ATM 101: A basic look at ATM technology

Asynchronous Transfer Mode (ATM) has recently been the subject of tremendous marketing promotion. ATM has been held out as the one-size-fits-all solution for the world's current and future communications needs. To better understand how ATM can actually be used - now and in the future - you need to understand the basic technology. In this article Lauren provides a very good introduction to data communication protocols and techniques, and then explains why ATM has become so popular.

By Lauren May

How we got to where we are today
Most of today's telecommunications networks are based on 2 fundamental concepts:

- Multiplexing
- Switching

Multiplexing
Multiplexing is used to share a resource by interleaving access to it. The most common form of multiplexing is Time Division Multiplexing (TDM). TDM allocates network bandwidth by first subdividing the data flow into *timeslots*, and then grouping some fixed number of timeslots into a frame.

In the simplest form of TDM, each multiplexed data stream is assigned one timeslot in each frame. Thus, each frame carries data from all of the data streams being multiplexed. Effectively TDM interleaves several independent lower-bandwidth data streams into a single higher-bandwidth data stream, for transport over a single path.

Switching
If multiplexing is not used, each data stream travels over a dedicated path, and *switching* determines which path the data stream takes each time the stream arrives at a network node. However, the situation is different when several independent data streams are multiplexed over a single TDM communication path. When a multiplexed data stream arrives at a node, some of the data streams may need to continue on together, while other data streams may need to be diverted into some other multiplexed data stream.

A TDM switch works by synchronizing itself to the beginning of an incoming frame, and then simply counting timeslots, to find the ones that need to be switched out of the stream. The same counting process is also used to locate the beginning of the next frame. Typically, individual timeslots contain a single bit but, in some instances, they might contain an octet (byte) or more.

Because this multiplexing/demultiplexing process is very deterministic and repetitive, time division multiplexing and time domain switching can easily be implemented in hardware. The relative simplicity and low cost of TDM hardware devices makes it possible to build cost-effective networks of widely varying:

- node count
- geographic extent
- capacity
- speed

This trait (called *scaleability*) has made TDM network technology extremely popular. In fact, the largest communication networks in the world (the public telephone networks) are based on TDM technology. Examples of TDM-based services include:

- T1
- E1
- Narrow-band ISDN
- Synchronous optical network (SONET).

Bursty communication
TDM networks are effective and very scaleable. However, they make rather inefficient use of the available transmission bandwidth when fed bit streams that are *bursty* in nature. Both the human voice and computer data tend to be very bursty. The consequence is that many timeslots go unfilled, and are therefore wasted.

When humans communicate with each other by voice, they are more concerned about immediate delivery of the data stream than they are about unused bandwidth, and are willing to pay the penalty for inefficient bandwidth use, as long as delays are negligible. To satisfy its subscribers, the telephone company provides a full bandwidth connection for the entire length of a call, and then charges the subscriber for that full bandwidth, whether it was used or not.

When computers communicate with each other, they are typically more tolerant of delays than humans. This has allowed the development of a technique that uses the bandwidth of a TDM network more efficiently. This technique is called *store and forward.*

Store and forward
Store and forward is just what the name implies - the bursty incoming data stream is stored as it arrives for some period of time. At some point, this stored data is then forwarded in a more continuous stream. This process of storing and then forwarding is repeated inside each node.

In one variation of store and forward, a separate call is placed to establish a temporary connection across the TDM network, after storing, and before forwarding. This scheme is *called message switching*. Message switching is only useful when the application is very delay tolerant, and when the cost of storage is less than the cost of wasted TDM bandwidth.

If the TDM network is transporting many different bursty data streams, it will probably be more efficient to store bursts when traffic is heavy, and then forward them when traffic is lighter. Since independent bursty sources typically generate their bursts

of data at different times, the interleaved data stream tends to be more continuous than its individual source data streams. In general, the larger the number of bursty sources, the more continuous the multiplexed data stream becomes.

With enough data sources, the combined stream becomes almost entirely continuous. However, there may still be times when the total bandwidth of all the combined bursts exceeds the bandwidth of the multiplexed data path. When that happens, portions of the individual data streams might need to be stored temporarily, until a lull in activity allows the stored data to be forwarded.

