Technical Report

UCAM-CL-TR-184
ISSN 1476-2986

Number 184

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Site interconnection and the exchange architecture

David Lawrence Tennenhouse

October 1989

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted September 1988 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Darwin College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

The users of a site's telecommunication facilities rely on a collection of devices, transducers and computers, to provide the primary communications interface. In the traditional approach to site interconnection, some of these devices are directly attached to specialized carrier networks. The remaining devices are attached to local networks that are tailored to support communication amongst compatible devices. These local networks are, in turn, attached to common carrier networks that support communication between compatible devices at remote sites. This arrangement does not reap the full benefits of network and service integration: each local network has its own common carrier interfaces; and there is no provision for device-independent processing, storage, and forwarding elements.

This dissertation describes a layered approach to site interconnection. Communication between peer sites is supported by the lower layer carrier networks, and *associations* between upper layer clients are supported by the local network layer. The *site interconnection layer*, inserted between the local and carrier layers, facilitates communication between peer local networks. This layer is composed of independent subsystems that offer the site interconnection service (SI-service) to their upper layer clients. Each SI-subsystem is a funnel through which various device-dependent symbol sets are encoded into a common digital format. The symbol streams of concurrent upper layer associations are multiplexed together for presentation at the shared carrier interfaces. Service integration is achieved through the encoding of many different styles of communication (voice, video, facsimile, file transfer, etc.) into a common symbol set.

The first part of this dissertation develops the connected argument supporting this layered architecture. The second part describes the experimental development and analysis of the *exchange architecture*, which provides an SI-service that supports *Asynchronous Transfer Mode (ATM)* communication. The ATM approach is characterized by the use of small packets in conjunction with switch fabrics that provide comparable performance to circuit switching, and permit much greater variability in traffic patterns. The switch fabric of the pilot implementation is based on the Cambridge Fast Ring: the CFR packet structure is the basis of the ATM encoding; and the VLSI ring technology has been used to construct the individual SI-subsystems. In this application, the CFR provides ATM-based switching and multiplexing facilities.

This work is distinguished by its emphasis on *site independence* and *universal access* to telecommunication services. The principal contributions of the thesis relate to: *site interconnection*; *ATM* encodings; *out-of-band* and *non-invasive* network management; *particular* analysis methodologies; and the design of *multi-service* networks.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Communication between individuals is based upon the exchange of meanings through a common system of symbols. Visual actions, still images, spoken languages, and written alphabets are all examples of systems of symbols that people use to support communication. *Telecommunication* is accomplished through the processes of *coding* and *transmission*. A transducer can be used to encode the common symbols understood by humans into electrical or optical symbols. These symbols can be transmitted over short or long distances and on reception they are decoded to their original form.

The modelling of communication in terms of coding and transmission can be applied in a hierarchical manner to describe communication between intermediaries, such as telex machines, on behalf of individuals. At some *upper* layer of the hierarchy the telex operators are the peer entities and their communication is based on the encoding of information into keystrokes and the transmission of those keystrokes. At a *lower* layer the telex machines are the communicating peer entities. The transmission of operator keystrokes is based on further encoding into sequences of electrical impulses which are in turn transmitted along copper wires.

Traditionally different encoding and transmission hierarchies have been used to support the exchange of different styles of communication. For example, the public telex and telephone networks utilize different codes which are tailored to their respective symbols: sequences of typewriter keystrokes and human speech. The transmission systems used to carry the encoded communication are physically distinct, and even when a telephone handset and a telex machine are adjacent to each other, each is serviced by a separate medium (usually a wire) which connects the transducer to its own communication network.

This dissertation is concerned with the architectural aspects of telecommunication between distinct physical sites. The goal of this work has been to design,

implement, and experiment with an architecture for multi-service site interconnection. Service integration is achieved through the encoding of many different styles of communication (voice, video, facsimile, file transfer, etc.) into a common symbol set. The use of a common encoding facilitates the development of service independent processing, storage, and forwarding components.

1.1 Background

Evolution of Voice and Data Communication Networks

Since the introduction of telephony, voice traffic has been the dominant form of narrowcast telecommunication and the public telephone network has evolved to provide near universal access to voice services. Until recently this evolution has been guided largely by the requirements of voice traffic and the principle of universal access. Although telephone networks have often been used to carry data communication, the process is somewhat cumbersome. Typically, modems are used to encode (modulate) and decode (demodulate) the digital symbols into analogue signals that fall within the frequency range supported by the voice service.

With the advent of computers, data communication has become an important form of communication with its own network tradition. The evolution of data networks has been guided by the great variability of the traffic. Most data networks concurrently support interactive traffic, arising from transaction-oriented and telemetry applications, and bulk traffic arising from file transfer and electronic mail applications. In recent years various research projects have used data networks to support a wider range of communication services including voice, images and video.

Voice traffic imposes stringent performance demands on a communication network but the traffic itself conforms to fairly fixed, well specified patterns. In contrast, the wide range of data communication applications impose highly variable demands on data networks while accommodating significant variations in network performance. For this reason the two network traditions have evolved in parallel developing their own local distribution architectures (PBX versus LAN) and distinct long haul architectures (circuit switching versus packet switching). The introduction of digital communication into the public telephone network has precipitated a great deal of interchange between the voice and data networking

communities.[1] With the confluence of the two networking traditions has come the realization that current communication architectures are not suited to the range and variability of integrated services.

Local and Common Carrier Networks

Communication networks can be broadly divided into common carrier and local networks. Local networks provide services to geographically proximate user communities that share common interests. The services provided by these networks can be tailored to the specialized needs of their relatively small user communities. In contrast, common carriers, such as the public telephone network, provide standardized services to diverse user communities spread over large geographic areas.[2] Carrier networks achieve universal access through the provision of *lowest common denominator* services to the widest possible community.

Local networks include traditional private business exchanges (PBXs) that support telephony and local area data networks (LANs). Since each local network is relatively small, the service provided can evolve, or new networks can be commissioned, to take advantage of improvements in technology and to provide enhanced services within specific communities.

Carrier networks are normally structured into distribution subnetworks that are linked together by a backbone network. The backbone can transparently evolve to take advantage of advances in technology. However, in contrast to local networks where new technology has led to new services, evolution of the backbone network is largely reflected in the economy and capacity of the network to provide the same service.

The burden of universal access largely falls on the distribution subnetworks, each of which services a user community which is far more diverse and geographically dispersed than that serviced by a local network. The introduction of new technology and services into the distribution network is a cumbersome process. In

---

[1]For example, in the ISLAND architecture [Calnan 87] voice services are carried over data networks using an encoding [CCITT G.711] developed for use in digital telephony. Similarly, data communication techniques now figure prominently in telephone network design and operation. The ISDN protocol reference model [CCITT I.320] is based on the OSI Reference Model [ISO 7498-1] for data communications. Furthermore, the ISDN signalling services and protocols ([CCITT Q.920], [CCITT Q.921], [CCITT Q.930] and [CCITT Q.931]) are descendant from the HDLC and X.25 data communication standards.

[2]Throughout this text, private services linking discrete sites are treated as special cases of carrier services and are not considered separately.

general, new services are of only limited use until they are either universally available or at least available between geographically dispersed communities with common spheres of interest.[1] Given the massive numbers involved, major changes to the distribution network take a long time to implement and must bring about quantum improvements in networks costs and services.[2]

1.2 Site Interconnection

The end users of a site's telecommunication facilities rely on a collection of devices, transducers and computers, to provide the primary communications interface. In the traditional approach to site interconnection, illustrated in Figure 1.1, some of these devices are directly attached to specialized carrier networks. The remaining devices are attached to local networks that are tailored to support communication

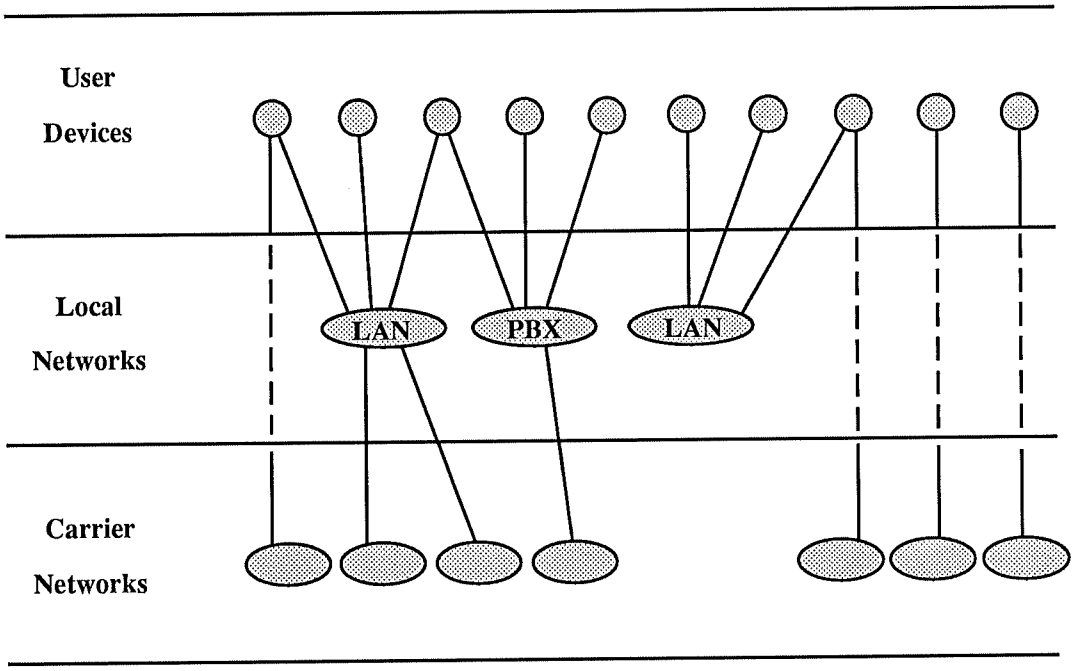

Figure 1.1: Traditional SI Organization

[1]For example, the medical community.

[2]ISDN, which represents a relatively minor upgrading of the distribution plant, was first approved by the CCITT in 1980. However, the service is still not widely available and its implementation may be overtaken by recent discussions concerning broadband digital services [Minzer 87]. The broadband project will involve the complete replacement of the copper distribution plant with optical fibre, and to justify this investment, it will be necessary to increase the bandwidth available for subscriber services from the ISDN range of 144-2048 Kbps to about 150-500 Mbps.

amongst compatible devices. These local networks are, in turn, attached to corresponding carrier networks that support communication between compatible devices at remote sites.

The local networks perform two important functions:

- They *contain* local communication between similar devices located at the same site; and
- They *concentrate* traffic, allowing a number of local devices to share the point(s) of attachment between the local network and the common carriers.[1]

While this traditional arrangement provides a measure of containment and concentration it does not reap the benefits of service integration: each local network has its own points of attachment to an assortment of specialized common carriers. One proposal is to replace the collection of local networks with a single integrated network capable of supporting many forms of telecommunication traffic. Within a site, every device is attached to this monolithic local network which is itself attached to a fully integrated common carrier network. This approach has been favoured by a number of research projects. Whilst this work may lead to the integration of a limited traffic subset, it is unlikely that any one technology will support the complete integration of all present forms of traffic, let alone every new application that may arise in the future.

1.3 Aims of This Research

This dissertation introduces an alternative approach to site interconnection, based on the hierarchical model illustrated in Figure 1.2. In this layered architecture, communication between peer sites is supported by the lower layer carrier networks. Similarly, *associations* between upper layer clients are supported by the local network

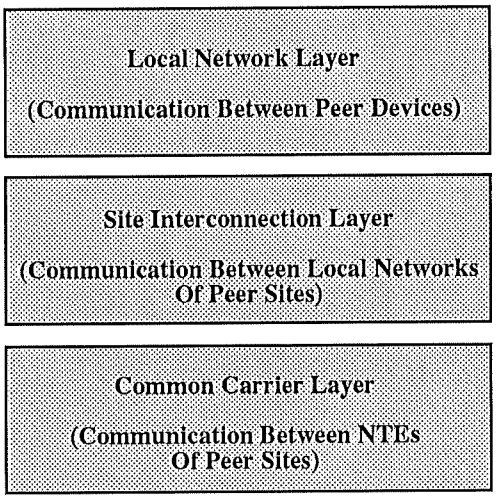

Figure 1.2: Network Layer Structure

[1]In a typical PBX environment the concentration function is directly supported by the switch fabric of the local network. In LAN environments concentration may be supported by relaying traffic through gateway devices or processing systems that are directly attached to the appropriate carrier.

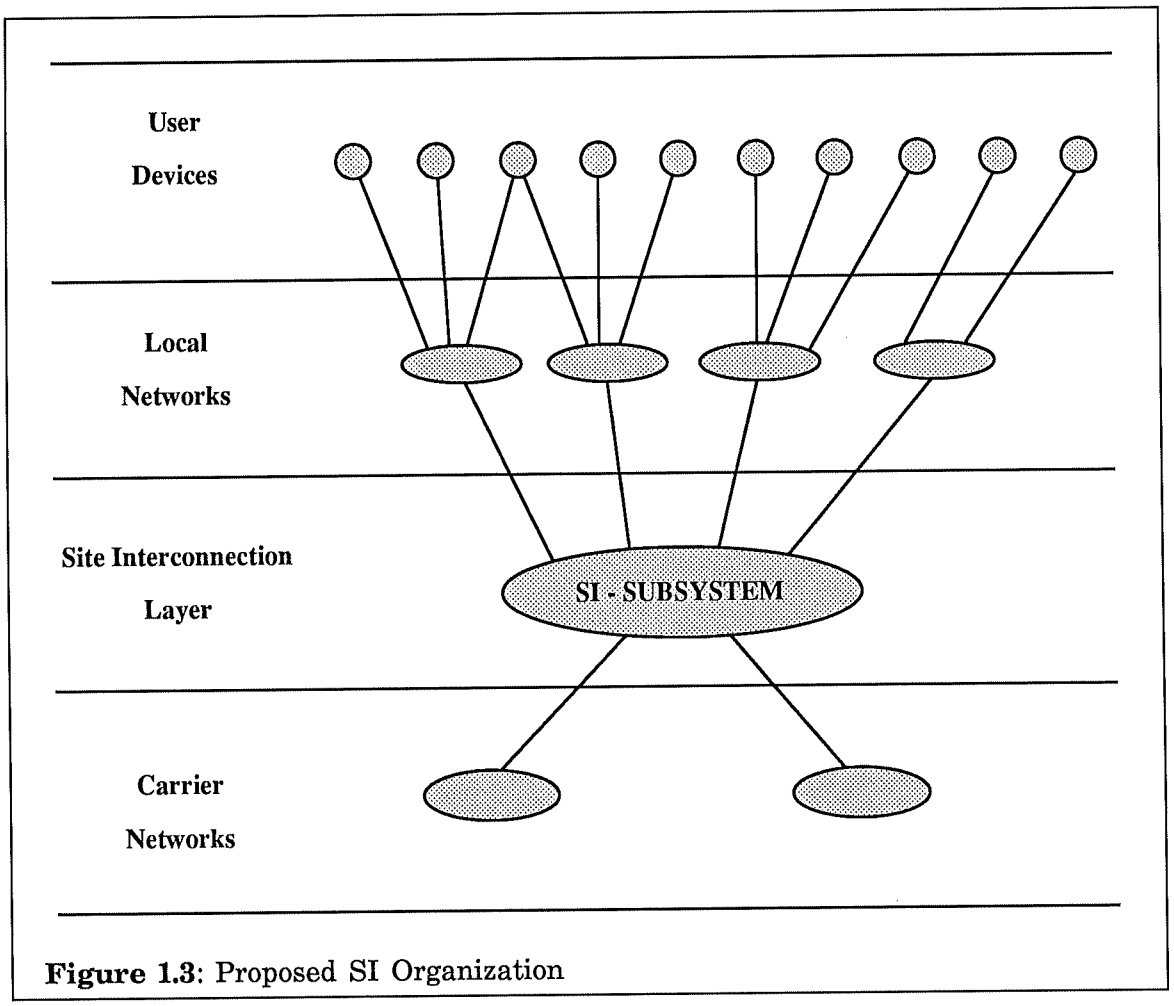

Figure 1.3: Proposed SI Organization

layer. The site interconnection layer (SI-layer), inserted between the local and carrier layers, facilitates communication between peer local networks. This layer is composed of independent subsystems that offer the site interconnection service (SI-service) to their upper layer clients. The subsystems within the SI-layer incorporate management components that facilitate addressing, routing, bandwidth management, security, accounting, etc.

The layer boundary between a site's local networks and its SI-subsystem is a funnel through which various device symbol sets are encoded into a common digital format. Within each subsystem, the independent symbol streams of concurrent upper layer associations are multiplexed together for presentation at a limited number of common carrier points of attachment as illustrated in Figure 1.3.[1]

[1]Each SI-subsystem also supports intra-site communication amongst similar, but otherwise disjoint, local networks.

Upper Layers

The SI-layer imposes few restrictions on the operation of its upper layer clients. The local distribution systems are based on symbol sets and interfaces that are compatible with device and end user requirements. Although the SI-service described in this dissertation relies on a digital encoding, there is no requirement that digital techniques be embedded within every device attached to a local network. For example, analogue telephones and video stations can be used in conjunction with digital encoders adjacent to the gateway between the local network and the SI-layer. Over time it is expected that the standardization process will result in the survival of only a small number of different local networks, thereby limiting the complexity of the boundary between the interconnection layer and its clients. Additional local networks will evolve for the attachment of devices that directly support the site interconnection encoding and for the support of new styles of communication.

Lower Layers

The SI-encoding used within the interconnection layer is independent of any one type of local or common carrier facility. However, below the SI-layer, different encodings can be used to support communication over a variety of carrier networks and transmission systems. The transparency of the site interconnection strategy improves the prospects for universal access[1] and facilitates the introduction of new technology. Each site's carrier configuration can be customized to suit particular traffic, geographic and environmental circumstances. Evolutionary changes within a site's configuration and within the individual carrier networks are transparent to the upper layer clients.

Physical Sites

Although the hierarchical model is somewhat abstract, the direct correspondence between SI-subsystems and distinct physical sites provides a more concrete frame of reference. Within the context of this dissertation, a site is a geographic location from which a community of users access telecommunication services. Typically a site is administered by a single organization such as an operating company, landlord, government, university, or hospital, and the members of the community share common spheres of interest. These common interest, geographic, and administrative bonds distinguish *sites* as appropriate domains for local area networks and shared common carrier points of attachment.

---

[1]Peer local networks can use the SI-layer to support communication even when the peer sites are serviced by different common carrier networks. Connectivity may be provided by carrier network gateways operating within the lower layers or by relay entities operating within the interconnection layer.

The evolution of carrier distribution networks is governed by the inertia arising from the vast size of the network and the time required to implement changes on a universal basis. The site interconnection architecture addresses this problem by reducing the number of carrier points of attachment and by targeting user communities that can derive immediate benefit from the availability of new service offerings.

Consider the case of a building, partitioned into separate offices leased by independent organizations that install local networks within their own premises. Rather than extending the common carrier distribution network throughout the building the landlord can provide a site interconnection facility. In this manner, a single common carrier point of attachment can service a large number of tenants. The high capacity shared service can be more responsive to the dynamic variability in individual user requirements than an equivalent number of low capacity private attachments.[1]

1.4 Original Contribution and Points of Investigation

The thesis supported by this dissertation is that the proposed site interconnection architecture represents a superior approach to the provision of flexible telecommunication services. The site interconnection service provides for evolution of the transmission substrate and the integration of a wide range of local network services through:

- The organization of the SI-layer into independent subsystems operating at distinct physical sites;
- The adoption of a common SI-layer encoding that facilitates the universal exchange of information between peer entities operating within these SI-subsystems;
- The use of flexible multiplexing and switching techniques that support the multi-service nature of SI-layer communication and enable each subsystem to process the traffic arising from its collection of local network and common carrier access points; and
- The provision of a management structure that supports the independent administration of distinct SI-subsystems.

---

[1]In the replacement of copper wiring with optical fibres, a large fraction of the cost is due to labour rather than the cost of the fibre or its termination. It is more economic to install a shared fibre of high capacity than to install a number of fibres operating at relatively low speeds.

In support of this thesis the following issues have been investigated and are described in this dissertation:

- The use of Asynchronous Transfer Mode (ATM) techniques as the basis of an SI-service. The ATM approach is characterized by the use of small packets in conjunction with switch fabrics that provide comparable bandwidth to circuit switching and permit much greater variability in traffic patterns;
- The use of the Cambridge Fast Ring (CFR) packet structure as the basis of the SI-encoding and the use of CFR technology in the implementation of the individual SI-subsystems. In this application, the CFR provides ATM-based switching and multiplexing facilities;
- The design of *out of band* management services that support the interconnection of peer SI-subsystems; and
- The experimental implementation and analysis of a pilot SI-service that operates over an ISDN network.

1.5 Outline of Dissertation

This dissertation can be divided into two parts. The first part provides background information and develops the connected argument supporting the site interconnection service. The second part describes the experimental work related to the development and analysis of a pilot SI-service.

Chapter two describes some of the styles of communication that can be expected within a multi-service interconnection environment and identifies the prominent characteristics of that traffic. This is followed by a general description of digital networks and a review of some of the techniques used to implement existing local and common carrier networks. Chapter four reviews previous work concerning site interconnection and lays the groundwork for Chapter five which identifies the principal site interconnection issues and develops the SI-service description.

The second part of the dissertation begins at Chapter six which describes the *exchange architecture* that is the basis of the pilot implementation. This is followed by four chapters describing the principal SI-subsystem components: the CFR-based switches; the *ramp* interfaces to the lower layer carrier networks; the *portal* interfaces to the upper layer local networks; and the management services that coordinate the operation of the independent subsystems. A number of experiments were performed using the pilot implementation and a summary of the results and the corresponding analysis is presented in Chapter eleven.

In conclusion, Chapter twelve presents a summary of the work done, reviews the insights that have been gained, and presents a number of recommendations for further research.

Except where otherwise indicated, the work described in this dissertation is my own. The design of the site interconnection layer and the use of the CFR as a switch fabric are original aspects of this research. Similarly, the experimental programme, the analysis, and the discussions are my own efforts. The *exchange architecture* was originally proposed in [Tennenhouse 85], and a number of individuals made significant contributions to its development. Much of this consultation took place in a series of meetings with Prof Roger Needham and Ian Leslie of the Computer Laboratory, and John Burren, Chris Cooper, and Chris Adams of the Rutherford Appleton Laboratory (RAL). Many other individuals assisted in the detailed design and implementation of the pilot exchange implementation. In particular, many of my colleagues within the Computer Laboratory contributed to the design and construction of the CFR-based local exchanges.

A great deal of this research was performed within the context of Project Unison [Clark 86], which is a joint research effort between the Computer Laboratory, the RAL, Loughborough University of Technology (LUT), Logica CES Ltd, and Acorn Computers Limited. Although the primary thrust of Unison is in the area of multi-media office applications, the project has adopted the exchange architecture as the basis of its site interconnection infrastructure. The pilot implementation could not have been developed without the considerable support and funding provided by the Unison principals.

Chapter 2
Digital Communication Services

The different styles of communication present in the multi-service network environment impose a variety of upper layer encodings and traffic characteristics on their supporting communication services. This chapter identifies the attributes of digital communications services that are of particular importance to the site interconnection layer.

2.1 Associations

Each occasion of communication is based on the exchange of symbols between *associated* peer entities. This exchange is normally structured into message units that reflect characteristics of the user encoding. The structure may be hierarchical and at the uppermost level of abstraction an entire occasion of communication, such as a telephone call, can be viewed as a single message unit. At a lower level this message is composed of individual *talk spurts* each of which consists of pauses and phrases that can be described in terms of more elementary units. Other examples of message units are pages of text, actions within a transaction, still video images, and individual frames of full motion video.

Associations are based on either interactive or bulk exchanges of encoded symbols. Interactive communication is characterized by the exchange of a series of message units, such as talk spurts. Bulk traffic can be unidirectional and may not involve a direct association between the peer entities.[1]

This dissertation is primarily concerned with the interactive traffic that arises from communication between devices, such as computers, and from the encoding of various styles of human communication. The many different styles of communication and encoding techniques lead to a wide range of demands on

[1]Bulk traffic, such as electronic mail, is often relayed through associations with intermediate agents.

communication services. Some encodings yield evenly distributed loads based on continuous fixed rate symbol streams. Other encodings present lumped loads arising from symbol bursts whose arrival patterns vary in terms of burst duration, density, and frequency.

Bulk traffic often arises from interactive communication that is locally captured and then further encoded for bulk transfer. The transfer encoding may be based on layered communication protocols[1] that provide the *envelopes* used to convey captured messages. An interactive encoding can be directly embedded within this envelope without further encoding or conversion.[2] Some researchers[3] have experimented with multi-media presentations that combine various styles of communication such as voice, text, and images. The presentations are captured interactively, transferred as bulk data, and replayed and edited interactively.

2.2 Digital Communication

At one or more levels of a communication hierarchy the symbols exchanged between peer entities may be encoded as sequences of digits. The results of this digital encoding process, referred to as digitization, are often expressed as sequences of binary digits. These encodings are naturally suited to communication between digital computers which are designed to process and store digitized information.

Digital encodings can also be used to convey a vast array of telecommunication services such as telephony, television, and facsimiles. In these cases the continuous analogue signals of the upper layer encodings are sampled at discrete points in time. At each sampling point, an analogue to digital converter is used to approximate the amplitude of the signal with one of a finite number of discrete values. The resultant digital encoding is a sequence of symbols that correspond to the discrete amplitude values. Fixed length *packets* of binary digits, such as octets,

---

[1]Data communication protocols can be viewed as layer encodings that have somewhat complex rules governing the exchange and interpretation of symbols on behalf of upper layer entities. In this dissertation the term *protocol* is sometimes used to refer to the rules themselves rather than the details of the coding.

[2]For example, the CCITT and ISO Message Handling System [ISO 9065] provides a bulk transfer service that supports electronic mail applications. This service embeds messages arising from voice, facsimile, videotext, and other styles of communication within standard envelopes. The embedded messages conform to media-specific encodings that normally support interactive communication.

[3]For example, [Thomas 85] and [Postel 82].

may be used to represent the individual symbols within the sequence. Once a signal has been digitally encoded the resultant stream of bits can be transmitted to a peer entity where, if necessary, it can be decoded back into analogue form. In order to accurately reproduce the analogue signal the timing relationship between digital samples must either be preserved during the communication process or regenerated by the peer entity.

One of the attractions of digital communication is that once a signal has been encoded it can be transmitted with little or no distortion. The digital signal can be accurately regenerated and so degradation arising from noise, attenuation, cross-talk, etc. does not accumulate as the signal passes through a concatenated series of transmission facilities. In digital communication the fundamental loss of quality arises from the initial digital approximation. The quality of this approximation depends on the accuracy of the sampling process and the intervals between samples.

Digital encodings have additional advantages in the areas of facility sharing and service integration. Facility sharing arises from the observation that if different styles of communication are encoded into similar digital bit streams then it is possible for different services to share a single communication service. In many instances one of the communicating entities may be a digital computer acting as a sink or source of information. Service integration between computers and end user devices is greatly simplified when standardized digital encodings are used to represent different styles of communication. For these reasons, the SI-service described in this dissertation is primarily concerned with the transfer of digitally encoded communication.[1]

Sampling and Encoding

Digitization processes can be divided into wideband and narrowband techniques. Wideband processing is based on regular sampling of the analogue signal and results in a *continuous* fixed rate digital signal. Narrowband processing makes use of particular characteristics of the analogue signal to adjust the sampling interval and/or accuracy to produce a *variable* rate bit stream. The two techniques can be combined. For example, Pulse Coded Modulation (PCM) can be used to convert the analogue signal arising from one end of a telephone conversation into a continuous stream of octets. Narrowband processing can then be used to eliminate silent

---

[1]In the remainder of this text, the *digital* prefix is normally implied when terms such as encoding, transmission, communication, etc. are used.

intervals from the PCM signal, leaving a variable rate digital signal consisting of distinct bursts of PCM octets. Similarly, analogue video frames can be digitized to yield a very high rate continuous stream, and this stream can be digitally processed to produce a variable rate encoding based on the differences between successive video frames.

Narrowband techniques take advantage of the structure of the end user communication. The resultant variable rate stream reflects discontinuities within this structure, and so there may be a substantial difference between the overall average bit rate and the peak rates generated during bursts. In comparison, wideband techniques ignore the structure of the communication: to achieve comparable quality the continuous bit rate of a wideband encoding must be greater than the average bit rate of an equivalent narrowband encoding.

Traditionally wideband encodings have been popular because they are simple to implement and present a known, fixed demand on the communication service. In multi-service environments, such as the site interconnection layer, a variety of encodings must be supported and the fixed demand attributable to any single encoding is of less significance. The lower average bit rates achievable with narrowband techniques make them particularly attractive for the provision of high bandwidth services such as video transmission. However, wideband encodings will continue to be popular in low bandwidth applications, such as PCM voice, where a standard encoding has gained widespread acceptance.

Temporal Coherence and Synchronization

In many styles of communication it is essential to maintain appropriate timing relationships in order to retain temporal coherence between the communicating parties. This can be especially important in the case of interactive communication, such as telephony, where pauses between message units convey useful information. For example, a speaker may pause between talk spurts to await a brief acknowledgement. When a satellite channel is used the long delay through the network lengthens the pause apparent to the speaker to the point where the speaker may find the silent period uncomfortable or continue without the acknowledgement. Multi-media presentations suffer from the additional complexity of maintaining temporal coherence across a variety of media.[1]

A lower level aspect of temporal coherence arises from the observation that digital sampling takes place at discrete instants in time. With *asynchronous* styles of

---

[1]For example, the synchronization of speech with video motions.

communication, such as the exchange of telex keystrokes, there is no need to retain information concerning the elapsed period between samples. In fact, the ability to perform *rate adaption*, i.e. to support communication between peer devices and individuals that operate at different rates, is an important feature of some communication services. In other styles of communication the timing relationship between samples must be reproducible. One approach, applicable to narrowband encodings, is to encode timing information within the symbol stream conveyed from source to sink. An alternative approach, used in *synchronous* wideband encodings, is for the source to sample on a regular basis at an agreed standard frequency while the sink uses a local oscillator, operating at the same frequency, to reproduce the timing information. Interactive traffic arising from such an encoding is referred to as *isochronous* traffic.[1]

2.3 Performance Considerations

Each style of communication imposes its own performance requirements on the facilities used to transfer symbols between communicating entities. The following paragraphs classify these diverse requirements in terms of four attributes of importance to site interconnection: bandwidth; delay; variation in delay, referred to as *jitter*; and fault tolerance. The discussion of each attribute identifies the requirements imposed by different styles of communication and the specific issues that must be considered in the evaluation of alternative approaches to site interconnection.

Bandwidth

Different encodings are characterized by wide variations in their transmission bandwidth requirements, both in terms of peak transfer rates and in terms of burstiness.[2] With the exception of video services, the bandwidth requirements of most existing encodings are within the range of planned or available services, such as the IEEE 802 local networks and the primary rate ISDN carrier service. In the site interconnection environment the important bandwidth issue is not the support of a single occasion of communication but the ability to support concurrent

---

[1]Pure synchronous encoding represents the special case when there is no variability in the sample arrival process. In practice the distinction is a matter of degree. For example, during a telephone call, a PCM encoder is expected to produce octet samples at a nominal rate of 8 Khz. This traffic is considered to be isochronous even though the individual sample periods may vary within specified bounds.

[2]The ratio of peak to average bit rate.

associations of various types by taking advantage of the bursty nature of many encodings.

Video traffic is characterized by its appetite for raw bandwidth. Furthermore, users familiar with broadcast video formats will expect the SI-layer to support a similar or superior quality of service. Analogue broadcast signals require between 4.5 Mhz and 6 Mhz of bandwidth and, using relatively simple encodings, digitized video signals have peak bit rates of 32 to 45 Mbps. Using more extensive encoding techniques these signals can be compressed further, though the cost of the necessary coding equipment may offset the savings in transmission bandwidth.[1] Although the equipment cost can be expected to decline in the future, the introduction of high resolution video services is likely to increase the analogue signal rate to about 20 Mhz thereby negating much of the saving. Given that the costs of encoding and transmission equipment are declining concurrently, it is difficult to determine the optimal bit rate for video services. The current consensus within the CCITT [Minzer 87] appears to be that basic video will be encoded at approximately 35 Mbps and high quality video at about 140 Mbps.

Delay

The maximum delay tolerated by communicating entities is determined by the upper layer protocols that are used to structure their exchange of encoded symbols. In general, bulk transfer clients can tolerate substantial delays and their end-to-end exchanges are often pipelined so that overall throughput is not directly dependent on the transfer delay.[2] Interactive communication is considerably more sensitive to delays since each interaction between associated peer entities involves at least one round trip. In the case of telephone conversations, for example, the upper layer protocol is operated by the individual participants and the maximum tolerable delay is somewhat subjective. It is known that the delay of 250 milliseconds imposed by a satellite channel is uncomfortably long and a site interconnection delay target of 25-50 msec would appear reasonable.[3]

---

[1][Eguchi 87] reports user satisfaction with a hybrid system that uses: an analogue encoding to support individual sites; a 32 Mbps encoding within metropolitan areas; and 6.3 Mbps and 1.5 Mbps encodings over longer distances where the cost of the coding equipment is offset by the savings in transmission facilities.

[2] However, many protocols, such as the ARPA Transmission Control Protocol (TCP) and the OSI Connection oriented Transport Protocol (TP), operate peer acknowledgement and flow control procedures that effectively constrain the pipeline capacity in the presence of lengthy round trip delays.

[3]More stringent delay requirements are sometimes imposed to limit the impact of echoes that arise when two wire analogue facilities ares used within a telephony circuit.

New styles of interactive traffic arise within distributed computing systems that utilize Remote Procedure Call (RPC) protocols to support transactions between components. These protocols lead to bursty traffic patterns where most of the messages are relatively short. Since each RPC interaction may involve a number of serialized transmissions, the overall transaction throughput is directly dependent on the end-to-end delay between the peer entities. In the local network environment this delay is on the order of a few milliseconds. Transfer delays must be kept to a minimum if the benefits of distributed computing are to be extended through the SI-service.

Jitter

The CCITT definition of jitter [CCITT G.701] is: *"short-term non-cumulative variations of the significant instants of a digital signal from their ideal positions in time"*. Isochronous traffic, such as PCM voice, depends on the smooth flow of octet packets from the encoding source to the decoding sink. A fresh packet is expected at the decoder during every period of a local oscillator operating at a pre-determined frequency. Short term variations in the delay through the transmission facility induce jitter in the incoming signal and interfere with the decoding process.

The jitter sensitivity of the PCM encoding is the reason the present day digital telephone system provides a synchronous transfer mode (STM) service. In STM environments, the communication service resembles a pipeline with exactly one packet accepted at the source and one packet delivered to the sink during each sampling interval. This arrangement means that all of the STM components must operate at exact multiples of the sampling frequency. In the case of telephony, considerable effort has been expended in the specification [Hall 79] and minimization [Kearsy 84] of jitter within the network. In practice many encodings are sensitive to variations in delay and in this text the term *jitter* is used, somewhat informally, to refer to the variation in delay experienced during individual occasions of communication.

STM services address the jitter problem by ensuring that the variation in delay experienced by service users is much less that the sampling interval of their encoding. The difficulty with this approach is that it ties the communication service to a specific sampling rate and is of limited value in multi-service environments where a variety of fixed rate and variable rate encodings must be supported.

An alternative approach is to provide an asynchronous transfer mode (ATM) service and to improve the jitter tolerance at the individual service access points (SAPs) by introducing compensating filters between the service and its clients. Different compensation techniques can be used to support different styles of communication.[1] In most data applications the upper layer protocols are sufficiently tolerant of jitter that the ATM service can be used directly without jitter compensation.

Jitter compensation increases the complexity of SAPs[2] and the average absolute delay between peer client devices. The additional complexity is a function of the maximum jitter tolerated and so there is a trade-off between the maximum jitter through an ATM service and the global cost of jitter compensation. There is a similar trade-off between ATM jitter and delay. For a given association the elastic delay must be less than the acceptable absolute delay and so, in practical ATM environments, the expected jitter must be constrained. In terms of the absolute delay limits previously discussed this implies that the maximum jitter through the SI-service should be measured in milliseconds rather than tens of milliseconds.

Fault Tolerance

A further characteristic of encodings is their ability to tolerate transmission faults that result in the incorrect or incomplete delivery of messages. Service users recover from these failures through selective retransmission. For example, an individual listening on a telephone will ask his correspondent to repeat talk spurts that suffer severe distortion. In the case of data and bulk transfer applications, where error-free communication is of importance and retransmission delays can be tolerated, upper layer protocols support the automatic detection and retransmission of corrupted or lost messages.

For some interactive services, such as voice and video, retransmission interferes with temporal coherence and is only invoked when the integrity of the entire association is at risk. On the surface this resistance to retransmission would

---

[1]A suitable scheme for use with PCM encodings is described in [Ades 86] where an *elastic buffer* is proposed. The nominal size of the buffer is equivalent to the expected maximum jitter and at the initiation of an association the buffer consumes samples until it expands to this nominal size. Thereafter samples from the buffer are presented to the decoder on a FIFO basis and at the pre-determined sample rate. When jitter results in the late arrival of incoming samples the buffer contracts and when samples arrive early it expands. So long as the jitter imposed by the network remains within the expected bounds, the elastic buffer ensures that a steady stream of samples is delivered to the decoder.

[2]Although the introduction of buffers into a telephone may only result in a marginal increase in the cost of each unit, a very large number of units will be built.

appear to require that transmission services supporting voice and video traffic must have extremely low error rates. Fortunately the styles of interaction and the encodings used provide a considerable degree of fault tolerance. Many of the words of a conversation or pixels of a video image have a limited lifetime during which they are of consequence. This is especially true of wideband encodings, such as intra-frame video, where the lifetime of a pixel within a frame is only 16-20 milliseconds and, even if an entire frame is lost, it will hardly be noticed by a human observer. Similarly, subjective studies [Gruber 83a][1] have shown that, during talk spurts, perceived speech quality is barely affected by the loss of one percent of all PCM samples provided the duration of individual outages is limited to 4 milliseconds. Longer loss durations of up to 50 milliseconds do not impair speech intelligibility to the point of triggering retransmission.

In summary, fault tolerance is a common characteristic of many styles of site interconnection traffic. The ability of the end users to recover from the loss of individual messages allows services to discard traffic in the event of transient overloads or transmission failures. Jitter compensation filters, such as the elastic buffer of the previous section, can take advantage of this characteristic to enforce absolute jitter margins through the judicious deletion of delayed samples.

2.4 Summary

Traditionally data traffic has been carried by communication services that have been designed to tolerate the bursty nature of data encodings and isochronous traffic has been carried by synchronous services that are tuned to handle specific sampling periods. The site interconnection environment is characterized by the mixture of traffic arising from bursty, variable rate and fixed rate encodings.

The discussion of performance considerations has highlighted the requirement for raw bandwidth and the susceptibility of interactive traffic to delay and jitter. In this dissertation it is assumed that the fundamental constraint on transmission bandwidth is dictated by the choice of transmission facilities and that alternative site interconnection architectures all try to deliver a significant fraction of the available bandwidth to their end users. On this basis, throughput is not the appropriate measure for the comparison of alternative proposals. Although propagation of signals through the transmission facilities is a fundamental source

[1]Also, [Gruber 83b].

of delay, different approaches to SI-layer implementation can significantly increase the delay and jitter experienced by communicating peer entities. Thus jitter, and to a lesser extent absolute delay, are the important metrics for the comparison of alternative site interconnection architectures.

Chapter 3
Digital Networks

The hierarchical model of telecommunication services, based on the layered application of coding and transmission, is sufficient to describe communication between a single pair of peer entities. In extending the model to describe larger communities it is important to distinguish between broadcast communication, involving large numbers of entities, and narrowcast communication in which only a fraction of the community participates in a given occasion of communication.[1]

The model can easily be extended to describe broadcast communication by allowing a single medium to be shared so that every entity can participate in every exchange of symbols. Broadcast communication is particularly attractive when communication is unidirectional, i.e., there is a single source of symbols and a very large number of recipients. This is the case in the provision of commercial broadcast services such as radio and television.

