A Review of the Resource Management Task in ATM Networks

Shane Naughton *

Department of Computer Science,
Trinity College, Dublin 2, Ireland.

Tel: +353-1-608-1800
E-mail: Shane.Naughton@cs.tcd.ie

September 5, 1995

1 Introduction

1.1 General Definition of ATM (Asynchronous Transfer Mode)

An ATM network is a connection-oriented, packet-switched network with the ability to transport heterogeneous services irrespective of characteristics across the same underlying physical network. ATM networks were designed with the extremely high quality and high speeds of existing transmission systems in mind, and the potential future trend in the development of a multitude of services with diverse communication requirements, resulting in a flexible and future-safe network.

This paper presents a review of the resource management tasks that are necessary to provide network users with their requested performance criteria for the diverse and wide-ranging services they use, while at the same time ensuring there is a sufficiently high and balanced utilisation of the network resources.

Section 2 provides an overview of the ATM network and the concepts of virtual connections and virtual paths. Section 3 discusses the need for resource management of ATM networks. Section 4 breifly discusses recommendations in developing resource management techniques. Section 5 considers on-line traffic control mechanisms, Section 6 higher-level network pre-dimensioning mechanismsm, and Section 7 high-level on-line adaptive network dimensioning mechanisms. Finally, Section 6 provides a conclusion.

2 The ATM Network

2.1 Physical Network

As mentioned, an ATM network is connection-oriented, packet-switched network, with network operation revolving around the transmission of all information within small fixed-size packets (called *cells*) containing 5 bytes of header information and 48 bytes of actual data.

*This research is supported by Broadcom Eireann Research Ltd., Clanwilliam Place, Dublin 2, Ireland

The physical network consists of a set of network nodes (or switches) interconnected by physical links, each with a fixed bandwidth capacity. There are two types of nodes: *switching* (transit) nodes which switch packets from incoming links onto outgoing links; and *access* nodes, which in addition to being switching nodes, allow for cells to be passed into and out of the network from / to users of the network through the User-Network Interface (UNI).

2.2 Virtual Connection

Every connection established in the network is represented by a Virtual Connection (VC). If a customer wishes to set up a flow of information between two nodes in the network, then a connection is explicitly established (connection-oriented) between these two nodes of the network. The data to be transferred is split into segments, encapsulated into ATM cells, and these cells are subsequently sent between these two nodes along a fixed route. The data is extracted from the cells at the destination. Each intermediate node along the route has knowledge of the connection, and switches the incoming cells for that connection, arriving on an incoming link onto the appropriate outgoing link.

2.3 Virtual Network

In addition to the concept of a Virtual Connection (VC), there is also the concept of a Virtual Path (VP): in essence a VP is a virtual link between two nodes in the network which do not necessarily have a direct physical connection. As with a physical link, it is considered to have a fixed allocated bandwidth through which connections can be routed. A Virtual Path represents a long-term flow of traffic between the two nodes that it connects.

While the physical links and nodes define the physical network, the access nodes and the Virtual Paths established between them define the Virtual Network topology.

2.4 Tagging and Switching

Virtual Connections are routed through Virtual Paths, and each cell of a given connection is tagged with two values (contained within the packet header):

Virtual Path Identifier (VPI): which uniquely defines the VP to which the cells belongs on that physical link, and

Virtual Connection Identifier (VCI): which uniquely defines the VC within the Virtual Path (defined by the VPI). The VCI is not sufficient in itself to uniquely identify a connection on a link, it only uniquely identifies the connection within the Virtual Path.

Virtual Paths introduce a hierarchy of routing into the network. Switches can be further subdivided into:

VP switches (cross-connects) These determine the route of Virtual Paths through the network, and do not consider the individual connections which make up the Virtual Paths. The switch examines the VPI value of each packet on each incoming link to determine the VP to which the packet belongs on that link. The routing table at a VP switch will have an entry for each VPI on each incoming link, with information as to which outgoing link the packet should be switched to, and the updated VPI value for the packet on that outgoing link. The VCI value remains unchanged.

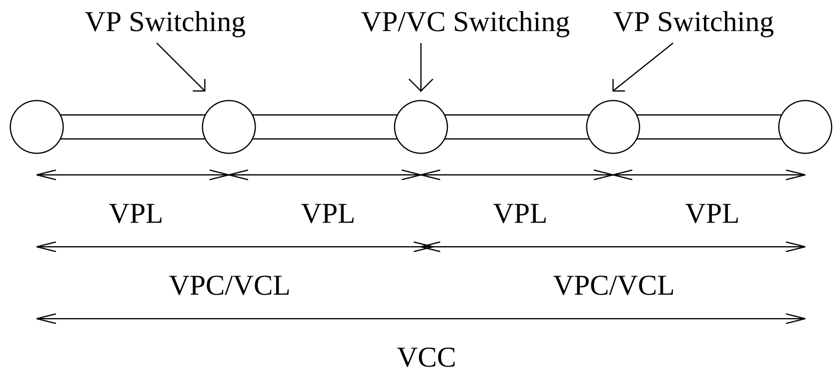

Figure 1: ATM network entity relationships

VP/VC switches (Virtual Path endpoints) These examine both the VPI and VCI values of the packet on each incoming link. The packet is switched onto the approporiate outgoing link (with updated VPI and VCI values), with the outgoing link and new VPI and VCI values determined by the incoming VPI and VCI values.