Statistical multiplexing

When the number of bursty sources is large and the information stream becomes continuous, the available bandwidth is almost entirely used. The number of bursty sources that can be accommodated in a multiplexed path depends upon the *average* data transmission rate of each source. Since the efficiency of such a multiplexed path depends upon the *average* behavior of the data traffic, this technique is called *statistical multiplexing*.

Handling variable-length bursts

Bursts from different sources are typically of varying lengths. To provide quick delivery of each burst, it's best to send the entire burst over the multiplexed path in one continuous data stream. However, some bursts might be very long. If a very long burst is sent in one continuous data stream, it might delay the transmission of shorter bursts from other sources.

To minimize the delay, the continuous data stream from any one source is limited to some maximum length. If the burst from that source is longer, the first portion of the burst might be sent, and then temporarily interrupted, to allow transmission of some other burst. Then the last portion of the burst would be sent.

To allow for handling of variable-length bursts (where fixed-length TDM counting schemes cannot be used) each burst in the source data stream is *delimited* with a mark as its beginning and end. Each delimited burst is then called a *packet*. Each packet is a variable-length string of octets (bytes).

Note: The term *frame* is sometimes used to describe a packet. However, this term is not recommended, since it leads to confusion with a TDM frame, which is a collection of *time slots* from several *different* data sources.

To provide the quickest delivery, packets from multiple data streams are typically transmitted on a first-come-first-served basis. Since packets might be arriving at the destination from several different sources, the receiver must sort the packets based on their source. To allow for packet sorting at the destination, each packet must have an *identifier* that uniquely tags its source.

Using statistical multiplexing with switched links

Statistical multiplexing was first used to allow a single point-to-point telecommunications link to carry multiple interleaved data streams from one location to another location. These early statistical multiplexing devices were known as *stat muxes* or as *concentrators*. These stat muxes provided what appeared to be a dedicated point-to-point link between each source/destination pair. Each of these links was called a *virtual circuit*.

Since all of the multiplexed packets originated at a single location, and were sent to a common location, the communication path was point-to-point, there was no need for switching between the source and the destination.

However, some subscribers wanted to have the bandwidth efficiency of statistical multiplexing, but also wanted to be able to send the multiplexed packets to *different* locations. This lead to a technology for switching statistically multiplexed packets, aptly called *packet switching*.

Packet switching

One of the most widely-deployed packet switching technologies was designed under the auspices of the CCITT international standards organization, and is commonly known as X.25.

Statistical multiplexors and packet switches are very dynamic and complex in their behavior, and require a microprocessor to control them, and memory to provide buffering. The memory cost and the software-based complexity of these devices make them much harder to scale than equivalent TDM devices that rely primarily on hardware implementations.

Most wide area networks (WANs) use the infrastructure of the telephone network to span large distances. Although the telephone networks of the developed nations have been digital for several years, subscriber *access* to those networks is still largely through analog circuits. The analog signals from individual subscribers are digitized when they reach the first switch in the telephone network.

These analog circuits are adequate for telephones, but analog signals are not very efficient for the transmission of the digital data streams originating in data communications equipment.

ISDN

In recent years, standards have been developed to allow direct connection of digital data sources to the digital telephone network. Several interface standards have been developed, but the most widespread and generally accepted is called Integrated Systems Digital Network, or simply ISDN.

The "Integrated Services" in ISDN comes from the concept that a properly-designed digital network can provide efficient service to a wide range of different device types, including:

- telephones
- facsimile machines
- digital data communications equipment
- low speed video
- high fidelity audio

Since many of these devices are already digital, it would actually be easier to build them with a digital interface than an analog interface. Current devices with analog interfaces (such as traditional telephones and video cameras) can be interfaced to the digital telephone network with an analog-to-digital *terminal adapter*, which incorporates a special circuit called a coder/decoder or CODEC.

Digital data streams can be compressed

In addition to converting analog signals to digital data streams, some CODECs also *compress* the data stream into a form that requires less bandwidth than the original signal. A matching CODEC at the destination then *decompresses* the data stream, to regenerate the original analog signal.

Some compression/decompression methods allow the original signal to be faithfully reproduced. However, other methods, which are designed for very high compression ratios, can regenerate only an *approximation* of original signal.

