load and buffer sizes decrease, the difference in packet loss between streaming and reassembly mode appears. It becomes readily apparent that reassembly mode servers have worse packet loss characteristics than streaming mode servers. In reassembly mode servers with small buffers, packets reside in the buffer for longer periods of time as they are reassembled. This longer buffer occupancy time results in a greater propensity for buffer overflow as other packets arrive. Streaming mode servers achieve better packet loss rates simply because cells flowing through them are buffered only for rate control purposes.

End-to-end packet loss probability is a second measure by which design factors can be studied. Figure 5 illustrates the overall loss rate of packets that traverse both *CLS-1* and *CLS-2*. This figure shows that streaming mode packet loss rate remains less than or equal to that for reassembly mode for all buffer sizes and loads. Two important conclusions can be drawn from this observation. First, recall that reassembly mode servers are capable of discarding entire packets when a single cell is lost, whereas streaming mode servers can only discard the remainder of a packet. According to the results in Figure 5, this effect is not important enough to significantly improve the overall packet loss rate of reassembly mode in comparison to streaming mode. The shorter buffer occupancy time of cells in streaming mode servers is clearly more important in determining end-to-end packet loss than the beneficial discarding characteristics of reassembly mode servers. Second, recall that streaming mode servers may have higher packet loss than reassembly mode due to consecutive cell loss. Based on the results in Figure 5, however, one can conclude that the impact of consecutive cell loss is only a minor one. It does not significantly diminish the performance of streaming mode relative to reassembly mode. Thus, packet loss in reassembly mode servers is only influenced to a slight extent by improved packet discarding, while streaming mode servers appear not to suffer significantly from the effects of consecutive cell loss.

The impact of network cell loss is also evident in Figure 5. With $l_{net} = 10^{-5}$, the network cell loss has very little effect on the packet loss when the packet loss probabilities are larger than 10^{-4}. However, when packet loss is less than this, both streaming and reassembly mode connectionless servers suffer in equal amounts from loss in the network. Although one might assume that reassembly mode would be affected less by network cell loss due to its utilization of packet interleaving, the fact that network cell loss was random and nonconsecutive led to a negligible effect on the relative packet loss probabilities of streaming and reassembly modes.

End-to-end delay is yet another useful measure by which streaming and reassembly mode can be compared. Figure 6 illustrates the end-to-end delay of streaming and reassembly mode servers for various buffer sizes and loads. A packet's end-to-end delay is defined as the time between the arrival of its first cell at *CLS-1* and the departure of its last cell from *CLS-2*. The delay for packets traversing reassembly mode servers is consistently greater than that for streaming mode servers. This fact is due simply to the greater amount of time required to reassemble packets before transmitting them to the next hop. Note that for both streaming and reassembly mode servers, delays increase as buffer size increases. This effect is to be expected since packet loss is greater in the case of smaller buffers.

Another important observation obtained from Figure 6 is the narrowing of the delays between streaming and reassembly mode servers as load increases. For low loads or large buffer size where the packet loss probability is relatively low, the difference in delay for the two modes is fairly constant. However, as load increases to a point where packet loss dominates, the difference in end-to-end delay becomes very small. This effect is due to the better discarding characteristics of reassembly mode. Reassembly mode's better discarding characteristics reduce the load on the the next hop server by discarding entire packets when a single cell is lost. This reduction in load in turn improves the delay of packets that have not been discarded, thereby narrowing the gap between streaming and reassembly mode end-to-end delays.

The effect of cell interleaving on packet delay can be seen by examining the output duration of packets. The output duration of a packet is the time between the departures of its first and last cells from a connectionless server. Recall that streaming mode servers have the potential to increase the output durations of packets as the number of hops grows. Figure 7 illustrates how the average output duration changes in relation to the input duration (*i.e.*, the time between the arrivals of the packet's first and last cells) at *CLS-1*. Reassembly mode shows little evidence of increased output duration since it utilizes packet interleaving. Cells from reassembly mode packets are transmitted unmingled with cells from other packets. On the other hand, streaming mode, which uses cell interleaving, exhibits at *CLS-1* output durations that are 50–70% longer than the input durations. As the packets arrive at *CLS-2* their average output duration becomes 60–85% longer than their original input duration at *CLS-1*. As expected, the output duration increases as more hops are traversed. However, the diminishing rate of increase was unanticipated. We hypothesize that the abatement in the increase of output durations as hop count grows is due to the cell interleaved nature of the stream arriving from *CLS-1*. As the stream is demultiplexed in *CLS-2*, cells destined for different output ports are separated. Because cells from interfering packets on other ports are able to fill the spaces in the demultiplexed traffic stream, output duration fails to increase significantly. Thus, as the number of hops increases, one finds the output duration for streaming mode servers stabilizing.

Finally, the effect of the number of ports (N) on the performance of streaming and reassembly mode operation is interesting to note. Figures 8 and 9 illustrate the effect of increasing N on packet loss and delay. The load on each output connection is kept constant ($\rho = 0.75$), while the total buffer space increases proportionally with the number of ports. Packet loss decreases, while packet delay increases. The reason is increased buffer sharing. As N increases and the buffer space per connection remains constant, there is more overall buffer space for each connection to share. Note however that as N increases, reassembly mode exhibits a slower improvement in packet loss and a faster worsening of delays than streaming mode. Reassembly mode's poorer relative performance is a result of the fact that increasing the number of ports increases the number of packets that fill buffers while they await reassembly and transmission.

In summary, the following generalizations can be made:

- Streaming mode connectionless servers provide superior performance to packet interleaving reassembly mode connectionless servers. They achieve lower end-to-end delay and packet loss rates for all loads.

- While the better discarding characteristics and consecutive cell loss resilience of reassembly mode make it seem competitive with streaming mode, these characteristics do not significantly impact the overall performance of reassembly mode servers.

- Random cell loss in the network does not affect one forwarding mode to a greater degree than the other.

- Although the output duration is longer under cell interleaving than under packet interleaving, this duration grows at a slower rate as more hops are traversed.

- As N increases, the packet loss rate of streaming mode becomes substantially less than that for reassembly mode.

4 Streaming Mode Connectionless Server Design

Simulation results have shown that streaming mode provides lower packet loss and and smaller delays than reassembly mode. Thus, we have concentrated our architectural efforts on designing and implementing a streaming mode connectionless server.

The goal of implementing a connectionless server is to empirically study the performance of streaming mode connectionless servers. Since the primary objective of this empirical analysis is to investigate the performance of connectionless servers, a number of simplifications have been made in order to avoid the complexity of designing components irrelevant to our research.

- *Link Speed*: It is not our goal to produce the fastest possible connectionless server. The target throughput of the connectionless server is 100 Mbps. Higher throughput can be achieved with improved VLSI implementations.

- *Permanent Virtual Connections*: The streaming mode connectionless server assumes that virtual connections are established permanently between connectionless servers. Switched virtual circuit operation requires more sophisticated routing, signaling and connection management protocols, issues which are beyond the scope of this paper.

