Scalable optical ATM networks

A dissertation submitted for the degree of Master of Research

Martin Biddiscombe
Department of Electronic and Electrical Engineering
University College London
Torrington Place
London
WC1E 7JE

August 1996

CONTENTS

Page

1 Introduction

This project considers some of the issues relating to large scale ATM (Asynchronous Transfer Mode) networks carrying a multitude of traffic types. Each traffic type can bear differing levels of performance in the areas of cell loss, cell delay and cell delay variation. In fact these different performance requirements are a key distinction between the traffic types. The other primary distinction between traffic types is the variability of their bandwidth requirements. It is both unnecessary and impractical to design a network that would offer uniform performance to all traffic types while guaranteeing the most stringent requirements of all users. If a particular traffic type is insensitive to cell delay variation but is intolerant of cell loss then the network need not be designed to eliminate delay variation *for that traffic type*, it should be optimised for minimising cell loss. On the other hand, a different traffic type may be highly sensitive to cell delay variation but relatively tolerant of cell loss (for example: uncompressed real-time voice or video). A network designed for this latter traffic type would be dissimilar to one optimised for the former.

The key issue facing ATM network design is encapsulated by the example above: how can a network be designed to satisfy the requirements of such diverse traffic types (indeed, traffic types could be conceived with any combination of cell loss and delay variation requirements) and *guarantee* to meet those requirements even in the most strenuous of operational environments? A particular concern relates to the concept of *statistical multiplexing*, where variable bit-rate traffic streams are combined in the knowledge that their combined mean bit-rates are within the system capacity even though their combined peak bit-rates may exceed that level. The problem facing ATM, and other packet technologies, is that of how to cope when the different streams all wish to transmit at their peak rate – exceeding the system capacity. Inevitably this will result in cells or packets being buffered, and eventually lost. How should this be handled, when the individual streams have different tolerances to delay or loss? The concepts considered during this project may offer a solution to this problem.

Two key ideas have been explored: super-symmetric network design and dynamic pricing. The examination of these ideas forms the core of this dissertation. Super-symmetric networks are discussed in chapter 3 and dynamic pricing in chapter 4. The ideas are brought together in chapter 5, which explains how dynamic pricing could be implemented on a hierarchy of super-symmetric networks. Chapter 2, below, investigates the motivation for the project from three perspectives: cell loss in wide area networks, distributed processing and the cost of implementing a global scale broadband network with unrestricted (end-user to end-user) communication patterns. The way in which the concepts of super-symmetric networks and dynamic pricing address these issues is outlined. The summary and conclusions (chapter 6) bring together the forgoing chapters and re-examine the motivations in the light of the material introduced and explained in those chapters.

Research into the concepts is still at an early stage, and the aim of this project has not been the investigation of the networks that could be constructed if the concepts were implemented, or any other *detailed* technical aspects that could arise from them. Instead, the aim has been to consider whether or not the concepts are likely to provide a solution to the problem of guaranteeing user quality of service parameters – such as cell loss rate and maximum cell delay variation – in a network capable of carrying a variety of traffic types and subject to the possibility of overload caused by user demand for bandwidth exceeding either the network's or an individual link's transmission capacity, or the network switching potential.

2 Project motivation

This chapter explains the motivation behind the project. The two ideas central to the project: super-symmetric network design and dynamic pricing have been suggested as possible solutions to two of ATM's biggest problems – cell loss and cell delay variation. These concepts are at an early stage in their development and as yet it is not possible to prove whether or not they will solve the problems. However, none of the work carried out to date has disproved their suitability. Indeed the work has strengthened the arguments in their favour, and provided further motivation for investigating the ideas. This chapter identifies the limitations of current network designs or protocols with respect to three drivers. The concepts of dynamic pricing and super-symmetric networks address these limitations. An indication is given of how these limitations could be overcome by the implementation of super-symmetric networks and dynamic pricing: these will be revisited and amplified in subsequent chapters.

As I have already indicated, cell loss and delay in wide area networks are serious problems facing ATM. The primary motivation for this project comes from a desire to eliminate these problems. Dynamic pricing is suggested as a bandwidth management scheme, to be implemented on super-symmetric networks, which have desirable properties in terms of routing choices and dynamic re-routing possibilities. Other benefits would flow from the implementation of these ideas. In particular, super-symmetric networks would be well adapted for use in distributed processing systems, and dynamic pricing provides a means of giving network users a choice of how to react to network overload. These three motives are presented and developed below.

2.1 Cell loss and delay in wide area networks

2.1.2 *Loss and delay*

There are several proposed ATM adaptation layers (AALs), which provide a means of transporting circuit-switched or packet-switched type data across an ATM network. There are also well established methods of implementing both circuit- and packet-switched networks. However a network that is optimised for circuit-switched applications such as telephony is not necessarily optimised for packet-switched applications such as data networks. Circuit-switched networks are ideal for carrying constant bandwidth or bit-rate traffic – such as plain old telephony – and guarantee uniform delay. Whilst such a network could be used to transfer data, it would be wasteful of bandwidth due to the non-constant bit-rate (bursty) nature of data traffic. Packet-switched networks on the other hand are ideal for carrying highly variable bit-rate traffic – offering statistical multiplexing gains in efficiency – but offer no delay or delay variation guarantees. This is due to the buffering at nodes necessary to prevent packet loss during periods of overload. Such periods are considered inevitable with packet networks due to the statistical nature of packet arrival distributions at nodes (switches, multiplexers, etc.).

Delay variation can be tolerated by certain traffic types (in essence, those which are not dependent on real-time user interaction or conversation). For example, file transfers do not require uniform inter-packet delay so long as the packets all arrive. Since it is possible for packet networks to send consecutive packets by different routes, it is even possible for packets to arrive in a different order to that in which they were sent. File transfer protocols can cope with this (within reason), and can reassemble the data from the packets into their original order. However, the user-perceived quality of real-time or conversational services is degraded by packet delay variation. Data generated at the source must be encoded and transferred to the destination to be decoded and passed to the receiver. If the data packets do not arrive at the same rate as they were transmitted, the decoding process is left with either too much or too little data from moment to moment. For example, software has recently become available that allows people with Internet access and suitable computer hardware to hold telephone conversations across the Internet. The speaker's software codes the speech and the resulting data stream is packetised. The packets are routed across the Internet as usual, and the listener receives the packets. Their software extracts the encoded speech from the packets and decodes it, replaying it to them. However, the large buffers in the Internet result in consecutive packets suffering delays different by up to several seconds. This packet delay variation results in abrupt periods of silence during the replay, making the speaker sound jerky and interrupted to the listener.

During conversation, if the listening party wishes to speak, they usually wait for the person speaking to pause. If these cues are distorted by the network, conversations can become awkward.

In essence, ATM is a packet technology, which can be used to implement *virtual* circuits. (In general packet networks allow variable length packets, and the packets contain addressing information. ATM has fixed length cells, containing routing information (virtual circuit and virtual path identifiers). So, ATM networks can be considered as a special case of packet networks.) However these virtual circuits can suffer from delay variation as much as any other packet based network. In ATM this is known as Cell Delay Variation (CDV). The CDV experienced on ATM networks designed with delay sensitive traffic in mind should not be as great as that on networks – like the Internet – which are primarily data networks. It is possible to reduce CDV by reducing buffer sizes across the network, and the CDV experienced by certain types of cells can be reduced by implementing a priority flag. If such a system is implemented, cells with the priority flag set take precedence over those without in any buffers they encounter. However, neither of these methods remove CDV altogether. In order to eliminate CDV completely it is apparent that there are only two practical solutions:

(i) impose delay (at the destination) on the first cell of a stream greater than or equal to the largest possible cell delay variation, and buffer incoming cells, processing them at regular intervals – restoring uniform inter-cell delay;

(ii) eliminate buffers from the core of the network, guaranteeing uniform inter-cell delay.

The first of these solutions, by definition, imposes extra delay on the majority of cells. This is *also* undesirable for real-time applications, particularly for those involving conversation across the network (such as telephony or video-conferencing). The concepts examined during this project could provide a means of implementing the second solution without incurring any cell losses: the buffering that is required to handle temporary overloads is located at the periphery of the network – possibly within the users' applications – and the network itself operates without buffers. This allows the possibility of constructing an entirely optical network – including optical multiplexing and switching of signals.

2.1.2 *Uncertainty*

Uncertainty is one of the greatest problems facing ATM. Buffer dimensioning relies explicitly on the statistical nature of users' cell generation. It is thus impossible to eliminate cell loss completely for at least two reasons:

(i) the buffers are dimensioned for an expected cell arrival distribution – there may well be times when the distribution differs significantly from that which was expected;

(ii) even if the cell arrival distribution fits the statistical model, there will be times when the buffer overflows, however infrequent.

Thus there remains a possibility of cells being lost due to buffer overflow. The statistical analysis may indicate that the probability of this occurring at a particular buffer is very low for a given buffer size, but it remains desirable to design a system that can guarantee zero cell loss probability.

Connection admission control (CAC) algorithms, in conjunction with usage parameter control (UPC) algorithms, should prevent congestion for well-behaved CBR and VBR sources. If there are insufficient network resources to sustain a new call given its requested cell-rate parameters (such as peak cell rate and sustainable or mean cell rate) the call will be rejected. VBR source description is a complex problem – questions still open to debate include how many and what parameters should be specified, and how accurately a VBR source can be expected to specify the parameters in advance. Hence, in the case of VBR traffic, the complexity of CAC algorithm design is an important factor in determining whether or not the new source can be adequately supported without adversely impacting either its own requested quality of service, or that guaranteed to existing calls.

If the CAC algorithm is too simplistic, the network resources allocated to a VBR source may not match its need. If the resources allocated are insufficient, cells will be lost. If superfluous resources are allocated, bandwidth will be wasted. A source is well-behaved if its actual cell output parameters match those it offered to the CAC algorithm. An ill-behaved source will not necessarily be detected by the UPC – typically a "leaky bucket" algorithm – since the algorithm must not penalise a user for network-induced jitter. The size of the bucket determines the maximum amount of jitter that will be ignored, the rate at which the bucket leaks determines the permitted mean cell rate. Since an ill-behaved source's output does not match its stated parameters, it is likely that the resources allocated will not match its requirements.

There are therefore clearly areas of uncertainty regarding whether VBR sources can be sustained by allocated network resources, and hence whether predicted cell loss rates for given buffer sizes, etc., will be achieved.

Services using the other traffic categories that have been defined by the ATM forum – UBR (unspecified bit rate) and ABR (available bit rate) – are intended for non-delay-sensitive applications that can vary their output cell-rate at the direction of the network. ABR sources may request a minimum cell rate (MCR), which will be processed by the CAC as for CBR and VBR call setups. However users of ABR sources will clearly hope to be able to transmit cells at a higher rate than their specified minimum cell rate (otherwise, they would have requested a CBR link – at the MCR). While sufficient VBR sources are transmitting below their peak rates, there will be spare bandwidth that can be assigned to UBR and ABR sources. However, changing network demands, and increases in the output rates of VBR sources can reduce the bandwidth available for UBR and ABR sources rapidly. These sources must be controlled by feedback indicating the congestion status of the network, reducing their rates during periods of increased CBR and VBR activity.

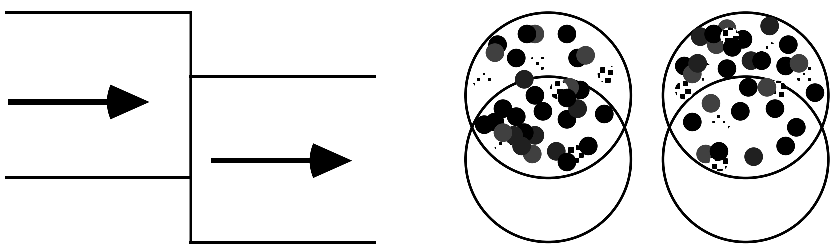
It is expected that eventually the majority of data applications will use the ABR or possibly UBR services. Currently, data applications generally communicate over packet networks where a common protocol used for data transfer is IP (Internet Protocol) or TCP/IP (Transmission Control Protocol/Internet Protocol). IP is responsible for transferring data packets from node to node and TCP verifies the correct delivery of data, providing error detection – initiating retransmission if packets are lost or corrupted. Any new network design, transportation method, or management technique must be compatible with existing (legacy) services and methods, since it is likely that there will be a period during which they will have to interwork. Legacy applications must remain unaware of the transition from one technology to another. ATM must therefore be able to cope with IP packets if it is going to be used to transport data for applications that currently rely on TCP/IP. An IP packet can be thousands of bytes long. It must therefore be segmented into consecutive ATM cells. If a single ATM cell is lost, the whole packet is corrupted, and must be retransmitted. This problem could conceivably be addressed by an error correction scheme. However, the strength of the scheme required to enable recovery from a single lost ATM cell – 47 bytes of user data if AAL5 is used – would impose significant overhead, particularly on shorter packets. It would also be necessary to operate the scheme at the interface between the ATM network and any network being used to route the packets further to their destination, since the packets would have to be returned to their native format for onward transmission. This would both require significant intelligence at the interface and impose further delay on the packets.

Cell loss in any ATM network is most likely to occur when the network is congested. If an IP packet is corrupted, it must be retransmitted – a single lost cell resulting in up to 100 extra cells (the whole packet) being resent (assuming a maximum of 5 Kbytes per packet). It is apparent that there is an increased likelihood of cell loss for the retransmitted packet than the overall network average cell loss probability, since congestion conditions may well have continued. It is conceivable that this could lead to a vicious circle of packet loss – a single cell lost incurring substantially increased cell transmission, producing further cell loss, resulting in additional increased cell transmissions. Testing a proposed solution to this problem, it was found that with no congestion control a sustained switch overload of just 4% resulted in 83% of 200-byte AAL5 PDUs requiring retransmission. Even after the proposed congestion control mechanism was implemented, the number of PDUs requiring retransmission had only reduced to 17% [Drury].

The likelihood of cells being lost may be extremely low, but if it remains at all, users are likely to remain wary of ATM for loss sensitive applications. Two analogies may provide an insight into this aspect of user behaviour. The first shows how network operators react when presented with two technologies that provide the same service: one considerably more demanding technically but without susceptibility to a particular problem. Even though considerable effort was devoted to showing that the likelihood of the occurrence of the problem was very small, once a new technology was developed that eliminated the problem, the old technology was rapidly superseded. The second analogy shows how end-users react to a problem with a reportedly very low likelihood of occurrence, if it is known that there is a way of eliminating the problem.

Multi-mode fibre vs. single-mode fibre for optical communication [Epworth]
As the demand for bandwidth and communication quality increased during the 1970s, focusing on fibre-optic transmission, a problem was discovered: modal noise. In a multi-mode fibre, light travels along many paths. Each path is of a different length and may have slightly different propagation characteristics from the other paths. If monochromatic light is passed

through a multi-mode fibre, a cross-section through the fibre will reveal a “speckle” pattern resulting from constructive and destructive interference between the rays travelling along each path since the phase of each ray at a given point depends on the path length it has travelled. The problem arises when joints in the fibre are misaligned, since the speckle pattern changes according to the physical conformation of the fibre and the wavelength of the light. Thus if the fibre is disturbed, however slightly, the speckle pattern can change dramatically – significantly altering the amount of light passing through the joint. Similarly, slight frequency changes in the light source cause significant changes in the pattern. Such frequency changes occur naturally as a result of diode laser modulation: as the laser power is varied according to the signal, the temperature of the laser cavity varies and so its length changes – changing the wavelength of the emitted light. These and other carrier induced frequency modulations at the source result in random amplitude modulations at the receiver. Figure 2.1 illustrates the effect of speckle at a misaligned joint in the fibre. As the fibre moves and the wavelength of the transmitted light varies, the speckle pattern changes, resulting in an effectively random variation in the amplitude of the received signal, sufficient to cause occasional bit errors.



Fibre core misalignment

Cross sections at different instants indicating effect of varying random speckle on portion of light coupled

Figure 2.1

Much effort was devoted to predicting the bit error rate resulting from modal noise, but the wide range of causes – degree of misalignment, exact fibre conformation, laser characteristics and so on – which themselves were not reproducible for experimental purposes, meant that the accuracy of prediction could not be guaranteed. However this form of modal noise can be eliminated by the use of single-mode fibre. It was initially believed that single-mode fibre transmission was technologically unfeasible due to the exacting tolerances required in joint construction, etc. However, once it was achieved, its advantages over multi-mode fibre were such that it rapidly became the dominant technology.

Pentium processor flaw [Sharangpani & Barton]

The Intel Pentium™ processor was originally released in early 1993. During 1994 it came to light that there was a flaw in the processor: certain values had been omitted from a look-up table used by the numeric processor for floating point division. Extensive research by Intel indicated that most users would encounter the flaw no more than once per 27,000 years, and that even then the impact of the flaw would be slight – affecting between the fourth and fifteenth significant digit of the result. Even users performing large numbers of floating point divisions would not expect to encounter the flaw more often than once in every 900 years: again the flaw would have an impact on between the fourth and fifteenth significant digit of the result of the division. However, news about the flaw spread rapidly resulting in significant user concern regarding the performance reliability of their computers, and eventually Intel was forced to offer replacement (corrected) chips to all end users who requested them.

