5 The ATM Adaptation Layer.

This is where the user sends the data to be transmitted, and where the data packet is changed into packets of a size appropriate to the ATM cells. Since the ATM cells have a 48B payload, the "actual" payload will be no more than 48B; some room will have to be given to the delivery of various services that will depend on the type of traffic and of connection supported. We will now look at these issues.

5.1 Types of Service.

There are several service classes identified up to now. The identification (and protocol definition) is still tentative, several changes have been made already, and much further understanding is needed before the requirements of the various service classes (or the final number, for that matter) are fully defined. For the most recent ATM Forum document on Traffic Management see

http://www.atmforum.com/atmforum/specs/approved.html,

in particular the Traffic Management 4.1 file. The discussion following represents a slightly older classification, which should be easily mappable to the most recent ideas.

Class X) User defined: the only involvement on the part of the network is in the provision of Quality of Service and bandwidth. All other actions are user defined. The AAL 0 portion of the ATM Adaptation Layer will provide this service. As one might expect, there is no agreed standard, yet (maybe none is needed).

Class A) Circuit Emulation. This is the "unmassaged" telephone traffic, although constant rate video would also be included. The applications supported have the following characteristics:

Constant bit rate at source AND destination.
Timing relationship between source and destination.
Connection between end users.

In contrast to Class X traffic, where the ATM Adaptation Layer is empty, the AAL for Class A traffic must perform several functions:

Segmentation and Reassembly of data frames into cells.
Management of cell delay variations.
Detection and handling of "bad" cells.
Recovery of source clock frequency.
Detection of bit errors in the user information field.

Class B) Variable-Bit-Rate-Services. Video and voice that must appear to the end-users to satisfy strict timing requirements, but can be coded – because of their nature – as variable-rate information. The services provided are:

Transfer of variable-bit-rate information between end-users.
Transfer of timing information between source and destination.
Indication of lost or corrupted information not recovered by the AAL itself.

This is also still lacking a standard. The reasons should be fairly obvious: we go from multiplexing telephone conversations where we use statistical techniques to “overstuff” the channel, to “movies-on-demand”, to videoconferencing (where you must deliver both synchronized acceptable video and acceptable audio). There is a substantial discussion of this in the ATM Forum document on Traffic Management v. 4.1.

**Class C** Connection-Oriented-Data. So-called “traditional traffic”. The services are connection-oriented with variable information flow:

- Segmentation and reassembly of frames into cells.
- Detection and signaling of errors in the data.
- Other services – e.g., multiplexing and demultiplexing of end-user connections may also be provided.

**Class D** Connectionless Data. This might be used to carry TCP/IP traffic. It needs to provide:

- Segmentation and reassembly of frames into cells.
- Detection and signaling of errors in the data.
- Multiplexing and demultiplexing of end-user data-flows.
- Network Layer addressing and routing.

The service categories described in *ATM Forum Traffic Management 4.1* are

- **Constant Bit Rate** (CBR). For voice, video and circuit emulation.
- **Real-Time Variable Bit Rate** (rt-VBR). For voice and video.
- **Non-Real-Time Variable Bit Rate** (nrt-VBR). For non-real-time bursty traffic. Amenable to statistical multiplexing.
- **Unspecified Bit Rate** (UBR). For file transfer and e-mail.
- **Available Bit Rate** (ABR). Same applications as UBR (for example), but feedback control is expected to reduce losses and increase fairness of bandwidth allocation.
- **Guaranteed Frame Rate** (GFR). The congested network may drop entire frames rather than cells. This guarantees a Minimum Cell Rate, where the cells are transported in complete frames.

### 5.2 The AAL Types.

Although the classes and the AAL types were expected to match, one has found that the association between traffic classes and the types of services provided by the AAL is a bit nebulous. Thus, the five AAL types overlap the traffic classes to varying degrees.

- AAL-0 does nothing.
- AAL-1 provides function for Class A.
AAL-2 provides for Class B.