- *Fixed Routing*: It is assumed that existing routing protocols (*e.g.*, OSPF) may be applied on top of connectionless servers. The authors have chosen, however, to use a fixed routing scheme in order to avoid this complexity.
- *Address Resolution*: The authors are implementing the streaming mode connectionless server using a flat address space. Furthermore, since the connectionless servers are to be tested in a small environment, only the least significant 16 bits of the destination interworking unit address are used to perform the next-hop lookup.

These assumptions have simplified server design and development without impeding study of all pertinent aspects of research.

For a throughput of 100 Mbps, a cell must be processed within 4.2 μs. Such time requirements place strong restrictions on server design decisions, mandating the use of application specific VLSI- or FPGA-based chip designs, or efficiently designed discrete device layouts. The authors have opted for HCMOS-based discrete devices in the first stage of implementation in order to foster on-the-fly modifications and to reduce implementation costs.

Figure 10 illustrates the functional model of a streaming mode connectionless server. Its major components include:

Forwarding Table Stores information for each packet currently being processed by the CLS.

Forwarding Table VPC Map Determines offset into the forwarding table based on the incoming VPI/VCI of each cell.

Address Resolution Map Performs the routing function by translating destination interworking unit addresses into outgoing VPI/VCI/MID combinations.

Protocol Engine Coordinates communication between the network and each of the components listed above.

An output rate control device is not implemented within the connectionless server, as such a device can be implemented externally. Detailed descriptions of each of these units follow.

Forwarding Table

The Forwarding Table keeps track of all packets that are currently in the process of being transferred through the connectionless server. It is a large memory containing for each packet an outgoing VPI/VCI, an outgoing MID, an expected cell sequence number, and other miscellaneous information about the packet such as the segment type (ST) and the deliver damaged frame (DDF) option. See Figure 11. A page of the Forwarding Table is indexed by the internal connection number provided by the Forwarding Table VPC Map (*i.e.*, there is one page per incoming connection). Each page is further indexed by the 10-bit MID field of the incoming cell. The MID field, provided by AAL 3/4, is used to distinguish up to 1024 unique packets on a single connection.

Forwarding Table VPC Map

The Forwarding Table VPC Map, depicted in Figure 12, performs a quick lookup on the incoming VPI/VCI of each cell and produces a page number for the Forwarding Table. This page number is then used as an index into the Forwarding Table memory, which stores a list of outgoing connection identifiers and other control information. The Forwarding Table VPC Map consists of a 48-bit content addressable memory (CAM) and control logic allowing access to it. The use of content addressable memory allows rapid lookup with a deterministic retrieval time. Each entry in the CAM consists of 28 bits for the incoming VPI/VCI pair and up to 20 bits for the corresponding forwarding table page number. The Forwarding Table VPC Map's control logic receives a START signal from the Protocol Engine when the data lines containing the VPI/VCI values from the incoming cell are ready to be looked up. Once the CAM has performed the lookup, the control logic sets the DONE signal to notify the Protocol Engine that the connection number is ready to be read from the register. The Protocol Engine then sends a HANDSHAKE signal to the Forwarding Table VPC Map, at which time the DONE signal is set low once more, and a new translation can take place.

Address Resolution Map

Upon receiving the first cell of a packet, the Address Resolution Map performs two functions. First, the Address Resolver component performs a lookup on the cell's destination internetworking unit address, calculates the next hop, and returns the VPI/VCI of the proper outgoing connection. Since a small, flat address space is being used, a simple content addressable memory suffices for address resolution. In fact, the implementation is similar to that of the Forwarding Table VPC Map except that it maps 16 bits of the destination address onto a 28-bit outgoing VPI/VCI value. Therefore, the only difference between the two is the format of the CAM entries and the setting of the mask values. An implementation supporting a hierarchical address structure can be supported using CAMs specially designed for routers [15].

Second, the MID Manager component allocates a unique multiplexing identifier for the packet on the outgoing connection. Within the MID Manager, each outgoing connection is allocated a page of pointers in static RAM with one entry per MID (*e.g.*, entry i is for MID value i). Available MIDs are managed by a single linked list as shown in Figure 13. MIDs that are unavailable are implicitly designated by not being in the available MID list. The entry in location 0 contains a head pointer to the first available MID entry in the list for the outgoing connection, while the entry in location 1023 points to the tail of the list. MIDs are allocated by removing the corresponding entry from the head of the list, and are freed by adding the entry to the tail of the list.

Protocol Engine and Network Interface

The Protocol Engine and Network Interface are the set of control logic and communication subsystems that coordinates and controls the data flow in the connectionless server. They are illustrated in Figure 14. A three-cell memory is utilized to allow pipelined operation of the reception unit, protocol engine unit, and transmission unit. Each of the these three units is attached to one of three cell memories through a contention-free memory access interface. The Memory Access Interface Controller rotates the memory assignments among the three units at cell slot intervals, thereby enabling pipelined processing of cells.

The Protocol Engine is largely a collection of control logic and registers responsible for coordinating the efforts of the Forwarding Table VPC Map, the Address Resolution Map, the Forwarding Table, and the Memory Access Interface. The Protocol Engine triggers a lookup on the incoming VPI/VCI of every cell in the Forwarding Table VPC Map. The page number produced by the Forwarding Table VPC Map is used along with the incoming MID of the cell to index into the Forwarding Table RAM. When the first cell of a packet (BOF cell, or beginning of frame cell) arrives, it is dispatched by the Protocol Engine to the Address Resolution Map for next-hop destination lookup and outgoing MID allocation. This information is stored in the Forwarding Table for use by subsequent cells of the same packet. To handle continuation of frame (COF) and end of frame (EOF) cells, the connectionless server simply uses information in the Forwarding Table to determine the output port for the next-hop connectionless server. Once next-hop connectionless server for a cell is determined, it is stored into the cell, while the CRCs (HEC and AAL 3/4 CRC-10) are generated as the cell is dispatched. Apart from registers, the Protocol Engine contains comparators for sequence number checks, counters to increment sequence numbers, and a large number of tri-state buffers and transceivers to coordinate internal bus access.

The Reception and Transmission units are based on the ATM Forum's physical layer specification for TAXI-based transport across 100 Mbps multimode fiber [16]. The Reception Unit receives ATM cells from a 125 Mbps 4B/5B TAXI interface. The serial input from the ATM switch is parallelized and written into the Reception Unit FIFO 16 bits at a time. The first four bytes of the cell are simultaneously passed through an 8-bit CRC checker to verify the cell's Header Error Check (HEC) code, while the payload of the cell is passed through a 10-bit CRC checker to verify the cell's AAL 3/4 CRC. If an error is detected in either case, the cell is simply discarded. The Reception Unit's control logic coordinates communication between the TAXI chip, the FIFO, the CRC checker and the Memory Access Interface.

The Transmission Unit reads data 16 bits at a time from its assigned memory cell and passes it to the TAXI transmission interface. The first four bytes of the cell are simultaneously passed through an 8-bit CRC generator to produce the cell's new HEC code, and the cell's payload is passed through a 10-bit CRC generator to produce the AAL 3/4 CRC code. The controller coordinates the generation of the CRCs as well as the transmission of the cell through the TAXI interface. It also communicates with the Memory Access Interface.

