Architecture and Design of Connectionless Data Service for a Public ATM Network

Donald F. Box, Duke P. Hong and Tatsuya Suda
dbox@ics.uci.edu, duke@ics.uci.edu and suda@ics.uci.edu
Department of Information and Computer Science,
University of California, Irvine,
Irvine, CA 92717-3425
(714) 856-4105 (voice)
(714) 856-4056 (facsimile) *

Abstract

The deployment of a high speed, wide area ATM network introduces a "backwards compatibility" dilemma. While emerging ATM networks must be capable of transporting traffic sources such as digital voice and video, many of the design choices that optimize ATM for voice and video make the transport of traditional connectionless data traffic difficult. This paper proposes the overlay of a virtual datagram network of connectionless servers onto a public ATM network to provide connectionless service. This method provides low transfer delay and efficiently utilizes network resources. Architectural issues are described, and an implementation of a connectionless server is presented.

1 Introduction

Asynchronous Transfer Mode (ATM) is a high speed connection-oriented network that will be used in B-ISDN networks. The ubiquity of Local Area Networks creates an initial application for ATM, both for the interconnection of existing LANs and the service of existing distributed applications that operate over LANs. To adequately interconnect these networks and provide reasonable performance to the applications that utilize them, some provision for interworking ATM and LAN technologies is necessary. However, integrating ATM into a LAN environment introduces a technological paradigm shift with respect to internetworking (*i.e.*, from connectionless IP or CLNP to connection-oriented

---

[*] This material is based upon work supported by the National Science Foundation under Grant No. NCR-8907909. This research is also in part supported by University of California MICRO program.

ATM). Thus, effectively interworking ATM and LANs requires an impedance matching between the connectionless nature of most LANs and the connection-oriented nature of ATM.

Providing connectionless service on an ATM internetwork is a complex task. This is due to the fact that the ATM network departs significantly from traditional packet switching networks in both its switch and protocol architectures and services for which it is optimized. One critical area of departure is that ATM is a connection oriented network. In order to provide connectionless service, a mapping of connectionless traffic on to connections (circuits) must be devised. Another ATM characteristic that makes provision of connectionless service difficult is that ATM cells are small and of fixed size (53 octets), while the frame sizes for traditional networks are much larger and usually variable. For example, the maximum frame size for Ethernet is approximately 1500 octets[1] and that for FDDI is 4500 octets[2]. Thus, segmentation and reassembly are required at the boundaries of the ATM network. ATM employs a different protocol layering architecture from traditional packet switching networks, which further increases the complexity of the problem. At a user-network interface rate of 155.52 Mbps using SONET STS-3c, cells are switched at a rate of one cell per 2.7 μseconds. To support this high speed, ATM utilizes a simplified protocol stack where much functionality has been streamlined for hardware or pushed up to end-to-end protocols.

For a connectionless service scheme to be successful in a wide area public ATM network setting, special attention must be paid to connection management. Excessive establishment and termination of connections can have a considerable impact on the performance of an approach. In a wide area public network, the high end-to-end latency will increase the connection set-up delay, potentially causing large amounts of data to be lost due to buffer overflow. However, a scheme that maintains a large number of permanent connections to accommodate all possible source-destination pairs will wastefully reserve network resources and reduce the overall utilization of the network. [1] Thus, a connectionless service scheme for a public ATM network must efficiently manage the use of ATM connections.

This paper proposes an architecture for connectionless transport on ATM. In Section 2, we discuss the general architectural options for connectionless service. Section 3 presents an overview of our architecture. Section 4 discusses issues in protocol encapsulation and conversion. In Section 5, we describe a possible connectionless server design. We conclude in Section 6 with a summary and discuss open issues.

2 Architectural Issues

A general model for connectionless service across ATM has been outlined by CCITT[3]. However, many of the details are left unspecified and many have multiple alternatives, none of which are clearly superior. In this section, we will examine and compare possible approaches suggested by [3].

---

[1] ATM connections require the allocation of connection identifiers (VPIs and VCIs) at each intermediate ATM switch. The ATM network will reserve some amount of bandwidth at each ATM switch based on the connections it must process. Thus, idle connections can prevent new connections from using a switch. Ultimately, this could prevent the connection from entering the network.

2.1 Direct vs. Indirect Support for Connectionless Data Service

Figure 1 shows the two main approaches recommended by the CCITT to providing connectionless service across an ATM network. The *indirect* approach, as shown in Figure 1(a), requires no explicit support from the public ATM network. Connectionless transport is achieved by maintaining IWU-IWU[2] virtual connections (VCs) at the ATM layer to forward datagrams from the source network directly to the destination network's IWU. The example in Figure 1(a) illustrates that the IWU for LAN 1 maintains a connection to LAN 2's IWU. In order to also send datagrams to LAN 3, LAN 1 must maintain a second connection to LAN 3's IWU. IWU-IWU connections can either be permanent or transient.

Permanent connections can be pre-established for network-network associations that are known to have regular internetwork traffic and/or cannot tolerate the connection establishment overhead. Transient connections can be established upon the first datagram destined for a given network, and terminated after some period of inactivity to allow other calls to enter the network and to reduce connection cost to the user. Permanent connections allow more efficient IWU implementations since no connection activity timers must be maintained and connection management processing can be performed prior to the actual reception of frames. However, permanent connections consume resources at the ATM layer even when idle. Since ATM networks admit connections based on the bandwidth allocation at the ATM switch layer, having a large number of idle connections can prevent other connections from entering the network. This limits the scalability of the indirect approach. For a large number of LANs to interconnect to one another, transient connections must be used. However, this increases the transfer delay for frames destined to LANs for which no connection currently exists.

The second alternative to providing connectionless service is the *direct* approach, where the ATM network directly supports the transport of connectionless data via Connectionless Servers (CLSs). Datagrams (frames) traverse a series of connections between CLSs that form a virtual connectionless network overlayed onto the connection-oriented ATM network. These CLSs must determine the next hop CLS for each frame that is received. This is accomplished by mapping incoming connection and frame identifiers onto those of an outgoing connection based on a destination address in the frame's header. Frames from multiple IWUs destined for the same IWU are multiplexed at an intermediate CLS onto one static CLS-CLS or CLS-IWU connection. Note that in the example in Figure 1(b), LAN 1 needs only one IWU-CLS connection to send datagrams to both LANs 2 and 3. CLS-CLS connections are permanent, while IWU-CLS connections can either be be permanent (as CLS-CLS connections are) or transient (as IWU-IWU connections are in the indirect method). Since, at a single IWU, there will be fewer IWU-CLS connections in this scheme than IWU-IWU connections in the indirect method, IWU-CLS connections individually will carry more traffic than individual IWU-IWU connections. This makes them likely candidates for permanent configuration to reduce frequent connection establishment/termination overhead.

The tradeoffs between the direct and indirect approaches involve balancing efficiency with delay. Both approaches potentially multiplex frames on a single connection. With the indirect approach, frames addressed to multiple hosts on *the same* local area networks are multiplexed on a single

---

[2] The IWU (interworking unit) between a LAN and the public ATM network is either a bridge or a gateway that transform data to/from ATM network from/to its connected LAN.

VC[3]. With the direct approach using CLSs, frames destined for multiple hosts on *different* LANs are multiplexed on CLS-CLS VCs. Unless permanent IWU-IWU connections are used, the transfer delay is higher using the indirect approach relative to the direct approach. In the indirect approach, the source IWU frequently may not have an open connection to the destination IWU, particularly if data is sent to many different destination networks. The frame must then wait for connection establishment before transmission. With the direct approach, as long as IWUs maintain at least one permanent connection to a CLS, no connection establishment is required. This assumes that all CLS-CLS connections are permanent. Thus, the transfer delay and connection processing load is lower with direct connectionless support. However, if the transfer session is long and the bandwidth requirements are sufficiently high, it may be preferable to use IWU-IWU connections rather than the CLS network. By using the indirect method, it is possible to dedicate ATM bandwidth to a single IWU-IWU association. Since only the ATM layer is used to transport data, all data frames will arrive in sequence and with the lowest delay possible due to more specific routing at the ATM layer. The use of Connectionless Servers in the direct approach not only increases the end-to-end delay due to frame processing at each CLS, but also due to an increased propagation delay, since in most cases, CLSs will lie on a less direct route than would be used by an IWU-IWU connection. The use of CLSs can also result in frame reordering since frames destined to the same IWU may follow different paths.

The efficiency of the direct and indirect approaches can be compared with respect to bandwidth and routing. To reduce the amount of processing due to connection establishment, IWU-IWU connections can be kept open when idle for long periods of time in the indirect scheme. However, while no data is transmitted, each of these idle connections still maintains an ATM VC, which can prevent other connections from entering the network. In this sense, the indirect scheme uses bandwidth inefficiently. The direct approach uses bandwidth more efficiently by maintaining fewer idle connections than in the indirect approach [5]. The CLS-CLS connections multiplex frames from more source-destination pairs than do IWU-IWU connections, which helps further increase the bandwidth efficiency of the direct approach.