Although individual occasions of narrowcast communication may involve only two parties it is desirable that each entity be capable of communication with many other peer entities through concurrent or sequential occasions of communication. Narrowcast communication can be supported through exhaustive replication of the transmission medium. While this might represent an ideal communication environment it is not altogether practical and does not take account of differences in distance or communication patterns between peer entities. Practical narrowcast communication is accomplished through the implementation of *networks* that support narrowcast services through the implementation of *multiplexing* and *switching* functions that operate in conjunction with layered coding and transmission.

---

[1]In this dissertation the term *narrowcast* usually denotes the common case where only two entities participate in each occasion of communication.

The first section of this chapter presents an overview of digital network organization with emphasis on the layered structure of networks and the implementation of multiplexing and switching functionality. The network attributes identified are relevant to the organization of the site interconnection layer itself and also to the understanding of both the upper layer local network clients and the lower layer carrier transmission services. Accordingly, the latter sections of this chapter describe the organization of local and common carrier networks in terms of these attributes. The organization of the SI-layer will be described in Chapter 5.

3.1 Network Organization

Multiplexing and switching are the principal functions that facilitate the construction of digital communication networks. Multiplexing partitions the communication *channel* provided by a transmission service into a number of logical *subchannels* each of which supports communication between peer entities. Switching allows a single channel or an entire collection of channels to be dynamically configured so that, at specific and controllable instants in time, signals can be exchanged by selected peer entities.

A digital communication network can be modelled as a mesh of switching nodes interconnected by multiplexed transmission services.[1] Each network user is attached to the network at one or more service access points (SAPs) and upper layer transmission is supported on a *bearer channel* that provides a path between the appropriate peer SAPs. This channel is supported through the concatenation of subchannels linking the appropriate nodes. The network *routes* individual associations by arranging for the switching nodes to relay symbols on behalf of the peer entities. The bearer channel provides a simple transmission service which can be embedded within some upper layer network.

3.1.1 Layered Network Services

In general, telecommunication networks can be hierarchically modelled in terms of coding, transmission, multiplexing and switching. At a given layer in the hierarchy, transmission may be based on communication between the entities of a

[1]This model encompasses the degenerate cases where the entire network consists of a single node or a single transmission medium.

lower layer. This communication can in turn be modelled by some combination of coding, transmission, multiplexing, and switching. The overall telecommunication environment is structured into a hierarchy of networks operating at various levels of abstraction.[1]

Each network supports the exchange of encoded symbols between the user entities attached to its SAPs. The symbols are represented by sequences, or packets, of binary digits where the values, lengths, and intervals between packets are functions of the encoding. The network transmission service may itself be realized through a hierarchy of network layers. The end user symbols will be encoded and decoded as they are passed across layer boundaries, and each encoding process involves the mapping of the packets of the upper layer into the packet format of the lower layer.

Relationship to OSI

The hierarchical network model is a generalization of the layered architecture developed in the ISO Reference Model of Open Systems Interconnection [ISO 7498-1]. The OSI model describes the communication process in terms of the operation of an arbitrary (N)-layer that supports the exchange of service data units (SDUs). In general, multiplexing and transmission functions may be performed within the (N)-layer.

The encoding of packets is modelled as a two step process. Client (N)-SDUs are mapped into the protocol data units (PDUs) that are exchanged between peer entities operating within the (N)-layer. These are in turn mapped into (N-1) SDUs that are presented at lower layer SAPs for transmission between peer (N)-entities. Individual (N)-PDUs transfer two types of information: protocol control information (PCI) conveys symbols that govern the communication between the peer (N)-entities; and user data conveys the individual bits of the packets being exchanged between the client (N+1)-entities. In effect, the binary digits of (N)-SDUs are embedded within user data carried within (N)-PDUs. Three mappings between (N)-SDUs and (N)-PDUs are distinguished:

- one to one mappings in which exactly one (N)-SDU is carried within each user data field;
- one to many mappings in which each (N)-SDU is *segmented* into a number of shorter packets which are carried within separate (N)-PDUs; and

---

[1]This is the view implicitly adopted in [CCITT I.412] where it is acknowledged that the switched bearer circuits delivered by an ISDN network can be used as the physical layer transmission medium of an X.25 network.

- many to one mappings in which consecutive (N)-SDUs are *blocked* together so that their contents are carried in a single (N)-PDU.

The need for segmentation and blocking arises from incompatibilities between the sizes of the packets exchanged at different layers. Segmentation is necessary when complete upper layer packets cannot be fully embedded within individual lower layer packets. In contrast, when a number of upper layer packets can be embedded within a single lower layer packet, blocking may be used to reduce the number of packets exchanged by lower layer entities.

Segmentation and blocking may occur at a number of logical layers within a communications environment. Furthermore, within a single layer, the bearer channel utilized by the entities of an upper layer may be constructed through the concatenation of subchannels that utilize different packet structures. In such cases, each of the relay points along the concatenated route represents a discontinuity at which it is necessary to convert between packet structures by reassembling (or deblocking) incoming packets and segmenting (or blocking) them to suit the outgoing subchannel. This processing increases the cost of the relay elements and introduces additive elements of delay and jitter into the bearer channel.

Network Transfer Mode

The transfer mode of a network has a considerable impact on the manner in which it supports different types of upper layer encodings. The OSI architecture was originally designed to support the open interconnection of computer systems and devices supporting traditional data communication applications. Many layers of the architecture implicitly assume a conventional packet switched transfer mode (PSTM) in which relatively large packets are asynchronously switched between peer SAPs. The service supporting client associations may be *connectionless* in which case each packet is dynamically and independently routed through the network based on the examination of the packet contents[1]. Alternatively the service may be *connection-oriented*, in which case packets are logically related to each other and are sequentially transferred in a pre-determined manner.[2]

In the analysis of multiplexing and switching that follows it will be shown that the asynchronous PSTM approach leads to a considerable degree of delay and jitter as packets are transferred through the network. The principal source of jitter is the

---

[1]Typically, only pre-determined address fields within the packet are examined.

[2]Although fields within a packet may still be examined, these fields are likely to be relatively short identifiers that bind the packet to the pre-arranged connection.

contention that develops at the switching components within the network. Since the packet lengths may be relatively long a collision with a single packet may induce a significant jitter element. Furthermore, the PSTM approach tends to involve nested segmentation and blocking functions which are expensive to implement and contribute further delay and jitter to the exchange of user symbols.

In contrast, the synchronous transfer mode (STM) described in the previous chapter rigorously constrains the jitter induced through the network. Furthermore it is possible to construct integrated STM networks in which packets are directly transmitted between end user digitization points without segmentation or blocking. The difficulty with this approach is that an STM network service is tuned to the support of a single encoding such as PCM voice, and so this transfer mode is not appropriate to the site interconnection environment.

The asynchronous transfer mode (ATM) represents a compromise between the conflicting requirements of network and traffic integration. This connection-oriented transfer mode depends on a common encoding, based on relatively small packets, that is suitable for many different styles of communication. The asynchronous nature of ATM provides the flexibility to support a wide variety of end user encodings whilst the common packet format facilitates the integration of switching and multiplexing.

Application to Site Interconnection

The site interconnection layer can be modelled as a loosely coupled network with:

- Switching nodes based on SI-subsystems located at independent sites; and
- Multiplexed transmission channels supported by lower layer common carrier networks.

The users attached to this *network* use the SI-service to support associations between peer local networks. At an even higher layer these SI-associations are concatenated together with the bearer channels of the local networks to support associations between client devices attached to different local networks.

A principal objective of the SI-layer is the use of a common digital encoding throughout all of its multiplexing and switching components. In practice this objective is difficult to realize. However, it is possible to construct ATM networks in which compatible packet formats are used: within all of the SI-subsystems; over all of the multiplexed transmission media; and at all of the SI-SAPs.

3.1.2 Multiplexing

There are two distinct, but related, levels at which multiplexing is used within communication networks. When a number of network nodes are interconnected by a common transmission medium, multiplexing techniques allow the single transmission channel to concurrently support a number of independent associations between peer nodes. When an individual node supports a number of concurrent associations the node can be modelled as a collection of entities. At this level multiplexing techniques allow the entities within the node to share a single point of attachment to the medium.

A number of multiplexing schemes have evolved to support different styles of communication. In frequency division multiplexing (FDM) the communication channel is divided into a number of different frequency bands each of which can be used to support a distinct occasion of communication. FDM is suited to the multiplexing of bandwidth limited analogue signals since each of the frequency bands can be used to transmit a distinct analogue signal. Time division multiplexing (TDM) is a complementary scheme whereby the time domain of the channel is divided into separate intervals referred to as timeslots. TDM is suited to the multiplexing of digital signals since each timeslot can be used to transmit the sequence of symbols arising from a digital encoding.

Time division multiplexing has become the dominant form of multiplexing used within digital networks. A number of different time division techniques have been developed and they are distinguished by the formatting rules used to divide the channel into time intervals, and the access rules that govern the assignment of intervals to specific occasions of communication.

Statistical Packet Multiplexing

When statistical multiplexing is used a transmitting entity can begin a transfer whenever the transmission channel appears idle. The start of the transmission defines the beginning of a TDM interval which, in general, continues until the transmitter relinquishes the channel. Typically a relatively long packet is transferred during each interval, and the maximum length of a packet is usually fixed so as to limit the delay experienced by other transmitters. Statistical schemes, which may be slotted[1] or otherwise, incorporate a contention resolution

---

[1]In an *unslotted* format the length and frequency of TDM intervals varies dynamically. In a *slotted* scheme the channel is formatted into fixed length intervals.

mechanism that ensures serialized access to the channel.

Synchronous Time Division

A synchronous frame structure can be imposed on a channel to format it into a regularly repeated sequence of fixed length *timeslots*. Each frame is of fixed duration and consists of a start of frame marker followed by a fixed number of timeslots. During every frame interval each timeslot can be used to transfer one packet of digital symbols. Frame and timeslot arrival processes are synchronous due to the regularity of the intervals between significant instants, i.e., the start of frame markers, and the fact that the individual timeslots appear at fixed offsets within each frame.

Synchronous time division (STD) occurs when the recurrent instances of a given timeslot are dedicated to the support of a specific association. The frame structure divides the channel into a number of synchronous subchannels each of which transfers fixed length symbol packets at regularly repeating intervals. A *signalling* mechanism can be used to assign individual timeslots for the duration of each association.[1]

A significant limitation of STD techniques is that they do not support the dynamic adjustment of the bandwidth assigned to specific associations. Individual implementations are normally tuned to support a single synchronous encoding: the frame repetition rate is matched to the sampling frequency; and the timeslot length corresponds to the packet length of individual samples.[2] A further limitation is that the number of timeslots in the frame limits the number of concurrent associations that can be supported.

Asynchronous Time Division

In asynchronous time division (ATD), the recurrent timeslots of a frame structure are dynamically assigned to associations on a frame by frame basis. In contrast to STD, there is no fixed relationship between the number of bearer channels and the number of timeslots in each frame. Transmitting entities contend for the use of timeslots and the ATD implementation must provide some mechanism for collision avoidance or resolution. Each transmitted packet contains information that allows the individual receiving entities, which monitor all of the timeslots on the medium,

---

[1]Receiving entities identify incoming packets by their timeslot offsets from the start of the STD frame.

[2]For example the STD structure used in digital telephony, based on 8 Khz frames of octet length timeslots, exactly matches the sample size and interval of the PCM encoding.

to correctly identify their own incoming packets. The resultant transmission service is asynchronous because the fixed length packets, born by individual timeslots, are transferred at irregular intervals.

Strictly speaking, statistical multiplexing is a variant of ATD. In the literature the two schemes are distinguished and the term *asynchronous time division* usually refers to the combination of relatively small packets[1] with some mechanism that limits the variation in delay arising from contention. This ability to constrain jitter is central to the ATD compromise between the STD and statistical multiplexing techniques. The flexible assignment of bandwidth facilitates the integration of variable rate encodings, whilst the stringent upper bound on jitter supports the transfer of isochronous traffic.

3.1.3 Switching

The switch fabric of a digital network supports the exchange of symbols between entities attached to the SAPs of the network. In a simple switching environment the individual entities participate in serial occasions of communication. In the more general case the SAPs are multiplexed and each entity can participate in a number of concurrent associations. Switching nodes are used within networks to support the flexible concatenation of transmission channels attached to their *ports*. At a lower level of abstraction, each node is an embedded network that supports the exchange of symbols between node ports. In this context the individual ports are the SAPs of the node's internal switch fabric.[2]

Traditional Packet Switching

A simple switch can be implemented by a central processor that copies packets of symbols between the input and output ports. In effect, the processor's time is statistically multiplexed into discrete switching intervals that are dedicated to the processing of individual packets. In traditional *store and forward* switches, such as the original *arpanet* IMPs described in [Kleinrock 76], the switching process is entirely asynchronous and variable length packets arrive at the ports at irregular intervals. There is no fixed assignment of processing intervals to associations or fixed mapping between the input and output ports. The processor waits until a complete packet has been stored at a port and then, based on information

---

[1]Typically 128 octets or less.

[2]For the purposes of discussion it is often useful to distinguish the input (source) and output (destination) components of each switch port.

contained within the packet, the processor forwards the packet to the appropriate destination.

An important advantage of this style of switching is that, due to the asynchronous nature of its operation, it can perform *rate adaption* between users and/or channels operating at different speeds. Packets arriving at a port attached to one channel can be distributed to ports attached to other channels operating at higher or lower rates. An unfortunate consequence of the manner in which this rate adaption is achieved is that it is possible for the source ports of a switch to supply packets faster than they can be transmitted at a target destination. Contention for a given output port should not be resolved by throttling, or *back-pressuring*, all of the input ports since this would unnecessarily delay traffic bound for other destinations. A common approach is to use upper layer mechanisms, such as flow and rate control, to limit the rate at which packets are generated by individual associations.

The traditional packet switch induces significant delay and jitter elements in the symbol streams traversing the switch:

- Individual packets experience store and forward delays as they are buffered at the switch ports; and
- Contention at the central processor and destination ports results in transient queueing delays.

Furthermore, the aggregate throughput of the switch is usually limited by the capacity of the central processor and, in many implementations, the packet processing capacity is relatively independent of packet size. Although the severity of the delay and jitter effects increase with packet length, long packets are often necessary to the support of high bandwidth associations.

Time Division Switching

There is an element of duality between time division multiplexing and switching techniques. A switch fabric can be constructed using TDM techniques to support concurrent associations between some number of ports attached to a physical medium. The switching of symbol streams is implemented by arranging for each output port to extract from the medium only those packets that have been transmitted by its peer. A central switching node can be constructed by confining the TDM medium to a small number of modules or even a single semiconductor device. Similarly, a distributed switch fabric, such as Ethernet [Metcalfe 76], is constructed by attaching discrete nodes, or stations, to a time division multiplexed co-axial cable.

Time division switching can be achieved using either STD or ATD techniques. With ATD techniques, fields within each transmitted packet identify the associated destination port. With STD switching, recurrent timeslots are assigned to the support of communication between pairs of actively communicating ports. The STD switch configuration is controlled by a network management service that determines the mapping between peer ports and timeslots.

The individual ports of a switch may themselves be time division multiplexed. A significant degree of integration can be achieved by matching the frame structures used within the switch to the frame structure used to format the multiplexed transmission channels attached to the switch ports. A further degree of integration is obtained if this structure is also compatible with the upper layer encodings transmitted over the network. For example, a fully integrated STM network can be constructed with STD switching and multiplexing structures that exactly match a synchronous client encoding. The format and frequency of client packets is maintained throughout the transmission process.

Space Division Switching

In space division switches, concurrent associations exercise different physical paths through an embedded network composed of simple switch elements and very short transmission links. The switch elements within the node route symbols between the input and output ports. Surveys of space division interconnection techniques are presented in [Thurber 74] and [Broomell 83]. A common characteristic of most schemes is that the switch fabric is based on the regular interconnection of simple cells. These cell-based topologies are particularly attractive because individual switch nodes can be efficiently implemented using VLSI technology and nodes can be cascaded together to form larger switches.

A circuit oriented approach to node configuration is to assign a fixed path to each association. A network management service configures the switch elements along the path when the association is initiated and the path remains available for the duration of the association. The physical paths, or circuits, provided in this way are of fixed bandwidth and can be used in conjunction with either analogue or digital encodings. The circuit approach is particularly well suited to the construction of STM networks provided the transmission bandwidth available through the switch is compatible with the STM encoding.

Alternatively, paths through the switch can be dynamically determined by the cascaded switch elements. A suitable scheme based on the interconnection of binary switch cells, each with two input ports and two output ports, is described in

[Hopper 79]. A sequence of routing bits appears at the start of each packet and as the head of a packet passes through a switch cell it is routed to one of the two output ports in accordance with a selected[1] routing bit. The routing decision remains in effect for the duration of the packet, and as the head of the packet threads it way between the input and output ports it pulls the remainder of the packet along with it. This approach to routing can be used in conjunction with ATD transmission to construct an integrated ATM network in which the packet structure recognized by the switching elements matches the timeslot structure used in transmission. One important limitation, arising from the regular structure and clocking of VLSI implementations, is that the individual paths of a space division switch have a fixed and common bandwidth. This characteristic extends to the switch ports and limits the ability of the switch to perform rate adaption between the ATD channels attached to its ports.

3.1.4 Management Issues

Network management provides the services which link the components of a network together to provide a useful communications environment. These services harmonize the operation of network components and are responsible for the identification, allocation, configuration, and monitoring of network resources. Management services operate along two axes:

- At a single physical location, such as within a computer system, a management service may operate across layer boundaries to coordinate the operation of adjacent entities located within different layers; and
- A management service may coordinate the operation of physically disjoint entities within a single layer.

Communication of Management Information

In the latter case, there must be some mechanism for the exchange of management information between distinct locations. This exchange of information can be implemented through the use of: *in-band* techniques, in which management communication is integrated with the transmission of end user information; or *out-of-band* techniques, in which the exchange of management information is independently supported.[2]

---

[1]Each cell along the path selects a different routing bit. This effect is sometimes achieved by arranging for the routing bits to be shifted or rotated as they are passed between adjacent cells.

[2]The term out-of-band is used within the context of a given layer. In an extreme case a network may provide entirely separate physical media that support out-of-band communication at all layers. Alternatively both the out-of-band and in-band

The in-band exchange of management information is useful in conjunction with services, such as routing, that are invoked to support individual occasions of communication. During the course of a client association, management functions can be dynamically invoked on a packet by packet basis. In this case, management information is exchanged through the embedding of management data within the packets exchanged by the in-band components supporting a client association. The cost of this flexibility lies in the increased complexity of the in-band components that must examine each packet and invoke the appropriate management functions.

When information is transmitted out-of-band, the management services can be modelled as collections of communicating entities. The management entities communicate over their own associations, that operate in parallel with those of the principal network clients. This organization is attractive because the design of the in-band network components can be streamlined to optimize the flow of client data.

Naming and Name Resolution

Within a communications environment names are used to identify and manage objects such as communicating entities, points of attachment, and switching nodes. Typically, different naming schemes will be used to identify different types of objects, and for each type of object mechanisms must be established for the structure, registration, and interpretation of names. Furthermore, the network management services must provide some interpretation mechanism that resolves entity names into paths that support communication between peer entities.

In the part of the OSI Reference Model dealing with Naming and Addressing [ISO 7498-3] a name is defined as *a linguistic construct which corresponds to an object in some universe of discourse.* Naming schemes can be distinguished by the degree to which a name approaches an *absolute* name. Absolute names are unique and identify the same object regardless of where they are pronounced, i.e., they have a global universe of discourse. In contrast, the interpretation of a *relative* name is dependent on the environment in which it is enunciated and the type of entity being identified.[1]

In a layered communications environment the interpretation of entity names may be accomplished through the step-wise resolution of the names used in one layer

---

associations of one layer may rely on the services of a common lower layer.

[1]Practical names are always related to some universe of discourse and so absoluteness is only be a question of degree.

into the names of lower layers. In the terminology of [ISO 7498-3], an (N)-title is a name that identifies an (N)-entity. Similarly, an (N)-address identifies the SAPs at which (N)-entities are attached to their client (N+1)-entities. An (N+1)-entity is located by its binding to one or more (N)-SAPs and an (N)-SAP is identified by one or more (N)-addresses. OSI name resolution proceeds in two stages:

- upper layer entities invoke directory functions to map the titles of their peers into lower layer addresses; and
- Within the lower (N)-layer these addresses are resolved into (N)-titles or directly into (N-1)-addresses.

This approach promotes layer independence as the use of (N)-titles is confined to the (N)-layer. For the most part, the distinction between titles and addresses has to do with the duration of the binding between names and objects. Titles are of long-term utility, while addresses, which are based on bindings between objects and locations, may change more frequently.[1]

Directory functions vary considerably depending on the layer at which a title is resolved. Some networks, such as Universe [Leslie 84], provide *active* services that partially or completely resolve titles into paths through the network. During the resolution process the service actively configures network elements to recognize the assigned address and perform the appropriate switching and multiplexing functions. The addresses supplied to the communicating entities are assigned to specific occasions of communication and, consequently, the name resolution service must be independently invoked in support of each association.

Other networks rely on *passive* directory services[2] that record bindings between object names, such as title, and properties, such as addresses. These bindings are relatively static and, since they change infrequently, communicating entities can retain the addresses of their peers and use them in support of subsequent associations. The static bindings of titles to addresses yields addresses that are not associated with fixed paths through the network and must be subjected to further analysis when they are presented for use.

Resource Allocation and Configuration

Network management allocates and configures resources to support client associations. This function involves the assignment of paths through the network and the configuration of the individual switching and multiplexing components

---

[1]Addresses represent the bindings between upper and lower layer entities and these bindings may be changed dynamically as entities are relocated or to suit the requirements of specific associations.

[2]Such as those described in [ISO 9594-1] and [Lampson 86].

along those paths to support the exchange of client symbols. This aspect of network management is dependent on the communication service offered by the lower layers and on the *signalling* mechanism that client entities use to invoke management functions.

In the connectionless style of communication, the service provider is not explicitly informed of the status of associations between peer client entities. Each packet contains the management information used to route the packet to its destination. This form of in-band signalling is suited to environments where the frequency of exchanges between associated entities is highly variable and, consequently, a fixed mapping between an associations and a path may not be appropriate.

When a connection-oriented service is offered, communicating entities explicitly notify management of the communication requirements of their associations. Each connection proceeds through three phases: establishment; symbol transfer; and release. A signalling mechanism is used at establishment time to provide the network management services with association-specific parameters such as the identity of the correspondents and the quality of service required. The transfer phase provides a context in which successive packets presented to the network are logically related. The separation of the transfer phase of a connection from the establishment and termination phases facilitates the use of out-of-band signalling to support the allocation and configuration of association-specific resources.

In a connectionless environment it is difficult for network management to control the overall allocation of resources and ensure that specific associations obtain the quality of service that they require. Furthermore, associations are subject to delay and jitter effects arising from the detailed and repeated examination of each packet as it traverses the nodes of the network. For these reasons, some packet-switched networks and most of the proposed ATM networks provide a *lightweight* connection-oriented service. The individual connections are sometimes referred to as virtual circuits and there is no fixed assignment of transmission or switching capacity to individual associations. At establishment time, management assigns a fixed path through the network and each of the peer entities is supplied with a connection identifier that it embeds within its packets. The switching nodes along the path are configured to process the embedded identifiers so that packets presented at one SAP are transferred through the network to the appropriate peer SAP.

Monitoring, Security and Accounting

Individual networks may provide ancillary services that are tailored to the operating environment of the network. Typically the need for monitoring, security and accounting services is greater in larger networks where it is important that these functions be automated and reliable. Monitoring services are particularly important in the geographically dispersed site interconnection environment where it would be inconvenient for maintenance personnel to physically examine all of the network components. Similarly, accounting and security services are essential if the site interconnection environment is to support communication amongst independent user communities.

Although this dissertation does not explicitly address these issues it is anticipated that these services will be provided using a hybrid approach based on the in-band operation of filtering and collection elements that are controlled by out-of-band configuration and collation components.

3.1.5 Summary

In the implementation of a given network the choice of strategies and technologies used will be dependent on traffic characteristics, such as those identified in Chapter two, and on the overall network environment including its physical dimensions and patterns of usage.

The local and common carrier environments differ significantly, and so their network architectures have evolved somewhat independently. The next two sections of this chapter describe specific examples of local and carrier networks. The architecture of each of these networks presents a particular combination of network attributes that affect the bandwidth, delay and jitter characteristics of the individual associations that will be used above, and below, the site interconnection layer.

3.2 Common Carrier Networks

Carrier networks are characterized by their large dimensions and are traditionally designed to provide universal access to a specific style of telecommunication service. An important step in the development of a universally accessible service is the

specification of the interfaces offered at the network termination points (NTEs).[1] Common specifications, that ensure compatibility between the services provided by different network operators, encourage the development of attachment equipment and facilitate the interconnection of similar carrier networks. The CCITT[2] is the primary international forum in which carrier service offerings are deliberated and specified. This section provides a brief overview of two types of common carrier networks that offer services[3] described in CCITT recommendations: Integrated Service Digital Networks (ISDNs); and Packet Mode Public Data Networks. The latter are known as X.25 networks in reference to the CCITT recommendation in which the network interface is described.

3.2.1 X.25 Networks

X.25 networks provide a PSTM service that is tailored to the support of bulk data transfer and relatively low speed computer communication. The principal advantages of the X.25 service are:

- The support of concurrent associations at the points of attachment;
- The dynamic allocation of bandwidth to associations; and
- The support of rate adaption between peer points of attachment.

The interface is specified in terms of physical, data link, and packet layers. At the physical layer a client device uses a synchronous channel to exchange symbols with the network. At the link layer this channel is statistically multiplexed to support the exchange of variable length packets arising from different packet layer associations. Elements of the link level encoding support error detection, flow control, and the retransmission of packets.

Whereas the lower layers are concerned with communication between a client device and the network, the packet layer supports connection-oriented communication between peer devices. Signalling is accomplished through the in-band exchange of distinguished packets. Although the configuration of virtual

---

[1]The points of attachment to carrier networks are referred to as the Network Terminating Equipment, and so the acronym NTE will frequently be used in place of SAP.

[2]A committee of the International Telecommunication Union (ITU).

[3]The specific configurations described in this dissertation correspond to the recommendations and parameters that have been adopted in the UK.

circuits is usually performed out-of-band, management activities are initiated when the network detects an in-band packet layer request to establish, reset, or terminate a connection. During the transfer phase of a connection, elements of the packet layer encoding are used to bind packets to virtual circuits and to provide flow control on a circuit-specific basis.

Although the X.25 recommendations do not specify the manner in which communication within a network is achieved, most implementations are based on store and forward packet switches that are interconnected by fixed rate digital transmission services. X.25 services are characterized by the substantial delay and jitter elements that accumulate as packets pass through the network. The jitter through the network has a high frequency component, that arises as a result of contention at the switch nodes and transmission links, and a further component that is induced by the retransmission functions present at the link and packet layers.[1]

3.2.2 IDN and ISDN

The Integrated Digital Network (IDN) and ISDN recommendations represent two steps along the evolutionary path towards a digital telephone network. The IDN *G-series* of recommendations describe the components of a digital backbone network that operates in conjunction with an analogue distribution network. The more recent ISDN *I-series* describes the extension of digital service into the distribution network and subscriber premises.

IDN

The IDN backbone provides a connection-oriented STM service tuned to the support of 64 Kbps PCM-encoded telephone calls. The backbone is a mesh of nodes, referred to as exchanges, that are interconnected by STD multiplexed transmission facilities. Each exchange consists of a switch fabric and some number of management processors. The exchange provides a *circuit-switched* service by assigning independent paths, or circuits, through the fabric to individual telephone calls.

---

[1]The retransmission and flow control procedures that support rate adaption and error recovery can induce transient delays that are many times larger than the average delay.

The 2 Mbps inter-exchange channels[1] are formatted to carry synchronous frames that are divided into 32 timeslots of one octet each. The 8 Khz frame repetition rate ensures that each timeslot provides an independent subchannel tuned to the basic STM service. Termination equipment, attached to each end of a channel, uses timeslot zero to exchange monitoring information and synchronization patterns that identify the start of each frame. Similarly, timeslot 16, is reserved for the transmission of signalling information between the management processors of peer exchanges. The thirty remaining subchannels support the in-band transfer of PCM octets.

In IDN, the analogue distribution network is based on voice frequency range transmission facilities that attach individual subscribers to exchanges. The subscriber service is based on an analogue encoding and signalling information is transmitted, in-band, using tones or dial pulses. At each exchange the signalling information is filtered out of the individual analogue channels and directed to a management processor. A synchronous PCM encoding is applied to the voice frequency signal to derive the 8 Khz octet stream that is fed into the exchange switch fabric. In some implementations separate *line shelves* are used to digitize a number of analogue channels and TDM multiplex them into the 2 Mbps format. This arrangement permits the use of a regular switch fabric that need only provide the standardized 2 Mbps interface. During the lifetime of each telephone call, the fabric is configured to exchange the successive packets of a given timeslot at a given interface with the packets of the appropriate peer timeslot at the peer interface.

ISDN

In ISDN, the STM service is extended to the individual subscribers and client devices use circuit-switched channels to support digital services such as PCM encoded telephony, facsimile transmission, or computer communication. Each NTE provides access to some number of 64 Kbps bearer channels and a common signalling channel. The transmission facility linking individual NTEs to the network operates at either the 2 channel *basic rate* or the 30 channel, IDN style, *primary rate.*

Signalling is accomplished through the exercise of an out-of-band protocol on the 16 Kbps or 64 Kbps signalling channel. Communication over this channel is

---

[1]The recommendations also provide for the higher order multiplexing of 2 Mbps streams between network nodes. At present the multiplexing structure is strictly hierarchical and is defined in terms of 8 Mbps, 34 Mbps and 140 Mbps streams all of which retain the 8 Khz frame repetition rate.

structured in accordance with the ISDN recommendations[1] which describe a layered organization that is similar to X.25. The data link layer of this structure supports multiplexing, and so the signalling channel can also be used to support access to an X.25 packet layer service.

Discussion

ISDN services are characterized by low delay and careful control of jitter. The delay through a network is a function of the propagation delay through the transmission media plus some number of 125 microsecond frame delays that are introduced by the packetization, switching and multiplexing processes. Although the CCITT has not recommended a specific constraint on the jitter through an ISDN, it is expected that the maximum jitter imposed on a 64 Kbps telephony signal will be on the order of one microsecond.[2]

In the ISDN environment, the bearer channels provide direct access to the high bandwidth IDN backbone and the X.25 service accessed through the signalling channel supports low bandwidth transaction and telemetry applications. The ISDN design achieves a substantial degree of network integration but little in the way of service integration. The digital links and the switch fabric of an ISDN are tuned to the provision of a specific STM service. Each connection provides a fixed bandwidth 64 Kbps transmission service and there is no provision for rate adaption or multiplexing.

3.2.3 Summary

Universal access to communication services is provided by common carriers, and the nature of each service is dictated by compromises reached within bodies such as the CCITT. A suitable site interconnection architecture must be able to operate in conjunction with presently available services, such as ISDN. Furthermore, the architecture must be sufficiently flexible to accommodate future offerings such as the broadband services that are likely to appear in the next generation of carrier networks.

---

[1][CCITT Q.920], [CCITT Q.921], [CCITT Q.930], and [CCITT Q.931].

[2][Kearsey 84] reports on some aspects of IDN jitter.

3.3 Local Networks

A variety of local network architectures have evolved to suit the requirements of different user communities. Since the dimensions of a local network are much smaller than in the carrier environment, it is reasonable for a single site to operate a number of different local networks, each providing a different style of service. Over time, different local services have evolved and a number of implementation strategies have been developed.

3.3.1 Telephone Services

Local telephone networks provide two related services: they facilitate communication between users attached to the same network; and they provide shared access to the carrier services that support communication between different sites. Due to the importance of the latter *concentration* function, the development of local telephone services has been dominated by the need to maintain compatibility with the carrier networks. Furthermore, the network usually supports transparent connections between its local users and peer users that are directly attached to the carrier distribution network, and so the basic service must be similar to that provided by the carrier.

The local network is normally based on a star topology in which the client devices are individually linked to a private business exchange (PBX)[1] consisting of a switch fabric and an attached management processor. PBXs closely resemble common carrier exchanges and the recent generation of digital PBXs use STD or space division switching to provide an STM service that is compatible with the IDN backbone. The management processors controlling these exchanges may provide signalling related services, such as call forwarding, that can be tailored to the requirements of the user community. Additional services such as accounting, security, monitoring, and routing functions support the shared utilization of the common carrier points of attachment.

---

[1]The acronym PABX is sometimes used to distinguish automated PBXs from their predecessors. In this dissertation, the PBX acronym always refers to an automated exchange.

3.3.2 Data Services

Local data networks are primarily concerned with the exchange of digital symbols between a variety of local devices. The networks are used to support diverse applications such as bulk data transfer, transaction processing, and distributed computing. Although PBX-based networks are sometimes used in this environment, the connection-oriented STM service they provide does not satisfy the full range of computer communication requirements.

Local Area Networks (LANs) are tailored to the support of communications amongst a relatively small number of devices located in a single geographic area. A number of different architectures have been developed. However, most LANs provide:

- Statistical multiplexing at the individual network SAPs. The multiplexing function permits upper layer entities within each device to participate in concurrent associations;
- Dynamic allocation of the aggregate network bandwidth in support of the bursty and variable rate encodings associated with computer communication; and
- Rate adaption functions that facilitate communication amongst upper layer entities of different capacity.

Most LAN implementations provide a connectionless PSTM service with routing performed in-band through the examination of address fields contained within the individual packets. Peer entities may operate an upper layer protocol over this service to provide applications with a connection-oriented transfer mode.

The PSTM service is often supported by a shared TDM medium that functions as a distributed packet switch. In some networks, such as the Ethernet, the medium is statistically multiplexed and directly supports packet switching. In other implementations, such as the Cambridge Ring[1], the medium is ATD multiplexed and entities operating within the peer devices segment and re-assemble individual packets for transmission within ATM timeslots.

The basic LAN service supports the exchange of packets between peer devices and, for the most part, LANs do not provide integral support for the interconnection of peer LANs. Access to peer LANs, via common carrier services, is usually supported by routing packets through distinguished devices that are attached to

---

[1]The Cambridge Ring [Wilkes 79], which is one of the earlier examples of an ATD-based network, is described in greater detail in Appendix A.

both the LAN and a common carrier network. In many cases the relaying function cannot be performed transparently and additional layers of protocol are required to support communication with distant devices. The interconnection of peer LANs is discussed in greater detail in Chapter 4, which describes previous site interconnection research.

3.3.3 Recent Work

Recent local network research has been directed towards the development of faster LANs and the integration of PBX and LAN networks.

Improving LAN Capacity

Whereas the first generation of LANs support communication at up to 10 Mbps, more recent networks operate at much higher speeds. The Fibre Distributed Data Interface (FDDI)[1] is a 100 Mbps statistically multiplexed ring network originally intended for the interconnection of high speed devices such as mainframe computers, disks and tape drives. This network uses a *token* access protocol that permits a single device to seize the entire 100 Mbps ring bandwidth and transmit a lengthy burst of symbols, such as a disk block, to a peer device. The Cambridge Fast Ring (CFR), described in [Hopper 86], is a second generation slotted ring with an aggregate transmission bandwidth of up to 100 Mbps that is shared amongst active transmitters.

The aggregate capacity of shared medium LANS, such as the Cambridge Ring, has been improved by increasing the rate of transmission over the medium. An alternative approach is to replace the single shared medium with a mesh of switch nodes and interconnecting links so that concurrent associations can be spread across more than one physical medium. [Milway 86] reports on the development of a *binary routing network* that uses a space division switch fabric to support LAN services.

LAN/PBX Integration

Some recent proposals address the integration of LAN and PBX services by using a single physical medium to convey both styles of service. These hybrid networks impose a synchronous TDM frame structure on the transmission medium, with each frame divided into two types of timeslots. The *isochronous* timeslots are

---

[1]FDDI and FDDI-II are proposed standards being developed through the American National Standards Institute (ANSI).

dedicated to providing STM services and recurrent timeslots are allocated to specific occasions of communication.[1] The remaining *asynchronous* timeslots are concatenated together to support a PSTM service.

The Integrated Voice Data (IVD) proposal currently under development within the IEEE P802 committee uses this hybrid approach to distribute the functionality of a digital PBX. IVD provides a fixed link between a desktop device and a wiring concentrator. At the concentrator the isochronous timeslots of a number of attached devices are routed to the PBX whilst packets arriving in the asynchronous timeslots are fed into a high speed LAN such as FDDI.

The FDDI-II proposal is an enhanced FDDI LAN that permits a portion of the ring bandwidth to be allocated to STM services. When IVD is used in conjunction with FDDI-II each IVD concentrator represents a single point of attachment to the hybrid LAN. Isochronous IVD timeslots are fed into corresponding FDDI-II timeslots and, in effect, the time division multiplexed medium provides STM switching between peer concentrators. A distinguished device attached to the LAN may be used to provide the remaining PBX-like functions: access to the common carrier IDN network; and the management of isochronous communication. The signalling function may be provided by operating the ISDN signalling protocols over the PSTM service.

Integrated Services Local Networks

Although hybrid LAN schemes permit the physical layer integration of asynchronous and isochronous traffic, they do not provide for the integrated operation of different services. The different encodings, access schemes, multiplexing, and management make it difficult for applications to support the concurrent and integrated use of both the STM and PSTM services.

An alternative approach is to provide a single lower layer service that supports both styles of traffic. In the ISLAND project [Calnan 87], a range of voice messaging and PBX services are supported using an ATM-like service. Since the various styles of ISLAND communication share a common packet format, applications such as voice messaging can integrate traditional PBX services with LAN-based services such as editing and bulk storage.[2] Experimental work, described in [Want 88], has confirmed that voice quality can be maintained over an

---

[1]The frame frequency is usually fixed at 8 Khz so that single octet isochronous timeslots can support PCM-encoded telephone conversations.

[2][Calnan 88] will describe an integrated voice messaging and editing service.

ATM service, and reliable call control functions can be supported using LAN-based distributed computing techniques.

3.3.4 Discussion

Although the monolithic local network has been the holy grail of many research efforts, standards bodies such as the IEEE have adopted a different approach. In an effort to reduce the number of incompatible implementations, their activities are directed towards achieving consensus on a relatively small collection of standardized services and architectures. Individual sites can be expected to adopt elements of this collection to suit the requirements of their user communities. An advantage of this approach is that, as new styles of service evolve, individual sites can support these services by installing new networks that operate in conjunction with their existing facilities.

3.4 Summary

The site interconnection layer bridges the gulf between the services provided in the local and common carrier environments. The transmission requirements of the SI-layer must be compatible with a variety of common carrier transmission services and management strategies. Similarly, the SI-service must be compatible with a number of different local network architectures supporting a mixture of STM, ATM, and PSTM services. Each type of local network will have its own management and network interconnection strategies that must be accommodated within the SI-layer.

The following chapter presents a review of previous work on site interconnection issues. In Chapter 5 the network attributes identified in this chapter will be used, in conjunction with the service characteristics of chapter two, to describe an ATM-based site interconnection service.

Chapter 4
Previous Work

This chapter reviews three networking strategies that support communication between peer devices located at different physical sites. For the most part these strategies have evolved within the computer communication domain and some aspects of their architectures are not appropriate for use in multi-service and universal access environments. The reader familiar with internet techniques, the Universe architecture, and/or metropolitan area networks need only peruse the *discussion* subsections and proceed to the final section of the chapter which reviews recent work in two areas related to site interconnection: LAN interconnection; and ATM-based carrier networks.

4.1 Connectionless Internet Services

A *connectionless* network service supports the exchange of variable length service data units (SDUs) that are often referred to as datagrams. This network layer is a distributed object composed of subsystems located at different computer systems. In an *internet*, the subsystems that make up the layer cooperate to support the exchange of SDUs over a set of interconnected, heterogeneous subnetworks.