The advantages which Virtual Paths offer are:

- reduced routing. The amount of routing and control necessary to manage connections at intermediate nodes is substantially reduced. A set of n connections between access nodes a and b would necessitate n entries in the routing tables of all intermediate nodes, whereas 1 Virtual Path into which these connections are bundled requires only 1 entry in each routing table.

- resource separation. Virtual paths between two nodes can be used to split the connections between these nodes into homogeneous groups according to their characteristics and network requirements. If n different groups were identified, then n Virtual Paths might be used to provide the network resources for these service connections. As will be explained, this can simplify resource management tasks.

2.5 Acronym Definitions

User-to-User traffic flows within the network are represented by Virtual Channel Connections (VCCs). VCCs are carried through the network within a sequence of Virtual Path Connections (VPCs). A VPC is a direct link between two nodes in the network which are connected via two or more physical links. Each individual component link of a VPC connecting two nodes is called a Virtual Path Link (VPL). Each Virtual Path which comprises a VCC is called a Virtual Channel Link (VCL). See Figure 1

3 The need for Resource Management

What do ATM networks offer that is significantly better than existing communication networks? The answer is *statistical multiplexing gain* — the ability to sell more bandwidth than is actually available. The crucial factor is that many existing services do not provide a constant fixed stream of information to be transferred across the network (μlaw encoded voice transfer), rather information is provided in bursts, punctuated by periods of silence (compressed video). Historically, a connection has been allocated bandwidth which corresponds to its peak transmission rate. The bandwidth of such a connection was reserved for it alone, and this bandwidth during periods of non-transmission could not be used by other connections, resulting in needless waste.

ATM networks work on the principle that the sum of the peak transmission rates of such connections on a given link are allowed to exceed its actual capacity, but only if the probability of any connection on the link experiencing congestion is at an acceptably low level. This congestion would result from the simultaneous transmission on a number of connections, such that the short-term combined bandwidth requirement of the connections exceeds the actual bandwidth of the link.

3.1 The need for node buffers

Network nodes can only process incoming cells at a fixed rate, and can only transmit cells at a fixed rate (defined by the bandwidth of the outgoing link). When packets arrive simultaneously on different incoming links that have to be transferred onto the same outgoing link, then all but one of the cells have to be delayed until the other cell has been sent.

It would be totally unacceptable to network users if the network nodes were to simply discard cells which could not be processed immediately, so network nodes provide buffers into which such cells can be placed until such a time that they can be processed. Buffers are of a finite size, so the amount of such simultaneous transmission that can be handled is limited. Any cells that arrive when buffers are full are discarded. A switch could not handle the situation where cells on two incoming links of 10 Mbits each were both being switched onto the one outgoing link of 10 Mbits — cells would build up indefinitely. Buffers therefore can only alleviate short term bursts from traffic sources.

While buffers add to the robustness of the network by reducing the amount of cell loss, they also introduce unwanted side effects, in particular cell delay. When cells arrive at a switch, and cannot be processed, they are placed into a buffer until they are ready to be processed. This introduces delays into the system, with any one cell possibly experiencing such delays at every switch along the route from source to destination. Cell delays are also introduced at connection multiplexing and demultiplexing, though these are usually fixed delays. Additionally, depending on the transmission status of other connections, the delays which different cells from the same connection receive can vary significantly. Some services can be very sensitive to end-to-end cell delays while others can be sensitive to the variation in cell delay.

3.2 Quality of Service

Many different customer services have very similar requirements in terms of the quality of the service they need to receive from the network. Various different *Quality of Service (QoS)* classes have been defined, each to represent a set of services that have the same QoS requirements. Each defines the Quality of Service that a Virtual Connection of that service class must receive from the network. Each QoS class is defined in terms of a subset of ATM cell transfer performance parameters, in particular the maximum acceptable cell delay variation, and the cell loss probability. The four defined service classes are:

Service Class A : Circuit Emulation, Constant Vit Rate (CBR) video

Service Class B : Variable Bit Rate (VBR) audio and video

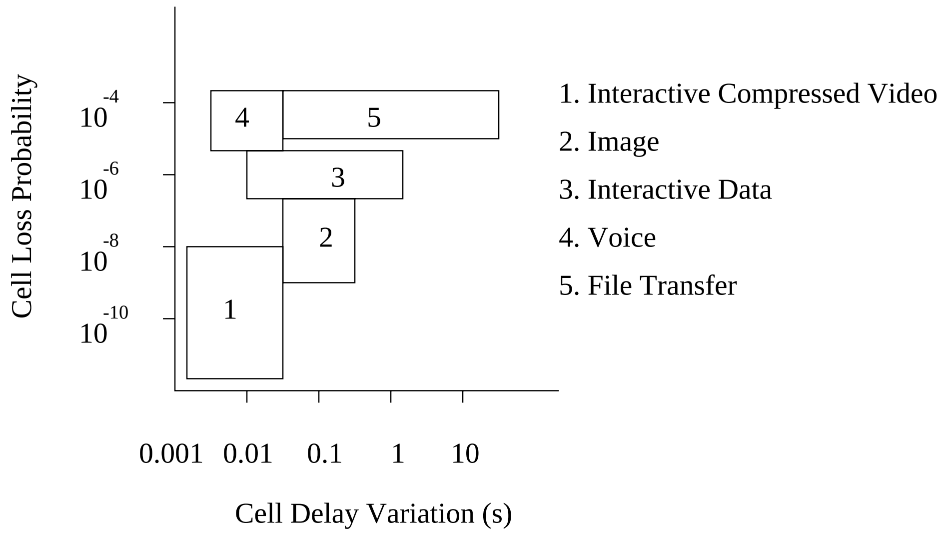

Figure 2: Some sample Quality of Service Requirements

Service Class C : connection-oriented data transfer

Service Class D : connectionless data transfer

For example, CBR voice (uncompressed) might want small cell delay in order to maintain a certain sampling rate, yet can be somewhat tolerant of cell loss, while a compressed video service might not tolerate any cell loss. Non-realtime services such as data transfer can be tolerant of cell delay and cell delay variation. Standard voice can tolerate a cell loss ratio of 10^{-3}, while HDTV can only tolerate a cell loss of 10^{-10}. See Figure 2 [6].