The streaming mode connectionless server described above has been fully designed and implemented, proving that such a relatively simple server is not only possible but eminently feasible.

5 Conclusion

Connectionless service provides a means by which high speed data and LAN/MAN interconnection services can be realized. The results derived in this paper can be applied directly to the Internet, particularly as it evolves into an even larger broadband environment. Each of the issues presented in this paper is relevant in wide area interworking environments, particularly in those which seek, as the Internet does, to interconnect widely distributed LANs and MANs.

This paper has presented architectural options for connectionless transport, examined performance issues, and outlined the design of a streaming mode connectionless server. Two packet forwarding techniques, streaming and reassembly modes, were studied via simulation, showing that streaming mode provides substantially lower packet loss and delay under realistic circumstances. Although reassembly mode has some advantages over streaming mode, namely better packet discarding and more flexibility in interleaving, it was shown that these factors have very little effect on overall performance.

In addition to simulation results, a design and implementation of a streaming mode connectionless server were presented. This connectionless server allows the fast lookup of packet information using a content addressable memory to index into a RAM-based table of packet forwarding information. Address resolution is also performed in a simple content addressable memory. With the design presented, the support of a 100 Mbps connectionless server is shown to be easily attainable using off-the-shelf components. The design is easily extended for larger, hierarchical address spaces (*e.g.*, E.164).

A possible extension of this work might include the development of a connectionless server design capable of scaling its throughput. In addition to providing scalable throughput, the design should minimize disruption of service when increasing bandwidth capacity. A potential scalable architecture might employ a number of connectionless server modules, with each module functionally identical to a nonscalable connectionless server. Packets may be distributed to modules by a central packet dispatcher, while address resolution and MID allocation are also performed centrally to avoid conflict on the outgoing connections. The authors have specified the detailed design of a scalable connectionless server in [17].

Another possible extension of this work may consist of the development of bandwidth management schemes for links between connectionless servers. Renegotiating or borrowing bandwidth on these connections when more is required would be useful in a situation where a scalable server is in use. Furthermore, bandwidth management streamlines the efficiency of connectionless service, keeping allocated network resources to a minimal level while at the same time preventing congestion by allowing further reallocations.

In future work, we intend to perform a mathematical analysis of streaming and reassembly mode connectionless servers in order to obtain performance measures that are not captured in simulations. For instance, simulation measurements of loss rates on the order of 10^{-9}, a typically expected cell loss rate for ATM networks, cannot be feasibly obtained. We also intend to obtain empirical benchmarks on the performance of our implemented connectionless server in an ATM testbed environment.

References

[1] P. Newman, "ATM Local Area Networks," In *IEEE Communications Magazine*, pp. 86–98, March 1994.

[2] J. Heinanen, "Multiprotocol encapsulation over ATM Adaptation Layer 5," *NIC Request for Comments 1483*, July 1993.

[3] M. Laubach, "Classical IP and ARP over ATM," *NIC Request for Comments 1577*, Jan. 1994.

[4] CCITT Study Group XVIII, *CCITT Recommendation I.327: B-ISDN Functional Architecture*, 1991.

[5] K. Yamazaki, Y. Wakahara and Y. Ikeda, "Networks and switching for B-ISDN connectionless communications — Issues on interworking of two connectionless services, network topologies and connectionless message switching method," *IEICE Trans. Commun.*, vol. E76-B, no. 3, pp. 229–236, March 1993.

[6] P. Crocetti, L. Fratta, G. Gallassi, M. Gerla, M.A. Marsiglia and D. Romanò, "ATM-based SMDS for LANs/MANs interconnection," In *Proceedings of ISS '92*, vol. 1, pp. 87–91, Oct. 1992.

[7] A. Biocca, G. Freschi, A. Forcina and R. Melen, "Architectural issues in the interoperability between MANs and the ATM network," In *Proceedings of ISS '90*, vol. 2, pp. 23–28, May 1990.

[8] M. Gerla, T.-Y. Charles Tai and G. Gallassi, "Internetting LAN's and MAN's to B-ISDN's connectionless traffic support," *IEEE J. Select. Areas Commun.*, vol. 11, no. 8, pp. 1145–1159, Oct. 1993.

[9] D.F. Box, D.P. Hong and T. Suda, "Architecture and design of connectionless data service for a public ATM network," In *Proceedings of INFOCOM '93*, pp. 722–731, March 1993.

[10] CCITT Study Group XVIII, *CCITT Draft Recommendation I.364: Support of Broadband Connectionless Data Service on B-ISDN*, June 1992.

[11] T. Kawasaki, S. Kuroyanagi, R. Takechi and K. Hajikano, "A study on high speed data communication system using ATM," In *Proceedings of GLOBECOM '91*, pp. 2105–2109, Dec. 1991.

[12] G. Fioretto, T. Demaria, R. Vaglio, A. Forcina, and T. Moro, "Connectionless Service Handling Within B-ISDN," In *Proceedings of GLOBECOM '91*, pp. 217–222, Dec. 1991.

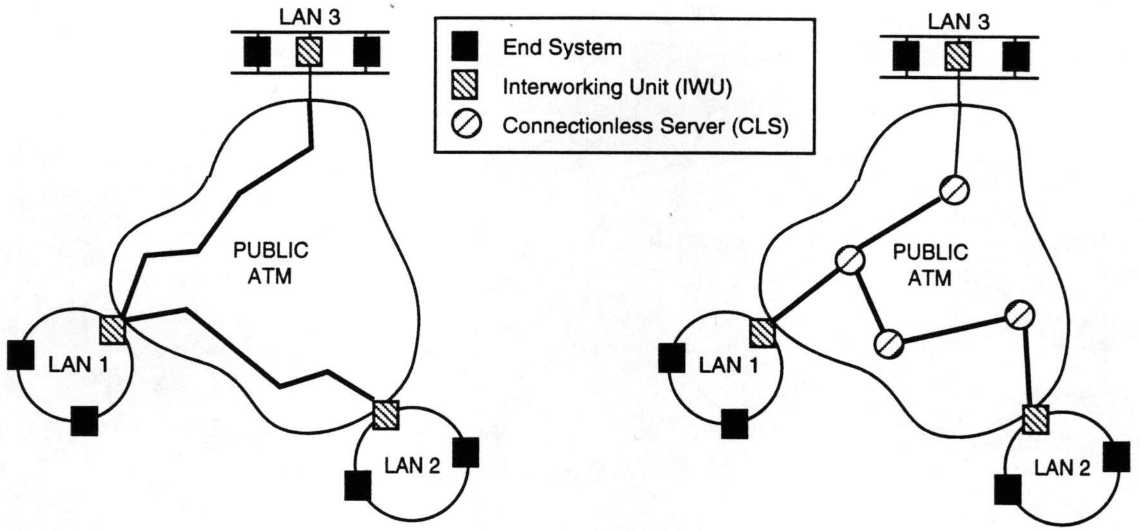
[13] T. Takine, T. Suda, and T. Hasegawa, "Cell Loss and Output Process Analyses of a Finite-Buffer Discrete-Time ATM Queueing System with Correlated Arrivals," In *Proceedings of INFOCOM '93*, pp. 1259–1269, March 1992.