If a method of managing cell flow in ATM networks can be devised which retains the bandwidth gains resulting from statistical multiplexing, but which can guarantee both zero cell loss and uniform cell delay for those users who require them, then, for the same reasons that Intel were compelled to offer replacement processors and that single-mode fibre transmission became the technology of choice, ATM network operators will be under pressure to implement such a method. The concepts investigated for this project will, if the assumptions they depend upon are shown to be justified, enable such a method to be implemented.

2.1.3 *Re-routing*

A related issue is that of dynamic re-routing of cell-streams. In general with today's networks, different routes between locations are of different lengths, and hence time delays. This means that should a particular link become congested it is not possible to re-route the calls in progress on that link without changing the propagation time between source and destination. However, with an unbuffered super-symmetric network design, the propagation time between nodes is uniform across the network. Should a particular link become congested there is no reason why calls in progress should not be re-routed away from the link. For instance, consider the network fragment shown below in figure 2.2. Each link (**AB**, **AC**, **BD** and **CD**) is the same length and data travel only in the directions indicated.

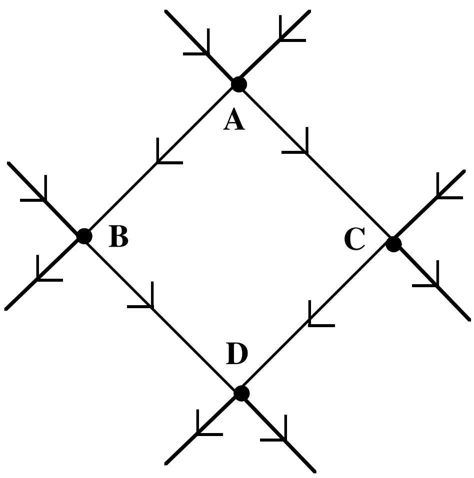


Figure 2.2

A call from **A** to **D** could be routed via **B** or **C**. Suppose it were routed via **B** – traversing links **AB** and **BD**. If, during the course of the call, link **BD** were to become congested – perhaps with calls from **B** to **D** which have no alternative route – the call in progress from **A** to **D** could be diverted via **C** without risk of interruption or data loss.

The super-symmetric network design enables such re-routing to be implemented. The dynamic pricing principles can be used to determine when such re-routing should be done. In the simple example above, a large volume of traffic from **B** to **D** would result in an increase of the price of the link **BD**. This, fed back to **A**, would result in the call from **A** to **D** being re-routed if the links **AC** and **CD** were cheaper (less congested) than **AB** and **BD**.

2.2 Distributed processing

A second motive for implementing a system based on bufferless super-symmetric networks is the potential application of such a system to distributed processing. In order for an operation to be distributed across several processing centres, it is essential that each processor knows the order in which data were input or instructions were issued. A simple example demonstrates the importance of such timing information:

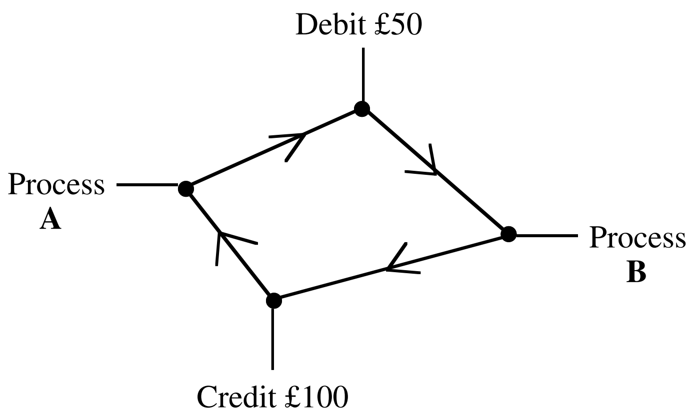


Figure 2.3

Suppose a bank accounting system is implemented at two sites: **A** and **B**, and that a customer's account currently contains £25, and that the bank levies a £10 charge if an account goes overdrawn. If two instructions arrive nearly simultaneously, as indicated on the diagram, the accounting system at **A** will receive the credit instruction before the debit instruction, and the customer's account will contain £75 after both instructions have been implemented. However at site **B**, the debit instruction will be received *before* the credit, resulting in the customer going overdrawn temporarily. As a result, the record of the customer's account at site **B** (after both instructions have been implemented) will differ from that at site **A**, indicating that it contains £65.

On a simple ring network, it is straightforward to correct this deficiency, as a node can be designated to "time-stamp" empty cells that pass it. Instructions and data can then be inserted into time-stamped cells. Processors compare time-stamps on incoming cells to determine the order of processing. In the above example, if the processor at **A** was the time controller, the cell into which "Credit £100" was inserted would have passed **A** before that into which "Debit £50" was inserted. Consequently, even though the system at **B** received the instruction to credit the account after the instruction to debit the account, by comparing the time-stamps of the cells carrying the instructions it would know that the credit instruction was to be implemented first.

However, as network topologies become more complex it becomes much harder to implement such a time-stamping system or other methods of coping with the timing issues. This results in a practical limitation on the size and complexity of distributed processors.

However, if the processors were connected by a super-symmetric network, the timing relations between events would be preserved by the network topology. The regular network structure and uniform link length provide this benefit. Figure 2.4 below indicates how a "wave" of time passes across the network:

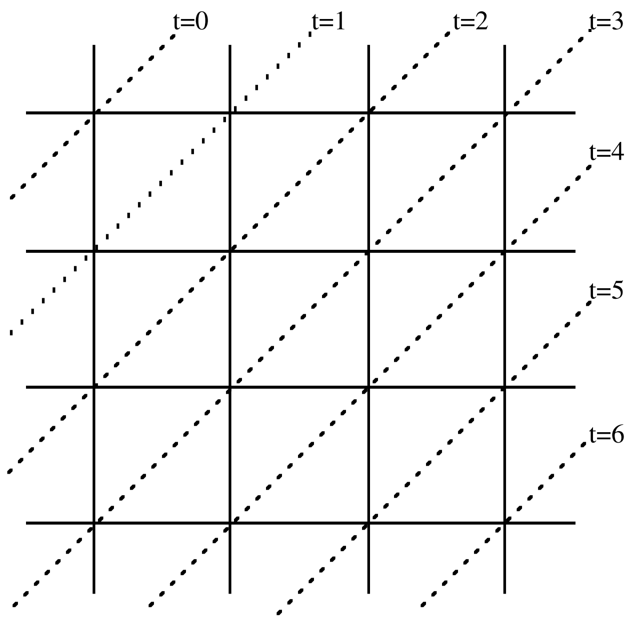


Figure 2.4

The structure of a super-symmetric network is explained in sections 3.1 and 3.2 below, and the potential application of super-symmetric networks to distributed processing is outlined in section 3.4

2.3 User choice

Today's Internet suffers from considerable delay at periods of peak usage. This can be particularly frustrating for those users who are attempting to use the Internet for work purposes and are being impeded by those who are just "surfing the net". The Internet is a collection of packet networks, and as such suffers from the problems of delay and loss common to all packet networks when overloaded. Current packet networks impose delay and, ultimately, packet loss on users when the network becomes overloaded. No discrimination is made between users as to who suffers the loss – it is spread uniformly over all incoming packets at the point(s) of overload. Users are required to pay for the overload they are subjecting the network to by suffering delay or packet loss.

The same considerations will apply to ATM networks offering variable bandwidth services to their users. When the network becomes overloaded, buffers will fill – increasing the delay and cell loss probability for all users. Protocols such as ABR (Available Bit-Rate) are intended to cope with this overload by signalling to users that they should reduce their bandwidth. However, this is also a non-discriminatory system: when the network becomes crowded, all ABR users have to decrease their cell rate until the network overload is relieved.

The dynamic pricing concepts offer users a third choice. Instead of paying for overload by suffering delay or loss, users can choose to maintain their current service levels (bandwidth, delay guarantees and maximum loss rates) by increasing the amount they are paying for the service. When the network becomes busy, those users who are sending data, or are involved in a high image quality (high bandwidth) video-conference, have a choice that would not be available with today's packet networks or existing ATM protocols. The increase in demand is signalled to the users as an increase in price for their current service levels. The user chooses whether to accept the increase in price and maintain their current bandwidth, or whether to maintain their current charges but reduce their bandwidth – increasing the length of time required for a file transfer or reducing the image quality on a real time video link.

2.4 Summary

The novel concepts of super-symmetric networks and dynamic pricing could prove beneficial in many areas and for many reasons. This chapter has provided an insight into some of the areas where the ideas could be applied, and suggested why the concepts could become invaluable tools in those areas. Chapters 3, 4 and 5 examine the concepts – individually and together – in more detail, giving further insight into their potential for application, particularly to the predicted future for telecommunications: global multimedia broadband information and communication.

3 Super-symmetric networks

Common network topologies include ring, mesh or partial mesh and star based layouts. A typical network will consist of a combination of such configurations: for instance a mesh or partial mesh in the core of the network with nodes at the periphery attached in a star-like layout. Other possibilities include nodes or sub-nets linked to backbones. As indicated previously, such networks frequently allow routing choices between nodes. However, the different routes are generally of different lengths and, in the case of packet networks, encounter different numbers of buffers. Hence the propagation time of different routes is different and, for packet networks, varies – even for the same route from moment to moment.

Routing decisions in packet networks are made packet by packet, which is one source of packet delay variation. In contrast in circuit switched networks, a pre-determined route is assigned to a circuit when it is set up and is not altered during the course of the call if at all possible. Route selection is based on historical measures of link utilisation and link reliability as well as current availability. The system is unable to react to sudden changes in local demand, as has been occasionally dramatically demonstrated in the telephone network. On such occasions, management intervention is required to protect the affected area from further overload.

Consideration of super-symmetric networks allows the possibility of considering dynamic re-routing of calls in progress with no change in propagation delay. Routing decisions can be based on the information provided by dynamic pricing.

3.1 Description

A three-dimensional object may be considered to be regular if its faces are identical and its edges are all the same length. Each vertex joins the same number of edges. It is straightforward to demonstrate that there are five such solids, based on square, triangular and pentagonal faces, known as the Platonic solids. The Platonic solid with the greatest number of vertices is the dodecahedron, which has twenty. A uniform network based on a dodecahedral arrangement of nodes cannot be scaled beyond the twenty nodes – a considerable limitation. Another factor to be considered is the number of edges meeting at each vertex – the links meeting at a node in a communication network. A dodecahedron has three edges per vertex. Assuming initially that each link could carry traffic in one direction only, it would never be possible to balance the loads on each link on a dodecahedral network. A condition that must be satisfied if the links are to be balanced is that each node must have an even number of links. These would have to be divided equally between incoming and outgoing traffic. The only Platonic solid that has an even number of edges per vertex is the octahedron – with four.

However the octahedron has only six vertices – an even greater restriction on the scale of the network. The table below summarises the properties of all five Platonic solids.

Solid	Number of vertices	Edges per vertex	Number of faces	Shape of face (edges per face)
Tetrahedron	4	3	4	Triangle (3)
Octahedron	6	4	8	Triangle (3)
Cube	8	3	6	Square (4)
Icosahedron	12	5	20	Triangle (3)
Dodecahedron	20	3	12	Pentagon (5)

No other three-dimensional polyhedra can be constructed from identical faces. For example, the "Buckyball" which has sixty vertices with three edges per vertex and thirty-two faces consists of pentagons and hexagons.

However, in four or more dimensions there are many more symmetric polyhedra. It is possible to project these super-symmetric objects into three dimensions. This process disturbs some of the objects' properties – such as equality of lengths, areas or enclosed volumes. This is equivalent to the distortion of lengths and areas when three-dimensional objects are projected into two dimensions. However, under projection, connected vertices remain connected and adjacent edges (those which meet at a common vertex) and faces (those sharing a common edge) remain adjacent, and the common vertex or edge of the projected object is the projection of the common vertex or edge in the original (four-dimensional) object.

Thus a network can be considered whose topology is that of a higher dimensional super-symmetric object – by projection. Equality of link lengths can be restored with padding as needed. The exact technical details of the methods by which this could be achieved in the "real world" are unimportant for the purpose of this project. Issues that would have to be addressed if such a network were to be implemented include determining the mismatch between installed link lengths and choosing a method that compensates for those differences. Possible methods include adding extra fibre to delay the signal if the mismatch is slight, inserting a high capacity electronic buffer if the delay mismatch is considerable (although this conflicts with one of the initial aims of such a network, which was that it should be possible to implement it all-optically), or feeding the delay mismatch information back to the user and requiring that the user varies their transmission timing to match the routing delay differences (this method may have a detrimental impact on the potential use of the network for distributed processing applications). The third method requires that the user's application should be aware of the route its transmitted data is taking and of any impending routing changes. All of these methods require an accurate method of measuring the mismatch in link lengths – a challenge in itself.

The particular network design that has been considered during this project is based on the four-dimensional hypercube, which has sixteen vertices, each connected to four others. Conceptually, a hypercube could be constructed from eight cubes in the same way that a cube is constructed by folding together six squares. The classical projection of a hypercube is shown in figure 3.1.

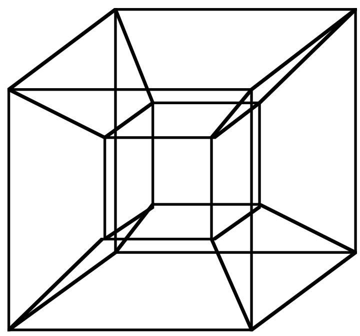


Figure 3.1

It is important to remember that all eight cubes are the same size and all thirty-two edges are the same length, and the four edges meeting at a given vertex are mutually perpendicular.

In order to consider constructing such a network, the hypercube must be first projected into three dimensions. The resulting object will also have sixteen vertices, each connected to four others, with four-sided faces. Such an object is a sixteen-node torus. By considering the projection above (figure 3.1), it is possible to imagine how a wire-frame hypercube could be stretched and manipulated to resemble a wire-frame torus, and vice versa. The resulting torus would resemble the sketch shown as figure 3.2.

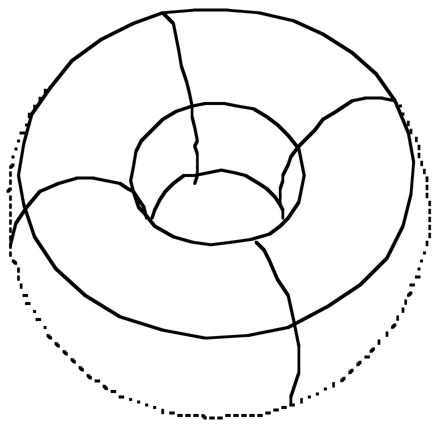


Figure 3.2

3.2 Routing

The hypercube forms the basis for all networks considered during this project. The principles derived can be extended to larger networks, or to hierarchies of hypercube networks – where each node of the network is connected to a smaller hypercube network. This provides scalability possibilities. The first situation to be considered was that of a single hypercube. In order to appreciate the underlying symmetry of the network and the impact of that symmetry on a practical realisation of the network, the routing options and traffic distribution on a hypercube network were considered. The ideas presented here do not have a direct impact on the discussion in Chapter 5 on the use of super-symmetric networks in conjunction with dynamic pricing, but this does not rule out the possibility of a link being developed. This section is therefore largely self-contained, but it was working through the principles of routing on a hypercube/torus based network that provided an insight into the problems that will have to be solved, and how they might be, in order to implement such a network and manage the routing on it.

In order to consider the routing across a hypercube network, it is redrawn yet again. It is simple to see how a torus can be cut to produce a cylinder, which in turn can be unrolled to leave a square plane. One of the first tasks undertaken was to use this device whilst deriving formulae for the number of routes between any pair of nodes on arbitrarily sized toruses. The hypercube network becomes a straightforward square grid after these manipulations, but with the added consideration that nodes on opposite sides of the grid are in fact adjacent and connected by links of the same length as any other pair of adjacent nodes. This also indicates how a larger scale super-symmetric network could be constructed: simply scale up the number of nodes in the square grid.

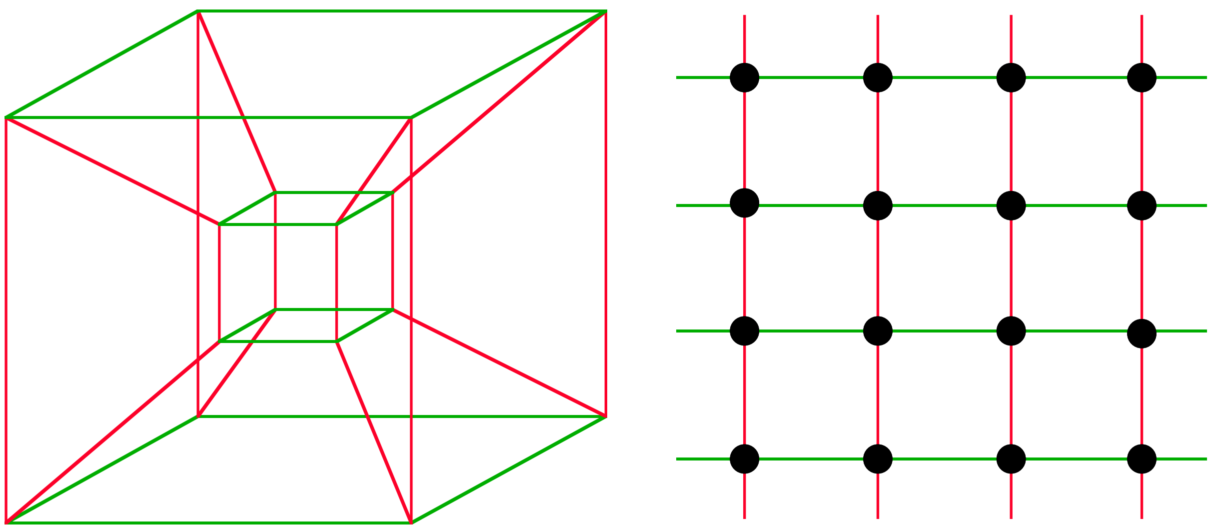


Figure 3.3 Unfolded sixteen node hypercube based network
showing correspondence between edges in hypercube and links in grid

The properties of grid networks have been investigated thoroughly [Vestmar]. It is a straightforward matter to determine the number of routes between arbitrary nodes on a grid network. The situation is complicated by the wrap-around nature of the grid derived from a hypercube. However the same combinatorial techniques can be applied to determine routing options on a hypercube as a plane grid. In combination with assumptions about communities of interest these results can be used to estimate traffic distributions across the network.