Internet protocols facilitate this style of operation by specifying a single protocol data unit (PDU) format that is used over all lower layer subnetworks. All communication between subsystems is based on this PDU format and the PDUs arising from an SDU can be passed through an arbitrary number of relay subsystems.[1] In the terminology of [ISO 8648] an internet protocol (IP) is a subnetwork independent convergence protocol. The PUP internet described in [Boggs 80] was one of the earlier schemes to make use of an IP and the approach

[1]From an architectural perspective there is no distinction between the terminal subsystems, to which users are attached, and the *gateway* nodes that perform the relaying function. However, some internet designs distinguish between these subsystems for management and implementation purposes.

has since been refined and standardized, initially within the DARPA community using the IP described in [Postel 81], and more recently on an international basis in accordance with [ISO 8473].[1]

Internet subsystems perform two basic functions, segmentation and forwarding, in order to support a store and forward PSTM service. The segmentation function allows a single PDU format to be used over a variety of lower layer networks that impose different constraints on packet lengths. Segmentation can be applied at any node along the path between correspondents, and so a source subsystem does not require information concerning the length constraints that may be encountered as PDUs are forwarded between source and destination. The forwarding function operates on a PDU-specific basis and supports the transfer of PDUs. An individual subsystem may be attached to a number of lower layer subnetworks and can forward packets across the subnetwork boundaries.

Shared access to common carrier services can be achieved by arranging for a subnetwork to have a *gateway* system that is attached to the common carrier. All of the inter-site traffic can be forwarded through this system, and its construction may be optimized to support this function. If the gateway is not directly attached to all of the site's local networks some of the traffic may traverse intermediate local networks and subsystems as it is forwarded to the gateway.

Internet addressing is based on the use of global network service access point (NSAP) addresses. Passive directory functions are used to resolve service titles into system titles, which are in turn resolved into NSAP addresses. Each subsystem must map the destination NSAP address within each PDU into the NSAP address of the next subsystem to which the PDU will be forwarded. The subsystem then resolves the next hop address into a title or address suitable for use over the lower layer network linking the peer subsystems. These routing functions are performed on a hop by hop basis but they may be affected by *source routing information* placed within the PDU by the source subsystem.

Discussion

The distinguishing characteristic of the IP approach is that physical devices, or systems, are the principal communicating objects. All of the lower layer channels that interconnect peer systems operate at the same level of abstraction, and there

---

[1]The IP description and terminology used in this section is based on the ISO standard. However, from an architectural perspective the ARPA and ISO protocols are closely aligned and most of this text is equally applicable to both standards.

is no architectural distinction between local and common carrier networks. The IP architecture takes no account of physical site boundaries. This approach leads to a very flexible topology of loosely coupled co-operating systems. However, it is not clear how independent management services can exercise control over network components in order to perform functions such as bandwidth allocation.

The internet approach leads to reasonable throughput if relatively long PDUs[1] are used to support the PSTM service. However, long packets induce significant delay and jitter elements that are not acceptable in multi-service communication environments. IP jitter performance is impaired by store and forward delays at the intermediate subsystems and by the requirement for in-band segmentation, reassembly, and routing processing.

4.2 The Universe Network Architecture

The principle aim of the Universe architecture [Leslie 84] was to extend the use of LAN communication techniques into the wide area. Universe implements *lightweight* virtual circuits[2] that provide for the transparent exchange of SDUs between peer upper layer services operating at different local networks. The architecture extends the geographic range of the Cambridge Ring basic block layer to support communication between peer entities operating at different LANs.[3] Universe virtual circuits preserve the sequence of blocks as they traverse the network, but end-to-end flow control and error recovery are not provided. When required, these functions can be provided by appropriate upper layer protocols. Since Universe is based on the pervasive use of basic blocks at all of the inter-connected LANs, there is no provision for the blocking or segmentation of PDUs as they are passed between adjacent networks.

Communication across LAN boundaries is supported by distinguished LAN nodes known as *bridges*. Local bridges support the interconnection of LANs within a

---

[1]Each PDU segment must carry at least two global addresses and, in the case of ISO IP, PDU headers can extend to up to 255 octets. In order to absorb this overhead each PDU will have to carry a fairly lengthy payload of upper layer symbols.

[2]In addition to virtual circuits, Universe supports a connectionless datagram service and a single shot transaction service.

[3]Although the work described in [Leslie 83] was based on the interconnection of Cambridge Rings, the architecture can be extended to support other LAN technologies.

physical site. Each local bridge is attached to two physically adjacent LANs and supports the flow of basic blocks across the LAN boundary. Off-site bridges are located at the points of attachment between LANs and wide area subnetworks. Peer bridges use the facilities provided by the wide area subnetworks to support the flow of basic blocks across site boundaries. Since bridges operate within the basic block layer, and do not themselves support upper layer services, their designs can be optimized to streamline their operation.[1]

In a single Cambridge Ring the basic block layer supports the exchange of SDUs between peer upper layer entities. The block layer implements the exchange of PDUs between peer ring stations[2] and supports the multiplexed use of the stations by concurrently active upper layer entities. Each SDU is encoded into a single PDU and the protocol described in Appendix A is used to effect the exchange of blocks between peer systems. Each PDU carries a *port* field that identifies the intended SDU recipient within the destination system.

The creation of a virtual circuit, supporting an association between peer upper layer services, is a two stage process involving an initial out-of-band transaction with a local nameserver. Each local network, or subnet, is equipped with a nameserver that resides at a well known service address. The initiating service uses the nameserver to resolve the title of a peer service into a service address that consists of a local station address and a *public* port value. The universe of discourse of a service address is limited to a single subnet: the port value is only valid within the context of a given station, and the station address is only valid within the context of a single LAN.

In the second stage of circuit establishment, the initiator sends a distinguished *open* block to the service address. In this open block the initiator quotes a *private* port value to support the reverse flow of blocks over the circuit. The responding service replies with a distinguished *openack* block containing an association-specific private port value that will support the forward flow of blocks.[3] Once the association has been established the peer services use the basic block layer to support the exchange of SDUs between their private ports. When the two services are located on the same subnet, the nameserver passively resolves the service title

---

[1]Within Universe four different bridge implementations were developed to suit different interconnection requirements.

[2]The points of attachment to a LAN are often referred to as stations.

[3]The open and openack blocks may contain additional information that is used to negotiate other association-specific parameters.

into the published station address and public port of the service. The exchange of open and openack blocks proceeds in a straightforward manner and the associated services communicate without the assistance of bridges.

When the recipient service is supported at a remote subnet the local nameserver actively resolves the service title into a bridge address and *public* bridge port through which the open block can be sent. To accomplish this task, the nameserver first resolves the service title into a static global address that contains site, subnet, station and port components. From this address, and some knowledge of its own site's bridges, the nameserver can identify the bridge that represents the first hop along a path between the initiator and the recipient. The nameserver then completes an out-of-band transaction[1] with a management component that resides at a well-known port of this bridge. The nameserver supplies the bridge manager with the global address of the recipient service, and the bridge manager in turn allocates the public port through which open blocks can be sent. This port value and the bridge station address are eventually returned to the initiator as the service address.

When a subsequent open block arrives at the bridge, its port value is used to identify the global address of the destination service. The bridge allocates a reverse port to the association and uses the site and local network fields of the global address to determine the next hop along the route. The global address and the contents of the open block are both forwarded to the next bridge where the forwarding process is repeated. When the block reaches a bridge attached to the destination LAN, the station address and port fields within the global address are used to forward the block to the destination service. At each hop along the route, the bridges allocate forward and reverse ports to support the virtual circuit that is being created. When the open block arrives at its destination, the private port value it carries identifies the reverse port allocated by the last bridge traversed. Using the bridge address and port value, the respondent returns the openack message along the reverse chain without reference to the global address of the initiator. When the openack arrives at the initiator it contains the forward port value allocated to the circuit by the initial bridge.

The forward and reverse port chains define a virtual circuit between the peer services, and bridges are designed to efficiently forward basic blocks along such chains. At each bridge the incoming port field within a block identifies the outgoing station address and port fields of the next hop. Each bridge port

---

[1]This transaction is nested within the initiator-nameserver transaction.

represents a dynamically allocated window from the service address space of one subnet to the address space of the adjacent subnet. The bridges support the extension of the basic blocks service by mapping the addresses of basic blocks as they cross subnet boundaries.

Discussion

In the Universe architecture, bridges and local networks support the exchange of messages between peer entities. There is a clear distinction between the client systems that support upper layer services and the bridge systems that support the relaying of messages. Furthermore, the architecture explicitly acknowledges the role of physical sites and provides for the shared use of carrier facilities. An interesting feature of the name resolution scheme is the deferred binding of service titles to addresses. While resolving a title associated with a remote service, the local nameserver can interact with the remote nameserver in order to acquire the global address of the service. This interaction provides for the late binding of titles to global addresses. Since individual sites do not have to publish the addresses of their services they retain greater independence with respect to the physical location of services within their site.

Universe virtual circuits have been used in a range of experimental applications including distributed computing [Richardson 83], voice [Adams 85] and images [Griffiths 84]. Some of these applications were found to be sensitive to the store and forward delays induced by bridges. Furthermore, the lack of in-block multiplexing within the basic block protocol can induce a substantial jitter component: a single station does not multiplex the exchange of basic blocks, and so a slow station can fully monopolize its peer for an extended period of time. If the peer node is a Universe bridge, the traffic on one virtual circuit will induce considerable jitter into the other circuits that traverse the bridge.

4.3 Metropolitan Area Networks

Some authors have proposed the development of Metropolitan Area Networks (MANs) that use shared media to interconnect local networks that are dispersed over metropolitan areas. The IEEE P802.6 committee, responsible for the development of a MAN standard, has considered various MAN implementations.[1]

[1][Mollenauer 88] reviews the history of P802.6 in a special issue of IEEE Communications magazine devoted to MANs.

The Multiplexed Slotted Token (MST) proposal [Sze 85] was based on a slotted ring that is similar to the CFR. This proposal lacked industrial backing and the committee now favours the Queued Packet and Synchronous Exchange (QPSX), proposed in [Budrikis 86]. QPSX relies on two unidirectional synchronous buses arranged in a loop topology. Both schemes are based on a hybrid service consisting of an ATM-like packet service and an IDN-compatible STM service.[1] In the current proposal, the ATM service provided by the MAN only supports the transfer of data link layer frames arising from traditional computer communication networks. In effect, only PSTM and STM services are provided to MAN users. There is no provision for the support of other styles of communication through direct access to the lower layer ATM service.

Current MAN work at the Computer Laboratory is based on a third generation slotted ring, the Cambridge Backbone Network (CBN), described in [Greaves 88]. This MAN can extend over a 50 km diameter and will operate at speeds of up to 1000 Mbps. Although the CBN packet format is identical to that of the CFR, the frame structure and slot protocol have been modified to relax the present limits on single node throughput. One application of the CBN is the interconnection of CFRs within a metropolitan area.

4.4 Recent Work

LAN Interconnection

A significant amount of recent work has concentrated on the implementation of LAN interconnection. Some of the work described in [Bux 87][2] has extended the internet approach whilst other work has been based on the direct bridging of local networks within the data link layer.[3] A common area of interest is the issue of routing PDUs through a concatenated chain of LANs, and schemes using flooding, source routing and hypercubes have been suggested.

One management proposal of interest [Estrin 87] uses *visas* to provide access controls within *interorganization* networks. Although this dissertation does not

---

[1]The synchronous frame structure imposed on the shared media is repeated every 125 micro-seconds. As with FDDI-II, octet timeslots within the frame structure are assigned to 64 Kbps STM connections.

[2][Bux 87] is a reference to the entire issue of IEEE JSAC, edited by Bux et al, devoted to LAN interconnection.

[3]This approach is often referred to as MAC layer bridging since it operates at the MAC sublayer described in Appendix F.

develop the design of SI access control and accounting functions, the visa approach could be adapted to provide these services. In this application, the scheme described by Estrin would be modified to use out-of-band transactions to issue the temporary visas honoured by the in-band components.

Other work of interest involves the specification of carrier-based LAN interconnection services. The Switched Multi-megabit Data Service (SMDS) described in [Hemrick 88] is a proposed carrier offering that relies on switching facilities within the carrier network to support communication between disjoint LANs. SMDS is tailored to the support of traditional LAN applications and makes little or no provision for the integration of other services such as telephony and video.

ATM Networks

In the past few years there has been considerable interest in the development of a broadband carrier network supporting subscriber transmission rates of about 150-500 Mbps. Although this network could provide an STM service, through the direct extension of the ISDN design, a number of investigators have proposed that an ATM service be provided. Consequently, a great deal of carrier domain research has proceeded in parallel with the site interconnection work described in this dissertation.

One of the earlier proposals for an alternative broadband architecture is the Integrated Services Packet Network (ISPN) described in [Turner 86]. This proposal extends the benefits of asynchronous bandwidth allocation and service integration into the carrier environment. ISPN is based on two layers of carrier encoding: the lower layer supports variable length packets, or frames; and the upper layer provides in-band support for both datagram and connection-oriented services. Although Turner's ISPN is concerned with voice and data transmission within the carrier environment, it addresses many of the architectural issues identified in this research. The principal differences in approach are that the SI-service described in this dissertation is oriented towards a wider multi-service environment. The SI-layer provides a uniform client service based on a common encoding and the use of out-of-band techniques.

A great deal of work has concentrated on the development of switch technologies to support an ATM-based broadband service. A number of different switch designs

are described in [White 87].[1] In particular, [Hui 87][2] describes a non-blocking space division switch based on a batcher-banyan switch fabric. [Newman 88b][3] describes an alternative design based on non-buffered binary routing networks. TDM switches have also been considered and [Coudreuse 87] describes an ATD design that *diagonalizes* packets so that the octets of different packets can be interleaved as they pass through the switch.

Although switch development has proceeded at a frantic pace, there has been little work on the overall design of ATM carrier networks and their services. Much of the published work has been undertaken within the context of a single switch technology. For example, [Wu 86][4] and [Littlewood 87][5] have described various approaches to the deployment of single technology networks using space division switches and slotted rings, respectively. Of greater long term significance is the work under way within the CCITT to develop a preliminary ATM service specification. [Minzer 87] reports on some of the deliberations and suggests that the CCITT will settle on a hybrid service in which a synchronous frame structure supports both STM timeslots and ATM packets. There appears to be little consensus on the structure of the ATM encoding or the delay and jitter characteristics of the service.

4.5 Summary

It is instructive to contrast the architectural structure of previous interconnection strategies with the site interconnection model proposed in this dissertation. The Internet approach concentrates on communication between peer systems. These systems operate at a common layer and exchange PDUs over a heterogeneous lower layer made up of local and common carrier networks. In contrast, the site interconnection approach, which has evolved from the Universe work, concentrates on communication between peer local networks located at distinct physical sites. This results in a structured hierarchy that distinguishes between: the interconnection of systems within the local network layer; the interconnection of

---

[1][White 87] is a reference to the entire issue of IEEE JSAC, edited by White et al, devoted to switching systems for broadband networks.

[2]Also, [Day 87].

[3]Also, [Newman 88a].

[4]Also, [Wu 87].

[5]Also, [Gallagher 86] and [Key 87].

local networks within the SI-layer; and the interconnection of sites within the common carrier layer.

A principal distinction between this research and other recent work is that, for the most part, the other research has focused on the use of a particular technology to achieve site interconnection. The architecture described in this dissertation is intended to operate in conjunction with a number of technologies and to survive the evolution of the transmission substrate. The design of the architecture has concentrated on the characteristics of the service offered rather than on the means of its implementation.

MAN research has been driven by the availability of specific technologies, and little effort has been made to locate MANs within the overall telecommunications environment. From a site interconnection perspective, MANs represent one of a number of technologies that can be used to implement the subsystems of the SI-layer. MANs can be used to extend the benefits of the SI-layer to distributed sites, such as residential housing estates, where the subsystem implementation techniques described in this dissertation may not be appropriate. Similarly, recent work on ATM-based carrier networks has been driven by the development of high capacity switches and fibre-based transmission facilities. Although an ATM-based carrier network would provide an appropriate transmission service for use within the SI-layer, it is not an essential component of the site interconnection architecture.

Chapter 5
The Site Interconnection Layer

The site interconnection layer supports communication between peer local networks. The layer is a distributed object composed of independent SI-subsystems located at participating sites. Each subsystem is positioned between the common carrier and local networks of its site. The local networks represent the upper layer users of the SI-subsystem and their points of attachment are referred to as site interconnection service access points (SI-SAPs). The carrier networks are the lower layer entities that support communication between peer subsystems.

Peer subsystems co-operate in order to provide a uniform service between their SAPs. This service can be modelled in terms of the transmission of encoded messages between peer SAPs. The attributes of the SI-service are determined by the nature of the common encoding, applied at the SI-SAPs, and the properties of the transmission channels linking peer SI-subsystems.

This chapter identifies the fundamental site interconnection issues and describes a suitable SI-service based on ATM encoding and transmission techniques.

5.1 Issues in Site Interconnection

5.1.1 Multi-Service Communication

The local networks attached to an SI-subsystem may support different styles of communication, and so the SI-service must cope with traffic arising from a variety of end user services. In particular, the SI-service must be suitable for the transport of both fixed and variable rate encodings.

The resultant combination of asynchronous and isochronous traffic places conflicting demands on the service. It must provide for the efficient transport of asynchronous communication whilst maintaining the integrity and temporal coherence of

isochronous traffic. The bursty nature of asynchronous traffic leads to a high peak-to-average bandwidth ratio which implies that the lower layer carrier facilities should be statistically shared amongst concurrent users. The difficulty lies in ensuring that this sharing takes place without inducing an unacceptable degree of jitter or delay into the fixed bandwidth isochronous traffic.

5.1.2 Shared Access to Common Carriers

A principal motivation for the imposition of a site interconnection layer is the economies that can be realized when the local networks of a site share the services available at a small number of common carrier NTEs.[1] In order to apportion access to these collective services the SI-layer must incorporate multiplexing, switching, and rate adaption functions within the independent subsystems. This degree of functionality has significant implications for the specification of the SI-service and its encoding. In particular, the implementation of SI-subsystems can be streamlined by choosing an encoding that is compatible with the integration of all of the multiplexing and switching functions located within the SI-layer.

Multiplexing

The subsystems that make up the SI-layer incorporate multiplexing functions that support concurrent occasions of communication, or associations, between layer entities. Three distinct levels of multiplexing can be identified within each SI-subsystem:

- At the upper layer, the SI-service supports communication between peer local networks. In general, each local network may exchange symbols with any number of peer networks located at the same or different sites. Therefore each subsystem must support the multiplexing of concurrent associations at its individual SI-SAPs;
- Within the SI-layer, carrier services are used to construct channels between peer sites. Communication between peer SI-subsystems should be multiplexed to ensure that these channels support concurrent associations between the local networks of the peer sites; and
- Below the SI-layer, some common carrier NTEs, such as those attached to ISDN networks, provide support for concurrent channels involving different peer sites. Although this level of multiplexing is not implemented within the SI-layer itself, its presence within the lower

---

[1]Common carrier points of attachment are usually referred to as Network Termination points (NTEs) rather than SAPs.

layer has implications for the design of individual SI-subsystems.

Switching

The SI-subsystems also incorporate switching functions that provide transmission paths between peer local networks. Three applications of switching can be distinguished:

- Switching between local SI-SAPs. In the case of intra-site communication, this switching function supports the exchange of symbols between local SI-SAPs;
- Switching as symbols are passed down from the upper layer SAPs to the NTEs. In the case of inter-site communication, this switching function provides a path, which passes through the subsystem itself, between the appropriate upper layer SI-SAP and the corresponding lower layer NTE; and
- Switching between NTEs attached to a common SI-subsystem. When the available common carrier networks cannot support a direct channel between peer sites, one or more relay sites may be used to concatenate a number of inter-site channels to provide a path between the peer sites. To support the relay function an SI-subsystem must be capable of switching symbols between its lower layer NTEs.

Rate Adaption

The multiplexing and switching components of the SI-layer must support communication between peer SI-SAPs that operate at different transmission rates. Individual SI-subsystems must support rate adaption between local SI-SAPs operating at speeds dictated by the local networks, and NTEs operating at speeds determined by the common carriers. Similarly, the SI-layer as a whole should facilitate communication between peer local networks that operate at different speeds.

5.1.3 Universal Access and Site Independence

The management of the SI-layer must support universal access without unduly constraining the independent administration of individual sites. An important implication of universal access is that the number of *potential* correspondents that can be accessed by a site vastly exceeds the number of *dynamic* correspondents with which the site concurrently exchanges symbols.

The SI-layer could be organized as a closely coupled mesh with peer SI-subsystems co-operating on an ongoing basis through the regular exchange of management

information. The difficulty with this approach lies in the scale of management information exchanged when the layer extends to include hundreds of thousands of sites. A further problem is that the close coupling limits the independence of individual SI-subsystems which may be required to provide detailed information on their configuration and availability to all of their potential peers.

An alternative approach, favoured in this work, is to operate a loosely coupled SI-layer that emphasizes the independent operation of individual sites. Peer SI-subsystems utilize common carrier services to dynamically construct inter-site channels between communicating sites. Management information is exchanged on a peer basis, as required, without the involvement of SI-subsystems located at other sites. This approach replaces the universal mesh of the closely coupled scheme with a number of small, dynamically created, meshes of communicating sites. Subsystems only retain information concerning their dynamic correspondents and delegate responsibility for universal access to the common carrier networks which have been expressly designed for this purpose.[1] The proposed organization allows individual sites to enforce their own security and accounting procedures and to retain control over the resources available at their NTEs and SI-SAPs. Communicating sites negotiate the allocation of resources, and security measures such as authentication, on a peer basis in accordance with their own requirements.

Naming and addressing are areas in which support of universal access complicates the operation of a loosely coupled SI-layer. One approach, using early binding, is to allocate a global SI-address to every local network service accessible at a site. Service titles and their bindings to global addresses are published in directories which can be accessed by SI-subsystems. The addresses can be hierarchically structured into site, SI-SAP, and local service components. The difficulty with this approach is that the global addresses will be quite long, and will in part duplicate the function of the NTE addresses supported by the common carrier. Furthermore, each address is likely to be related to the physical configuration of the site, and so once an address is published it becomes difficult to revoke or otherwise modify. In effect, early binding limits a site's control over its internal configuration.

The alternative, and favoured, approach involves the late binding of service titles to addresses. In this case a site only exports information identifying the local network services it supports. The information recorded in directories can be

---

[1]In the general case, relaying by third party sites can be supported through the concatenation of inter-site channels by switching functions within intermediate SI-subsystems. The decision to support relaying involves consultation with the intermediate sites which retains control over their own resources.

restricted to the binding of service titles to site titles.[1] The directory information is used to establish communication between peer sites and the actual binding of service titles to SI-addresses is negotiated by peer SI-subsystems on an *as required* basis. The addresses used can be statically assigned from a global address space whose universe of discourse encompasses all potential correspondents, or an active name resolution scheme can be used to dynamically assign addresses from a smaller name space whose universe of discourse is limited by the number of dynamic correspondents.

5.1.4 Avoiding Obsolescence

The site interconnection architecture should speed the evolution of the telecommunications environment by facilitating the introduction of new services and technology. A major force driving this evolution will be technical advances leading to higher speed digital transmission, multiplexing and switching. Furthermore, since different sites will have different bandwidth requirements, it is likely that a variety of technologies will be used to implement SI-subsystems. Communication between SI-subsystems must be structured so as to ensure the inter-operability of different implementations:

- The SI-encoding must be scalable to support communication over a wide range of transmission rates, and should not be tied to a particular bandwidth or sample frequency; and
- The rate adaption function must support communication between different subsystem implementations.

5.2 The Site Interconnection Service

The service provided by the SI-layer can be described in terms of the encoding applied at the SI-SAPs and the properties of the transmission channels constructed by peer SI-subsystems. The selected encoding should be compatible with the integration of all of the multiplexing and switching functions located within the layer. Time division techniques can be used to achieve this level of integration provided the SI-encoding is based on the blocking of symbols into packets. The length, format, and frequency of packets are functions of the transport mode used within the SI-layer.

---

[1]The binding of site titles to NTE addresses may be recorded in directories operated by common carriers.

The Asynchronous Transfer Mode (ATM) has been chosen as the basis of the pilot SI-service described in this dissertation. The ATM approach is suited to the support of rate adaption and traffic arising from a mixture of continuous and variable rate upper layer encodings. The specification of an ATM encoding, which is divorced from a specific clock frequency, should permit inter-operation between different SI-subsystem implementations.

ATM represents a compromise between traditional packet switched transfer (PSTM) and synchronous transfer (STM). PSTM supports rate adaption but is not compatible with stringent delay and jitter performance requirements. In contrast, STM-based systems can be designed to meet strict performance targets but they do not provide rate adaption nor do they efficiently transport traffic arising from variable rate sources.

5.2.1 Encoding

ATM communication is based on the exchange of fixed length packets that can be inserted into the slots of time division frames. In general, the packet arrival process is asynchronous, and so the length of individual packets must be constrained in order to minimize the effects of packetization delay and the jitter arising from contention at multiplexing and switching points.

Each packet must contain sufficient addressing information to associate it with a given occasion of communication. The use of short packets implies that the in-band components of SI-subsystems must support high packet rates, and the packet format should be structured so as to reduce per-packet overheads. This requirement can be accommodated through the adoption of fixed length addresses and the division of packets into distinct address and symbol fields. The length of the address field should be chosen so that the overhead associated with the transmission of addressing information does not represent an undue proportion of the aggregate bandwidth required to support packet transmission.

5.2.2 Transmission

The SI-service supports the simple transfer of ATM packets between *associated* user entities attached to peer SI-SAPs. The service provided at the SI-SAPs is

structured so that service characteristics, other than performance, are independent of the underlying common carrier facilities.

The SI-service supports a wide range of upper layer services with varying packet transfer requirements. The approach adopted here is to specify a lowest common denominator transfer service whose basic properties can be enhanced, at the peer SI-SAPs, on an end-to-end basis. In particular, it is important to exclude non-pervasive transfer functions if their inclusion in the SI-service is likely to impair the performance experienced by all SI-users. In practice, this means the exclusion of functions that are likely to require in-band processing by the shared components of the SI-subsystems.

The following paragraphs describe the packet transfer characteristics of the SI-layer. The terminology and organization of this description are based on the *elements of layer operation* clause of the OSI Reference Model [ISO 7498-1].

Multiplexing

The SI-service provides for the multiplexing of concurrent associations at individual SI-SAPs. Clearly this level of multiplexing is directly accessible to SI-users. The additional aspects of multiplexing and switching, described earlier in this chapter, are embedded within the SI-subsystems and their operation should be transparent to users of the service.

Splitting

Splitting may be used to allow a number of common carrier channels to be aggregated together to form a single inter-site channel. This form of splitting is related to the encoding of SI-layer packets for transmission over lower layer channels. Different splitting schemes may be adopted for use over different types of common carrier networks. When it is implemented, the splitting function is transparent to SI-users.

Segmentation and Blocking

The ATM encoding used at the SI-SAPs should be compatible with the integration of all multiplexing and switching components, and so there is no requirement for segmentation or blocking within the SI-subsystems. As with splitting, segmentation and blocking may be associated with the encoding of SI-packets for transmission over common carrier facilities, and their effects should be transparent to SI-users.

Peer SI-users may implement both functions during the encoding of upper layer symbols into SI-packets. For example, segmentation can be used to encode relatively large upper layer messages such as Internet datagrams into trains of SI-packets. Similarly, blocking may be used to collect a sequence of PCM octets into a single SI-packet.

Error Detection

An error detection function must protect the address fields of packets as they are transferred between SI-SAPs. In order to avoid the incorrect delivery of packets, the SI-service must arrange for the deletion of all packets whose address fields are distorted in transit. Error detection can be performed on an end-to-end basis at the SI-SAPs, or on a hop-by-hop basis within the SI-subsystems.

The extension of error detection to the packet symbol fields is an open question. Once in-band detection is provided for the address fields it may be possible to extend protection to the symbol fields without incurring an additional performance penalty. The provision of this function within the SI-layer would reduce the overall overhead experienced by upper layer applications dependent on error-free symbol transmission. On the other hand some upper layer encodings, such as PCM voice, may be tolerant to errors in the symbol field and their performance might be impaired by the arbitrary deletion of packets containing slightly distorted symbol fields. In the absence of frequent symbol distortion, this latter consideration is of diminished importance.

Acknowledgement and Retransmission

The SI-service does not support the acknowledgement of packet delivery or the detection and retransmission of deleted packets. Many styles of communication do not require these functions, and, where necessary, they can be provided through the operation of appropriate upper layer protocols. Their inclusion within the service would necessitate additional in-band packet processing and the retention of a considerable amount of association-specific state information within subsystem components.

Sequenced Delivery

The sequence of SI-packets must be preserved during their transfer between peer SI-SAPs.[1] Out of sequence packet delivery would significantly impair the value of the service to temporally sensitive applications such as telephony. In the absence

---

[1]When the SI-SAPs are multiplexed, packet sequences need only be preserved within the context of individual associations.

of sequence permutation by common carrier channels, sequenced delivery can be achieved, without incremental in-band processing, by maintaining packet sequences on a hop-by-hop basis.

Flow Control

Flow control functions add to the demands placed on the in-band operation of the SI-subsystems and introduce contention effects within the switching and multiplexing components. From a performance perspective, flow control impairs the delay and jitter performance experienced by all SI-users and should be completely excluded from the SI-service. This position is not compatible with the requirement for rate adaption, and so a secondary objective is to structure flow control functionality so as to localize its effects. In particular, the support of in-band end-to-end flow control should be avoided.

The SI-layer supports rate adaption between SI-SAPs and carrier NTEs operating at different transmission rates. The flow control functionality necessary to this form of rate adaption can be achieved through the application of localized *back pressure* at the subsystem interfaces. In an ATM environment the implementation of back pressure will be dependent on the contention resolution mechanisms adopted by the switching and multiplexing components. In choosing an ATM implementation it is important to assess the implications of back pressure on the performance and fairness of contention resolution.

Although localized back pressure assists the rate adaption process, it does not prevent SI-users from flooding the service at rates that exceeds the capacity of the SI-subsystems, the inter-site channels, or the peer SI-users. On a transient basis, the subsystems can cope with flooding through the elimination of excess packets. However, the random deletion of packets may lead to end user retransmissions, which in turn increase the load on the subsystems. Packet deletion should be reserved for statistically infrequent events and the SI-service should include some mechanism for limiting the rate at which individual SI-users generate packets. Since in-band flow control is undesirable, it is proposed that out-of-band techniques be used to prevent flooding through the assignment of maximum transmission rates to individual associations.

Expedited Transfer

The overall utility of the service can be improved by providing SI-users with a mechanism for distinguishing packets that are temporally sensitive, i.e., those packets arising from applications that cannot tolerate significant amounts of delay and jitter. The SI-service should support the expedited, or priority, transfer of

these distinguished packets. This function can be supported on a packet-specific or an association-specific basis. The latter approach is preferred as it permits out-of-band techniques to be used to configure subsystem components when an *expedited* association is initiated. The assignment of a common priority level to all of the packets arising from a single association avoids undue conflict between the sequenced delivery and expedited transfer functions.

Layer Management
Independent management entities within each SI-subsystem must interact with local network, common carrier, and peer subsystem management components. These entities must support a variety of layer management functions including: name resolution and association establishment; resource allocation; routing; monitoring; and error control. The support of management functions should be structured so as to streamline the in-band operation of subsystem components that directly support the packet transfer service.

5.3 Relationship to OSI

In the OSI architecture, the Network service supports communication between Network layer SAPs located within peer Open Systems. The internal organization of the OSI Network layer [ISO 8648] is defined in terms of network layer entities that communicate with each other over services provided by a collection of loosely-coupled subnetworks. Local and common carrier networks are both modelled as subnetworks whose entities perform relaying functions on behalf of the client entities within the peer end systems.

The site interconnection architecture imposes a somewhat more structured organization on the Network layer by dividing it into distinct local network, site interconnection, and carrier network sublayers. The service provided by the uppermost sublayer is roughly equivalent to that provided by the OSI Network service [ISO 8348]. Entities within this sublayer support communication between peer systems either directly within their local subnetwork or indirectly through services provided by their SI-subsystem. Within the intermediate SI-sublayer, peer subsystems support communication between local networks and exercise the carrier services provided by the bottom sub-layer. In the terminology of [ISO 8648], the SI-sublayer supports a uniform Subnetwork Independent Convergence function that is applicable to all of the subnetworks within the layer.

5.4 Summary

This chapter has identified the issues that must be addressed within the site interconnection layer. The analysis of these issues has led to the selection of ATM as the basis of a site interconnection service and a description of the corresponding encoding and transmission characteristics. The remaining chapters of this dissertation describe the design, implementation, and performance of the *exchange* architecture which is the basis of the experimental work associated with this thesis. The exchange supports a pilot SI-service that has been used in the experimental investigation of many aspects of site interconnection. In many respects the issues and service characteristics that have been described in this chapter represent the *results* of this experimental work.

Chapter 6
The Exchange Architecture

The *exchange* architecture has been developed as a concrete example of the organization and operation of the site interconnection layer. This chapter provides an overview of the principal exchange components and describes the pilot implementation which has been used within the experimental programme.

The distinguishing characteristic of the exchange approach is that each SI-subsystem is built around a high speed switch fabric that provides the physical link between a site's local networks and its common carrier NTEs. The switch fabric is referred to as the *local exchange* and its points of attachment to the site's local networks and common carrier facilities are referred to as *portals* and *ramps*, respectively. In addition, each site has its own complement of management entities that control the site's resources and communicate with peer management entities at remote sites. A typical SI-subsystem, illustrated in Figure 6.1, consists of a local exchange and the collection of ramps, portals, and management systems that are attached to it. The site interconnection service is offered at SI-SAPs which are logically embedded within the individual portals.

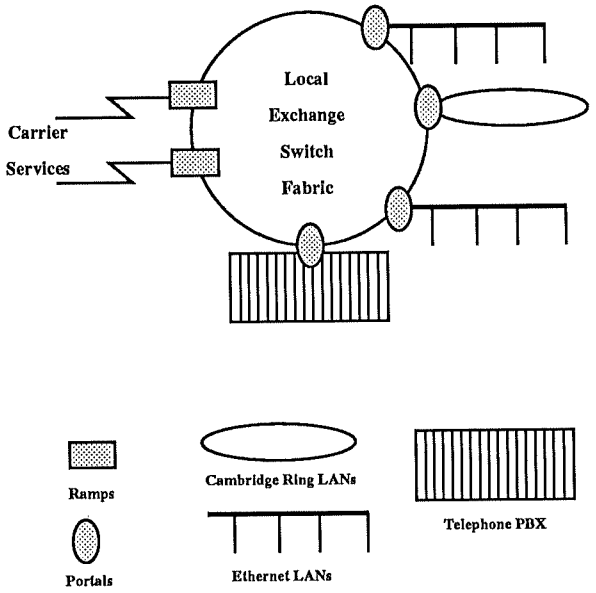

Figure 6.1: Local SI-subsystem

An important aspect of the exchange approach to site interconnection is the emphasis on site independence. Each exchange-equipped site represents an autonomous community of devices that are attached to local networks and share

access to common carrier services. The exchange architecture facilitates site interconnection through the specification of conventions that allow peer sites to dynamically knit their SI-subsystems together. Specific occasions of site interconnection are directly negotiated by the independent management entities of the communicating sites.

6.1 The Local Exchange

The decision to build the individual SI-subsystems around high speed switch fabrics follows directly from the requirement for extensive switching, multiplexing, and rate adaption functionality within the SI-layer. Furthermore, if the SI-service is to be ATM-based, it is reasonable to restrict the choice of available switch technologies to those that are compatible with packet oriented ATM encodings: asynchronous time division (ATD) switches; and self-routing space division switches.

The Cambridge Fast Ring (CFR) has been selected as the switch technology used within the exchange architecture. The CFR is a slotted ring whose design [Temple 84] and implementation [Hopper 86] are based on experience with the Cambridge Ring and a number of multi-service network applications.[1] Although the CFR is intended for use as a high speed local area network it satisfies most of the requirements for a site interconnection switch fabric. The slotted ring access protocol, small fixed packet length, and distributed implementation make the CFR a particularly convenient form of ATD switch.

6.2 Exchange Interconnection

Although the implementation of an ATM-based carrier network may be desirable, such a network is not indispensable to the realization of an SI-service. The exchange architecture operates an ATM overlay above available transmission services without compromising the service requirements developed in Chapter 5.[2] The pilot exchange implementation described in this dissertation is based on

---

[1]For example, [Leslie 84] and [Calnan 87].

[2]Clearly the quality of the ATM service will be dependent on the performance attributes of the carrier networks. For example, if an IP network is used the peer ramps will embed short ATM style packets within much longer IP datagrams. The per packet overheads incurred within the IP switches will induce jitter and delays that mitigate the performance advantages of the SI-service.

primary rate ISDN facilities. Exchange ramps superimpose the CFR packet structure on the synchronous switched circuits supported by the ISDN carrier. This approach permits an SI-service implementation that benefits from the present availability of large scale circuit switches and high bandwidth transmission facilities.

The exchange architecture relies on public common carrier facilities to support universal access between peer sites. However, the architecture can also be used to interconnect a closed group of sites into a private network similar to Universe. The sites may be connected in a fairly arbitrary mesh with some of the local exchanges acting as intermediate relay points for traffic between non-adjacent sites. Interconnection is achieved using private transmission facilities, such as fixed ground links or satellite channels, or through the operation of a *closed user group* facility provided by a common carrier. A site's exchange may concurrently support operations within both public and private domains, using public carrier facilities to access peer sites outside its closed groups, and private facilities to interact with its closer associates.

6.3 Ramps

Exchange ramps are attached to the termination points (NTEs) of each carrier network. The physical layer of each ramp consists of a carrier-dependent interface to the NTE and a CFR interface to the local exchange. Above the physical layer, the carrier transmission services are formatted into *channels* that support CFR packet transmission. These channels are used to construct *bridges* that support the transparent flow of packets between peer exchanges.

A ramp attached to a fixed transmission facility cooperates with its peer to support a single bridge, whilst a ramp attached to a switched carrier network can support an arbitrary number of bridges. The number of *potential* bridges is a function of the number of exchange sites serviced by the carrier network. Typically the design of the ramp and the capacity of the carrier facility limit the number of *dynamic* bridges supported at a given point in time. Bridges to peer sites are dynamically constructed on demand and are demolished when they are no longer required.

Given the short CFR packet length, ramp designs must minimize the in-band processing associated with the packet transfer function. In the case of fixed transmission facilities, the channel formatting function is fairly straightforward and

per packet overheads can be kept to a minimum. The service provider specifies the physical interface presented at the NTE, and, provided the ramps conform to this specification, there are few restrictions on how the service is used.

For NTEs attached to switched networks, the type of carrier service provided has a significant impact on ramp design and complexity. The service provider specifies the signalling conventions that are used to communicate with the management of the carrier network, and exchange ramps must conform to these specifications in order to route channel data to peer ramps. When a ramp is attached to a connection-oriented network a great deal of the signalling functionality can be physically removed from the ramp and placed within management entities attached to the local exchange.[1] In contrast, ramps attached to IP-like connectionless networks must embed addressing information within every data unit that is presented to the carrier. Although these ramps may support dynamic access to a very large number of peer sites, it is difficult to streamline the design of their in-band signalling functions.

6.4 Portals

Exchange portals support communication between devices attached to their own local networks and remote devices attached to peer networks. These portals can be viewed at two distinct levels of abstraction. At the physical level a portal provides a path between its local network and the exchange switch fabric. At the peer level, portals support the transparent flow of digital symbols between peer local networks *of the same generic type*.[2]

Portals encode the symbols presented by their local networks into the CFR packet format which is the common SI-encoding supported by the exchange. In the case of traditional local data networks, the client symbols are already packetized and the

---

[1]This is particularly true for ISDN networks where signalling functions are performed out-of-band with respect to the flow of user data. Even when in-band signalling is required, as is the case with X.25 networks, the connection establishment procedures can usually be separated from the normal flow of user data. This approach was adopted in the implementation of the X.25 signalling components of the Universe Basic Block Tunnel [Tennenhouse 84].