[14] R. Caceres, P. Danzig, S. Jamin and D. Mitzel, "Characteristics of Wide-Area TCP/IP Conversations," In *ACM SIGCOMM '91*, pp. 101–112, Sept. 1991.

[15] A. McAuley and P. Francis, "Fast Routing Table Lookup Using CAMs," In *Proceedings of INFOCOM '93*, pp. 1382–1391, March 1993.

[16] The ATM Forum, *ATM User Network Interface Specification, Version 3.0.* Englewood Cliffs, New Jersey: Prentice-Hall. 1993.

[17] D.P. Hong, B.J. Vickers and T. Suda, "Connectionless Data Server for ATM Networks: Design and Performance," Technical Report #94-10, University of California, Irvine, Jan. 1994.



a) Indirect Connectionless Support

b) Direct Connectionless Support

Figure 1: Direct and Indirect Connectionless Service Architectures

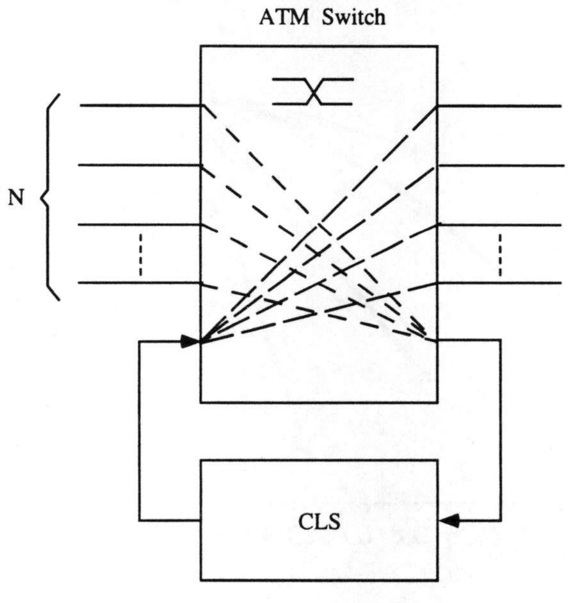


