CHAPTER II
LITERATURE REVIEW

In order to meet the objective of this project, areas such as ATM network, ATM traffic management (congestion control), dynamic bandwidth allocation scheme, fair bandwidth sharing, buffer occupancy, artificial Neural Network, and Fuzzy Logic are surveyed.

2.1 Asynchronous Transfer Mode (ATM)

ATM was developed as part of the work on Broadband Integrated Services Digital Network (B-ISDN). ATM supports several data rate and allows multiple logical connections to be multiplexed over a single physical interface. A fixed-size packets (cells) flow in these logical connections. ATM network is suitable for integration of variety of traffic because of the fixed-size cell that reduces the variance of delay. ATM takes advantage of reliability and fidelity of modern digital facilities to provide faster packet switching than X.25.

2.1.1 ATM Network Operation

A typical ATM network consists of a set of ATM switches is interconnected by point-to-point ATM links or interfaces. ATM switches support two kinds of interfaces:

- user-network interfaces (UNI), and
- network-node interfaces (NNI), sometimes also known as network-network interfaces.

UNI connects ATM end-systems (hosts, routers, and so on) to an ATM switch. An NNI may be imprecisely defined as an interface connecting two ATM switches together. There are slightly different cell formats defined across UNI and NNI. In NNI cells, there is no Generic Flow Control (GFC) field, and the first four bits of the cell are used by an expanded (12 bit) VPI field. The GFC is rarely used. In practice, no functional difference between UNI and NNI cells, other than in the fact that the NNI cells can support a larger VPI space.

A NNI is any physical or logical link across the two ATM switches using the NNI protocol. For this reason, the connection between a private ATM switch and a public ATM switch is a UNI, known as a Public UNI, since these switches do not typically exchange NNI information [6].

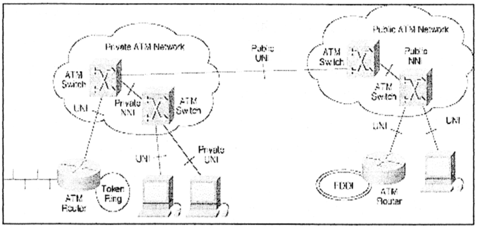


Figure 2.1 ATM network interfaces [6].

2.1.2 ATM Protocol Architecture

Based on the ATM standards issued by ITU-T, the protocol architecture is shown in Figure 2.1. This figure illustrates the basic architecture for an interface between user and network. ATM is a layered architecture allowing multiple services like voice, data and video, to be mixed over the network as shown in Figure 2.2.

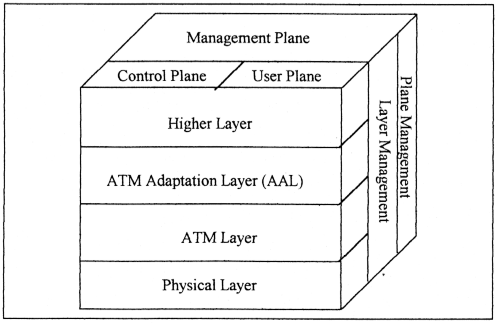


Figure 2.2 ATM protocol architecture.

Three lower level layers have been defined to implement the features of ATM. The use of ATM creates the need for an adaptation layer to support information transfer protocols that are not based on ATM.

The ATM Adaptation Layer

The AAL interfaces the Higher Layer protocols to the ATM Layer. This layer relays ATM cells both from upper layers to the ATM Layer and vice-versa. AAL segments the data into ATM cells when receive information from the Higher Layer. When relaying information to the Higher Layer, the segmented information must be reassembled into a format that the Higher Layer can understand. This is called Segmentation and Reassembly (SAR).

There are four types of AALs, each supporting a different type of traffic or service. Associated with each service class is a different adaptation function/protocol. These 48-byte are then relayed to a Lower Layer, which is concerned with adding the correct cell header information. The service classes and the corresponding types of ALLs are [8]:

1. Class A – AAL 1: Supports a connection-oriented service in which the bit rate is constant.
2. Class B – AAL 2: Supports a connection-oriented service in which the bit rate is a variable. Requires bounded delay for delivery.
3. Class C – AAL 3/4: Supports a connection-oriented data service in which the bit rate is a variable. Does not requires bounded delay for delivery.
4. Class D – AAL 5: Supports a connectionless data service.

The ATM Layer

The ATM Layer provides an interface between the AAL and the Physical Layer. The ATM Layer receives a stream of cells from the Physical Layer and transmits cells with new data or empty cells if there is no data to be sent. When residing in a switch, the ATM Layer determines where incoming cells should be forwarded. Further more, it buffers incoming and outgoing cells, and handles various traffic management functions such as cell loss priority marking, congestion indication, and generic flow control access. This layer also monitors the transmission rate and the conformance to the service contract [8].

The Physical Layer

This layer defines the bit timing and other characteristics for encoding and decoding the data into suitable electrical/optical waveforms for transmission and reception. This layer also provides cell delineation function, Header Error Check (HEC) generation and processing, performance monitoring, and payload rate matching of the different transport formats used at this layer [8]. Synchronous Optical Network (SONET), a synchronous transmission structure, is often used for framing and synchronisation at the Physical Layer.

2.1.3 ATM Logical Connections

An ATM connection is identified by its Virtual Channel Identifier (VCI)/Virtual Path Identifier (VPI) value. ATM networks are fundamentally connection-oriented. This means that a virtual circuit needs to be set up across the ATM network prior to any data transfer. There are two types of ATM circuits:

- virtual paths, identified by VPI, and
- virtual channels, identified by the combination of a VPI and a VCI.

A virtual path is a bundle of virtual channels (as shown in Figure 2.3). All VCI and VPI, however, have only local significance across a particular link, and are remapped, at each switch. In normal operation, switches allocate all UNI connections within VPI=0.

Logical connections in ATM are referred to as Virtual Channel Connection (VCC). A VCC must set-up between 2 end users before the transmission begins. This VCC has variable bit rate. When information needs to be communicated, the sender negotiates a "requested path" with the network for a connection to the destination. When setting up this connection, the sender specifies the type, speed and other attributes of the call. Sometimes, VCCs are used for user-network exchanged in control signalling and network-network exchanged in network management and routing.

The concept of Virtual Path is introduced in ATM. A Virtual Path Connection (VPC) is a bundle of VCCs that have the same end points. All of the cells flowing over all of the VCCs into a single VPC are switched along the same path. The virtual path concept is such that the controlling effort is made easier by grouping connection sharing the same paths into a single unit. Network management concentrate on the small group of VP rather than a large number of VC.

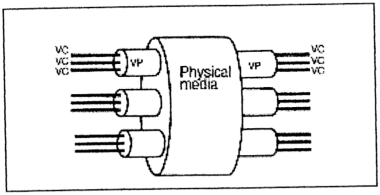


Figure 2.3 ATM connection relationships [42].

There are several advantages using virtual path technique [4] as follows:

- Simplified network architecture.
- Increased network performance and reliability.
- Reduced processing and short connection setup time.
- Enhanced network services.

At both end points of a VCC, there can be end-users or network entities. With the VCC, the cell sequence integrity is preserved (cells are delivered in the same order in which they are sent). There are 3 kinds of VCC:

- User-to-user: Mainly for transporting the end-to-end user data and control signalling. The VPC between 2 end points is with sufficient resources. Extra VCC can be set up, provided that the set of VCCs doesn't exceed the VPC capacity.
- User-to-network: Mainly for transporting the user-to-network control signalling. May be used to carry traffic from end user to a network exchange or network server.
- Network-to-network: Used for network traffic management and routing functions. A network-to-network VPC can be established as a common route for the exchange of network management information [4].

Table 2.1 shows the terminology of virtual path and virtual channel.

Table 2.1 Virtual Path/Virtual Channel Terminology

Virtual Channel (VC)	A generic term used to describe unidirectional transport of ATM cells associated by a common unique identifier value.
Virtual Channel Link	A means of unidirectional transport of ATM cells between a point where a VCI value is assigned and the point where that is translated or terminated.
Virtual Channel Identifier (VCI)	A unique numerical tag that identifies a particular VC link for a given VPC.
Virtual Channel Connection (VCC)	A concatenation of VC links that extends between two points where ATM service users accesses the ATM layer. VCCs are provided for the purpose of user-user, user-network or network-network information transfer. Cell sequence integrity is preserved for cells belonging to the same VCC.

Table 2.1 Virtual Path/Virtual Channel Terminology (continued)

Virtual Path (VP)	A generic term used to describe unidirectional transport of ATM cells belonging to virtual channels that are associated by a common unique identifier value
Virtual Path Link	A group of VC links, identified by a common value of VPI, between a point where a VPI value is assigned and the point where that value is translated or terminated.
Virtual Path Identifier (VPI)	Identifies a particular VP link.
Virtual Path Connection (VPC)	A concatenation of VP links that extends between the point where the VCI values are assigned and the point where those values are translated or removed. VCs are provided for the purpose of user-user, user-network or network-network information transfer.

Virtual Path/Virtual Channel Characteristics

Another key concept is that ATM is a switched based technology. By providing connectivity through a switch (instead of a shared bus), several benefits are gained, namely:

- Dedicated bandwidth per connection
- Higher aggregate bandwidth
- Well defined connection procedures
- Flexible access speeds

Control Signalling

Control Signalling is the information involved to establish and release the VPCs and VCCs. This signalling takes place in separate connections from those that are being managed. One or more of the methods below will be used to establish and release the VCCs and VPCs:

1. Semi-permanent VCCs may be used for user-to-user exchange. In this case, no control signalling is required.
2. If pre-established call control signalling channel is not available, a new call Control Signalling channel must be setup. A control signalling must take place between the user and the network. Hence, a low data rate permanent channel can be used to setup VCCs that can be used for call control. This channel is called a meta-signalling channel because the channel is used to setup signalling channel.
3. Meta-signalling channel can be used to setup a VCC between the user and the network for call control signalling. This user-to-network signalling virtual channel can then be used to setup signalling channel.
4. The meta-signalling channel can also be used to setup a user-to-user signalling virtual channel. Such a channel must be setup within a pre-established VPC. This allows users to establish and release user-to-user VCCs without network intervention [4].

ATM signalling is initiated by an ATM end-system that desires to set up a connection through an ATM network. Signalling packet are sent through a well-known VC. This means that this VC is reserved for signalling traffic, and no other types of information can be transmitted across this connection. All switches are also pre-configured to receive any signalling packet and pass them to a signalling process. In general, all VCI of value below 32 will be reserved for such control purposes. Hence, for VCI 32 and above will be allocated to data connections.