[2]For example, a portal that attaches an IEEE 802 network to an exchange supports communication with peer 802 networks. Portals do not directly support conversion functions that facilitate heterogeneous communication between devices that operate different end-to-end protocols or are attached to different types of local networks. Conversion services could be provided by attaching separate *conversion entities* to individual exchanges.

peer portals need only agree a scheme for the segmentation and reassembly of lengthy client packets. In contrast, PBX portals collect samples arriving on synchronous voice streams and transmit the *blocked* samples within CFR packets. In the opposite direction, incoming packets must be deblocked and the individual synchronous streams regenerated.

Portal designs must take account of the requirement to support dynamic connectivity involving some number of concurrently active peers. Although universal access is a desirable goal, the degree of potential connectivity may be limited by the management and addressing schemes of individual local networks.

6.5 Exchange Management

Portal communication is based on a simple SI-service that supports the exchange of CFR packets between *associated* peer portals. The portals invoke exchange management services to create and maintain associations on an out of band basis. This arrangement reduces the complexity of the in-band components that directly support the SI-service.

Exchange management services are divided into: the *layered* services that manage associations and exchange resources; and the *ancillary* services that provide additional management functions. Although the layered services and their protocols are fairly extensive they operate out of band with respect to each other and the normal flow of SI-user data. The services are built up in the traditional layered fashion but the protocol elements of the upper layers are not embedded in those of the lower layers as they are when the Open System Interconnection architecture is used. Some of the ancillary services provide functions, such as accounting and security, that may be embedded within individual ramps and portals. Others, such as directory functions, may be entirely separable from the exchange architecture.[1]

The uppermost of the layered services is the *secretary* service. This service, which is directly accessible to the SI-users, supports the out of band negotiation of inter-portal associations. Once an association is established CFR packets flow directly between the correspondents without further secretary intervention. In the case of a

---

[1]Directory functions could be supported by a public directory service accessed through a common carrier network.

request for a local association, between peer portals at the same site, the secretary provides each of the participants with the CFR address and association specific parameters specified by its peer. When the portals are located at different sites the service provides an out of band path for the negotiation of association specific parameters and address information. In these cases the secretary must also access the lower layer *window* service that is responsible for exchange resource allocation and the negotiation of inter-site bridges. The window service may, in turn, access the *channel* service which performs carrier dependent functions that facilitate the out of band control of the ramps and their attached NTEs.

Each of the layered management services is implemented by a collection of entities that reside within processing systems attached to the individual local exchanges. These entities exchange information with each other and with management *stubs* that reside within the in-band ramp and portal systems. Management interactions are supported by the SI-service over separate associations that operate in parallel with client portal associations.

6.6 Exchange Protocols

The exchange of information over SI-associations is structured in accordance with the *exchange protocol suite* described in Appendix F. This layered structure is based on:

- Lower layer services that support ATM packet transmission within the SI-layer;
- The Unison Data Link (UDL) service that supports the SI-service between associated SI-users located within exchange portal and management systems; and
- Upper layer services that operate over UDL.

The upper layer services operate over end-to-end associations between peer SI-SAPs and can be structured to suit the requirements of the individual SI-users. The upper layer services supported between peer portals are normally tailored to the specific requirements of their attached local networks. In contrast, the upper layer services used by management entities and their stubs conform to exchange conventions that facilitate management transactions.

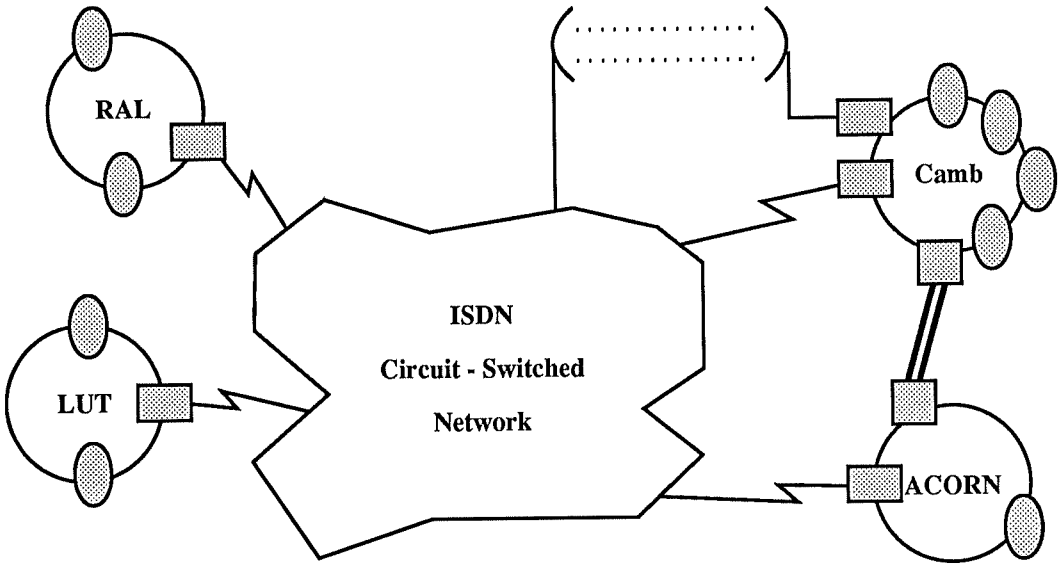

Figure 6.2: The Pilot Exchange Implementation

6.7 The Pilot Exchange Implementation

A pilot exchange implementation has been developed as the basis of the site interconnection infrastructure used within Project Unison. CFR-based local exchanges have been deployed at each of the participating sites as illustrated in Figure 6.2. Each site has some number of portals, ramps, and management systems attached to its exchange.

The NTEs at each site are supported by a prototype ISDN. Within this carrier network the individual NTEs are linked to a central circuit switch by primary rate (2 Mbps) transmission facilities. Each site is serviced by one NTE supported over conventional land lines, and the Computer Laboratory has a second NTE that is supported over a microwave transmission channel. In addition, a fixed 2 Mbps facility links the Computer Laboratory and Acorn sites.

The pilot implementation can be viewed as a large piece of experimental apparatus that can be arranged in a variety of configurations. In particular, a variety of exchange interconnection topologies can be investigated by varying the operation of the circuit switch to support public, closed user group, or hybrid operation. The flexibility of this apparatus makes it particularly useful for the experimental determination of exchange performance characteristics, and the evaluation of alternative routing and resource allocation strategies.

6.8 Summary

One of the goals of this research work has been to gain insights into site interconnection issues through the detailed design and implementation of the principal exchange components. The next five chapters of this dissertation describe the design and experimental analysis of the local exchanges, ramps, portals, and layered management services of the pilot implementation.

Chapter 7
Exchange CFRs

This chapter examines the use of the Cambridge Fast Ring as the switch fabric of the local exchange. The requirement for an ATM switch was identified early in the design of the exchange architecture. Two factors led to the rather pragmatic choice of the CFR as a switch fabric: the CFR design meets most of the exchange requirements; and implementation of the CFR was already under way. A great deal of time could have been spent developing a new switching technology but this work would not have provided insight into the overall site interconnection problem. Instead, the emphasis of this research has been on the experimental development of an exchange prototype. Within this context, the CFR has served as a good first approximation to the ideal switch. This experimental work with CFR-based prototypes has provided new insights into the desirable properties of the exchange switch fabric.

The design of the CFR Networking System is based on experience with its predecessor, the Cambridge Ring. The CFR is a distributed ATM switch intended for use as a Local Area Network. Host devices are attached to *stations* that support the exchange of client packets. Each packet consists of a 16 bit destination address, a 16 bit source address and a 256 bit data field. Peer stations exercise the empty slot protocol described in Appendix A to effect the transmission of individual packets. Each packet is embedded within a CFR slot that is ATD-multiplexed onto the ring transmission medium. The design provides for multiple ring configurations in which adjacent rings are linked by *bridges* that copy selected packets between their attached subnetworks. This arrangement is illustrated in Figure 7.1.[1]

[1]The CFR diagrams in this chapter have been adapted from [Hopper 86] which presents a detailed description of the CFR and its VLSI implementation.

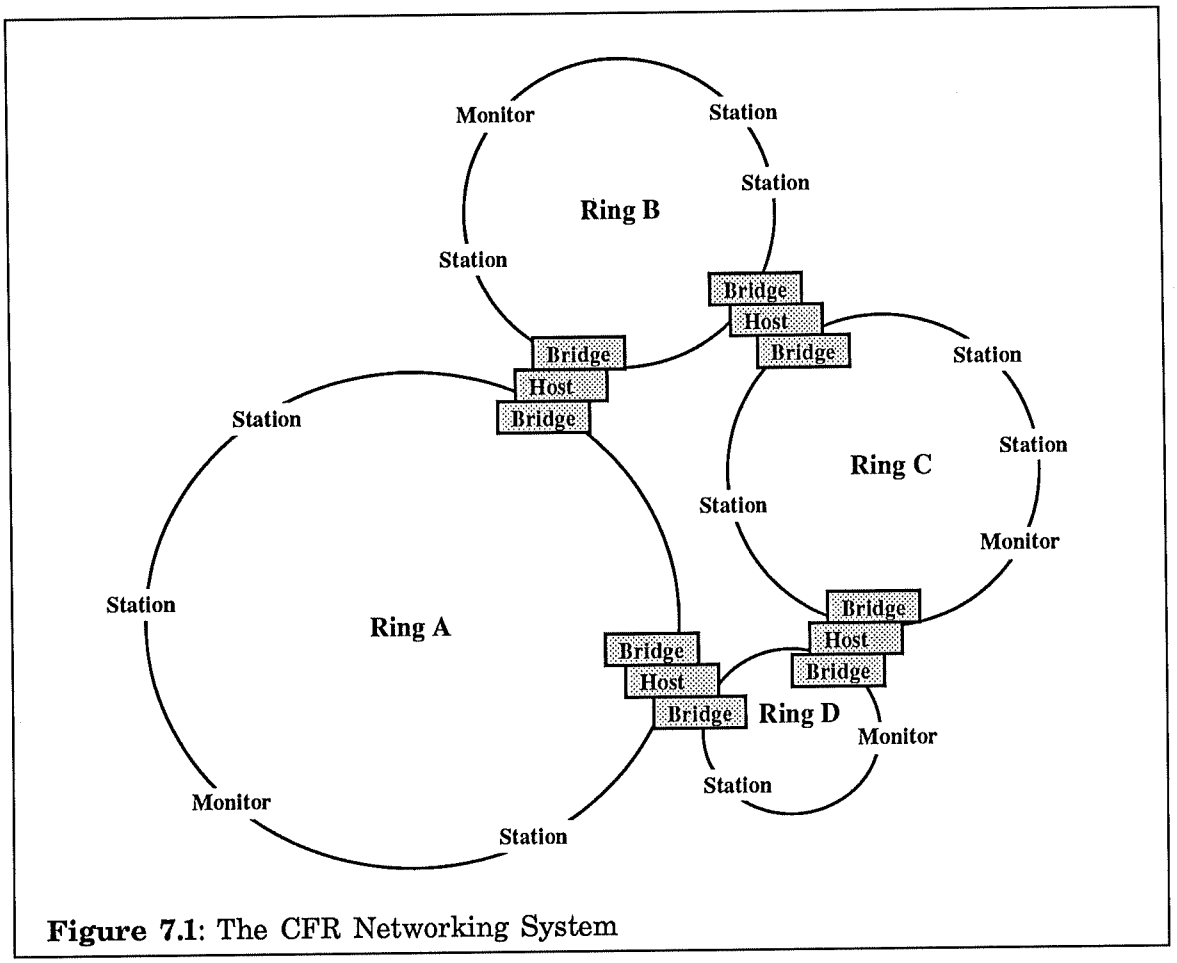

Figure 7.1: The CFR Networking System

7.1 Site Interconnection Encoding

The most important consequence of the decision to use the CFR technology has been the adoption of the CFR packet format as the basis of the common encoding used within the SI-layer. This choice of encoding has had an iterative effect on the design of many exchange components. For the most part the packet structure has proved satisfactory. The recommendations for improvements to the CFR reported in Appendix D are related to the access protocol and station implementation rather than the packet format itself.

The Host Data Field

The host data field of each packet is passed between SI-SAPs without modification. The CFR nodes treat the data field as an opaque structure whose contents are not examined or interpreted. Many of the considerations that led to the choice of a 256 bit data field for LAN applications are equally applicable to the exchange environment. The local exchange must allow traffic originating from a wide variety of local networks to be multiplexed and switched without incurring prohibitive ramp or portal overheads.

Some portals will be attached to local networks that support bulk transmissions and distributed systems traffic. If very short packets are used, then the portals' per packet overheads will be aggravated by the frequent requirement that packets be linked together to form larger blocks. A study of the Cambridge Distributed Computing System [Needham 82] reported in [Temple 84] has shown that, in that system, the distribution of block lengths is bimodal: over 80% of the blocks exchanged between hosts contain less than 32 bytes; and most of the remaining blocks are over 800 bytes long. Each of these short blocks can be embedded within a single CFR packet.

Other portals will be attached to local networks that support synchronous applications such as telephony. Long data fields result in packetization delays, sensitivity to lost packets and jitter arising from the coarse multiplexing of concurrent traffic streams. For example, when the SI-encoding is used to transport ADPCM encoded speech at 32 Kbps, the packetization delay per 256 bit data field is 8 msec. This falls safely below the maximum limit of 16 msec recommended in [Gruber 83a][1]. In comparison, the corresponding packetization delay for one Kilobyte data fields would be 256 msec.

In summary, a very short data field generates prohibitive overheads and long data fields impede the transmission of multi-service traffic. The 256 bit data field represents a compromise that permits exchange portals and ramps to support a wide range of communication services.

Address Fields

The length of the packet address fields is a function of the universe of discourse in which the addresses will be interpreted. In the exchange architecture, rings at different sites will be interconnected so that portals can exchange packets across site boundaries. Since the potential connectivity between portals should be universal, a global addressing scheme appears desirable.[2] On closer examination it can be seen that, from the perspective of a given portal, the universe of discourse can be restricted to the set of peer portals with which the portal is dynamically exchanging packets.

---

[1]Also, [Gruber 81] and [Gruber 83b].

[2]One could use an internationally recognized scheme such as that proposed in [Dalal 81] to assign unique addresses to CFR stations. This approach requires address fields that are relatively long relative to the 256 bit packet payload and therefore represent excessive overheads.

If shorter address fields are to be used then the universe of each address must be constrained. In the exchange architecture each site has its own CFR address space which is partitioned into a number of *windows*. One of the windows is statically allocated and an address within that partition is assigned to every portal or management station attached to the site's CFR. When communication is first established with a peer site, a window is *dynamically* assigned to facilitate communication with that site, and an address within that window is assigned to every station attached to the peer site's CFR. In this way the stations of the peer site are dynamically knitted into the address space of the local site. *Within the context of a single site's CFR* there is a unique address for every station, local or distant, that is dynamically accessible. To facilitate site interconnection, address field values are automatically translated whenever packets cross address space boundaries.[1]

The number of peer sites dynamically accessible from a single site is limited by the number of windows available. When communication with a peer site lapses, the addresses assigned to that site are reclaimed so that the window can be reassigned to some other peer site. The exchange addressing scheme permits a great deal of dynamic connectivity without incurring high per packet overheads. Management services dynamically establish communication between peer sites and retain control over each site's dynamic address space. Only the entities supporting these services operate within the large universe of discourse associated with universal connectivity.

7.2 CFR Implementation

The VLSI implementation of the CFR is based on the node design illustrated in Figure 7.2. The repeater logic performs clock recovery, byte alignment and the conversion of the incoming serial bit stream into an eight bit parallel byte stream. The incoming byte stream and a byte clock are presented to the controller logic which implements the slot protocol and provides the host device interface. The controller generates an outgoing parallel byte stream to the repeater, which in turn produces the outgoing serial byte stream.

The division of responsibility between the logic components means that only the relatively simple repeater, implemented in an ECL gate array of about 350 gates,

---

[1]The appropriate mapping is performed by the ramp nodes in accordance with the exchange addressing scheme described in Appendix B.

is clocked at the full CFR transmission rate. The much more complex controller, implemented in a CMOS chip of about 25000 gates, is clocked at one eighth the speed. The cost of this simplification is that the electrical delay through every node is substantial: three bytes of delay through every repeater and two bytes of delay through every controller.

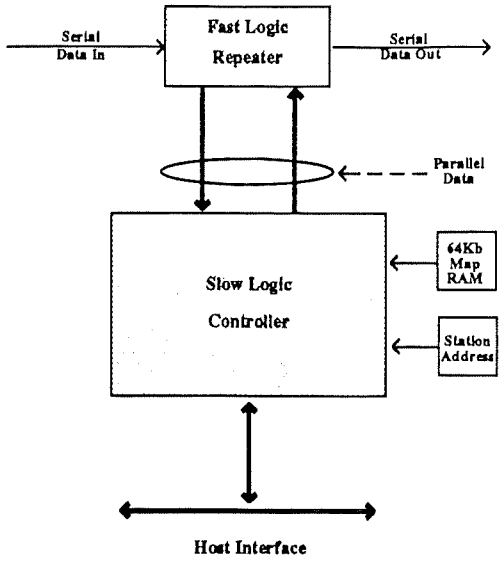

Figure 7.2: CFR Node Design

The controller chips are quite flexible and can operate in a number of configurations. In particular, each chip can operate in one of three modes: *monitor*, *station*, or *bridge*. In monitor mode the controller is used to create and maintain the ring slot structure, to monitor ring operation, and to receive maintenance information from other nodes.

Station Mode Controllers

In station mode the operation of the controller is optimised for use by packet sources and sinks. To transmit a packet, the host device writes the two byte destination address field into the controller's destination register and copies the 32 byte data field into the transmit FIFO. When the last data byte is written the transmit logic of the controller is *armed*, and a packet containing the FIFO data, the destination address, and the fixed source address of the station is transmitted within the next available slot.

The transmit logic waits for the slot to circulate around the ring and examines the returned response code. If the transmission has been successful, or cannot be safely repeated, then the host is notified that the transmission has been completed. If the response indicates that the transmission should be repeated, the controller automatically retries the slot without further intervention by the host device. This process continues until either the transmission terminates normally, or a preset repeat limit is exceeded. In this case the host is notified that the packet has been thrown on the ground (TOG'd) by the controller.

The station mode receive logic examines every full slot on the ring and compares the destination address of the packet to the station address of the controller. If the controller's receive FIFO is not empty then the station is busy and the packet

is rejected. If the station is not busy and the packet is *selected* for reception[1], then the source address and data fields embedded within the slot are copied into the source register and receive FIFO of the controller. The host device is notified that a packet has been received and the controller cannot receive another packet until the host device empties the receive FIFO by reading or discarding the packet data.

Bridge Mode Controllers

A simple fast ring bridge can be constructed using two controller chips attached to different rings. In a typical bridge configuration, the controllers are connected to a common host device which acts as the bridge manager. The manager uses the host interfaces to configure the controllers, adjust their address lists, and load/unload packets. The receive logic of each controller extracts packets from its own ring so that they can be copied to the opposite controller and injected into the adjacent ring. The manager always copies all 36 bytes of each packet, including the 256 bit data field and both of the address fields.

When a controller chip operates in bridge mode its receive logic must decide whether or not a given packet should be selected for reception. Each controller maintains a list of the station addresses associated with destinations whose packets are to be routed through the bridge.[2] When a slot containing a packet bearing one of these destination addresses passes the controller the packet contents are copied into the controller's receive FIFO and the response field is marked as if the bridge were the final destination of the packet.

7.3 The CFR as an Exchange Switch

The use of the CFR as the switch fabric of a local exchange has resulted in various extensions to the CFR architecture, especially with respect to bridge construction and the management of the CFR address space. The bridge features of the CFR node controller provide hardware support for the window-based service supported by exchange ramps.

---

[1]The station mode configuration provides two selection mechanisms that can be used to restrict the source(s) of received packets.

[2]The list of favoured addresses is represented by a bit map stored within the attached memory chip.

7.3.1 Ramps and Bridges

The node that attaches a ramp to a local exchange is built out of a CMOS controller chip operating in bridge mode. Packets extracted from the local CFR are transmitted over one of the channels supported by the ramp. The ramp at the other end of the channel presents the incoming packets to its controller, which injects the packets into the peer site's CFR. These ramp-supported bridges support the direct exchange of packets between peer portals attached to different exchanges. So long as the address lists within the bridge mode controllers are correctly configured, packets will flow between sites with minimal in-band intervention.

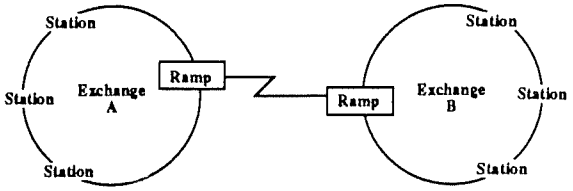

Figure 7.3: An *Elongated* Bridge

If the ramp channel operates over a fixed transmission facility, then the effective result, depicted in Figure 7.3, is an *elongated bridge* which is not much different from the ring-ring bridges illustrated in Figure 7.1. The bridge manager functions have been distributed amongst the peer ramps which exercise the bridge mode features of their attached controller chips. The principal difference is one of administration: the elongated bridge allows packets to cross site boundaries, and so the ramps must perform additional in-band functions such as address field translation.

If the channel is constructed using a switched carrier service, then the bridge topology can be dynamically adjusted. Furthermore, each ramp can concurrently support a number of bridges by constructing separate channels to different peer sites. In effect, the ramps are supporting a dynamic *dial-a-bridge* service by overlaying the CFR packet encoding on the switched carrier service.

7.3.2 Physical Organization

At most sites the number of exchange nodes will be quite limited and it is possible to arrange the topology of the local networks and the carrier termination points so that all of the exchange nodes can be located in a single chassis. In this case the

length of the entire ring will be about a metre, and the CMOS controller chips can be centrally clocked and directly linked into a byte parallel ring.[1]

This organization, which has been used in the pilot implementation, is illustrated in Figure 7.4. An exchange is constructed by plugging some number of controller modules into a chassis backplane. The backplane distributes the clock signal and daisy chains the ring output of each controller to the ring input of its downstream neighbour. A single ribbon cable closes the ring by carrying the parallel output of the last module back to the input of the first. The electrical length of these relatively short rings is almost entirely a function of the two byte delay through each of the controller nodes. When the exchange CFR is less than one slot in length, a programmable shift register is inserted into each ring to ensure that it is long enough to support a frame structure consisting of a single slot and a few bytes of gap.

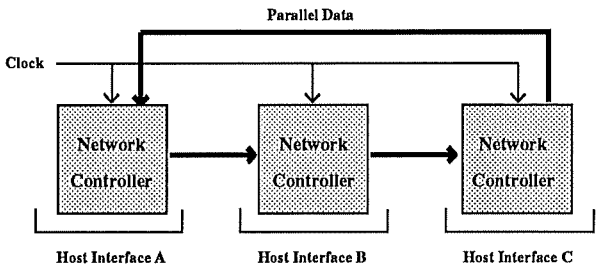

Figure 7.4: An Exchange CFR

7.3.3 Performance Considerations

The CFR provides a high aggregate bandwidth which is dynamically shared by the concurrently active portals and ramps. Furthermore, the empty slot protocol limits the maximum access delay experienced by jitter-sensitive traffic, and the automated transmission retry scheme facilitates ATM-based rate adaption. The performance characteristics of the CFR are similar to those of the Cambridge Ring and the equations developed in Appendix A are directly applicable.

The aggregate system bandwidth, *SysBw*, is determined by the clocking rate which is limited by the CMOS controller implementation. Although the controller chips operate correctly at 8 Mhz, a more conservative 5 Mhz byte clock is used in the pilot implementation. For a single slot frame structure with a four byte gap, equation A.1 yields:

---

[1]The parallel organization avoids the complexity and byte delays associated with the repeater logic.

SysBw = 30.5 Mbps at the present 5 Mhz rate; or
48.9 Mbps at the 8 Mhz rate that would be used in a production environment.

The maximum transmission bandwidth, *MaxTxBw*, that can be consumed by a single ring node can be calculated using equation A.5. The actual throughput that can be achieved is dependent on *D*, the delay between successive transmissions. The minimum delay incurred using the present controller chips corresponds to 2 ring slots, and so:

MaxTxBw = 7.6 Mbps at the 5 Mhz rate; or
12.2 Mbps at the 8 Mhz rate.

In practice, the delay between transmissions will be a function of the attached host device. Experience with the Universe portal described in Chapter 9 has shown that a nominal delay of 3 slots can be achieved using a program controlled host device. It is to expected that the higher values of *MaxTxBW* can be realized using more sophisticated devices.

The minimum transmission bandwidth, *MinTxBw*, guaranteed to every CFR node, is governed by equation A.6. On a pilot CFR with ten nodes the guaranteed allocation when all nodes are concurrently active is approximately 2.8 Mbps. Under these circumstances, *BusyDelay*, the worst case access delay given by equation A.7, evaluates to 3024 bits, which corresponds to 75.6 usec at the 5 Mhz clock rate.

The overall transmission performance is consistent with exchange applications involving a number of ramps and portals that individually operate at peak rates of about 10 Mbps and offer sustained loads of around 3 Mbps. This operating range is especially suited to pilot exchange configurations where ramps are attached to synchronous carrier services such as the 2 Mbps ISDN facility, and portals are attached to asynchronous local networks such as the 10 Mbps Ethernet.

A final performance consideration is the contention that arises when a number of CFR nodes are concurrently transmitting packets to a common target. The present design does not ensure that the receive bandwidth available at a given node is *fairly* distributed amongst the active transmitters. Receiver contention is a particularly serious problem for ramps which are funnels for off-site traffic arising from concurrently active portals. A detailed analysis of CFR performance and the implications of receiver contention is presented in Appendix D.

7.4 Alternative Switch Fabrics

The exchange architecture and its packet format are not dependent on the present CFR technology. As new semiconductor devices become available, they can be used to construct exchange CFRs supporting higher transmission throughputs. For even higher speed applications an alternative switch fabric may be employed. An important feature of the ATM-based design is that its rate adaption capability supports the inter-operation of local exchanges operating at different rates or using different switch technologies.

One alternative technology is the CBN slotted ring described in Chapter 4. In addition to its higher aggregate capacity, the CBN's slot protocol allows a single node to seize a larger fraction of the available bandwidth. Other distributed ATD structures such as QPSX or Orwell[1] could also be used. However, these schemes lack the low level *response* mechanism provided by the CFR. In the exchange application this feature is used, in conjunction with automated retransmission, to facilitate rate adaption and the detection of receiver contention. There is no similar mechanism in the QPSX and Orwell schemes, and so, brief but periodic bursts of receiver contention could effectively block the flow of segmented upper layer messages.[2]

A further alternative is to use a centralized switch fabric based on ATD or space division techniques. In [Milway 86] it has been suggested that a binary routing network (BRN) could perform local exchange switching, and a suitable BRN design is described in [Newman 88b]. The major obstacle to the use of central switch fabrics arises from the regular nature of their VLSI implementations. These fabrics are suited to switching environments involving large numbers of ports operating at similar transmission rates. In contrast, the local exchange environment is characterized by a relatively small number of ramp and portal nodes that generate asymmetric traffic loads.[3] A further difficulty with these switches is that contention at their output ports generates back pressure that results in queue build-up at the input ports. Newman's design addresses this

---

[1]The Orwell slotted ring [Falconer 85b] uses a *destination delete* protocol that increases available system bandwidth and controls access delay.

[2]In the absence of a response and retry scheme, a short contention burst will cause single ATM packets to be dropped thereby corrupting the reassembly of upper layer messages.

[3]For example, exchange ramps are expected to act as *funnels* that concentrate traffic arising from a site's local networks.

problem by incorporating two switch planes and a priority arbitration scheme that supports the expedited transfer function.

7.5 Summary

The CFR networking system has many attributes that recommend it for use in the exchange environment:

- The CFR packet structure provides a suitable SI-encoding;
- The VLSI implementation simplifies the construction of the switch fabric; and
- The overall performance characteristics are consistent with exchange requirements.

The station and bridge mode controllers embedded within the exchange deliver packet-based services normally associated with the OSI network layer. Exchange ramps extend the CFR bridge design to support inter-site communication by transferring packets between sites with minimal intervention. From a portal's perspective, there is little difference between the intra-site and inter-site communication. Furthermore, since the packet switching function is distributed over the nodes of a ring, the dimensions of the fabric are easily altered: the number of portals and ramps attached to a local exchange is easily changed by adding or removing controller modules.

Chapter 8
Exchange Ramps

This chapter discusses various aspects of ramp design within the context of the site interconnection issues and SI-service description presented in Chapter 5. A carrier independent description is first developed and this generic model is then used to describe the existing ramp implementations. The first of these, the Bailey ramp, supports a single fixed channel to a statically determined peer site. The second design, the ISDN ramp, represents a more general scheme that supports switched, variable bandwidth channels to dynamically selected peer sites.

8.1 Ramp Design Issues

Ramps operate within the SI-layer to extend the range of the CFR packet transfer function. Exchange ramps must ensure the sequence-preserved and error-free delivery of packets as they are transferred between exchange CFRs. They should avoid the introduction of acknowledgement, retransmission, flow control and related in-band functions that would impair the overall performance of the SI-layer.

The following subsection reviews the manner in which ramps and their associated channels are used to support exchange address windows. This is followed by a generic description of a single channel ramp. Functions such as expedited delivery, multiplexing, switching, segmentation and splitting are introduced through the progressive addition of functionality to this basic ramp description.

8.1.1 Channels, Windows and Ramps

Exchange ramps extend the packet transfer function by supporting CFR bridges between the local exchanges of peer sites. Each bridge is a concatenated path that traverses a pair of peer ramps and a common carrier supported channel. The ramp at one end of the bridge extracts selected packets from its local exchange and

transmits them over the channel to the peer ramp, which injects the packets into the peer exchange.

Channels

Channels may be constructed using a variety of services supported by private or common carrier networks. Peer ramps operate an agreed encoding to *format* these lower layer services into channels that support CFR packet transfer. Depending on its design, a ramp may be capable of supporting a number of concurrent channels, and each of these channels represents a distinct inter-site bridge.

Windows

Address windows represent dynamically assigned addressing paths between peer exchanges. Window creation and deletion is negotiated on an out-of-band basis by the window entities of the peer sites. Each window is identified by a locally assigned window value that is used in the destination window field of packets destined for the peer site. Associated with the window is a <source, destination> pair of window values assigned to the window by the window service of the peer site. The window entity binds each window to a specific bridge by arranging for a local ramp to support the window on a specific channel.[1] The designated ramp is supplied with the window value, the pair of values supplied by the peer site, and other window-specific information. The binding of windows to bridges may be transient, and the window service can unbind windows and either suspend them or bind them to other bridges.

Ramps

The designated ramp supports the window by extracting corresponding packets from its local exchange and transmitting them along the specified channel.[2] When a packet crosses a bridge it traverses the boundary between the independent addressing domains of the peer sites. The ramp uses the supplied window value pair to translate the window values within the packet address fields in accordance with the exchange addressing scheme described in Appendix B. A number of windows can be bound to a single channel, and the ramp must ensure that the appropriate pair of peer site values are used during address translation.

---

[1]When intermediate sites are used to support relaying, each bridge represents a single hop along a concatenated path between the end sites.

[2]When concurrent channels are supported the ramp must ensure that each extracted packet is routed to the appropriate channel.

Discussion

The CFR packet format is the common encoding that is used within all of the in-band components of the SI-layer. The portals, ramps, and CFRs support the integrated switching and multiplexing of packets as they are routed between peer SI-users:

- Within a source portal, the packets of different associations are multiplexed together into a packet stream that is presented to the CFR station node. This stream is ATD-multiplexed onto the CFR medium, where individual packets are demultiplexed on either a *portal-specific* or a *window-specific* basis. Packets destined for a peer portal at the local site are switched to the corresponding station node, and packets destined for a portal located at a remote site are switched to the CFR bridge node of the appropriate ramp;
- The multiplexed packet stream arriving at a portal consists of packets generated at a variety of peer portals. This stream is demultiplexed so that the packets corresponding to individual associations are presented to the appropriate SI-users; and
- The multiplexed packet stream arriving at a ramp contains packets generated at a variety of source portals. The stream is demultiplexed on a window-specific basis, and the resultant streams are switched and multiplexed so that each packet is transmitted on the appropriate inter-site channel. At the peer ramp, packets arising from concurrent channels are multiplexed together and presented to the CFR node where they are ATD-multiplexed onto the medium. These packets are switched, on a portal-specific basis,[1] to the appropriate peer portal.

8.1.2 Ramp Organization

Figure 8.1 illustrates the major functional units within a basic ramp that supports a single bidirectional channel.[2] By convention, the stream of packets from the local CFR and towards a peer ramp is referred to as the transmit stream, and the components associated with this stream make up the transmit half of the ramp. Similarly, the incoming stream of packets received from a peer ramp is referred to

---

[1]When relaying is performed, incoming packets are switched, on a window-specific basis, to the next ramp along the relayed path.

[2]Since most carrier services are bidirectional in nature, the windows, ramps, and channels described in this dissertation are assumed to be bidirectional. There is no architectural requirement for bidirectional channels.

as the receive stream, and the associated components make up the receive half.[1]

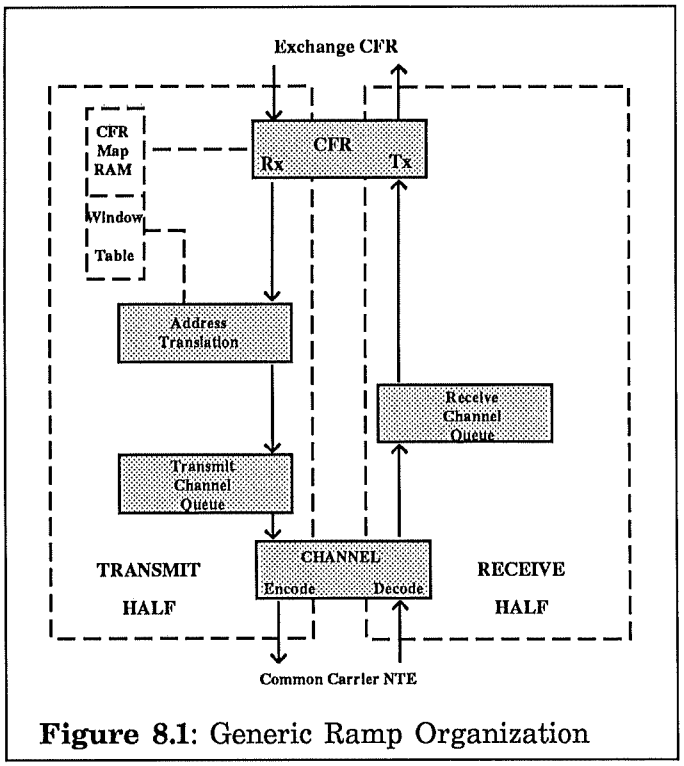

Figure 8.1: Generic Ramp Organization

The Transmit Half

The transmit half uses a bridge mode CFR node to extract selected packets from the exchange CFR. The node receives packets bearing destination addresses that have been selected in the bit map stored within the node's map RAM. The transmit half configures the bit map to select all of the addresses that fall within the window(s) bound to the ramp's channel.

Selected packets, complete with their address fields, are unloaded from the node by the transmit half. The values in the window components of the address fields are translated from the local address space to that of the peer site, and the packets are placed in the transmit channel queue. When multiple windows are concurrently bound to the channel, the address translation process uses the destination window value within each packet as an index into a table of window value pairs. Address translation can be viewed as a simple filtering operation that is performed as packets are unloaded from the CFR node.

The Receive Half

Incoming packets are placed in a common receive channel queue and are presented to the CFR node in the order of their arrival at the receive half.[2] The address fields within these packets have been translated by the peer transmit half, and so the packets can be directly loaded into the node which transmits them on the local CFR.

---

[1]This distinction is logical rather than physical. Some of the ramp components, such as the CFR and common carrier points of attachment are physically shared by the two halves of the ramp.

[2]In some cases, described later in the text, variations in this strict service discipline may be necessary.

The Channel

At the level of abstraction depicted in Figure 8.1 the formatted channel can be viewed as a sequence preserving stream that removes complete packets from the transmit channel queue of a ramp half, and inserts the packets in the receive channel queue of the peer half. Entire packets may be lost as they traverse the channel but packets that are inserted in the receive queue are complete and error-free. The transmit half formatting functions include the encoding of CFR packets for transmission over the carrier service. Similarly, the receive half functions include the decoding of receive carrier symbols and the regeneration of the transmitted packets.

Summary

The ramp halves operate on individual packets using window-specific state information, retained in the window table and the CFR map RAM. Ramp operations are independent of specific associations between peer SI-users and the ramp halves do not acquire or maintain association-specific state information. Furthermore, the service provided by the ramp has been structured so that the implementation of the receive and transmit halves can be streamlined and isolated from each other.

8.1.3 Rate Adaption

The packet retry mechanism implemented within the CFR nodes provides a limited form of rate adaption between ramps and their CFR-based clients. The queues, located at the channel termination points within the ramps, function as elastic buffers that provide further rate adaption between the ATM-based packet stream and the carrier supported channel. This buffering is particularly important when channels are built upon STM-based carrier services.

At the transmit half, the CFR node can receive bursts of packets at speeds which exceed the fixed channel capacity. The transmit queue absorbs the bursts thereby smoothing the packet stream that is presented to the channel. At the receive half, incoming packets arrive synchronously at channel speed, while the CFR transmission rate is adapted to match the capacity of individual clients. The receive queue expands to store incoming packets during transient intervals when packet transmission is constrained by relatively slow clients.

8.1.4 Concurrent Channels

The basic design is easily extended to describe multiple channel ramps. The receive half becomes a funnel that multiplexes traffic from a number of channels, onto a single CFR node. The transmit channel queue is partitioned into independent channel-specific queues, and the transmit half routes each incoming packet to the appropriate queue. Channel assignment, which is based on the destination window value within each packet, can be performed in parallel with address translation. The window table within the transmit half is expanded so that the table entries, accessed at translation time, record the channel binding of each window.

The overall ramp design must ensure that a burst of packets associated with one channel does not exert *back-pressure* on packets associated with the remaining channels. In particular, transmit queue overflow associated with one channel must not interfere with the processing of parallel channel traffic at the CFR node.

8.1.5 Expedited Transfer

The SI-service supports expedited packet transfer on an association-specific basis. When associations traverse site boundaries, the layered management services arrange for *expedited* and *normal* associations to be supported through different address windows. The ramp halves arrange for packets passing through expedited windows to receive priority as they are passed between exchange CFRs. Each half will have some bottleneck component, at which the flow of packets is constrained and packet queues may develop. When expedited transfer is implemented the expedited packets must be identified before they reach this queue so that they are not unnecessarily delayed by normal packets waiting at the bottleneck.

Within the transmit half, the sorting of packets into expedited and normal streams should be performed as soon as the packets are unloaded from the CFR node. As is the case with address translation and channel routing, expedition processing is based on the examination of the destination window value located within each packet. The window priority is recorded in the Window Table entries so that all three operations can be performed in parallel.[1]

---

[1]Window priorities are assigned and supplied by the window service when windows are bound to ramps.

The channels themselves will be the bottlenecks within many transmit halves, and so packets arising from an expedited window are granted priority access to their channel. Each channel queue is subdivided into separate expedited and normal queues. Access to the channel itself is based on a nonpreemptive priority queueing discipline that grants packets in the expedited queue priority over those in the normal queue.

Receive half bottlenecks develop when the aggregate throughput of the channel(s) exceeds the transmission capacity available at the CFR node. This capacity varies dynamically depending on the level of CFR rate adaption being used to support communication with ramp clients. At receive halves that implement packet expedition, the queue of packets to be injected into the CFR is subdivided into separate expedited and normal queues, and access to the CFR is based on a nonpreemptive priority queueing discipline. Sorting is based on the examination of the source window value located within each incoming packet.[1] The priority assigned to a window is determined by the management of the *receive half's* exchange and is independent of the priority assigned at the transmit half.