Figure 2: Switch-to-CLS Connection

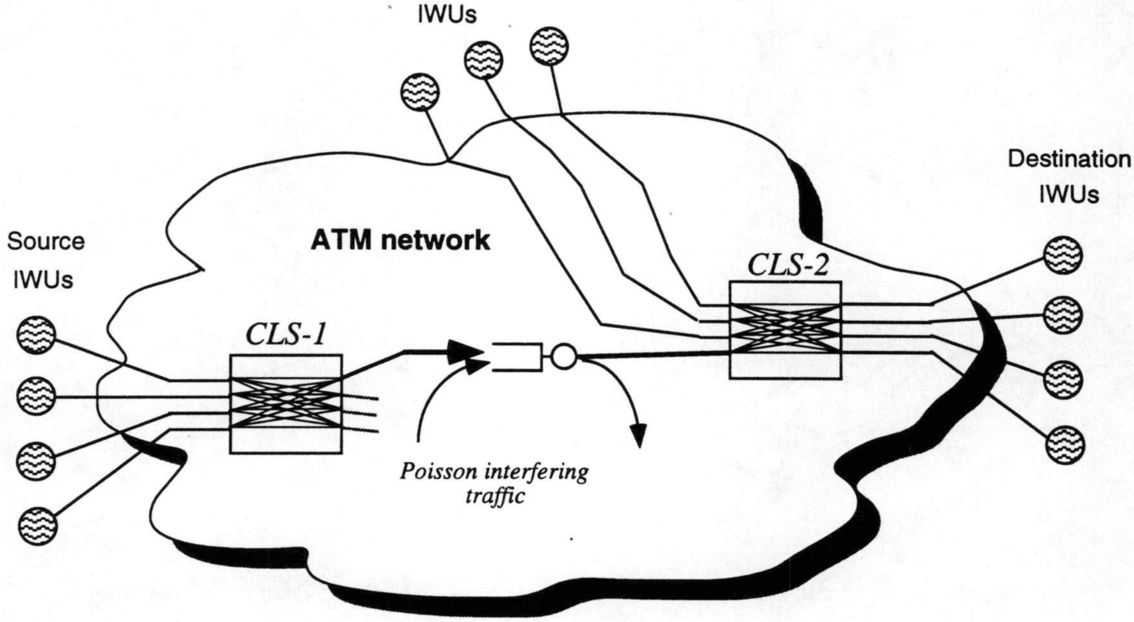


Figure 3: Network Simulation Model

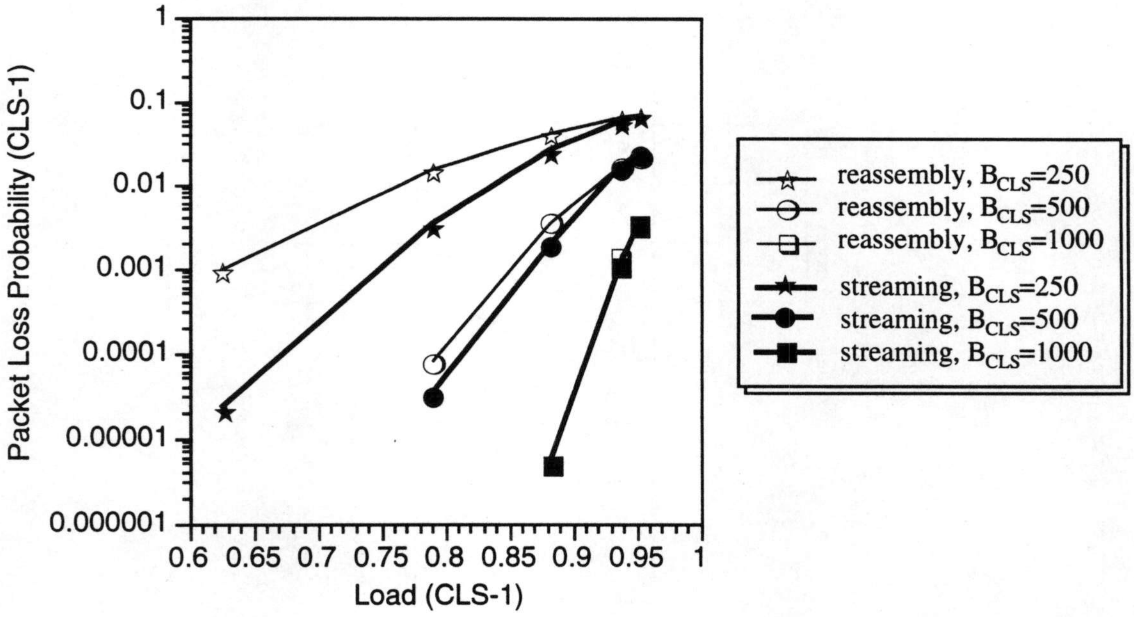


Figure 4: Packet Loss Probability at *CLS-1*

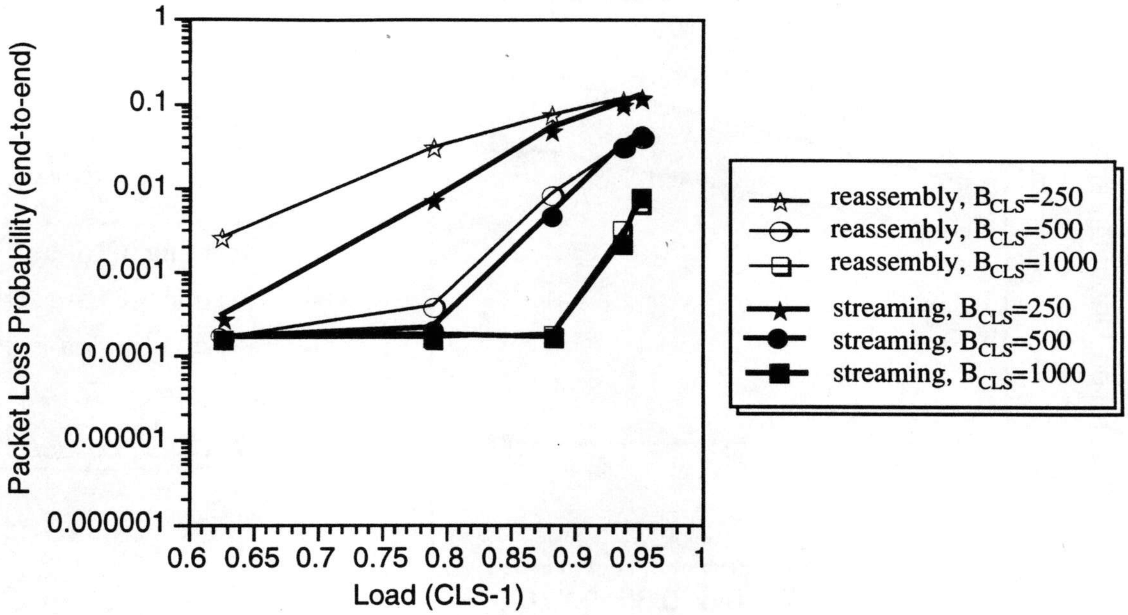


Figure 5: End-to-End Packet Loss Probability

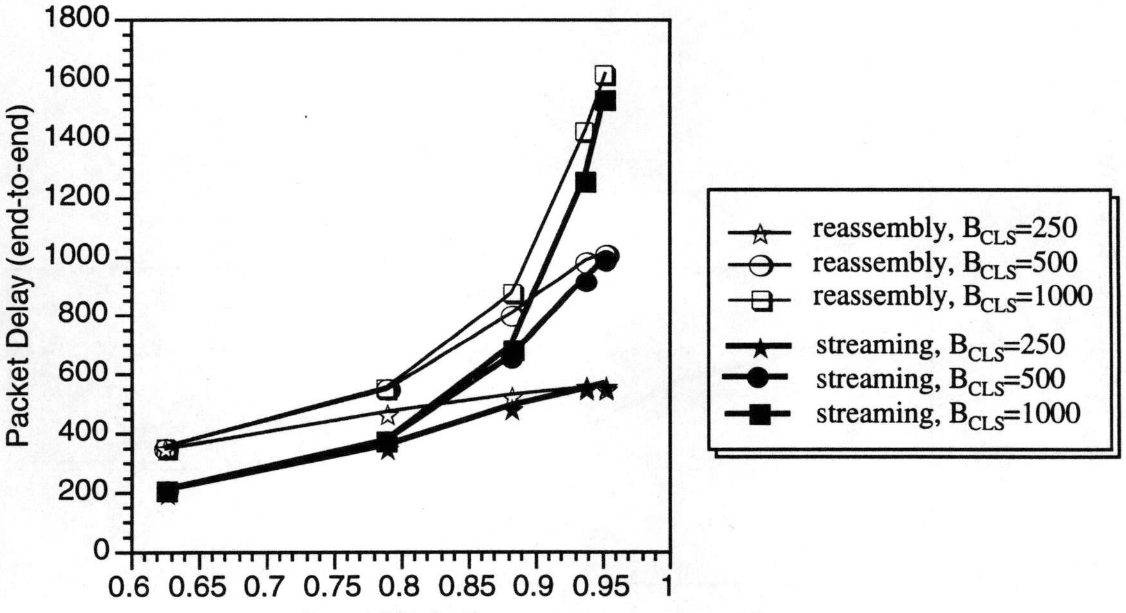


Figure 6: End-to-End Packet Delay