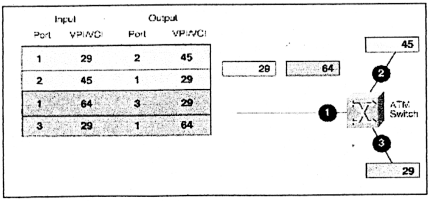
The signalling is routed from switch to switch. However, for the sake of robustness and performance, most of the vendors will integrating the call control capability into each switch, rather than supporting them on an off-board processor. When the signalling packet reach the destination end-system, the end-system can either accept the connection request, or reject it. Note that because the connection is set up along the path of the connection request, the data also flows along the same path.

The connection identifiers (that is, VPI/VCI values) for a particular connection are typically allocated in each direction of a connection, but the traffic parameters in each direction can be different.

2.1.4 ATM Switching Operation

The operation of an ATM switch (refer Figure 2.4) is as follows:

1. Receive a cell across a link on a known VCI or VPI value
2. Looks up the connection value in a local translation table to determine the outgoing port (or ports) of the connection and the new VPI/VCI value of the connection on that link
3. Retransmit the cell on that outgoing link with the appropriate connection identifiers[6].



Input		Output	
Port	VPI/VCI	Port	VPI/VCI
1	29	2	45
2	45	1	29
1	64	3	29
3	29	1	64

Figure 2.4 ATM switching operation [6].

Two fundamental types of ATM connections determine the manner in which these tables are set up:

- Permanent Virtual Connections (PVC): A PVC is a connection set up by some external mechanism, in which a set of switches between an ATM source and destination ATM system are programmed with the appropriate VPI/VCI values. ATM signalling can facilitate the set up of PVCs, but PVCs always require some manual configuration.
- Switched Virtual Connections (SVC): A SVC is a connection setup automatically through a signalling protocol. SVCs do not require manual interaction. All higher layer protocols operating over ATM primarily use SVCs. Figure 2.5 shows the process of setting up a connection through ATM signalling (SVC).

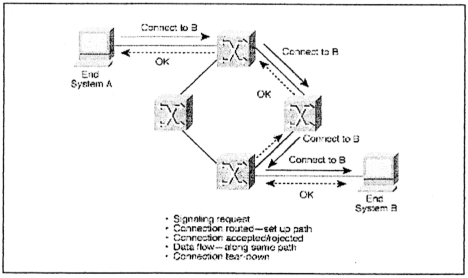


Figure 2.5 Connection setup through ATM signalling (SVC) [6].

2.1.5 ATM Cells

The ATM cell has a fixed length of 53 bytes. Being fixed length allows the information to be transported in a predictable manner. This predictability accommodates different traffic types on the same network. The cell is broken into two main sections:

i.) The header (5 bytes) - the addressing mechanism.

ii.) The payload (48 bytes) - the portion, which carries the actual information.

Header Format

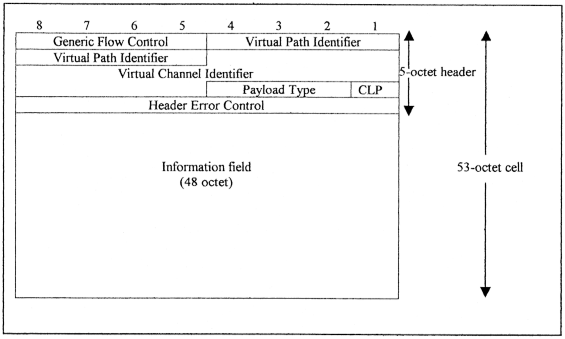


Figure 2.6 ATM cell format – User-Network Interface (UNI) [4].

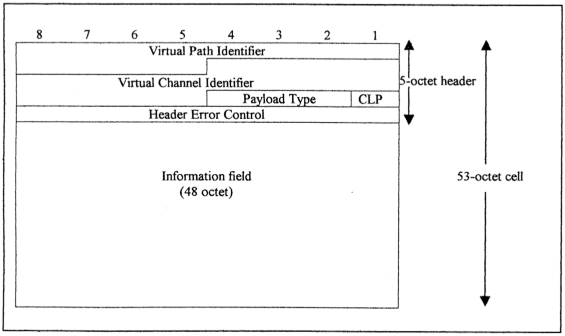


Figure 2.7 ATM cell format – Network-Network Interface (NNI)[4].

Figure 2.6 shows the cell header format of an User-Network Interface (UNI). Figure 2.7 shows the header format of a Network-Network Interface (NNI). For the NNI, the Generic Flow Control (GFC) field, which is performing local functions (to regulate the entry of cells by user into the network or flow control), has been omitted (combined with the VPI) because the control of cell flow is only at the local UNI. The GFC field can be used to assist in controlling the flow of traffic for different service [4]. The short term overloading conditions can alleviate by GFC mechanism. On the other hand the VPI field is expanded from 8 to 12 bits. This allows support for an expanded number of VPCs internal to the network.The virtual path identifier (VPI) (8 bits for UNI and 12 bits for NNI) is for routing, switching and identifying purposes within the network. The virtual channel identifier (VCI) is for routing to and from the end-user. If the first bit of payload type is set to 0, this means the corresponding cell is carrying user data. On the other hand, if the first bit is set to 1 the corresponding cell is carrying management information. If congestion encountered, the second bit will set to 1. The third bit or service data unit (SDU) bit can be used to discriminate two types of ATM SDUs associated with a connection.

The Payload Type (PT) field indicates the type of information. Table 2.2 shows the PT field coding:

Table 2.2 Payload Type (PT) Field Coding [10]

PT Coding	Interpretation
000	User data cell, congestion not experienced, SDU type=0
001	User data cell, congestion not experienced, SDU type=1
010	User data cell, congestion experienced, SDU type=0
011	User data cell, congestion experienced, SDU type=1
100	OAM segment associated cell
101	OAM end-to-end associated cell
110	Resource management cell
111	Reserved for future function

SDU=Service Data Unit, OAM=Operations, Administration, and Maintenance

The Cell Loss Priority (CLP) provides guidance to the network in the condition of congestion. Cells on each link may be discarded during congestion [4]. Bit 0 indicates higher priority and should not be discarded unless there is no other alternative. Bit 1 indicates lower priority and subject to discard within the network if congestion occurred. The network may set the CLP field of any cell to 1 automatically if the agreed traffic parameters is violated.

Header Error Control

An 8-bit header error control (HEC) field is generated by the Physical Layer and included in each ATM cell. This field is calculated based on the remaining 32 bits of the header. The polynomial used to generate the code is X^8+X^2+X+1. However HEC is only for error detection and correction. When a single-bit error is detected, correction will be done automatically [4].

2.1.6 ATM Service Categories

An important networking issue is how application programs determine and indicate their expected bandwidth requirements to the underlying network [7]. Applications may specify parameters such as: peak and average data rate, maximum delay, and cell-loss probability. Then, it is up to the network to allocate its resources to satisfy the application's requests.

ATM network is naturally designed to be able to carry different types of traffic simultaneously, including real-time and non-real-time traffic, voice, video and data. These traffic travel through a virtual channel in a stream of 53-octet cell. There are different ways to handle the different types of traffic. For example, real-time video is delay sensitive and must be delivered within minimum variation in delay.

The ATM forum has defined the service categories as follows:

- Real-time Service
  - ➢ Constant bit rate (CBR)
  - ➢ Real-time variable bit rate (rt-VBR)
- Non-real-time service
  - ➢ Non-real-time variable bit rate (nrt-VBR)
  - ➢ Available bit rate (ABR)
  - ➢ Unspecified bit rate (UBR)

Constant Bit Rate (CBR)

This category is used for emulating circuit switching. This class of service provides fixed data rate that is continuously available through out the whole transferring session and a relatively tight upper bound on transfer delay. Cell loss ratio is specified for CPL=0 cells and may or may not be specified for CPL=1 cells [7]. Uncompressed video and audio data are likely to use this class of service. Examples of CBR application:

- Videoconferencing
- Interactive audio
- Audio or video distribution (multicast/broadcast) and retrieval

Real-time Variable Bit Rate (rt-VBR)

This class of service does not provide a constant or fixed bit rate instead a variable bit rate is provided during a connection session. This category allows users to send data at a variable rate. However, rt-VBR category is intended for time-sensitive application. Bit rate provided by this service may vary over time. Statistical multiplexing is used [7]. Applications using this service can compress the data before transfer. For example, video image frame may compress and vary in size before transfer. This class of service allows the network more flexible than CBR.

Non-real-time Variable Bit Rate (nrt-VBR)

This class of service is suitable for those applications that does not have constraint on delay and delay variation. Bit rate provided by this class of service varies over the whole transmission session. With this class of service, the end system specifies a peak cell rate, a sustainable or average cell rate and burst size. Examples of nrt-VBR application are airline reservations, banking transactions, and process monitoring.

Unspecified Bit Rate (UBR)

This class of service uses the bit rate (capacity) left by the CBR and 2 classes of VBR. UBR takes the advantage of left over capacity because:

- not all resource will be consumed by CBR and VBR, service of UBR can improve the utilisation of network resource, and
- the bursty nature of VBR means that at some time less than the committed capacity is being used.

All this unused capacity is available for UBR service. Applications that can tolerate variable delay and some cell losses are suitable with this service. Such connections are not rejected on the basis of bandwidth shortage and not policed for their usage behaviour. With UBR, ATM cells are forwarded in First In First Out (FIFO) basis. There is no initial negotiation and commitment before the transmission, and no feedback concerning congestion is provided; in other words this is the best-effort service [4]. Some applications using UBR are:

- Text/data/image transfer, distribution and retrieval
- Remote terminal
- Email

Available Bit Rate (ABR)

This class of service is designed for normal data traffic such as file transfer and email. Although, the ABR does not require the cell transfer delay and cell loss ratio to be guaranteed or minimised, it is desirable for switches to minimise the delay and loss as much as possible. Depending upon the congestion state of the network, the source is required to control its rate. The users are allowed to declare a minimum cell rate, which is guaranteed by the VC of the network. Most VCs will ask for a Minimum Cell Rate (MCR) of zero. Those with higher MCR may be denied the connection if sufficient bandwidth is not available [7].

ABR is defined as an alternative and to improve the service for application with bursty traffic. Applications that wish to use ABR service must specify a peak cell rate (PCR) and minimum cell rate (MCR). If the permission is granted then the applications will receive at least the MCR that they requested.

Figure 2.8 shows how the whole capacity is shared among all the classes of service during a steady state period of time (no additions or deletions of virtual channels).