In order to perform expedition processing the receive half must have some means of associating priority information with window values.[2] The window service can explicitly supply the receive half with a window-specific priority value whenever a window is bound to an incoming channel. Alternatively, the window service can establish an implicit assignment convention whereby window values that fall within a certain range are only assigned to *expedited* windows. The receive half of a ramp can then sort incoming packets through the simple examination of their source window values.

Discussion

The exchange management services can arrange for expedited associations, carrying isochronous and other jitter-sensitive traffic, to be bound to expedited windows. At exchange ramps the expedited traffic will be placed in separate queues and thereby insulated from bursty asynchronous traffic. This scheme should work well provided the packet rate of the expedited traffic is limited and does not exceed the bandwidth of any ramp bottlenecks. Within a ramp the maximum queue lengths

---

[1]This value associates the packet with a window allocated from the address space of the receive half's exchange.

[2]In contrast, a receive half that does not implement the expedition function does not require any information concerning the windows that are bound to its channels. Individual receive half designs should be carefully examined to determine whether or not the benefits associated with expedition justify the additional complexity.

associated with expedited packets will be quite short. In comparison, the queue lengths associated with normal traffic may grow to be quite long as the ramp absorbs bursts of asynchronous traffic.

It is claimed that the dual priority scheme will support a reasonable mix of multi-service network traffic. Although the degree to which additional priority levels would improve this support is an open question, the present scheme could easily be extended. Similarly, the present priority queueing discipline could be replaced with a weighted discipline that gives each window priority access to an assigned portion of the channel bandwidth

It has been suggested that the CFR packet header be extended to include an explicit expedited transfer flag. One difficulty with this approach is that the initial packet priority would be determined by a source node at a given site. It may prove difficult for exchange management to allocate inter-site channel bandwidth without exercising control over the use of expedited transfer. Similarly, the priority of a packet that is determined by a portal at one site may not be appropriate at the peer site. Thus, in order to preserve site independence, exchange ramps may be required to modify priority bits as packets traverse site boundaries. At the present time the relative merits of packet-specific expedition over window-specific expedition remain unproven. It is hoped that further work within Unison will lead to the experimental evaluation of both schemes.

8.1.6 Sequenced Delivery

Peer ramps can preserve the sequence of packets as they are passed between local exchanges by:

- Labelling packets as they are extracted from the CFR at the transmit half and reordering them as they are injected into the CFR at the receive half; or by
- Arranging for packet sequencing to be preserved by the channel and within the ramp halves.

Since most common carrier services are themselves sequence preserving, the latter approach to sequencing is preferred. At the single channel ramp of Figure 8.1, packet sequences are preserved if the channel is sequence preserving and the channel queues obey a FIFO discipline.

The sequence preservation function of the SI-service applies to individual associations between SI-users. The ramps need not preserve the sequence of packets arising from different associations, and so the strictly sequential regime

proposed above is unnecessarily restrictive. Although it simplifies the implementation of the single channel ramp, it is not compatible with the support of concurrent channels or the expedited transfer function. On the other hand ramp operations are meant to be association-independent and so some compromise regarding the degree of sequence preservation is desirable. An alternative policy is to insist that the peer ramps preserve the sequence of packets received through the same window. This degree of sequence preservation is consistent with the other window-specific functions of the ramps.

At the transmit half this policy is easily implemented by enforcing a FIFO queueing discipline at each of the independent channel queues.[1] At the receive half the packets arising from a number of windows are multiplexed onto the CFR node. When a window-specific queueing discipline is operated a burst of packets destined for a relatively slow CFR node can severely delay packets bound for other destinations. At the risk of architectural inconsistency, but without otherwise complicating the ramp, the window-based policy can be relaxed so that sequence preservation only applies to packets bearing identical destination addresses.[2] This approach minimizes the interference attributable to rate adaption without introducing association-specific processing into the ramp.

8.1.7 Channel Formatting

An exchange channel implements an error-protected sequenced packet stream operating over a common carrier service. Typically the carrier service supports sequenced symbol streams of bits or octets, and the peer ramp halves operate an agreed encoding based on the lower layer symbols. The following paragraphs review some of the generic channel formatting issues that affect ramp design. Many aspects of channel formatting are network-dependent, and so the specific formatting techniques that have been investigated are discussed in later sections of this chapter.

---

[1]This approach satisfies a somewhat stronger condition in that it is sequence preserving on a channel and priority basis rather than window-specific.

[2]This policy can be partly realized without implementing destination-specific queues within the receive half. The FIFO discipline of a single queue can be altered, so that when the packet at the head of the receive channel queue cannot be immediately delivered to the peer CFR node, subsequent packets in the queue bearing different destination addresses are selected for transmission.

Packet Synchronization

The transmit ramp half segments CFR packets into lower layer symbols and transmits the segments to the receive half where they are reassembled into complete packets. The channel encoding, which specifies how packets are segmented, may require that distinguished *packet synchronization* symbols be embedded at appropriate points in the lower layer stream. These symbols, which are stripped from the stream at the receive half, are used to identify packet boundaries.

Error Detection

The SI-service supports error detection on a hop-by-hop basis: paths across exchange CFRs are protected by CRC processing within the node controller chips; and the ramp halves ensure that inter-site paths are similarly protected. Unless the common carrier service supports error detection within the lower layer symbol stream the ramp halves must provide this function themselves. Transmitted packets are accompanied by check symbols, such as CRCs, that are stripped by the receive half and used to detect errors in reassembled packets.

Splitting

Peer ramps can use a number of lower layer symbol streams, operating in parallel, to support a single channel. In this case the channel encoding specifies how the packet stream is split into parallel symbol streams at the transmit half, and recombined into a sequence-preserved channel stream at the receive half. The encoding may require that distinguished *channel synchronization* symbols be embedded within the parallel lower layer streams.

8.1.8 Ramp Management

Exchange ramps can be viewed as filters that retain minimal amounts of state information in relatively simple tables. Although many channel functions, such as expedition and segmentation, must be performed in-band within the ramps, most ramp management functions can be extracted from the individual ramps and supported within separate processing elements. These management services configure the ramps by issuing simple commands that examine and update their table entries.

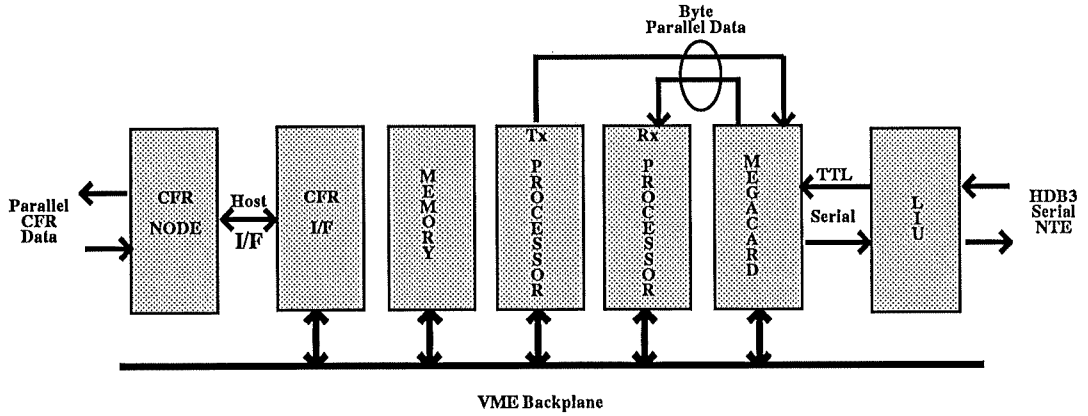

Figure 8.2: Bailey Ramp Organization

8.2 The Bailey Ramp

The Bailey ramp supports a single packet channel over fixed transmission facilities. The principal ramp components are a CFR interface module, the channel interface modules, a memory board, and two processor boards. The channel interfaces of peer ramps are linked by a carrier circuit that terminates at 2 Mbps NTEs that conform to CCITT recommendation G.703.[1] In the present implementation, the configuration of the ramp, including the window table, is statically determined when the ramp processors are initialized.

8.2.1 Ramp Organization

The ramp components are mounted in a single chassis and are linked by a VME backplane bus as illustrated in Figure 8.2. The processor and memory boards are off-the-shelf modules purchased from a commercial supplier.

The CFR Interface

The CFR interface is a general purpose card developed at the Computer Laboratory to connect VME-based systems, including portals and exchange management components, to ring nodes. In the Bailey ramp the interface card is attached to a CFR node that is configured to operate in bridge mode. Processors can access the memory-mapped interface over the VME bus in order to:

- Load packets into the node;

---

[1] [CCITT G.703] specifies the electrical characteristics of the NTE. In the UK, British Telecom offers a suitable service referred to as Megastream. North American carriers offer a similar 1.5 Mbps service based on their T1 technology.

- Unload packets from the node;
- Examine the node status; and
- Modify the node configuration.

When the ramp is initialized one of its processors initializes the nodes's address RAM to receive packets bearing addresses associated with the windows supported by the ramp.

The Channel Interface

The Channel Interface is divided into two modules, referred to as the line interface unit (LIU) and the megacard. The LIU supports the HDB3 encoding and electrical signal levels that must be provided at a G.703 interface. The megacard imposes the timeslot frame structure specified in CCITT recommendation G.732 on the carrier circuit.[1]

The LIU is attached to the NTE and performs the appropriate code conversions so that symbols and clocking information can be exchanged with the megacard which operates at standard TTL levels. The serial receive and transmit symbol paths are synchronously clocked in accordance with a clock recovered from the incoming HDB3 signal or a local clock supplied by the LIU.

When configured for use with the Bailey ramps, peer megacards implement serial octet streams by exchanging client octets during successive selected timeslots. Up to thirty timeslots within each frame are made available to the megacard client at two directional TTL busses carrying clock, timeslot address, and parallel data signals.[2] Transfers over these busses are synchronous and are controlled by the megacard which supplies the clock signals. During each receive interval the megacard shifts incoming serial symbols into a buffer, and during the subsequent interval the corresponding timeslot address and parallel data are presented to the client on the receive bus. Similarly, during each transmit interval the megacard places a timeslot address on the transmit bus. The client supplies the corresponding data octet, and during the subsequent interval the megacard shifts the individual bits of the octet onto the outgoing serial stream.

---

[1][CCITT G.732] specifies the 32 timeslot frame structure used in IDN and ISDN. [Rainforth 87] contains a detailed description of the design which was jointly developed by the RAL and Logica.

[2]The first timeslot of each frame (TS0) supports the exchange of frame synchronization and maintenance information. The octets of the sixteenth timeslot (TS16), which ISDN reserves for signalling purposes, are converted into a 64 Kbps serial stream that can be routed to an out-of-band signal service. In the Bailey ramp configuration the timeslot address information provided on the client busses is ignored, however, this information is of significance in other configurations.

Although the VME bus does not support the in-band flow of megacard data, it does provide access to the on-board registers used to monitor and configure the channel interface. One of these registers allows the client to select a subset of the 30 data timeslots. The client busses are only strobed at selected timeslot intervals and during unselected timeslots the megacard does not drive the outgoing signals to the LIU. This arrangement permits a number of megacard modules to be attached to a single LIU interface, provided the timeslot sets selected at the parallel megacards are mutually disjoint.

8.2.2 Transmit Half Operation

Activity within the transmit processor[1] is triggered by the arrival of a packet at the CFR node. The packet is unloaded through the ring interface and a simple table lookup is used to perform address translation. The processor calculates check symbols for the packet, and a parallel interface located on the processor board is used to pass the octets of the packet and the check sequence to the megacard. The transmit processor inserts a distinguished *start* symbol into the data stream immediately before the first octet of the packet. Once the start code is transmitted the processor polls the clock signal supplied by the megacard to ensure that successive octets of the packet are transmitted in successive selected timeslots.

All of the transmit operations are performed sequentially. There is no channel queue within the transmit half, and each packet is completely processed before the processor returns to the CFR interface for another packet. The asynchronous nature of the incoming traffic and the sequential organization of the transmit half will lead to idle periods during which packets are not being transmitted on the channel. The transmit processor arranges for the parallel interface to pass distinguished *idle* symbols to the channel during the periods between packet transmissions.

---

[1]The ramp processor attached to the megacard transmit bus is referred to as the transmit processor and the processor attached to the receive bus is referred to as the receive processor.

8.2.3 Receive Half Operation

The receive processor scans the incoming octets supplied by the megacard. When a start symbol is detected the processor copies consecutive incoming octets into a packet buffer located within the shared VME memory. The number of octets copied corresponds to the length of a complete packet and its accompanying check sequence. Once the packet has been copied into memory the receive processor updates a buffer pointer and continues to scan the incoming stream, ignoring idle symbols and waiting for the start symbol that delineates the next packet.

The *transmit* processor is used by both halves of a Bailey ramp. When it is not processing a transmit half packet, the processor removes receive half packets from the shared buffer and transmits the packets over the CFR. In the current implementation the transmit processor is also responsible for verifying the check sequence of receive half packets and deleting packets that have been corrupted during transmission. This arrangement has two advantages:

- Only one processor has access to the CFR interface; and
- The operation of the receive processor is streamlined so that the processor does not miss incoming start symbols.

8.2.4 Summary

The Bailey ramp processors segment CFR packets into octets that are transmitted over a serial stream. The megacards perform the splitting function by placing successive octets of the stream into timeslots associated with different synchronous subchannels. Within the megacards, the G.732 frame structure is used to support the STD multiplexing of up to thirty synchronous subchannels onto a single G.703 circuit.

The megacard design permits a number of ramps to be attached to a common LIU and thereby share access to a single NTE. Time division switching equipment, located within the carrier network, can be used to direct the timeslots associated with different ramps to circuits terminating at different peer sites. In this configuration each ramp attached to the LIU supports a single channel to a designated peer site.

Like its civil engineering namesake, the Bailey bridge, the primary ramp design criterion was ease of construction[1] as opposed to throughput. These ramps have been used to support the testing and integration of other exchange components including the CFRs and portals. The description of their design has been included in this text to illustrate how easily single channel ramps can be implemented. The following section of this chapter describes a substantially more complex design that supports greater throughput and functionality.

8.3 The ISDN Ramp

The ISDN ramps provide the full range of channel configurations and bandwidths that can be supported at the primary rate ISDN interface. These ramps exercise multiplexing and switching functions, embedded within the ISDN, to construct channels between peer sites. Routing services within the ISDN support the dynamic establishment of 64 Kbps calls between any pair of peer NTEs. The ISDN interface is based on the G.703 encoding and G.732 frame structure, and, at each of the peer NTEs, individual calls are bound to locally designated timeslots. Circuit switching equipment within the ISDN arranges for the octets of the timeslot designated at one NTE to be exchanged with the octets of the corresponding timeslot at the peer NTE.

Using the 2 Mbps primary rate interface, each ramp can establish up to 30 concurrent ISDN calls to a variety of peer sites. Although each call can be used to support a separate 64 Kbps channel, calls between the same peer ramps can be aggregated together to form higher bandwidth exchange channels. The assignment of calls to each channel can be varied dynamically, and so the 64 Kbps ISDN call represents the basic unit of channel bandwidth.

8.3.1 Ramp Organization

Inmos Transputer chips are the processing elements used within the ISDN ramps. Each transputer is a single chip processor complete with on-chip memory and serial links to neighbouring processors and devices. Figure 8.3 is a block diagram of the ISDN ramp which consists of a CFR interface, an ISDN interface, and the mapper and framer transputers that make up the ramp halves. Transputer links are used

---

[1]The Bailey ramps were assembled out of existing components by C Adams of the RAL.

to interconnect the components, and the resultant pipelined organization facilitates the parallel operation of the principal ramp components.

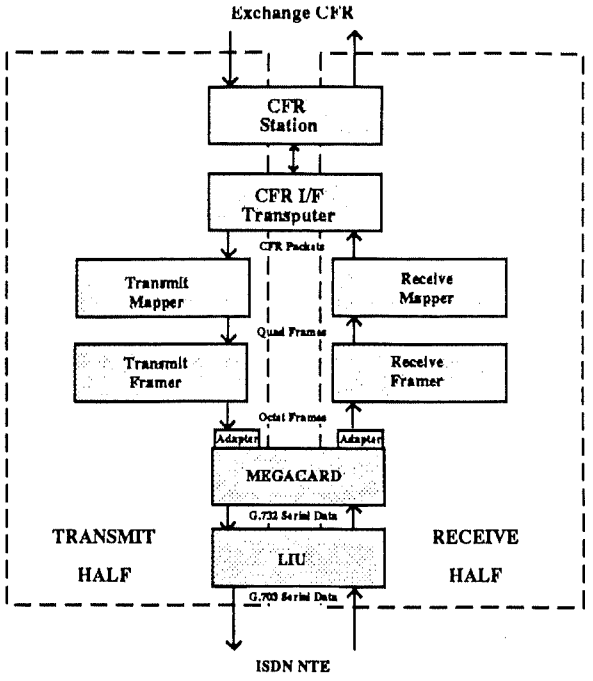

Figure 8.3: Transputer Ramp Organization

The CFR Interface

The CFR interface is based on a single transputer that has been provided with memory-mapped access to a bridge mode CFR node. This transputer performs CFR operations on behalf of both ramp halves. It is responsible for the configuration and monitoring of the node as well as the loading and unloading of CFR packets. Packets extracted from the local exchange are passed over a transputer link to the transmit half mapper, and packets destined for the exchange are acquired from the receive half mapper.

The ISDN Interface

Although the ISDN interface is based on the LIU and megacard modules used in the Bailey ramp, the style of operation is somewhat different. Transputer link adapters are used to connect each of the megacard's client busses to the appropriate framer processor. The megacard clocking is configured so that, regardless of which timeslots are selected, octets are exchanged during every data timeslot interval: the framers recognize the G.732 frame boundaries associated with TS0, and during each frame period they exchange a 30 octet sequence with the megacard. With the exception of TS0 and TS16, which are handled by the megacard, each of these sequences represents a complete G.732 frame.[1]

---

[1]Octets corresponding to unselected timeslots are ignored by the megacard and are not presented to the LIU. The megacard's TS16 interface provides access to the ISDN signalling function. This 64 Kbps stream is routed to an out-of-band signal service that exercises the ISDN signalling protocol. This signal service is controlled by the exchange channel service which is responsible for the configuration and management of the ramp.

Ramp Processors

The principal ramp processors are the mapper and framer transputers that are dedicated to each ramp half. During each 125 microsecond (usec) frame interval the framers must exchange a 30 octet frame with the megacard. The frame data is in turn exchanged with the mapper processors that perform most of the channel formatting functions. The transputers can complete most 32 bit (word) operations in the time required to complete the equivalent 8 bit (octet) operation, and so the mappers produce and consume *quad frames* consisting of 30 words each. One frame is exchanged with the framers during every 500 usec interval. The framers perform the appropriate mapping between the individual quad frames exchanged with the mappers, and the four corresponding octet frames exchanged with the megacard.

8.3.2 Channel Encoding

Segmentation and Splitting

Peer ramp halves implement segmentation and splitting functions that map the octets of each CFR packet onto the timeslots associated with ISDN calls. One approach is to encode each packet channel into an octet stream by segmenting packets as they are passed to the channel. The resultant octet stream is then split across the available timeslots: one packet is transmitted at a time and the individual octets of a packet may be transmitted in different timeslots. To recombine the channel the receive half requires some knowledge of the order in which timeslots arriving at the receiver were used at the transmitter. The number of 125 usec frame delays associated with the transmission of each packet decreases as the channel width, measured in timeslots, is increased.

An alternative approach is to perform the segmentation and splitting operations in the reverse order. The packet channel is first split into timeslot-specific subchannels, and after splitting the packets are segmented into independent octet streams. All of the octets of a packet are transmitted over the same timeslot and the higher bandwidth channel is realized by using multiple timeslots to support the concurrent transmission of different packets. The receive ramp half must ensure that the overall packet sequence is preserved when the subchannels are recombined. Using this scheme the number of frames delays associated with packet transmission is independent of the channel width. If 40 octets are transmitted for each packet, the 8 Khz frame rate will result in a fixed 5 msec frame delay.

The present ISDN ramp software implements the first of the above options for two reasons:

- Propagation delays within the UK are quite low, and so this option should permit the construction of high bandwidth channels with low overall delays; and
- the channel recombination problems associated with this option are more interesting and its solution may have other applications.[1]

Channel Recombination

The recombination of an octet encoded channel that has been split across a number of ISDN calls presents two interesting problems which, in this text, are referred to as timeslot reordering and frame skew.

When an ISDN call is established, timeslots at each of the peer NTEs are independently assigned for the duration of the call and different timeslots may be assigned at each interface. For example, an ISDN call may associate TS1 at NTE_a with TS4 at NTE_b and a second call may associate TS2 at NTE_a with TS3 at NTE_b. If the transmit half attached to NTE_a aggregates the two calls into a channel by transmitting successive channel octets in successive timeslots of a frame then the octets transmitted over the first call, during TS1, must precede the octets transmitted over the second call, during TS2. However, at NTE_b, the octets will arrive out of order, with the octets transmitted over the first call arriving in TS4, after the other octets which arrive in TS3. In effect, the octets of the channel are reordered as they pass through the circuit switching equipment embedded within the ISDN.

The delay through an ISDN is made up of propagation delays through the transmission components of the network, and frame delays which are incurred at circuit switches along the route. In a simple ISDN concurrent calls between the same pair of NTEs will follow the same route and will experience the same delay. Octets transmitted in different timeslots of a frame may be reordered but they will arrive at the peer NTE during the same G.732 frame. In a more complex network the calls may follow different routes involving different transmission paths and a different number of switches. In this case it is possible that the octets transmitted during a single frame will experience frame skew and will be presented to the peer NTE during different G.732 frames. The maximum frame skew (*MaxSkew*) is network dependent and is related to the maximum difference in propagation

[1]The recombination scheme is of particular interest when longer packets, such as IP datagrams, are being transmitted.

distances and switches traversed.

The splitting and recombination scheme used by the ISDN ramps permits a number of calls to be aggregated together without restriction or *a priori* knowledge of call routing.[1] Each transmit half maintains a *splitting map* that records the binding of timeslots to channels. Similarly, each receive half maintains a *recombination map* that binds incoming timeslots to channels. For each timeslot the map records the relative order and skew of the timeslot in relation to the other timeslots that make up the channel.

The transmit half places channel octets in successive timeslots of the channel so that the corresponding timeslots of frames arriving at the receive half contain the skewed and reordered octets of the channel. As each frame is received it is placed in a *frame buffer* which is capable of storing at least *MaxSkew* frames. The receive half uses its map to recombine the channel: it cycles through the channel timeslots in the order specified by the map entry, instead of the order of arrival, and extracts octets from the frame buffer. Successive octets may be extracted from different frames stored within the buffer and the frame accessed is determined by the timeslot skew recorded in the map.

In practice the operation of the ISDN ramp is slightly different in two respects: Each of the peer halves is concurrently supporting a number of channels to the same or different peer ramps; and the splitting and recombination functions are distributed between the mapper and framer transputers.[2]

Channel Synchronization

The transmit half embeds a sequence of synchronization symbols within the stream of channel octets. Each sequence contains distinguished symbols that identify the start of the sequence, and timeslot-specific symbols that are used to compute the

---

[1]The ISDN could provide direct support for multiple timeslot calls through the judicious choice of timeslots and routes. Alternatively the ISDN could compensate for skew and reordering by using internal routing information to adjust the frames presented at the peer NTEs. The first approach limits the flexibility of switch design and call routing whilst the second approach would complicate the establishment of calls that traverse concatenated ISDNs. The current ISDN recommendations make limited provision for multiple timeslot calls, and for the most part the issue is reserved for *future study*.

[2]The receive mapper incorporates a quad frame buffer, and its component of the recombination map accounts for quad frame skew and timeslot reordering. The framer incorporates an octet-based buffer, and its component of the recombination map accounts for octet frame skew. The framer ensures that, for each timeslot, the 32 bit words presented to the receive mapper are *aligned* with respect to the words generated by the peer transmit mapper.

relative order and skew of channel timeslots. The receive half uses the synchronization information to determine the corresponding entries within its recombination map.

The synchronization sequence is transmitted at fixed periodic intervals. Although the timeslot order will only change when the channel configuration is adjusted, uncontrolled *frame slips* within the ISDN can affect the skew of individual timeslots. Periodic synchronization allows the receiver to recover from these infrequent events.[1] Furthermore, timeslots can be added to or deleted from a channel at every synchronization point and there is no need to suspend the flow of packets when the channel bandwidth is varied.

Packet Synchronization

The packet synchronization scheme is similar to that used in the Bailey ramps. The transmit half inserts start and check symbols at the beginning of every packet, and idle symbols during periods between packets. The error detection function in the present ramps is somewhat limited and the check symbols are primarily used to validate channel and packet synchronization.

8.3.3 Transmit Half Operation

The CFR interface performs address translation and channel assignment on received packets. The transmit half implements expedited transfer, and so there are two sequenced queues assigned to each channel. Each queue can hold approximately 1000 packets allowing the ramp to absorb bursts of packets from peer CFR nodes.

The mapper removes packets from the channel-specific queues at the interface and delivers assembled quad frames to the framer. For each supported channel the mapper maintains a short *pending* queue of packets that have been extracted from one of the corresponding queues in the interface. The appropriate start and check symbols are added to each packet as it is admitted to the pending queue and packets in the *expedited* queue are always admitted in preference to packets in the *normal* queue.

---

[1]The check symbols associated with incoming packets are used to discard packets arriving during the interval between a slip and the next synchronization point.

Every 500 usec the mapper uses its splitting map to control the assembly of a quad frame. For each selected timeslot it removes a four octet word from the appropriate channel queue. If the timeslot is not associated with a channel or if there is no packet in the pending queue then idle symbols are inserted in the frame. The assembled frame is passed to the framer, which generates the octet frames that are presented to the megacard at 125 usec intervals. At fixed intervals, presently once every 100 frames, the mapper transmits a synchronization frame instead of a data frame. This action causes synchronization sequences to be concurrently transmitted on all of the active channels.

8.3.4 Receive Half Operation

Every 125 usec the framer accepts a frame from the megacard and places the frame in the octet frame buffer. For every four incoming frames, one quad frame is assembled from the buffer in accordance with the framer's recombination map. This frame is passed to the mapper where it is placed in the quad frame buffer.

The mapper performs recombination and reassembly concurrently. For each active channel, the mapper maintains a reassembly buffer of words associated with the current packet and a count of the number of words in the buffer. Between quad frame arrivals the mapper cycles through the entries in its recombination map extracting words from the frame buffer and appending them to the appropriate channel packet buffer. As words are processed the mapper performs the reassembly function by ignoring the idle symbols between packets, detecting start symbols, and updating the count associated with partially reassembled packets. Fully assembled packets are forwarded to the CFR interface where they are queued and eventually transmitted.[1]

The receive half may operate channels originating at a number of peer transmit halves and so it cannot expect all of the channel synchronization sequences to arrive during the same frame. For each channel, the mapper maintains a count of the number of words received since the last synchronization sequence. At fixed intervals the appropriate words in the frame buffer should contain the synchronization sequences inserted by the peer transmitter. The symbols are stripped from the channel and used to update the entries of the recombination

---

[1]The expedited transfer function is not supported in the present receive half software, and so all incoming channels receive equal priority at the CFR queue. The software is currently being modified to implement expedited transfer by subdividing the queue and sorting packets on a window-specific basis.

map. Should the synchronization pattern not be detected the receive mapper assumes that channel synchronization has been lost and the channel is suspended. The symbols within incoming timeslots are subsequently scanned until a suitable pattern is detected.

8.3.5 Summary

A number of individuals have collaborated on the step-wise refinement of the basic ISDN ramp description [Tennenhouse 85] into the present implementation. The earlier stages of problem definition were aided by I Leslie [Leslie 85] and a continuous synchronization scheme [Porter 85] was developed at the Computer Laboratory with the assistance of Prof D Wheeler, J Porter, I Leslie and A Hopper. J Burren was responsible for the consolidation of this effort and its generalization to suit a range of channel synchronization applications.[1]

The use of transputers in the ramp implementation has allowed virtually all of the ramp functions to be performed in software, and as such the ramp represents a valuable research tool for the analysis of alternative ramp implementations. The ISDN ramps have been used to perform a number of exchange experiments, and the results, including the evaluation of ramp throughput, delay, and jitter, are reported in Chapter 11 and Appendices D and E. It is hoped that, within Project Unison, the ramps will be used to further evaluate the effectiveness of expedited transfer and alternative approaches to splitting.

---

[1]The burden of detailed ramp design and the production of the working prototypes was borne by J Burren, D Clarke and their colleagues at the RAL.

Chapter 9
Exchange Portals

A portal is the point of attachment of a local, or client, network to a site's exchange. This chapter examines the network independent aspects of portal design with particular emphasis on the generic structure common to different types of portals. The network dependent issues are characterized through the description of specific portal designs that support the attachment of individual client networks.

9.1 Portal Organization

At an upper level of abstraction, a portal extends the service of the client network to support communication between peer devices attached to physically disjoint client networks. At a lower level, the portal supports the physical gateway between the client network and the exchange CFR. Figure 9.1 is a block diagram

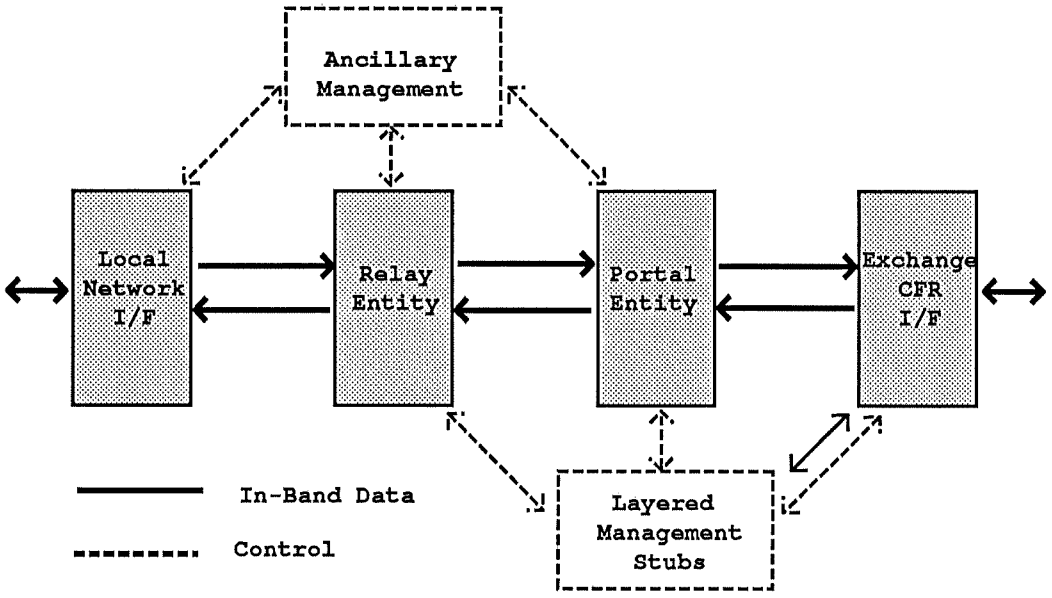

Figure 9.1: Generic Portal Organization

of a generic portal in which the local and exchange interfaces support the lower level gateway functions, whilst the relay, portal, and management components support the upper level functions.

Local Network Interface

The local network interface conforms to the standards and conventions governing access to the client network. The interface provides physical layer attachment to the local medium and implements the access scheme and encoding used to exchange relay entity messages with client devices attached to the medium. The design of the interface will vary with the client network, and the service provided to the relay entity will be dependent on the style of communication that is used by the client devices. For example, at an OSI-compatible LAN the interface provides a data link layer service that is tailored to bursty asynchronous traffic and based on the exchange of varying length service data units (SDUs). At other networks, suited to voice and video applications, the client traffic may be isochronous and the exchanged SDUs relatively short and of fixed length.

Relay Entities

The relay entity is divided into transmit and receive halves that rely on services provided by the portal entity to exchange data units with the relay halves at peer portals. The transmit half processes incoming messages from the local interface and performs client dependent operations to route each message to the appropriate peer relay entity. The peer receive half will subsequently use the data link service provided by its local network interface to relay the message to the peer client device. In order to support their activities, in particular the routing function, the peer relay entities may exchange additional information unrelated to the relaying of specific client messages.

Relay mechanisms vary greatly in terms of their complexity and the logical layers at which they operate. The implementation of low level relays that interconnect physically identical networks is likely to be streamlined, whilst relays operating at higher levels, though less efficient, can be used to interconnect diverse local networks.[1] Relay entities also cooperate with the management of their local networks and the complexity of this interaction will vary with the level at which relaying functions are performed.

---

[1]Although relay entities do not support protocol conversion functions, they may support communication between peer devices that support a common data link protocol or operate a subnetwork independent convergence protocol within their network layers.

For example, a telephony relay, attached to a digital PBX, may use out-of-band techniques to interact with the management of the local network. The routing of each telephone call is fixed when it is established, and the client SDUs, composed of PCM samples, are opaque. In contrast, the data link layer routing of datagrams between IEEE 802 networks is based on the examination of the 48 bit address fields located within each datagram. Although these SDUs are transparently passed between networks, fields within them are examined by the relays. Similarly, when relaying is based on the operation of a common Internet Protocol (IP) over the peer networks, routing can be based on address fields defined at the network layer. In this case the datagrams may be subject to both examination and modification as they are passed between client devices.

Portal Entities

Associations between portal entities support the exchange of messages between peer relay entities. When the relay presents a message for transmission, the portal entity assigns the message to an appropriate peer portal association and encodes the message for transfer over the association. The peer portal entity will regenerate the message and present it to the receive half of the peer relay.

The complexity of the association assignment process will vary amongst portal implementations. A simple scheme is to operate a single association for each peer portal, so that all of the traffic between a given pair of local networks is assigned to the same association. New associations are established as required and dormant associations are expunged. This scheme can easily be extended to support expedited transfer by operating separate *normal* and *expedited* associations. An alternative proposal is to use separate exchange associations to support discrete occasions of communication between peer client devices. This requires a more sophisticated assignment process in which the relay entity must relate individual SDUs to specific client associations. The suitability of this scheme depends on the frequency and duration of client associations: It is suited to telephony applications but not appropriate to datagram services.

The encoding of transmitted messages is dependent on the length and variability of the relayed SDUs. Messages that exceed the maximum length supported by the exchange interface must be segmented and reassembled by the peer portal entities. Similarly, relay messages that are short and of fixed length, i.e., single octets, should be blocked and deblocked.[1]

---

[1]Portals attached to PBX networks are expected to support the blocking function. There is no requirement for segmentation or blocking at the other portals described in this chapter.

Portal Management

Management functions related to the client network are embedded within the relay and local interface components. Management functions related to exchange operation are performed by out-of-band management stubs and ancillary management components.

The stubs perform local management functions in conjunction with the layered management services of the exchange. In particular, the portal entities rely on the secretary service to establish and maintain their peer associations. The secretary stub within each portal interacts with the secretary service to negotiate association-specific parameters, and configures the local exchange interface in accordance with the negotiated values.

Within the exchange domain, the support of security and accounting services governing peer site communication is fundamental to the preservation of site independence. Although further work is required in this area, support for these functions should be provided by efficient in-band filters that are configured by ancillary management services.

The local networks attached to an exchange may fall within different administrative domains, and in these cases the portals straddle administrative as well as physical boundaries. A *chinese wall* can be drawn between exchange components, so that, logically, the exchange interface and portal entity fall within the domain of the exchange operator whilst the local interface and relay entity fall within the domain of the local network operator.[1] The portal entities operate in conjunction with the ancillary management services to enforce the exchange operator's security, accounting, and monitoring policies. The relay entity is expected to provide similar functions on behalf of the local network's administration.

Exchange Interface

The exchange interface implements the lower layers of the exchange protocol suite by providing the UDL service described in Appendix F. The interface multiplexes CFR transmissions and receptions arising from the concurrent associations operated

---

[1]In the extreme case the portal components could be physically separated into distinct devices that communicate over a private interface.

on behalf of the portal and management components.[1] The implementation of multiplexing within the UDL service ensures that, above the interface level, the traffic arising from management associations does not interfere with the *in-band* traffic arising from portal entity associations.

9.2 Universe Portal

The Universe portal relies on the exchange site interconnection service to link Universe subnetworks. The portal provides an alternative to the Universe satellite bridge [Waters 84] by substituting the common carrier oriented exchange service for the private satellite channel used in Universe. From the perspective of the Universe LANs the portal appears to be a Universe bridge conforming to the specification in [Waters 82].

Portals can be used in conjunction with, or as alternatives to, the ring-ring bridges [Leslie 83] that link the local networks of a Universe site. In the Universe experiments each site had a single point of attachment to the satellite service: the satellite bridge was attached to one of the site's local networks and all off-site traffic was routed through that subnetwork. In the exchange environment any number of the site's LANs can be directly attached, via portals, to the exchange CFR. The portals support local communication amongst the attached LANs and provide them with direct access to peer local networks attached to remote exchanges.

9.2.1 Portal Organization

The portal hardware organization is based on a VME backplane bus that links: two processor cards; a shared memory module; an interprocessor *mailbox* device; and a CFR interface. One of the processors is dedicated to the support of the Universe basic block protocol and is directly attached to the local network. The other processor supports the Tripos operating system kernel [Richards 79]. Tasks running within this processor perform the relay and portal entity functions and use Tripos device drivers to access the CFR interface and local network processor. Figure 9.2 illustrates the internal organization of the portal software.

---

[1]The interface does not directly participate in association management as this function is performed, at an upper layer, by the secretary stub.

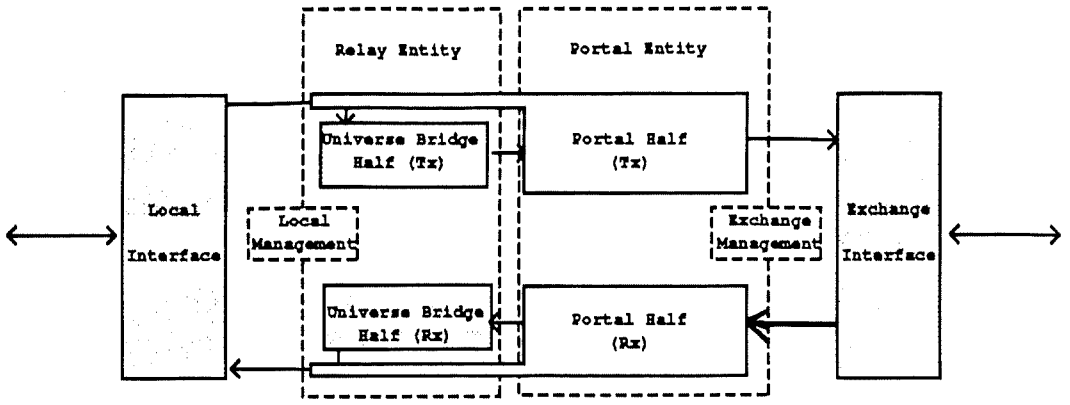

Figure 9.2: Universe Portal Organization

Local Network Interface

The present LAN interface, which attaches the portal to a Cambridge Ring, supports a basic block service that is similar to the SPECTRUM system described in [Garnett 83]. Software running within the portal processor uses shared VME memory and the mailbox device to exchange commands, responses, and data with the interface processor.

Exchange Interface

The exchange interface is provided by a Tripos device driver, running within the portal processor. This software implements the UDL service and is responsible for the loading and unloading of CFR packets. The portal processor is attached to a station mode CFR node through the memory mapped VME interface that is used in the Bailey ramps.

The UDL software supports the transmission and reception of blocks related to specific associations. When a block is to be transmitted on an association, the driver accesses an association-specific table entry to determine the port field value and other information to be placed in the header fields of outgoing packets. When a packet arrives at the interface, the UDL port field within the header is used to demultiplex concurrent associations. The port value is mapped into a table entry that contains association-specific reassembly information. The device driver accesses the association table entries but is not responsible for their creation or for the assignment of port values to specific associations.