AAL-3/4 provides some functions for classes C and D.

AAL-5 Also provides functions for classes C and D.

Signaling AAL. This does not provide any user-to-user services.

AAL-0 has no functionality associated with it. One might see it as a way to provide "sandbox" space on the existing network, or it might be viewed as an invitation to anarchy. Dutton and Lenhard seem to take the second view, while the past success of "naturally evolved standards" would argue for always leaving some room. We will examine in some more detail the AAL-1 and AAL-3/4 types, leaving the remainder as an "exercise", along with a perusal of ATM Forum Traffic Management 4.1.

5.2.1 Some terminology.

The ATM Adaptation Layer contains three sublayers. The user would access the functionality through a Service Access Point into the topmost sublayer – the Service Specific Convergence Sublayer – and the data so inserted would be eventually moved to the ATM layer via another Service Access Point. The intermediate layers are: the Common Part Convergence Sublayer and the Segmentation and Reassembly sublayer. The interfaces between the sublayers are NOT Service Access Points: this is just a way of saying that no external programming interface may access the sublayers.

Service Data Unit. This is the name for the data item as received by a layer.

Interface Data Unit. A Service Data Unit may be segmented at an interface between layers and each of the segments becomes an Interface Data Unit.

Protocol Data Unit. This is the name for the data item as it is about to be passed to the lower layer: headers and trailers have been added, all the needed massaging has taken place.

5.3 AAL-1.

The network is prepared to support the transmission of bits generated at a constant rate and expected at the destination at the same constant rate. Thus AAL-I-nnnn must be capable of receiving the data, packaging them, sending them across the network, and reassembling them at the other end with the same timing relationships they had at the start.

Unfortunately, the transmission is in packets, so that the time interval between bits in transit bears no relationship to the original one. Furthermore, the packets themselves may arrive with different timing relations among themselves. This basically means two things:

i) The amount of buffering in the intermediate network nodes must be kept to a minimum – to minimize both trip delay and the possibility of jitter.

ii) The receiver must have adequate buffer space (and delay capability) so that there is always a bit available when needed.
iii) Both sender and receiver have synchronized their clocks so that the timing relationship is correctly maintained. The problem is that a link of any duration has to deal with "clock drift": no two clocks are ever the same.

iv) Recall that the ATM environment performs no error recovery – recognized bad cells are just thrown away.

What does the payload of the ATM cell look like? It can’t be all "user payload", since that would mean that no real service is performed by AAL-1. As it turns out, the user payload consists of only 47 bytes. The first byte of the payload is divided into:

i) A 3-bit Sequence Number.

ii) A 1-bit Convergence Sublayer Indicator (CSI).

iii) A 4-bit CRC field to protect against errors in the Sequence Number field.

The "standard" way to use the Sequence Number is to have a large enough SN field so that the round trip time from sender to receiver and back will not be larger than the time for the sender to have sent a "whole set" of cells (8 in this case), thus forcing repetition of the sequence number. It is clear that this will not work in this case, since at 155Mb/s you could have many megabits in a long connection. On the other hand, none of the ATM layers guarantee delivery – they only guarantee good reliability and correct sequencing. In practice, when coupled to the decision of NOT allowing multiple concurrent physical paths from source to destination for the same Virtual-Path/Virtual-Channel pair (the physical path could change in mid-transmission only in case of network failure), we only need to guarantee that a missing packet will be recognized: the 3-bit SN will be effective, except in the case when an exact multiple of 8 packets is lost (a, supposedly, vanishingly small probability event). This will alert the user process that something was lost, and it may trigger the re-sending of a (possibly very) large number of cells – but NOT under control of the AAL-1 functionality.

Some provision needs to be made in case the sender process is sending at a slow enough rate so that waiting to fill an ATM cell would result in timing problems at arrival: a cell CAN contain fewer than 47 bytes of used payload space – that all depends on prior agreement of the two user processes.