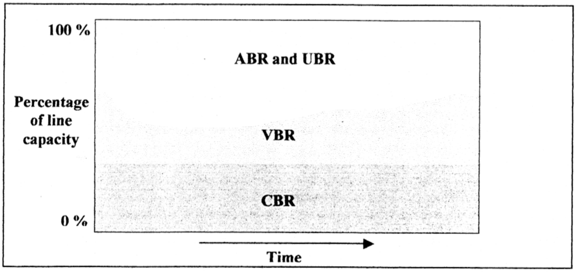


Figure 2.8 ATM bit rate services [4].

2.2 Quality of Service (QoS) and Traffic Attributes

Some of the traffic parameters [8] are as follows:

- Peak Cell Rate (PCR): The maximum rate that an application can transmit. PCR is the inverse of the minimum inter-cell interval.
- Sustained Cell Rate (SCR): This is an average rate as measured over a long time interval.
- Cell Loss Ratio (CLR): The percentage of cells those are lost in the network. This maybe caused by error or congestion. During congestion, cell with CLP bit set will be dropped by network.
- Cell Transfer Delay (CTD): The delay experienced by a cell between network entry and exit points is called the Cell Transfer Delay. This includes propagation delays, queuing delays at intermediate switches, and service times at queuing points.
- Cell Delay Variation (CDV): This is a measure of variance of CTD. High variation needs larger buffer for delay sensitive traffic such as voice and video.
- Cell Delay Variation Tolerance (CDVT): For source transmitting at a certain rate, a slightly variation in inter cell time is allowed. A leaky bucket type of algorithm called "Generalised Cell Rate Algorithm (GCRA)" is used to determine if the variation in inter cell times is acceptable.
- Maximum Burst Size (MBS): The maximum number of back-to-back cells that can be sent at PCR without violating the SCR.

PCR, SCR, CDVT, and MBS are the input traffic characteristics and they are enforced by the network at the network entry. CLR, CTD, and CDV are qualities of service provided by the network and they are measured at the network exit point [7].

2.3 Connection Acceptance Control and Usage Parameter Control

During the call setup phase (or the call renegotiation phase), the Connection Admission Control (CAC) decides to accept a new connection on a link and consequently, to allocate a certain portion of bandwidth to it, if the required QoS can be guaranteed for both the connections already established and the new one [44-46]. CAC will reject a request if the connection is expected to degrade the QoS of the existing connections. As the CAC connections rely on the negotiated parameter values, it can only perform correctly if all bandwidth contracts are respected. The more parameters that can be negotiated, the more exact the representation of the behaviour of the service will be and the more accurate the police function can react. Usage Parameter Control (UPC) is required in order to ensure that each source conforms to its negotiated parameters. The UPC function can be defined as the set of actions taken by the network, during the entire phase of the call, to monitor and control the offered traffic, with the purpose of protecting network resources from malicious and unintentional misbehaviour. CAC takes an appropriate action after violation of negotiated parameters are detected. These actions can either be:

- Cell-dropping,
- Cell-marking, or
- Shaping the source rate.

2.4 Congestion Control Techniques

Congestion happens whenever the input rate is more than the available link capacity [7]:

$$\Sigma \text{ Input Rate} > \text{Available link capacity}$$

Congestion control is the heart of the general problem of traffic management for ATM network. The main mission of congestion control is to ensure good throughput and delay performance while maintaining fair allocation of network resources to the users [8].

The traffic patterns of ABR service class, are naturally highly bursty and unpredictable. As described in [11], one way to classify congestion control schemes is based on the layer of International Standardisation Organisation/Open Systems Interconnectivity (ISO/OSI) reference model. For example, there are data link, routing, and transport layer congestion control schemes. Typically, a combination of such schemes is used. The selection depends on the severity and duration of congestion. Figure 2.9 shows how the duration of congestion affects the choice of the method. The best solution is to install higher speed links and redesign the topology.

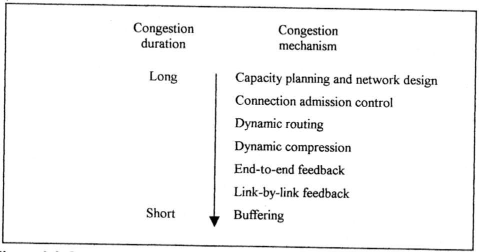


Figure 2.9 Congestion technique for various congestion duration [7][8].

The most popular method that can be used to avoid the network congestion is to accept a new ATM connection only when sufficient resources are available. This is due to the fact that once the connection is admitted, the congestion may persist for the duration of the connection, so CAC is effective only for medium duration congestion. An end-to-end control scheme can be used when the congestion lasting is less than the duration of connection. For the shortest congestion duration, a sufficient size buffer in the switches is the best solution.

2.4.1 Feedback Facilities

UNI V3.0 specified two different facilities for the feedback control [7]:

1. Generalised Flow Control (GFC): GFC field at the UNI header will be used by the network to control the source flow.
2. Explicit Forward Congestion Indication (EFCI): Payload type field in the cell header is used to convey congestion information in a binary (congestion or no congestion) manner. The destination can monitor the EFCI bits and request sources to increase or decrease their rate.

2.4.2 Selection Criteria

A number of congestion schemes were presented. Thus, in order to select the rigt congestion scheme, the following criteria must be satisfied:

1. Scalability: Basically this ensures that the same scheme can be applied for LAN as well as WAN.

2. Optimality: In sharing link capacity environment, a certain source depends upon the demands by other source. The most popular criterion is "Max-Min Allocation" [12]. This method provides the maximum possible bandwidth to the source receiving the least among all contending sources.
3. Fairness Index: Suppose a scheme allocates $\{x_1, x_2, \ldots x_n\}$ instead of the optimal allocation $\{y_1, y_2, \ldots y_n\}$, then the normalised allocations $z_i = x_i/y_i$ for each source can be calculated. Finally, compute the fairness index as follows [13][14]:

$$\text{Fairness} = (\Sigma_i x_i)^2/(n\Sigma_j x_j^2)$$

4. Robustness: Slight mistuning of parameters or loss of control messages should not bring the network down.
5. Implementability: The scheme should not dictate particular switch architecture.
6. Simulation Configuration: A number of network configuration were also agreed upon to compare various proposals.
7. Traffic Patterns: Various traffic patterns are proposed. The following three are the most common:
   - Persistent Sources – Greedy, always have cells to send, network always congested
   - Staggered Sources – The source start at different times.
   - Bursty Sources – These sources oscillate between active state and idle state. During active state, they generate a burst of cells [16]. This is a more realistic source model. If the total load on the link is less than 100%, then throughput and fairness are not an issue.

2.5 Neural Network

The expression Neural Networks come from the similarities with the neurons in the human brain. The human brain has many advantages that are highly attractive such as:

- Robustness. Nerve cells can die without affecting the performance.
- Flexibility. The brain can adapt to new environments easily.
- Fuzziness. The brain can deal with fuzzy decisions for fuzzy inputs.
- Recognition. The brain has excellent pattern recognition and processing.
- Parallel. Calculations in neurons are executed simultaneously.

- Compact. The brain is small and consumes very little power.

Artificial neural systems consist of numerous, simple processing units or "neurons". We can program or train Neural Networks to store, recognise, and associatively retrieve patterns or database entries; to solve optimisation problems; to filter unwanted noise from measurement data; and to control ill-defined problems. The human nervous system can be viewed as a three-stage system, as shown on Figure 2.10 [33].

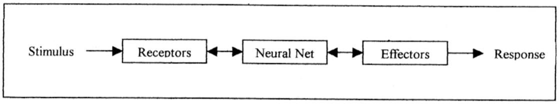


Figure 2.10 Human nervous system [33].

2.5.1 The Biological Neuron

The human brain contains roughly 100 billion neurons. The brain represents an asynchronous, non-linear, massively parallel, and feedback dynamical system. Many feedback Neural Networks can learn new patterns and recall old patterns simultaneously [32]. Neural Networks store pattern or function information with distributed manner. A very simplified schematic view of a biological neuron is shown in Figure 2.11 [34].

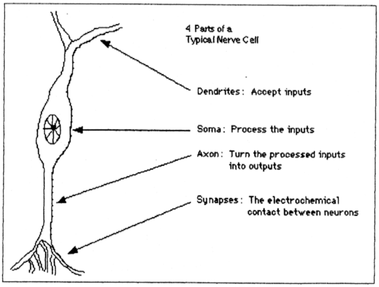

4 Parts of a Typical Nerve Cell
Dendrites: Accept inputs
Soma: Process the inputs
Axon: Turn the processed inputs into outputs
Synapses: The electrochemical contact between neurons

Figure 2.11 Biological neuron [34].

1. Soma or body cell is the large, round central body in which almost all the logical functions of the neuron are realised.

2. The axon (outputs) is the nerve fibre attached to the soma and can serve as the final output channel of the neuron.
3. The dendrites (inputs) represent a highly branching tree of fibres and attached to soma. Dendrites connect the neuron to a set of other neurons. Dendrites either receive inputs from other neurons via specialised contacts called synapses or connect other dendrites to the synaptic outputs.
4. Synapses are specialised contacts on a neuron, which are the termination points for axons from other neurons.

According to the simplified model of the neuron, the cell body (soma) receives input from other neurons through adjustable or adaptive synaptic connections to the dendrites. The output signal (consisting of nerve impulses) from cell is transmitted along a branching axon to the synapses of other neurons. When a neuron is excited it produces nerve impulses, which are transmitted along an axon to the synaptic connections of the other neurons. The output pulse rate depends on both the strength of the input signals and the weight or strength of the corresponding synaptic connections [34].

2.5.2 Model of Artificial Neuron

A neuron is an information-processing unit that is fundamental to the operation of a Neural Network. As in nature, the network function is determined by the connections between elements (neurons). We can train a Neural Network to perform a particular function by adjusting the values of the connections (weights) between elements (neurons). Here are three basic elements of the neural model:

1. A set of synapses or connecting links, each of which is characterised by a weight or strength of its own. A signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}.
2. An adder for summing the input signals, weighted by the respective synapses of the neuron.
3. A transfer function is use to limit the amplitude of the output of the neuron.

Commonly Neural Networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in Figure 2.12.

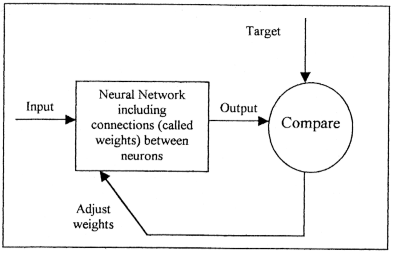

Target
Input
Neural Network including connections (called weights) between neurons
Output
Compare
Adjust weights

Figure 2.12 Training with input and target [43].