In contrast, the direct approach is routes inefficiently. Although it uses bandwidth efficiently by having fewer idle connections, it is routes inefficiently since more links than necessary are used to route frames via CLSs than would be used by an IWU-IWU connection. Since frames must traverse more links, they impose more traffic on the network overall, increasing congestion.

While each of the two approaches has distinct advantages, the remainder of this paper will concentrate on the direct approach. As we are primarily interested in a solution suitable to a public ATM setting, we favor the direct approach because of (1) its low transfer delay for short lived traffic streams, (2) its scalability with respect to the connectivity/connection ratio, and (3) its permanent connection configuration will facilitate efficient IWU implementations.

---

[3] More than one VC may be used under certain circumstances, (*e.g.*, using AAL5 to support more than one higher layer protocol[4].)

2.2 Cell Based Forwarding vs. Frame Based Forwarding

The direct approach described in Section 2.1 requires each CLS to make routing decisions and forward frames. There are two possible methods for forwarding frames at a Connectionless Server[6, 5]: cell based forwarding and frame based forwarding. In CLSs based on cell based forwarding, incoming cells are transported in a *streaming* mode, where the arrival of the first cell of a frame creates an internal mapping from the incoming connection identifier (VPI/VCI) and frame identifier (MID) to the appropriate outgoing connection and frame identifiers. The first and all successive cells of the same frame then are forwarded based on the mapping created by the first cell in a stream without the need for frame reassembly. Servers that operate in this manner are referred to as *Streaming Mode Connectionless Servers*.

In frame based forwarding, incoming cells are first reassembled into a frame at the CLS, the address of the next CLS is determined for the frame, and the frame is (re)segmented into cells that are delivered to the next CLS via the appropriate connection and frame identifiers. Connectionless servers that operate in this mode are referred to as *Reassembly Mode Connectionless Servers*.

For Reassembly Mode CLSs, there are three options available for transmission of multiple frames: *frame*, *cell*, or *hybrid* interleaving. In frame interleaving, all frames are transmitted contiguously. Only after the complete transmission of a frame can other frames be transmitted. In cell interleaving, the transmission of cells from multiple frames are interleaved as the frames are ready for transmission. In hybrid interleaving, separate *frame* interleaved transmission queues are maintained for each outgoing connection. The frames from each of these queues are then *cell* interleaved for subsequent transmission on the physical medium. This has the effect of frame interleaving with respect to a VC, but cell interleaving with respect to the ATM switching fabric.

The use of cell or hybrid interleaving reduces cell loss at the ATM switch by decreasing the burstiness of cells destined for the same output port[6]. However, due to the consecutive cell loss nature of ATM[7][4], frame interleaving will isolate cell loss periods to fewer frames, reducing the overall frame loss rate. Using frame interleaving, only a small number of frames may be affected during a loss period. Using cell or hybrid interleaving, many frames may be affected by a lossy period. Thus, for the same cell loss rate, cell or hybrid interleaving may show a higher frame loss rate than frame interleaving. This effect is less evident in the hybrid scheme compared to the cell interleaving scheme however, since, in the hybrid scheme, cell interleaving only occurs on those physical links that outgoing VCs share.

The use of Reassembly Mode CLSs may introduce large delays relative to that experienced with Streaming Mode CLSs. This additional delay is the result of:

1. *Reception delay:* delaying the transmission of the first cell of a frame until its last cell arrives at the CLS. The total additional delay is *at least* the transmission time of the frame, but could be higher due to the interleaving scheme that is used by the previous hop CLS.
2. *Queueing delay:* delaying the transmission of the first cell of a frame until all cells from previous frames are transmitted. The total additional delay is proportional to the frame length and

---

[4] Because cell loss occurs when switch buffers are full due to congestion, the loss of one cell has a high probability of being followed by another loss. Thus, it is likely to cells will be lost consecutively.

the number of frames in the transmit queue waiting to be transmitted (buffer occupancy). This factor is highly dependent on the interleave scheme used at the CLS.

3. *Processing delay:* increased processing that takes place at each CLS (*e.g.*, buffer management, buffer timers, address resolution/routing, optional frame-level error checking). The total additional delay is proportional to the memory bandwidth, timer implementation, and processing speed of the CLS.

Since each of these delays are experienced at each CLS hop, routes with a greater number of hops will experience proportionally higher delays.

Reassembly Mode CLSs must maintain buffer space for the reassembly of frames. Depending on the buffer management scheme, this limits the number of frames that can be in transit and possibly the maximum frame size. The buffer management scheme can either use variable sized or fixed sized buffers. The use of variable sized buffers requires more sophisticated management schemes to find a buffer that is the "best fit" for a given frame. Additionally, the use of variable sized buffers can result in inefficient buffer utilization due to fragmentation caused by "islands" of allocated buffers that split free regions of buffer space into multiple, smaller regions.

Fixed size buffer schemes either require all buffers to be large enough to accommodate the maximum frame size, or require the use of chained buffers (*e.g.*, BSD UNIX *mbufs*). Maximum size buffers require less space and processing for buffer list information, but potentially waste memory when used for small frames. This waste can be aggravated by either a high multiplexing factor (*e.g.*, many frames are arriving with their cells interleaved, resulting in all buffer space being allocated to the first k frames) or the use of large frames (*e.g.*, AAL5's maximum size (64K octets) frames) Chained buffers require additional space and processing to manage buffer list information but allow buffer space to be allocated based on the size of the frame. The chained buffers can either be individually allocated to a frame as its cells arrive to increase efficiency (a dynamic buffer allocation scheme), or all at once upon the arrival of the first cell of a frame based on a length field in the header (a preallocated buffer scheme) to guarantee that adequate resources will be available to reassemble the frame. A preliminary comparison of dynamic and preallocated buffer schemes is given in the Appendix.

Both Streaming and Reassembly Mode CLSs utilize the services and protocols of the ATM adaptation layer. Streaming Mode CLSs require an adaptation layer that avoids reassembly and dispatches the first cell of a frame up to the connectionless interworking protocol layer. This functionality, also known as *cut-through routing*, has not yet been defined in the CCITT Standards. Although this approach requires less buffer space at CLSs, calculation of the next-hop VPI/VCI/MID identifiers must take place in roughly one cell transmission time (*e.g.*, 2.7μs for 155Mbps throughput), severely impacting the amount of processing that can take place. Several of the functional elements must be implemented using dedicated and/or high speed circuitry. This will increase the overall cost and reduce the allowable number frames in transition through a CLS due to the limited size of faster memories. Reassembly Mode CLSs can perform more processing on each cell of a frame once it is copied into the CLSs buffer space, which requires fewer of of its functional elements to complete their processing in 2.7μs.

Both Reassembly Mode and Streaming Mode CLSs must use a Message Identifier (MID) to disambiguate interleaved cells belonging to different frames. This MID field must be changed at each

CLS to ensure uniqueness across all frames in transit on a given ATM connection. This places a protocol-imposed limitation on the number of frames that can be actively in transit between two CLSs, and consequently, the throughput relative to frame length between two CLSs[5]. These performance limitations are magnified in the Streaming Mode CLS because of the larger number of frames in transit. This larger number is the result of Streaming Mode CLSs inherently using a cell interleaved transmission scheme. This allows cells to continue in transit while remaining cells of its frame may be delayed. This situation is compounded at each hop up to the destination IWU, as more multiplexing takes place. This increases the burden on MID management because entries must remain valid for increasingly longer durations. This also increases the interarrival time between the first and last cell of a frame, particularly as the number of hops travelled increases.

We are currently considering both Streaming and Reassembly Mode CLSs. We are developing analytical and simulation models to determine the performance of the various alternatives mentioned above. We are also implementing a CLS in hardware to empirically verify the feasibility of our design and to identify potential performance bottlenecks. Sections 4 and 5 describe this implementation.

2.3 ATM Adaptation Layer Alternatives

The ATM adaptation layer is responsible for encapsulating and transmitting connectionless frames. As most frames from existing LANs are greater than 48 octets in length, segmentation of LAN frames must take place at the source IWU, and reassembly must take place at the destination IWU. Additional segmentation and reassembly will also be required at intermediate CLSs if Reassembly Mode Connectionless Servers are used. These functions are delegated to the ATM Adaptation Layer (AAL) in ATM.

There are currently two AAL protocols being proposed for transporting connectionless data, AAL4 and AAL5[6]. AAL4 requires an MID (multiplexing ID) field to be present in its PDU, while AAL5 does not. AAL5 is more efficient due to its lower overhead (48 octet payload in AAL5 vs. 44 octet payload in AAL4) and better payload alignment (payload begins at 4 octet boundary in AAL5 vs. 2 octet in AAL4)[8, 9]. The primary functional difference between the two schemes is at the protocol level where frame multiplexing takes place. In AAL4, cells belonging to a frame are identified by a unique MID value. Thus, several frames can be cell interleaved on one VPI/VCI connection.