The human brain has built-in mechanisms for handling slightly degraded video or audio signals. However, in digital data communications the loss of some of the digital information is usually considered unacceptable.

Wide-area networks (WANs)

By evolution, wide-area networks have developed a hierarchical structure.

- The lowest tier is provided by the digital TDM transport and switching technology of the telephone network.
- The next tier is provided by *packet switching* that occurs in packet switching devices that are interconnected via the point-to-point telephone network.

Many computers and packet switching devices can be interconnected to form complex networks. The packet switching devices are located at intermediate points in the network, where they can direct incoming data streams from one point-to-point link to several different outgoing point-to-point links.

Devices attached at the edge of the network, or at intermediate points in a network, are called *nodes*. Software-based data communications protocols identify each data stream's source and destination, and route data streams through the appropriate network nodes to its destination.

In the 1970's and into the 1980's, most computers and packet switches were interfaced to the telephone network with modems. To compensate for the noisy and lossy characteristics of the analog network segments, protocols included complex (and rather inefficient) error detection and recovery mechanisms.

However, these mechanisms required each node to wait until a complete packet was received and checked for errors, before forwarding that packet over the next segment of the network. If an error was detected, the damaged packet was simply discarded, and a special *control packet* was sent back to the sending node. The sending node would then retransmit the packet.

This process required whole packets to be buffered in both the sending and receiving nodes. The sending node had to maintain its copy of the packet in an output buffer until the receiving node signaled with an acknowledging control packet that the entire data packet was received intact.

To lower the control packet overhead, multiple data packets were typically acknowledged by a single control packet. While this reduced the number of packets exchanged between the sending and receiving node, it also increased the amount of buffer storage needed in each node. Until the acknowledging control packet arrived, the sending node had to retain copies of all the data packets it had sent.

Thus, we see that there were 2 problems with this approach:

- Network nodes had to be equipped with large amounts of buffer memory.
- Each node introduced at least a packet length delay, because it could not begin sending a packet until the whole packet was received and verified to be correct.

The cost of the buffer memory, and the delays contributed by each node, discouraged the liberal use of intermediate nodes.

Local-area networks

However, several technological developments in the 1980s have brought about radical changes in the design of data communications networks. One of these was the introduction of cheap and powerful microprocessors and computer memory devices. Suddenly it became more cost effective to network small computers than to rely on a few large systems. The complexity of network topologies rapidly rose by an order of magnitude.

To provide for the interconnection of microcomputer-based systems, local area networks (LANs) were developed. LANs allowed several computers to be easily interconnected as long as they were located in the same building - or sometimes on the same campus. Rather than interconnecting these computers in a point-to-point manner with intermediate nodes, these LANs allowed several computers to be attached to a single *shared* communication path.

To allow the computers to communicate over this shared media without overwriting each other's transmissions, special LAN protocols were developed. These LAN protocols incorporated mechanisms that allow only one computer to transmit at any given time.

Interconnecting WANs and LANs

Since WAN protocols were developed to support *point-to-point* network topologies with intermediate nodes, and since LAN protocols were developed to support *shared* communication links, they were used in relative isolation. Typically WAN protocols were not used on LANs and conversely, LAN protocols could not be used on WANs. To make matters worse, LANs typically operated at speeds 10 to 100 times faster than most WANs.

To allow computers on one LAN to communicate with computers on another LAN, specialized packet switching devices were developed to communicate between LANs over WANs. The simplest of these devices are called *bridges*.

Bridges

Bridges take a packet from one LAN, transport it verbatim across the WAN, and then transmit it over the remote LAN, just as if it had originated from some computer on the destination LAN.

This approach provides only a limited solution, because it is not scaleable. Each connection between LANs requires a bridge at each end, and a dedicated point-to-point WAN link between those bridges.

Additionally, the network layer protocols of the first LAN must match the network layer protocols of the second LAN.

Routers

This problem was overcome by increasing the capabilities of bridges by combining them with devices called *routers*. Routers can connect networks using the same *transport layer* protocols, but different *network layer* protocols. These combined devices, known as bridge/routers (or more simply as *brouters*) can route packets between multiple LANs over packet switched, as well as TDM based, WANs.