Since AAL-1 manages constant bit-rate traffic, it must manage clock synchronization between sender and receiver, so that the rate at which bits enter the connection is the same as that at which they leave it. There appear to be two ways to do this.

**There is a Network-Wide Reference clock (Synchronous Residual Time-Stamp).** This clock is usually slower than the "Network Clock" (its period will be an integer multiple of the network clock period). As data is received, the number of **input clock cycles** in a Reference Clock Cycle is counted; this number is sent to the receiver, which sets up a matching output clock. The information is updated with some regularity to maintain synchronization. This method puts most of the burden on the network having and maintaining a strictly synchronized reference clock.
There is no Network-Wide Reference clock (Adaptive Clock). This requires maintaining a "constant length" queue at the receiver: it has to be large enough so that 47-byte fluctuations make no difference to the "discharge rate", and small enough so that an acceptable delay between source and destination can be maintained. When the queue fills up beyond a pre-set value, the discharge rate is increased, when it empties below another pre-set value, the rate is decreased.

The CSI bit. This has something to do with the actual physical transport used: it indicates that a second byte is used for transport management, giving a user payload of 46 bytes. We will discuss the details of this when we discuss the physical layer.

5.4 AAL−3/4.

This is the most complex of the AALs defined at this point as it attempts to deal with both connectionless (Class D) and connection oriented (Class C) data. It is derived from the SMDS standard, and was designed to be "compatible" with it - in the sense that SMDS traffic could be mapped to it with relative ease.

We thus begin by giving a quick summary of SMDS. The protocol was designed to handle LAN to LAN traffic, which was expected to be quite bursty: a relatively short high-speed transmission, followed by a period of silence. This was attained by using a packet that contained a source "telephone number" followed by a destination "telephone number", both of 15 decimal digits, both prefaced by a 4 bit code. Each decimal digit was packed into a distinct 4 bit segment. The telephone numbers could be set up so that broadcasting and "exit and entry screening" were included.

In order to avoid overcommitment of resources, the user would purchase an "average rate" and the router would have a counter incremented so as to support that rate. When a packet arrives at a router, the length of the packet (≤ 9188) is checked against the counter, and packets longer than the counter are discarded, while the others are passed, decrementing the counter by their length in bytes.

Given this picture, we must find a way to transform these packets - and others, from most any other protocol - into ATM packets: AAL−3/4 must take a user frame of any size appropriate to the data entering via the Service Access Point, transform it into ATM cells, and ensure that the reverse process can be carried out.

The Service Specific Convergence Sublayer may "massage" the AAL Service Data Unit (if this sublayer is not empty) and then pass the resulting data to the Common Part Convergence Sublayer.

The (massaged?) user data frame will be padded to a multiple of 4 bytes, and will receive a header (4 bytes) and trailer (4 bytes).

The new frame is now segmented into 44 byte segments. Each segment receives a 2 byte header and a 2 byte trailer. The last segment is padded to 44 bytes (if needed) before receiving the trailer.

The 48 byte frames are passed to the ATM layer for insertion into ATM cells.
What do the various headers and trailers contain?

The AAL Service Data Unit becomes the Common Part Convergence Sublayer Protocol Data Unit (CPCS-PDU – ain’t the alphabet soup wonderful??) The four bytes of the header are as follows:

Common Part Indicator: this tells us (possibly among other things) how the Buffer Allocation Size (the last two bytes of the header) and the Length (the last two bytes of the trailer) are to be interpreted – at the moment, the only value for the CPI is all zeros, meaning that the Buffer Allocation Size and Length fields are coded as the integer number of bytes used.

Beginning Tag: matches the End tag and is incremented for each successive Protocol Data Unit.

Buffer Allocation Size indicator: tells the receiver how much space to allocate for the rebuild of the user frame (at least as much as the incoming Protocol Data Unit).

The four bytes of the trailer (beyond the pad, which is added to ensure 32 bit width of all "important" fields) consist of an alignment byte (just to make up 32, for now), the End Tag (to match the Beginning Tag) and a two byte Length field, so that the actual PDU could have been anywhere between 1 and 65,535 bytes in length.