[10] discusses the framing of IP connectionless datagrams over an ATM - IEEE 802.6 (DQDB) internetwork. IP and ARP packets are encapsulated by an IEEE 802.2 LLC/SNAP header, an 802.6 MAC Convergence Protocol (MCP) header, and an 802.6 common header and trailer. The encapsulated frame is then delivered to the Segmentation and Reassembly Sublayer of the AAL4 Layer. Recent standardization activities by CCITT have recommended a similar encapsulation method in

---

[5] Each frame currently being forwarded on a connection consumes an MID value. The amount of time a frame must hold this MID value is directly proportional to the length of the frame and the number (and length) of all other frames being forwarded simultaneously through the CLS. If a new frame must be forwarded along a connection for which all MIDs are currently in use, the new frame must be discarded, irrespective of the bandwidth capacity of the outgoing connection.

[6] AAL5 was designed for *connection-oriented* service, but it has been proposed for use with connectionless frames as well.

Draft Recommendation I.cls[11]. The draft also calls for a connectionless network access protocol layer (CLNAP) above AAL4 with SMDS as the recommended interface. [4] discusses a possible frame encapsulation scheme ATM using either NLPID, NLPID/SNAP, or LLC/SNAP.

In AAL5, the interleaving of cells from multiple frames on to one VC is not possible due to its lack of an MID field. Because of this, AAL5 requires a separate VC connection for each frame simultaneously in transit. Frames either must be delivered frame-interleaved onto a VC using Reassembly mode CLSs or must be interleaved onto a VP, with different frames identified by their VCIs. To allow greater throughput, CLSs could potentially maintain permanent VP level connections between each CLSs to reserve multiple VCs for multiplexing. However, this reduces the amount of VPI/VCI space available for other traffic. The reservation of VPs and the dynamic creation and destruction of VCs may be feasible in an ATM-LAN setting. However, in a public data network, this may not be a reasonable approach, as it requires the network provider to reserve entire VPs for each source-destination pair. The large number of potential source-destination pairs in a large public network makes VP reservation an unlikely option.

AAL4 was designed for the direct connectionless approach and can be used with either cell based for frame based forwarding. AAL5 favors the indirect approach and/or the direct approach with frame based forwarding. When the indirect approach is used with AAL5, frame encapsulation can be eliminated by implicitly identifying higher layer protocols according to VCIs [4] in lieu of NLPID or LLC/SNAP, further increasing bandwidth and processing efficiency. However, because of ATM's high transmission rates, the minor bandwidth efficiency gained from AAL5's reduced overhead may be insignificant. As our primary focus is large scale public ATM networks directly providing connectionless service, the remainder of this paper will assume the use of AAL4.

3 An Architecture for Connectionless Service over ATM

We are currently designing and implementing an architecture for providing connectionless service over ATM. Our architecture is based on the direct approach to connectionless service and uses the AAL4 encapsulation scheme. Providing connectionless service requires three major entities: (1) Connectionless Servers (CLSs), (2) Interworking Units (IWUs), and a (3) Connectionless Interworking Protocol (CLIP). Figure 2 illustrates the layering model used with the various entities for providing connectionless service on ATM. IWUs maintain (at least) two interfaces, one to the private network (such as IEEE 802 LANs) and one to the public ATM network that is used for interconnection with other private networks. Hosts on the private network exchange data frames with remote networks via an IWU. The IWU maintains connections to one or more CLSs that forward the connectionless data frames to the destination IWU, which in turn forwards the frames to the destination host. CLSs either forward the frame directly to the destination IWU or indirectly via one or more intermediate CLSs. IWUs must also be capable of absorbing frame format and size differences by segmenting and reassembling frames. This architecture assumes that an existing network layer protocol is used at the IWU to resolve the destination *host* address to the CLIP address of the destination *IWU*. IWUs and CLSs then resolve this destination address to the next hop CLS (or IWU). The address space of CLIP is hierarchical to allow for efficient hardware resolution of addresses.

In the layering model shown in Figure 2, CLSs operate above and within the AAL layer using CLIP receiving ATM cells and forwarding them based on routing and forwarding information maintained by the CLSs. Streaming Mode CLSs operate within the AAL layer and process incoming ATM cells. CLIP Service Data Unit (SDU) header information is stored within the AAL Protocol Data Unit (PDU) of the first cell of a frame. All cells in the Streaming Mode CLS are modified at the AAL layer header and trailer (by replacing the cells' MIDs) and at the ATM layer header (by replacing the cells' VPI/VCIs).

Reassembly Mode CLSs may be implemented above the AAL layer by operating at the frame level on AAL Service Data Units (SDU). After determining the next hop, a new AAL SDU can be created by a Reassembly Mode CLS and delivered to the AAL. Reassembly Mode CLSs may optionally operate within the AAL layer at the cell level to gain efficiency by pipelining address translation and reducing memory-to-memory copying. AAL SDUs are not recreated at Streaming Mode CLSs.

An IWU provides a superset of the functionality of a CLS. In addition to sending and receiving cells to and from the public ATM network, an IWU must be capable of sending and receiving MAC frames from some private network. To accomplish this, an IWU maintains a protocol stack appropriate to the private network it is connected to, up to the network layer (*e.g.*, IP, LLC, LAN-MAC). It also maintains a CLIPAAL/ATM protocol stack for connections to one or more CLSs on the ATM network.

In the following sections, we describe the designs for CLIP (Section 4) and a CLS (Section 5). The designs presented assume the use of AAL4 and are for both a Streaming mode and a Reassembly mode of operations. Designs based on AAL5, although they are not discussed in this paper are also being considered, but are not presented due to space limitations.

4 Connectionless Interworking Protocol (CLIP)

Designing a connectionless server architecture that provides efficient and flexible service requires investigating issues in both hardware implementation performance and protocol efficiency. To match the high channel speeds ATM offers, it is likely that a large portion of the connectionless server will be implemented in custom silicon. CLS architectures designed without consideration of hardware implementation issues may exceed the capabilities of available VLSI technology. To afford a hardware implementation, protocols must have simple state machines and fixed packet formats. Complex protocols that require traditional software processing increase the cost and complexity of the implementation. Implementations and designs that do not adequately consider these issues will not provide the level of performance necessary for wide spread acceptance.

In this section, we will describe the Connectionless Interworking Protocol (CLIP). CLIP is used by IWUs and CLSs to efficiently encapsulate and transmit connectionless frames. Its encapsulation scheme is designed such that the header information for many encapsulated protocols will appear in the first cell of a frame, allowing the use of Streaming Mode IWUs and/or traffic shaping based on higher layer protocol information, if desired. To accommodate loss tolerant applications, our design provides for delivery and indication of incomplete or damaged frames.

4.1 CLIP SDU Format

Figure 3 illustrates the format of a CLIP service data unit (SDU) used by IWUs to encapsulate connectionless data from private networks using the Network Layer Protocol ID (NLPID)[12] and the Sub-Network Access Protocol (SNAP)[13] to identify higher layer protocols. The encapsulation method shown here is derived from encapsulations found in [14], [15], and [11]. It is designed to allow the significant fields (*e.g.*, addresses) of several possible encapsulated protocols (*e.g.*, IP, ATM-LAN) to be present in the first cell of a frame, allowing intermediate CLSs and/or IWUs to optionally process cells based on higher layer protocol information. This processing could take the form of alternative routing based on the encapsulated destination address or using encapsulated protocol information to identify cells that are loss tolerant (*e.g.*, forwarding loss-sensitive cells to the next CLS, but discarding loss-tolerant cells to reduce congestion). Having complete addressing information in the first cell also allows IWUs to operate in Streaming Mode by binding incoming VPI/VCI/MIDs to outgoing MAC addresses. Even for Reassembly Mode IWUs, the reception of the first cell could trigger the address resolution process, which could then potentially be complete prior to the complete reception of the frame.

In a CLIP SDU, the presence of the SNAP header is optional, and only required for protocols not explicitly identified by a unique NLPID value. The primary difference between CLIP and 802.6 based frames is the location of several control fields (*e.g.*, HEL, CIB), which, to reduce complexity, has been moved to a fixed position rather than following the variable length header extension. The following is a description of each CLIP SDU component.

CLIP Common Header: The Dly field indicates the delay sensitivity of the frame, with zero being the lowest sensitivity and seven being the highest. The Pri bit indicates that the frame is carrying high priority data and should be delivered as reliably as possible. The CRC Indicator Bit (CIB) field indicates the presence of a CRC in the CLSA Common Trailer. The HEL field is the length of the Header Extension (in 4 octet words), which can range from zero to twenty octets. The Deliver Damaged Frame (DDF) field is set to one for frames that are to be delivered with missing or damaged cells. The BETag header field contains the same value as the frame's BETag trailer field. These two fields are compared to detect the loss of a header or trailer. The BASize field is used to indicate to the receiver how much buffer space will be needed for the frame. The Destination Address field is the E.164 address of the destination IWU. The Source Address field is the E.164 address of the originating IWU. Addresses of intermediate CLSs do not appear in the frame. The address of the next hop CLS must be derived from the Destination Address field. The Header Extension is a variable length field (0 – 20 Octets, in 4 octet increments) that is reserved for future expansion.