The network can be considered to consist of "1-way" or "2-way" links. For consistency and maintenance of symmetry, the links in the "1-way" case are oriented so that each node has two incoming ports and two outgoing ports, which are the same as each other node. Regarding the network as a set of interlocking rings, the "1-way" case is equivalent to the situation where the rings are unidirectional, and are oriented such that when the network is unfolded and considered as a grid, all the "horizontal" rings point the same way, and all the "vertical" rings point the same way.

In order to calculate the number of routes between nodes it is necessary to apply a numbering or labelling scheme to the nodes. An intuitive and suitable scheme is to label each node with an (x,y) co-ordinate according to its location in the grid. The symmetry of the network is such that any node can be selected as the origin, (0,0) without loss of generality.

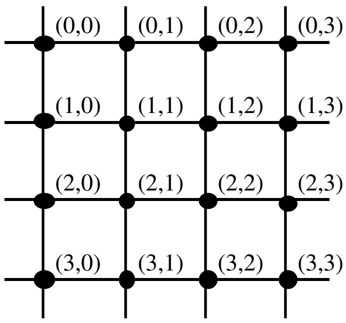


Figure 3.4 Unfolded hypercube with co-ordinate labels

On an arbitrary, non-wrap-around, grid network, the shortest routes between (0,0) and (x,y) traverse $x+y$ links. An arbitrary route will traverse $x+y+2n$ links, where the $2n$ are extra links made up of n links deviating from the shortest route and n links returning from the deviation. Thus, on an arbitrary, 2-way, grid network, the set of second shortest routes between (0,0) and (x,y) traverse $x+y+2$ links. Any deviation from the from a shortest route is made by altering the first co-ordinate by 1 (a vertical deviation) or the second co-ordinate by 1 (a horizontal deviation). Each detour must be made up either of a horizontal link and a reverse horizontal link, or a vertical link and reverse vertical link. So, on any given route of length $x+y+2n$, there will be a horizontal detour links and b vertical detour links such that $a+b=n$. The total number of routes on an arbitrary, 2-way grid network between (0,0) and (x,y) for a maximum detour length of n is given by $\sum_{a=0}^{n}\frac{(x+y+2n)!}{a!(n-a)!(x+a)!(y+n-a)!}$ which can be rewritten as $\binom{x+y+2n}{x+n}\cdot\binom{x+y+2n}{n}$ (for $n\neq 0$), where $\binom{a}{b}=\frac{a!}{b!(a-b)!}$. If n=0 the number of routes is given by $\binom{x+y}{x}=\frac{(x+y)!}{x!\,y!}$, which is the total number of *shortest* routes on either a 2-way (non wrap-around) grid or a 1-way grid with or without wrap-around.

Considering the grid representation of a hypercube network, the situation is complicated by the possibility of wrapping round the grid. However, the formula for the number of shortest routes on a 1-way network is unchanged. For example the numbers of shortest routes between nodes on a 1-way 16-node hypercube labelled as in figure 3.4 is shown in the table below:

Destination address		x co-ordinate			
		0	1	2	3
y co-ordinate	0		1	1	1
	1	1	2	3	4
	2	1	3	6	10
	3	1	4	10	20

Routes from (0,0) to (x,y) on a 1-way 16-node hypercube

If the hypercube has 2-way links, the basic method for calculating numbers of routes between nodes remains unchanged, but the resulting formula is somewhat different since there are sets of alternate routes if either the x or y (or both) co-ordinates of the destination are offset from the source by exactly half the length of the ring. However, the number of routes can be derived from the 1-way case by realising that (in the case of a 16-node hypercube) the number of routes is doubled when x=2 and when y=2, and that the table entries are symmetric about x=2 and y=2. In section 3.3 I shall show how a much simpler method can be used by considering an alternative node labelling scheme.

Destination address		x co-ordinate			
		0	1	2	3
y co-ordinate	0		1	1×2	1
	1	1	2	3×2	2
	2	2×1	2×3	2×6×2	2×3
	3	1	2	3×2	2

Routes from (0,0) to (x,y) on a 2-way 16-node hypercube

If the nodes are sources and sinks of data and it is assumed that there is a global community of interest, then other things being equal, it would be appropriate to divide the traffic from one node to another evenly among the available routes, which in turn prompts the question: how much of the traffic arriving at a node is intending to terminate at that node? Answering this question involves several steps. The formulae developed for calculating the number of routes from (0,0) to (x,y) are readily extended to calculate the number of routes between nodes that either pass through or avoid another given node. For example, the number of routes from (0,0) to (x,y) via an intermediate node (u,v) is: $\frac{(u+v)!}{u!v!} \cdot \frac{(x-u+y-v)!}{(x-u)!(y-v)!}$ (where $0 \le u \le x$ and $0 \le v \le y$). This is simply the number of routes from (0,0) to (u,v) multiplied by the number of routes from (u,v) to (x,y). Using these principles to calculate the amount of traffic passing a given node involves calculating the number of routes between all possible pairs of source and destination nodes that could route their traffic via the node in question, and scaling the routes by their traffic density.

Hypercube with 1-way routing

Using the 16-node hypercube with 1-way routing as our example again, we can calculate the amount of traffic routed via (0,0) and the amount of traffic terminating at (0,0). Consider figure 3.5 below. The 16-node grid is shown duplicated four times in order to demonstrate how traffic may route across the network via (0,0).

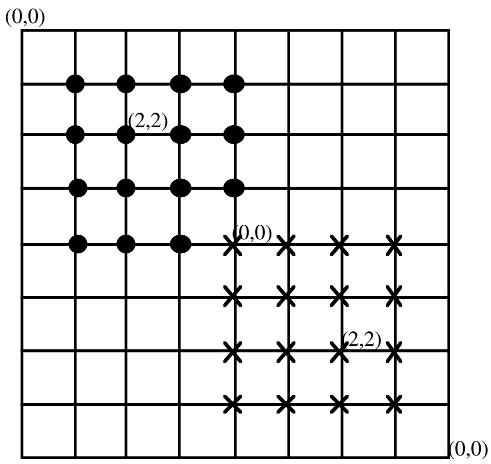


Figure 3.5

The symmetry of the network guarantees that the results obtained for one node will be the same as those for any other node – so it is sufficient to determine the routing and traffic loads for just one of the nodes. Traffic originates at any of the nodes marked • and terminates at the nodes marked ×, bearing in mind that only shortest routes are considered – so for example traffic from (2,2) to (3,3) would not be routed via (0,0), however traffic from (3,3) to (2,2) could route via (0,0). In general, a route from (x,y) to (u,v) could only be routed via (0,0) if

$x>u$ and $y>v$ (where a source $(0,y)$ or $(x,0)$ is written as $(4,y)$ or $(x,4)$ respectively) There are fifteen possible source nodes (all except (4,4)) that involve routing and sixteen possible destinations. Calculating the number of *routes* via (0,0) is a straightforward matter of applying the equations already derived (remembering that the labels (a,b), $(a+4,b)$, $(a,b+4)$ and $(a+4,b+4)$ all refer to the same node). There are $\frac{(u+4-x+v+4-y)!}{(u+4-x)!(v+4-y)!}$ routes from (x,y) to (u,v) of which $\frac{(4-x+4-y)!}{(4-x)!(4-y)!}\cdot\frac{(u+v)!}{u!v!}$ go via (0,0). However, calculating the amount of *traffic* using these principles is somewhat more complicated, since each route must be weighted by the amount of traffic on it relative to other routes. If it has been assumed that traffic is divided equally between the available routes, this is equivalent to calculating the proportion of routes via (0,0) of all the routes between source and destination. The large number of routes involved, even on a small network such as the sixteen node hypercube, make it impractical to write down the complete set of equations that would have to be evaluated to calculate the total traffic passing a given node – for the 16-node network there are 256 pairs of nodes to consider. However, using a spreadsheet to total up the traffic values per route is not a problem. Each possible source and destination pair is tabulated, and the number of routes between them calculated. If the source/destination pair satisfy the condition for (0,0) to lie on (at least) one of the shortest routes between them, the formula for the number of routes via (0,0) can be applied. (In the case of a 16-node network, only 100 of the 256 pairs satisfy the condition for (0,0) to lie between them.) Hence the proportion of routes from source to destination that pass the specified node (0,0) can be calculated – both for individual source/destination pairs and for all pairs.

For example, there are ten routes between the nodes labelled (2,2) and (0,1). Of these, six pass through (0,0). Therefore, 60% of the traffic from (2,2) to (0,1) can be expected to be routed via (0,0). On the other hand, there are three routes between (0,1) and (2,2) but none pass through (0,0). So, none of the traffic from the node labelled (0,1) to that labelled (2,2) would be seen at node (0,0).

If the amount of traffic between nodes is normalised to 1 unit per time period (i.e. assuming uniform global community of interest), then the total amount of traffic flowing on the network is 256 units per time period (16 possible sources to each of 16 possible destinations). The amount of traffic between sources that *could* route via (0,0) amounts to 100 units, and of that 64 units are actually routed via (0,0). Of those 64 units, 16 terminate at (0,0). So, a quarter of the traffic routed via a given node terminates at that node.

Hypercube with 2-way routing

The same principles can be applied to determine the routing choices and implications for traffic distribution on a hypercube network with 2-way routing. As explained previously, the underlying formulae for calculating the numbers of routes remain unchanged, but additional routes are available if the source and destination nodes are offset from one another by half the length of the rings in either direction. Also, rather than routing to a node *beyond* the halfway point, a shorter route will be found by routing from that node back to the source node – since links carry traffic in both directions in this case.

Given the cyclic nature of the network and the possibility of sending data either way along the links, there are up to four possible directions that could be taken from source to destination:

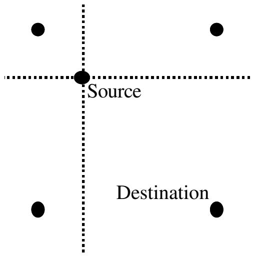


Figure 3.6

One, two or four of these directions will include shortest routes. In figure 3.6 above – which indicates the four possible representations of a destination node for a particular source/destination combination – it is apparent that the shortest routes would only be located in the upper-left quadrant with respect to the source node. In order to analyse the routing options for all four quadrants, it is necessary to standardise on a default configuration for the source/destination nodes. This can be achieved by relabelling the destination node. If the source is (x,y) and one label of the destination is (u,v) then

$$\left.\begin{array}{l}\text{If }\begin{cases} x\le u & \text{let } U=u-4 \\ x>u & \text{let } U=u \end{cases} \\ \text{If }\begin{cases} y\le v & \text{let } V=v-4 \\ y>v & \text{let } V=v \end{cases}\end{array}\right\} \text{guaranteeing that } U < x \text{ and } V < y.$$

Then the routes into the four quadrants can be examined by considering routes between

(x, y) and $\begin{cases}(U,V)\\(U+4,V)\\(U,V+4)\\(U+4,V+4)\end{cases}$ and asking whether any of the routes in each quadrant pass through (0,0)

(or its equivalents $(4m,4n)$ where m,n are integers), bearing in mind that we are only interested in routes of least length. The length of route in each quadrant is given by $\left|(x_{\text{co-ord}})_{end} - (x_{\text{co-ord}})_{start}\right| + \left|(y_{\text{co-ord}})_{end} - (y_{\text{co-ord}})_{start}\right|$. For an arbitrary pair of points (a,b) and (c,d), the rectangle they define (which contains all the shortest routes between (a,b) and (c,d)) contains (0,0) or one of its equivalents if $c > a$ and (c mod 4 < a mod 4 or a mod 4 = 0) and if $d > b$ and (d mod 4 < b or b mod 4 = 0).

If these conditions are satisfied, then the values of m and n which define the equivalent of (0,0): $(4m,4n)$, are $m=(c - c \bmod 4)/4$ and similarly $n=(d - d \bmod 4)/4$.

Applying these principles and using the equations for calculating numbers of routes as before, makes possible the calculation of the amount of traffic expected to pass an arbitrary node, as compared to the amount terminating at the node. As in the case where the network is configured with 1-way links, the amount of traffic between pairs of nodes is normalised to 1 unit per time period. The total traffic on the network is thus 256 units. In this case, the total traffic between sources that could route via (0,0) is reduced to 81 units, and the total number of units *actually* routed via (0,0) is reduced to 48. Again, the amount of traffic terminating at each node is 16 units per time period. So, a third of the traffic routed via (0,0) is terminating traffic.

Observation

Given the assumptions of a uniformly global community of interest and an even traffic distribution over the available routes, it can be seen that there are differences in the complexity of switches required if 1-way or 2-way networks are to be implemented. If a 1-way network is implemented, the switches need fewer "positions": traffic on the incoming links can pass straight through, be turned on to the perpendicular outgoing link, or be terminated (the switch is located at the destination node). If a 2-way network is implemented, extra switch "positions" are required that switch the traffic onto the other perpendicular link. Figures 3.7 and 3.8 indicate the increased complexity of switching in the 2-way case. (Terminating traffic has been omitted for simplicity.)

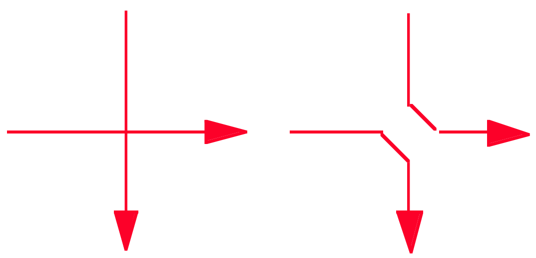


Figure 3.7 Ring to ring switching options in a one-way network

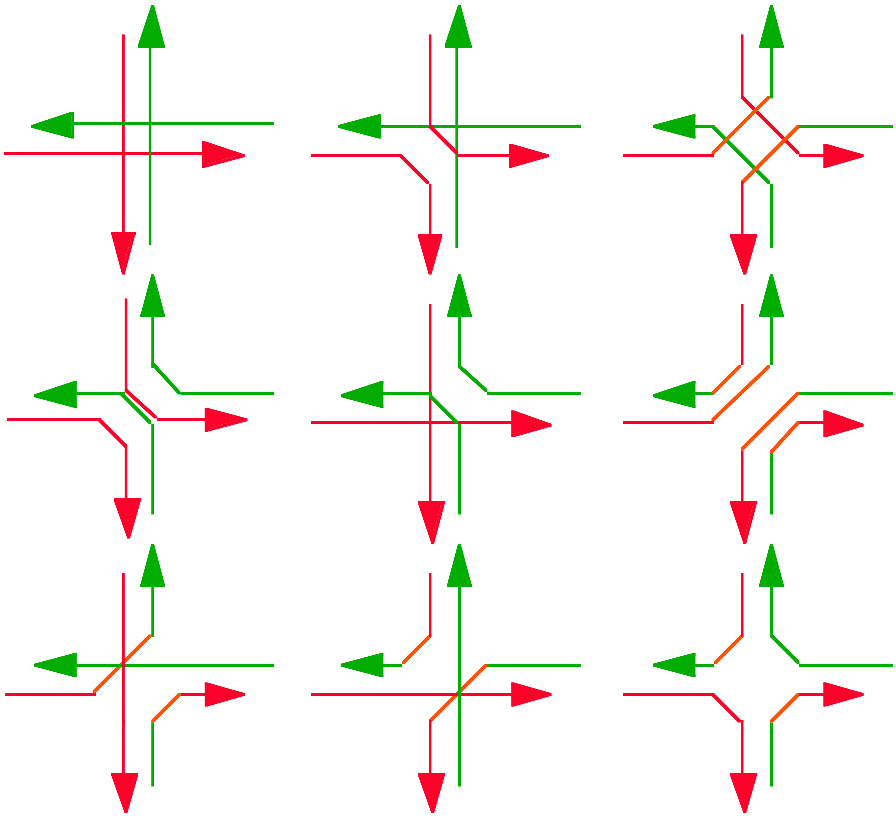


Figure 3.8 Ring to ring switching options in a two-way network

However, the increased switch complexity in a 2-way network is balanced by a reduction in load. For a given volume of traffic, the switches in a 2-way network handle 81% of the traffic handled by switches in a 1-way network.