One of the implicit conclusions to be drawn is that, if the user wishes to send packets larger than 64KB, an earlier segmentation must be performed, and a higher layer (or, at least, the Service Specific Convergent Sublayer) will have to handle that function.

The Segmentation And Reassembly Layer takes the CPSC-PDU, segments it (44 byte segments), adds a two byte header and a two byte trailer to each segment, thus creating the SAR- Protocol Data Unit. The header and trailer must contain enough information to either reassemble the segments into the CPSC-PDU, or detect an error. The header consists of

A two bit Segment Type: 10 = beginning of message; 00 = continuation of message; 01 = end of message; 11 = single segment message.

A four bit sequence number. We assume that losing 16 cells in sequence is vanishingly unlikely – a long user packet may give rise to many cells with the same sequence number, so we can only use this number to detect errors – no recovery is possible here.

A 10 bit Multiplexing Identification Code.

The trailer consist of two fields:

A six bit Length Indicator – how many of the 44 bytes of the payload are "live".

A 10 bit Cyclic Redundancy Check field – computed in the usual manner: we are protecting $2^5 + 44*8 + 6 = 374$ bits, and $374 + 10 + 1 = 385 \leq 2^{10} = 1024$, so we have enough bits for the job. Again, the job of the CRC field is to DETECT errors in the SAR-PDU, but NO correction is attempted at this level.
One of the things that should be obvious by now is a near obsession with error checking; at the ATM cell header level for correct addressing; the data content of a cell is protected by the CRC of that cell; sequence numbers should catch dropped or out of sequence cells; the segment type lets us determine the beginning and the end of a message; the MID field must also be the same for all the cells in the sequence; the BTag and ETag fields must match; the Payload Length field must match the actual length. It may be possible to envision circumstances in which all of these safeguards are circumvented, but, we hope, the probability that this will actually occur is too small to worry about...

What is the Multiplexing Identification field? This identifies the CPSC-PDU from which the SAR-PDU is derived; all SAR-PDUs constructed from the same CPSC-PDU will have the same MID - so they can be recognized by the receiving SAR layer and reconstructed correctly. This allows a single Virtual Path/Virtual Channel (corresponding to a single connection) to carry multiple message streams - up to 1024 of them.

**Delivery Assurance.** With all these checks, what is the ATM responsibility as far as delivery?

i) *Assured operation.* In this case, it will be the Service Specific Convergence Sublayer that will have responsibility for managing retransmission of missing or corrupted packets. This seems to be still missing from the standard...

ii) *Non-assured operation.* All others.

Basically, the expectation appears to be that the end-user will implement whatever is needed at the transport layer to manage assured operation, with ATM defaulting only to delivering cells and their contents.

One of the problems this leads to is of classification within the ISO Reference Model: the *Data Link Layer* in some protocols supports some level of reliability (it is supposed to provide error free, transparent communication between network layer processes); the *Network Layer* usually does not do much; the *Transport Layer* is the one usually entrusted with the responsibility of guaranteeing error free communication to the *Application Layer* above it. We have several places where different responsibilities come into play. On the other hand, to confuse the issue, telephone companies would like to add some reliability properties to the *Network Layer*. Depending on the context, you should be prepared to understand what view of ATM somebody might have.

### 5.5 AAL-5.

As it turns out, a long series of discussions seems to have led to the conclusion that the AAL-3/4 protocol has too much overhead and provides services that may be just too much overkill. Accordingly, it appears that the AAL-5 protocol, which was introduced as a much cut down AAL-3/4, has won the day. It does does away with the per-cell sequence and integrity checks, leaving the decision on whether the message was received correctly to the software that sets up the reassembly. The protocol involves padding the original message to a multiple of 48 bytes. According to some sources a 4-byte header is under consideration (but is empty). There is an 8-byte trailer divided as follows
A 1-byte Common Part Convergence Sublayer User-to-User Indicator, which exchanges information between the two CPCs.