NLPID/SNAP Header: The Network Layer Protocol ID (NLPID) field is used to identify the protocol that the encapsulated frame corresponds to. As the NLPID field in only one octet long, some protocols (*e.g.*, IP) may be uniquely identified by a NLPID value, as shown in Figure 4, while other protocols require the use of Sub-Network Access Protocol (SNAP)[13], as shown in Figure 3, to further distinguish the encapsulated protocol.

Payload: The Payload area is used to transport network layer protocol information. Its length can range from 0 to 9184 octets, depending on the encapsulated protocol. The total length of the frame may not exceed 9240 octets (210 cells).

CLIP Common Trailer: The Padding field is used to pad network layer protocol information to a four octet boundary, and its length is derived from the two low order bits of the BASize field. The CRC-32 field, when present, provides error protection from the Dly field to the end of the Padding field. The Reserved field is reserved for future use. The BETag is equivalent to the BETag field of the header and is used to detect lost frame headers and trailers. The Length field indicates the total length of the payload portion of the frame.

4.2 CLIP PDU Format

As we are primarily interested in designing an architecture for public network use that cannot easily reserve entire VPs for connectionless service, we are using AAL4 as our cell encapsulation scheme. This allows us to multiplex several frames on one CLS-CLS VC, preserving valuable VP/VC space. Figure 5 shows the first cell of an an IP datagram, which is encapsulated within a CLIP frame. Note that even when this less efficient AAL is used, (*i.e.*, AAL4 with 44 octet payload, as opposed to the 48 octet payload in AAL5), the first cell contains complete addressing information for both the IWUs and the source and destination end systems (which will appear at the head of the CLIP SDU payload as IP addresses). This arrangement allows IWUs to forward cells in either Streaming or Reassembly modes.

4.3 CLIP Protocol Semantics

To allow high throughput rate throughout the system, the protocol processing that must be performed at CLSs is minimal. Figure 6 shows the finite state machine for frame reception at a CLS. This state machine is simple (three states) and can be easily implemented in hardware as the transitions are derived by examining two fixed position fields in the AAL4 header and one fixed position field in the first cell of a frame (DDF). CLSs extract addressing information from BOF cells and resolve the destination address to an outgoing connection. This and all subsequent cells from the same connection with the same MID field value are forwarded via this connection until either (1) an EOF cell arrives, indicating that the frame is complete, or (2) a new BOF cell arrives, indicating that the previous EOF cell was lost. Each cell from the same frame will have a sequence number one greater than the previous cell's. However, (assuming that the ATM layer will not reorder cells), sequence number gaps can occur when either (1) a cell is discarded by the ATM layer due to congestion, or (2) a cell that has an invalid CRC10 field is discarded by the CLS. In such cases, if the frame's Deliver Damaged Frame (DDF) field is zero (*i.e.*, the frame's FSM is in state RECV), the arrival of an mis-sequenced cell allows the CLS to discard all cells belonging to that frame and causes the FSM to enter the IDLE state. For frames with a DDF field of one (*i.e.*, the frame's FSM is in state RECV_DDF), the missing cells are ignored at the CLS and are replaced with zero-filled cells at the destination IWU.

As it is possible that cells from multiple incoming connections will have the same MID value, CLSs will need to change the MID of incoming cells to a value that is unique on the outgoing connection. This change necessitates recalculating the CRC10 field of each cell. The new MID value must be chosen from the set of free MIDs such that the MID values selected is uniformly distributed across all possible 1024 values. This ensures that all possible MID values will be used, which is necessary for CLSs that perform load balancing of frames based on MID value.

The routing table maintenance protocols will be executed offline in software by an attached processor and routing updates will occur via special control cells. We are investigating alternatives for routing algorithms based on OSPF and/or BGP. The choice of routing protocol does not affect the design or performance of the main data path in our CLS.

5 Connectionless Server (CLS)

CLSs are located throughout the ATM network, providing a virtual datagram network interconnecting IWUs. Each CLS must maintain virtual circuit connections with two or more CLSs or IWUs, possibly segment and reassemble frames, and resolve destination IWU addresses to the appropriate next hop CLS.

In this section, two design alternatives for a CLS are illustrated, one that operates in Streaming Mode (Section 5.1) and one that operates in Reassembly Mode (Section 5.2). Both designs assume the normal processing and routing of cells is performed by the ATM layer. A connectionless server is physically connected to the ATM network via a duplex physical link to an ATM switch. We assume that statically maintained virtual circuits interconnecting CLSs and IWUs are used to forward cells, and are only established or terminated by external control. It is also assumed that special *Route Servers* will be implemented to perform actual link and CLS state monitoring, routing advertisements and calculations. These Route Servers then update the internal routing tables of CLSs by sending special control cells. Offloading this task to an offboard system allows the CLSs to be implemented using simplified hardware, while still taking advantage of the power of a general purpose processor for route calculation.

5.1 Streaming Mode Connectionless Server

In a Streaming Mode CLS, an arriving *BOF* cell is allocated a forwarding table entry that binds the incoming VPI/VCI/MID to an appropriate outgoing VPI/VCI/MID. All subsequent cells with the same incoming VPI/VCI/MID are forwarded according to their corresponding entry in the forwarding table. To do this, all processing must be performed in roughly one cell transmission time. Thus, Streaming mode CLSs must be capable of forwarding cells at a rate comparable to the channel speed. Figure 7 illustrates a hardware based design that is capable of efficient cell forwarding. This design consists of four major subsystems: the Protocol Engine, the Forwarding Table VPC Map, the Forwarding Table, and the Address Resolution Map.

Protocol Engine: The Protocol Engine is responsible for receiving cells from the ATM layer and dispatching them to the proper subsystems. All incoming cells trigger a Forwarding Table lookup based on the incoming connection number (calculated by the Forwarding Table VPC Map). This number combined with the cell's MID is used to select the entry in the Forwarding Table that corresponds to the cell's incoming connection and frame. Cells that are the first cell of a frame (Beginning-Of-

Frame or BOF cells) also trigger a next hop lookup from the Address Resolution Map, the result of which is used to update the outgoing VPI/VCI/MID/Connection Number fields of Forwarding Table entry. Once the Forwarding Table entry is loaded for a cell, the cell's MID field is replaced with that of the Forwarding Table entry, a new CRC is generated, and then the cell is transmitted via the VPI/VCI specified by the Forwarding Table entry. End-of-Message (EOM) cells free the Forwarding Table entry and its MID for use on subsequent frames.

The data path of the protocol is streamlined and allows several operations (forwarding table lookup, address resolution) to be performed in parallel. The CRC regeneration is be performed in hardware as the cell is being transmitted.

Forwarding Table VPC Map: The Forwarding Table VPC Map derives the internal connection number from the VPI and VCI values of each incoming cell. This number is then used to select the corresponding set of entries in the Forwarding Table for frames received from that connection. It is expected that a CLS will maintain a relatively small fraction of the total possible connections allowed by the 28 bit VP/VC identifiers. Some mechanism must be used to map sparsely distributed but large VPI/VCIs values onto smaller compact internal connection numbers. By building this small translation table with Content Addressable Memory (CAM), reasonable cost is maintained while still satisfying the performance constraints of the system. The CAM accepts the 28 bit VPI/VCI as input, and produces a unique connection number for use by the forwarding table. The number of words in the CAM is equal to the maximum number of connections supported, reffered to in this paper as MAX_CONNECTION_NO.

Forwarding Table: The Forwarding Table for a Streaming mode CLS is used to keep track of state information for each frame currently being forwarded. The design of the Forwarding Table is shown in Figure 8. Its main component is a array of Static RAM with MAX_CONNECTION_NO pages of 1024[7] entries, each of which are 8 octets wide. For each incoming cell, an entry from this array is selected by loading the incoming connection number derived from the VPC Map and the cell's MID and asserting the proper control lines. This causes the Forwarding Table to select a line from RAM by catenating the connection number derived by the VPC Map with the cell's MID value to calculate the address. From this line, the Forwarding Table produces (1) the outgoing VPI/VCI/MID values to be used to transmit the cell to the next hop CLS, (2) the outgoing internal connection number, which is needed for use by the Address Resolution Map when the MID value is deallocated, (3) the expected sequence number for the cell, and (4) a seven bit *state* variable for use by the Protocol Engine. This state variable is used to implement the receive finite state machine as well as to store the value of the DDF bit from the frame header. Once the Protocol Engine retrieves the Forwarding Table information and processes the cell, the Protocol Engine then stores the new state variable and new next sequence number back to the cell's entry by asserting the proper control lines.

Address Resolution Map: The Protocol Engine uses the Address Resolution Map to calculate the outgoing VPI/VCI for the destination address embedded in the first cell of a frame. The Address Resolution Map also selects a unique MID value that will distinguish the frame on the outgoing connection. These VPI/VCI/MID values are then substituted in each subsequent cell of the frame.

---

[7] There are 1024 entries per connection to accommodate the maximum number of frames that can simultaneously be multiplexed on a connection.