The Neural Network is adjusted, based on a comparison of the output and the target, until the output matches the target. In supervised learning, many such input/target pairs are used to train a network. Neural Networks have been trained to perform complex functions in various fields of applications. Neural Networks can be trained to solve problems that are difficult for conventional computers or human beings. The supervised training methods are commonly used. Unsupervised networks can be used, for instance, to identify groups of data.

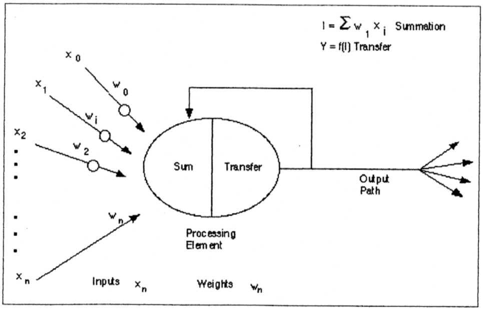


Figure 2.13 Model artificial neuron [33].

Figure 2.13 shows the model of a neuron, which forms the basis for Neural Networks. The artificial neuron model in Figure 2.13 also includes an externally applied bias, denoted by b_k. The bias b_k has the effect of increasing or decreasing the net input of the transfer function.

In mathematical terms, we can describe a neuron *k* by writing the following pair of equations [33]:

$$U_k = \sum_{j=1}^{m} w_{kj} x_j$$

and

$$y_k = \varphi(u_k + b_k)$$

where $x_1, x_2, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{km}$ are the synaptic weights of neuron k; u_k is the linear combiner output; b_k is the bias; $\varphi(\ )$ is the transfer function; and y_k is the output signal of the neuron.

The use of bias b_k has the effect:

$$v_k = u_k + b_k$$

2.5.3 Types of Transfer Function

1. Figure 2.14 shows the Threshold Function. For this type of transfer function, for example we have the following example:

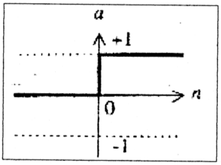


Figure 2.14 Threshold function [33].

$$\varphi(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{if } n < 0 \end{cases}$$

2. Figure 2.15 shows the Linear Function. For this type of transfer function, we have the following example:

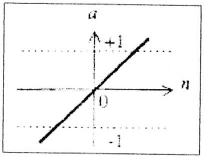


Figure 2.15 Linear function [33].

$$\varphi(n) = n$$

3. Log-Sigmoid Function. The log-sigmoid function, whose graph is s-shaped (as shown in Figure 2.16), is by far the most common form of transfer function. An example of log-sigmoid function is:

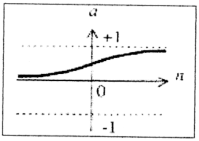


Figure 2.16 Log-Sigmoid function[33].

$$\varphi(v) = \frac{1}{1+\exp(v)}$$

2.5.4 Types of Network Architectures

In general, four fundamental different classes of network architecture are defined.

1. **Single-Layer Feed-forward Networks**. This is the simplest form of layered network; an input layer of source nodes that projects onto an output layer of neurons, as illustrated in Figure 2.17. None of the neurons are connected to each other.
2. **Multi-layer Feed-forward Networks**. This class of network distinguishes itself by the presence of one or more hidden layers. The function of hidden neurons is to intervene between the external input and the network output, as illustrated in Figure 2.18. The output values of one layer are weighted and fed as inputs to the next layer of neurons. Only neurons in adjacent layers are connected.

3. **Back-Propagation Neural Networks**. This class of network distinguishes itself from others in that it has at least one feedback loop. Information for updating the weights is fed backwards, as shown in Figure 2.19.
4. **Hopfield Neural Networks**. As shown in Figure 2.20, all neurons are connected to each other. The inputs to a neuron are the weighted output from all the neurons and an input value.

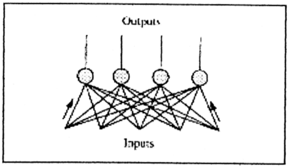


Figure 2.17 Single layer feed-forward [42].

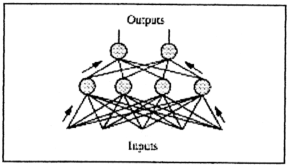


Figure 2.18 Multiple layer feed-forward [42].

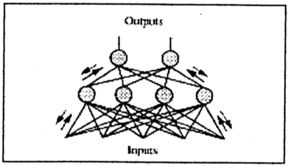


Figure 2.19 Back-propagation [42].

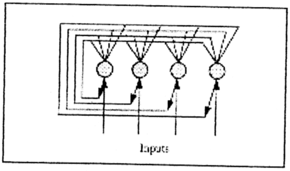


Figure 2.20 Hopfield [42].

2.5.5 Prediction Using Neural Network

In many scientific, economic and engineering application there arise the problem of predicting (forecasting) the future on the basis of some collected historical data. The prediction of future sample values of a time series can be done by extracting knowledge from its past values [35-41]. Predicting the future in real-time (updating prediction on the fly) has potential applications in such fields as adaptive control and system modelling. In fact the problem of predicting the future is extremely general.

The most powerful approach to the problem of prediction is to find a law underlying the given phenomenon [35]. If such a law can be discovered and analytically described by a set of differential equations, then by solving the equations we can predict the future. Unfortunately, he information about a dynamic process under investigation is often only partial and ncomplete; so the prediction cannot be based on a known analytical model.

n the second less powerful approach, an attempt is made to discover some strong empirical egularity in the observation of the time series. A time series consisting of samples of a eriodic process that can be modelled by the superimposition of sinusoids generated by a set f second-order differential equations. Unfortunately, in many real world problems, some egularity such as periodicity are masked by noise and even some dynamic processes phenomena) are described by chaotic time series in which the data seem random without pparent periodicities [36][37]. Since the error (uncertainty of the prediction) increases xponentially in time, a long-term prediction is not possible for chaotic time series. Though haos precludes any long-term predictability, a short time prediction is possible and very omising results have been obtained by using Neural Network approach [36][37].

Ct.

2.6 Fuzzy Logic

Zadeh, in his seminal 1965 [48], completed the task by following through the speculation of previous logicians and showing his "fuzzy set". The theory of Fuzzy Logic deals with two problems [83]:

1. The fuzzy set theory, which deals with ambiguity found in semantics.
2. The fuzzy measurement theory, which deals with the ambiguous nature of judgements and evaluations.

The primary motivation of Fuzzy Logic is the possibility of exploiting tolerance for some inexactness and imprecision. Precision is often very costly. Fuzzy Logic and classical logic differ in the sense that the former can handle both symbolic and numerical manipulation; while the latter can handle symbolic manipulation only (as in Figure 2.21). In Fuzzy Logic, exact (crisp) reasoning is considered to be the limiting case of approximate reasoning; one can see that everything is a matter of degree.

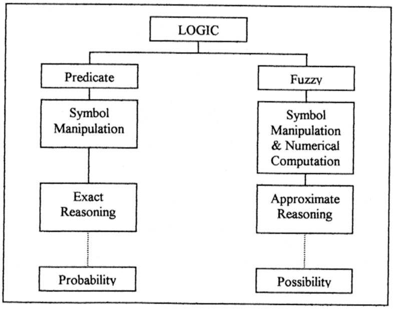


Figure 2.21 A comparison of predicate and Fuzzy Logic [83].

"Fuzzification" translates the crisp knowledge, to a linguistic or fuzzy knowledge. Fuzzy sets may be represented by a mathematical formulation (often known as the membership function). This function gives a degree or grade of membership within the set. The crisp sets only allow full membership or no membership at all, but the fuzzy sets allow partial membership. In other words, an element may be partially belong to a set. A fuzzy set in a universe of discourse U is characterised by a membership function m_f, which assumes values in the interval [0,1]. A fuzzy set F is represented as a set of ordered pairs, each made up of a generic element $u \in U$ and its degree of membership $m_f(u)$.

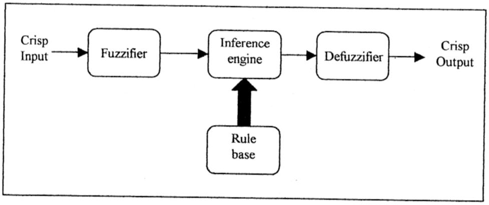


Figure 2.22 Block diagram of a typical fuzzy controller [49].

A linguistic variable x in a universe of discourse U is characterised by a set $W(x)$ = $(W_{1x},\ldots,W_{nx})$ and a set $M(x) = (M_{1x},\ldots,M_{nx})$, where $W(x)$ is the term-set, the set of names which the linguistic variable x can assume, and W_{ix} is a fuzzy set whose membership function is M_{ix}. For example, if x indicates a temperature, $W(x)$ could be the set $W(x)$ = (Low, Medium, High), each element of which is associated with a membership function.

The rules governing Fuzzy Logic are often written using linguistic expressions. If x and y are taken to be two linguistic variables, Fuzzy Logic allows these variables to be related by means of fuzzy rules of the following type:

"IF (x is A) THEN (y is B)"

A fuzzy controller takes the form of a set of IF-THEN rules whose antecedents (IF part) and consequents (THEN part) are themselves membership function. The degree of membership function of the antecedents is calculated. Figure 2.22 shows the block diagram of a typical fuzzy controller. A fuzzy controller used crisp data and through the process of fuzzification, these data are changed to linguistic or fuzzy membership functions (fuzzified). Through application of a Fuzzy Logic inference method (typically MAX-DOT or MAX-MIN), it allows the output y to be determined. The fuzzy output will then be changed back into crisp values through a process called "defuzzification". Consequents from different rules are numerically combined and then collapsed to yield a single real-number (binary) output [44].

In general, the antecedents is made up of a statement in which fuzzy predicates P_j of the general form (X_j is A_j) are combined by different operators such as union, intersection, complement, OR, AND, etc. For example:

IF P1 AND P2 OR P3 THEN P4

where,

$$P1 = (X_1 \text{ is } A_1);\ P2 = (X_2 \text{ is } A_2);$$
$$P3 = (X_3 \text{ is } A_3);\ P4 = (X_4 \text{ is } A_4).$$

To apply an inference method to the conclusion, it is first necessary to assess the degree of membership, θ, of the premise, through assessment of the degrees of membership α_i of each predicate P_i in the antecedents [44]. The membership degree α_i is calculated by assessing the degree of membership of X_i in the fuzzy set Ai. If X_i is made up of a fuzzy set, its degree of membership α_i is determined by making an intersection between the fuzzy value of X_i and the fuzzy A_i.