Routers vs. packet switches

Routers differ from traditional packet switches in a subtle, but very important way. Packet switching schemes (such as X.25) require a *logical connection* be established between the source and destination *before* any packets can be sent. To reach its

destination over a packet switching network, a packet carries the identifier of the *connection*, not the *address of the destination*. This is called *connection-oriented* service.

In contrast, packets in a router-based network each contain the *address of their destination*, and therefore do not require a logical connection to be established between routers. This is called *connectionless* service.

Connection-oriented vs. connectionless services

The pros and cons of connection-oriented and connectionless services are heavily debated. The details of that debate could easily fill an entire article. For a much more detailed discussion of network technologies (including traditional packet switching and routing) see TANNENBAUM [1].

In Europe, most WANs are based on X.25 packet switched networks. However, in the US, most WANs have been based on dedicated point-to-point telephone network TDM circuits. However, this has not prevented interconnection of these WANs. Routers have been developed that can relay packets between packet switched and TDM-based WANs. Probably the best-known example is the Internet, which is a huge global data network that runs over a patchwork of TDM and packet switched carrier services.

TCP/IP

In the US during the 1970s, a very fortunate development occurred. A very adaptable set of communications protocols called Transmission Control Protocol and Internetwork Protocol (TCP/IP) were developed for communication between computers manufactured by different companies.

The development of TCP/IP was funded by the US Department of Defense, to allow it to build large computer networks, using computers from different companies. TCP/IP was essentially a common protocol suite for transferring packets between computers over WANs.

TCP is a computer-to-computer communication protocol that allows the destination to verify that each packet arrived intact. TCP does not assume that intermediate WAN nodes detect or correct errors. Instead, the destination computer is responsible for detecting errors and requesting retransmission from the source computer.

IP provides the information needed to route a TCP packet through any intermediate networks to the destination.

TCP does not assume that the underlying network service is connection-oriented. This allows TCP to be used over connectionless environments (such as LANs or IP) as well as over connection-oriented networks (such as X.25 or the telephone network).

Because of this, TCP has been widely adopted for use on LANs. Then, as routers were developed to interconnect LANs, IP (and a number of related protocols) were used to route from one LAN to another. As more and more LANs were interconnected over WAN links, TCP/IP protocols became the key protocols in the global Internet.

Frame relay

In the last 5 years, 2 more significant events have occurred.

- Digital access circuits to telephone networks became widely available and affordable.
- A new simplified packet switching network technology, called Frame Relay, was developed.

Frame Relay is essentially a stripped-down version of X.25. It removes the burden of packet retransmission from the intermediate packet switching nodes. (This is practical because fully-digital circuits have very low error rates.) When errors do occur, they are handled by transport protocols (such as TCP/IP) which place all the responsibility for retransmission on the destination and source computers.

By eliminating the need for retransmission, Frame Relay greatly reduces the buffering requirements in intermediate nodes and the algorithm complexity within each node. This has allows the construction of Frame Relay networks that provide good performance, are cost effective, and are reasonably scaleable.

In the last few years, many dedicated point-to-point WAN circuits between routers have been replaced by Frame Relay circuits. In fact, the rate of acceptance of Frame Relay has been so fast that AT&T has not been able to keep up with demand.

Personal computers are fueling rapid networking growth

The number of personal computer workstations continues to mushroom. Also, personal computers are becoming increasingly powerful and affordable. Many people now have computers at home, as well as at work, and want to interconnect these computers. WANs provide a way to do this.

In addition, computers are constantly being connected to existing LANs. As the number of computers attached to a LAN increases, users find it necessary to segment the LAN, to keep down the number of computers competing for the LAN's bandwidth. Bridge routers are typically used to do this.

As the total number of LANs increase, the number of needed interconnections between these LANs also increases, causing the number of bridge routers between LANs to increase dramatically. This causes a tremendous administrative headache for LAN managers.

The traditional concept of a personal computer has been a stand-alone machine that *collects* and *processes* information. However, personal computers are increasing being seen as simply a component in a vast communication network which allows users to *exchange* information.

Correspondingly, WANs are no longer seen as just a way to get information from one computer to another. They are now seen as a way to allow computers to access resources on remote LANs, just as if it were directly attached.

Unfortunately, WANs are in most instances 10 to 100 times slower than LANs. Certain forms of information transfer (particularly telephony and video) do not tolerate delay very well. Some high-bandwidth information forms, (such as full motion video) tax the resources of even the fastest LANs, and completely outstrip the capabilities of all but the fastest and most expensive WANs.