3.3 The Traffic Contract

The task of the network manager is to provide all customers with an agreed *quality of service* while at the same time, ensuring that network resource usage it maximised. This can only be done if the requirements and characteristics of each connection are accurately defined. The network manager must have sufficient information to determine accurately whether or not the addition of a new connection to the network will violate the quality of service being provided to any existing connections.

To facilitate the network in determining whether a proposed connection can be accepted, a *traffic contract* is negotiated at the User-Network Interface (UNI). A negotiated traffic contract consists of

- a QoS class for each direction
- a **Connection Traffic Descriptor**. The Connection Traffic Descriptor in turn consists of
  - a **Source Traffic Descriptor**. This consists of a set of traffic parameters such as
    - Peak Cell Rate
    - Sustainable Cell Rate

    * Burst Tolerance
    * Burst length
  - a conformance definition. What actions need to be taken if the connection violates its contract.

Such a negotiated and agreed traffic contract is used to ensure that reliable connection acceptance or denial decisions can be made. The network has to ensure that the QoS contracts of the existing connections are maintained, while also providing the QoS to the new connection.

By providing such a flexible system of traffic requirements and QoS levels, the task of connection management has been made more challenging, not least in deciding whether to accept connections, but also in ensuring that they do not violate their contracts during their lifetimes and that the QoS levels agreed at connection setup are maintained.

3.4 Efficient Resource Allocation

While it is very easy to provide the necessary resources to a set of connections, it is not so easy to do so in an efficient manner such that there is a significant statistical gain. For example, to meet variable bit-rate connection resource requirements, bandwidth equal to their peak transmission rates could be allocated to each. No statistical gain would be achieved, however,and there would be significant wasteage of resources.

The objective is to provide to each connection the *minimum* resources necessary such that its QoS requirements are met, thus maximising resource usage. However, in achieving this, the margin of error in the network is reduced, it is possible that a connection which violates its agreement will adversly affect the performance of other connections with which it shares bandwidth. Policing techniques to ensure connections do not violate their contracts are discussed in the next section.

4 Resource Management

Resource Management needs to be performed at two levels. At the lower level, there is a need for on-line traffic control mechanisms that work with a given virtual network topology which aim to prevent long-term and short-term congestion in the network. At the higher level, a set of mechanisms are needed to determine the actual virtual network topology from high-level models of source-destination traffic requirements, and its mapping to the underlying physical network topology.

4.1 Guidelines for generation of Resource Management Techniques

There are four recommendations when developing any ATM resource management techniques, and they are as follows [3]:

simplicity The need to achieve high resource efficiency while maintaining simple control structures, due to the high speeds involed in the network.

flexibility The need for the flexibility to adapt as needed to new situations that may arise. e.g. the emergence of new services

robustness The ability to be relatively insensitive to imperfect assumptions or insufficient information. e.g. graceful degradation of service.

controllability The need to control traffic and congestion so that efficient resource utilization can be achieved without performance penalty.

Techniques which aim to *prevent* congestion before it occurs, rather than *react* to congestion when it actually does occur are favoured, for a variety of reasons.

Firstly, due to the high speeds that ATM networks operate at, and the small size of cells, a large number of cells can be in transit at any one time. Once congestion has been detected, and a remedial technique decided upon, then this cannot affect cells already in transit (a 150Mbps link of 1000km in length has a propagation delay of 3ms, enough to load the link with 1000 cells [11]).

Secondly, cell loss should be avoided due to the positive feedback effects — the receiving node has to inform the sending node to re-send the lost information, and the information then has to be re-sent, both transmissions adding to the existing congestion in the network. This can drive the network into a congestion collapse state [15].

4.2 Organisation

The discussion of Resource Management is split into three sections. Firstly, on-line traffic control mechanisms at a cell and call level are discussed. Secondly, network pre-dimensioning mechanismsm are presented, and finally, on-line adaptive re-dimensioning techniques are considered.

5 On-line traffic control mechanisms

These control mechanisms work at two levels. At the *call level*, they aim to regulate the admission of calls into the network; at the *cell level*, they try to ensure that the cell stream of each call conforms to its agreed contract. Given that these operate on-line, there is a requirement that they be simple, so that no significant overheads are added to the system. However, simplicity is only achieved at a cost of reduced accuracy, so an agreeable balance has to be found.

5.1 Call-level controls

Call-level controls aim to avoid long-term congestion in the network and maintain the traffic load at a manageable level by regulating the admission of new calls into the network. The task of determining whether or not a new connection can be accepted into the network is called *Connection Admission Control (CAC)*, and is performed at the entry point to every VP which the connection traverses in reaching the destination node.