The degree of membership, θ, of the antecedents can thus be calculated by assessing the fuzzy operations on the predicates. The Fuzzy Logic operator AND when applied to two predicates P_i and P_j, returns the minimum of (α_i, α_j); the Fuzzy Logic OR, when applied to the fuzzy predicates, returns the maximum of (α_i, α_j).

Once the value of θ is known, an inference method can be applied to assess the consequence. The latter is expressed in the form

$$(Y_1 \text{ is } B_1) \text{ AND } (Y_2 \text{ is } B_2) \text{ AND } (Y_3 \text{ is } B_3)$$

where Y_i are linguistic variables and B_i are names belonging to the term set $W(Y_i)$.

In this case, the fuzzy operators AND acting on the output fuzzy predicates have a different meaning than in the antecedents. They only express in a single rule with the same antecedents but different consequents. If an output variable is present in more than one rule, the fuzzy values of the output variable can be obtained in the single fuzzy set through a union operation.

To obtain the final output membership function, it is possible to obtain a crisp value by adopting one of several defuzzification techniques, for example the centre of gravity method seems to give the best results [44].

2.7 ATM Traffic Prediction using Neural Network

In [49], they classified the use of Neural Network (NN) techniques in ATM control in to four general categories:

1. Neural Network based admission control
2. Neural Network based congestion control and policing
3. Neural Network based traffic characterisation and prediction
4. Neural Network based switch control

In the prediction category, the properties of Neural Network are used to either predict incoming traffic or combine the prediction of expected performance of the system with a congestion notification. For such problem Feed-Forward Neural Networks are used. The proposed networks differ in their number of layers, size, training technique used, and input/output representation.

Traffic prediction is an inherent property of Neural Networks. Neural Networks used in congestion control need first to predict the rate of arrival so that they can suggest optimum control actions.

In [50] a three-layer Feed-Forward Neural Network (FFNN) with back propagation learning algorithm is used for traffic prediction. It is argued that the use of Auto Regressive Integrated Moving Average (ARIMA) models is not sufficient for prediction of non-linear and chaotic types of traffic such as encountered in broadband networks. Good results in terms of Mean Square Error are reported in the second-order properties. Over-training seems to be a problem when the noise signal is very strong.

In [51], Neural Networks are used for prediction of load based on queue status and estimated arrival rate number of estimated assigned slots. A Doubly Stochastic Poisson Process arrival process is assumed. For such traffic it is shown that a single neuron is a good enough estimator.

2.8 Dynamic Bandwidth Allocation

Dynamic Bandwidth Allocation means that the bandwidth allocated to a logical connection (same end points) is changed from time to time, in order to be adapted to the traffic variations and to obtain better resource utilisation [52]. This subsection discusses few proposed and developed dynamic bandwidth allocation technologies.

2.8.1 Dynamic Time Slice (DTS) Scheme

Sriram developed this scheme in 1992.

Traffic Classification In Broadband/High-Speed Networks

In [52], they classified traffic (which are likely to be integrated in broadband ATM networks) into 4 categories:

- Type 1A: Delay-sensitive Isochronous High Bandwidth Services
  - ➢ It is isochronous because it requires a fixed bandwidth for the duration of a call.
  - ➢ It requires real-time service.
  - ➢ Known as Real-Time High Bandwidth Service (RT-HBS).
  - ➢ Example of this traffic is CBR conference video.
- Type 1B: Delay-sensitive Non-Isochronous High Bandwidth Services
  - ➢ This traffic is real-time and non-isochronous.
  - ➢ Each call alternate between active and inactive periods.
  - ➢ Also known as RT-HBS.
  - ➢ Example of this type of traffic is VBR video and LAN-to-LAN interconnects.
- Type 2: Delay-insensitive High Bandwidth Services
  - ➢ Non-real-time bulk information transport services.

- The user must be specified are the extent of delay that is acceptable.
- Examples of this traffic type are: document/image/video delivery services (non-real-time, sometimes overnight delivery).

- Type 3: Low-bandwidth Statistically Multiplexed Services (SMS)
  - Delay sensitive with end-to-end delay requirements ranging from a few tens of milliseconds to a few hundreds of milliseconds.
  - Network will reserve a certain bandwidth for this type of service.

Bandwidth Allocation and Congestion Avoidance in B-ISDN

The DTS scheme can be explained with the help of Figure 2.23. Each Type 1A, Type 1B, Type 2 and Type 3 are assigned with separate queues. The DTS server shown in Figure 2.23, services all the queues by cyclically visiting each queue and allocating a slice of time to it. The time slice allocated to a queue would be proportional to the bandwidth required by that queue. The maximum number of cells that can be transmitted in a DTS (M_c) cycle must be large enough so as to facilitate a choice of a wide range of fractions of link bandwidth. However, M_c should be chosen so that the cycle length remains in the order of a few milliseconds. This will guarantee that the maximum delay for the cells of Type 1A calls will be limited to a millisecond or two.

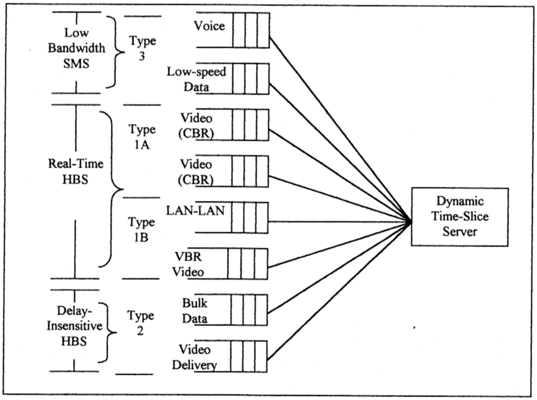


Figure 2.23 An illustration of operation of the DTS scheme [52].

In general, n changes as the number of traffic classes that offer calls varies. Also, n varies as the number of Type 1A and Type 2 calls vary. As n varies, the time-slice allocation $T1$, $T2$, ..., Tn for the n queues, are also re-assigned value to reflect the proportion bandwidth requirements for all traffic classes or HBS calls in service. The units of the time-slice are in terms of the number of ATM cells.

Computations of DTS Time-Slice Parameters

Suppose that $n+1$ queues are to be set up with bandwidth requirements in the fractions of $f0$, $f1$, ..., fn of the link bandwidth. The DTS time parameters ($T1$, $T2$, ..., Tn) are chosen such that the following relationships hold:

$$f_i = \frac{T_i}{\sum_{i=0}^{n} T_i}, \quad o \leq i \leq n, \tag{1}$$

$$\sum_{i=1}^{n} f_i \leq 1 - f_o, \tag{2}$$

$$\sum_{i=0}^{n} T_i \leq D_t, \tag{3}$$

where $D_t = M_t\tau$ is the DTS service cycle time, M_t is the number of cells in a DTS service cycle , and τ is the cell transmission time on the link. Equation (1) ensures that a fraction f_i of the link bandwidth is guaranteed for queue i. Equation (2) indicates that the total allocated bandwidth to all traffic classes cannot exceed a fraction ($1 - f_o$) of the link capacity. Fraction f_o is reserved for the signalling traffic. Equation (3) indicates that the DTS service cycle time should be about one to two millisecond.

2.8.2 Virtual Bandwidth Scheme

Tsai, Wand and Chang proposed this scheme in 1993. They proposed a mathematical model for the analysis of their dynamic bandwidth allocation scheme. By using the analytical formulas, the bandwidth utilisation and management cost (measured in normalised bandwidth and update frequency) of two experiments are obtained [53]. ATM allows cell stream with various characteristics multiplex into the transmission links. ATM network employs the concept of Virtual Path and Virtual Channel and allows bandwidth management at both levels. For example, in [54-56], a dynamic scheme for VP bandwidth control, has been proposed for single-link and single-traffic-type system.

Dynamic Bandwidth Allocation Scheme (Virtual Bandwidth)

To accept an incoming call, the residual capacity of each transmission link on the designated route needs to be larger or equal to the requested bandwidth. It is more complicated in ATM network, which has the mixture of heterogeneous traffic. The CAC turn out to be dependent on the traffic parameters, priority control mechanisms, and bandwidth assignment algorithms. The virtual bandwidth of a call is defined to be the link capacity divided by the maximum number of acceptable calls, provided only this type of calls is accepted.

For connection acceptance control, the number of calls of a stream, say stream-i, denoted as x_i, needs to be monitored. An additional parameter x'_i represents the smallest integer multiple of s which is greater than or equal to x_i, where s is an integer called bandwidth allocation unit. Thus, the bandwidth allocated for the stream is then implicitly equal to x'_i times the virtual bandwidth of that type of call.

- If a call arrives when $x_i = x'_i$, implying that the allocated bandwidth is completely utilised by connected calls, then the bandwidth needs to be re-allocated and x'_i shall be increased by s.
- If a call terminates such that x_i turns to be equal to $x'_i - s$, then an amount of bandwidth shall be released and x'_i is decreased by s.

By keeping the two records x'_i and x_i, a connection establishment procedure combining the dynamic bandwidth control and the connection acceptance control can be efficiently realised.

For example, they implement the dynamic bandwidth allocation scheme on the VP-level bandwidth control. The VC connection request could be processed as follows:

- If $x_i < x'_i$ then QoS is guaranteed and the VC is established without further processing.
- If $x_i = x'_i$ the network tries to add bandwidth to this VP by increasing x'_i to $x'_i + s$. In allocating bandwidth, the network simply checks whether the state vector composed of x'_i is feasible.

Intuitively, one can predict that usage of the network resources is less efficient than call-by-call allocation strategy, because bandwidth may be allocated but is not immediately used.

2.8.3 Minimum Overflow Traffic Algorithm (MOTA)

This scheme was proposed by Han Zhou, Chang and Han in 1995. In [57], this scheme was proposed to assign the bandwidth b_i for each traffic class in the hierarchical admission control structure. An overflow traffic function Q is used to adjust the bandwidth assignment each time a new connection is required. They used virtual cell loss probability (VCLP) as the measurement of performance. The main advantage of this algorithm are their minimum values in direct ratio and the bandwidth allocation that can be adjusted based on a single call request, thus it can use bandwidth fairly and efficiently.