Relay Entities

The relay entities are based on the bridge software [Leslie 83] that normally links adjacent Universe LANs. In the bridge application, the software is divided into two directional halves, each of which receives the basic blocks arriving on one of the subnetworks, performs the appropriate bridge processing, and transmits the blocks on the other subnetwork. In the portal application only one of the subnetworks is a physical LAN. The other *logical* subnetwork is composed of all of the peer Universe portals that are accessible over the exchange. The bridge software, operating within peer portals, uses this logical subnetwork to exchange blocks in the same way that a physical LAN is used to relay traffic between peer Universe bridges.[1] A similar scheme was used in the implementation of the Universe Tunnel [Tennenhouse 84] which substituted a public X.25 service for the Universe satellite channel.

The principal relay components are the two bridge half tasks. These tasks rely on the portal entities to support communication over the logical portal subnetwork. They perform port mapping and *open* block processing on the basic blocks exchanged between portals. The transmit bridge half receives blocks from the local LAN and presents them to its portal entity for transmission over the exchange. The receive half accepts blocks from its portal entity and presents them to the local interface for transmission over the attached LAN.

In practice, the portal organization is somewhat complicated by the desire to maintain compatibility with the bridge software. In the original application, each bridge half accepts incoming blocks from a Tripos device driver and returns processed blocks to the same driver. The device interface is configured to receive blocks on one LAN and transmit on the other. In the portal application this message passing sequence has been preserved by arranging for the portal tasks to emulate the bridge device drivers. Each portal task includes a thin *veneer* that shuffles blocks between its bridge half and the network interfaces.

The relay software includes additional tasks and components that interact with the management of the local network and configure the bridge halves to:

---

[1]In the original application, the bridge half functions are symmetric and both halves execute the same program. Although a common program is also executed by the bridge halves of the portal, the functions exercised by each half are not identical. For example, the transmit half may receive Universe OPEN blocks originating from clients attached to the local network. The receive half accepts blocks from the portal subnetwork and, since only peer bridges are *attached* to this subnetwork, OPEN blocks will not be processed.

- Support Universe path setup and path tracing;
- Support the allocation and reclamation of subnetwork ports;
- Determine the configuration of the rings and bridges accessible via the local network; and
- Supply bridge status information.

Portal Entities

Peer portal entities use UDL associations to support the transfer of relay entity blocks. The portal entity is divided into transmit and receive halves and each half is represented by a Tripos task that logically encapsulates the bridge task of the corresponding half of the relay entity.

Exchange Management

The dynamic creation of associations is not supported by the present portal implementation. The Universe site topology is fixed and associations are statically created when the portals are initialized. Provided the destination identifiers supplied by the relay entity are valid, the portal table lookup will always be successful and will yield the pre-determined identifier of the appropriate peer portal association.

In the future the portal may be upgraded to use the exchange secretary service to support the dynamic creation of peer portal associations. The portal table will be initialized with null entries and whenever a lookup operation fails a request for a new association will be forwarded to the portal's secretary stub. This task will resolve the Universe site and subnetwork identifiers into the title of an appropriate peer portal service. The stub will then perform a remote procedure call (RPC) to the secretary service in order to initiate an association with the peer portal. When the association has been established the stub will configure the UDL driver to support the association and add the corresponding entry to the portal table. The secretary stub will also processes incoming RPCs from the secretary service, notifying it of remotely initiated requests for new associations.

9.2.2 Transmit Half Operation

Basic blocks are received by the local interface and are presented to the transmit relay half by the veneer portion of the transmit portal task. The blocks will have been error checked by the driver and are known to have arrived on one of the ports reserved for peer bridge and management interactions, or on a port

previously allocated by the bridge software.

The port field value and other information contained within the block header are used to perform the appropriate Universe bridge operations. This will usually result in modifications to the block header and the determination of the Universe site and subnetwork values that identify the peer bridge to which the block should be forwarded. The modified block is returned to the portal task for onward transmission over the exchange.

The portal entity inserts header information at the front of each block,[1] and presents the message to the UDL driver for transmission over the outgoing peer portal association. As each block is processed, a portal table is used to map the peer site and subnetwork identifiers, supplied by the relay half, into the appropriate peer portal association identifier.

In keeping with Tripos conventions, the messages passed between the device drivers, portal tasks, and relay tasks contain pointers to data blocks that are stored in the shared VME memory. Once the blocks have been received they are not copied until they are transmitted on the other network.

9.2.3 Receive Half Operation

The operation of the receive half of the portal is somewhat simpler. As each message is received on the exchange, the portal entity strips the header and passes the body of the block to the relay entity. The relay task performs the appropriate bridge processing, further modifying the block forwarded to it by the source relay entity. The transformed block and local network address are returned to the veneer of the portal task for onward transmission over the local network.

9.2.4 Portal Performance

The peer portal transmission bandwidth is limited by the inter-site channel capacity supported by the ramps or by the bandwidth available at the Cambridge Ring interface. In experiments performed using the pilot exchange implementation the measured unidirectional throughput between peer client devices was 565 Kbps.

---

[1]Space for the header is allocated by the *veneer* portion of the portal half when the block is received at the local interface.

This compares favourably with the 655 Kbps maximum transmission bandwidth, *MaxTxBw*, available between peer nodes attached to the client LANs.[1] For sufficiently wide channels the throughput between sites is ultimately limited by the speed and configuration of the client slotted rings: there was no observed difference in the throughput between peer LANs at the same site and peer LANs at different sites.[2]

In the presence of contention traffic, the portals will induce a significant degree of jitter into client block streams. The principal sources of jitter are the lack of multiplexing within the Cambridge ring basic block layer and the relatively large limit on the size of individual blocks. In the absence of other traffic, the delay through the portals is dependent on the length of the blocks being processed. For relatively short blocks, the delay is dominated by the processing delays incurred within the exchange ramps and portals. In an experiment involving the exchange of short messages between peer devices attached to different LANs within a site, the observed round trip delay was 13.3 msec. When the experiment was repeated between devices at different sites the round trip time was extended by 12.1 msec.[3]

For longer blocks, the delay is dominated by the store and forward processing of Cambridge Ring blocks. Each basic block is completely received at the source portal before transmission to the destination portal begins. The destination portal waits until the corresponding UDL block has been completely received before it begins transmitting the basic block on the destination LAN. These delays could be reduced through the use of *cut through* switching, allowing the source portal to process the basic block header as soon as the first mini-packets of a block are received. Subsequent mini-packets are blocked into UDL segments and transmitted over the exchange as they are received. At the destination portal the segments are queued for onward transmission on a segment by segment basis. If the source portal detects an error in the incoming block it signals this condition to the destination portal which aborts the LAN transmission. In the absence of contending traffic, the cut through approach can significantly reduce the store and forward delay through the portals. However, it requires a much greater degree of

---

[1] *MaxTxBw* is derived from equation A.5 of Appendix A. For the longer of the two rings used in the experimental programme: N_s = 3 slots; P_s = 38 bits; P_d = 16 bits; G = 8 bits; F_r = 10 Mbps; and D = 2 slots.

[2] In fact, the throughput remains stable even when the traffic is routed through intermediate exchanges supporting the SI-layer relaying function.

[3] In Chapter 11 this delay will be shown to be primarily attributable to the ramps themselves rather than the carrier transmission facility. A similar delay is incurred when the messages are routed through an intermediate relay site.

interaction between the upper and lower layer portal components and this additional complexity may reduce the overall portal throughput.

9.3 Other Portals

9.3.1 Internet Portals

Internet portals link disjoint clusters of end users that communicate in accordance with a common Internet architecture such as the ARPA protocol suite. The portal relay entities operate within the client network layer by arranging for IP datagram segments to be transported between peer portals. Each of these client segments is transported as a single UDL block, which is in turn segmented into exchange packets. These portals will support many existing client services such as file transfer, electronic mail, and terminal access. Given the limitations of the Internet approach described in Chapter 3, the performance characteristics of these services will be somewhat limited.

In the case of an ARPA Internet, the portal relay elements can be based on available IP gateway software. The addresses contained within the datagram segments arriving at the relay entity are hierarchical and each segment's destination address can be decomposed to yield a unique destination network identifier. This identifier is resolved into the identifier of the IP network, serviced by an exchange portal, that represents the next hop to the destination network. The datagram segment and its next hop identifier is supplied to the portal entity for transmission to the peer portal.

The portal entities provide little additional functionality beyond the transfer of datagram segments between peer relay entities. The transmit portal entity maps the IP network identifier into the appropriate peer portal association identifier, and the segment is presented to the exchange interface for transmission within a single UDL block. The dynamic creation of peer portal associations is supported by the secretary stub, which must resolve IP network identifiers into the titles of exchange portal services.

9.3.2 IEEE 802 Portals

Although Internet portals can be used to support the network layer interconnection of IEEE 802 LANs, it is also possible to build lower level portals that operate within the data link layer. These portals provide enhanced performance and their operation is independent of the network layer protocols exercised by the clients.

The relaying algorithm is derived from the scheme described in [Hawe 84]. The local interface operates in *promiscuous* mode so that the relay entity can observe the 48 bit source addresses of all MAC[1] layer data units transmitted on their local network. Each observed address is recorded in a *forwarding database* that identifies the local clients accessible via the portal.

In Hawe's scheme, a single relaying entity is locally attached to a number of networks, and the forwarding database contains the aggregate addressing information derived from all of the networks. When a datagram is received at a network interface its destination address is extracted and the database is accessed to determine whether or not the block should be transmitted on one of the other network interfaces. If the address is not recorded in the database then the block is transmitted on all of the attached networks. The scheme includes mechanisms for dealing with group addresses and duplicate transmissions.

In the case of a portal, the relaying entities are geographically disjoint. Associated peer portals must exchange forwarding information so that each portal builds up a database of the addresses accessible through a current or recent associate. As datagrams are received from the local network the database is used to determine whether the destination address is accessible through the local network, in which case the block is ignored, or whether the destination address can be mapped to a peer portal identifier. In the latter case, the datagram and the identifier are forwarded to the portal entity. The portal entity uses a UDL association to transport the message to the appropriate portal where the datagram is injected into the peer client network.

If the destination address is not recorded in the forwarding database then the relay entity dispatches an address resolution request to an out-of-band management service. This service may access a global directory that records bindings between

---

[1]Media Access and Control (MAC) is a sublayer within the IEEE 802 data link layer.

IEEE 802 addresses and client network identifiers.[1] The eventual result of the address resolution process is the establishment of an association to another portal and the exchange of forwarding database information.[2] A portal's forwarding database represents its local cache of directory information. It is expected that a large fraction of each portal's traffic will be directed to a small group of peer portals, and so this cache will considerably reduce the requirement for directory access.

9.3.3 PBX Portals

Two different styles of digital PBX portals have been identified: isochronous portals that exchange PCM samples at the standard 125 microsecond sample rate; and ATM gateway portals that exchange blocks of PCM samples contained within ATM packets.

Isochronous Portals

Isochronous portals link peer PBXs by using an exchange association to emulate a G.703-style fixed transmission service. During every PCM sample interval, the portal and its PBX exchange one 32 octet G.703 frame. Each frame is embedded within a single CFR packet and transmitted to the appropriate peer portal where its is presented to the peer PBX. The portals incorporate elastic buffering in order to smooth the flow of frames over the exchange and maintain the isochronous nature of the service.

The isochronous portal provides a PBX with access to the shared common carrier facilities attached to the site's exchange and replaces fixed links between peer PBXs with dynamic switchable links. The advantage of the isochronous approach is that it requires very little relay processing on the part of the portals. The peer PBXs use the G.732 frame structure to multiplex concurrent 64 Kbps telephone calls onto the association. The ISDN signalling protocols can be exercised over timeslot 16 to support the exchange of signalling information between peer PBXs.

The disadvantage of the isochronous approach is that each peer portal association statically consumes enough bandwidth to support thirty 64 Kbps timeslots

---

[1]Or some equivalent form of site identification such as a list of NTE addresses.

[2]There is clearly a requirement for an algorithm that effects the efficient exchange and updating of databases by a group of co-operating portals.

regardless of the number of timeslots that are actively in use. This arrangement may be acceptable in broadband environments, where bandwidth is plentiful, or when the individual peer portal associations are heavily utilized. In other cases, such as the Unison pilot network, the available inter-site bandwidth is limited and should be allocated on a more dynamic basis.

ATM Gateways

When an ATM gateway is employed, sequential octets arising from each 64 Kbps telephone call are collected into larger blocks that are transmitted within CFR packets. In effect, the gateway converts each isochronous stream into an ATM stream that is compatible with the basic transfer service of the exchange. Bandwidth is only consumed by active associations, and the peer portals can further reduce their overall bandwidth requirement through the use of compression and silence suppression techniques. The relaying performed by an ATM gateway portal is relatively complex and has both in-band and out-of-band components.

On an in-band basis, the portal maintains a separate peer portal association on behalf of each of the active 64 Kbps streams. During each sample interval, the transmit half of the portal demultiplexes the samples arriving from the local PBX[1] and places each sample in an association-specific buffer. When a buffer is filled, its contents are transmitted, over the appropriate peer portal association, in the data field of a CFR packet. The receive half of the portal maintains association-specific elastic buffers which are filled by incoming packets and emptied as the individual samples are passed to the local PBX during each PCM interval.

On an out-of-band basis, the portal relaying entity emulates a switching node terminating the ISDN signalling protocol exercised over timeslot 16 (TS16) of the PBX interface. For example, the PBX can utilize TS16 signalling to request the establishment of a new 64 Kbps call, and to negotiate the timeslot in which the call's samples will be exchanged between the portal and the PBX. The portal uses the exchange secretary service to establish an association with the appropriate peer portal, which in turn exercises TS16 signalling to notify the destination PBX of the incoming call.

One advantage of the gateway approach is that it exploits the existing SI-service to support switching and bandwidth allocation on a per-call basis. A further advantage is that the gateway explicitly converts the isochronous PBX streams into

---

[1]At the physical level, the portal uses the G.703/G.732 interface to exchange frames with its local PBX.

the ATM format used throughout the exchange architecture. The gateway represents an evolutionary step towards the deployment of new PBXs based on *native* ATM telephony.[1]

9.3.4 CFR Portals

The Cambridge Fast Ring, which is the basis of the local exchange switch fabric, is also used as a multi-service local area network. Within Project Unison, CFR-based LANs represent an important testbed for the development of new telecommunication services based on the ATM transfer of digitized voice, images, and video. Although the communication rate and transmission media of future local networks will be dependent on the deployment of new technologies, it is expected that many of these networks will provide an ATM service similar to that provided by the CFR. The viability of the exchange architecture is dependent on the development of portals that support the interconnection of ATM-based client networks such as the CFR.

At the physical layer, it is fairly simple to construct a portal that functions as a CFR bridge, by copying complete packets between the client and exchange CFRs. The difficulty arises in arranging for the portal to be multiplexed so that it can concurrently support associations between different devices attached to the client CFRs. Since the client devices support a range of MSN applications, it is important that the delay and jitter experienced at the portals be constrained. Accordingly, portal relay entities should avoid the use of blocking and segmentation functions.[2] Since there is an exact match between the maximum payloads of the client and exchange SDUs it is not possible to embed additional multiplexing fields, such as the UDL header, within the packets transferred over the exchange. Two solutions to this problem have been identified and both require a considerable degree of cooperation between the management of the exchange and the client CFR.

The relay entities can operate at the CFR packet level by examining and mapping the address fields within individual packets. In this scheme, each portal is allocated a group of exchange addresses, which are in turn assigned to specific

---

[1]The design of an ATM-based PBX is described in [Want 88].

[2]The previously described LAN portals perform relaying at the client data link or network layer, and rely on the exchange UDL layer to segment the relatively large SDUs of the client network into the CFR packets that are transferred over the exchange.

associations between clients. In effect, the portal resembles an exchange ramp that uses dynamic address assignment to extend the SI-service into the client networks. The advantage of this approach is that the data fields of CFR packets pass through the exchange without modification and there is no limitation on the upper layer protocols exercised by the client devices.

Alternatively, the relay entities can operate at the UDL level by examining and mapping the port field values located within the UDL headers of individual packets.[1] The port mapping performed by the portal is similar to the mapping performed in the Universe bridges, and, after processing, each packet is forwarded to the portal or client station that represents the next hop along an association-specific route. The association establishment procedure is dependent on RPC interactions involving the exchange secretary service, the secretary stubs within the portals, and the individual secretary services of the client LANs. A suitable scheme is described in [Adams 87].

9.4 Summary

This chapter has discussed the generic features common to many portal designs and has described a number of specific portal implementations.

The Universe portal has been developed, as part of this research, to demonstrate the operation of the exchange architecture and to provide a development environment for a variety of multi-service network applications. An electronic office demonstration, developed at the LUT, used this portal to support the transfer of voice, video and data traffic between offices located at different sites. The portal components have also been used within the experimental programme described in Appendix C. In this configuration, the relay and portal tasks are replaced with artificial traffic sources and sinks. The resultant *synthetic* portals have been used to emulate the load and performance of portals supporting various types of exchange traffic.

---

[1]This approach to portal construction is dependent on the use of the UDL protocol and a local secretary service within the client network. This is not considered to be a significant disadvantage since UDL has been universally adopted by the CFR user community.

Although an IP portal has yet to be assembled and tested, the relaying of IP segments has been demonstrated.[1] The IP software, operating on SUN workstations, uses the CFR to relay datagrams on behalf of client systems attached to Ethernets. Similarly, the relaying of IEEE MAC data units has been demonstrated using Olivetti M28 processors. This design is now being extended to use transputers to perform in-band aspects of the relaying function. For both the IP and IEEE 802 portals the principal area of difficulty, yet to be resolved, is the distribution of directory information that supports the resolution of client addresses into next-hop and peer portal identifiers.

In collaboration with Project Unison, Olivetti Research (Cambridge) is building an ATM-based PBX gateway. The in-band frame assembly operations are the inverse of those performed by the ISDN ramp described in Chapter 8, and so the gateway is based on the existing ramp hardware and custom frame processing software. Although its primary application will be the support of ATM communication between PBX-based telephones and CFR-based telephones[2], the gateway can also be used as an exchange portal linking peer PBXs.

Finally, a UDL level CFR portal has been developed at the RAL and replicated at the various Unison sites. This portal has been constructed by interconnecting two of the CFR interface transputer boards that make up the ISDN ramp. As part of this effort, the exchange secretary service, described in Chapter 10, has been extended to support the association establishment functions.

---

[1]Joe Dixon of the Computer Laboratory developed the IP software on behalf of Olivetti Research. Brian Robertson of Olivetti Research is responsible for the development of the IEEE 802 portal.

[2]Developed by Roy Want of Olivetti.

Chapter 10
Exchange Management

The principal exchange function is the transfer of CFR slots between *associated* client portals. Additional functions, such as the establishment of associations, are provided on an out-of-band basis by exchange management services.

This chapter describes the *layered* management services that support exchange site interconnection:

- The *secretary* service supports association establishment;
- The *window* service provides address space and bandwidth management; and
- The *channel* service is responsible for ramp configuration and monitoring.

Each service is supported by a collection of cooperating entities resident within processing systems attached to exchange CFRs. These entities are themselves dependent on the packet transfer function to support associations with clients, peer entities, and other management services.[1]

10.1 The Secretary Service

Application entities[2] access the service through secretary *stub* entities resident within each exchange system. Each site has a unique secretary entity that communicates with its local secretary stubs and with peer secretaries operating at remote sites.

---

[1]Associations between management entities are structured in accordance with the exchange portocol suite described in Appendix F. Transactions involving management entities are supported at Remote Procedure Call interfaces.

[2]For the purposes of this discussion, the exchange management entities and the portal entities described in Chapter 9 are collectively referred to as application entities.

Service Description

Association establishment is accomplished through the nesting of Remote Procedure Calls. An initial parameter set, originating at a source application, is embedded within a chain of RPCs until it is delivered to a destination application.[1] The application's response is passed back along the links of the chain until it is returned to the initiator. The parameter sets may be modified as they are passed along the chain, with individual parameters added, removed, and transformed as appropriate.

The process is initiated when an application invokes its system secretary stub and supplies it with:

- A destination application title;
- Its own application title; and
- Application specific parameters.

The stub will first attempt to resolve the destination title within the context of its own system, and if successful, it establishes the intra-system association without reference to the site secretary. Otherwise, the stub allocates a UDL port to support the proposed association and embeds the port identifier, the system's CFR address, and the initiator's parameter set, within a nested RPC to the local secretary.[2]

The secretary attempts to resolve the destination application title within the context of the local site. If the application is supported on a local system, the title will be bound to a local system identifier.[3] In this case, the secretary acts as a message switching service that forwards the request to the secretary stub of the target system.

---

[1]The proposed association may be rejected or redirected at any stage during construction of the RPC chain. On rejecting an association an entity may transform the parameter set in order to redirect the chain to an alternative application, system, or site as appropriate.

[2]The local secretary maintains an association with the secretary stub resident within each local system. This association provides an out-of-band path, that in effect, supports communication between peer secretary stubs. The secretary association is itself established through the use of a primitive association that is bound into the individual stubs.

[3]The stubs may use their secretary associations to dynamically advertise the application titles supported within their own systems.

The recipient secretary stub allocates a local UDL port to support the association and matches the destination application title with a resident application entity.[1] The target application is notified of the proposed association and is supplied with a local association identifier[2], generated by the stub, together with the initiator's parameter set.

The unravelling of the RPC chain begins when the recipient responds to its secretary stub. Application specific information and a responding application title may be included within the returned parameter set. The stub conveys the application's response, the local CFR address, and the UDL port identifier to the site secretary. The secretary passes this information back to the initiating stub which assigns a local identifier to the association. The process is finally completed when the association identifier and the respondent's parameter set are returned to the initiating entity.

When the local secretary determines that a destination title is not supported within the local site, it attempts to resolve the application title within a global context.[3] If the application is supported at a remote site, the title will be bound to a remote site identifier and the association request is passed to the peer secretary responsible for the remote site.[4]

The remote secretary resolves the application title into a remote system identifier and forwards the request to the system secretary stub. The response, which is threaded back along the RPC chain, passes through both secretaries *en route* to the initiator.[5] As part of the establishment procedure, each secretary consults its local window entity, which is responsible for the binding of associations to exchange

---

[1]The process supporting the application entity may be dynamically created as a consequence of the incoming RPC.

[2]The stub configures the UDL driver so that the local association identifier is bound to the peer UDL ports and CFR addresses.

[3]The secretary may invoke public directory services, such as those described in [Lampson 87] and [ISO 9594-1], to assist in the resolution of application titles. Directory entries may include application attributes, such as QOS requirements. These attributes may be passed to the window service and/or added to the RPC parameter set exchanged by the peer secretaries.

[4]If necessary, an association between the secretaries of the peer sites is established as a side effect of this action.

[5]In general, the RPC chain may span any number of secretary entities. This feature is exercised by the CFR portal [Adams 87] which attaches CFR-based local area networks to the local exchange. An exchange secretary operating within the SI-layer can interact with local secretaries operating within client networks.

address windows. The returned window value is used to transform the CFR addresses embedded within the RPC parameter sets.[1]

Discussion

In many respects the present design is an extension of the out-of-band name resolution scheme introduced in Universe [Leslie 84].[2] In the secretary service the nameserver transaction has been extended into an end-to-end exchange involving the peer applications.

The service supports the active resolution of application titles into local association identifiers. Furthermore, the secretary associations provide an out-of-band medium for the exchange of:

- Application-specific parameters between peer application entities;
- UDL port identifiers and CFR addresses between the secretary stubs at peer systems; and
- Address window values between peer secretaries, and, indirectly, between peer window service entities.

From a performance perspective, the secretary service facilitates the streamlined implementation of exchange ramps and of the UDL software embedded within portals. The present design has additional architectural advantages including the initial exchange of parameters between applications, and the deferred binding of application titles into specific sites, windows, systems, and entities.

The initial parameters represent a medium for the negotiation of various attributes including the association quality of service (QOS) and the selection of upper layer protocols. In the corresponding OSI procedures the transport connection is established, and many session and presentation layer parameters are determined, before the destination application can be consulted.[3]

---

[1]Whenever a CFR address crosses a site boundary, its window field values must be transformed in accordance with the address assignment procedures described in Appendix B. The interaction with the window service may have a number of side effects including the creation of new address windows, the allocation of channel bandwidth, and the construction of new channels.

[2]The deferred binding of remote application titles through peer secretary interaction is an extension of the remote lookup function supported by the nameserver. The active configuration of ramps triggered by the window service interaction corresponds to the Universe bridge setup function.

[3]The bottom-up establishment process also leads to the early binding of end system addressing information. During the specification of the OSI Naming and Addressing architecture [ISO 7498-3] a concerted attempt was made to limit the impact of early binding. Nonetheless, each application title entry within the OSI directory includes the values of the transport, session, and presentation layer

The deferral of association-specific bindings complements a *need to know* approach to the distribution of naming information. The overall design reinforces the administrative independence of the individual sites and their client networks:

- Secretaries bind remote application titles to sites (represented by peer secretaries) but not to individual systems. Sites do not export bindings between applications and specific systems;
- Secretaries bind local application titles to systems (represented by secretary stubs) but not to individual application entities. Systems do not export information concerning the internal organization of their upper layers. The CFR addresses and UDL ports exchanged by peer secretary stubs are dynamically determined and association-specific;[1]
- Secretary stubs bind application titles to specific application entities but do not restrict the choice of upper layer protocols exercised by associated applications; and consequently
- Applications exchange titles without reference to the physical location of their peers. The association identifiers provided to each entity are strictly local and association-specific.

Although experience with the present secretary service[2] is somewhat limited, the streamlined designs of the in-band exchange components have confirmed some of the anticipated benefits. Other aspects of the secretary design have already been proven by its predecessor, the Universe nameserver. These include the use of deferred bindings, and the overall reliability and capacity of a dynamic name resolution service. One of the major secretary limitations, the round trip delay incurred during association establishment, is not a significant obstacle to the experimental applications, which include file access, telephony and video conferencing.

---

selectors used to access the application entity. In effect, these selectors allow the initiating system to navigate its way through the upper layers of the recipient.

[1]The secretary entity does not retain state information concerning individual application associations. The secretary stubs reclaim UDL ports and association identifiers when associations are released or found to be inactive. The only fixed bindings required by the secretary service are the well-known CFR address and UDL port used by the primitive secretary association.

[2]The secretary service [Tennenhouse 86a] is a component of the CFR communications architecture that is present within LAN and exchange environments. Ian Wilson of Olivetti Research Ltd developed the secretary and correspondent stubs presently used within CFR-based LANs. Further work at the RAL has extended the secretary software to support the peer RPCs that arise within the exchange environment.

10.2 The Window Service

The window service occupies an intermediate position in the management hierarchy. Above the window layer, associations between peer applications are bound to address windows. These windows represent end-to-end addressing paths between peer sites, and associations terminating at the same sites may be bound to a common window. Below the window layer, windows are bound to channels that correspond to transmission facilities linking pairs of exchange ramps. A number of windows can be bound to a single channel, and windows passing through relay sites can be supported through the concatenation of channels.

Each site has a unique window entity that supports the window service function and effects the local bindings between associations, windows, and channels.[1] This entity interacts with the local secretary, the channel service, and peer window entities operating at remote sites. It dynamically allocates address windows, and, in conjunction with the channel service, it manages the available common carrier bandwidth.[2]

Service Description

Requests for window values are presented as RPCs that specify the peer site identifier and association-specific attributes. If the new association can be supported on an existing window, the corresponding window value is returned to the secretary. Otherwise, the window entity establishes a new window by:

- Allocating a window value from the local address space;
- Negotiating window creation and an exchange of source and destination window values with the peer window entity at the specified site; and
- Selecting a channel to support the window, and interacting with the channel service to complete the binding.

---

[1]Although a window assignment is fixed for the lifetime of an association, the physical routing of associations can be altered through adjustments to the bindings between windows and channels.

[2]The distinction between window and channel service arises because the ramps of an exchange may be attached to different common carrier networks. Distinct entities within the channel layer cope with the different styles of carrier service and present a relatively uniform channel model to the upper layer.

The channel selection process effects a routing decision in terms of the ramps, carrier networks, and relay sites traversed by the window.[1] If an appropriate channel is not available, the window service must construct a new channel or arrange for the window traffic to be relayed at intermediate sites.[2]

To construct a new channel, the window entity selects the peer ramps that will support the channel[3], and invokes the channel service to perform the carrier dependent signalling functions associated with channel creation. Once the channel has been created, each of the peer window entities binds an initial window to the channel, and an association is established between them.[4] This association can be used to negotiate adjustments to the channel, the construction of new channels, and the creation of additional windows.

In addition to responding to secretary and peer requests, the window entity performs background functions related to the monitoring and management of exchange resources. It maintains a shared management database that is accessed and updated by the channel service. Although this database records the current status of exchange windows, channels, and ramps, its does not retain state information concerning individual associations or their window bindings. Windows are automatically deleted when they are found to be idle as a result of channel failure or the cessation of association traffic.[5]

Discussion

The quality of service provided to an individual association depends on the service available at its assigned window. This service is in turn dependent on an appropriate allocation of resources to the supporting channel. In managing the

---

[1]The choice of common carrier is of particular importance since it may have a substantial impact on the QOS that can be obtained. For example, ISDN and X.25 services differ substantially in terms of their delay, jitter and throughput performance.

[2]In this case, the overall route is determined through negotiation with the window entities of the intermediate sites.

[3]A directory service can be used to resolve the peer site identifier into the common carrier NTE addresses that identify the peer site's ramps.

[4]The initial window values may be determined through the use of default values or through the embedding of window entity messages within the signalling stream exercised by the channel entities.

[5]In the event of window deletion, the corresponding window values are reclaimed and quarantined. Application entities recover from window deletion by re-establishing their associations. During brief periods of inactivity, peer entities may maintain an idle *handshake* that exercises their association and its corresponding windows.

bindings between associations, windows, and channels, the window entity must operate a resource allocation strategy that determines the bandwidth requirements of each channel and resolves contention for common carrier and ramp resources.

One approach involves the static allocation of resources for the duration of each association. This strategy, which is commonly used within STM networks, has a number of disadvantages in the ATM-based exchange environment:

- There must be some means of verifying compliance with association-specific resource allocations. The correlation of arriving packets with individual allocations will substantially increase the complexity of in-band ramp operations;
- The window service must be explicitly notified of any variation in an association's resource requirement; and
- The service must retain state information concerning individual associations and reclaim resources upon termination. In the absence of a reliable notification procedure the service must operate a deadman's handle or similar timeout mechanism.

An alternative strategy is to monitor the actual traffic flowing through exchange windows and dynamically react to variations in the aggregate traffic pattern. If the window service implements the strategy on an out-of-band basis then the approach has the added benefit of being *non-invasive* with respect to individual associations and exchange ramps: the ramps collect window and channel statistics but do not concern themselves with individual associations or the enforcement of compliance.

The non-invasive strategy relies on the statistical benefits of allocating resources on an aggregate basis. The arrival pattern at each channel is determined by the aggregate arrivals on all of the associations bound to its windows. The bandwidth allocated to each channel can be based on an *average* traffic measurement that is not subject to high frequency variations. The service can respond to long term variations in traffic by adjusting channel resources[1] or by rebinding windows to alternative channels. Each allocation should include an additional reserve that, in conjunction with the elastic buffers within the ramp, smooths out the peaks and troughs induced by high frequency variations in the aggregate traffic. The service must monitor each window's status to ensure that persistent queues do not develop within the ramps.

---

[1]Adjustments to channel characteristics are network dependent, and so the implementation of this function is delegated to the appropriate channel entity.

The support of jitter sensitive traffic is of particular concern to the window service. This traffic must be partitioned from bursty traffic that can induce transient queues within exchange ramps and portals. The partitioning is effected by assigning jitter sensitive associations[1] to *expedited* windows that obtain priority service at the ramps. The resource allocation strategy must ensure that the bandwidth assigned to each channel is at least sufficient to support the peak demands of its expedited windows.[2]

10.3 The Channel Service

Channels are supported by exchange ramps which are attached to common carrier networks. The ramps format the bearer service provided by the common carrier into a packet transfer service suitable for the exchange of CFR packets. Although the channel formatting function must be performed in-band within the ramps, the management of the channels is performed by the out-of-band channel service.

The channel service of each site is made up of distinct entities that are responsible for the configuration and monitoring of the individual exchange ramps. In addition to its captive ramp, each channel entity interacts with the site's window entity and the management of the ramp's common carrier network.

Service Description

Channel entities process window service requests to:

- Construct new channels;
- Adjust window bindings; and
- Adjust existing channels.

A channel creation request specifies channel QOS attributes and the NTE address of a peer ramp accessible through the common carrier. The channel service allocates a local channel identifier which is returned to the window entity for use in subsequent transactions. When the carrier service is connection-oriented, the channel entity performs the appropriate signalling function[3] to establish a

---

[1]These associations are identified through the attributes supplied by the secretary during association establishment.

[2]If the expedited traffic arises from continuous encodings then this requirement is already accounted for within the channel's *average* traffic.

[3]In the Unison pilot network the channel entity accesses a separate *signal service* that implements the ISDN signalling protocol.

connection between the peer ramps, and configures its ramp to format the connection into an exchange channel. In the case of a connectionless service the channel entity simply configures the ramp to associate the QOS attributes and peer NTE address with the local channel identifier.

Requests for window binding adjustments specify the window value to be added to or deleted from a channel. A request to add a window must also specify the channel identifier, the remote window values supplied by the peer site, and the relative priority of the window. The channel entity interacts with the ramp to ensure that:

- the CFR interface extracts slots with destination addresses that fall within supported windows;
- The window address fields within extracted packets are translated to the corresponding values expected at the peer site;
- Packets received at each window are transmitted on the appropriate channel; and
- Expedited windows obtain priority service at ramp queues.

Requests for channel adjustments specify the local channel identifier and the revised QOS. The channel entity may interact with its ramp and/or the management of the carrier network to fulfil individual requests. For example, the bandwidth available at an ISDN-based channel is adjusted through the addition or deletion of circuit-switched connections. Additional bandwidth is provided by establishing new connections and adjusting the ramp configuration to merge the corresponding timeslots into the existing channel. Reductions in bandwidth are accomplished by: selecting the connections to be deleted; extracting the corresponding timeslots from the channel; and using ISDN signalling to drop the connections.

In addition to servicing window requests, the channel entity monitors the status of its ramp and the associated common carrier interface. It maintains the corresponding entries within the shared management database, and responds to carrier indications, such as the deletion of existing connections or unsolicited proposals to establish new connections. The latter arise when peer sites initiate the construction of channels. The recipient channel entity interprets the signalling information and generates an *upcall* RPC to its window entity. The window service processes the request for channel establishment and authorizes the acceptance or rejection of the incoming connection.

Discussion

The window service may take advantage of the close relationship between the channel entities and their ramps by delegating aspects of its resource allocation function to the channel service. In particular, individual channel entities may be authorized to adjust the resources allocated to their channels within margins specified by the window service. The margin assignment ensures that the local window entity retains its authority over the site's resources.

The distribution of responsibility can improve the responsiveness of the non-invasive strategy employed by the window service. Each channel entity operates a dynamic allocation algorithm that balances the demands of its channels with respect to the local capacity available at the ramp. This algorithm responds to variations in the channel traffic patterns which are detected through the periodic interrogation of the ramp.[1] To improve its responsiveness to increased load, the algorithm may include a predictive component that allocates otherwise idle resources in advance of demand. This may be combined with a hysteresis property that delays the reclamation of excess resources during temporary lulls in channel activity.

The design and implementation of the lower layer services is presently being investigated within the Computer Laboratory and a survey of resource management policies and considerations can be found in [Harita 87]. Though some promising simulations have been conducted by Harita, the overall effectiveness of the non-invasive strategy remains to be proven. Further issues that remain to be investigated include: the transient interference observed by existing traffic when a new association is bound to a window; and the specification of an appropriate policy for the rejection of traffic under saturation conditions.

10.4 Summary

The preservation of site independence is a recurrent theme within the exchange management architecture. At the secretary layer, site independence is protected by the *need to know* approach to name distribution. At the window and channel layers each site retains control over its address space and physical resources.

---

[1]Ramps may also generate alarm indications when internal queues exceed designated high water marks.

A further theme is the exploitation of out-of-band communication. The association establishment procedure of the secretary service and the *non-invasive* strategy of the lower layers depend heavily on the use of out-of-band techniques to support the configuration and monitoring of the streamlined in-band components.

Chapter 11
Analysis and Experimental Results

This chapter presents an analysis of the exchange architecture and the results of experiments performed using the pilot exchange implementation. The analysis proceeds through the development of performance models that are based on the structural organization of the exchange components. The purpose of the analysis is to confirm that the individual components behave as expected, and to gain insight into the overall performance of the pilot service. The principle performance characteristic of interest is the variation in delay experienced by individual SI-layer associations. Accordingly, the performance models developed in this chapter[1] are suited to the *particular* analysis of individual symbol streams as opposed to the *general* analysis of the overall exchange environment.

General Analysis

In the general analysis of a network, the model workload is usually defined in terms of a single arrival process that is identically applied to all of the input ports. Typically the global parameters of interest are: aggregate throughput; utilization of shared network resources; and the traffic levels above which network resources are saturated. The definition of identical arrival processes considerably simplifies the application of analytical and simulation techniques. This simplification is appropriate for the analysis of a single network component, such as a crossbar switch, when it is known that the ports will be symmetrically loaded.

Particular Analysis

Whereas general analysis investigates the aggregate impact of the workload on the network as a whole, particular analysis investigates the aggregate impact of the network on individual symbol streams. Since general analysis provides little or no insight into the performance of complex networks, whose input ports may be asymmetrically loaded, it is claimed that particular analysis techniques provide a more appropriate basis for the evaluation of alternative ATM architectures.

---

[1]And the accompanying Appendices D and E.

Particular symbol steams are transformed as they traverse a concatenated path of network resources linking the appropriate pair of network ports. In this chapter each transformation is separated into a *basic* component, observed in the absence of other network traffic, and a *contention* component, that arises from the presence of parallel symbol streams flowing through shared network resources. The basic and contention components are expressed in terms of three observable effects: delay; jitter; and the throughput limit at which the network *clips* the symbol stream. These effects correspond to the traffic performance attributes identified in Chapter 2, and, for the reasons elaborated at the end of that chapter, emphasis will be placed on the analysis of jitter.

11.1 Performance Models

Hierarchical techniques, described in [Svobodova 76] and [Lazowska 84], can be used to model the performance of the exchange architecture. The upper layer model of a particular path is an open queueing network of cascaded service centres that represent individual network resources. The queueing networks are assembled from a small number of generic service centres that represent the primary exchange resources: CFRs; portals; receive and transmit ramp halves; and ISDN transmission facilities. Although the individual service centres model physical resources, the boundary between adjacent centres is sometimes adjusted to ensure that each centre corresponds to a separable stage of the pipeline connecting the peer SI-SAPs.

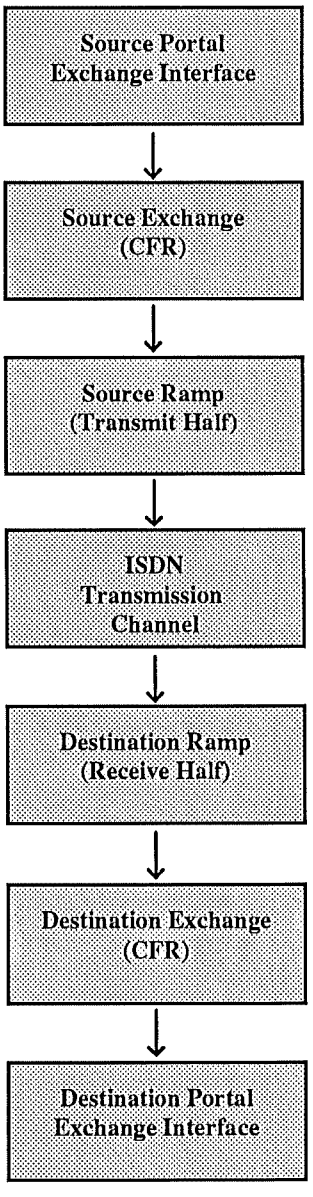

Figure 11.1: Basic Performance Model

Basic Performance Model

Analysis of the Basic Model characterizes exchange performance in terms of the throughput limit, delay and jitter imposed by each service centre. Figure 11.1 is the upper level basic model of a two hop path through an exchange configuration. The domain of the model is bounded by the portal exchange interface components which correspond to

the service access points of the SI-service. The remaining portal components have been excluded from the modelling process as they are outside the scope of the SI-service and their performance is dependent on the type of client network supported.