CAC determines whether there are sufficient resources available in the network to open a connection with the required QoS, while at the same time maintaining the QoS requirements of all existing connections. CAC makes its decision based upon the connection's anticipated traffic characteristics, its requested QoS, and the current link load. While there are defined guidelines what tasks the CAC should perform, the actual implementation is left unspecified, and therefore there are many instantiations of the CAC of varying degrees of complexity and accuracy.

CAC also determines the traffic parameters required for cell-level control (defined later), and determines the routing of the connection through the network and allocation of network resources for the connection request, if successful. CAC also covers dynamic call renegotiation when a call is determined to be violating its contract such that a modification of its traffic contract is necessary.

Central to the task of call-level control is determining the amount of bandwidth that a call is expected to use, called the *equivalent bandwidth*. Equivalent bandwidth is normally bounded on top by the peak transmission rate of the connection, and on the bottom by the sustainable cell rate, and is determined by examining the requested traffic parameters. Some of the effects of traffic parameters on equivalent bandwidth are as follows [6]:

- a higher burst length can increase cell losses and cell delays (by swamping buffers), necessitating a rise in the equivalent bandwidth.
- the ratio of peak bandwidth of a call to the *link* bandwidth can be important, with lower values (small peak compared to link) increasing the potential multiplexing gain and decreasing the equivalent bandwidth.
- burstiness (the ratio of peak to average bandwidths) can be important if there is a low value for the above peak-to-link bandwidth. As burstiness increases, equivalent bandwidth tends towards the peak bandwidth.

The main traffic-dependant factors which can cause a deterioration in the provided QoS are cell loss and cell delay. The CAC has to ensure that the combined load on the link or VP of the current connections and the new connections does not cause a deterioration in these factors to such an extent that the agreed QoS levels of current connections are violated, or that the QoS of the new connection cannot be provided. In such situations, the requesting connection is rejected.

5.1.1 Some call admission schemes

There are many existing implementations of the CAC algorithm, some focusing on simplicity and ease of implementation (low hardware costs, minimal information storage), while others focus on accuracy. As always however, accuracy is only achieved at a penalty of higher processing times, higher amounts of stored information, or increased on-line realtime processing.

The simplest method is to admit calls as long as resources are available. This is acceptable in situations where VPs are dedicated to homogeneous services, but can lead to unfairness if connections of service classes with differing bandwidth requirements are allowed to share the same link or VP.

Fixed CAC admission schemes use only the parameters of the traffic contract to determine whether or not a call can be accepted, while *dynamic* [12, 11] CAC schemes uses the on-line measurement of actual cell streams (measurement of the number of arriving calls) as well as the traffic contract. The dynamic schemes are useful in cases where it is difficult to accurately define values for the traffic descriptor parameters at the start (connections belonging to a new service whose characteristics have not been defined or examined). Dynamic schemes can also achieve higher resource utilization than fixed schemes.

While primarily used to ensure the required QoS levels are provided to connections, call-level controls also function as preventative congestion control mechanisms — resources are only allocated if available, thereby preventing over-allocation of resources which can lead to congestion. Enforcement of individual connections to their contracts is also required to prevent congestion however, which is the task of cell-level control mechanisms.

- Neves et al. [9] propose using feed-forward neural networks (back-propagation) to aid in Call Admission Control (CAC) by predicting the QoS changes caused by each new call. It achieves this by learning traffic patterns in previous traffic characteristics.

  Each node processor along the proposed route for the call asks the neural network for the expected traffic load pattern on the relevant outgoing link, with and without the inclusion of the new connection. Quality of Service is evaluated in both cases, and the call is accepted if a deterioration in quality is not predicted at any of the nodes.

Neural network inputs are the allocated bandwidth to each QoS class for that link, and the outputs can be the expected delay, cell loss rate, and the maximum and minimum buffer occupation, the latter leading directly to the cell delay variation.

To train the neural network, patterns of the traffic load in a node or a link can be collected during the operation of the network in different traffic situations.

- The approach of Nordstrom et al. [10] is to attempt to reproduce the Quality of Service estimations (probability of cell loss, ...) by using a multi-layer perceptron as a function approximator, for use again in Connection Admission Control (CAC).

  The system uses a fluid flow performance model, modelling the disrete cell flows as a continuous fluid flow, resulting in an input state vector for the neural network containing six parameters, derived from knowledge of the buffer queueing system. Three describe the distribution of the arrival rate, two describe the correlation structure, and one reflects the overall behaviour of the queueing system.

- Hiramatsu [5] proposes an ATM controller using back-propagation (three layered, fully connected) networks for learning the relations between the offered traffic and service quality. There is one neural network per ATM processing node.

  The inputs are the observed node status (cell arrival rate, cell loss rate, call generation rate, trunk utilization rate, number of connected calls), a history of past observed node status, and the declared parameters in the call setup request (average bit rate, ...)

  The outputs from the neural network consist of predicted service quality parameters (cell arrival rate, cell loss rate), and decision values for deciding upon the acceptance or rejection of the proposed call.

- Duffield et al. [2] propose a method for estimating the Quality of Service parameters for ATM traffic, by modelling a buffer and its QoS parameters (cell-loss frequency and cell delay). Incoming traffic on a given line is modelled using a two-state Markov process, with defined probabilites for switching from an active state to a dead state, and vice-versa. At a switch, there are l lines, the traffic on each line being independent, but having the same distribution, and all are to be multiplexed onto one output line.