Structure of the Bandwidth Allocation and Admission Control System

Each traffic source is characterised by its peak rate λ_i bits/second, average active period of $1/\alpha_i$ second and average silent period of $1/\beta_i$ second. The traffic density of a traffic source in class i is denoted by r_i and defined as

It is the average time fraction for an active traffic source.

$$\sum_{i=1}^{M} b_i \leq C$$

When a new connection is required, bandwidth allocations are recomputed on-line by an allocation controller, whose goals reflect overall cell loss and refused traffic. The multiplexer is assumed to operate synchronously. Each admission controller i implements a decision rule for the acceptance of incoming calls. The bandwidth allocation controller "sees" the traffic density and peak rate of each class. It divides the total capacity C of ATM channel (in Mbit/s) into virtual link capacities b_i, i=1, 2, …., M. The assignment is made on the basis of the dynamic variations in the traffic flows and the assignment should be fair to all the admission controllers. The acceptance algorithm is based on individual virtual cell loss probability requirement.

Virtual Cell Lost Probability

The individual virtual cell loss probability for class i denoted by VCLP(i) is based on the link overflow model in which cells are discarded when instantaneous total traffic load R_i in class i exceeds the link capacity b_i assigned to class i. It is the ratio of the expected value of the excess loads X_i, $X_i = E[r_i - b_i]$, and the average load of the class i, S_i. So,

$$\mathrm{VCLP}(i) = X_i/S_i$$

where

$$X_i = \sum_{r_i}^{L_i} p_i(r_i)(R_i - b_i)$$

$$R_i = r_i * \lambda_i$$

L_i is the number of connections in class i, i =1,2, .., M and $p_i(r_i)$ is the probability r_i out of L_i calls are active.

Overflow Traffic Function (*Q*)

MOTA defines an overflow traffic function Q and uses Q as the bandwidth assignment optimisation object function. The VCLP due to the lack of assigned bandwidth ($b_i < R_i$) will trend to its minimum value with optimising the function Q. The overflow traffic function is expressed in:

$$Q\min = \sum_{i=1}^{M} \sum_{n_i \in (R_i - b_i > 0)}^{n_i = L_i} P_i(n_i)\left(\frac{\sigma_i R_i - b_i}{b_i}\right)$$

It measures the mean blocked peak traffic to bandwidth ratio over the ATM link for all the traffic classes in the hierarchical admission control structure, where $\sigma_i > 0$. The problem to be solved in bandwidth allocation controller is to optimise the function Q as below:

$$\mathrm{VCLP}(i) < \mathrm{VCLP}_{\max}(i),\ i = 1, 2, \ldots, M$$

Where VCLP(i) is the individual virtual cell loss probability in class i under current conditions, while $\mathrm{VCLP}_{\max}(i)$ is the maximum value of virtual cell loss probability in class i.

Bandwidth Assignment Algorithm

This algorithm can be described as:

$$m_i \le b_i \le L_i \lambda_i$$

$$Q\min = \sum_{i=1}^{M} \sum_{n_i \in (R_i - b_i > 0)}^{n_i = L_i} P_i(n_i)\left(\frac{\sigma_i R_i - b_i}{b_i}\right) \quad \text{and} \quad \sum_{i=1}^{M} b_i \le C$$

where m_i is the minimum capacity, which is needed to support the active connections by guaranteeing their performance requirement. The requirement can be defined as

$$\mathrm{Prob}(\text{loading of class } i > G_i) < \tau$$

Where G_i is the maximum permitted load and τ as a given threshold. According to [58], they assume that channel loading arises from a large number of sources multiplexed so that the loading of the channel can be approximated as a Gaussian process and τ is 10^{-8}, then m_i can be described as:

$$m_i = \frac{5.243(\lambda_i\sqrt{L_i r_i(1-r_i)} + L_i\lambda_i r_i}{G_i}$$

2.8.4 Adaptive Hidden Markov Model (HMM) Prediction

Several bandwidth allocation policies for connection oriented guaranteed bandwidth networks have been introduced in [59]. These policies can be decomposed into three groups:

1. Static Algorithms in which the bandwidth allocation does not change.
2. Periodic Algorithms in which the bandwidth allocation is adjusted periodically, in fixed equal size time interval.
3. Adaptive Algorithms in which the bandwidth allocation maybe changed whenever is necessary, as long as the changes are not too frequent.

A change in the VC bandwidth allocation is made by either closing the existing VC (or opening a new one with the new allocation) or by changing the allocation of the VC without closing it. Both methods involve some overhead, thus these adjustments should not be made too often. However, avoiding changes in a VC bandwidth allocation may cause either long queues and high latency, or low network utilisation.

The periodic and adaptive algorithm modifies the bandwidth allocation dynamically. The main problem is in determining the new bandwidth allocation. If the future traffic is unknown and packet arrive in an on-line manner, then it is important that when a change is made, the new bandwidth allocation matches the future traffic as best as possible. To this end, forecasting methods are developed that predict the session's arrivals rate in the near future.

Saran and Keshav [60] have introduced and studied the question of VC holding cost (once a VC established, how long should the circuit open while no packets are arriving).

Model and Problem Statement

The smallest time units that they considered in [59] and which were in their simulations are called slots. Thus time is measured in slots (in the simulations a slot was 10^{-3} seconds). The term session refers to the time interval in which the session is alive; that is, from time in which the first packets of the session arrives to the adaptation layer until the time that the last packets leaves the adaptation layer. To change the bandwidth at the beginning of each slot according to the number of packets that are expected to arrive during the time slot is impractical.

They assumed that a change in a VC bandwidth allocation is possible during each period of r time slots. Session is divided into k periods where the k period, $k = 1, 2, \ldots, n$. At the end of each period (of r time slots) the algorithm is given the arrival times of all the packets in the previous periods. The output of the algorithm at this point of time is the requested bandwidth allocation for the session to be used in the next period. They use the following notations:

- n – the number of periods in a session (1 to n)
- r – the number of time slots in a period
- p – a parameter that represents the "desired utilisation" (ranges in the interval [0,1])
- $T_i(t)$ – the number of packets arriving in time slot t of period i
- T_i – the number of packets arriving in period i
- λ_i – the number of packets per time slot in period i, T_i/r
- λ - the number of packets per time slot over the entire session, $\sum_{i=1}^{n} \lambda_i / n$
- μ - the bandwidth allocation for the entire session
- μ_i – the bandwidth allocation in period i
- $\hat{p}$ - the average session utilisation, $\frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n} r\mu_i}, \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{n} p_i}$
- q_i – the queue length at the beginning of period i

The Algorithm

There are three categories of algorithm:

1. Static algorithm – opens a VC and fixes its bandwidth once at the beginning of a session and does not modify the allocation during the session.

2. Periodic algorithm – adjusts the bandwidth allocation of a session at the beginning of every period (r time slots). The output of the algorithm is the requested bandwidth for each period. The periodic algorithms select the bandwidth allocation μ_i for the duration of period i at the beginning of each period i. The various periodic algorithms compute μ_i as follows:
   - prev - $\mu_i = (\lambda_{i-1})/p$ for i=1,2, …., n and $\lambda_0 = 0$; that is, to allocate the next period with the amount of the bandwidth, it is necessary to transmit the amount of traffic arrived in the previous period with the desired utilisation.
   - queue - $\mu_i = (q_i/r)/p$; that is, to allocate the next period with the amount of bandwidth, it is necessary to transmit the current content of the queue with the desired utilisation.
   - queue + prev - $\mu_i = ((q_i/r) + (\lambda_{i-1}))/p$ for i=1,2, …., n and $\lambda_0 = 0$; that is, to allocate the next period with the amount of bandwidth, it is necessary to transmit the current content of the queue plus the same amount of traffic arrived in the previous period with the desired utilisation.
   - queue + next - $\mu_i = ((q_i/r) + (\lambda_i))/p$, $\lambda_0 = 0$; that is, to allocate the next period with the amount of bandwidth, it is necessary to transmit the current content of the queue plus the amount of packets will arrive in the next period with the desired utilisation.
3. Adaptive algorithm – which is more flexible than the periodic algorithms. The bandwidth of a session may be adjusted at arbitrary times.

Hidden Markov Model (HMM)

In a HMM, each state of a random variable that measures the number of packets generated in a time slot, is distributed according to the next state (idle, active, and burst) given the current state. A HMM is a doubly stochastic process with an underlying hidden stochastic process. The states of the underlying stochastic process behaviour cannot be observed directly. To predict the number of packets arriving during the next period, HMM learns a state machine that describes the behaviour of the network in the recent past using the learning algorithm of [61].

Let the random variable $R(s)$ denotes the number of packets generated during a period, given that start at state s. The expected value of $R(s)$ for different states s should be significantly different. In addition, the variance of $R(s)$ should be small.

Adaptive Algorithms

In the periodic algorithm, the bandwidth allocation at the end of each period will change. Simple adaptive algorithm will try to avoid some of the changes, in the case that the difference between the new and old bandwidth allocation is insignificant. The idea is to save unnecessary bandwidth modifications without affecting the performance of the algorithm.

The simple adaptive algorithm has a parameter,

$$0 < CHANGE \leq 1$$

As in the periodic algorithms, in the beginning of period i calculate the desired μ_i denoted by μ_{inew}. Now μ_i is determined as follows:

$$\text{IF } (CHANGE \leq \mu_i - 1 / \mu_{inew} \leq 1 / CHANGE) \text{ THEN}$$

$$\mu_i = \mu_{i-1}$$

ELSE

$$\mu_i = \mu_{inew}$$

For CHANGE =1 this simple adaptive algorithm behaves exactly as the periodic algorithm. For CHANGE <1, the bandwidth allocation will remain unchanged, if the new bandwidth allocation is not much different with the old bandwidth allocation. As expected for smaller CHANGE, the number of bandwidth allocation changes becomes smaller and decreases the bandwidth utilisation. There is a trade-off between saving in the bandwidth modification and increasing the average queue length.

Adaptive HMM (AHMM)

HMM is used to predict the traffic in the next period, but occasionally the HMM changes its state much sooner than expected. AHMM learns the HMM in the same way HMM does. However, changing the bandwidth allocation every period, AHMM may change when the state machine changes its state.

2.8.5 Self-Sizing Network

In [62], Saito proposed the "Self-Sizing Network" concept [63], which assigns network resources automatically to each traffic class and to each pair of origin and destination quickly, and with flexibility based on traffic demand. It meets the demands at every moment rather than using inflexible static resource allocation. Its self-sizing operation systems use measurement-driven traffic technologies [64][65], which simplify the traffic models as much as possible.

The first target of the Self-Sizing Network is dynamic bandwidth allocation, which uses flexibility characteristics of ATM [66]. They implemented a dynamic VP bandwidth control system on the TMN platform [67] that covers the nationwide ATM testbed in Japan. The real VP bandwidth in use is measured. This measured bandwidth is used to determine the VP bandwidth assigned at the next control epoch. A control epoch occurs every 15 minutes, so each operation cycle is 15 minutes. To assign enough bandwidth to the VP to maintain a CLR of 10^{-8}, the following algorithm is used.