Skyrocketing bandwidth demands can only be met with a highly scaleable networking technology

The function of computers and televisions is converging. Computers are rapidly becoming both information and entertainment centers. In recognition of this fact, telephone and television cable companies are squaring off to compete against each other. To many it seems that our current communication networks will not scale up sufficiently to keep up with the demand for higher bandwidth connectivity.

To compound this problem, the telephone companies make the large majority of their multibillion dollar revenues from a single low-bandwidth service - voice telephony. If these big telephone companies make large numbers of relatively inexpensive high-bandwidth circuits available, enterprising independent telephone service providers will likely use those circuits to build networks that bypass the services of the big telephone companies, potentially causing huge revenue losses. As a result, large telephone companies are strongly demotivated from providing affordable high bandwidth circuits until they see a major revenue source for high bandwidth service. The current candidate to replace telephony is video entertainment and communications.

ATM is highly scaleable

ATM is a unifying technology that provides scaleable and cost efficient solutions to the problems listed above, by providing:

- Efficient support of many different commercial applications - voice, data, facsimile, video.
- Seamless integration of different operating environments - LAN and WAN; home and office.
- Compatibility and effective integration with the existing infrastructure - largely TDM or packet based.
- Cost effective support of the existing installed base - particularly phone, lease line data, and cable TV.
- Flexible and cost effective support of a wide variety of bandwidth needs

To see how ATM might solve these varied problems, let's take a close look at the fundamentals of ATM.

ATM basics

ATM is an implementation of a technology called *cell relay*. Cell relay is effectively a hybrid, combining the best traits of both TDM and packet switching. The basic switched unit of cell relay (the cell) is similar to a packet, except that a cell always has the same length.

An ATM cell is always exactly 53 octets (bytes) long. Each cell starts with a 5-octet header, which contains information identifying the cell's destination, and also specifics about the cell's *payload*. The remaining 48 octets of a cell contain its payload, which is the information being transported by the cell. Roughly speaking, 10% of each cell is consumed by this 5-octet overhead.

ATM cell formats

The format and the content of ATM cell headers vary, depending upon the payload, and where in an ATM network the cell is used. Cell structures used between terminal devices and ATM switches (at the edge of an ATM network) have a somewhat different format than the cells used inside the network (between ATM switches).

Separate interface specifications exist for terminal-to-switch and switch-to-switch interfaces. The former is called the User-Network Interface (UNI) and the latter is called the Network-Network Interface (NNI). Additional interface specifications exist for Private Network-Network Interface (PNNI), and Intercarrier Interface (ICI).

The ATM Forum

These ATM specifications (and several more) were developed under the auspices of the *ATM forum* [2], a private non-profit industry consortium made up of equipment manufacturers, carriers, semiconductor manufacturers, and network users.

Initially, most of the ATM forum membership was from North America, but the forum has actively promoted international membership, and holds a number of international meetings each year. The successor body to the CCITT (the ITU) is developing an international standard for ATM that encompasses much of the work of the ATM forum.

ATM topologies

Each *terminal device* in an ATM network is attached directly to an ATM switch. Simple networks might be star-configured, with one multiported ATM switch at the center of the star. Larger ATM networks are typically tree-structured, with an ATM switch at each branch. In this way, ATM networks are very similar to TDM-based telephone networks.

ATM data streams

At the physical level, ATM cells are transmitted in a *continuous stream*. When no payload is available to fill a cell, it is simply sent empty - much as individual timeslots may be left empty in a TDM stream. The use of a continuous stream of cells of equal length makes it possible to implement the basic process of cell switching in *hardware*, using counting schemes akin to those used for TDM switching.

ATM bandwidth

The aggregate bandwidth of all the ATM switches at the edges of the network is typically greater than the bandwidth available across the network. However, if the bandwidth of all the cell streams arriving at an intermediate ATM node exceeds its available outgoing bandwidth, that node is permitted to discard any *empty* ATM cells, and use the outgoing bandwidth that would have been consumed by those empty cells to send full cells from some other cell stream.