  At any one clock cycle, the number of active lines defines the arrivals process, and a bound for the cell-loss frequency can be got using the Duffield-Buffet formula, which is dependant upon the number of service lines, the size of the buffer, and the service rate, and uses parameters derived from the Markov model parameters.

- Morris and Samadi [8] propose a neural network system which in the long term learns which call combinations result in acceptable performance, and in the short term reacts to changing network conditions. Key network performance parameters are observed while carrying various combinations of calls, and their relationship is learned by a neural network structure. k neural networks, one for each performance metric of interest, and a call is only admitted if the resulting k-dimensional performance vector is satisfactory.

5.2 Cell-level controls

Cell-level controls are used to avoid short-term congestion in the network by ensuring that connections do not violate their contracts and consequently impede on the performance of other connections with which they share links. The task of cell-level control on individual connections is called *Usage Parameter Control (UPC)* and is performed at the entry point of the connection to the network.

UPC consists of a set of actions taken by the network to monitor and control its traffic in terms of the validity of each ATM connection at the UNI — the detection of violations by connections of the negotiated traffic parameters and the taking of appropriate actions. Ordinarily, the UPC is performed for each traffic parameter in a source traffic descriptor. At present however, the only standardised parameter is the peak cell rate.

Again, UPC is a guideline and any algorithm can be employed. The ideal algorithm should [7]:

- detect and respond to illegal traffic situations
- have a rapid response time to violations. Some violations are easier to detect than others. Violations of parameters like maximum allowed burst length can easily be detected, while violations of parameters like mean transmission rate are harder to detect. In particular what sort of acceptable time interval should be used in measuring the mean transmission rate ?
- be simple. The operation of the UPC, since it is an online monitoring task, can cause connection performance degradations. Simplicity is necessary to ensure that UPC itself does not cause the violation of QoS objectives.

There are two main parts to cell-level control: *traffic policing* and *traffic shaping.*

5.2.1 Traffic Policing

Traffic policing ensures that connections do not violate their contracts, indirectly ensuring that connections sharing the link are provided with their agreed QoS levels. There are various algorithms for implementing these, the most popular being:

leaky bucket algorithm [6, 11] This polices connections using a conceptual bucket which has a fixed capacity but is leaking at a constant rate. The "leaking" or processing rate correpsonds to the rate to be policed (peak or average cell rate), the bucket size ensures tolerance of cell delay and cell delay variation factors. If the bucket is full on the arrival of a new cell, then the UPC determines that the connection has violated its contract, and the cell is discarded. The disadvantage of this mechanism is that traffic policing is enforced even when there is a low network load (and no need for enforcing traffic contract compliance). The mechanism is also prone to mistake non-excesive traffic as excessive, which is wrongly discarded. This problem is avoided with the virtual leaky bucket algorithm.

virtual leaky bucket algorithm [6, 11] This is similar to the previous case except cells are not discarded, rather they are tagged as being discardable, leaving the task of discarding them to network switches along the connection path which might be experiencing congestion.

windows [11] This involves counting the number of cells in a time window, marking the connection as violating if the count exceeds a certain threshold. There are three variations:

- **jumping window method** : The windows do not overlap, but stick to each other.
- **overlapping windows** : The windows overlap, resulting in more accurate results, but significantly increased processing.
- **exponential windows** : The same as the jumping method, except the threshold is a function of cells accepted in previous windows.

flag UPC [11] A special cell is sent periodically during which up to a specified number of cells may be sent by the user of the connection.

Neural networks [14] This aims to produce an accurate estimation of the probability density function (pdf) of traffic by using two interconnected neural networks. One is trained to learn the pdf of ideal non-violating traffic, the other trained to capture the pdf of "actual" characteristics of the actual offered traffic during the progress of the call.

The output of both are compared, and an error signal is generated whenever the pdf of the offered traffic violates the ideal one. This system has the advantage that actual pdfs of traffic include all the statistical properties of the traffic.

5.2.2 Traffic Shaping

Traffic shaping is performed at the input stream source and is used to smooth the incoming traffic by providing buffers at the entry point of each connection to the network. Traffic can be reshaped to desirable or to originally defined traffic characteristics. While CDV is minimised, it is minimised at a cost of increased delay, resulting in possible cell loss. Traffic shaping can also affect peak bandwidth due to clumping of cells. Also, a hardware device for cell shaping is required for every connection.

6 Network Pre-Dimensioning Methods

These work at a higher level than the on-line mechanisms just discussed, and relate to the generation of the virtual network topology, which encompasses the identification of potential virtual paths, the calculation of their bandwidths, and the mapping of the resulting VPs onto physical routes within the network. It is important to note that these tasks are subject to pragmatic network operator constraints such as load balancing in the network, and the maximisation of network traffic throughput (measured in terms of the overall number of accepted calls).

These set of tools operate at a pre-dimensioning stage - the initial dimensioning of the ATM network given expected future traffic characteristics. To adequately describe the pre-dimensioning task, it is necessary to provide a network model and a dimensioning model.

6.1 Network Model

Traffic is separated into a set of *traffic models*, each traffic model representing a flow of traffic between two particular nodes in the network for a particular homogeneous service. The traffic model describes such characteristics as the average rate of generation of connection requests, and the mean holding time of accepted connections, as well as the QoS requirements for the connection.