The Algorithm

This algorithm takes into account the bandwidth/traffic variation of the previous day. Classify

- x as the measured bandwidth in use
- level i as $x : x_{i-1} \leq x < x_i$ where x_{i-1} and x_i are given constants
- $x(t)$ as the measured bandwidth in use at the tth measurement

Assume that $x(i)$ is in level i, that is, $x_{i-1} \leq x(t) < x_i$. $y(\mu_i)$ is the one measurement-period ahead of variation in the measured bandwidth in use. For each day and for each level, derive y_i, the largest variation based on the measured bandwidth in use; that is, y_i = max $y(\mu_i)$, to determine the VP bandwidth, the largest variation derived for the previous day is used. If the measured VP bandwidth in use during the latest measurement period is x, and if x is in level i, and if y_i is the largest variation in level i in previous period, then $x+y_i$ is the modified bandwidth in use. This bandwidth is an estimate of the bandwidth needed in the next measurement period. The bandwidth determined from the modified bandwidth in use is sufficient if the largest variation is equal to or less than the one occurring in previous period.

2.8.6 Dynamic Bandwidth Management in ACTS REFORM Project

The aim of the REFORM project (EU ACTS project AC208) [69-71] is to specify, implement and test a reliable system that offers ATM a multi-class and switched services. A hierarchical network resource management model was proposed in this project. One of the components of this model is Bandwidth Distribution [72].

Bandwidth Distribution

The Bandwidth Distribution (BD) component is responsible for the management of the bandwidth allocated to working Virtual Path Connections (VPCs) according to actual traffic conditions. BD adjusts the bandwidth allocated to the working VPCs to their actual usage to avoid situations where in the same links some VPCs tend to become over-utilised while other VPCs remain under-utilised. The dynamic management of the working VPC, is achieved by distributing portions (viewed as a common pool) of the link working bandwidth among the working VPCs.

The activities of BD are required to compensate the inaccuracies in traffic predictions. The management entity estimates the bandwidth that needs to be allocated to the VPCs so that satisfies traffic predictions. This is viewed as the mean value of the bandwidth that needs to be allocated to the VPCs.

The proposed algorithm [72] for bandwidth redistribution assumes that there is a common pool of bandwidth per link to be redistributed to the VPCs. The algorithm assumes that this common pool of bandwidth per link is the links' unallocated bandwidth. This pool of bandwidth is not totally allocated to the VPCs at any instant. Each VPC grabs and returns portions of its allocated bandwidth to the common pool according to its congestion level.

Redistribution Approaches

In distributing resources, there are three main approaches:

1. Generous – allocating as much bandwidth as possible to every VPC.
2. Greedy – allocate "just enough" bandwidth to each VPC.
3. Fair – taking into account information and dynamics of the other components of the system.

The chosen solution consists an Admission Zone (AZ) in which the VPC bandwidth is allocated. The AZ is defined around the Vreq (bandwidth allocated at initial establishing connection) value given by the management entity. BD is not allowed to allocate bandwidth outside these boundaries. The operating point of VPC as the last measured Vused is send to BD, this is the freshest measurement of the used bandwidth that BD knows. Around the operating point of the VPC (as shown in Figure 2.24) they defined a working zone (WZ), which defines an upper and lower threshold. Above this working zone is a buffer zone (BZ).

The Algorithm

Each time the bandwidth used crosses one of the working zone boundaries an alarm is raised. BD then centres the working zone around the new operating point, and allocates the VPC a new bandwidth equal to the upper boundary of the WZ. When this bandwidth reaches the upper limit of the admissible zone or when this bandwidth is not available because of link capacity restrictions along the VPC path or its protection path, the WZ being centred around the used bandwidth, BD will not repeatedly receive alarms since the upper boundary of the WZ will be above bandwidth allocated. If the VPC is empty or has very low traffic; again BD will not receive repeated alarms since the lower boundary of the WZ will be negative (or zero).

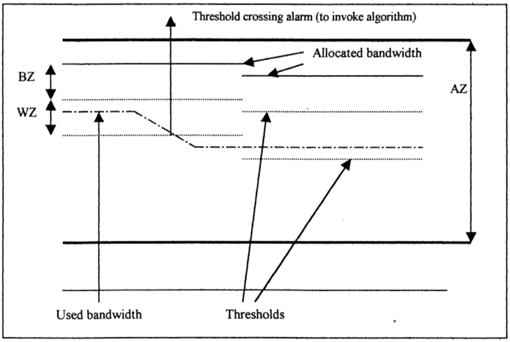


Figure 2.24 Tracking used bandwidth on VPCs [72].

The Admissible Zone (AZ)

- Narrow AZ means that little flexibility in altering the VPC's bandwidth.
- Wide AZ means that the Vreq shouldn't be taken too seriously (inaccurate prediction).

The Working Zone (WZ)

- Small WZ means that larger number of threshold crossing will happens and BD will invoke frequently.
- Larger WZ reduced the threshold crossing rate and processing load on BD.

The Buffer Zone (BZ)

- Small BZ leads to unacceptable cell blocking rates.
- Large BZ leads to an inefficient use of the link bandwidth (low utilisation).

2.8.7 Adaptive Neural Network for Dynamic Bandwidth Allocation

In order to allocate the slots in a frame ("hybrid" Time Division Multiplexing (TDM), where slots in a frame can be dynamically assigned to different traffic types) among different traffic classes and ensure QoS [85], while at the same time obtaining an efficient use of the bandwidth resources, some form of control must be exerted on the TDM multiplexer. Several attempts have been made to allocate the available bandwidth between service classes, and to find the optimal allocation policies [66-91].

The use of neural approximations in this kind of problems (whose dynamics is characterised by finite state Markov chains) becomes a practical interest mainly in the adaptive control case. In [85], they divided their approach into two parts [92] to introduce the structure of the controller. Neural Networks have been widely used in Connection Admission Control (CAC) and bandwidth allocation problems, especially in the context of ATM networks [93-95].

Implementing the related control laws on multiplayer Feed-Forward Neural Networks is quite a natural choice. This family of Neural Networks is characterised by three very important properties:

- possesses the ability of approximating non-linear functions (optimal control functions) by using a number of parameters that maybe surprisingly smaller than the one required by traditional polynomial or trigonometric expansions [96-99].

- allows a very easy and possibly distributed computation of the partial derivates of the process cost function [100].
- renders the use of Neural Networks in Markov Decision Processes (MDP).

Dynamic System Model and Bandwidth Allocation Problem

Consider a TDM frame of b seconds duration with a capacity of C slots. A free slot maybe assigned to one of two traffic types:

- a circuit-switched isochronous traffic with average connection request arrival rate λ_1 requests/seconds and average holding time $1/\mu$ seconds, or,
- packet-switched traffic with average arrival rate λ_2 packets/seconds, with fixed length packets (one slot).

Packets can be stored in a buffer, which has a finite capacity K. The circuit-switched traffic requires continuation of service (one slot/frame per connection request) until the end of the connection. The slots should have enough bytes to allow the transport of packets with reasonable length (53 bytes as in ATM networks).

The state of the whole system at a certain time instant is represented only by the number r of calls in progress ($r \in \{0, 1, 2, \ldots\}$). The control mechanism works as follows.

1) Let $t = 0, 1, \ldots,$ represent a discrete-time variable, counting the occurrence of an event in the isochronous traffic process.
2) Let r_t be the number of calls in progress at time t.
3) Whenever an incoming connection request for a circuit appears, it is either accepted with probability $\beta_f(r_t)$ or it is blocked with probability $1 - \beta_f(r_t)$.

Since the channel only has C slots, so $\beta_f(C) \equiv 0$. Thus $\beta_f(r_t)$ represents a randomised control strategy that maps r_t into probability of call acceptance. Suppose that packets can occupy all slots unused by the isochronous traffic, this strategy plays the role of a "probabilistic movable boundary" [67], and realises a partition of the available capacity in terms of slots/frame. The process r_t is a controlled birth-death process with upward rate $\lambda_1\beta_f(r_t)$ and downward rate μr_t. More specifically

$$P\{r_{t+1} = j \mid r_t = i\} = \begin{cases} \dfrac{\lambda_1 \beta_t(i)}{\mu_i + \lambda_1 \beta_t(r_t)} & j = i+1 \\ \dfrac{\mu_i}{\mu_i + \lambda_1 \beta_t(r_t)} & j = i-1 \end{cases}$$

To simplify the computational involved in the minimisation problem, it is more convenient to transform the continuous time process into an equivalent Markov chain [101].

Let $\mu C + \lambda_1$ be the maximum rate at which events can occur; then consider the embedded Markov chain with transition probabilities $P^t_{ij} = P\{r_{t+1} = j \mid r_t = i\}$ where

$$P\{r_{t+1} = j \mid r_t = i\} = \begin{cases} \dfrac{\lambda_1 \beta_t(i)}{\mu C + \lambda_1} & j = i+1 \\ \dfrac{\mu_i}{\mu C + \lambda_1} & j = i-1 \\ 1 - \dfrac{\lambda_1 \beta_t(i)}{\mu C + \lambda_1} - \dfrac{\mu_i}{\mu C + \lambda_1} & \mathrm{j = i} \\ 0, \quad \text{otherwise.} \end{cases}$$

2.9 Fair Bandwidth Share in Bandwidth Allocation

The ABR service is designed to fairly allocate the bandwidth unused by higher priority services. The network indicates to the ABR sources the rates at which they should transmit to minimise their cell loss [68]. The feedback from the switches to the sources is indicated in Resources Management (RM) cells, which are generated periodically by the sources and turned around by the destinations. RM cells contain current cell rate (CCR) and explicit rate (ER). The ER field is initialised to a rate lower than the PCR (peak cell rate) [7].

2.9.1 ERICA Algorithm

This algorithm aims at computing a fair and efficient allocation of the available bandwidth to all contending sources.