The Address Resolution Map has two subcomponents: the Address Resolver and the MID Manager. The Address Resolver reads the destination IWU address one octet at a time. Once all eight octets are read in, it then produces an outgoing VPI and VCI, as well as an internal connection number that is used by the MID Manager. The MID Manager takes this connection number and selects a MID value from the available (unused) MID values for that connection.

The IWU addresses are allocated hierarchically to allow efficient routing lookups. The Address Resolver takes advantage of this hierarchical organization by implementing the routing table as a hardware realization of a trie[16, 17]. As shown in Figure 9, the trie is implemented in static RAM in multiple pages of 256 entries, with each page corresponding to the possible values of a single byte of an address. Each entry has two fields, a Connection_no that corresponds to the outgoing internal connection number for the destination address, and a Page_No field that indicates the page that corresponds to the next octet of address information. The 0th page is reserved for the first octet of the address. If no further octets of the address are significant, the line will have 0 as its Page_No entry indicating it has no additional pages. When an entry is encountered with 0 as its Page_No (*i.e.*, no more addressing information is needed to determine the outgoing connection), the entry's Connection_No is taken as the connection number for the address. Th connection number is used to index into the Connection Number Free List to look up the actual VPI/VCI values to be used.

Once the connection number is found, it is also input to the MID Manager. The Address Resolution Map then asserts the appropriate control lines of the MID Manager. This causes the MID Manager to reserve an MID from the list of unused MIDs for the connection indicated. As shown in Figure 10, the MID Manager is a simple hardware implementation of a linked list. It maintains a static RAM array of MAX_CONNECTION_NO pages of 1024 entries, one for each possible MID value. Each page corresponds to an outgoing connection number. Each entry contains the MID value of the next free MID value to be used on the page's connection. Entry 1023 and 1022 are the head and tail of the list, respectively. MIDs are allocated from the head of the list, and deallocated MIDs are placed at the end of the list. This ensures that all possible MID values will be used, which is necessary for CLSs that perform load balancing of frames based on MID value.

5.2 Reassembly Mode Connectionless Server

The alternative to a Streaming Mode CLS is a Reassembly Mode CLS. It mainly differs from the Streaming Mode CLS in that is has a Frame Manager for the reassembly of cells into frames. In the Reassembly Mode CLS shown in Figure 11, an arriving *BOF* cell is assigned a buffer slot in the Frame Manager. Each buffer slot maintains a list of fixed size buffers, each of which contains a cell belonging to the slot's frame. An outgoing VPI/VCI/MID can be determined after the frame has been reassembled. Alternatively, as in this design, the outgoing VPI/VCI/MID is determined after arrival of the first cell so that next hop calculation can be performed in parallel with reception of following successive cells.

A Reassembly mode CLS is composed of the following subsystems: the Protocol Engine, the Frame Manager, the Forwarding Table VPC Map, the Forwarding Table, and the Address Resolution Map.

Protocol Engine: The Protocol Engine for a Reassembly mode CLS is responsible for receiving cells from the ATM layer and dispatching them to the proper subsystem. An internal connection number from the Forwarding Table VPC Map is used to select a page of entries in the Forwarding Table, which, when combined with the cell's MID field, selects the appropriate Forwarding Table entry. The information in the Forwarding Table entry is then used to dispatch the cell to the designated Frame Manager Slot. This slot maintains the list of buffers that correspond to the cells of the frame. The Frame Manager will allocate a buffer for the cell and insert the buffer into the frame's slot. Cells that are the first cell of a frame (BOM cells) cause a slot in the Frame Manager to be reserved, moving the slot from the free list to the active list, and the index of the slot is stored in the corresponding Forwarding Table entry. End of message (EOM) cells move the slot from the active list to the transmission list where they are eventually transmitted to the next hop CLS. When the last cell has been transmitted, the slot is moved onto the free list.

Frame Manager: The Frame Manager of a Reassembly mode CLS is the subsystem that is responsible for reassembling incoming cells into frames, performing frame level procedures such as error checking, forwarding cells of reassembled frames, and managing reassembly buffer space. The Frame Manager maintains a collection of entries called *Frame Manager Slots*. A Frame Manager Slot consists of (1) a head and tail pointer to a list of fixed size buffers in the buffer cache, and (2) the outgoing connection number. Each buffer in the buffer cache has 16 octets of control information and 48 octets of payload information. The buffer cache is logically organized into three lists: a free list, an active list, and a transmit list. The free list contains all of the buffers that are not currently in use. The active list contains those that are currently being reassembled, but that still have at least one cell to yet be received. The transmit list contains those that have been completely reassembled and are in the process of being transmitted. The Frame Manager forwards cells from the buffer cache in the transmit list, placing each buffer on the free list once it has been transmitted. We are currently evaluating interleaving and allocation schemes for use by the Frame Manager. Appendix A includes a preliminary comparison of dynamic and preallocated buffer schemes.

Forwarding Table VPC Map: The Forwarding Table VPC Map for a Reassembly mode CLS is identical to the Streaming mode version. In combination with the Forwarding Table, these two components allow a quick and efficient access to frame slots in the Frame Manager.

Forwarding Table: The Forwarding Table for a Reassembly mode CLS is a simplified version of its counterpart for a Streaming mode CLS. As much of the state related to frame reception is present in the Frame Manager data structures, Forwarding Table entries need only contain a reference to a Frame Manager Slot.

Address Resolution Map: The Address Resolution Map and Address Resolver for a Reassembly mode CLS are identical to the Streaming mode versions except that outgoing VPI/VCI values are stored in Frame Manager Slots, not in Forwarding Table. If cell interleaving is used, then MID management will be similar to that used for Streaming mode CLS. If frame interleaving is used, then the MID management can be implemented as a simple counter.

6 Conclusions

Support for connectionless data is critical to the success of ATM. This paper has presented architectural options for connectionless transport and has outlined our design of a connectionless service architecture. We believe that for a connectionless service scheme to be scalable to public ATM networks, some provision for multiplexing traffic for multiple IWUs onto a smaller number of connections is required. Connectionless servers can be implemented to perform this function, allowing flexible allocation of resources across all IWUs. Among the many open questions to be answered with respect to connectionless service schemes for ATM, we are currently using analysis, simulation, and empirical experimentation to determine (1) how the performance of direct and indirect connectionless support are affected by traffic characteristics, (2) how the frame loss rate is affected by frame, cell, and hybrid interleaving, (3) how is the delay affected by the choice of frame or cell based forwarding, and (4) how does buffer management affect the performance of the frame based CLS, particularly when complexity cost is considered. We are using our preliminary findings in these areas to guide the design process as we implement a network of CLSs.

Acknowledgments

We would like to thank Jaime Bae, Masaaki Yoshimura, Hung Huang, and Brett Vickers for their participation in this project.

References

[1] IEEE, *Std-802.3 1985: Carrier Sense Multiple Access with Collision Detection*. New York, 1985.

[2] ANSI, *American National Standard, FDDI Token Ring Media Access Control (MAC)*, ANSI X3.139-1987 ed.

[3] CCITT, Geneva, *CCITT Recommendation I.327: B-ISDN Functional Architecture*, 1991.

[4] J. Heinanen, "Multiprocotol Interconnect over ATM," Tech. Rep. Draft RFC, IETF, June 1992.

[5] T. Kawasaki, S. Kuroyanagi, R. Takechi, and K. Hajikano, "A study on high speed data communication system using ATM," in *Proceedings of the Conference on Global Communications (GLOBECOM)*, pp. 2105–2109, IEEE, 1991.

[6] G. Fioretto, T. Demaria, R. Vaglio, A. Forcina, and T. Moro, "Connectionless Service Handling Within B-ISDN," in *Proceedings of the Conference on Global Communications (GLOBECOM)*, 1991.

[7] 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," *submitted to IEEE Trans. Commun.*, June 1992.

[8] T.-. ANSI, *Simple and Efficient Adaptation Layer (SEAL)*, Aug 1991.

[9] T.-. ANSI, *AAL 5 - A New High Speed Data Transfer AAL*, November 1991.

[10] E. Cooper, "Transmission of IP Datagrams over Asynchronous Transfer Mode (ATM) Networks," Tech. Rep. Draft RFC, IETF, Pittsburgh, PA, November 1991.

[11] CCITT, Geneva, *CCITT Draft Recommendation I.cls: Support of Broadband Connectionless Data Service on B-ISDN*, December 1991.

[12] ISO/IEC, "Information technology - Telecommunications and Information Exchange between systems - Protocol Identification in the Network Layer," Tech. Rep. TR 9577 1990 (E) 1990-10-15., ISO/IEC, 1990.

[13] IEEE, *Draft Standard P802.1A–Overview and Architecture*. New York, 1989.