This traffic is transported through the network along one or more allowable *routes* between the source and destination nodes, where each route is catered for along a specific sequence of virtual paths. (Each virtual path can therefore carry more than one route at any one time, the routes not necessarily connecting the same source-destination pairs). Again, there are two levels — the first, *connection-level dimensioning* determining the bandwidth allocations to VPs, the second, *VP-level dimensioning* determining routings for virtual paths on top of the physical network topology.

6.2 Dimensioning Model

There are three distinct tasks to be solved (See Figure 3):

1. Determining the bandwidth necessary to meet the connection-level Quality of Service objectives for the traffic source.
2. Determining the bandwidth of VPs in the system, given the bandwidth requirement of each traffic source, and the allowable routes for the traffic source. Operator constraints such as load balancing and network throughput maximisation may apply here, and statistics such as the load sharing coefficients (distribution of traffic source bandwidth over allowable routes) are returned.
3. Determining a routing for these virtual paths on top of the physical network, given the virtual path topology (source and destination nodes) and associated virtual path bandwidths.

6.3 Step 1: Connection level dimensioning

This entails providing agreed *connection-level* Quality of Service requirements to users. The primary task at this connection level is to satisfy the customers' connection loss probability objective - the number of connections requested by the customer but rejected by the network must be below an agreed threshold.

To provide the connection-loss probability objective to the user, it is necessary to compute the bandwidth necessary to keep the rate of rejection of connections beloq a certain level. The required bandwidth is calculated given the source traffic model and QoS requirements for the connections, and is a non-trivial task, given that the necessary bandwidth is a non-linear function of the traffic model characteristics and the required QoS.

If, as assumed, we are to restrict traffic to homogeneous traffic with the same QoS requirements, then the problem can be solved as follows.

1. Connection-level computation. Determine the number of connections, n, which the route must be able to handle simultaneously such that the connection-loss probability objective is met. This can be acquired by using an Erlang-B formula which uses the poisson arrival rate and the mean holding time of calls.
2. Cell-level computation. Determine the necessary bandwidth which should be allocated to the route such that the CAC function will allow n connections on that route at any one time. This is a complicated function of the number of connections, n, the cell-level QoS requirements, and the traffic characteristics, and is non-linear with respect to n (the necessary equivalent bandwidth of a connection decreases as the number of connections increases)

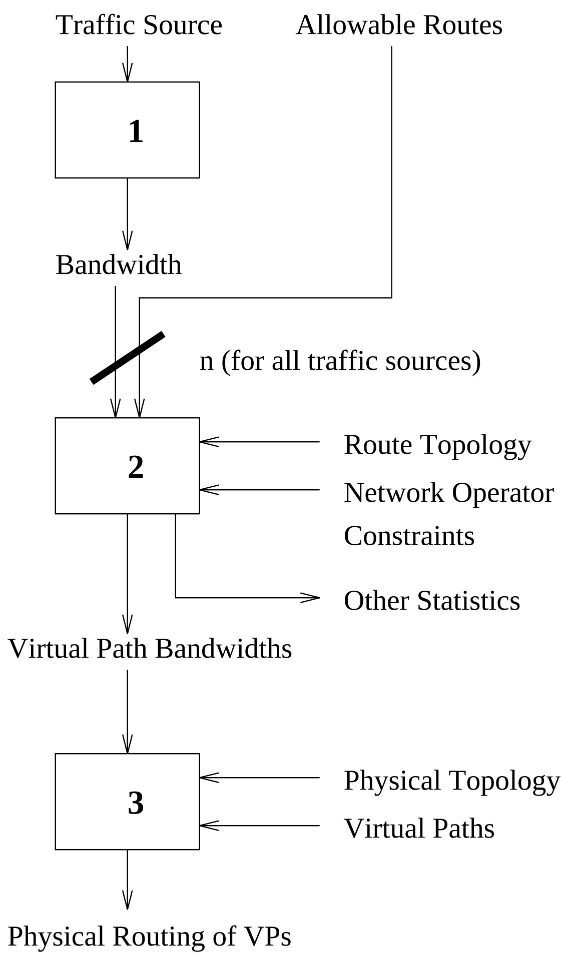

Figure 3: Pre-Dimensioning Model.

If we are not to assume homogeneous traffic , then the task becomes more complicated, requiring an algorithm which iterates between the connection-level and cell-level computations.

6.4 Step 2: VP Bandwidth Computations

- Equivalent Link Blocking (ELB) method [13]

  The goal of the algorithm is to find virtual path capacities, given route capacities, such that the connection-loss probability objectives for all routes are satisfied. The solution is not optimised.

  CAC has to be performed at the entry-point to each virtual path along the route between the source and destination in order to determine if a connection can be accepted, therefore each of these contribute to the overall connection-loss probability. The probability of accepting a connection is given by the product of the probabilities at each VP (connection has to be accepted at VP1 *and* then at VP2 and then ...). The responsibility for providing the connection-loss probability objective for a given source-destination pair can be distributed evenly across all links which the route traverses (hence the name Equivalent Link Blocking), and can be easily calculated from the route connection-loss probability objective and the number of VPs used within the route.

  From the perspective of a virtual path, there are a set of routes passing through it, each of which has a connection-loss probability objective for that link. Logically, the link must provide a connection-loss probability objective which is *at least as stringent* as the most stringent one specified.

  So, we can evaluate the bandwidth requirements for each link using Erlang's formula, given the bandwidth requirements, offered Poissonian traffic, and calculated probability objective of each route using the link.