Contention Performance Model

Contention Model analysis focuses on the jitter arising from the presence of concurrent traffic at queueing centres. Figure 11.2 is the upper level contention model of a two hop path through an exchange configuration. This model, which is a direct extension of the basic model, identifies the traffic present at each network component. Instead of exhaustively accounting for each of the contention sources, the model groups parallel streams into abstract *contention elements* that represent the multiplexed traffic joining and leaving the path at a unique pair of network components.[1]

The primary (*vertical*) input of each centre represents traffic arriving from the previous component along the route, whilst the secondary (*horizontal*) input represents the set of contention elements joining the route at the modelled component. Similarly, the primary output represents the merged traffic, including the traffic of interest, proceeding to the next downstream component, and the secondary output represents the contention elements leaving the route at this centre.

An important aspect of queueing analysis is the requirement that each traffic element be characterized in terms of an arrival process that adequately represents a multiplexed symbol stream. Typically the symbols originate from either continuous or bursty encoding processes that can be approximated using

[1]The label associated with each element identifies the point at which it joins the path and its corresponding subscript denotes the point of separation. The present model and experimental programme only consider unidirectional traffic at the ramps and portals. The interpretation of the secondary elements could be extended to account for bidirectional and relaying activities. Some components, such as the ISDN and the exchange CFRs, are bidirectional in nature. In the case of other components, there may be further dependencies arising from contention at duplex modules such as CFR controller nodes.

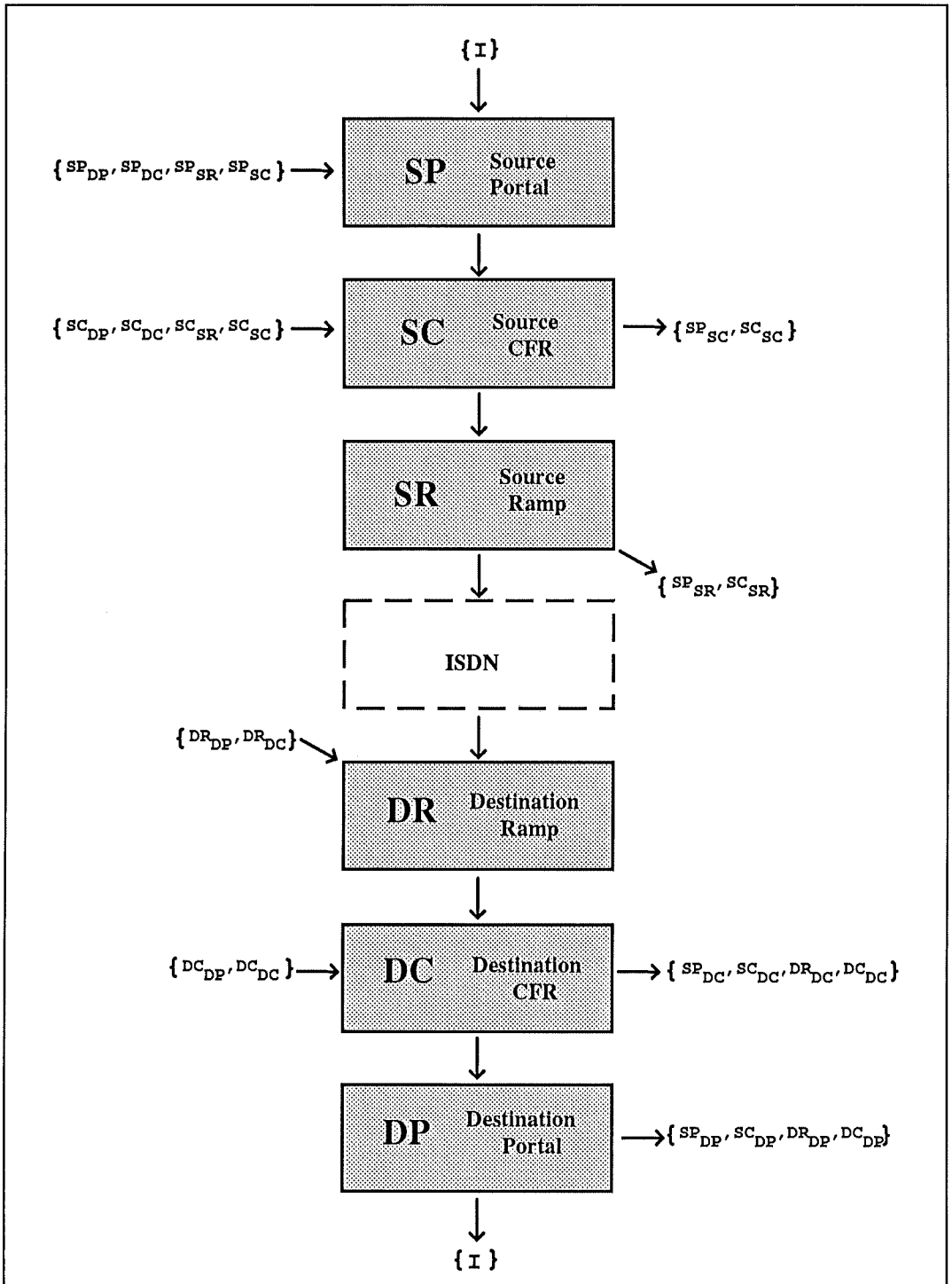

Figure 11.2: Contention Performance Model

deterministic or erlangian distributions.[1] For modelling purposes these streams have been multiplexed together at portal, ramp, and CFR components that are external to the particular path, and whose service characteristics are unknown.

Lower Level Models

The analysis of both performance models is based on the development and substitution of lower layer models representing the more complex service centres. Suitable models, representing the exchange CFRs and ramps, are developed and analyzed in Appendices D and E. In some cases the analysis of the lower layer models is based on their further decomposition into well known queueing problems that can be solved using conventional methods described in the literature.

11.2 Basic Results

The experimental programme included a number of experiments that investigated the *basic response* of the principal exchange resources.[2] The following subsections review the results obtained and their relationship to the successive stages of the basic model. These results have been used to validate and parametrize the individual lower layer models.

11.2.1 Exchange CFRs

CFR and peer portal performance was investigated using a *single hop* configuration to support the direct exchange of packets between local peer stations. Figure 11.3 is a delay histogram that illustrates the basic response of the CFR. This response incorporates:

- 15.8 usec of fixed delay, representing the physical delay associated with packet transmission; and
- 8.2 usec of jitter, induced by the alignment of asynchronous packet arrivals with the synchronous CFR frame structure.

---

[1]In the literature, Poisson arrival processes are commonly used to model the aggregate arrivals arising from a reasonable number (n >= 10) of concurrent streams. For a small number of streams a hyperexponential arrival distribution may be used. However, [Kim 83] reports that the Poisson approximation may remain valid so long as the overall utilization is relatively low (<0.1). Furthermore, the jitter induced by the exchange CFRs will help to randomize the inputs arriving at ramps and destination portals.

[2]The basic experiments were performed in the absence of contention traffic. The experimental apparatus and procedures are described in Appendix C.

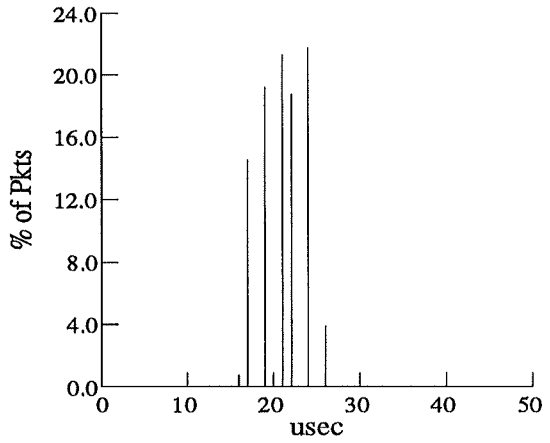

Figure 11.3: Basic CFR Response

The maximum throughput between stations is limited by this basic response and by the packet loading and unloading times at the peer portals and ramps. In practice, it is the individual stations, in particular the ramps, that determine the packet transfer rate. These results are consistent with the overview of slotted ring performance, presented in Appendix A, and the detailed CFR model developed in Appendix D.

11.2.2 Portals

The analysis of portal performance is restricted to the interface components that are responsible for the loading of CFR packets at the source portal and the unloading of packets at their destination. For the portals used in the experiments, the delay attributable to packet loading was found to be 22 usec (±1 usec), and the corresponding unloading delay was 28 usec. The difference in delay is less than the CFR transmission time, and so the unloading of each packet may be fully overlapped by the loading and transmission of its successor. Although the destination portal contributes to the fixed delay of the basic model it does not limit the overall packet throughput.

UDL Block Size (Packets)	Synthetic Portal Throughput (KPPS)
1	2.8
2	4.7
4	7.9
10	13.2
147	22.6

Table 11.1:
CFR Block Throughput

The measured portal delays exclude the processing performed at the beginning and end of every UDL block. For modelling purposes this overhead is beyond the scope of the SI-service and is therefore excluded from the performance model. In practice, the per-block processing reduces the peer portal throughput and is particularly significant when short UDL blocks are exchanged. Table 11.1 illustrates the relationship between UDL block size and the measured peer portal throughput.

11.2.3 ISDN Ramps

Ramp experiments were performed using a *dual hop* configuration in which packets transit a path comprising a pair of CFRs and a connecting channel supported by the subject peer ramp halves.[1] The basic response, illustrated in Figure 11.4(a), has delay and jitter components that are two orders of magnitude greater than those exhibited by the exchange CFRs.

The bulk of the basic delay is accounted for by the deskewing buffer within the receive half and the pipelining of frame processing stages within the transmit half. These stages, which are synchronized to the ISDN, are clocked at 500 usec quad frame intervals. A consequence of this design is that each of the pipeline and buffering stages makes a substantial contribution to the fixed delay.

The jitter spectrum is broadly organized into a leading spike, a densely populated central cluster, and a sparsely populated trailing cluster. The central cluster accounts for the bulk of the samples, and its 500 usec width corresponds to the alignment of asynchronous packet arrivals with the synchronous quad frame structure. The trailing cluster is also about 500 usec wide and is offset from the central cluster by a quad frame interval. The samples falling within this cluster correspond to the small fraction of packets that are delayed by the periodic transmission of the channel synchronization sequence.[2]

---

[1]In these experiments, the ramp halves were directly connected to each other via a *null* transmission channel. Although the experimental results include delay and jitter elements arising from CFR packet transfers, the overall response is dominated by the ramp performance. A report on the experimental results, and their analysis in terms of the ramp performance model, can be found in Appendix E.

[2]It has been suggested that the spike at the leading edge of the spectrum represents a quantum effect of the choice of process schedules within the ramp transputers.

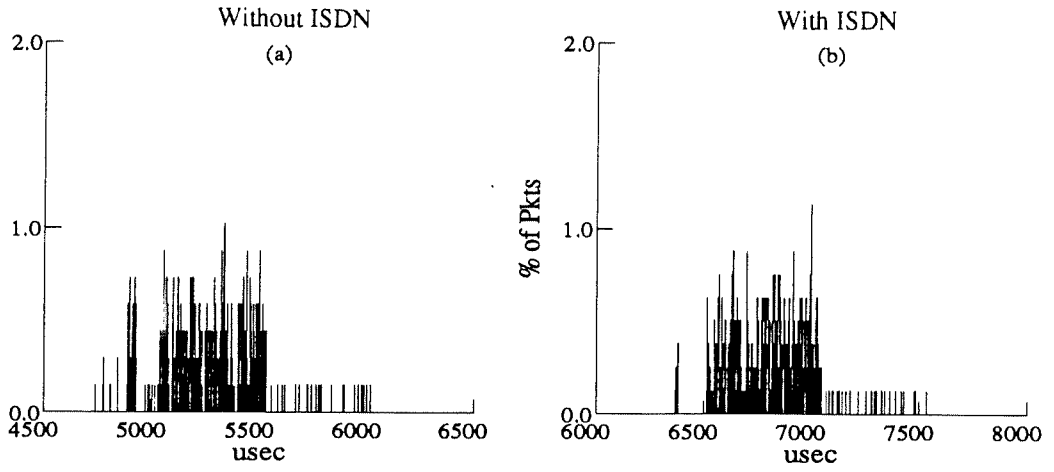

Figure 11.4: Basic Ramp Response

The maximum throughput between peer exchanges is limited by the capacity of the inter-site channel, which can be adjusted in single timeslot increments. The maximum width supported by the primary rate interface is 30 timeslots, which corresponds to a transmission capacity of 6 KPPS.[1] In practice, the overall throughput is limited by the packet loading and unloading delays induced by the CFR software embedded within the ramps. The aggregate capacity of the transmit half is limited to 4.4 KPPS and the receive half is limited to 3.6 KPPS. Accordingly, the receive half software is the throughput bottleneck whenever the channel width exceeds 18 timeslots.

11.2.4 ISDN Transmission

The ISDN supports a fixed throughput STM service between peer ramps. Delay within the network arises from propagation delay along the transmission media and ISDN frame delay at the circuit switches along the route.[2] The basic delay through the network is determined by the routing of the timeslots associated with the channel. This routing is fixed when the timeslots are assigned, and so, in the absence of high frequency fluctuations in timeslot assignment, there is little or no variation in ISDN delay.

---

[1]Kilopackets per second.

[2]The jitter induced by these network components is insignificant relative to the jitter induced at an ISDN ramp.

Figure 11.4(b) was obtained using traditional land lines as the ISDN transmission media.[1] This histogram illustrates the differential impact of ISDN transmission on basic response. The shape of the jitter spectrum, which is the same as in Figure 11.4(a), confirms that significant jitter components are not induced within the ISDN. The horizontal position of the sample cluster indicates that the ISDN adds about 1600 usec to the overall basic delay. It is believed that up to 125 usec of this delay is attributable to ISDN frame alignment at the circuit switch, with the balance accounted for by propagation delay through the two land lines.[2]

The experiment was repeated after substituting a microwave link for one of the two land lines. Although the overall shape of the spectrum remained unchanged, the differential delay was reduced to 1120 usec. Comparison of the two results suggests that the propagation delay over the microwave link is between 250 and 318 usec. This result is consistent with the propagation of radio waves over the estimated 75 km of the route between the Computer Laboratory and the circuit switch.

11.2.5 Relaying

The performance models can be extended to encompass the exchange relaying function through the introduction of additional service centres representing the CFR, ramp, and ISDN transmission facilities associated with relay sites. The principal performance implications are the delay and jitter components introduced by each pair of ramp halves along the relay path of interest.

Figure 11.5(a) and Figure 11.5(b) are the basic responses of particular paths that transit one and two relay sites, respectively. These jitter spectra are consistent with the results anticipated by the extended performance model. To a first approximation, the imposition of each relay site is equivalent to convolution with the basic response of Figure 11.4. With the addition of each site the jitter spectrum is: attenuated by the jitter induced within the relaying ramp halves; and shifted horizontally by an amount corresponding to the fixed delay through the ramps and the ISDN.

---

[1]The network was configured so that all 15 of the channel timeslots followed a common route through a single circuit switch and two primary rate carriers.

[2]Since the physical length of the land lines is unknown it is difficult to validate this result.

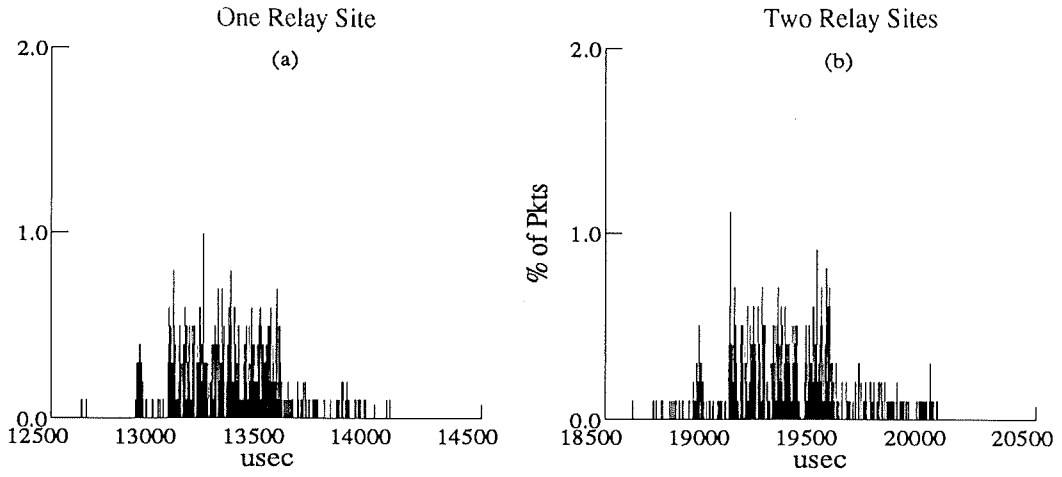

Figure 11.5: Basic Relay Response

11.2.6 Summary

The experimental results confirm that the CFR is suited to the exchange application. Exchange performance is constrained by the ISDN ramp, which constitutes the throughput bottleneck and dominates the basic response. Although the bottleneck can be eased through tuning and refinement, the bulk of the jitter arises from alignment and synchronization components that are fundamental to the present ramp design.

In Table 11.2 the delay probe results for the single and dual hop configurations have been normalized with respect to a variety of user data rates. The table entries give the relative impact, in packet intervals, of the basic delay and jitter experienced by a particular symbol stream.[1]

11.3 Contention Results

The investigation of contention effects has been restricted to the CFR and ramp service centres. These are the principal points of contention interaction that fall

[1]Appendix C discusses the relevance of the normalized packet interval measure. For the purposes of delay analysis these unit intervals express the delay arising within the network as a multiple of the packetization delay absorbed at a source point of attachment. From a jitter perspective the packet interval measure is indicative of the extended length, in packets, of an elastic buffer implementing jitter compensation at the destination point of attachment. The values reported in the table are derived from impulse measurements and do not take account of various effects arising from the continuous flow of packets.

Configuration	User Data Rate (Kbps) (Packet Rate (KPPS))				
	64 (0.3)	256 (1.1)	800 (3.6)	2000 (8.9)	5000 (22.3)
Single Hop (CFR Only)	packet intervals				
Average Impulse Delay =14usec	0.0042	0.015	0.050	0.12	0.31
99% Impulse Jitter = 8usec	0.0024	0.088	0.029	0.071	0.18
Two Hops (CFRs, Ramps, ISDNs)	packet intervals				
Average Impulse Delay = 6797usec	2.0	7.5	24	---	---
99% Impulse Jitter = 1104usec	0.331	1.3	4.0	---	---

Table 11.2: Basic Results Summary

within the domain of the site interconnection service.

11.3.1 CFR Contention

In contention environments, transmitting CFR stations compete for available ring bandwidth and for the receiver capacity available at particular stations. Ring contention is resolved by the empty slot protocol which defers packet transmission until a slot is available. In the event of receiver contention, transmissions are repeated until the packet transfer is successful.

Figure 11.6(b) is a histogram that illustrates the jitter induced by receiver contention.[1] This spectrum consists of a number of clusters that are positioned along the horizontal axis. The number of samples within each cluster represents the fraction of transfers completed during a given transmission attempt, and the interval between clusters corresponds to the delay between transmissions.

Although there is no guarantee that a transfer will be successfully completed within a fixed number of attempts, the CFR station logic imposes a limit on the number of automatic retransmissions. Although this limit constrains the maximum jitter associated with a successful transfer, some residual fraction of all transfers will be lost.[2] The complete response to a given contention pattern can be expressed in terms of the transfer failure rate and the delay distribution associated with

---

[1]Figure 11.6(a) is the basic response in the absence of contention traffic.

[2]The CFR does not ensure the fair resolution of receiver contention and complete starvation may occur. This issue is discussed within the detailed analysis of Appendix D.

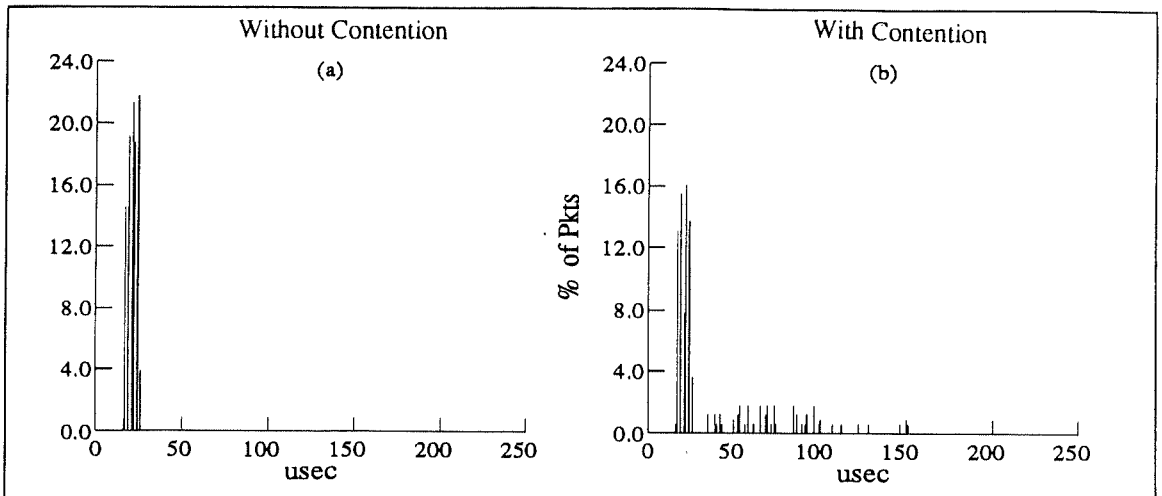

Figure 11.6: CFR Contention Response

successful transfers.

11.3.2 Ramp Contention

Ramp contention arises from the multiplexing and switching of the parallel traffic streams generated by exchange portals. The overall effect is the emergence of packet queueing that induces jitter into the particular symbol stream of interest. The following paragraphs present a summary of the ramp contention effects that are discussed in Appendix E.

At the transmit half, the capacity of the CFR interface may be instantaneously exceeded. This form of contention provokes CFR receiver contention, which can be viewed as a form of back pressure distribution that results in packet queueing within the individual contention sources. Similarly, channel dependent queues develop within the ramp whenever the aggregate traffic destined for a shared channel exceeds the available capacity.

At the receive half, packets arriving on parallel channels are recovered from incoming ISDN frames and are presented to the CFR interface software for transmission. Substantial queues may develop at this interface which is the channel independent throughput bottleneck. Furthermore, the arrivals within each incoming frame are presented to the interface as a single batch. The relative positioning of a packet within its batch influences the delay experienced at the CFR interface.

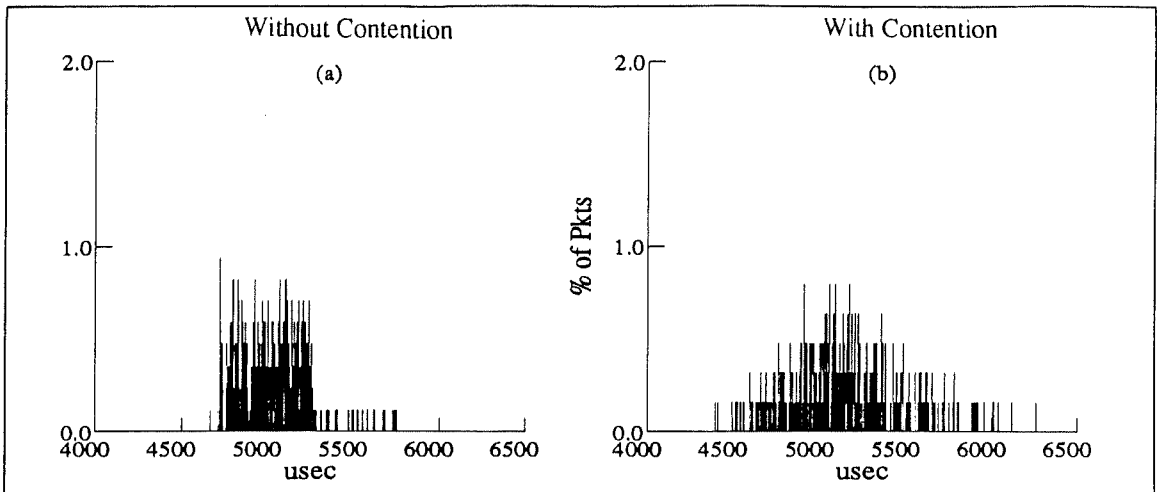

Figure 11.7: Ramp Contention Response

Figure 11.7(b) is an example of ramp response in the presence of periodic contention loads.[1] Although the overall traffic in this experiment was not sufficient to induce queueing within the ramps, the jitter spectrum has been attenuated by the batching of packet arrivals at the receive half CFR interface.

11.4 Summary

This chapter has presented experimental and analytical evaluations of exchange performance. Although the cases studied are not exhaustive, the results demonstrate the operation of the exchange, and have validated some aspects of the performance models. In some cases, discrepancies between the analytical predictions and the experimental observations were identified. These results were analyzed in greater detail in order to: refine the models; gain further insight into the operation and performance of individual components; and improve the present implementation.

The experimental results have been encouraging and indicate that the exchange architecture can deliver reasonable jitter and throughput performance. However, the contention results illustrate the implications of multiplexing jitter-sensitive traffic with bursty symbol streams. Fortunately this is an area where expedited transfer can gainfully be employed. Where necessary, the expedited queues within exchange components can employ finite storage limits in order to guarantee a

[1]Figure 11.7(a) is the response in the absence of contention traffic.

maximum jitter constraint on delivered packets.[1] In these cases the overall jitter performance will be characterized in terms of the delay density *and* the fraction of packets that are successfully delivered.[2]

The contention results also demonstrate the importance of the bandwidth allocation function of the exchange management services. Although expedited transfer provides some relief from transient events, the longer term allocation of expedited capacity must be monitored and controlled.

The experimental characterization of delay and jitter has been based on the response to single packet impulses. This *single shot* approach does not provide a comprehensive measure of a configuration's delay response but it has proven to be a simple experimental technique that provides insight into the performance and operation of exchange components. Furthermore, when the experiment is performed in the presence of contention traffic generated by synthetic portals, the impulse response is indicative of the incremental performance that could have been achieved by those portals.

A final observation concerns the application of *particular* analysis to the evaluation of overall exchange performance. It is suggested that, for a given exchange configuration and workload, a global performance *snapshot* could be built up through the exhaustive application of particular analysis to each of the concurrently active streams. Further work is required to determine whether this technique yields acceptable results and whether or not it can be used to dynamically determine the optimum channel configuration for a given workload and set of exchange resources.

---

[1]When the expedited demand instantaneously exceeds the overall capacity, new arrivals will be lost. For many jitter-sensitive encodings it may be more appropriate to completely or partly flush a full queue so that *older* packets are discarded in preference to *newer* arrivals. The strategy used for non-expedited, or *normal*, traffic may be quite different. For example, relatively long *normal* queues can be used to permit queue build-up during bursts, and dissipation during gaps.

[2]A similar observation is made in Appendix D with respect to the maximum retransmission limit imposed by the CFR nodes.

Chapter 12
Conclusion

The initial motivation for this work was an interest in the transparent interconnection of a relatively small number of geographically dispersed LANs. The benefits and complexity of service integration had already been studied within Project Universe. Experience within that project has shown that the performance demands of the multi-service environment are not fully satisfied by a traditional PSTM service. However, it is also apparent that many desirable characteristics, such as dynamic bandwidth allocation and rate adaption, are not supported by STM transmission. Accordingly, ATM has been identified as a sensible compromise between the flexibility offered by PSTM and the performance advantages of STM.

During the course of this work it became clear that a longer term view of site interconnection was required, and so the architecture presented in this dissertation emphasizes site independence and universal access issues that are not addressed within fixed topology networks. A distinct SI-layer has been identified, and the independent subsystems that make up this layer cooperate on an *as required* basis to provide the SI-service. Universal access is achieved through the use of lower layer common carrier networks in place of, or in conjunction with, fixed topology networks. The service is provided through the operation of a common ATM encoding that supports service integration and shared access to carrier facilities. Furthermore, the SI-layer provides increased scope for evolutionary changes by divorcing the upper layer local networks from the common carrier substrate.

Experience with the pilot exchange implementation has substantiated many of the claims made for the proposed site interconnection architecture. Furthermore, the characterization of the SI-service has been refined as a result of the design, implementation, and experimental evaluation of the exchange environment. Accordingly, the SI-layer description presented in Chapter 5 is one of the principal results of this investigation. The remainder of this chapter describes additional insights that have been gained and the areas for further investigation that have been identified during the course of this research.

12.1 Insights Gained

Telecommunication Models

The hierarchical network model developed in Chapters 1 through 3 is a generalization of the layered structure used in the OSI reference model. The model presented here identifies encoding, transmission, multiplexing, and switching as the fundamental operations that can be supported at any layer within a *hierarchy* of networks. In addition, the OSI concept of layer *protocols* has been extended to represent *encodings* arising from a wide variety of telecommunication services. The author plans to further the development of this abstraction and investigate its usefulness in teaching and research applications.

ATM Encodings

An important question in the design of an ATM service is the packet format used in the common encoding. The principal issue is the symbol payload to be transferred within each packet. The length of this symbol field determines the packet rate necessary to achieve a given throughput. It also influences the selection of an addressing scheme, as the address fields represent transmission overhead associated with each packet of symbols.

A relatively small packet size simplifies the hardware implementation of ATM multiplexing and switching components. Larger packets increase the absolute jitter incurred as a result of packet collisions within the network, and increase the symbol loss associated with the non-delivery of individual packets. However, larger packet sizes also reduce the aggregate packet rate, and associated processing overhead, at terminal nodes performing segmentation and reassembly functions.

In the development of an ATM broadband service, designed to operate at high transmission rates, a 4 kilobit packet payload may seem reasonable:

- At 500 Mbps the packet transmission interval is only 8 microseconds[1] and so single packet delays induce little absolute jitter; and
- There is sufficient transmission capacity that low bandwidth applications can use lower density encodings to increase their tolerance to packet non-delivery.[2]

---

[1]For the purposes of this discussion, slot structure and address field overheads can be ignored.

[2]For example, if the payload represents too long a sample interval for the transfer of 64 Kbps PCM encoded telephony, then: a different encoding can be used; partly

However, if the 4 kilobit packet format is used within a 32 Kbps mobile data network the packet transmission interval will be 125 milliseconds. The choice of packet length limits the range of transmission rates over which ATM rate adaption can be achieved. If the rate adaption and integration potential of ATM is to be fully exploited, then the packet length must be determined by the degree of jitter and symbol loss that can be tolerated *at the lowest transmission rate to be used within the integrated environment.*

There is clearly a strong argument for keeping the packet length as short as possible. Experience with the earlier Cambridge Ring showed that very small packets led to excessive segmentation overheads and complicated the support of concurrent transmissions. Experience with the present 256 bit payload has shown it to be a reasonable compromise. Although the segmentation overheads are still significant, multi-megabit symbol rates can be sustained. It is expected that broadband rates can be achieved through the use of faster processing components and increased parallelism[1] within the terminal devices.

The SI-Service and the Exchange Architecture

The pilot implementation has demonstrated the ability of the present CFR-based exchanges and ISDN ramps to support the interconnection of a number of local networks and a variety of upper layer client protocols. Although an ATM-based carrier service would lead to simpler ramps, the present configuration demonstrates the feasibility of operating an ATM service over STM transmission facilities.[2]

The design of the exchange architecture illustrates the benefits to be derived from the convergence of the telephony and computer communication traditions. For example, the out-of-band approach to network management is an extension of the common channel signalling approach adopted in ISDN. Similarly, results derived from distributed computing research[3] have been applied to the design of the exchange management components. The combination of these two strategies has proven especially powerful. Out-of-band management techniques have helped to streamline the design of in-band components, and ensure that the management of each exchange retains control over its own site's resources. For example, the

---

populated packets can be transmitted; or packet transfers can be replicated.

[1]This issue is discussed in the CFR section, below.

[2]This may be a useful result in itself. Some form of STM hybrid or overlay may be required during the deployment of ATM-based carrier services.

[3]In particular, the remote procedure call paradigm.

deferred binding of titles to addresses, achieved by the secretary service, limits the degree of configuration information that is exported to peer sites.

One of the novel aspects of this research has been the emphasis on the jitter performance of the site interconnection service. For the most part, the jitter induced by the exchange multiplexing and switching components is acceptable. The principal sources of jitter, within the ISDN ramps, are attributable to the common carrier frame rate rather than the exchange architecture or its packet format. Although the present level of jitter is compatible with many voice and video encodings, further work is required in order to establish an appropriate jitter objective for the SI-layer.

The experimental programme and the accompanying analysis have provided a better understanding of the detailed operation of many exchange functions. In some cases this analysis has identified ways in which the operation of the existing components can be improved through relatively minor changes to the software controlling their operation.[1] In other cases experience with the pilot exchange has provided further insights concerning the functionality provided within the SI-Layer:

- The implementation of the expedited transfer function has proven to be cumbersome, because packet expedition must be considered at every point where contention can arise. A more uniform approach to expedited transfer would appear desirable; and
- The present hop-by-hop approach to error detection is not sufficiently robust to prevent address or data field corruption within the SI-layer.[2] The ATM packet format could be extended to include a check sequence that is initially computed by the source client device and is recomputed and validated after each hop.[3]

---

[1]For example, there has been considerable improvement in the expedited transfer support provided by the ISDN ramps. Similarly, the analysis of synchronization effects, described in Appendix E, has suggested a simple software modification that will halve the observed frequency of synchronization vacations.

[2]These errors, which are very infrequent, occur when packets are stored in unprotected memory or are presented at unprotected interfaces such as the host data bus of the CFR controller nodes.

[3]Simple end-to-end validation of the check sequence is not sufficient since the integrity of the packet address fields must be protected in order to avoid the incorrect routing and delivery of packets. In applications requiring error-protected symbol transmission, the destination client can also use the check sequence to detect and discard corrupted packets. An improved error detection capability would improve the monitoring capability of the SI-layer management.

CFR Experience

The pilot exchange implementation has been the first design to use the CFR network technology. Aside from some initial teething problems the overall experience with the CFR has been quite good. The experimental results indicates that the capacity of the exchange CFRs is appropriate for use with existing local networks and ISDN carrier facilities. The provision of bridge mode controller operation has considerably simplified the design of exchange ramps and the present CFR portals.

One difficulty with the present CFR implementation is the substantial per-packet overhead incurred at the host interface. The segmentation and reassembly functions are processor intensive, and so the performance benefits to be gained from customized interface devices are likely to be limited. The present portals use microprocessors[1] to support software implementations of the UDL protocol. The principal throughput limitations arise from:

- Interrupt handling overhead within the processors;
- The eight bit width of the host interface data bus; and
- The maximum transfer rate of the controller chips.

The utility of the CFR could be immediately improved by providing a full 32 bit data bus. In the longer term, it would be attractive to encapsulate the CFR controller logic as a VLSI mega-cell that could be incorporated into a variety of processor and interface devices.

A more substantial difficulty with the CFR design is the receiver contention problem described in Chapter 11, and analyzed in greater detail in Appendix D. This contention problem is common to most PSTM and ATM architectures. However, its impact is proportionally greater in ATM environments, where each SDU may be segmented into a large number of packets. In the absence of a low level retransmission mechanism, the periodic loss of individual packets may preclude the transfer of complete SDUs. Although the response and retry mechanisms of the CFR provide some tolerance to receiver contention, they do not ensure the fair distribution of receiver capacity.

Analysis Methodologies

The methodology used in the exchange performance analysis is based on the hierarchical application of queueing models. On the whole this approach has proven effective with respect to the structured decomposition of the exchange analysis problem into a number of manageable units. However, it is not clear that

---

[1]Such as the Motorola 68020, the Acorn Risc Machine, and the Inmos Transputer.

the lower level queue-based models are appropriate to the ATM environment. Although queueing techniques are commonly used to determine the overall throughput and delay bounds of traditional PSTM networks, they are less frequently used to evaluate delay distribution, i.e. jitter.

In this research a *signal-oriented* approach has been taken to the characterization and analysis of ATM jitter, and some of the terminology that has been adopted is commonly associated with signal analysis. The emphasis has been on determining the manner in which symbol streams are transformed as they traverse the network. The relatively simple experimental programme has characterized the impulse response of exchange components in the presence of various contention elements. The corresponding jitter results have been presented in the spectral form, commonly used in signal analysis, rather than the delay distribution form, normally associated with queueing systems.

It is suggested that more effective ATM modelling tools may be derived through the composition of results derived from both signal and queueing theory.[1] Convolution techniques are firmly embedded within both disciplines and it is likely that a suitable compromise can be achieved. A composite methodology may provide an enhanced theoretical framework for the understanding of the PSTM-STM compromise that is the principle attraction of the ATM environment.

12.2 Recommendations for Further Work

A number of suggestions for further research have been incorporated into the preceding section of this chapter.[2] Other research issues that have been identified are primarily concerned with the performance, management, and application of the site interconnection layer.

The apparatus of the pilot implementation can be configured for use in a wide range of experiments related to exchange performance and resource management. Some aspects that remain to be investigated are:[3]

- The routing and bandwidth allocation strategies to be used within the exchange window and channel services.

---

[1]Elements of information theory may also be of interest.

[2]These include: telecommunication modelling; ATM encodings; receiver contention; and ATM analysis tools.

[3]Some of these issues are discussed in [Harita 87] and may be investigated within the context of his PhD research.

An important issue is the responsiveness that can be achieved using the non-invasive and predictive approach suggested in Chapter 10;

- The application of rate control techniques to the prevention of congestion within the SI-layer. Although transmission throttles could be applied on an association-specific basis, there is a requirement for a suitable rate determination algorithm and a mechanism to enforce compliance at the individual SI-SAPs;
- The application of priority queueing disciplines to the suppression of jitter. The present exchange components implement a two-tier priority scheme. It is important to determine whether greater flexibility is required and, if so, whether a multi-tiered scheme would prove satisfactory; and
- Alternative schemes for the splitting of channel traffic across parallel ISDN calls. For an exchange-like architecture to be widely adopted, it will be necessary to standardize the channel formatting technique that supports the ATM overlay.

A further performance issue is the application of alternative switch technologies to the local exchange environment.[1] Although a number of switch fabrics have been developed for use in the backbone of an ATM-based carrier network, it is not clear that these designs are suitable for use in local environments that are characterized by asymmetric traffic patterns and relatively small dimensions.

If successful, the introduction of multi-service networks and site interconnection facilities will substantially alter existing telecommunication patterns. As new services and applications are developed, the iterative cycle must be continued through the further refinement of the site interconnection service. One area of particular relevance is the development of video encodings. In STM environments it is important to maximize the utilization of a fixed capacity transmission service. In ATM environments there is greater scope for the application of variable rate encodings that generate substantial transient loads, but have considerably lower average transmission requirements. It is expected that video transmission will be a major site interconnection application, and so the jitter tolerance of the encoding adopted will be a major consideration in the refinement of the SI-layer jitter objective and the ATM packet format.[2]

---

[1]This issue is discussed in greater detail in Chapter 7.

[2]The corollary of this observation is that ATM considerations should be introduced into the present CCIR deliberations on the standard encoding to be used for the transmission of high definition television services.

Given the present pilot implementation, one of the most important areas of further research is in its field application to the provision of telecommunication services. Within Project Unison, work is proceeding on the development of the distributed computing techniques that are essential to the orchestration of multi-service applications. An important issue that must be addressed is the maintenance of temporal coherence across parallel associations supporting different styles of upper layer encodings. Further work, within the Unison project, will involve the field trial of multi-service applications coordinating telephony, video, and image transfer services.

Appendix A
Slotted Ring Networks

This appendix provides background information concerning the organization and performance of the Cambridge Ring and Cambridge Fast Ring (CFR) slotted ring networks that have been developed at the Computer Laboratory.

A.1 Ring organization

The Cambridge Ring [Wilkes 79] is a local area network that uses standard TTL technology to link digital devices.[1] The ring is a distributed packet switch that allows digital information to be transferred between host devices through the exchange of packets over an ATD multiplexed medium.