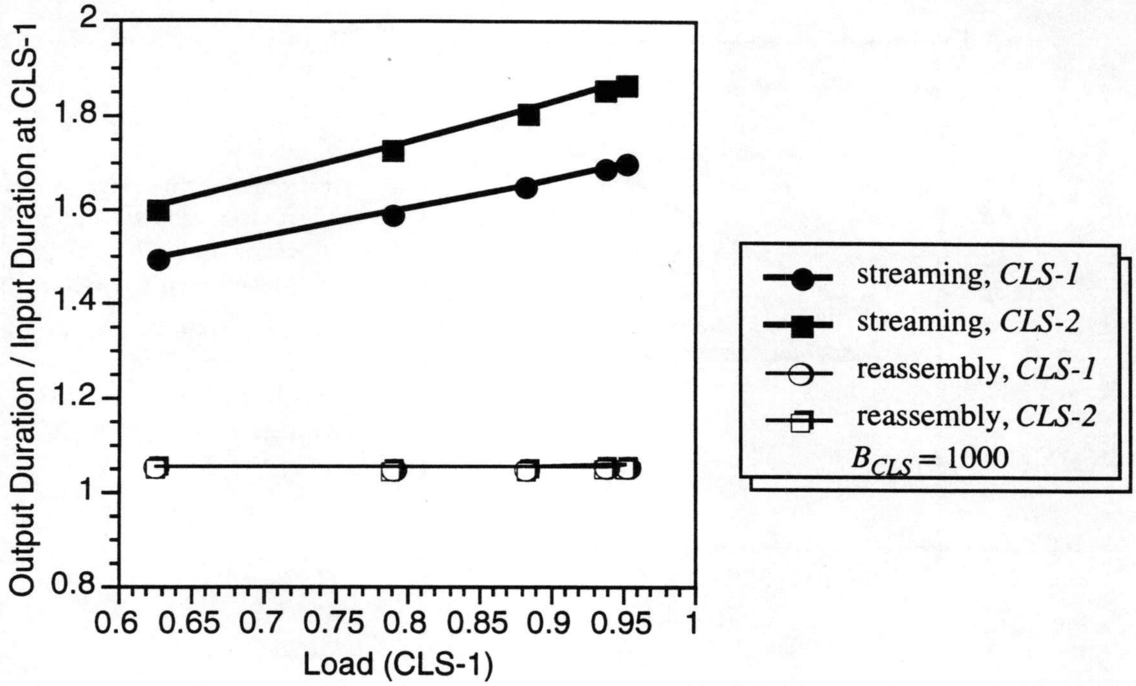


Figure 7: Packet Input Time and Output Times at *CLS-1* and *CLS-2*

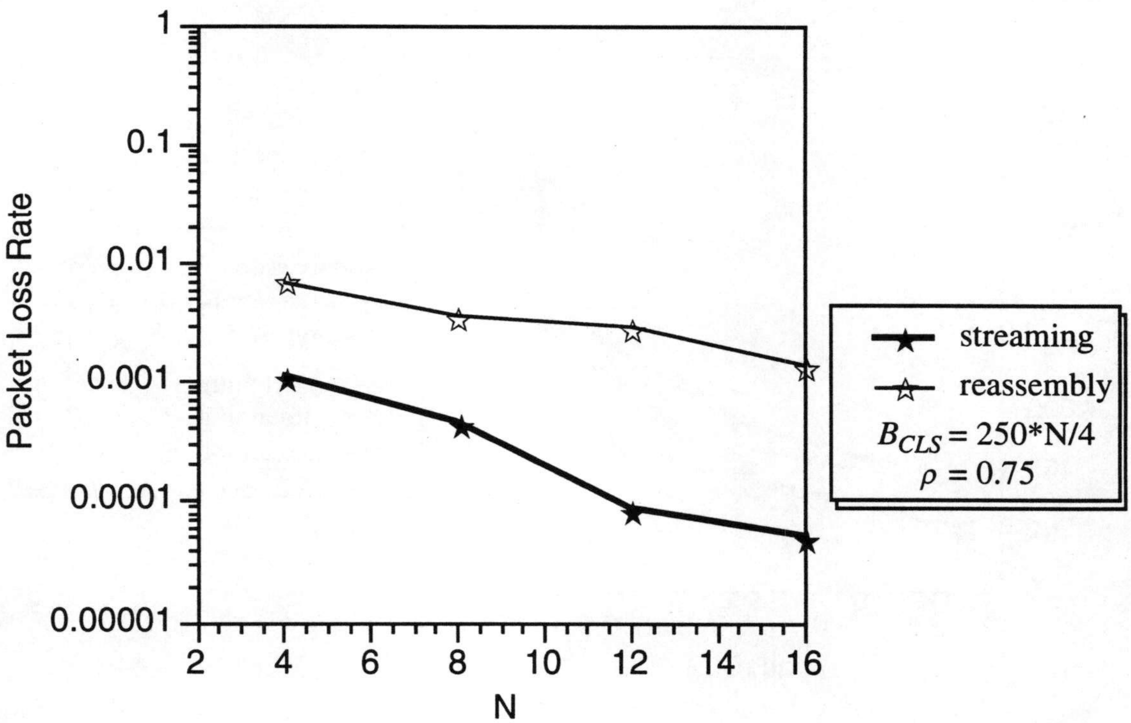


Figure 8: Packet Loss Probability vs. Number of Ports, Single CLS

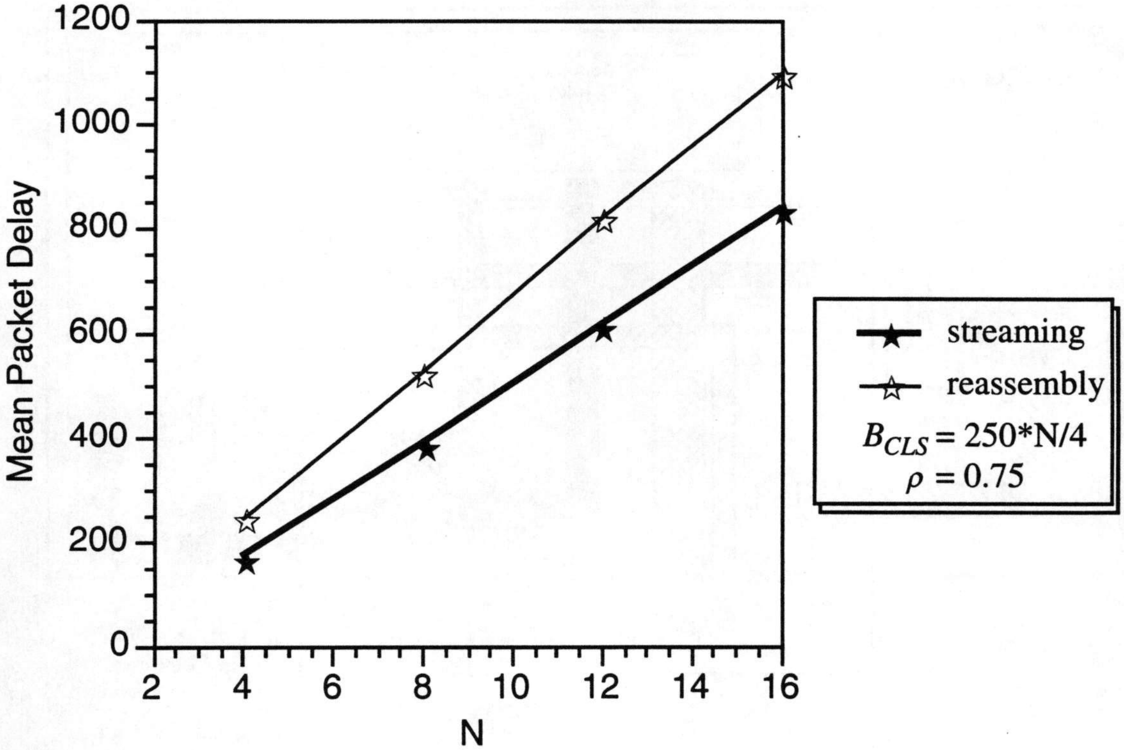


Figure 9: Packet Delay vs. Number of Ports, Single CLS

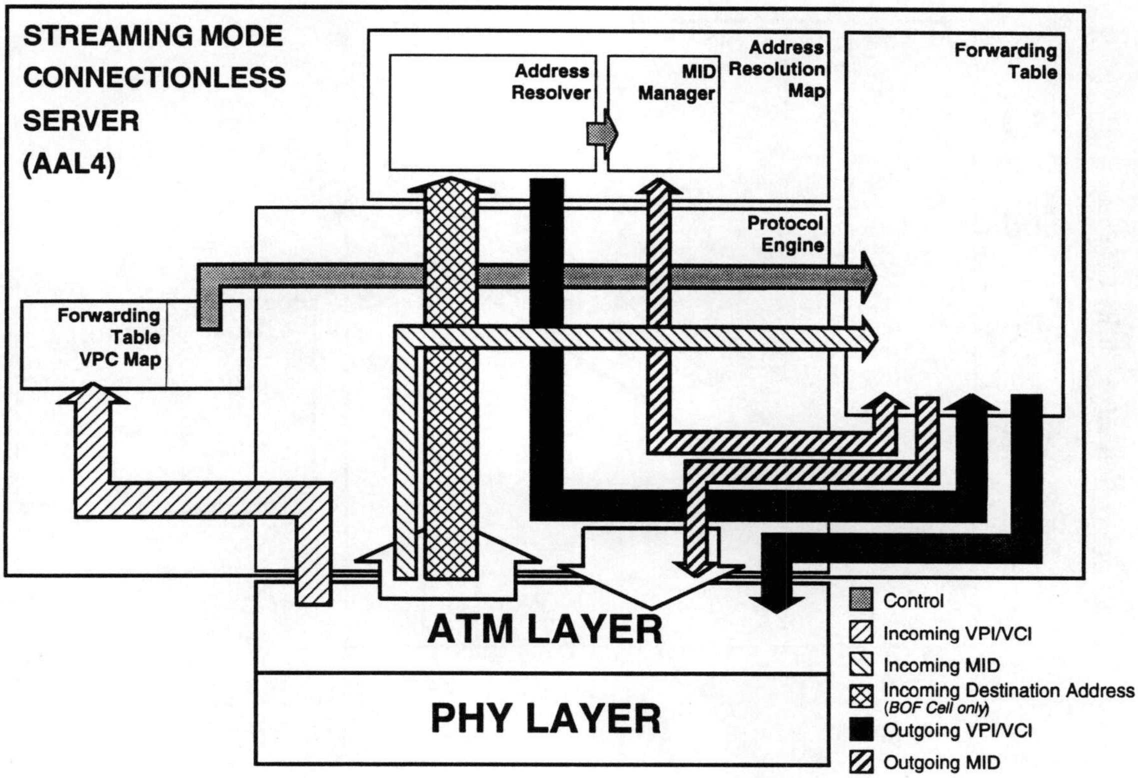