- Kelly's derivatives [13]

  An even more general and more complex algorithm. For each combination of source-destination pair and traffic class, there is a set of allowable routes through the network (again defined by a set of links), and a poissonian arrival rate. The traffic for this particular traffic class is to be statistically shared between all the routes (such that the sum of the traffic on the routes equals the traffic originating for that traffic class).

  The goal of the algorithm is to find these *load sharing coefficients* for the source-destination pair traffic classes (what proportion of the traffic is allocated to each allowable route), and as before the virtual link capacities, such that the network revenue is maximized, where the completion of a call on route r contributes a constant c_r to the total revenue. If the constants for all routes are the same ($c_1 = c_2 = \cdots$), the the objective reduces to maximizing the total number of calls which pass through the network. The algorithm is extremely complex due to the very large number of parameters involved, and uses a gradient-based search to find a local optimum of the network revenue.

- Iterated ELB method [13]

  Similar in objective to the previous Kellys derivatives - the maximisation of network revenue, except that the optimisation for load sharing parameters and logical capacities have been separated in order to decrease complexity.

  Firstly, load sharing parameters are calculated to minimise blocking probabilities and bandwidth capacities in the network. An objective function measuring the bandwidth consumption of the network is defined for given load sharing values, which reduces to a linear programming task, and a minimum found.

Secondly, the selected load sharing parameters are used to generate a set of routes through the network using the original ELB algorithm, and logical capacities calculated.

Finally, these logical capacities are varied with the goal of finding a local optimum for network revenue, using an algorithm employed in the 'Kellys derivatives' method.

- Fixpoint method [13]

  The aim here is to maximise the total carried traffic in the network (in terms of the number of accepted calls). This is achieved by summing across all the routes the carried traffic on each route, which again is a function of the connection-loss rate on each of the links the route traverse (and the fraction of the VP capacity which the route uses, which is fixed). This is done by determining the VP capacities and blocking probabilities such that the carried traffic is maximised by using Erlangs fixed point equations (unique solution solved by an iterative method, and proven to find a global optimum of network performance). The algorithm assumes calls of equal bandwidth, where only the number of calls are taken into account.

- Dynamic Virtual Path Bandwidth Management [13]

  "Dynamic" is a misnomer since it does not dynamically manage VP bandwidth during network operation. The objective is to maximise the carried traffic of the network by calculating VP capacities, but makes lots of simplifying assumptions: virtual paths are assumed to be end-to-end connections with given poissonian traffic demands, and there no logical routes. The objective function, which calculates a measure of the blocked traffic (again using Erlang's formula), is minimised for given VP bandwidths, using a gradient projection method.

6.5 Step 3: VP routing

The task here is to determine a routing for VPs on top of the physical network topology, given their required bandwidths, subject to the physical constraints such as physical link bandwidths capacities, and network manager constraints (e.g. a maximum number of intermediate nodes for each VP).

6.6 Alternative approach

It is possible to simplify the network model, and consequently reduce the complexity of the pre-dimensioning model. If we only allow one route for every traffic source, and restrict each route to being carried by *end-to-end* dedicated virtual paths, then Step 2 disappears from the equation - route bandwidths map directly to virtual path bandwidths.

7 On-line Network Re-Dimensioning Techniques

All of the discussed algorithms deal primarily with the pre-dimensioning of ATM networks at a VP level. A "static" view of source traffic characterisics is given, and VP bandwidths are calculated based upon these characteristics. However, the traffic characteristics displayed by a network site change and evolve with time (more users at the site, a new service provided to users at the site), therefore traffic source bandwidth requirements are changing. Traffic source bandwidth requirements will diverge from the bandwidth initially allocated to it, allowing for potential performance problems (for both customers and network operators). So while the initial dimensioning of the network may be adequate for a short period of time, without some method of on-line monitoring of network resource usage, network efficiency will deteriorate, and the network will no longer be able to guarantee agreed QoS levels.

An on-line mechanism is needed to monitor the network for any changing conditions and adapt the virtual topology as necessary to maintain network efficiency and agreed QoS levels.

The traffic and performance characteristics along each VP in the network must be monitored to ensure that

- the actual traffic characteristics match the agreed traffic model. It is possible that the initially provided source traffic model was invalid, or that over time, the characteristics of a given traffic source might change (more users, different user requirements). Both will result in different bandwidth requirements than currently allocated to the VP carrying that the traffic for that source. Such changes need to be detected, and the agreed source traffic model and necessary VP bandwidth should be updated accordingly. If not, then connection-level QoS requirements of the traffic source might not be met (theoretically, if CAC and UPC are performing correctly, then cell-level QoS objectives should still be met).
- the allocated bandwidth of a VP is accurate. Even if the monitored traffic source characteristics correspond to the traffic model, the method of translating this model into a bandwidth requirement might be inadequate. If insufficient bandwidth is allocated, then connection-level QoS objectives will again be violated. If too much bandwidth is allocated, then there is a wasteage of resources which could be used more effectively elsewhere.

The network characteristics must be monitored to ensure that

- there is a balanced load on the system (where link load is defined by the amount of bandwidth allocated on each physical link to VPs). If VP bandwidths can be dynamically updated, then the initial "load-balanced" VP configuration might change over time to an unbalanced configuration (some physical links might be relatively unused, while others are heavily used). If the system is deemed to be unbalanced, then a reconfiguration of the VPs in the system needs to be performed in order to restore a balanced load in the system. This is achieved by rerouting VPs in such a manner that the load on links is satisfactorily redistributed.