A 1-byte Common Part Indicator, whose use is still under study (zero for now).

A 2-byte length of payload field (64K byte max payloads).

A 4-byte CRC field.

The Length-or-Payload and CRC fields are used to determine integrity. Since there is no way in which the end-of-message can be extracted from the successive payloads, one of the bits of the ATM header needs to be used for the purpose. Bit 2 (the third) of the three bits of the Payload Type field is used for this purpose: bit2 = 0 means the cells is at the beginning of or interior to the message, bit2 = 1 means end of message.

One thing to observe is that the other protocols do not seem to require any "reassembly related services" from the ATM Layer itself, while AAL-5 requires a bit of the header to be set according on whether the cell forwarded is the last or not of a user message.

5.6 High-Speed Problems.

Tananebaum has some observations that are fairly relevant.

1) **Sequence Numbers.** We have seen that some kind of sequence numbers are used in ATM. In the Internet, sequence numbers have to be handled more carefully, for several reasons. ATM guarantees two things: order and no copies lurking about. The guarantee of order maintenance means that the sequence numbers will not be used to reassemble a large message that has been broken up, they are simply used as a further error check method. As such, they can be defeated, but with very low probability. The absence of copies means that there is no possibility that a copy of a message could arrive at the destination a second time and thus need to have its sequence number checked against previous arrivals to be discarded correctly. These are possibilities in the Internet Protocol, where datagrams can arrive out of order and they can also exist as copies in the network for as long as 120 seconds. Using $2^{16}$ or $2^{32}$ sequence numbers, which was quite adequate for 10Mb/s short networks is NOT adequate for thousands of kms long multi-Gb/s networks: you can run out of numbers well before the 120 seconds are up. If you up the sequence numbers to $2^{64}$ (which will be probably adequate for a few years), you start getting into size of the overhead - which is something you want to minimize anyway. Adding eight Byte sequence numbers in the 48 Bytes of the ATM cell payload would be unacceptable.

2) Communication vs Computing Speeds. The ARPANET of the 1970s ran at 56kb/s with computers that executed 1 MIPS. The packets were 1008 bits long and so a connection would handle at most about 56 packets/second, which meant that the average CPU would execute 18,000 instructions during a "packet time". This means that it could devote several thousand instructions to manage each packet, and have enough left over for "normal" processing. At Gb/s speeds and 100 MIPS the processor would have $4.24 \times 10^{-7}$ seconds to handle an ATM packet, or 42 instructions. Even assuming a 1000MIPS processor (this would require, at present, that all the pipelining works without any "rethinkings", and that memory access be performed at high enough
speeds), we are still dealing with 424 instructions. Probably not enough to manage anything, especially if the processor is doing something else. Dedicated processors require synchronization with a "main processor", and thus would not be an easy solution. One solution would be to increase the size of the packet to "several thousand bytes", but this would preclude using ATM for telephone transmissions. Packets that are non-uniform in length would require much more processing than short, uniform packets... So, you just can't win.

3) go back n error management. A bad packet recognized by the receiver results in no acknowledgment being sent and all subsequent packets being discarded. Then the sender will notice something amiss and re-send from the first unacknowledged packet. This is OK if there are relatively few packets "in the wire", since just a few packets will have to be retransmitted. This may not be a good strategy if there are packets for several megabytes already sent - and this is quite possible for connections of hundreds or thousands of kms. Furthermore, acknowledging each large (several KB) packet may be feasible, but acknowledging every ATM packet (53 Bytes) is definitely not: the backward acknowledgment traffic would take up just as much bandwidth as the forward "true" traffic. The solution that "piggy-backs" acknowledgments on regular data-frames will not make any difference because of the size of the ATM frames.

4) selective repeat error management. Buffer all packets following a bad one. Since acknowledgement will be missing, the sender will, eventually, notice, retransmit the bad packet, and go on from there. This requires a potentially very large amount of buffering. Since none of the ATM layers perform acknowledgement (and 3) above indicates why they may never do), who, and when and what is still a problem.