3.3 Alternative labelling scheme

An alternative labelling scheme for the nodes of a hypercube network is closely related to the 4-spatial co-ordinates of the corners of the unit hypercube. The co-ordinates of the unit hypercube range from (0,0,0,0) to (1,1,1,1). The nodes of the network are described by the binary number *xyzt* where the corners of the hypercube have co-ordinates (x,y,z,t). Figures 3.9 and 3.10 indicate how these labels are derived and how they translate to the familiar two-dimensional representation.

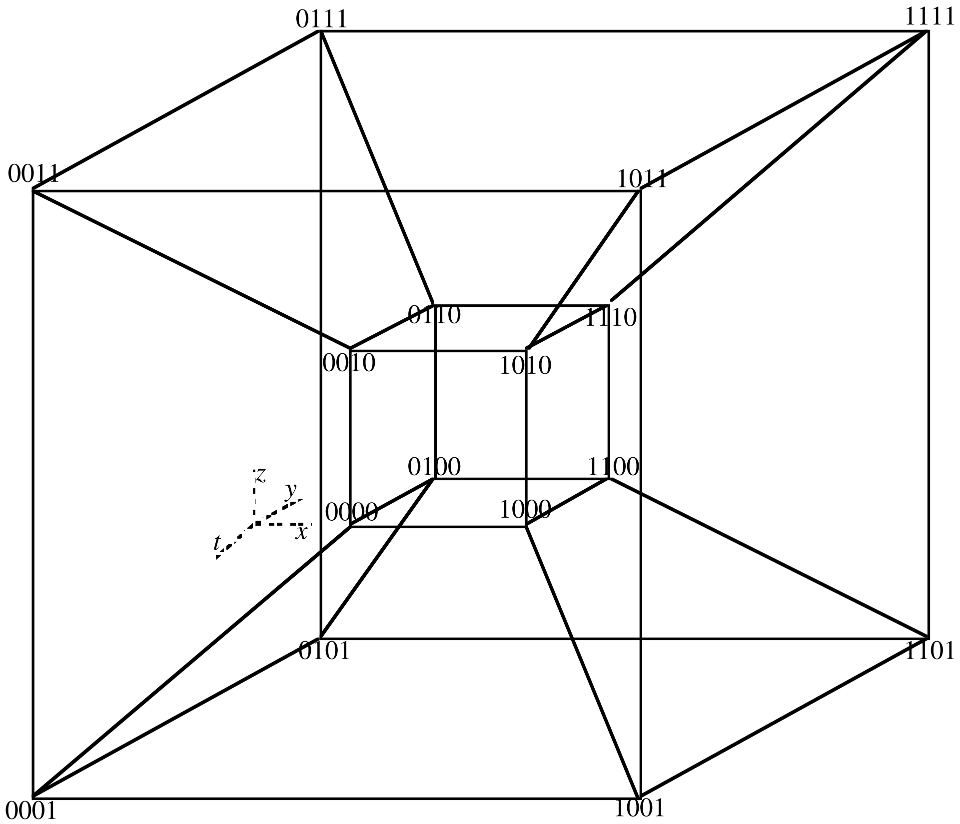


Figure 3.9 Labels derived from unit hypercube

With the grid representation labelled in this manner, the binary labels of adjacent nodes differ in only one bit position, and no two nodes have the same label. Any shortest route between arbitrary pairs of nodes can be found by successively reducing the distance between the current position and destination. This can be achieved by routing to a node whose Hamming distance from the destination is less than that of the current node. On a 16-node network with 2-way links, the number of such routes is simply the factorial of the Hamming distance between source and destination nodes. It is straightforward to show why this is the case.

Any node adjacent to the current node will have a Hamming distance to the destination node of either one more or one less than the current node, since each of the four adjacent nodes' labels differ in exactly one bit position from that of the current node. Therefore, if the Hamming distance from the current node to the destination is n, any one of n bits can be changed to reduce the Hamming distance to n-1. The same reasoning can be applied to each step on the route between source and destination. Hence, the total number of shortest routes between arbitrary nodes is the factorial of the Hamming distance between the labels of the nodes.

For example, the number of routes, on a 2-way network, between 0000 and 1011 is 3!, or 6, since the Hamming distance between the nodes labelled 0000 and 1011 is 3. Under the previous labelling scheme, these nodes could be represented as (0,0) and (2,1). Using the equations derived with that system, the number of routes between them can also be evaluated as 6 since $2 \cdot \frac{(2+1)!}{2! \cdot 1!} = 6$. Notice how much simpler the new expression is in the case of a 16-node network. However, if the network is based on a larger grid the expression cannot be used, since the maximum number of links from the source node that are first steps in a shortest route is always 4 – the number of links adjacent to the node – whereas the Hamming distance from source to destination could be more than 4.

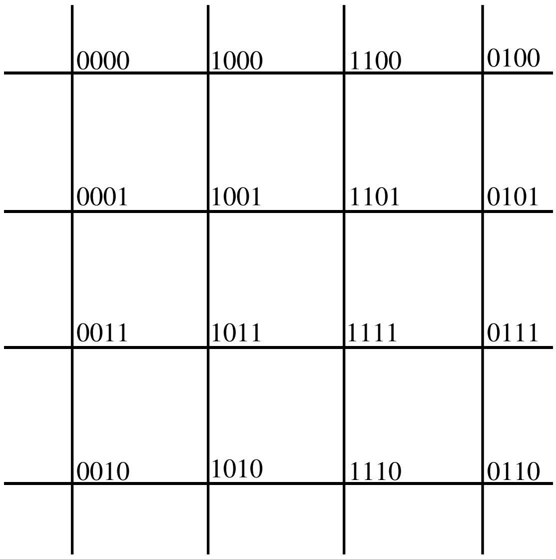


Figure 3.10 Translation of labels from hypercube to grid representation

3.4 Application to distributed processing

Chapter 2, the project motivation, indicated the scalability problem of networks used to carry the data associated with distributed processing, and that super-symmetric networks provide a potential solution to the problem. Having outlined the structure of super-symmetric networks, I am now in a position to be able to provide a further insight into how such networks could address the issue of maintaining timing relations between events.

Super-symmetric networks, as described, have uniform inter-node propagation times and a regular structure. Figure 2.4, which is repeated below, indicates how a "wave" of time can be considered to pass across the network (the diagram is shown for the case of a 1-way network). Data injected onto the network at the top left node (as indicated on the diagram), will reach either of the adjacent nodes at the same time – t=1 – where the unit of time is how long it takes to propagate along a single link. Similarly the data would reach any one of four nodes at time t=3, regardless of the route taken to reach those nodes.

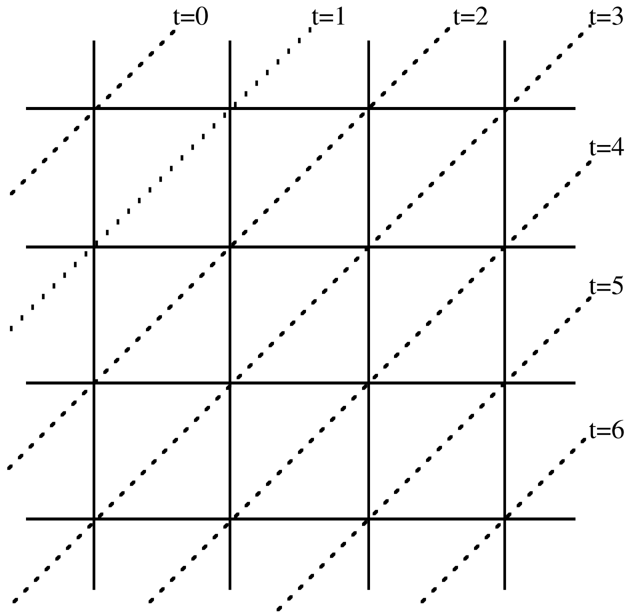


Figure 3.11

In order for a processor located at a node to be able to determine a unique timing relationship between two incoming data packets, all it needs to know is the address (location) of the node that originally transmitted the data. This tells the processor how far away each data source is, and thus how long it took for the data to propagate across the network. This in turn enables it to determine the order in which data packets (which could of course contain data to process or instructions to follow) from separate sources were generated, and thus the order in which they should be processed.

3.5 Summary

Super-symmetric networks, with their uniform inter-nodal propagation delays, have many useful properties. Their simple structure makes them ideal candidates for being implemented without buffering, and being controlled from the periphery, where any necessary buffers would be located. This would allow the routes within the network to be dynamically optimised, according to the traffic distribution offered to the network. However, the control mechanism for such a network must be able to guarantee prevention of overload at any point in the unbuffered core of the network, and respond to increasing traffic levels by balancing the traffic load across the network as far as possible and providing sufficient feedback to the traffic sources to prevent overload from occurring. In chapter 4, such a control mechanism – dynamic pricing – is introduced and discussed. The potential for its implementation on super-symmetric networks is considered in chapter 5.

4 Dynamic Pricing

In chapter 3 I described the concepts surrounding super-symmetric networks, and some of the routing implications that can be drawn, based on assumptions about the users' communities of interest. The second idea at the core of the project is that of *dynamic pricing*. In this chapter I shall discuss the desirability of implementing a dynamic pricing scheme, and then go on to describe the early ideas as to how such a scheme could be implemented and the drawbacks of the envisaged implementation. Further ideas were pursued, resulting in significant progress in the case where network load is not varying (the static case). Realising these ideas in the dynamic situation is a considerably harder task, but the concepts and direction of current research can be outlined. These are linked to the implementation of dynamic pricing on super-symmetric networks, which is described further in chapter 5.

4.1 Motivation

There are times when it would be very nice if it were possible to "jump the queue" when using the Internet. However the Internet being a conglomeration of typical packet networks operates on a "best effort" basis. Quality of service (in terms of packet delay or loss) is not guaranteed, and there is only one level available – which is offered to all users irrespective of the urgency of their need to use the system or the value they place on the data they are transferring.

Many services do not usually require rapid transfer of data (for example: e-mail). Others do not require a constant packet inter-arrival time (for example: file transfer). On the other hand, there is an increasing use of the Internet for real-time services ("Internet phone" services, etc.), whose packets require both rapid transfer and uniform inter-arrival time. ATM networks must address the issues that these conflicts raise if they are to offer guaranteed performance levels successfully to a variety of users.

Buffer dimensioning in ATM networks is a complex problem, involving a trade-off between preparing for potential bursts of cells above the local capacity and maintaining a low overall delay and minimum cell delay variation for delay sensitive services. Regardless of the size of buffers employed (a network topology has been suggested in the previous chapter that could be implemented optically – without buffers) access to network resources must be managed and shared among the users in a fair and responsible manner. This is not a serious problem when the demand is less than the supply. However, as bandwidth requirements increase beyond availability – either in the short term (bursts) or medium term (period of exceptional, unforeseen demand) – the situation must be managed. Connection admission control (CAC) algorithms are intended to reduce the likelihood of such periods. However, if any statistical multiplexing gains are to be realised, the potential combined peak cell rate of the calls in progress must exceed the system capacity.

Current methods – primarily ABR – rely on reducing the rate of all users. As a simple example, if the bandwidth demanded by all users of a 2Mbits^{-1} capacity system totalled (for a short while) to 4Mbits^{-1}, all users would be required to reduce their cell rates to 50% of their current levels in order that cells should not be lost due to buffer overflow. When the combined rate has decreased below the system capacity, users may be permitted to start increasing their rates once more.

Unfortunately, this system *also* treats users indiscriminately – regardless of the relative importance of their communication. Dynamic pricing – which responds to the level of congestion in the network – is a way of managing the available bandwidth, while maintaining maximum data throughput for those users who value their data sufficiently. As demand increases beyond capacity, the price increases. Services that are the least delay sensitive are throttled back first. If the overload continues, the price increase will prompt those users on a tight budget (whether “real money” or company provided credits on an intranet) to reduce their rate – by for instance using a lower bit-rate (and thus reduced quality) speech coder – or to stop altogether. Continuing overload results in more and more users reducing their rates or terminating their calls/data transfers. The users who place the highest value on their communication (those with urgent calls to make, or senior managers with more “credits”) are able to maintain their data rates for much longer.

The net effect of a dynamic pricing system will be to maintain the cell rate within the system capacity, but to maximise the accumulated value of data transferred.

4.2 Early ideas

The first mechanism considered for allocating bandwidth to competing users was based on the concept of an auction. Each user submits a bid, consisting of the amount of bandwidth they require at a given time and the upper limit on the amount they are willing to pay per unit bandwidth. Since the network structure in mind is based on a set of interlocking rings, it seemed that the cycle time for a ring was a natural choice for the frequency at which auctions should be held, although it is possible to auction access to several cycles at once. The auctioneer monitors the bids from the users. When all bids for a given cycle have been received, the auctioneer sorts the bids by willingness to pay, and starting from the user willing to pay the most per unit bandwidth, selects users to admit until no more could be accommodated. These users are informed of their success in the auction, and allowed to transmit their data. They are not all charged at the amount they indicated as their upper limit in their bid, but at the price per unit bandwidth of the last user admitted. This is technically similar to a Vickrey auction [MacKie-Mason & Varian]. The following example shows how the method would work.

Suppose a system has an availability of 20 units. Six users wish to compete for those units. Each user has a different requirement and different willingness to pay for units and hence a different maximum amount they will pay. These amounts are shown in the table below. The users submit their bids, and are reordered in terms of their willingness to pay per unit. User 4 is the most willing, and so is the first to be guaranteed access. As successive users are admitted, the running total of requirements reserved is kept. When further users' demands would exceed the availability, the process stops. In this case, user 2 was refused access. The last user to be selected was user 6, with a willingness to pay of $2 per unit. Hence all users are charged $2 per unit. All users, except user 6, make a nominal profit – since they are charged less than the maximum they would have been willing to pay – and the total value (requirement × willingness) transmitted across the network is maximised. The table below demonstrates the process of selecting users according to their willingness to pay.

User	Require	Willing	Max	**R**	User	Require	Willing	Σ req	Sent?	Charge	Profit
1	2	$5	$10	**e**	4	3	$8	3	Yes	$6	$18
2	7	$1	$7	**o**	1	2	$5	5	Yes	$4	$6
3	1	$3	$3	**r**	5	7	$4	12	Yes	$14	$14
4	3	$8	$24	**d**	3	1	$3	13	Yes	$2	$1
5	7	$4	$28	**e**	6	5	$2	18	Yes	$10	$0
6	5	$2	$10	**r**	2	7	$1	25	No		

The total value transmitted, when users are selected according to their willingness to pay per unit bandwidth, is the maximum possible – $75 – as can be seen from the table below. The table shows the remaining demand for units, and the value of the units that would be transmitted if successive users were refused access. The total offered value is $82. The selection of users resulting in optimum utilisation of the resource – user 6 refused access – does not, in this case, result in optimum value transfer.

Since the price charged for access depends on the willingness to pay of the last user admitted, rather than a flat rate, there is no incentive for users to *claim* a higher willingness than their true willingness to pay. Stating a higher willingness will not provide the user with a better service, unless the user *is* the last to be admitted, when they will be charged according to their *stated* willingness. In that case, the user gets the service but pays the penalty by being charged more than they were *really* willing to pay.

User(s) refused	Remaining requirement	Sufficient available?	Value transmitted
1	23	No	
2	18	Yes	\$75
3	24	No	
4	22	No	
5	18	Yes	\$54
6	20	Yes	\$72
1,4	20	Yes	\$48

This method could be revised – at the expense of increase complexity – to allow users to specify a range of willingness/requirement parameters. For instance, a user may need to make an urgent call that must get through, however it is not essential in this case that the video component be transmitted. The user could specify a separate willingness for each component, or else a scale of values – very high for the minimum voice requirements, but decreasing with increased bandwidth. Thus the user will be as sure as possible of their voice component getting through, but under different network conditions may also be able to transmit a low, medium or high quality video component (according to the output rates of the codecs that are available to them and the size of the image they transmit).

The primary advantage of the method is that since resources are allocated to each user as they demand them, and the users' allocations vary according to their needs, the users can be monitored closely for adherence to their contract – the network can stipulate exactly which cell-slots they should be using – and since users are only permitted to transmit if resources have been allocated for their exclusive use, the *network* will never overload. Overload is handled by the auction process: once the bids have been processed, and resources allocated, the network *knows* who will be transmitting how much, and that the total transmitted will be within its capacity.

The key drawbacks of this method of implementing dynamic pricing are the time it takes to negotiate access to the resources, and the amount of network resources devoted to implementing the auction – particularly if access to each ring cycle is auctioned individually. The situation becomes more complex if the network is not simply a single ring, but a set of interlocking rings – such as the hypercube network – but the principle can be extended, with the network auction controller allocating resources on a link-by-link basis. The optimisation procedure evidently becomes significantly more complicated since each call in progress routes across a selection of links, and if a user is unable to connect to their destination by any route, the links to which they would have gained access are freed up for other users' traffic.