The Algorithm

The ERICA switch monitors the load on each link and determines a load factor, z, the available capacity, and the number of current active virtual connections. The load factor is calculated as follows:

$$z = \frac{\text{ABR Input Rate}}{\text{ABR Capacity}}$$

$$\text{ABR Capacity} = \text{Target Utilisation} \times (\text{Link Bandwidth} - \text{VBR Usage} - \text{CBR Usage})$$

The above steps are executed at the end of the switch measurement interval (input rate and output link ABR capacity are measured). Target Utilisation is a parameter, which is set to a fraction (close to, but less than 100%). The fair share of each VC, *FairShare*, is also computed as follows:

$$FairShare = \frac{\text{ABR Capacity}}{\text{Number of Active Connections}}$$

If a connection is sending at a rate below the *Fairshare*, then the switch will allow the connection sending rate rise to *FairShare*. If the connection does not use all of its *FairShare*, then switch fairly allocates the remaining capacity to the connections, which can use it. For this purpose, the switch calculates the quantity of *VCShare* as belows:

$$VCShare = \frac{\text{CCR}}{z}$$

If all VCs changed their rate to their *VCShare* values, then, in the next cycle, the switch would experience unit overload ($z = 1$). *VCShare* aims at bringing the system to an efficient operating point, which may not necessarily be fair. A combination of the *VCShare* and *FairShare* is used to rapidly reach optimal operation as follows:

$$ER\ Calculated = \text{Max}(FairShare\ ,\ VCShare)$$

The calculated ER value cannot be greater than the ABR capacity, which has been measured earlier. Hence,

$$ER\ Calculated = \text{Min}(ER\ Calculated\ ,\ ABR\ Capacity)$$

To ensure that the bottleneck ER reaches the source, each switch computes the minimum of the ER it has calculated as above and the ER value in the RM cell, and indicates this value in the ER field of the RM cell.

ERICA does not converge to max-min fair allocation. To overcome the fairness problem in ERICA, the algorithm is extended to remember the highest allocation made during each measurement interval, and ensure that all eligible connections can also get this high allocation. To do this, a variable called *MaxAllocPrevious* stores the maximum allocation given in the previous interval. For $z > 1 + \delta$, where δ is a small fraction, using the basic ERICA algorithm and allocate the Max(*FairShare* , *VCShare*). On the other hand, for $z < 1 + \delta$, equalise all the rate allocations, by assigning ER to the Max(*FairShare* , *VCShare* , *MaxAllocPrevious*).

The Measurement Interval

The length of the measurement interval limits the amount of variation, which can be eliminated. It also determines how quickly the feedback can be given to the sources. Longer intervals produce better averages, but slow down the rate of feedback.

2.9.2 An Accurate Method to Determine the Fair Bandwidth Share

In [67], Fahmy, Jain, Kalyanaraman, Goyal and Vaudalore proposed few ideas to compute accurate fair bandwidth share.

Basic Idea

They redefined the *FairShare* quantity to be the maximum share a VC could get at this switch under max-min fairness criteria. Hence, the *FairShare* is calculated as follows:

$$FairShare = \frac{\text{ABR Capacity}}{\text{Effective number of active connections}}$$

The main innovation is the computation of the effective number of active connections. The value of the effective number of active connections depends on the activity level of each of the connections. The activity level of connections is defined as follows:

$$ActivityLevel = \text{Min}\left(1, \frac{\text{Source Rate}}{\text{FairShare}}\right)$$

Thus, connections that are operating at or above the *FairShare* are counted as one. The connections that are operating below the *FairShare* only contribute a fraction. The connections that are bottlenecked at this switch are considered fully active while other connections are considered partially active.

The effective number of active connections is the sum of the activity levels for all connections.

$$\text{Effective number of active connections} = \sum \text{Activity level of connection}$$

2.9.3 MIT Scheme

This scheme is presented in [68][69]. This scheme computes max-min fair allocations for connections within a certain number of round trips. According to MIT scheme:

$$FairShare = \frac{\text{ABR Capacity-} \sum_{i=1}^{N_u} Ru_i}{N - N_u}$$

where:

Ru_i = Rate of i^{th} underloading source ($1 \leq i \leq N_u$)

N = Total number of connections

N_u = Number of underloading connections

Substituting N_o for the denominator term, becomes:

$$FairShare = \frac{\text{ABR Capacity-} \sum_{i=1}^{N_u} Ru_i}{N_o}$$

where:

N_o = Number of overloading connections ($N_u + N_o = N$)

or

$$FairShare \times N_o + \sum_{i=1}^{N_o} Ru_i = \text{ABR Capacity}$$

Factoring *FairShare* out in the left hand side:

$$FairShare \times \left(No + \sum_{i=1}^{No} \frac{Ru_i}{FairShare} \right) = \text{ABR Capacity}$$

or

$$FairShare = \frac{\text{ABR Capacity}}{No + \sum_{i=1}^{No} \frac{Ru_i}{FairShare}}$$

Substituting N_{eff}, we get:

$$FairShare = \frac{\text{ABR Capacity}}{N_{eff}}$$

where

$$N_{eff} = No + \sum_{i=1}^{No} \frac{Ru_i}{FairShare}$$

This means that the effective number of active connections N_{eff} is equal to the number of overloading sources, plus the fractional activity of underloading sources.

2.9.4 Min-Max Algorithm

In [73], Biswas and Izmailov proposed 4 algorithms were based on min-max policy to re-allocate bandwidth among all contending links. Periodic recalculation and reassignment of portions of bandwidth according to some metrics involving buffer occupancies and the traffic rates can improve bandwidth utilisation [59][74-77].

Problem Description

A single server serves N traffic streams (as shown in Figure 2.25), and the service time is slotted. The formulation is denoted as follows:

A – available capacity (bandwidth)

λ_k^i - bandwidth requirement for the ith stream in kth slot

x_k^i - buffer occupancy of the ith stream in kth slot

Assuming that the excess traffic flow at the next time slot will be the same as that in the current slot k, the ith stream requires ($x_k^i + \lambda_k^i$) for slot $k + 1$ to clear the traffic.

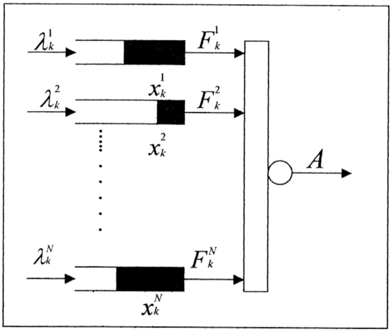


Figure 2.25 Dynamic Capacity Allocation Model [73].

Proportional Linear Algorithm

To allocate the available capacity A in linear proportion to the requirements $(x_k^i + \lambda_k^i)$, therefore, the allocation is determined by the formula:

$$F_{k+1}^i = A \frac{x_k^i + \lambda_k^i}{\sum_{i=1}^{N} (x_k^i + \lambda_k^i)}, \quad \mathrm{i} = 1, \ldots, \mathrm{N}.$$

This method was found to be very effective in [78], but unfairness (large queue becomes larger) occurred when congestion happened [73].

Proportional Polynomial Algorithm

This method could achieve by a non-linear allocation procedure, where the bandwidth is allocated in proportion to polynomials of the capacity requirements. The allocation for $(k + 1)^{th}$ can be rewritten as:

$$F_{k+1}^i = A \frac{(x_k^i + \lambda_k^i)^n}{\sum_{i=1}^{N} (x_k^i + \lambda_k^i)^n}$$

where n indicates the degree of polynomial. In this algorithm, relatively more resource is allocated to the streams with larger requirements, and the allocation towards the faster growing queues gets more skewed as n increases.

Proportional Exponential Algorithm

The idea of this algorithm is to allocate capacity in proportion to exponents of the excess requirements of the competing streams:

$$F^{i}_{k+1} = A\frac{\exp(x^{i}_{k}+\lambda^{i}_{k})}{\sum_{j=1}^{N}\exp(x^{j}_{k}+\lambda^{i}_{k})}$$

This algorithm has the same behaviour of buffer contents as the min-max algorithm while being linear on N in terms of complexity.

Min-Max Algorithm

During congestion, if there is not enough capacity to keep the buffer contents at the same level, a fair distribution would be to keep the maximum queue length as small as possible. For given x^i, λ^i and A, the task is to find suitable F^i values so that the quantity

$$\max\{ x^i + \lambda^i - F^i \}$$

is minimised, subject to

$$\sum_{i=1}^{n} F^{i} = A, \quad F^{i} \le x^{i} + \lambda^{i}, \quad i = 1,\ldots,N. \quad F^{i} \ge 0.$$

The solution of the problem consists of two steps.

1. Rearrange all the requirement numbers ($x^i + \lambda^i$) in descending order ($x^1 + \lambda^1$) ≥ .... ≥ ($x^N + \lambda^N$).
2. Min-Max algorithm sequentially performs the following operations:
   - Calculates the portion a_1 of A that needs to be allocated to ($x^1 + \lambda^1$) so that the remaining requirement to ($x^1 + \lambda^1 - a_1$) is equal to the next largest requirement, which is ($x^2 + \lambda^2$).
   - Calculates the portion a_2 of the remaining capacity A - a_1 that needs to be allocated to both ($x^1 + \lambda^1 - a_1$) and ($x^2 + \lambda^2$), so that the remaining requirements ($x^1 + \lambda^1 - a_1 - a_2$) and ($x^2 + \lambda^2 - a_2$) are equal to the next largest requirement, which is ($x^3 + \lambda^3$).
   - Continues to allocate the remaining bandwidth in the same way until all the available capacity is exhausted.

When this process stops because of exhausted bandwidth, a has been effectively split into portions $F^1,\ldots,F^n$.

2.10 Buffers

Within the networks, buffering is required to smooth out periods of overload to avoid cell losses. The requirements of these buffers often vary from one type of service to others:

- Buffers must be kept short in order not to cause unwanted delays, for delay sensitive traffic such as voice.
- Buffers must be made long to avoid unwanted losses, for loss sensitive traffic such as data.

Obviously these demands are contradictory, and there are two possible ways, either [79]:

- Use separate buffers for different services, or
- To go for a compromise such as some sort of medium long buffer.

Separate buffers are more complicated to handle and longer buffers might have to rely on low speed technology, hence such buffers are useful only when transmission speeds are not too high. Common buffers are simpler to handle. Short lengths promote the use of high-speed technology.

2.11 Chapter Summary

This chapter explored the areas of ATM networks, ATM network management (congestion control), dynamic bandwidth allocation scheme, fair bandwidth share, artificial Neural Network, Fuzzy Logic, and buffer occupancy. ATM provides variety and better quality of services to the users. Dynamic Bandwidth Allocation improves the quality of service by increasing the bandwidth utilisation and reducing the buffers occupancy and the cell dropping rate.

The core problem of dynamic bandwidth allocation is to allocate "just-enough" bandwidth to the connections while retaining low buffer occupancy and cell dropping rate. This problem can be overcome by using the Neural-Fuzzy algorithms. The following chapter focuses on implementation of the Neural-Fuzzy algorithm.