This ability to discard empty cells, allows a network to support a large number of bursty cell sources. The empty cells are simply discarded wherever additional bandwidth is needed within the network.

The ability to insert or discard empty cells also makes it possible to join ATM carrier circuits operating at significantly different rates, using an ATM switch or multiplexor. This ability to easily adapt to varying rates allows ATM to scale from relatively low rates up to extremely high rates.

The main limiting factor at both ends becomes cost. Below T1 speeds, (1.554 Mbits/sec) ATM's high overhead ratio makes it progressively less cost effective than protocols specifically intended for lower-bandwidth operation. At extremely high speeds, the cost and complexity of building very large and very fast switching hardware becomes a limiting factor.

However, it is possible to use multiple parallel trunks and switches, to provide effective transmission rates much higher than the maximum rate for a single trunk and switch.

ATM vs. TDM

ATM resembles TDM packet-switching networks when combining streams from different cell sources. However, in a TDM

network, unused timeslots are wasted. In an ATM network, the ability to replace empty cells from one cell source with the full cells from some other source allows full use of the available bandwidth.

ATM is connection-based

Before cells can be sent across an ATM network to a given destination, a network route must be determined and assigned between the source and destination terminal devices.

The assignment of an ATM cell stream to a specific route across an ATM network is called a *virtual connection*. A virtual connection can be provided by a pre-allocated permanent connection, or it can be created dynamically by placing an *ATM call*. A virtual connection over a permanent connection is called a *permanent virtual connection* or PVC. A connection created by an ATM call is termed a *switched virtual connection* or SVC.

Each virtual connection is assigned a unique identifier, called an *ATM address*. For PVCs, ATM addresses are pre-assigned. For SVCs, the ATM address is assigned by the endpoint switches at the time the call is placed, for the duration of the call. Every cell sent across a connection contains an ATM address. Typically, an ATM address has only local significance, and therefore, might change as the cell travels end-to-end.

An ATM terminal device can have zero or more SVCs and or PVCs active at the same time. ATM is intended to transport different kinds of payload information between endpoints. The same 2 endpoints might be connected for several different payload types. In this instance, to conveniently differentiate payload types, separate virtual connections could be used simultaneously between the 2 end points. For efficiency's sake, virtual connections that share the same endpoints can be routed identically across the network, using a single virtual path. Accordingly, an ATM address is made up of two subfields: the *virtual path identifier* and the *virtual connection identifier* - VPI and VCI, respectively.

Traffic contracts and quality of service

As part of the process of an ATM call, the network and the calling terminal negotiate a *traffic contract* which guarantees a specific quality of service (QOS) for all cells that do not exceed a corresponding maximum cell rate.

This traffic contract includes QOS related parameters, such as:

- maximum burst size
- sustainable cell rate
- peak cell rate
- cell misinsertion rate
- cell delay variation

The need to guarantee cell delay variation is the reason for the 48-octet payload length of ATM cells. For data networking, a larger 64-octet payload size would have imposed less cell overhead, and yielded a higher throughput at low data rates. However, for voice telephony, a 64-octet payload might result in too high a cell delay variation. Telephone network designers advocated a 32-octet payload. As a compromise, the 48 octet payload was chosen.

To complete a call across an ATM network, adequate bandwidth must be available end-to-end to provide the requested QOS. If not, the call is rejected and must be placed again, with a lower level of requested QOS. If the network is heavily loaded, this process might need to be repeated several times.

Handling heavy load conditions

In very heavy load conditions, a connection may simply not be possible. A similar situation occurs in a traditional TDM network when calls are temporarily blocked because all timeslots are already allocated in some segment along the path across the network. When this happens, the telephone subscriber hears a fast busy signal.

In a TDM network, this is an all-or-nothing situation. Any call successfully connected in a TDM network is guaranteed its full allocation of bandwidth for the duration of the call. However, if that full allocation is not available, no connection is possible.

In the case of an ATM connection, the subscriber can actually experience a better-than-guaranteed quality of service when network traffic is light. This is because a terminal device (such as a video CODEC) can exceed its traffic contract. When it does, any cell in excess of the contracted cell rate will have its *cell loss priority* (CLP) bit set to 1 by the ATM switch attached to the terminal.