In the case of a simple ring network it takes at least two complete cycles of the ring to allocate resources. On the first cycle the nodes submit their bids. These are processed, and responses sent on the next cycle. Users may then transmit their data at the prescribed time. If network access is allocated cycle-by-cycle, then bids must be submitted every cycle, processed every cycle and responses sent every cycle. If the simple access/no access method is implemented, the amount of information contained in a bid is relatively small – the total bandwidth required and the user's value per unit bandwidth. However, the result of implementing such a method would be that a call in progress would be suddenly terminated or interrupted when the demand on the system was such that the price exceeded the user's willingness to pay. This is clearly undesirable. The enhanced auction eliminates this problem as users specify a range of bandwidths, and if the demand increases a call in progress may reduce its bandwidth offered to the network – for example by reducing the output bit-rate of a video or voice codec – rather than by losing access altogether. Of course, the information contained in the bids for the enhanced auction is significantly more than that contained in the basic bids – a range of bandwidth/value pairs, or a function describing the user preferences must be sent. The bids themselves place demands on the network resources which must be permanently reserved, and processing the bids requires significant rapid computing power.

4.3 Progress (static case)

It becomes much harder to determine the effects of a flow control mechanism on the network when the offered load is varying, since the inherent fluctuations in the load are combined with the effects of the flow control mechanism. However, a better appreciation of the situation where the offered load does not vary – the static case – should assist in understanding the more complex dynamic case.

The second pricing method considered involves the network determining a price threshold per unit bandwidth for a given network loading, without considering the value individual users attribute to their offered load. The price is advertised across the network and those users who are willing to pay the price may transmit their data.

4.3.1 *Assumptions*

In order to analyse the effect of a flow control system it is necessary to know relevant properties of the offered traffic. Understanding a system based solely on the level of demand and the number of users requires only knowledge about the expected level of demand and how many users are connected. If the flow control method is based on the price of access, then it is also necessary to know about the price users are willing to pay, and how that price relates to their offered traffic.

Since no information is available on how users would *actually behave* if a dynamic pricing system were implemented, it is necessary to make certain assumptions regarding the distribution of offered bandwidth and value. Most point-to-point communication currently requires only limited bandwidth, and it is to be expected that this will not change significantly with the introduction of higher bandwidth services. For most purposes a low bandwidth channel will remain adequate – many conversations can be carried out perfectly well using only a voice channel, or by plain text e-mails. It can also be argued that much of the traffic in a communication network is of relatively low value, and this is also likely to remain true. If a large user base multimedia network becomes a reality it is easy to imagine that much of today's "junk mail" will be transferred to, or duplicated on such a network. This is inherently of low value, especially to most recipients! It is also apparent that bandwidth need not be related to value. It is possible to have any combination of low to high bandwidth with low to high value – as demonstrated in the table below.

Example services	Bandwidth required	
	Low	High
Low value	unimportant e-mails	multimedia WWW browsing
High value	phone call to emergency services	video conference to discuss business deal

Given these observations it seemed reasonable to assume that an individual's offered bandwidth and their perceived value for successful communication are independent. It was also assumed that value and bandwidth could be treated as random variables with semi-Gaussian distributions:

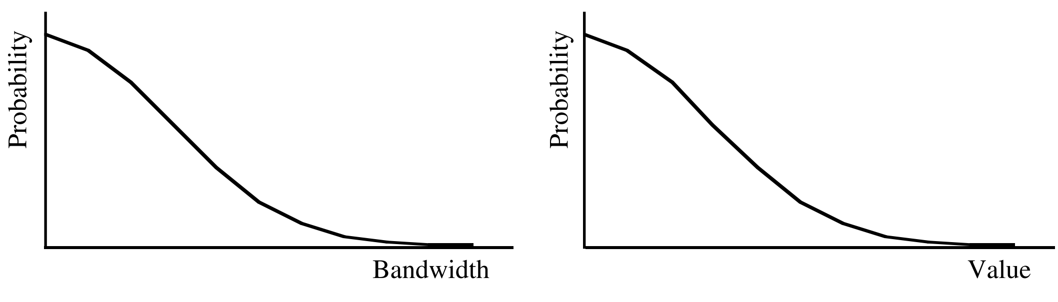


Figure 4.1

With these assumptions, the probability of a user requiring a high bandwidth connection is low. The probability of a user's willingness to pay being high is also low – most calls or data transfers are low value. Note that in the table above I have cited making a call to the emergency services as an example of a high value application. This does not mean that such calls should be subject to high charges, rather that the network should treat such calls as though they had the maximum possible willingness to pay.

Every call has associated with it a bandwidth and value – using these semi-Gaussian distributions it is expected that the majority of connections will be low bandwidth, low value, and that there will be very few high value, high bandwidth calls. This is consistent with the description of services above and the estimated value of such services (there will be many more unimportant e-mails sent than high value video conferences set up).

4.3.2 *Discoveries*

If a pricing scheme is implemented on a network, it is evident that since there is a finite supply of money – either real money or company-provided usage credits – then a sufficient increase in price will reduce the network load. Users may temporarily accept an increase in price without altering their usage patterns, but if the increase is large or sustained it would impose too great a drain on the users' money and they would back off. For instance, a user may be willing to spend £50 per month for their combined telephony/network access and use. During periods of light to normal loading, calls may be charged at £1 per 50 Mbytes transferred (equivalent to approximately £0.01 per minute at 64kbits^{-1} – British Telecom's current off peak rate for local telephone calls). A 2Mbyte file transfer would cost £0.04 – a small proportion of the user's budget. However, during periods of heavy use, the price may increase to £1 per Mbyte. The same 2Mbyte file transfer would now cost £2 – which is a significant proportion of the user's monthly budget. Clearly different users have different tolerances to such price variations. However, given a large user base, it seems reasonable that most users would have a low rather than high budget, and would therefore respond quickly to price increases. Again, this is consistent with the assumption that the distribution of value across calls is Gaussian, with most calls being low value – only a few users (those with a lot of money or urgent needs) are willing to absorb significant price increases without changing their use of the network.

In general, only those users whose value per unit bandwidth (that they wish to transmit) exceeds a network imposed threshold – which will be related to the level of utilisation or degree of congestion – will transmit their data.

The effect of changing the value per unit bandwidth threshold is demonstrated in the graph below (figure 4.2).

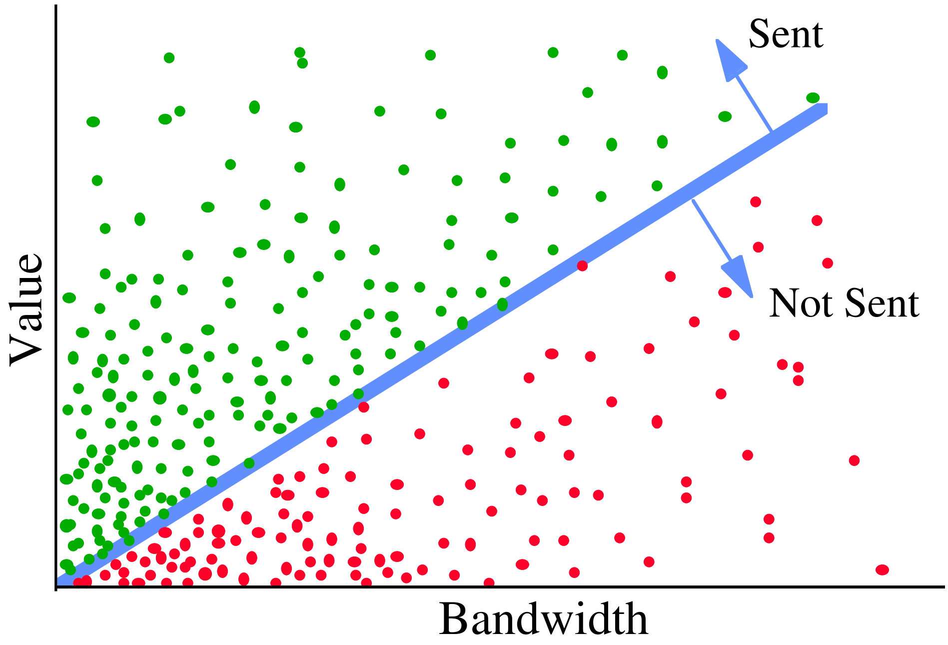


Figure 4.2

The slope of the line is the minimum value per unit bandwidth that the network will admit. All users' offered traffic is represented by a point whose co-ordinates are the bandwidth and value of the traffic. All points lie either below the line or above (or on) it. Those users whose offered traffic lies below the line are denied access at their current proposed bandwidth/value combination. As the threshold increases from zero, the first users to terminate their calls or reduce their bandwidth are those using high bandwidth but whose data is low value (i.e. is inherently of low value, or is not urgent and could be sent when the network is less busy). As the minimum willingness-to-pay increases further, the users affected are those with successively lower bandwidth requirements or increasing value. As can be seen, a significant number of low value, low bandwidth users will have to terminate their calls (or reduce their bandwidth) before a small number of high value, high bandwidth users are affected, since there are many more calls with low value and bandwidth than with either high bandwidth or high value.

Given the assumption that value and bandwidth have independent Gaussian distributions, it is possible to derive an analytic expression for the amount of bandwidth transmitted as compared to the amount offered (i.e. how much would be transmitted if the threshold were zero) as a function of the price. Hence it is also possible to determine the required minimum willingness-to-pay (from now on referred to as a price) that will result in a given level of resource utilisation.

Suppose a user (i) wishes to transfer data across the network with bandwidth b_i and associated value v_i. If there are N such users, the total (offered) bandwidth will simply be $\sum_{i=1}^{N} b_i$. The *mean* bandwidth is thus $\sum_{i=1}^{N} \frac{b_i}{N}$.

In terms of the probability of a user's bandwidth being b: $\mathrm{P}(b_i = b) = \alpha e^{-\beta b^2}$ for some α, β, (α,β are related such that $\int_0^\infty \alpha e^{-\beta b^2} db = 1$[1]) the mean bandwidth, for large numbers of users, tends to $\int_0^\infty b\alpha e^{-\beta b^2} db$, which is $\frac{1}{\sqrt{\beta\pi}}$ (see Appendix 1 for evaluation). Similarly, the probability of a user's value assigned to the call being v is: $\mathrm{P}(v_i = v) = \alpha e^{-\beta v^2}$ (α,β are the same for the two distributions: bandwidth and value are scaled so that the numerical values of their distributions match).

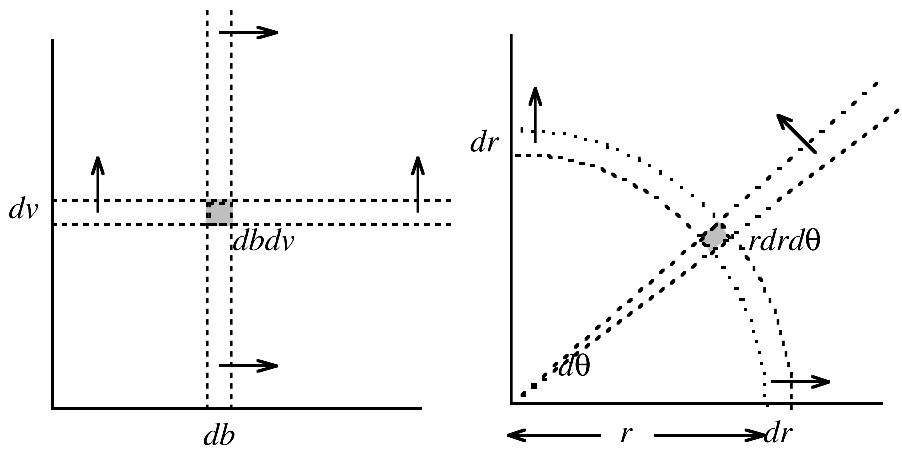
The equation for the mean bandwidth can be rewritten to take account of the different values users place on their calls, in terms of the joint probability of a user's bandwidth demand and their value, giving $\int_{b=0}^{\infty} b\mathrm{P}(b_i = b) \int_{v=0}^{\infty} \mathrm{P}(v_i = v) dvdb$ (since $\int_{v=0}^{\infty} P(v_i = v) dv = 1$ by definition). This is expanded to $\int_0^\infty b\alpha e^{-\beta b^2} \int_0^\infty \alpha e^{-\beta v^2} dvdb$ or equivalently $\int_0^\infty \int_0^\infty b\alpha^2 e^{-\beta(b^2+v^2)} dvdb$.

Integration over the plane defined by the rectangular co-ordinates (b,v) can be replaced by integration over the plane described by polar co-ordinates (r,θ). The change is shown diagramatically below in figure 4.3. The element of area $dbdv$ over which the probability distribution is integrated when the rectangular co-ordinates are used becomes $rdrd\theta$ when the integration is done with respect to the polar co-ordinates.

---

[1] Remembering that $\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \mathrm{erf}(x)$ then

$$\int_{b=0}^{\infty} \alpha e^{-\beta b^2} db = 1 \Rightarrow \frac{\sqrt{\pi}}{2} \frac{\alpha}{\sqrt{\beta}} = 1 \Rightarrow \alpha = 2\sqrt{\frac{\beta}{\pi}}$$



Rectangular co-ordinates   Polar co-ordinates

Figure 4.3

The bandwidth, b, and value, v, are transformed to polar co-ordinates r and θ, by the pair of equations: $r^2 = b^2 + v^2$ and $\theta = \tan^{-1}\left(\frac{v}{b}\right)$, or equivalently, $b = r\cos\theta$ and $v = r\sin\theta$.

Integration over the entire quadrant when described with rectangular co-ordinates requires b and v to vary between 0 and ∞. If polar co-ordinates are used instead, r must be allowed to vary between 0 and ∞, and θ between 0 and $\frac{\pi}{2}$, in order to integrate over the whole quadrant.

The equation for the mean offered bandwidth thus becomes:

$$\int_0^\infty\int_0^\infty b\alpha^2 e^{-\beta(b^2+v^2)}dvdb = \int_0^\infty\int_0^{\frac{\pi}{2}} r(\cos\theta)\alpha^2 e^{-\beta r^2} rd\theta dr$$

by the change of variable and co-ordinate system.

This in turn can be simplified to: $\alpha^2\int_0^\infty r^2 e^{-\beta r^2}dr\int_0^{\frac{\pi}{2}}\cos\theta d\theta$. Of course, $\int_0^{\pi/2}\cos\theta d\theta = \left[\sin\theta\right]_0^{\frac{\pi}{2}} = 1$ and so the mean (offered) bandwidth is given by $\alpha^2\int_0^\infty r^2 e^{-\beta r^2}dr$. As expected, this also evaluates to $\frac{1}{\sqrt{\beta\pi}}$ (see Appendix 1 for proof).

It is the transformation to polar co-ordinates that enables us to calculate how much of the offered bandwidth is actually sent, if a price is set for access. Figure 4.2 provides the clue as to how this is done. The price is defined as the slope of the cut-off line. However, using polar co-ordinates, the line can simply be defined by an angle – Θ. All users whose bandwidth-value combinations transform from b_i, v_i into r_i, θ_i where $\theta_i \geq \Theta$ are admitted, and those whose $\theta_i < \Theta$ are denied access. Figure 4.4 demonstrates this relationship.

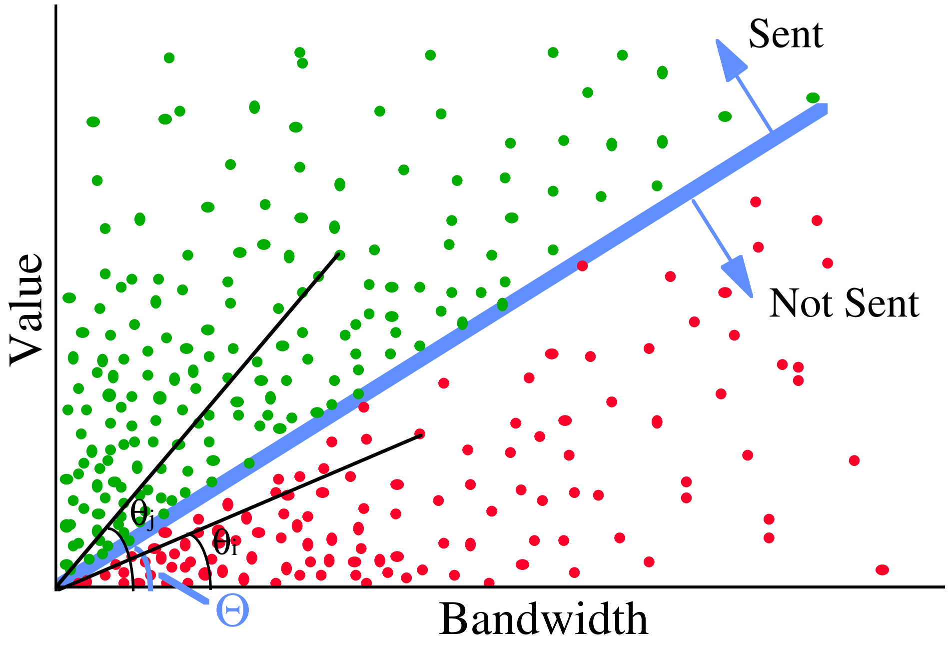


Figure 4.4

The equation for the mean offered bandwidth: $\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_0^{\frac{\pi}{2}}\cos\theta d\theta$ can be modified to enable the ratio of bandwidth transmitted to bandwidth offered to be calculated. The bandwidth of those users willing to pay the price for access is $\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_\Theta^{\frac{\pi}{2}}\cos\theta d\theta$. Note that this is not actually the *mean* transmitted bandwidth since the total probability associated with those users above the threshold, $\alpha^2\int_0^\infty re^{-\beta r^2}dr\int_\Theta^{\frac{\pi}{2}}d\theta \neq 1$.