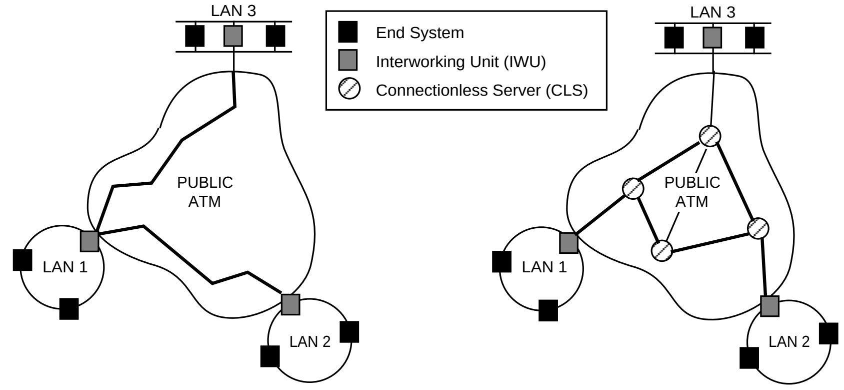
[14] IEEE, *Std-802.6 1990: Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)*. New York, 1990.

[15] D. Piscitello and J. Lawrence, "Transmission of IP datagrams over the SMDS Service," *Network Information Center RFC 1209*, pp. 1–11, March 1991.

[16] D. E. Knuth, *The Art of Computer Programming, Vol. 3: Searching and Sorting*. Reading, MA: Addison Wesley, 1973.

[17] T.-B. Pei and C. Zukowsli, "Putting Routing Tables in Silicon," *IEEE Network*, pp. 42–50, Jan. 1992.

Figures



a) Indirect Connectionless Support

b) Direct Connectionless Support

Figure 1: Direct vs. Indirect Connectionless Service Architecture

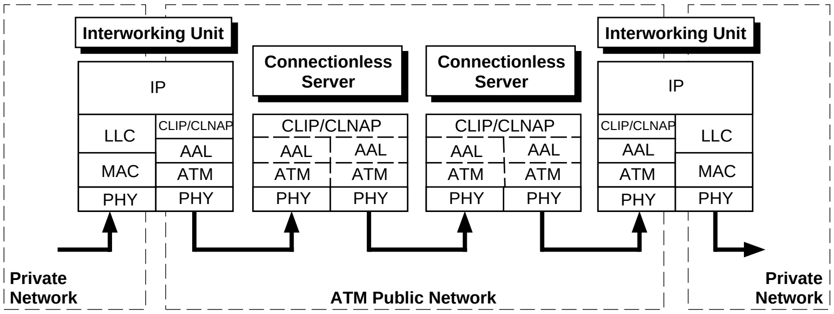


Figure 2: Connectionless Service on ATM

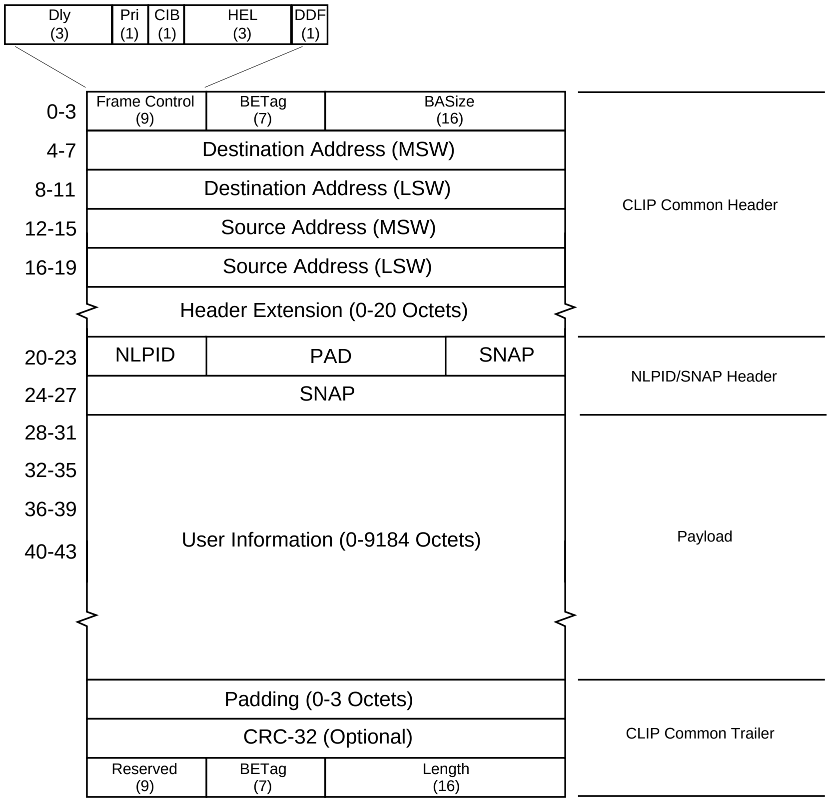


Figure 3: CLIP Encapsulation Format (NLPID/SNAP)

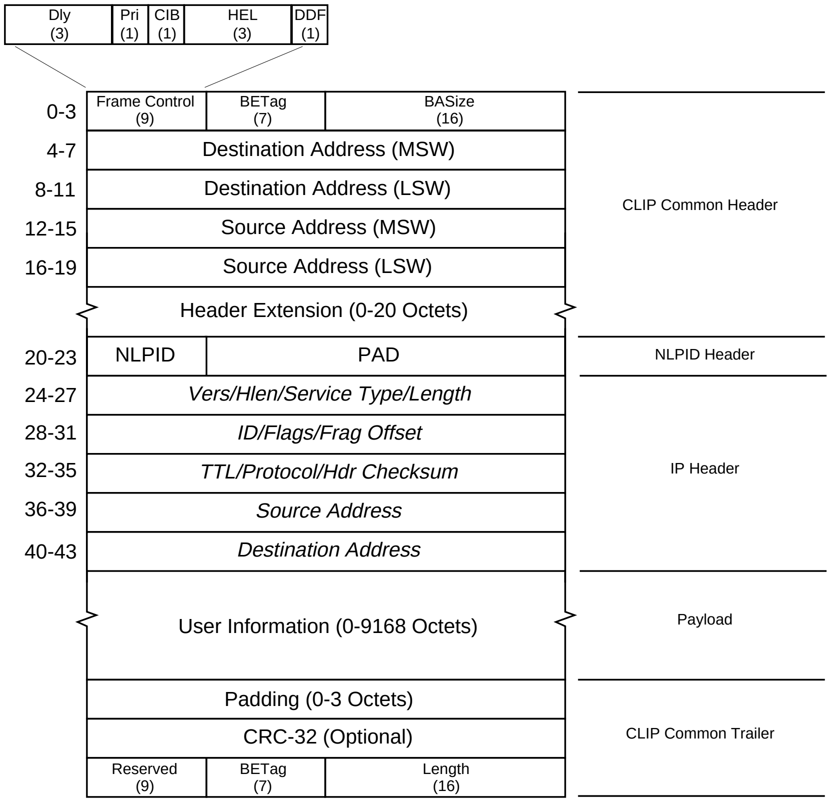


Figure 4: CLIP Encapsulation Format (NLPID/IP)