5) delay-limited vs bandwidth-limited. At 1Mb/s, the amount of time it takes to put the bits in the pipe dominates the total transmission time; at 1Gb/s it is the speed of light in the medium that determines how long the transmission is going to take - along with any queueing delays found along the path. The "time to put the bits in the pipe" is a very small percentage of the total (at least for medium to long distances). Protocols that wait until receiving packet acknowledgment to transmit the next packet will waste most of their time waiting, with nothing in the pipe. This is not acceptable for several obvious reasons, both of content delivery and economics.

6) Multimedia applications are new and require new transmission characteristics, which have to be worked into whatever protocol we design. One of their characteristics is keeping the variance in inter-arrival times very low.

Transmission (and re-transmission) efficiency would suggest large packets with small headers; heterogeneity of traffic requires that some of the packets be small (real-time voice interactions). The two requirements are contradictory, and the design of ATM reflects the contradiction. Any other high-speed protocol that would attempt to serve such a wide variety of traffic would run into the same problems: a different solution would make some kinds of traffic easier to handle, while making others harder.

5.7 Signaling AAL.

This protocol must provide reliable transport for signaling traffic. It uses a slightly modified AAL-5 protocol.
Figure 5.1: Signaling AAL Stack

The layers perform the following functions (from the top):

1. This maps the Service Specific Connection Oriented Peer-to-Peer Protocol (SSCOP) to the needs of the user of the Service Specific Coordination Function (SSCF). There may be multiple versions of this layer, depending on the needs of the end user.

2. This transfers variable length Service Data Units between users. It recovers lost or corrupted SDUs by re-transmission. It thus contains some kind of end-to-end protocol.

3. Builds header and trailer records onto user data frame. Assures integrity at the frame level. This must also contain some end-to-end protocol (possibly empty if the previous layer has adequate such facilities). The Peer-to-Peer protocol and SAR functions are, supposedly, only trivially different from those of AAL-5.

4. Converts CPCS frames into ATM cells, adding cell headers and trailers to provide integrity at the cell level (but AAL-5 does NOT provide integrity at the cell level beyond the services provided by the ATM Layer).

6 Signaling.

This explains how to set up and clear ATM connections (this function is carried out by the Transport Layer in the ISO-OSI model – Tanenbaum, Ch. 6). In some instances (e.g., X.25), the establishment of a connection involves security considerations (the delayed duplicate packet problem), but ATM leaves most of those difficulties out of its specification.

What is the Delayed Duplicate Problem? Timeouts may lead to retransmission. If the packet whose missed arrival caused the timeout was merely delayed, we now have two equal packets in the network. The problem of recognizing that one is now illegitimate is not easy to handle: sequence numbers may not be adequate, limited lifespans may not be adequate, etc..
The signaling process is used for dynamic management of connections at the User Network Interface. Connections can be permanent – and set up by OA&M (Operations Administration and Management) processes – or switched, in which case they are set up by signaling. Permanent connections are re-established by the network in case of failure, while switched ones must be re-established by the endpoint processes.

Signaling is carried out on dedicated VPI/VCI s (out-of-band signaling) – the ”normal” channels carry only user data traffic. A number of channels have been reserved for various functions: meta-signaling (i.e., the setting up of more signaling channels, if needed); broadcast signaling; point-to-point and point-to-multipoint signaling; etc. Some of these functions have been specified, others are still in discussion. Some (Class D), need nothing beyond connection to a ”connectionless” server. At the moment, no third party can request to set up a connection for others (except in case of an OA&M procedure to set up a ”permanent” connection).

Permanent connections are quite ”manual” – in the sense that it is not the signaling layer’s responsibility to consider anything beyond the parameters of the connections: compliance by the network endpoints must be assured separately. Switched connections are handled with better completeness: AAL parameters are passed and even higher layer functions (? which ones ?).