In fact the ratio: $\dfrac{\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_\Theta^{\frac{\pi}{2}}\cos\theta d\theta}{\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_0^{\frac{\pi}{2}}\cos\theta d\theta}$ is the proportion of the offered bandwidth that is actually transmitted: ${}^{B_{tr}}\!/\!_{B_{off}}$

But of course, $\dfrac{\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_\Theta^{\frac{\pi}{2}}\cos\theta d\theta}{\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_0^{\frac{\pi}{2}}\cos\theta d\theta}$ can be simplified substantially, giving:

$$\frac{\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_\Theta^{\frac{\pi}{2}}\cos\theta d\theta}{\alpha^2\int_0^\infty r^2e^{-\beta r^2}dr\int_0^{\frac{\pi}{2}}\cos\theta d\theta} = \frac{\int_\Theta^{\frac{\pi}{2}}\cos\theta d\theta}{\int_0^{\frac{\pi}{2}}\cos\theta d\theta}$$

$$= \frac{\left[\sin\theta\right]_\Theta^{\frac{\pi}{2}}}{\left[\sin\theta\right]_0^{\frac{\pi}{2}}}$$

$$= \frac{1-\sin\Theta}{1}$$

$$= 1-\sin\Theta$$

In other words, $\frac{B_{tr}}{B_{off}} = 1-\sin\Theta$.

The step following the derivation of this concise expression for the ratio between transmitted bandwidth and offered bandwidth must be the determination of a method of setting the price for a given level of resource utilisation, and thus controlling the traffic level that the resource has to handle. Consideration of a geometric representation of the offered bandwidth, transmitted bandwidth and resource availability gives an insight into this step.

Two configurations are shown below. In the first, the price is set such that the transmitted bandwidth matches a predetermined – desired – utilisation. This may be (say) 80% or 90% of the system capacity, to allow for more rapid fluctuations in the transmitted bandwidth than the response time of the pricing system can handle. In the second configuration, the transmitted bandwidth is guaranteed to be less than the desired maximum utilisation, and as the utilisation increases, users are throttled back to a greater and greater extent.

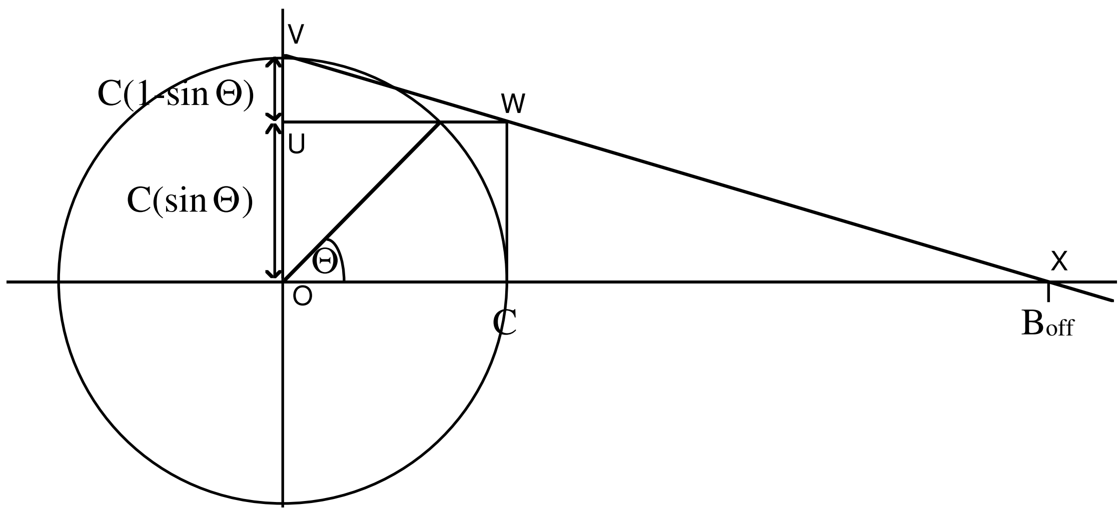


Figure 4.5 First configuration

In figure 4.5 above, B_{off} is the total offered bandwidth, and C the desired level of utilisation (notionally the resource capacity). Simple geometry shows that $\frac{C}{B_{off}} = \frac{C(1-\sin\Theta)}{C} = 1-\sin\Theta$ (triangles UVW and OVX are similar, so $\frac{OV}{OX} = \frac{UV}{UW}$). In other words, if the price threshold is $\tan\Theta$, the transmitted traffic will be C – the desired level of utilisation. An expression for $\tan\Theta$ can be derived from the information in figure 4.5:

Since $\frac{C}{B_{off}} = 1-\sin\Theta$, the price threshold, $\tan\Theta = \dfrac{\sin\Theta}{\cos\Theta} = \dfrac{1-\frac{C}{B_{off}}}{\sqrt{1-\left(1-\frac{C}{B_{off}}\right)^2}}$.

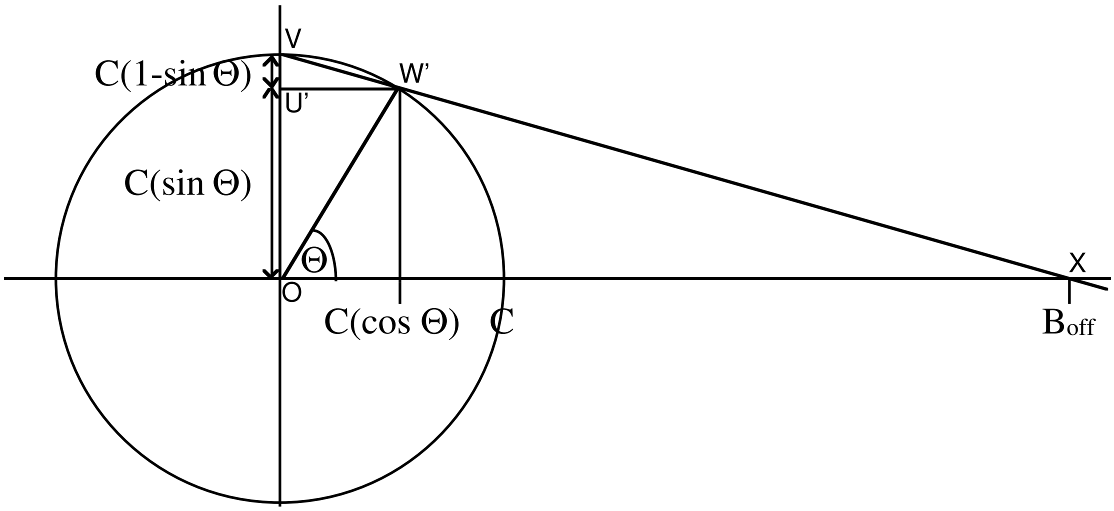


Figure 4.6 Second configuration

In the second geometric representation, B_{off} is again the offered bandwidth. C is the desired *maximum* level of utilisation. In this case, the similarity of the triangles U'VW' and OVX shows that $\frac{C\cos\Theta}{B_{off}} = \frac{C(1-\sin\Theta)}{C} = 1-\sin\Theta$ so that if the price is set at $\tan\Theta$, the amount of traffic transmitted will be $C\cos\Theta$. Again, an expression for $\tan\Theta$ can be derived by considering the geometry of the figure 4.6. In this configuration, $\Theta = 90° - 2\alpha$ (or $\frac{\pi}{2} - 2\alpha$) where α is the angle between OX and VX.

Therefore $\tan\Theta = \frac{1}{\tan 2\alpha} = \frac{1-\tan^2\alpha}{2\tan\alpha}$. But of course, $\tan\Theta = \frac{OV}{OX} = \frac{C}{B_{off}}$ so the price threshold required for $B_{tr}=B_{off}(1-\sin\Theta)=C\cos\Theta$ is given by:

$$\text{price} = \frac{1-\left(\frac{C}{B_{off}}\right)^2}{2\left(\frac{C}{B_{off}}\right)}$$

It is when the network is under the greatest degree of overload that any fluctuations in the actual transmitted traffic – caused by users not conforming fully to their traffic contracts, or by cell rate changes occurring faster than the price mechanism can regulate – could result in unresolvable resource contention, and thus cell loss. For this reason, it is thought that the second method of calculating the price threshold from the offered traffic and resource or network capacity should be employed: $\text{price} = \dfrac{1-\left(\dfrac{C}{B_{off}}\right)^2}{2\left(\dfrac{C}{B_{off}}\right)}$.

4.3.3 *Current direction of research*

In the preceding sections I have shown how, by considering the value users place on their communications, the flow of cells onto an ATM network can be controlled – by setting a value per unit bandwidth threshold. The amount of traffic admitted to the network can be regulated under any degree of overload (i.e. when the offered traffic exceeds the network or resource capacity) by changing the price according to a function of the ratio of offered traffic to capacity.

So far, it has been assumed implicitly that the value a user places on their network transactions is simply a scalar value – for example "I am willing to pay X for 2Mbytes per second" – or a range of values – "I am willing to pay X for 2Mbytes per second or Y for 1Mbyte per second". However, it is possible to consider a user's value assignment as being made up of different components or dimensions. Certain traffic types are more sensitive to delay than others, and if cells are delayed beyond a known limit they become effectively worthless. For example, a user's application at the receiving end of a real-time conversation may expect cells to arrive at regular intervals. If a cell is delayed by more than a predetermined amount, the application assumes that the cell has been lost and inserts "dummy" output – which may be random or an extrapolation from previous outputs. Therefore, if a cell carrying data from a real-time conversational application is delayed beyond that predetermined amount, it becomes worthless – there is no longer any need to guarantee its arrival. However, if the cell has not yet been delayed sufficiently to render it worthless, its urgency of transmission is considerable.

Two dimensions that can therefore be considered to make up value are the *intrinsic value* and the *urgency*. The *intrinsic value* is, or is derived from, the value that a user places on the overall success of the transmission – this is equivalent to the value as considered previously. The *urgency* is largely dependent on the traffic type – as indicated above. Cells carrying data for real-time or conversational communications have high urgency unless they are delayed beyond an application-determined amount, whereas non-conversational applications (file transfers, etc.) are not affected by increased cell delay to such an extent. These two basic components can be combined to give an overall value, which depends on both the user-assigned *intrinsic value* and the application dependent *urgency*.

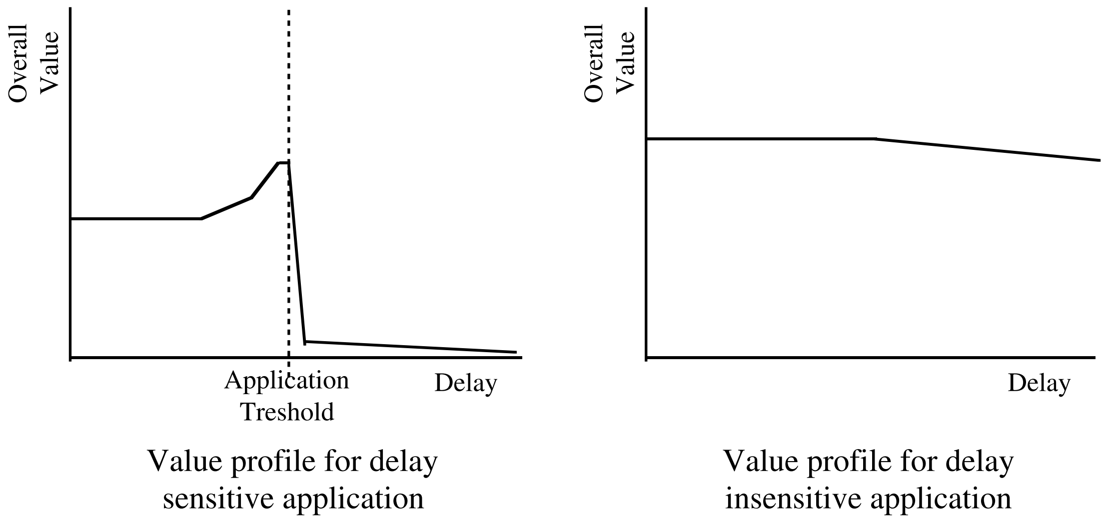


Figure 4.7

The graphs above indicate how the overall value could change for two applications – one sensitive to delay, the other relatively insensitive. The initial value level is determined by the user specified intrinsic value, while the shape of the graph is determined by the application type and the associated urgency. Even for applications that are relatively insensitive to delay, cells do not retain their value indefinitely (this is indicated by the slight drop in the overall value on the second graph).

A second argument in favour of considering value to be made up of these two components is that they allow traffic with low intrinsic value but high urgency to be transmitted ahead of traffic with high intrinsic value but low urgency. Consider figure 4.7 again. If two users were competing for access to the same resource, and one placed lower value on successful communication, but was using a delay sensitive application, under certain conditions the network would allow them access in preference to the user who placed higher value on their delay insensitive communication. For instance, if the next available slot for communication came after the delay sensitive application's delay threshold, the value of that communication would have decreased considerably. On the other hand the value of the delay insensitive application's data would not have changed appreciably. Therefore, greater total value of data would be transmitted – when integrated over time – by admitting the lower *intrinsic value* (but higher *urgency*) traffic in preference to the high intrinsic value, low urgency traffic. Both users would be satisfied with the service provided.

4.4 Dynamic case

The discussion of dynamic pricing so far has dealt with the case where the offered load is static. This has made the development of an overload control strategy based on pricing possible. However, in a real network levels vary from moment to moment as user demands fluctuate and as calls are initiated or terminated. The dynamic pricing traffic management scheme employed must be able to cope with these effects while delivering its goals of network stability, resource utilisation and value transfer. In order to meet these goals it is necessary that rules should be imposed that govern how rapidly traffic sources are allowed to vary their bandwidth. This is due to the time it takes for the network to respond to changes in load and offered traffic.

4.4.1 *Feedback*

It is apparent that, whatever the application, for a feedback mechanism to operate successfully, amongst other things the driving force must not be able to fluctuate so rapidly that the feedback can not respond or more extensively than the feedback can anticipate. If these conditions are not met, then by the time the feedback is applied to the force, its level can have changed to such an extent that the feedback is either unnecessary or insufficient.

In the case of an ATM network, the time it takes for cells to travel from source to the bottleneck being controlled by the pricing mechanism and for the price to propagate back to the source must be considered when determining how rapidly a source can increase its cell rate. For any given network configuration and set of users with their particular communication needs, it seems likely that there will be particular network regions that are more susceptible to overload. In order for the pricing mechanism to control overload anywhere in the network, the value per unit bandwidth required for network access should be dominated by the region or resource in greatest demand *that lies on the route that the user's cells will take*. In effect, for those users routing through the congested region, or bottleneck, the price they see is the price of routing via that bottleneck.

There is no particular reason why users’ output cell rates should not fluctuate rapidly and significantly. Therefore it is essential that rules are imposed which control the rate at which users can vary their offered bandwidth. These rules must limit the amount that a user’s offered bandwidth can change from moment to moment so that the price mechanism is not frustrated by rapid changes in cell rate, which are beyond its ability to control. It is anticipated that the choice of pricing mechanism will also have an impact on the success of dynamic pricing. As mentioned previously, at the end of section 4.2.2, it is when the network is under the greatest pressure that changes in offered bandwidth can have the most significant effects. If the network is lightly loaded, an increase in cell rate by a source will not have an adverse effect of the ability of the network to cope with the aggregate demand. However, if the network, or a region of the network through which a user's traffic is routed, is heavily utilised, then an unpredicted increase in cell rate by that user, or one that has not been negotiated, could very easily result in unresolvable resource contention.

4.4.2 *Shaping*

If a user has a block of data to transmit, there are several ways in which they could shape their transmission. For example they could send it as a short – high bandwidth – burst, or as a long, slow trickle. Both of these structures have sharply defined edges and have either constant bandwidth within the pulse or else no particular constraint on bandwidth within the pulse. However, I have already indicated the necessity for rules governing the rates of changes of users' cell rates. It is therefore natural to consider what effect these rules could have on the shape of a data pulse offered to the network, and what pulse shapes would be most compatible with the pricing mechanism.

Clearly a high bandwidth burst that has not been explicitly negotiated with the network is not an acceptable option. A simple simulation demonstrates the impracticability of such a pulse shape: suppose two such pulses were initiated by users, and the pulses met at a resource – a switch or multiplexer for example – and their combined cell rate exceeded the capacity of the resource. In this case, the resource would be overloaded instantly, and no matter how rapid the feedback mechanism or how short its propagation delay, the sources would not be able to modify their cell rates rapidly enough to prevent cell loss.

By extension, any pulse shape that has a sharply defined rising edge will be susceptible to the same danger. As the network load increases, the minimum cell-rate rise that could cause such contention decreases. It is therefore apparent that a user wishing to transmit at a high cell rate must either negotiate a route across the entire network at that rate, or else start by transmitting at a low rate and increasing within the limits imposed by the pricing mechanism and its associated rules.