If such a cell encounters no congestion on its trip to the destination, it might traverse the entire network without being discarded. In this way, ATM resembles statistical multiplexing, allowing unused bandwidth allotted for one connection to be used by other connections.

Temporary saturation of bandwidth in a packet network or a cell network causes a condition called *congestion*. As congested levels are approached in any ATM switch in the network, the switch sets a congestion indication bit (CI) in all the cells that it forwards. However, before the congestion indication can propagate out to the edges of the network (to cause terminal devices to temporarily slow their cell rates) the congestion may become worse. If congestion increases, the switch begins to discard an increasing number of cells that are in excess of their traffic contracts, and therefore have their cell loss priority (CLP) bit set.

A generic cell rate algorithm (GRCA) can be used to choose (on a cell-by-cell basis) which cells to discard. Typically, a GRCA provides a degree of fairness by tracking traffic contract compliance for each virtual connection. Cell discarding is weighted against the connections with more recent and/or larger traffic excesses.

As a last resort, the switch will discard cells that do not have their cell loss priority bit set. Note: This should never be necessary if all traffic contracts are negotiated to avoid exceeding the capacity of the network. A helpful discussion of traffic management can be found in FLANAGAN [3].

Adapting ATM

To serve the wide range of applications envisioned for ATM, several different *classes* of information must be efficiently transported. These classes can be grouped into 2 main types:

- constant bit rate (CBR) user data
- variable bit rate (VBR) user data

Four classes of user data (labeled A through D) are currently defined:

- Class A includes all CBR services including standard voice telephony, TDM network trunks, and certain forms of video.
- Class B typically supports analog VBR services, such as packetized and compressed voice and video.
- Class C includes traditional connection-oriented data communication protocols, such as Frame Relay, TCP, SNA and X.25.
- Class D includes connectionless (CL) datagram-based communications protocols.

Note: Datagrams are packets that contain enough addressing information to allow each datagram to be individually routed, without first establishing a *connection*. ATM does not inherently provide this routing capability. Instead, it must be provided by a higher level service such as Switched Multimegabit Data Service (SDMS).

Most applications within these classes are designed to transport units of information larger than the 48-octet payload of an ATM cell. To transport these larger information objects, ATM terminals provide a fundamental service called *Segmentation and Reassembly* (SAR). It splits large information objects into 48-octet cells as they are transmitted, and then recombines them back into their original form at the destination, as they are received.

The ATM layering

ATM cell generation (and the mating of ATM cell streams with underlying carrier services, which are typically TDM based) such as...

- T1 (1.5 Mbits/sec)
- DS3 (45 Mbits/sec)
- OC3 (150 Mbits/sec)

...are considered functions of the ATM *Physical Layer*.

The switching of ATM cells is performed in the next higher layer, called the *ATM Layer*. This switching is provided by hardware devices.

The next higher ATM Layer is the *ATM Adaptation Layer*. Segmentation and Reassembly (SAR) is the lowest level sublayer of the ATM Adaptation Layer. Several semiconductor device manufacturers have introduced first generation SAR hardware devices. Additional SAR devices will no doubt be developed by many of the major semiconductor manufacturers, as the technology matures, and as the market grows.

In order to provide the classes of services required by ATM applications, several ATM Adaptation Layer Protocols have been specified to run above the SAR sublayer. Currently there are 4 such protocols specified:

- AAL1
- AAL2
- AAL3/4
- AAL5

AAL1 - Class A constant-bit rate

AAL1 supports Class A constant-bit-rate user data. A key component of this protocol is clock recovery, allowing TDM streams to be transported with minimal frame slippage. Class A service is categorized as *unreliable* because it does not provide for any form of error recovery.

AAL2 - Class B variable-bit rate

AAL2 supports Class B user data. This protocol is intended to serve variable-bit-rate user data that requires clock recovery. This service is also unreliable. AAL2 is still being specified. The complex requirements of supporting different compressed video formats will likely require refinements to this protocol for some time.

AAL3/4 - Some Class C and all Class D variable bit rate

AAL3 and AAL4 were originally conceived as separate protocols, but were later combined to form AAL3/4. AAL3 was originally designed to provide reliable support for Class C variable-bit-rate connection oriented data transfers. However, the relatively high level of overhead it imposed to provide error correction for large bulk data transfers made it less than ideal for burst-oriented applications, such as bridging LAN packets. These applications were subsequently shifted to the “lighter weight” AAL5.