Figure 10: Streaming Mode Connectionless Server Architecture

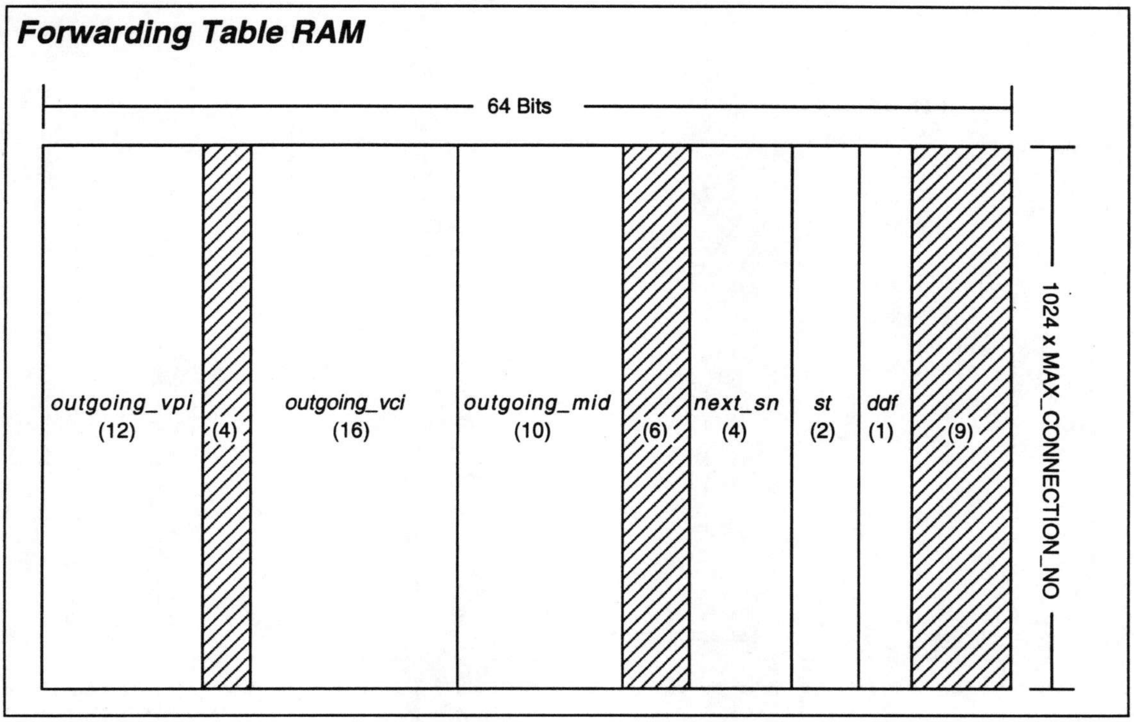


Figure 11: Streaming Mode Forwarding Table Format

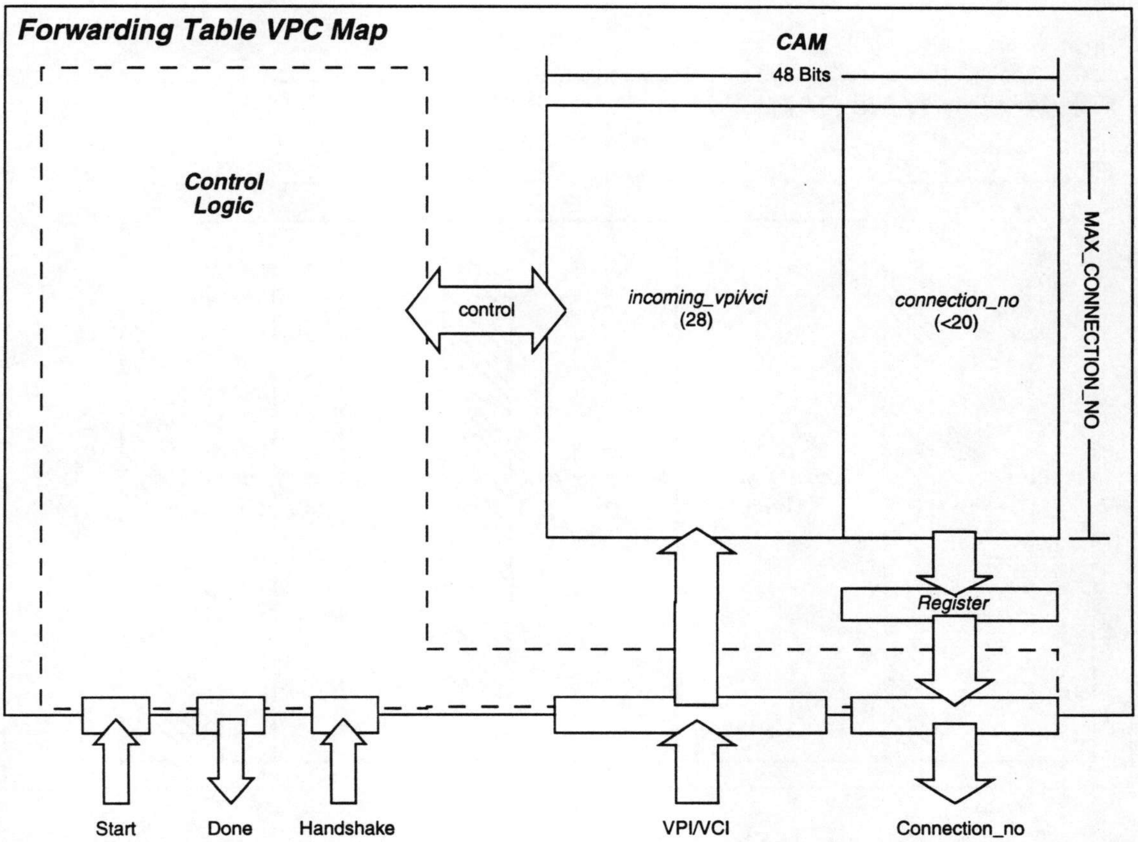


Figure 12: Forwarding Table VPC Map Design

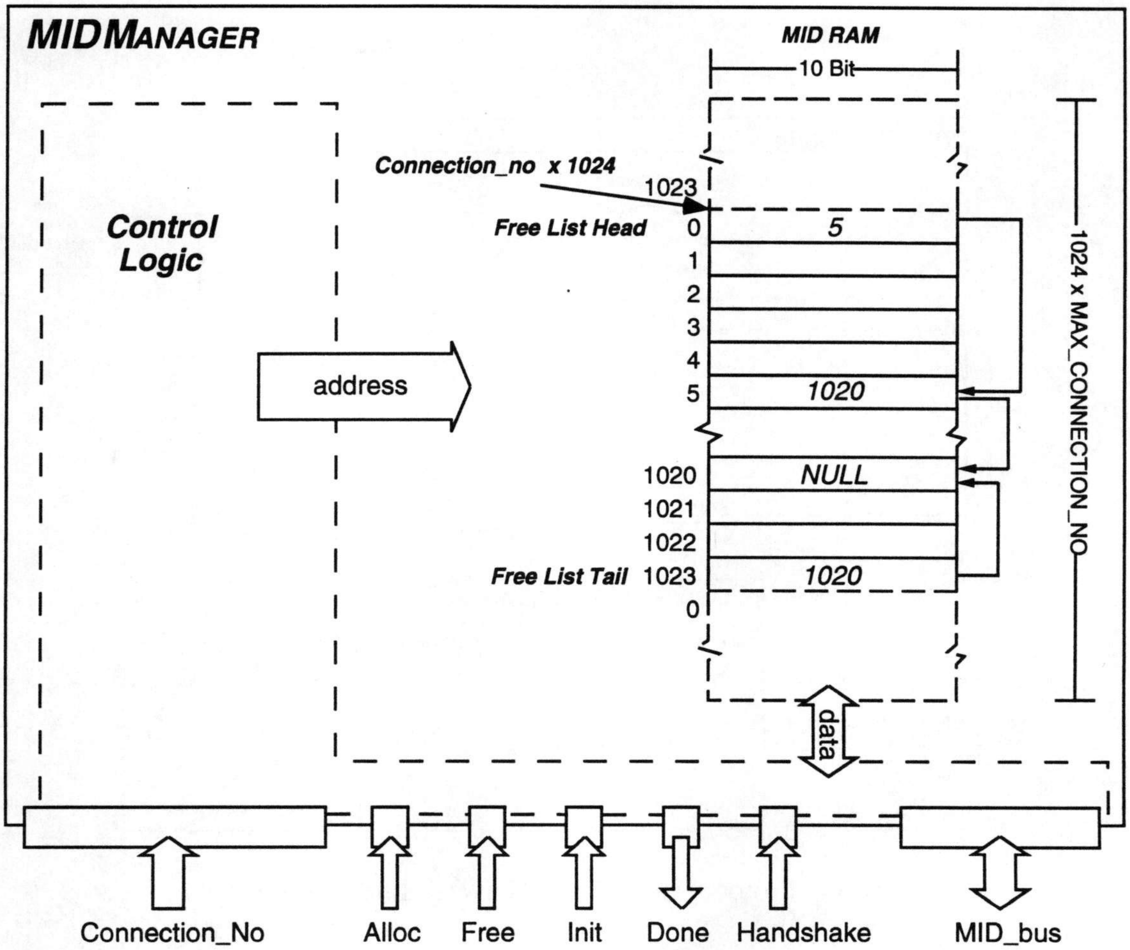


Figure 13: MID Manager Design

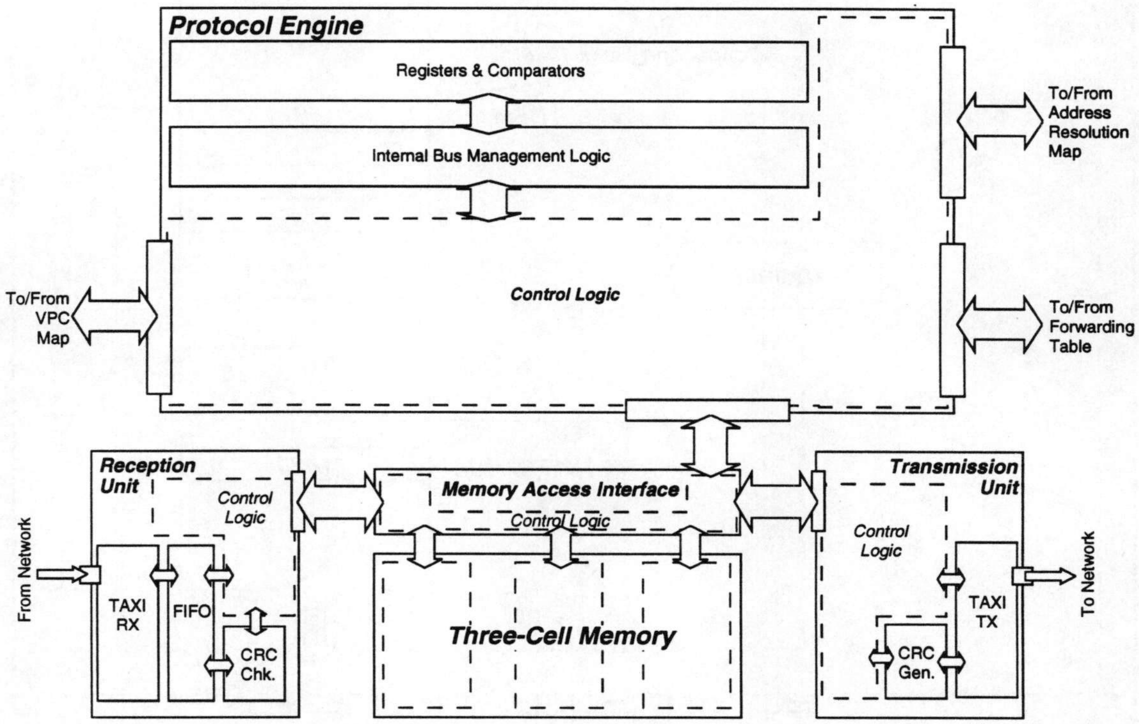


Figure 14: Protocol Engine and Network Interface Design