It is also arguable that a sudden drop in cell rate should be avoided. The undesirable consequences of such a drop are not as immediately obvious, since a drop in cell rate would result in available network capacity being increased. However, if a user's cells were routed through a region suffering from congestion, a sudden, significant drop in the user's cell rate would result in a sudden change in the degree of congestion in that region. Consequently, the price would drop significantly. This would allow immediate access to more users, resulting in increased congestion once more. This would push the price back up again, in turn prompting those users whose value on communicating was now below the threshold to stop transmitting. It is easy to see how this could result in a system oscillating between two levels of utilisation. If, on the other hand, users were generally not allowed to drop their cell rate suddenly, the price would change only slowly – resulting in a more stable level of utilisation. Given that a stable level of utilisation is *likely* to require less management and signalling overhead than a constantly changing or oscillating level, the second scenario is more appealing.

4.5 Further work

The implementation of a dynamic pricing scheme where offered traffic levels are varying is evidently going to be complex. It has been demonstrated that, given certain assumptions about the users' communication requirements, the network traffic load can be successfully managed in the static case. It remains to be shown that the same can be said for the dynamic case. It is entirely plausible that a dynamic pricing scheme would work. However the increased complexity of the situation makes developing an analytic approach to the problem much harder. Indeed, the extra variables introduced, such as feedback propagation delays, and the changes that can occur to network loading between feedback being initiated and received could result in a system that cannot be modelled both analytically and realistically. By this, I mean that the increased complexity may result in it being necessary to introduce further constraints or assumptions to facilitate the analytic development, or else the results obtained from the analysis will themselves be so complex as to render them impossible to understand or interpret.

For this reason, it may prove necessary to demonstrate the effectiveness of a dynamic pricing scheme by extensive simulation, or small-scale implementation. The contribution of each of the scheme's parameters could then be empirically tested. In particular, the choice of rules that regulate cell-rate changes will be critical to the efficiency and even success of the scheme. If the rules chosen are too restrictive, the network will not operate efficiently. However, if the rules are too lax or are inappropriate for the mix of traffic types carried by the network, then the pricing mechanism will not be able to control the traffic flow tightly enough to prevent resource contention under some circumstances.

Similarly, the allowable shape of data pulses, and the relation between network loading and initial non-negotiated cell-rate or cell-rate change must be explored. As indicated above, the shape of an individual user's transmission could have a noticeable impact on the price presented to all users. For this reason it is important that the optimum combination of pricing mechanism and shaping be found. A potential pulse shape is that of a Gaussian curve. It has been commented that the equations describing the dynamic pricing scheme and its effect on traffic levels are similar to those found in quantum mechanics, and that since Gaussian shaped pulses also occur in quantum mechanics, they may well prove applicable in the case of a system controlled by dynamic pricing [Kirkby].

If the network is not congested, the data could be sent as a (discretised) Gaussian curve shaped pulse. However, if the network became congested, the curve would be flattened and the data sent in a pulse whose shape resembled the error function, the cell-rate eventually decreasing when the majority of the data had been transmitted.

4.6 Summary

When a network is lightly loaded, controlling the flow of traffic is not a significant problem and resource overload can be avoided without serious difficulty. However, as network loading increases the likelihood of overload at any point in the network increases and the necessary resource management requirements become more complex or have a noticeable impact on the flow of traffic in the network. Similar observations can be made in any situation where the supply of a commodity is limited and demand varies: as the mean level of demand increases, there will be times when demand exceeds supply, and not all consumers can be provided with their desired quantities of the commodity.

In the marketplace, periods of elevated demand encourage suppliers to increase their prices. When demand drops, the price drops – as suppliers attempt to stimulate demand. Dynamic pricing applies these basic “supply and demand” principles to a network, where the only significant difference is the frequency at which prices can change – the utilisation of a network (the demand for bandwidth) can vary far more rapidly than the demand for most other commodities.

Chapter 5 considers the application of dynamic pricing to super-symmetric networks, and outlines the concepts and the initial work that has been carried out in this area, with particular reference to the potential for creating an arbitrarily scalable network, whose bandwidth allocation is managed from the edges. Implementation of such a system could lie anywhere between a small office network and a global broadband multimedia network capable of carrying the traffic associated with all projected communication demands – including those of future telephony and Internet-style use as well as other predicted services.

5 Dynamic pricing on super-symmetric networks

In chapter 3, I introduced the concept of super-symmetric networks, and explained some of their properties and demonstrated how their symmetry makes it relatively easy to predict traffic loading across such a network. The advantages of super-symmetric networks in terms of their dynamic re-routing potential were also indicated. Chapter 4 described dynamic pricing and its advantages – in terms of providing users with a choice of how to react to network congestion, and how it could be used to manage resource utilisation successfully. In this chapter the benefits of combining these concepts – implementing dynamic pricing on super-symmetric networks – will be outlined.

5.1 Concepts

As already mentioned, a major issue facing ATM network designers is that of coping with the requirements of diverse traffic types. Each traffic type has its own performance requirements – maximum sustainable cell loss or delay variation and so on – and the needs of traffic types conflict. It is difficult to design a network optimised for carrying traffic sensitive to cell loss as well as traffic sensitive to cell delay variation, particularly if any statistical multiplexing gains in bandwidth utilisation are to be realised.

However, the implementation of dynamic pricing on super-symmetric networks may provide a solution to this problem. In order for a network to provide a good service to users of delay sensitive applications, it is essential that cell delay variation across the network be minimised – or eliminated if at all possible. This can be achieved by reducing the size of buffers in the network, or by removing them altogether. If either of these options is implemented, it becomes essential to control the flow of cells onto and across the network to guarantee that no part of the network could be overloaded. If overload occurs, cells will be lost – degrading the service provided to users of loss-sensitive applications. Overload can be avoided by using the dynamic pricing mechanism, which signals the current level of resource utilisation on the route being taken by loss-sensitive data. As the price increases, users transmitting loss-sensitive data may either decrease their cell-rate, providing their own buffer space, or bear the price increase and continue transmitting at their current rate – relying on other network users to decrease their cell-rates.

It is evident that a dynamic pricing scheme could be implemented on any network as a flow control system. Since each network user has a limited amount of money (or network-use credits), a sufficient increase in price will result in a decrease in the amount of traffic transmitted by each user, and thus will control the flow of traffic across the network.

However, the advantage gained by implementing dynamic pricing on super-symmetric networks is that if a particular network region becomes congested, not only can newly arriving traffic be routed away from the congestion, but calls *in progress* can be dynamically re-routed onto less congested links. If the price of a user's current route increases the user may investigate alternative routes, and select the cheapest. Alternatively, dynamic re-routing may be managed by the network, without the need for user intervention. In section 2.1.3 I gave an example of how dynamic re-routing could occur on a small fragment of a super-symmetric network.

In section 4.4.1, I commented that for the pricing mechanism to control overload anywhere in the network, the value per unit bandwidth required for network access should be dominated by the region or resource in greatest demand *that lies on the route that the user's cells will take*. If the pricing mechanism is implemented on a super-symmetric network, there are routing choices between many pairs of nodes. Therefore unless the congestion is at the source or destination it is likely that an alternative route can be found which avoids the congestion. This has the effect that the traffic load is automatically balanced across the network.

5.2 Global implementation – some thoughts

There are two basic methods of increasing the scale of the super-symmetric networks described in chapter 3. The first is to scale up the size of the square grid that results from unfolding the hypercube. The hypercube has 16 vertices, and so a network constructed to match the connectivity of the hypercube would have only 16 nodes. However, as I commented in section 3.2, if the grid size is scaled up, a greater number of nodes can be connected. Although the four-dimensional object that would have to be "unwrapped" to produce the larger grid would not be a hypercube, the essential properties of the network would be unchanged: uniformity of link length and multiplicity of routing options would be maintained.

An alternative method of increasing the number of nodes in the network is by the use of a hierarchy of hypercube networks. Each of the nodes in the topmost level is a hypercube in the next level down. Each node in that hypercube is itself a hypercube in the next level. This pattern may be repeated as often as necessary, until sufficient coverage has been provided. Of course, the two methods may be combined – to give a hierarchy of hypercube-like networks. The nodes at the lowest level in the hierarchy are the network access points for individual users.

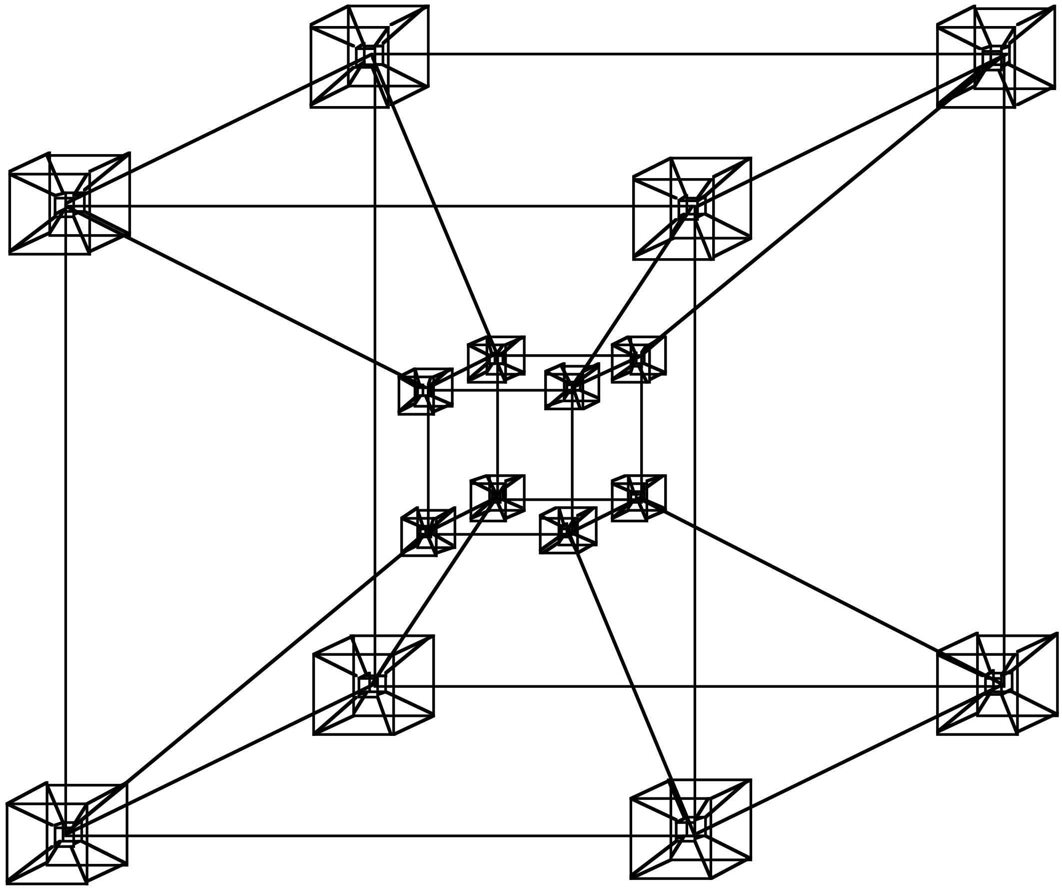


Figure 5.1 Hierarchy of hypercubes – two levels

Such a hierarchy of networks could be scaled up to global proportions without difficulty. The symmetry would be maintained at each level in the hierarchy. By considering the three-dimensional projection of the hypercube – the torus – it is possible to imagine how the network could be mapped onto the globe. The torus could be wrapped around the world – rather like sticking a golf ball in the middle of a ring doughnut!

An important concept that is being considered is that at the lowest level in the hierarchy, individual ATM cells are switched and routed, but at successively higher levels, packets of cells are grouped together and these *supercells* share routing and switching. The number of cells in a supercell is determined by the ratio of the link lengths in parent and child hypercubes. If the link length of the parent hypercube was sixteen times longer than the link length of the child hypercube, then the supercells would consist of sixteen ATM cells – with a common destination on the parent hypercube. Similarly, at the next level up, the packets would consist of sixteen supercells. When a packet reaches its destination and is passed to the lower level hypercube, the individual cells or supercells it contains are routed onwards individually.

The uniformity of link lengths within each level of the hierarchy allows switching to be synchronised and to occur at a relatively slow speed, since on the higher levels consecutive groups of cells are routed to a common node. The principles of this hierarchical routing and cell grouping scheme are indicated in figure 5.2.

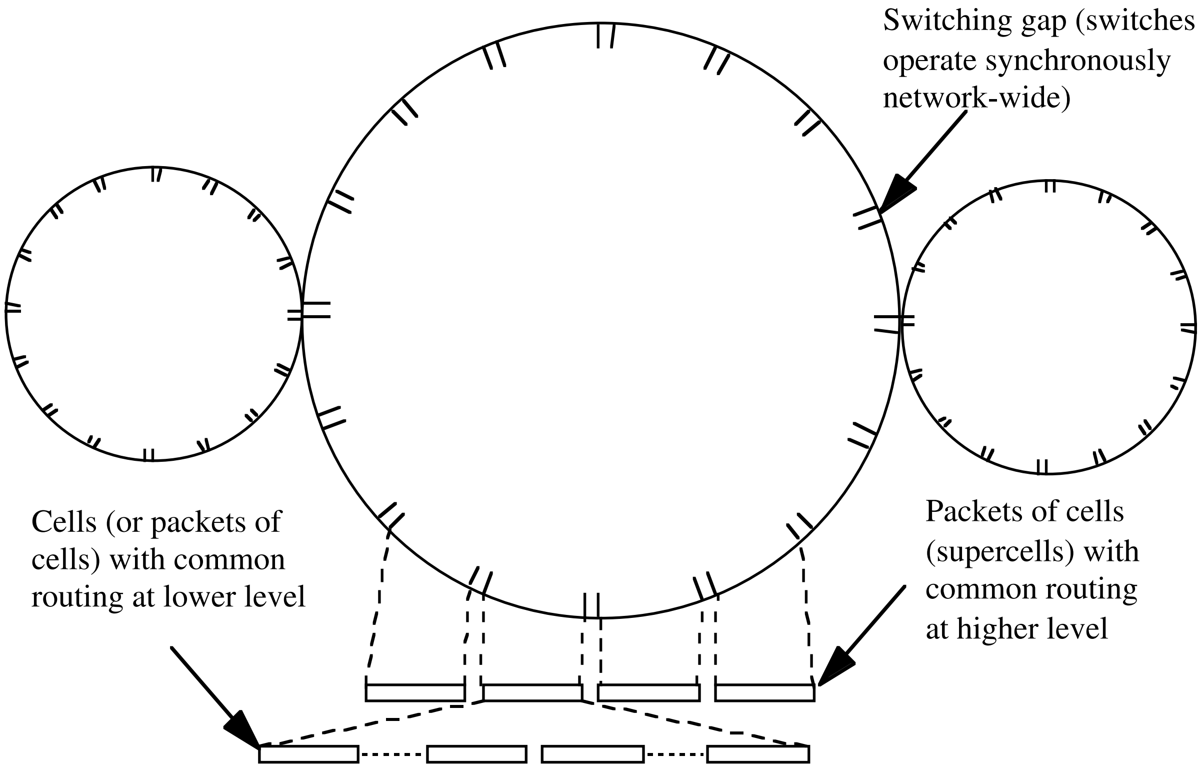
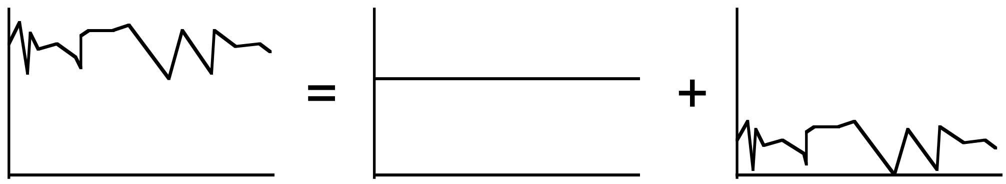


Figure 5.2

Access to the network, and bandwidth management on the network, would be controlled by dynamic pricing. The rules regarding allowable rates of changes of bandwidth – which are necessary primarily because of the propagation delay between congested regions and traffic sources – would be related to the number of levels in the hierarchy that a call was traversing. A call to a "local" node (on the same low level hypercube) could vary its bandwidth more rapidly, since the response time of the dynamic pricing would be less, than a call to a "distant" node (on another hypercube – involving routing up and back down the hierarchy). Alternatively, since a variable bandwidth connection can be considered to be made up of a constant bandwidth and a variable bandwidth part, a user wishing to initiate a variable bandwidth call to a distant node may be required to "reserve" a larger proportion of their peak bandwidth than a user initiating a call to a local node. Increases to the peak would be a smaller proportion of the reserved part – as demonstrated by figure 5.3 below:



Alternatively

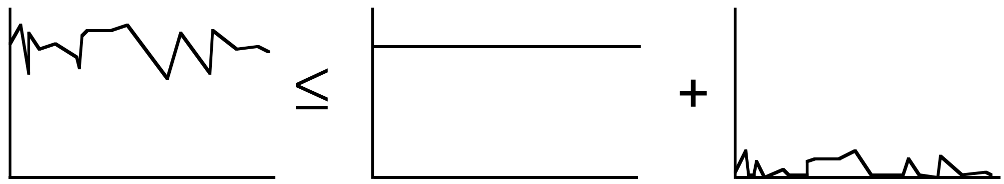


Figure 5.3

In the second diagram, a larger proportion of the peak bandwidth has been taken up with a constant bandwidth element. The variable bandwidth part of the communication has a smaller amplitude, but this advantage is offset by reduced potential for statistical multiplexing.