7.1 Monitoring VP performance

Traffic and network performance characteristics need to be continually monitored in order to check that the actual traffic characteristics for each VP / traffic source match the theoretical traffic model, that the bandwidth allocated is sufficient to guarantee connection and cell-level QoS objectives for that traffic source. Relevant statistics to achieve this are:

- measured call-level traffic characteristics at the VP level
  - Mean number of active connections
  - Mean connection request rate

  - Mean connection acceptance rate
  - Mean call holding time

- measured cell-level QoS values at the VP level
  - mean cell arrival rate
  - mean cell loss rate
  - mean cell delay

VP bandwidth maintenance can be operated as a reactive control or a preventative control. Again, the reactive control is simpler, as it only has to *react* to congestion that occurs, but has the disadvantage that congestion has to occur before it activates. Bandwidth maintenace as a *preventative control* is more difficult, as it has to predict in advance that the VP bandwidth is going to be inadequate, but has the advantage that if it works properly, then VP problems will be detected and avoided before they actually occur. To perform this prediction, the system must be able to extract trends in observed performance statistics, use these to predict their values at some point in the future, and calculate a bandwidth.

The following is a summary of one study which aims to predict future traffic characteristics in "generic" communication networks.

- Chu and Beight [1] aim to approximate the short-term characteristics of the arriving traffic, which then can be used to predict the future arrival characteristics and to assist network control. Neural networks are used because the transient reponse and temporal behavior is normally too complicated to be modelled analytically.

  Estimation must not only approximate the transient behaviour, but also detect the change from the incoming traffic and react accordingly.

  The method uses short-term averaging over time-periods (windows) which can capture time-varying behaviour. A function f estimates (or predicts) the behaviour of the future window by using the information from the current window and the past d windows. A large time slice for these windows can be used to describe "periodic" regularities, and a small slice can be used to describe the transient behaviour.

  Inputs to the backpropagation neural network are the short term average of arrival rates at each window, short-term average of required service time at each window, average number of calls in burst phase at each window, and average cell arriving rate at the window. The output is the required quantity to be estimated for the next window (estimated cell arrival rate for future window). The size of the future window may vary, and can be different from the size of the history windows.

While the type of neural network used here is strictly feed-forward, it can also be classed as a recurrent neural network. Recurrent neural networks are widely used in learning time sequences. The temporal sequence of information in the above study is converted into a spatial pattern to be presented to the input nodes of the neural network. The backpropagaion then can learn and recognise different sequences. These architectures are termed *time-delay neural networks* [4].

7.2 Load balancing

A metric is required which will give an indication of how balanced the load is in the network, and a threshold is required — a level belowabove which the degree of load balance is unacceptable. If this occurs, then the virtual topology must be re-routed.

8 Conclusion

This paper has presented an overview of ATM communication networks and discussed the need for resource management techniques to meet the needs of both the users of the networks and the network managers. These resource management tasks are broken down into various levels, the goals at each level are defined, and various realisations of these goals are discussed.

References

[1] Pong P. Chu and Timothy O. Beight. Estimating Transient Traffic with a Neural Network. *World Congress on Neural Networks*, 4:107–111, 1993.

[2] N. G. Duffield, J.T. Lewis, and Neil O'Connell. The Entropy of an Arrivals Process: A Tool for Estimating QoS Parameters of ATM Traffic. School of Mathematical Sciences, DCU and DIAS.

[3] A. E. Eckberg. B-ISDN/ATM Traffic and Congestion Control. *IEEE Network*, pages 28–37, September 1992.

[4] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Addison Wesley, 1991.

[5] Atsushi Hiramatsu. ATM Communications Network Control by Neural Networks. *IEEE Transactions on Neural Networks*, 1(1):122–130, March 1990.

[6] Duke Hong and Tatsuya Suda. Congestion Control and Prevention in ATM Networks. *IEEE Network*, pages 20–16, July 1991.

[7] ITU-T. *I.371: Traffic Control and Congestion in B-ISDN*, March 1993.

[8] Robert J. T. Morris and Behrokh Samadi. Neural Network Control of Communications Systems. *IEEE Transactions on Neural Networks*, 5(4), July 1994.

[9] Joaquim E. Neves, Luis B. de Almeida, and Mario J. Leitao. ATM Call Control By Neural Networks. *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications*, pages 210–217, 1993.

[10] Ernst Nordstrom, Olle Gallmo, Mats Gustafsson, and Lars Asplund. Statistical Preprocessing for Service Quality Estimation in a Broadband Network. *World Congress on Neural Networks*, 1:295–299, 1993.

[11] Hiroshi Saito. *Teletraffic Technologies in ATM Networks*. Artech House, 1994.

[12] Hiroshi Saito and Kohei Shiomoto. Dynamic Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 9(7):982–989, September 1991.

[13] Aron Szentesi. Comparison of ATM Network Dimensioning Methods. Technical report, HSN Lab, Budapest, 1994.

[14] Ahmed A. Tarraf, Ibrahim W. Habib, and Tarek N. Saadawi. A Novel Neural Network Traffic Enforcement Mechanism for ATM Networks. *IEEE Journal of Selected Areas in Communications*, 12(6):1088–1096, August 1994.

[15] Marek Wernik, Osama Aboul-Magd, and Henry Gilbert. Traffic Management for B-ISDN Services. *IEEE Network*, 6(5):10–19, September 1992.