AAL4 was devised to support Class D variable-bit-rate connectionless data. AAL3 and AAL4 had enough common requirements for strong error correction facilities that they were combined, to form AAL3/4. AAL3/4 includes facilities to facilitate segmentation of the very large data units that AAL3 was originally designed to support. Each segment is accompanied by a Cyclic Redundancy Code (CRC) to support error correction. AAL3/4 also contains facilities to directly support multiplexing.

AAL5 - Additional Class C

AAL5 is used to support Class C variable-bit-rate connection-oriented data when high throughput is of greater concern than an occasional retransmission. AAL5 is streamlined to minimize overhead. The error detection and multiplexing facilities of AAL3/4 are dropped to provide more room in each cell for the actual data payload.

These services provided by the ATM Adaptation Layer are just building blocks that allow various applications to use the underlying capabilities of ATM. In many applications, software is needed to implement a portion of the AAL. The long-term prospect is that hardware devices will largely absorb this function, increasing speed and driving costs down.

Making ATM connections

Most applications using ATM require the ability to create and terminate Switched Virtual Connections. It was decided early on that it would be most expedient to adopt an existing proven protocol to support ATM calling. Both the ISDN Digital Signaling System Number 1 (DSS1) and Signaling System Number 7 (SS7) were considered as models.

Over objections that it might prove to be too inefficient, ISDN DSS1 was chosen. The core protocol of DSS1 (Q.931) was then adapted to the task of ATM signaling. The new protocol was originally dubbed Q.93B. The ITU has subsequently undertaken the task of providing formal specifications for ATM signaling, renaming it as Q.2931.

The signaling capabilities provided by Q.2931 are quite powerful, but the protocol is also quite complex. The task of implementing this protocol requires a man-year or more of investment. Most equipment manufacturers entering the ATM market have chosen the simple expedient of purchasing ATM signaling software from a vendor that specializes in signaling protocols.

One good example of this ATM signaling software is a product called Adaptable ATM Software from Telenetworks. It provides

a very efficient implementation of the ATM forum's specifications for Q.2931 - both UNI 3.0 and 3.1, and the variant of AAL required for signaling known as the Signaling ATM Adaptation Layer (SAAL).

To facilitate customization, this software is licensed in C source code form, and has many features that allow it to be adapted to both workstation and embedded applications. The company also provides custom ports and device driver development services for customers looking for even quicker time to market. Ω

References:

[1] Computer Networks, Second Edition by Andrew S. Tannenbaum. Prentice Hall ISBN 0-13-162959-X.

[2] The ATM Forum can be contacted for specifications and membership information at 415-578-6860 (phone), 415-525-0182 (FAX), or info@atmforum.com (e-mail). An anonymous FTP site (for approved specifications only) is located at info@atmforum.com. A FAX-on-demand server is available at 415-688-4318. A World Wide Web home page can be examined at http://www.atmforum.com. The ATM Forum's Worldwide Headquarters is located at: 303 Vintage Park Drive, Foster City, CA 94404-1138.

[3] ATM User's Guide by William A. Flanagan. Flatiron Publishing ISBN 0-936648-40-6.

Lauren May is the Director of Marketing at Telenetworks. Lauren has a 15 year software engineering background in data communications including a BS in Computer Science from California State University Northridge. He joined the Telenetworks team in 1991 after completing a very successful ISDN terminal adapter project that incorporated Telenetworks Adaptable ISDN Software. Lauren's current responsibility is the promotion and licensing of the *Telenetworks Adaptable ATM Software*.

About Telenetworks

Telenetworks has been providing ISDN signaling software to the communications industry for over 8 years, and has the largest customer base in the business. Much of today's field proven ISDN equipment uses Telenetworks' ISDN Adaptable Software, including most of the bridge/router and remote access products manufactured by North American equipment vendors.

If you have questions about Telenetworks ATM Adaptable Software, or other communication protocol products (including ISDN BRI and PRI signaling, Frame Relay, Multi-link PPP, and X.25) you can contact Telenetworks at:

Telenetworks
625 Second Street #100
Petaluma, California 94952
Tel: 707-778-8737
Fax: 707-778-7476
email: info@tn.com.