Combining these principles with the dynamic pricing flow control mechanism and with suitable rules governing cell-rate changes will be a complex task, but appears to be a promising technique for maximising statistical multiplexing possibilities and guaranteeing quality standards (cell loss and delay variation) to the users.

6 Summary and Conclusions

Designing ATM networks for a variety of traffic types – voice, video, data, etc. – has traditionally been seen as managing the trade-off between minimising cell loss probability and controlling cell delay variation. A novel approach to this problem has been suggested which is built on two independent concepts: super-symmetric networks and dynamic pricing. Research into these areas is still at an early stage, but the progress made so far continues to indicate their suitability as elements of the solution to this problem.

Three areas were presented as motivation for the project. The major motive was resolving the conflict between minimising cell loss probability and managing cell delay variation. Other motives were the potential application of super-symmetric networks to distributed processing and the introduction of user choice if dynamic pricing was implemented.

In this chapter, I shall summarise the properties of super-symmetric networks and the implications for routing and dynamic re-routing. I shall then outline the principles of dynamic pricing and the progress that has been made in quantifying its effectiveness as a bandwidth management system. Before presenting my overall conclusions, I shall briefly re-examine the three motives in the light of what has been learned during this project.

Super-symmetric networks

Most practical network designs incorporate at least some redundancy, link protection and routing choices. However, in the case of circuit-switched networks, it is not generally possible to implement dynamic re-routing (changing the route of a call in progress), since the alternative routes are of different lengths. Changing the route would result in a change in the time taken for data to propagate from source to destination. This would result in either an interruption to the data stream, or a brief period when too much data was received – depending on the relative lengths of the original and new routes. Both of these options result in a degradation of the user's perceived quality of service. In packet networks, dynamic re-routing is commonplace but variation in packet inter-arrival times is not considered important.

The topology of a super-symmetric network is based on the structure of a four-dimensional symmetric object – the hypercube. In such a network, link lengths are uniform, and depending on the relative positions of source and destination nodes, a variety of equal length (and hence equal propagation time) routes exist. The super-symmetric network structure is inherently scalable, either by increasing the number of nodes and maintaining the regular connection patterns and uniform link lengths, or by creating a hierarchy of networks and switching *supercells* of increasing size at successively higher levels in the hierarchy.

If a circuit switched network were to be built with the super-symmetric structure, it would be possible to re-route dynamically without disrupting a user's service. Similarly, if a bufferless super-symmetric packet network were constructed, packet inter-arrival time would match the inter-departure time. However, without the implementation of an effective bandwidth management scheme, the packet loss resulting from the removal of buffers at the switches would impact adversely on the users' perception of the services provided by the network.

Dynamic pricing

When heavily loaded, packet networks suffer from considerable delay, and in the worst case packet loss. The effects of congestion are distributed equally among all users by current congestion control schemes – passive or active. A passive congestion control scheme can be implemented by providing large buffers at bottlenecks. An example of an active congestion control scheme is the ABR (Available Bit Rate) method for ATM, which signals all users to reduce their cell-rate when the network is congested.

However, the effects are not necessarily felt equally by all users. If the data from an interactive application is being carried across a packet network, the user's perception of the network or application performance will be reduced if the network imposes noticeable delay on their data. On the other hand, other applications do not depend on rapid data transfer (e.g. e-mail, off-line file transfer). Dynamic pricing offers a solution to this problem, by redistributing the congestion effects. Those services dependent on low delay pay more per unit bandwidth, and are guaranteed low delay, while those services that do not require low delays or minimal delay variation may pay less, but suffer extra delay when the network is congested.

By using price to control congestion, the *value* of data transferred across the network during periods of congestion is maximised. It should be noted that maximising the value transfer does not necessarily equate to maximising the utilisation of the network, particularly if the number of users is small and they have clearly defined responses to price changes.

Since a dynamic pricing scheme such as is being proposed has not been implemented, it is not possible to say how its users would respond to the price changes. In order to draw any conclusions about the effectiveness of a dynamic pricing system, it is necessary to make certain assumptions about how users would value their communications and how they would respond to price variation. It seems apparent that a lot of communication is relatively unimportant, or of low intrinsic value. Similarly most point-to-point communication does not require a high bandwidth channel. Given these broad estimates, and the fact that there need not be a relation between communication bandwidth and value, it was assumed that user assigned value and required bandwidth could be represented as independent random variables with Gaussian distributions.

On the basis of these assumptions it is possible to derive an analytic expression for the effectiveness of a pricing scheme at managing bandwidth, for a system with a large user base. Since no user of a network has an infinite supply of money, it is evident that *whatever* the distribution of required bandwidths and user-perceived values, a pricing scheme can be used to control the traffic admitted to the network. However, if different assumptions were made about the distributions of bandwidth requirements and user-perceived values, the analytic expression quantifying the effects of a pricing scheme would also be different (if possible to derive at all).

As they stand, the dynamic pricing principles make no distinction between delay sensitive traffic and delay tolerant traffic. Therefore, two cell streams with the same user-perceived value but different tolerances to delay would be treated identically, regardless of the degree of network congestion and likelihood of delay. However, it is not difficult to conceive of a way in which this deficiency could be corrected. By considering the value to be made up of different components – the *intrinsic value* (the user-perceived value) and the *urgency* (traffic type dependent) – and requiring that the goal of the network should be to transfer the maximum value *over time*, delay sensitive traffic will be given priority over delay tolerant traffic. The overall value of delay sensitive traffic decays rapidly by comparison with that of delay insensitive traffic, and so a greater total value will be delivered by the network if the urgent traffic is transmitted ahead of the traffic insensitive to delay.

In order for dynamic pricing to be used as a flow management system, account must also be taken of the effects of the time taken for price changes to reach the users. Rules must be set in place that prevent users from varying their cell rates too rapidly. If a user's traffic is routed through a remote congested region of the network, rapid cell rate changes could have a significant impact on the level of congestion in that region. Therefore it is imperative that the user should be aware of the potential for overload that would result if they increased their cell rate rapidly, and instead vary their rate slowly – allowing the price feedback to keep up with their changes.

Cell loss and delay

If a network operator could implement an ATM network that guaranteed zero cell loss and eliminated cell delay variation for those traffic types sensitive to it but retained the flexibility of "traditional" ATM, then they would have a significant advantage over operators unable to make those guarantees. The combination of bufferless super-symmetrical networks and dynamic pricing should enable this goal to be realised. Eliminating buffers from the core network makes CDV impossible. Dynamic pricing allows the network operator to control the traffic level – preventing overload, and thus preventing cell loss in the network. The implementation of dynamic pricing on super-symmetric networks makes dynamic re-routing of calls possible, which in turn allows traffic loads to be balanced.

Distributed processing

If the elements of a distributed processing system – data sources, storage, processors, and so on – were located at the nodes of a super-symmetric network, it would be possible to construct a scalable system, without incurring the penalty of implementing a complex timing recovery procedure. The regular structure of the super-symmetric network would allow straightforward provision of the information necessary for the successful operation of a distributed processing system.

User choice

Dynamic pricing can be used to provide network users with a range of options if the network becomes congested. Current congestion management techniques for packet networks – large buffers and, eventually, packet loss – are applied indiscriminately to all users when congestion strikes. This results in packet delay variation and packet loss. If the same techniques are applied in ATM networks, the result would be cell delay variation and cell loss during periods of overload. However, if price is used as a congestion control technique, users are given a choice: suffer the consequences of network overload, but have cheap communications, or retain high quality service (low loss and delay variation) at increased cost.

Conclusions

The concepts of dynamic pricing and super-symmetric networks have potential applications in a wide variety of areas. This project has concentrated on examining the concepts, and suggested several possible uses for them. The primary motivation for investigating the concepts was their potential use as fundamental elements in constructing ATM networks and network management systems that could give *guaranteed* performance – in terms of cell loss and cell delay variation – to their users.

This study of the concepts strongly suggests that they could indeed be used to make it possible to construct scalable ATM networks with guaranteed zero cell loss and guaranteed uniform cell delay. The cores of such networks could be fully optical, with buffering provided only at their peripheries – scalable optical ATM networks. Feedback from the network nodes would provide the users with the information and instructions necessary to prevent overload in the optical part of the network by buffering their traffic or reducing their demand for bandwidth. This feedback would take the form of the price charged for access. The pricing mechanism could be implemented either with “real” money, users being billed according to their use of the network, or with “access tokens” which may be pre-purchased or within a company could, for example, be allocated to staff according to seniority. Those users with a greater budget (of money or tokens) or who regarded their communication as being of sufficient importance would continue sending their data, while those users with a tight budget or non-urgent traffic would wait for the price to fall before sending their data.

When the traffic offered to a dynamically priced network exceeds the network’s capacity the price increases. This price increase results in a decrease in traffic levels, which in turn prevents the core of the network from becoming congested. The traffic carried during periods when a potential for overload exists is of maximum value (as perceived by or imposed on the users). When the overload has passed, and traffic levels drop, the price will fall – allowing those users on a lower budget to increase their bandwidth utilisation.

If a dynamic pricing scheme were implemented on a private network, it could be used to provide certain users with a superior service (for example, a senior manager is guaranteed a high-quality video link, even when the network is busy). Alternatively, or additionally, such an implementation could be configured to provide certain services or applications with guaranteed minimum standards.

The combination of super-symmetric networks and dynamic pricing promises to be an exciting solution to the problem of designing ATM networks for a variety of traffic types. The traditional trade-off between minimising cell loss probability and controlling cell delay variation is eliminated altogether. The resulting networks should continue to perform well even when the offered traffic exceeds the capacity, either locally or globally, whether momentarily or for sustained periods, delivering the maximum throughput of value.

7 Acknowledgements

I would particularly like to thank each of the following for their assistance and insightful comments or contributions during the course of my MRes project: Paul Kirkby (Nortel), Mark Searle and Phil Lane (UCL), Richard Epworth, Robin Thompson and Mark Gibson (Nortel). I would also like to express my gratitude to the Engineering and Physical Sciences Research Council and to Nortel (Northern Telecom), without whose financial assistance I would have been unable to conduct this research.

References

Drury, D M	ATM traffic management and the impact of ATM switch design. Computer Networks and ISDN Systems, 28 (1996), 471-479.
Epworth, R E	Modal Noise, Causes & Cures. Laser Focus, September 1981, 109-115.
Kirkby, P A	Personal communication.
MacKie-Mason, J K & Varian , H R	Some Economics of the Internet. University of Michigan Tech. Report. *and* Pricing the Internet. University of Michigan Technical Report.
Sharangpani, H P & Barton, M L	Statistical Analysis of Floating Point Flaw in the Pentium™ Processor. Intel Corporation, November 1994.
Vestmar, B J A	Design considerations for the grid type of communication network. 1975.

Appendix 1

The offered bandwidth (calculated using rectangular co-ordinates) is given by: $\int_0^\infty b\alpha e^{-\beta b^2}\,db$.

The offered bandwidth (calculated using polar co-ordinates) is given by: $\int_0^\infty \alpha^2 r^2 e^{-\beta r^2}\,dr$.

It is straightforward to confirm that these are equivalent, as shown below.

Evaluating the rectangular co-ordinate based equation:

$$\int_0^\infty b\alpha e^{-\beta b^2}\,db$$

$$= \int_{b=0}^{\infty} \tfrac{-\alpha}{2\beta} e^u du \text{ where } u = -\beta b^2$$

$$= \left[\tfrac{-\alpha}{2\beta} e^u\right]_{b=0}^{\infty} = \left[\tfrac{-\alpha}{2\beta} e^{-\beta b^2}\right]_0^{\infty}$$

$$= \frac{\alpha}{2\beta}$$

$$= \frac{1}{\sqrt{\beta\pi}} \text{ since } \alpha = 2\sqrt{\tfrac{\beta}{\pi}}$$

Similarly, the polar co-ordinate version can be evaluated as:

$$\int_0^\infty \alpha^2 r^2 e^{-\beta r^2}\,dr \int_0^{\frac{\pi}{2}} \cos\theta d\theta$$

$$= \int_0^\infty \alpha^2 r^2 e^{-\beta r^2}\,dr \cdot [\sin\theta]_0^{\frac{\pi}{2}}$$

$$= \int_0^\infty \alpha^2 r^2 e^{-\beta r^2}\,dr \cdot 1$$

$$= \int_0^\infty \alpha^2 r^2 e^{-\beta r^2}\,dr$$

(continued overleaf)

This can be evaluated by integrating by parts, and remembering that $\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt = \operatorname{erf}(x)$.

Since $\int u\frac{dv}{dx}dx = uv - \int v\frac{du}{dx}dx$ and $\frac{d}{dr}r = 1$, $\int re^{-\beta r^2}dr = \frac{1}{2\beta}e^{-\beta r^2}$ then

$\int_0^\infty \alpha^2 r^2 e^{-\beta r^2}dr$ can be rewritten as $\alpha^2\int_0^\infty r\cdot re^{-\beta r^2}dr$ and therefore,

$$\int_0^\infty \alpha^2 r^2 e^{-\beta r^2}dr = \alpha^2\left[\left[\frac{-r}{2\beta}e^{-\beta r^2}\right]_0^\infty - \int_0^\infty \frac{-1}{2\beta}e^{-\beta r^2}dr\right]$$

$$= \alpha^2\left[0 + \frac{1}{2\beta}\int_0^\infty e^{-\beta r^2}dr\right]$$

$$= \frac{\alpha^2}{2\beta}\frac{\sqrt{\pi}}{2\sqrt{\beta}}\operatorname{erf}(\sqrt{\beta}\infty) \quad \left(\text{since } \int_0^x e^{-at^2}dt = \frac{\sqrt{\pi}}{2\sqrt{a}}\operatorname{erf}(x\sqrt{a})\right)$$

$$= \frac{\alpha^2\sqrt{\pi}}{4\beta\sqrt{\beta}}$$

And, since $\alpha = 2\sqrt{\frac{\beta}{\pi}}$, this can be simplified giving

$$\int_0^\infty \alpha^2 r^2 e^{-\beta r^2}dr = \frac{4\frac{\beta}{\pi}\sqrt{\pi}}{4\beta\sqrt{\beta}}$$

$$= \frac{1}{\sqrt{\beta\pi}}$$

as expected.

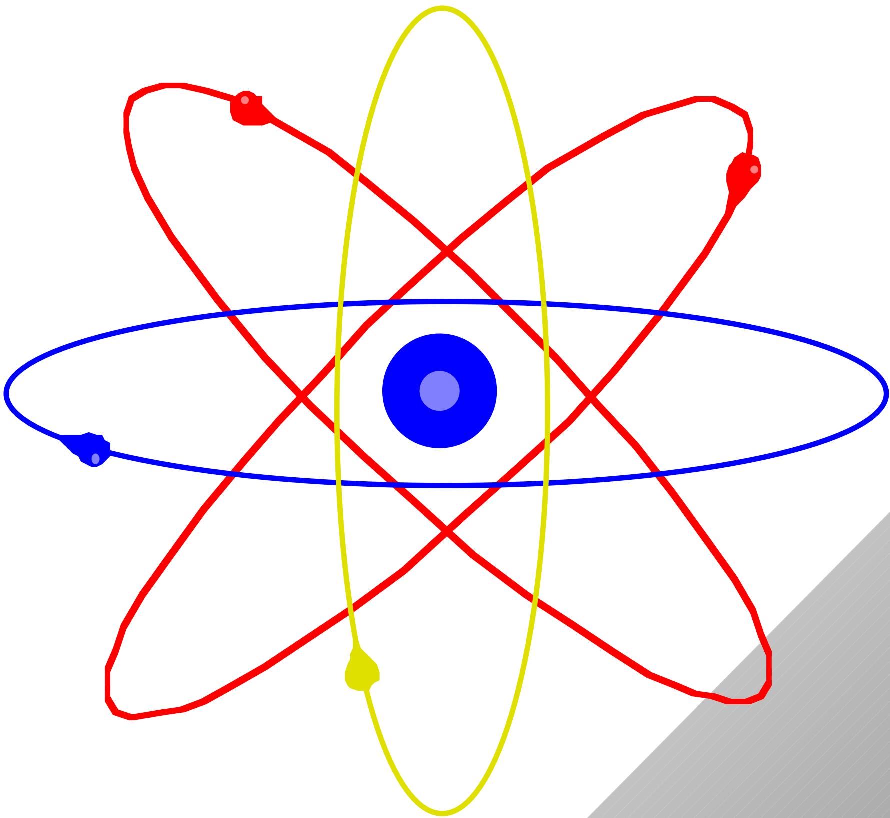
Appendix 2

The following list summarises the acronyms and abbreviations used in the body of the dissertation.

AAL	ATM Adaptation Layer
AAL5	ATM Adaption Layer number 5
ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
B_{off}	Offered bandwidth
B_{tr}	Transmitted bandwidth
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDV	Cell Delay Variance
IP	Internet Protocol
$kbits^{-1}$	Kilobits per second
$Mbits^{-1}$	Megabits per second
Mbyte	Megabyte
MCR	Mean Cell Rate
PDU	Protocol Data Unit
TCP	Transmission Control Protocol
UBR	Unspecified Bit Rate
UPC	Usage Parameter Control
VBR	Variable Bit Rate

Scalable Optical ATM Networks

Martin Biddiscombe



University College London
1996