6.1 ATM Address Formats.

ITU-T (E.164) Format. The number of bytes devoted to each field is indicated over the field.

\[
\begin{array}{cccccc}
1 & 8 & 2 & 2 & 6 & 1 \\
AFI & ISDN Address E.164 & RD & AREA & End System Identifier & ESI \\
\end{array}
\]

*Figure 6.1: ITU-T Format.*

AFI = Authority and Format Identifier.
This is a ”telephone number” of some sort.
RD = Routing Domain.
AREA = Area Identifier.
ESI = End System Identifier.
SEL = NSAP Selector.

DCC (Data Country Code) Format

\[
\begin{array}{cccccc}
1 & 2 & 1 & 3 & 2 & 2 & 6 & 1 \\
AFI & DCC & DFI & Amin, Authority & Res. & RD & AREA & End System Identifier & ESI & SEL \\
\end{array}
\]

*Figure 6.2: IEEE 802 Format - DCC.*
DCC = Data Country Code.
DFI = Data Format Identifier (?).

ICD Format

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>1</th>
<th>3</th>
<th>2</th>
<th>2</th>
<th>6</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFI</td>
<td>ICD</td>
<td>DFI</td>
<td>Amin. Authority</td>
<td>Res.</td>
<td>RD</td>
<td>AREA</td>
<td>End System Identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ESI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SEL</td>
</tr>
</tbody>
</table>

Figure 6.3: OSI Format - ICD.

They all use a 20 byte address. Its is structured - so it uses more space than might be minimally necessary. On the other hand, the only "completely satisfactory" addressing scheme would be one that allows us to address every single electron in the universe: anything less will, eventually, be overrun by our greed.

6.2 Signaling Messages.

The following classes of signaling messages have been identified:

1) Call Establishment

   Setup
   Connect
   Call Proceeding
   Connect Acknowledge

2) Call Clearing

   Disconnect
   Release Complete
   Release

3) Status

   Status Enquiry
   Status

4) Point-to-Multipoint Messages Status

   Add Party
   Add Party Reject
   Drop Party
   Add Party Acknowledge
   Drop Party Acknowledge
   Drop Party Acknowledge
6.3 Meta-Signaling.

This will be used for the management of additional (beyond the default ones) signaling channels; the pair VPI = 0, VCI = 1 will be used for the purpose. It will allow the management of three kinds of signaling channels: i) Point-to-Point; ii) Selective Broadcast (Multi-cast); iii) General Broadcast. The messages have a very simple format and they use no AAL protocol. Three procedures are defined, with appropriate messages:

Channel Setup: with messages

<table>
<thead>
<tr>
<th>Assign Request</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denied</td>
<td></td>
</tr>
</tbody>
</table>

Channel Clearing: with messages

<table>
<thead>
<tr>
<th>Remove Request</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denied</td>
<td></td>
</tr>
</tbody>
</table>

Check Request: this checks the status of the system, either as a regular activity or as a timeout is experienced on a setup or clearing message.

The format of a meta-signaling ATM cell is given as follows:

<table>
<thead>
<tr>
<th>ATM Header</th>
<th>PD</th>
<th>MT</th>
<th>RI</th>
<th>PSVCI</th>
<th>BSVCI</th>
<th>CAU</th>
<th>BW</th>
<th>SPID</th>
<th>Fill With NULLs</th>
<th>CRC</th>
</tr>
</thead>
</table>

*Figure 6.4: Meta-Signaling Format.*

where

PD = Protocol Discriminator, 1 byte.

MT = Message Type, 1 byte.

RI = Reference Indicator, 1 byte.

PSVCI = Point-to-Point Signaling VCI, 2 bytes.

BSVCI = Broadcast Signaling VCI, 2 bytes.

CAU = Cause, 1 byte.

BW = BandWidth, 1 byte.

SPID = Service Profile Identifier, 1 byte.

CRC = Cyclic Redundancy Check, 10 bits.