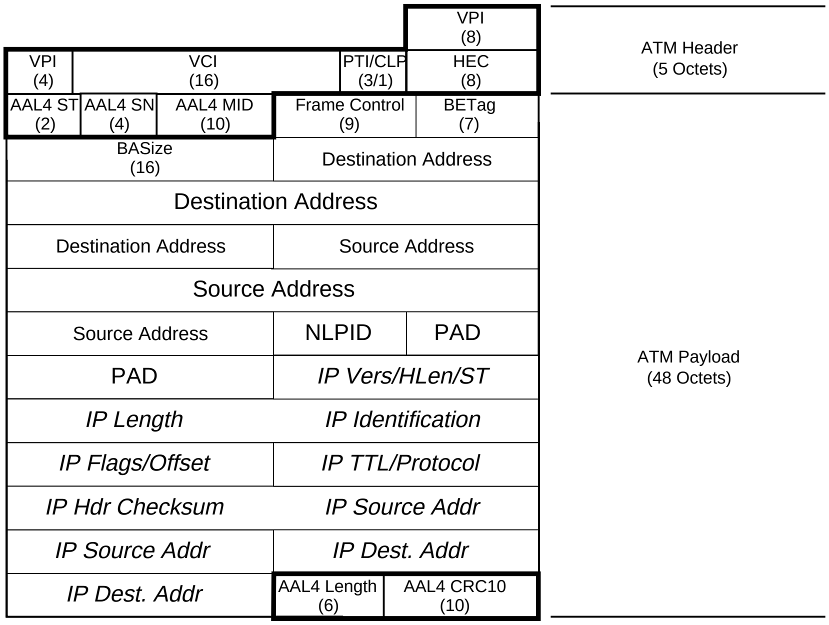


Figure 5: CLIP Protocol Data Unit Format

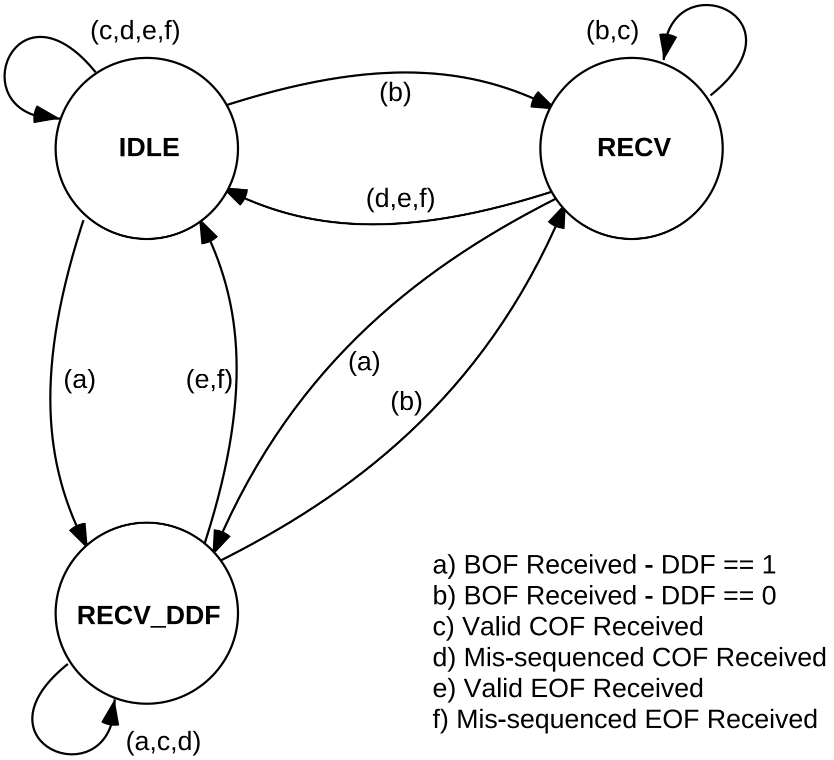


Figure 6: CLIP Receive Finite State Machine

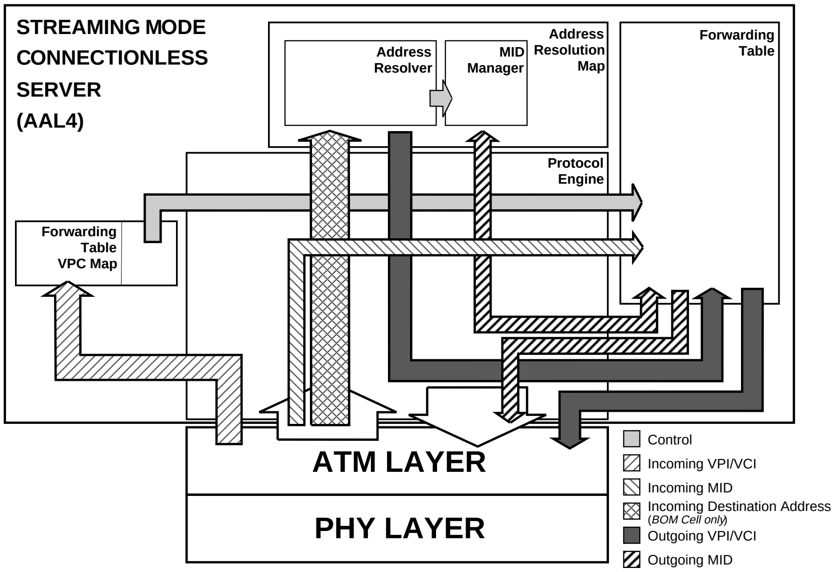


Figure 7: Streaming Mode Connectionless Server

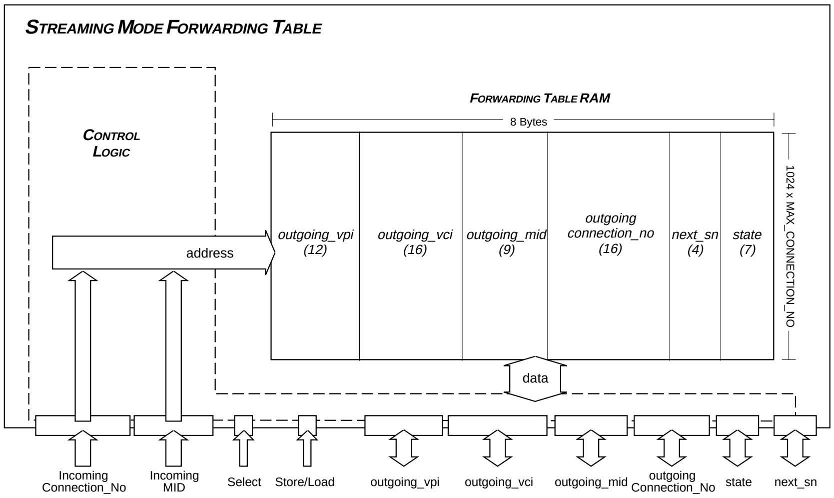


Figure 8: Streaming Mode Forwarding Table

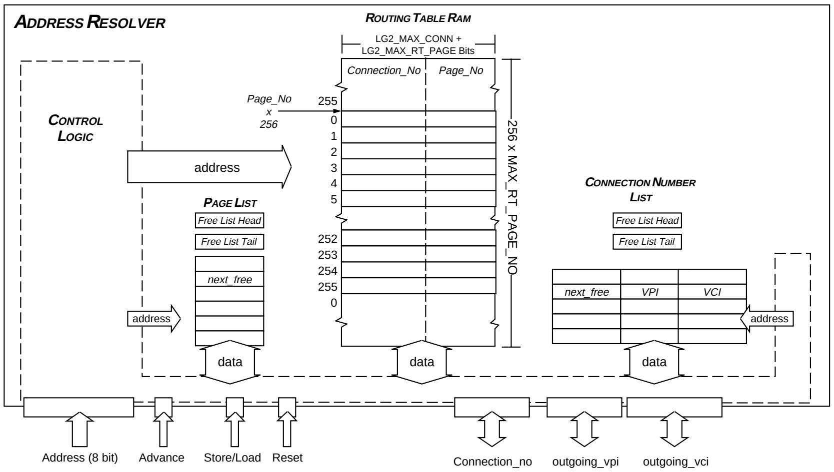


Figure 9: Address Resolver

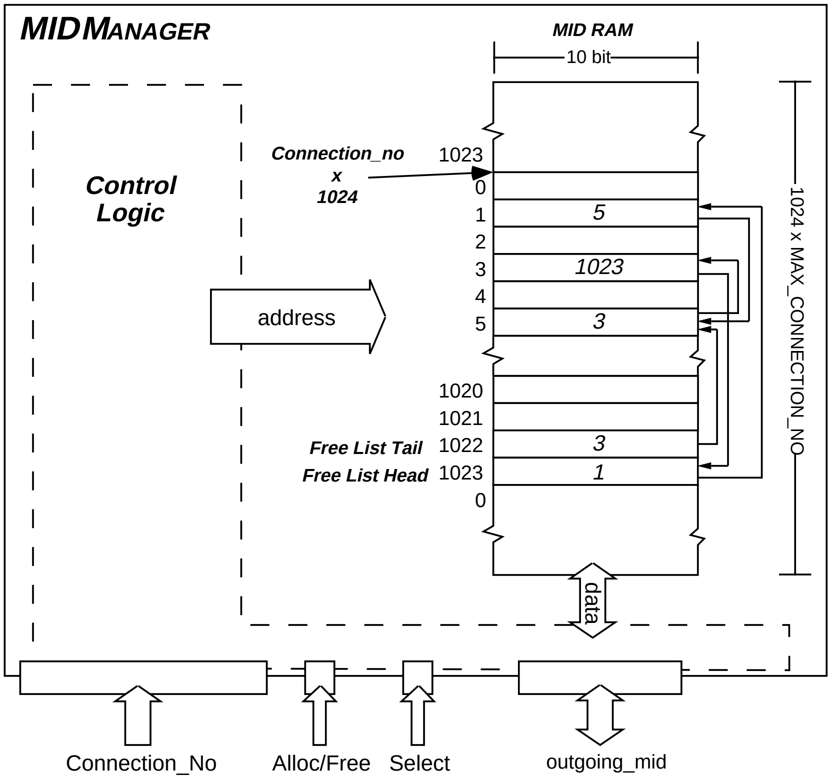


Figure 10: MID Manager

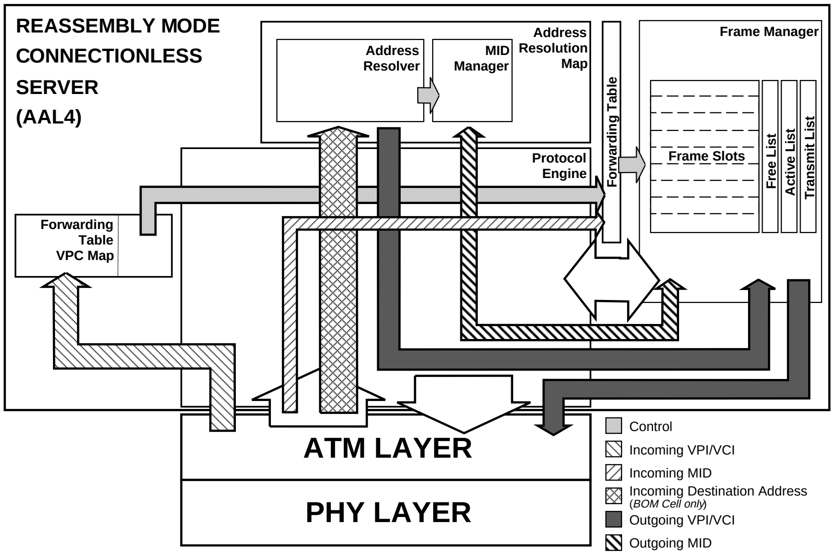


Figure 11: Reassembly Mode Connectionless Server

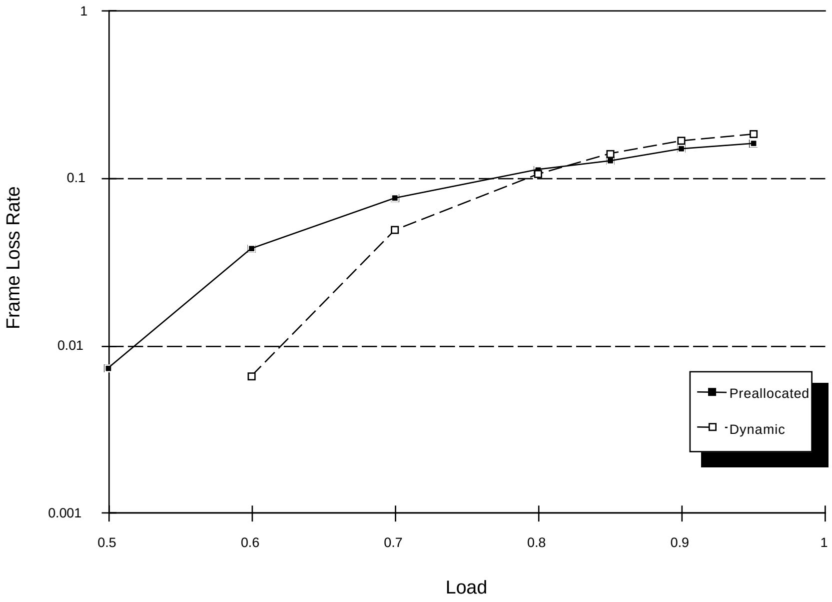


Figure 12: Frame Loss Rate vs. Load, N = 32, B = 3200

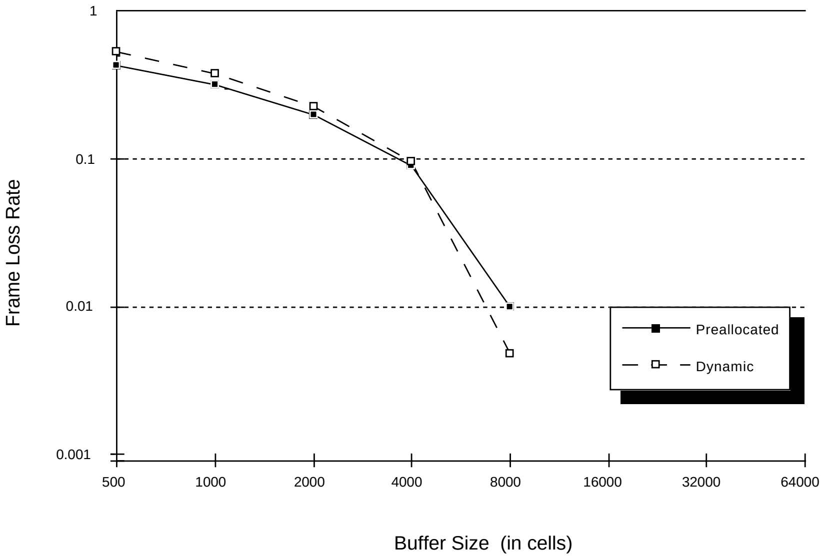


Figure 13: Frame Loss Rate vs. Buffer, N = 32, ρ = 0.83

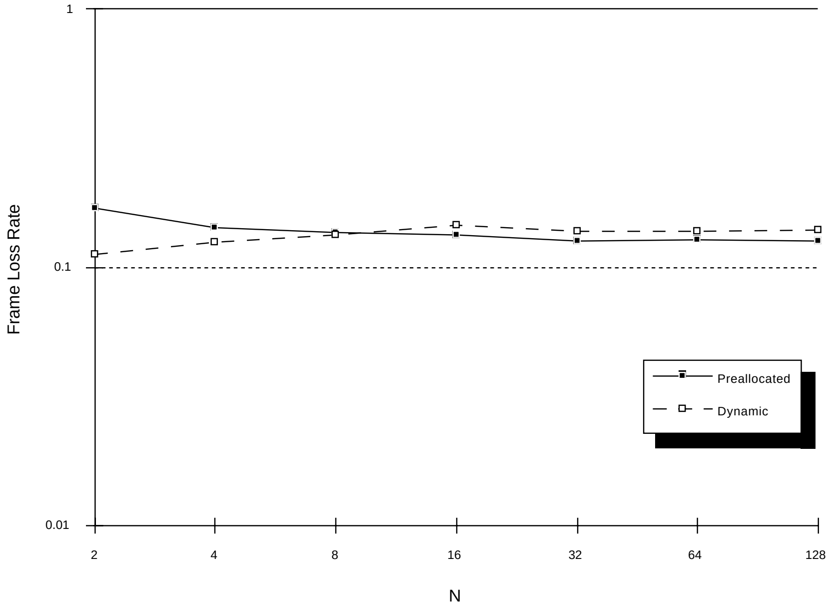


Figure 14: Frame Loss Rate vs. N, $B = 100N$, ρ = 0.85

Appendix A: Comparison of Preallocated vs. Dynamic Reassembly Buffer

We compared through simulations the frame loss performance of preallocated and dynamic buffering schemes for a Reassembly Mode CLS with frame interleaving. We assume normalized geometrically distributed frame sizes with a mean of 50 cells per frame. Frame sizes are normalized with a minimum of one cell per frame and a maximum of 210 cells per frame. Since frames are reassembled at each CLS and then segmented into cells and transmitted cell by cell, cells are transmitted at a regular interval at the sending CLS. However, because of the possible queueing delay and interfering traffic at intermediate ATM switching nodes between two CLSs, cells arrive at the next hop CLS with random intervals. We assume this fluctuation is normally distributed with a standard deviation of 10% of the original time interval at the sending CLS.

Figure 12 shows the frame loss rate for the two buffering schemes as a function of the traffic load ρ. It is assumed that the number of connections that the CLS supports (N) is 32 and that the buffer size (B) is 3200 (cells) for both schemes. It is shown that the dynamic buffering scheme provides a lower loss rate for loads less than approximately $\rho = 0.82$, while the frame loss rate for the preallocated buffering scheme is lower for higher loads.

Figure 13 shows the frame loss rate as a function of the buffer size B. Traffic load is assumed to be $\rho = 0.83$, and the number of connections that the CLS supports (N) is 32 in this figure. It is shown that the dynamic buffering scheme provides a smaller frame loss rate when the available buffer space is large and that the preallocated buffering scheme performs better when the buffer space is small.

These two figures show that the preallocated buffer scheme shows a smaller loss rate under high buffer load conditions, (*i.e.*, when traffic load is high or the buffer size is small). The dynamic buffer scheme shows a lower loss rate under low buffer load conditions. This can be explained as follows. Under low buffer load conditions, the dynamic buffer scheme allows multiple frames to share some buffer space over time that they would not otherwise be able to share. However, under high buffer load conditions, the dynamic buffer scheme deteriorates. High load conditions occurs when the input traffic load is high or the total buffer space is small. Under these conditions, frames may occupy buffer space for some time, possibly causing other frames to be dropped, but may itself be dropped later on because of the lack of buffer space to store all of its cells. The problem may be aggravated by a high frame arrival rate which increases the probability that frames already resident in the buffer will be dropped.

Figure 14 shows the frame loss rate as a function of the number of connections (N) that a CLS supports. Traffic load ρ is 0.85, and the buffer size B is 100 $\times$ N. As in Figure 12 and Figure 13, the tradeoff between two buffer schemes is shown in this figure. As N increases, the statistical multiplexing gain increases. This is evident in the lower loss probability of the preallocated scheme. However, in the dynamic buffering scheme, when the buffer becomes full, with larger N, there can be more frame arrivals which can cause loss of frames already residing in the buffer. This can cause more frame loss as shown in Figure 14.

Although a simple traffic model was assumed, some general observations can be made regrading the performance of the two buffering schemes. The preallocated buffer Reassembly Mode CLS ensures that enough buffer capacity is allocated to a frame as soon as the first cell of the frame is accepted. It does this at the cost of rejecting frames from entering the buffer if there is not enough space to accommodate those frames in their entireties. The dynamic buffer reassembly mode CLS allows statistical sharing of buffer space to increase the number of frames which can be admitted into the buffer. This is done at a risk of dropping those frames later on if the buffer occupancy does not decrease. The simulation study shows that under high load buffer conditions, the preallocated buffering scheme provides a smaller frame loss rate; under low load buffer conditions, the dynamic buffering scheme provides a smaller frame loss rate.