Each fixed length packet contains an eight bit destination address, an eight bit source address and a sixteen bit host data field.[2] At the ring points of attachment the devices are attached to ring *nodes* which are, in turn, attached to ring *repeaters*. The repeaters are linked together into a loop and the concatenated inter-repeater links form a single shared medium.

ATD Structure

Each repeater recovers a node clock from the incoming modulated signal received from the upstream repeater. The clock is used to regenerate the incoming data which is converted into a bit stream routed through the host node.[3] The node returns an output bit stream to the repeater and the recovered clock is used to generate the outgoing modulated signal transmitted to the next downstream repeater. In the absence of traffic to or from the host device the nodes acts as a short shift register that copies the incoming bit stream to the outgoing bit stream.

---

[1]The CFR [Hopper 86] is a second generation ring based on VLSI components.

[2]In the CFR, the data field is 256 bits long and each address field is 16 bits.

[3]In the CFR, the repeater-node interfaces are 8 bits wide.

If there is no node present, or if the node is not operational, then the repeater bypasses the node and the received data is used to generate the outgoing signal.

The ring can be thought of as a circular shift register of some exact number of bits that exceeds the slot length. When the ring is initialized, a designated *monitor* node formats the bits in this shift register into an ATD frame. Each frame consists of at least one fixed length timeslot (or simply, slot) and any remaining bits in the frame are set to zero and are referred to as the *gap*. The number of bits in the logical shift register is a function of electrical length of the ring which is determined by the cumulative delay through all of the links, repeaters and nodes of the ring. If the ring is shorter than one slot then it must be extended through the insertion of a shift register at the monitor node. If the ring is more than two slots long, then it can be formatted so that each frame, or ring *revolution*, carries a train of two or more slots followed by the gap.

A.2 The Empty Slot Protocol

Ring Access Protocol

Peer devices use the ring to exchange packets by arranging for their nodes to embed packet symbols in the slots circulating around the ring. Each slot conveys 3 header bits, a packet field, and 3 trailer bits.[1]

An eight bit address is assigned to every node and each address is unique within the context of a single ring. When a device attached to a node wishes to transmit a sequence of packets to a single destination it loads a register within its node with the address of the peer node. The value loaded will be placed with the destination address field of every packet in the sequence.[2] The source address of a packet is already known to the node.

For each packet of the sequence the device 'arms' the node by loading the node data register with the date field symbols. Once the node is armed its transmit logic scans the bits of the incoming ring data stream looking for a slot header that identifies an empty slot. When an empty slot is found the full/empty bit of that slot's header is changed on the outgoing bit stream. The packet field of the incoming slot is discarded and is replaced by the address and data of the packet

---

[1]In the CFR, the header is four bits long and the trailer is 12 bits.

[2]The same procedure can be followed to transmit individual packets. In the CFR, there is also a bridge mode of operation that is described in Chapter 7.

being transmitted. The check bit(s) within the slot trailer are set in accordance with the previous bits of the packet. The transmit logic waits for the slot to circulate around the ring during which time it will pass the destination node. When the slot header returns to the source node, the full/empty bit of the slot is restored to empty and the contents of the response bit(s) within the slot trailer are copied into a register within the node.[1] The slot cannot be used by the same node until the next revolution, and so, it becomes available for use by other nodes.

The receive logic of a node scans the destination address field of every full slot to see if it matches the address fixed in the node hardware. When a match is identified the receive logic may copy the source address and data fields of the packet into registers within the node. The packet will be rejected if the destination is busy, the packet is not selected, or a transmission error has occurred along the forward path. The response bit(s) of the slot are used as a reverse channel to inform the source node as to whether or not the slot has been *accepted* by the destination. The copy attempt will fail if the node is *busy*, i.e., the receive registers contain the contents of a previously received packet which has yet to be retrieved by the host device, or if the node is *unselected*, i.e., the host device has set the node to only accept slots transmitted by a selected source node.[2]

A.3 Upper Layer Protocols

Cambridge Ring Basic Block Layer

In the Cambridge Distributed Computing System [Needham 82], the Cambridge Ring is used to interconnect a number of host devices that provide a variety of services. Terminal concentrators, file servers, print servers, an authentications server and a bank of general purpose processors use the ring as a common packet

---

[1]This scheme is sometimes referred to as a *source delete* protocol. Some slotted ring designs use *destination delete* protocols in which slots are freed as soon as they pass the destination node. Although destination delete protocols may lead to greater aggregate capacity, they do not provide the low level response mechanism available with the source delete scheme.

[2]In the CFR there is an additional filtering mechanism described in [Hopper 86]. Furthermore, the CRC check sequence within the slot trailer is used to convey the response information. If the packet is rejected the destination node inserts a valid CRC field into the slot trailer. If the packet is accepted the node inserts an invalid CRC field. At the originating node a valid CRC field in a returned slot indicates that the packet definitely *was not* accepted by the destination and the packet can be retransmitted without risk of duplication. An invalid CRC field indicates that the packet was either accepted or that the slot contents were corrupted along the return path and the fate of the packet is unknown.

switch. The messages exchanged between these hosts are usually longer than 16 bits and so the Basic Block encoding described in [Leslie 84] is used to segment a message into a sequence of packets. Each basic block consists of some header packets followed by the message and a trailer packet.

When a host is prepared to accept a block it sets the select register of its node to accept any packet bearing its own node address. When the header packet of a block is received, the response field of the slot is marked *accepted* and the node's select register is set to only accept packets whose source addresses match the source address of the header. When the source device observes that its header packet has been accepted, it transmits the remaining packets of the block. The destination accepts a specified number of packets[1] and at the end of the block it resets its node to select all sources.

Although the ring itself is ATD multiplexed, the basic block protocol inhibits the degree of multiplexing performed at the individual points of attachment. In practice, ring nodes do not multiplex the transmission or reception of packets associated with different basic blocks. While a block exchange is in progress, header packets arriving from unselected nodes are rejected and the response bits of their slots are marked unselected. An unselected source node retries the header packet until it is accepted. The lack of multiplexing within the Cambridge Ring block protocol has led to performance restrictions at nodes supporting concurrent associations.[2]

CFR Data Link Layer

The larger packet data field of the CFR permits the use of a packet-level protocol that supports multiplexing and thereby extends the ATM characteristics of the ring to the upper layer service. The Unison Data Link layer supports the standard block level service of the CFR. This layer is briefly described in Appendix F and further details can be found in [Tennenhouse 86b] and [Tennenhouse 86c].

---

[1]The block header specifies the block length. The block trailer includes a block check sequence that is used to protect the integrity of block transmissions.

[2]Such as bridges and gateways or nodes supporting multi-service applications.

A.4 Slotted Ring Performance

The analysis developed in this section describes general aspects of slotted ring performance.[1] For the most part this discussion represents an extension of previous work[2] though the discussion on *practical limitations* introduces some new observations. A more detailed analysis of CFR performance is presented in Appendix D.

System Bandwidth

The total system bandwidth available to the nodes attached to a ring is:

$$\text{SysBw} = \frac{\text{Number of data bits on ring}}{\text{Length of ring in bits}} * \text{Clocking rate} \qquad \text{(A.1}^{3}\text{)}$$

For a ring operating at frequency F_r , where each frame contains N_S slots and a gap of G bits, equation A.1 can be expressed as:

$$\text{SysBw} = \frac{N_S * P_s}{(N_S * P_s) + G} * (P_d/P_s) * F_r \qquad \text{(A.2)}$$

where

P_s is the total length of each slot, measured in bits, and P_d is the data portion of each slot.

Shared Bandwidth

In order to ensure that the available ring bandwidth is shared amongst all active transmitters the access protocol prohibits a node from immediately re-using a slot. Returning slots must be marked empty and some ring bandwidth is lost as empty slots are passed between nodes. The ring bandwidth shared amongst a fixed number of actively transmitting nodes is:

$$\text{SharedBw} = \frac{\text{SysBw} * A}{N_S + 1} \qquad A \leq N_S \qquad \text{(A.3a)}$$

---

[1]The operation of a source delete protocol is assumed in this analysis.

[2]In particular, [Hopper 78] and [Temple 84].

[3]Equation A.1 is taken from page 39 of [Temple 84].

$$\text{SharedBw} = \frac{\text{SysBw} * A}{A + 1} \qquad A \geq N_s \qquad \text{(A.3b)}$$

where

A is the total number of concurrently active transmitters.

Transmission Bandwidth

A transmitting node cannot consume all of the shared capacity because once a slot has been injected into the ring the node must wait a full ring revolution for the slot to return. Since the returning slot cannot be immediately re-used a single node can only transmit in one out of every $N_s + 1$ slots. The practice of marking returning slots empty and making them available for use by downstream nodes ensures that the ring bandwidth is equally shared amongst the active nodes. Each node is *guaranteed* a transmission bandwidth of:

$$\text{TxBw} = \frac{\text{SysBw}}{N_s + 1} \qquad A \leq N_s \qquad \text{(A.4a[1])}$$

$$\text{TxBw} = \frac{\text{SysBw}}{A + 1} \qquad A \geq N_s \qquad \text{(A.4b)}$$

Once a slot has been injected into the ring the transmitting node must wait a full ring revolution for the slot to return before transmitting again. Since the returning slot cannot be immediately re-used a single node can only transmit in one out of every $N_s + 1$ slots. In practice, it is difficult to design node and host logic that will examine the response bits of the returning slot and be prepared to transmit again in the immediately following slot. Therefore, in the absence of other active transmitters:

$$\text{MaxTxBw} = \frac{\text{SysBw}}{N_s + 1 + D} \qquad \text{(A.5)}$$

where

D is the delay, in unused slots, between transmissions[2].

---

[1]Equations A.4a and A.4b are equivalent to equation 3.13 of [Hopper 78] with the substitution of $(N_s * P_e + G)$ for $(N * B_e)$ and (A) for $(N * p(z))$.

[2]The value of D may vary amongst the nodes of a single ring. The cumulative effect of the delays has been excluded from the previous equations for SharedBw and TxBw which are, therefore, slightly optimistic.

As the number of active transmitters increases the TxBw available to each of the transmitters gradually declines from MaxTxBw to MinTxBw, the value of txBW when A is equal to N, the total number of nodes attached to the ring:

$$\text{MinTxBw} = \frac{\text{SysBw}}{N + 1} \qquad N \geq N_s + D \qquad \text{(A.6)}$$

Access Delay

An attractive feature of the empty slot protocol is the short access delay experienced by nodes waiting to use the shared medium. Two factors contribute to ring access delay:

- Packets may be presented to the node asynchronously. On receipt of a packet the node must wait for the next slot header. This *alignment* delay may vary between zero and (P_s + G) bits; and
- The node must wait for an empty slot. This delay will be governed by the number of concurrently active transmitters. The delay attributable to busy slots will be:

$$\text{BusyDelay} = (A - 1) * (P_s + G/N_S) \qquad \text{(A.7[1])}$$

Practical Limitations

In practice, slotted ring performance is affected by the relatively small number of slots per frame and the fact that real values of G and D do not normally correspond to integral numbers of slots. Furthermore, the presence of degenerate steady states can significantly affect performance. Cambridge Ring simulations reported in [Falconer 85a] have shown equation A.3 to represent the upper bound on SharedBw. In certain quasi-stable states the shared bandwidth decreases to:

$$\text{SharedBw} = \frac{\text{SysBw} * A}{A + N_S} \qquad \text{(lower bound)} \qquad \text{(A.8)}$$

A corollary of this observation, omitted from [Falconer 85a], is that there are quasi-stable states (prevalent as SharedBw approaches its upper bound) where the upper bound on access delay increases significantly to:

$$\text{BusyDelay} = (A - 1) * (P_s + G/N_S) * N_S \qquad \text{(upper bound)} \qquad \text{(A.9)}$$

---

[1]Equation A.7 is equivalent to equation 3.7 of [Hopper 78].

Quantum effects, observed at slot and frame boundaries, can affect the stability of MaxTxBw. Consider a node for which the value of D is slightly more than one slot. If the gap is small, then two slot headers will pass the node during this delay and the node will only transmit in one out of every $N_s + 3$ slots. If the gap is set slightly longer then, when the gap passes the node during the period of delay, the node may transmit in one out of every $N_s + 2$ slots. In some cases the electrical and gap lengths of rings have been *tuned* in order to improve the MaxTxBw experienced by particular node implementations.

A.5 Summary

Slotted rings exhibit a number of properties that have proven useful in the distributed computing and multi-service network environments. The relatively short packet length and the empty slot ATD access protocol mean that the ring medium is statistically multiplexed on a very fine per packet basis. Since a single node cannot monopolize the ring for longer than a single slot time, a lengthy block exchange between two hosts does not arbitrarily delay the exchange of data by other hosts. This has two implications:

- The ring can concurrently support various types of upper layer encodings with widely differing message lengths. In combination with the fine degree of multiplexing and low packetization delay afforded by the small packet size the ring is suited to multi-service applications; and
- The ring's response and low level retransmission mechanism considerably enhance its rate adaption capability. No ring bandwidth is consumed between packet transmissions, and so, a slow transmitting device can pause for significant and/or variable periods between packets. Similarly, by marking incoming slots busy, a slow receiving device can exert *back-pressure* on the rate of packet transmissions initiated by its peer.

Appendix B
Exchange Addressing

In the exchange architecture, the management services of each site maintain control over the CFR address space of the local exchange. One or more CFR addresses is statically assigned to each exchange node and the remaining addresses are available for dynamic assignment to the nodes of distant exchanges.[1] Dynamic assignments are made when associations with distant peers are initiated. The assigned address values remain valid until they are reclaimed by the exchange management and returned to the pool of unassigned addresses.[2]

Consider the case of portal P_1 at site A communicating with peer portal P_2 at the same site. Each of these portals is assigned a static address from site A's address space and these addresses can be written as P_1^A and P_2^A. When an association between the portals is initiated the local secretary service, S^A, provides each of the portals with the static address of its peer.

If the same portal P_1 attempts communication with P_3, located at site B[3], then the association establishment procedure is more complex. The addresses provided by a local secretary are always assigned from the site's own CFR address space and not from an address space shared with the distant site. S^A provides P_1 with a dynamic address, P_3^A, for communication with P_3. Similarly, S^B provides P_3 with a dynamic address, P_1^B for communication with P_1. Portal P_1 can now generate CFR packets that contain its own static address, P_1^A, in the source address field and the dynamic address, P_3^A, in the destination address field. The exchange window service must arrange for packets with this destination address to be extracted from the local CFR and transmitted along the appropriate channel. As the packets traverse the physical channel between the peer sites, they also cross the borders of

---

[1]Since there are relatively few nodes attached to each exchange, a substantial fraction of the 64K address space will remain available.

[2]Reclaimed addresses may be quarantined before being returned to the pool.

[3]At site B, the local address P_3^B is statically assigned to P_3.

the independent addressing domains. Accordingly, the exchange ramps supporting the inter-site channel translate the CFR address fields embedded within transmitted packets. When packets from P_1 are injected into the CFR at site B their source address fields contain the value P_1^B and their destination address fields contain the value P_3^B.

Address Space Management

In practice the scheme outlined above is somewhat difficult to implement. Management services at each site have to keep track of the individual addresses assigned to each of the distant portals and of the address mappings to be performed when site boundaries are crossed. The problem is reduced by partitioning every CFR address into separate *window* and *node* fields. In the present implementation each of these fields is eight bits long.

Every site's address space is divided into a number of windows each of which can be used to address all of the nodes at some site. Within a given site a unique node value is assigned to each exchange node and so the maximum number of nodes attached to a site's exchange is limited by the length of the node field. Local communication, between nodes at the same site, is performed by specifying the locally assigned node values together with the distinguished *local* window value.[1]

When communication between peer sites is initiated the management *of each exchange* creates a window through which CFR packets can be transferred to the peer site. At each site a portion of the CFR address is dynamically associated with the peer site. Windows are implemented through the dynamic assignment of window values. The addresses of a node at a distant site are specified using the node value statically assigned to the destination node (by the distant site) together with the window value dynamically assigned by the management of the local site.[2]

A window represents a path to a peer site and each path may be supported through the concatenation of a number of channels. Similarly, a single channel

---

[1]Although strictly speaking the only nodes that must be assigned addresses are the portals of the exchange it is convenient, for management purposes, to assign addresses to all nodes including those supporting ramps, management and other services.

[2]Each site assigns a window value from its own CFR address space without reference to the *reverse* window value assigned by the peer site. The assignment of window values to sites is not static or global and in the absence of traffic a window value can be reclaimed and assigned to support communication with some other peer site.

can concurrently support a number of windows.[1] The advantage of the window-based addressing strategy is that, when a single window value is assigned to a peer site, distinct CFR addresses are concurrently assigned to all of the peer site's nodes. It is expected that once a window to a site is established it will be used to support a number of concurrent associations between different pairs of nodes at the peer sites. The management and ramps of a site need only maintain address mapping information on a window-specific basis instead of maintaining separate entries for every association. The disadvantage of this strategy is that the present window scheme limits each exchange to a maximum of 256 nodes and 255 concurrently active windows. In a large exchange most of the CFR nodes are portals and it is unlikely that a single site will have more than two hundred local networks. Similarly, it is unlikely that the nodes of a single site will be concurrently accessing nodes at a large number of peer sites.

Address Notation

It is useful to define some notation for referring to address fields within CFR packets:

A: source ⇒ destination

describes a packet in circulation at site A. The source and destination address fields of the packet are separated by the ⇒ symbol. Within the context of the site A address space, the address of a local CFR node is expressed as:

A: node@local

where

node represents the node field value assigned to the node in question; and

local represents the distinguished window field value statically assigned to support local CFR addressing.[2]

The CFR address of this node at some distant site, B, is expressed as:

B: node^A@window^A

where

node^A represents the node field value assigned to the node by the management of site A. This value is used to refer to the node from any site; and

window^A represents the window field value dynamically assigned by site B to support direct access to nodes at site A.

---

[1]Provided the ramp maintains separate mapping entries for each window bound to the channel.

[2]Although different *local* values can be used at each site it is convenient if a common value is always used.

Address Assignment

The CFR nodes that attach portals to the local exchanges operate in *station* mode and their CFR addresses are read from switches. It is easy to arrange that, for each of these nodes, the window byte of the address is set to the *local* window value and the node byte of the address is unique within the context of the exchange. In this case, the CFR node logic will only accept packets with destination addresses of the form *node@local* and will only generate packets whose source addresses are of the same form.

Communication between two portals attached to the same exchange proceeds in a straightforward manner through the exchange of packets of the form:

A: $portal_1$@local ⇒ $portal_2$@local

and

A: $portal_2$@local ⇒ $portal_1$@local.

The only significant issue in local communication is that, prior to the exchange of packets, there must be some mechanism by which each portal can determine the CFR address of its peer. The exchange *secretary* service[1] is an intermediary that performs this function. A distinguished node value is assigned to the node supporting the secretary service so that a portal can always contact its local secretary by sending a packet of the form:

A: $portal_n$@local ⇒ secretary@local.

Communication between two portals at different sites is somewhat more complex. In order to supply a portal at site A with the dynamic address of a distant peer at site B the local secretary must first ascertain that an address window is assigned to support communication with the nodes of the peer site. When a window is available the local secretary can exchange packets with the peer secretary at the distant site and solicit its assistance in the out-of-band exchange of association specific parameters and CFR addresses.

Once the association is established the local portal will send packets of the form:

A: $portal_1$@local ⇒ $portal_2$@$window^B$

and receive packets of the form:

A: $portal_2$@$window^B$ ⇒ $portal_1$@local

When a window to a site is not immediately available the local secretary invokes the exchange *window* service which manages the site's address space. The window

---

[1]The secretary service is described in Chapter 10.

service assigns a window value to the distant site and arranges for the appropriate ramp to support the window.

Each ramp is attached to the exchange by a CFR node controller operating in bridge mode. When a ramp is configured to support a window, the 256 CFR addresses within the window are added to the list of destination addresses recognized by the receive logic of its bridge mode controller. The controller extracts from the CFR any packets whose destination addresses specify the window value. The ramp transmits the extracted packets to a peer ramp at the distant site, and the packets are injected into the peer exchange.[1]

One problem that arises is that $portal_2$ expects to receive packets of the form:

B: $portal_1$@$window^A \Rightarrow portal_2$@local

and send packets of the form:

B: $portal_2$@local $\Rightarrow portal_1$@$window^A$

where:

$window^A$ is the window value, assigned by site B's window service, to support communication with site A.

The window fields of CFR addresses must be mapped whenever packets cross site boundaries. In the exchange architecture this mapping is performed after a packet is extracted from a ramp's CFR and before it is transmitted to a destination ramp.[2] The source ramp is equipped with the mapping function when it is instructed to support the window. In the case of direct peer site communication described above, the mapping of destination addresses is straightforward: the window value is always replaced with the distinguished *local* window value. The mapping of source addresses is somewhat more complex. In order to correctly map the source address windows of every packet the ramp must be provided with the *reverse* window value assigned by the peer site.

A ramp that supports channels to more than one peer ramp must examine each extracted packet's destination address window field in order to route the packet to the appropriate peer ramp. It is claimed that the mapping of the address field can be performed in parallel with the routing decision and that locating the mapping operation in the source ramp has little or no impact on ramp performance.

---

[1]In practice the flow of packets between sites is bidirectional and whenever a site's window service establishes a window in one direction the peer site's window service will instruct its ramp to support a *reverse* window in the opposite direction.

[2]The destination ramp simply injects the packets it receives into the local CFR without further examination.

Relaying

In the previous example it is assumed that the ramps of the peer sites are linked by a fixed transmission facility or that a switched channel can be constructed when a window is opened. In some cases it may be sensible to route inter-site traffic via an intermediate *relay* site.[1]

At intermediate sites, the in-band relaying of CFR packets is automatically performed by the bridge mode CFR nodes of the ramps. The relay function consumes some of the site's CFR and common carrier bandwidth but the switching of relayed traffic is transparent to the intermediate site's portals. The main impact at the relay site is the additional out of band functionality required of the site's window service which must assign address windows and transmission resources to support relayed traffic.

Each of the communicating end sites allocates one window from its own address space to access nodes at the peer site. Each of these windows is bound to a channel that leads to the relay CFR at site R. This site assigns two window values, $window^{AB}$ and $window^{BA}$, to support the relay function. The AB window is bound to a channel that leads from site R to site B and the BA window is bound to a channel that leads from site R to site A. Packets arriving from site A will have their destination addresses mapped so that they carry the $window^{AB}$ value when they are injected into the CFR at site R. These packets circulate around the site R ring until they reach the appropriate ramp which extracts them and transmits them to site B.

In the A to B direction, a transmitted packet is of the form:

$$A: portal_1@local \Rightarrow portal_2@window^{B}$$

At site A, $window^{B}$ is supported by a channel to R and the packet is mapped into:

$$R: portal_1@window^{BA} \Rightarrow portal_2@window^{AB}$$

At site R, $window^{AB}$ is supported by a channel to B and the packet is mapped into:

$$B: portal_1@window^{A} \Rightarrow portal_2@local$$

In the opposite direction a transmitted packet is of the form:

$$B: portal_2@local \Rightarrow portal_1@window^{A}$$

---

[1]Relaying can take advantage of existing switched channels or of leased point-to-point facilities.

At site B, $window^{A}$ is supported by a channel to R and the packet is mapped into:

R: $portal_2@window^{AB} \Rightarrow portal_1@window^{BA}$

At site R, $window^{BA}$ is supported by a channel to A and the packet is mapped into:

A: $portal_2@window^{B} \Rightarrow portal_1@local$

Appendix C
The Experimental Programme

C.1 Apparatus

The apparatus used in the experiments was based on the equipment commissioned for the pilot exchange implementation. The principal objects subjected to measurements were: exchange CFRs; ISDN ramps; and Universe Portals. To simplify the experimental procedures all of the equipment was located at a single physical site within the Computer Laboratory. In some configurations the facilities of the circuit-switched ISDN were used to evaluate multiple site operation and relaying. Detailed evaluation of the ISDN was not undertaken as its contributions to delay and jitter are relatively limited.

Exchange CFRs
Using the notation introduced in Appendix A, the two CFRs used in the experiments are characterized by:

F_r = 40 Mbps, the operating frequency of the ring;
N_s = 1, the number of slots in the frame structure;
P_s = 38*8 = 304, the number of bits per slot; and
G, the length, in bits, of the gap between frames.

The *source* CFR was used in single hop experiments, between peer stations attached to the same CFR, and as the first hop of multiple hop experiments involving ISDN ramps. To improve the available system bandwidth this ring was manually configured to operate with a relatively short inter-frame gap, G, of only 24 bits. The gap of the *destination* CFR, used in multiple hop experiments, defaulted to 152 bits which corresponds to one half of a CFR slot.

Synthetic Portals

Synthetic portals were used to determine the basic point-to-point throughput of certain exchange configurations and to generate contention traffic during delay measurement intervals. These portals rely on the same hardware configuration, UDL/CFR driver, and Tripos kernel used by the Universe Portals. Their portal-specific software generates and sinks a synthetic workload. The generated traffic pattern is defined in terms of the intervals between UDL blocks and the length of the individual blocks that are exchanged.

Delay Probe

A modified version of the VME-based exchange interface card was used to measure the delay associated with the forwarding of CFR packets. Each interface card provides access to two CFR station cards which may be attached to different exchanges. Using appropriate software the VME host arranges for a packet to be transmitted between the two stations. During each packet transfer the modified interface card captures two interval measurements which are subsequently recorded by the host software.

The interval timers are both started when packet transmission commences, a condition distinguished by the negation of the *Transmit FIFO Available* (TFA) signal at the source's CMOS station chip. The *transmit* timer, triggered by the reassertion of TFA, measures the interval during which the source station attempts packet transmission. The *path* timer, triggered by the assertion of the *Receive FIFO Available* (RFA) signal at the destination's CMOS station chip, measures the total transit time between source and destination.

When the source and destination stations are located on the same CFR the two intervals differ by the period, after packet reception, during which the packet completes its ring revolution and is inspected at the source station. When the two stations are located on peer exchanges the intervals measured will differ substantially. The path timer measures the time taken to travel along the complete path between the peer stations, however, the transmit timer only accounts only for the first *hop*, i.e., the time taken to transmit the packet to the ramp that extracts it from the source CFR.

C.2 Procedures

Many of the experiments investigate the performance of a particular path between two exchange points of attachment by measuring the throughput achievable between a pair of peer portals or the delay between the source and destination stations of the delay probe.

C.2.1 Throughput Measurements

Basic throughput was measured, in the absence of other traffic, by gradually increasing the load offered through a source portal and noting the point at which congestion results in packets failing to reach their destination.

Throughput measurements are reported in units of *kilopackets per second* (KPPS) which denote the number of CFR packets transferred per unit time. A more conventional measure of throughput, in bits per second, can be derived, when the number of upper layer symbols encoded in every packet is known.[1]

C.2.2 Delay and Jitter Measurement

The delay probe is used to measure how particular exchange configurations respond to single packet impulses. For a given configuration the delay measurement is repeated a number of times to determine the average absolute delay and to produce a histogram, or *jitter spectrum*, depicting the variation in delay. Additional relative measures are based on the width of the delay bounds applicable to a given fraction of the observations. For example, the 95% jitter margin is the range within which 95% of all observations fall. The maximum jitter margin is the difference between the maximum and minimum delay measurements.

The period between successive delay measurements is relatively long to permit the exchange components to return to their quiescent states. This *single shot* approach does not provide a comprehensive measure of a configuration's delay response but it has proven to be a simple experimental technique that yields useful insight into the performance and operation of exchange components. Furthermore, when the experiment is performed in the presence of contention traffic generated by synthetic

---

[1]For example, at the M-Access layer every packet carries 32 octets of upper layer symbols.

portals, the impulse response is indicative of the incremental performance that could have been achieved by those portals.

The interval timers of the delay probe are calibrated in units of 1.6 microseconds (usec) with a measurement accuracy of +.8 usec and -3.3 usec. The individual interval measurements were rounded to the nearest usec prior to statistical analysis and the production of the delay histograms. These integral units are used in the remainder of the text without further reference to the error bounds on the individual or derived measurements.

The absolute measures of delay and jitter can be normalized with respect to the inverse of the packet throughput of a particular symbol stream. The resultant values, expressed in *packet intervals* (PI) are a convenient measure of the relative impact of the network on particular styles of communication. For the purposes of delay analysis these unit intervals express the delay arising within the network as a multiple of the packetization delay absorbed at a source point of attachment. From a jitter perspective the packet interval measure is indicative of the extended length, in packets, of an elastic buffer implementing jitter compensation at the destination point of attachment.

Appendix D
CFR Performance: Analysis and Results

This appendix presents an analysis of the CFR performance observed by a particular symbol stream. The first section of this appendix develops and validates a basic performance model that predicts the response of the CFR to a single packet impulse.[1] The second section describes how contention elements, arising from parallel symbols streams, affect the observed performance of the exchange.

D.1 Basic Performance

Figure D.1 is a three stage queueing network that models the impulse response of the CFR. The first stage is a delay centre that corresponds to the loading of packets into the source CFR station. The next stage of the model is an alignment *gate* that delays a loaded packet until the arrival of the next CFR slot header. The final stage corresponds to the interval during which the packet is injected into the ring and circulates around it.

The CFR model is not internally pipelined and each packet is sequentially processed by the cascaded stages of the network. The delay attributable to the

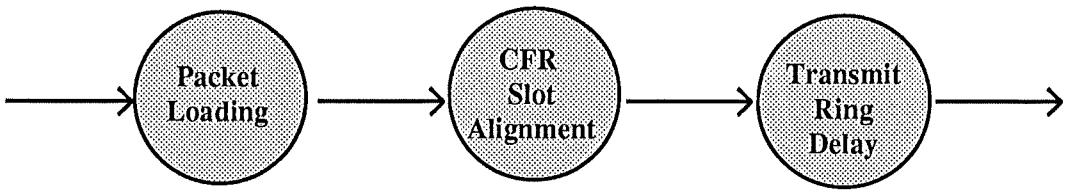

Figure D.1: Basic CFR Model

[1]Where appropriate, this appendix includes results derived from the experimental programme described in Appendix C. Various aspects of the CFR design are described in Chapter 7, Appendix A, and [Hopper 86].

unloading of packets is excluded from the CFR model and is accounted for within the next downstream stage of the upper level model. This exclusion arises from the observation that the unloading of a packet, at the destination, can overlap the loading of a subsequent packet at the source. In contrast, packet loading delay is accounted for within the CFR model even though it is largely determined by the upstream packet source. The inclusion of this stage is dictated by the design of the CMOS station chip which does not support the concurrent loading and transmission of packets.

There is a slight ambiguity in the transmission stage of the CFR model. A discrepancy arises because the present model does not allow for the potential overlap between the transmission and downstream unloading of a single packet. From the perspective of the transmitting station a packet is resident at this stage for one frame delay period after is injected into the ring. From the perspective of the receiver the transmission is completed as soon as the packet arrives at the recipient station. In this case the time spent circulating around the ring is dependent on the relative position of the peer stations and ranges from a few bytes, for adjacent stations, to a complete frame delay, when the transmitter is sending to itself.

The CFR model can be substituted into a variety of upper level models using one instance of the lower level model to represent each *hop* across an exchange switch fabric. In the basic model of Figure 11.1 separate instances of the CFR model are tailored to describe the behaviour at the source and destination exchanges. The dominant parameter of each model is the packet loading time, $R_{loading}$, which is a function of the packet source. At the source exchange $R_{loading}$ is determined by the exchange interface component of the source portal and at the destination exchange $R_{loading}$ is dependent on the receive half of the ramp.

Delay and Jitter

When a single packet impulse is applied to the CFR model the queue of packets waiting to enter service will be empty and so the CFR packet residence time, R_c, is the sum of the residence times at each of the three stages of the lower level model[1]:

$$R_{CFR} = R_{loading} + R_{alignment} + R_{transmission} \qquad \text{(D.1a)}$$

[1]Strictly speaking, the pdf of R_c is the convolution of the pdf's at the individual stages.

Since there is no internal queueing within the lower level model the residence time of each stage is the same as its service time. From the perspective of a transmitting station attached to the source CFR:

$R_{alignment}$ = up to one frame period
= 0 to $(P_s+G)/F_r$
= 0 to 8.2 usec

$R_{transmission}$ = one slot period + one frame period
= $(P_s+G)/F_r + P_s/F_r$
= $(2*P_s+G)/F_r$
= 15.8 usec

For the portals used in the experiments, $R_{loading}$ is a fixed delay of 22 usec (±1 usec). This values was obtained by attaching oscilloscope probes to the user bus of a portal's CFR station and monitoring the loading process.

The residence times can be characterized in terms of basic delay components and superimposed jitter components. The loading and transmission stages, which do not contribute to the variation in delay, have fixed delay components and null jitter components. The alignment gate, which is the principal source of jitter, contributes a null delay component and an 8.2 usec jitter component. On this basis the total residence time, R_{CFR}, has a delay component of about 38 usec with a superimposed jitter of 8.2 usec. This characterization is somewhat simplistic since it describes the jitter range but not its density. In the case at hand the impulse arrivals are asynchronous with respect to the CFR slot structure. The jitter component is evenly distributed across all packets and so the average residence time is 42 usec (±1 usec).[1]

The delay probe was used to validate the last two stages of the source CFR model. The probe's transmit and path timers measure the cumulative residence time from the perspectives of the transmitting and receiving stations respectively. In Figure D.2 histograms are used to depict the frequency with which individual

[1]Graphically, $R_{loading}$ and $R_{transmission}$ can be represented by unit area impulses that are displaced from the origin by 22 usec and 15.8 usec respectively. The jitter component arising from $R_{alignment}$ can be represented by an 8.2 usec pulse of unit area positioned at the origin. The convolution of the three components is an 8.2 usec pulse of unit area that is offset from the origin by 37.8 usec.

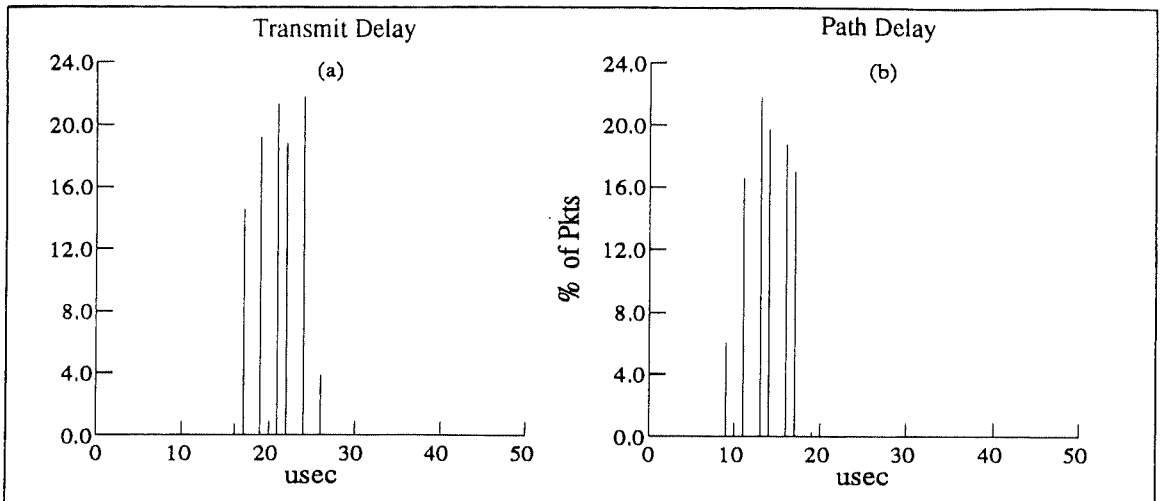

Figure D.2: Basic CFR Response

residence times were observed.[1] Given the limited accuracy of the apparatus, the transmit timer's delay range of 16 to 26 usec is consistent with the model's 15.8 usec delay component and 8.2 usec jitter component. The average of the transmit delay measurements is 21 usec versus an expected average delay of 19.9 usec.

Throughput

The nominal throughput of the CFR model is the inverse of the average residence time. From the perspective of a portal attached to the source CFR the derived throughput is 1/42 usec, or 23.8 KPPS[2]. This nominal throughput is based on the delay experienced when single packet impulses are transmitted across the CFR.

When a large burst of packets is presented for transmission the complete transmission cycle becomes synchronized with the frame structure of the ring. The alignment interval, which previously reflected the asynchronous nature of packet arrivals, becomes a fixed delay that expands the total residence time into an integral multiple of the frame period. For some value of D, corresponding to the number of unused CFR slots between transmissions:

$$R_{loading} + R_{alignment} = (D * (P_s+G) + G) * (1/F_r)$$

and accordingly:

[1]Figure D.2(a) is the *transmit* delay seen by the transmitter, and Figure D.2(b) is the *path* delay measured at the receiver.

[2]Kilopackets per second.

$$R_{CFR} = (D * (P_s+G) + G + (P_s + G) + P_s) * (1/F_r)$$
$$= ((D + 2) * (P_s+G)) * (1/F_r) \qquad (D.1b^1)$$

For the Universe and synthetic portals, which have a 22 usec packet loading delay, the value of D at the source CFR is 3 slots and Equation D.1b evaluates to 41 usec. This corresponds to a burst throughput of 24.4 KPPS. Note that for a given packet loading time and clock frequency the value of G can be adjusted to minimize the residence time. In effect, a ring can be *tuned* so as to increase the point-to-point throughput available from a given packet source at the expense of a reduction in the overall system bandwidth.

An experiment was conducted to measure the throughput between peer synthetic portals exchanging large UDL blocks across the source CFR. There was no evidence of CFR back-pressure being exerted by the destination portal and this implies that the source portal's ability to transmit packets is the principal bottleneck. The maximum measured throughput of 22.6 KPPS, which is tempered by portal delays arising at UDL block boundaries, is in close agreement with the predicted value.

Destination CFR

The delay at the destination CFR can be derived from the maximum throughput available at the CFR interface of the receive ramp half. To measure this throughput the experimental equipment was configured in a two hop configuration with a high bandwidth channel between the peer ramps. A synthetic portal was used to generate packets which were routed over the source exchange, across the channel to the receive ramp half, and then over the destination exchange to a peer synthetic portal. The rate of packet generation was gradually increased until the ramp's ability to transmit on the CFR became a bottleneck.

The ramp's CFR queue begins to overflow at a throughput of 3.64 KPPS. This value represents the maximum throughput at the destination CFR and corresponds to a modelled residence time of 275 usec. Clearly this delay is dominated by the performance of the packet loading stage implemented within the ISDN ramp.

---

[1]Equation D.1b corresponds exactly to the inverse of the previously derived maximum transmission bandwidth, MaxTxBw, after substitution of N_s=1 into Equation A.5 and multiplication by P_d to convert from bits per second to packets per second.

Jitter at the destination CFR arises from jitter at the packet loading stage within the ramp and alignment jitter at the CFR itself. The alignment component is evenly distributed over the range 0 to $(P_s+G)/F_r$ which, for the destination CFR, evaluates to a maximum jitter contribution of 11.4 usec. The packet loading jitter was not measured, however, both the alignment and packet loading components are believed to be relatively small in comparison to other jitter effects induced at the ramps.

D.2 Contention Analysis

Slotted ring access protocols have been represented using a variety of modelling techniques. The available models emphasize the general analysis of overall ring throughput and the access delay experienced at transmitting stations. In [Zafirovic 88], vacation intervals are used to account for the empty slots that are passed between active CFR stations. In the exchange environment, however, this loss of shared bandwidth is not as significant as the contention effects that can arise at receiving stations. In [King 82], a network of independent multi-class queues is used to model the reception of Cambridge Ring blocks at each station. Although this model provides some insight into the general effects of receiver contention, it does not support the jitter analysis of a particular path across a CFR. The following subsections describe a CFR model and analysis techniques that support the particular analysis of both empty slot and receiver contention effects. A subset of these techniques will be applied to the configurations studied in the experimental programme.

D.2.1 Contention Elements

Figure D.3 illustrates the overall traffic along the path between a particular pair of CFR stations. The primary inputs, which arise at the transmitting station are: P_i, the traffic of interest; P_c, representing parallel contention traffic between the peer stations; and P_b, the background traffic destined for other stations. The secondary inputs, which arise at the recipient or at third party stations are: S_c, contending traffic destined for the same station as P_i; and S_b,

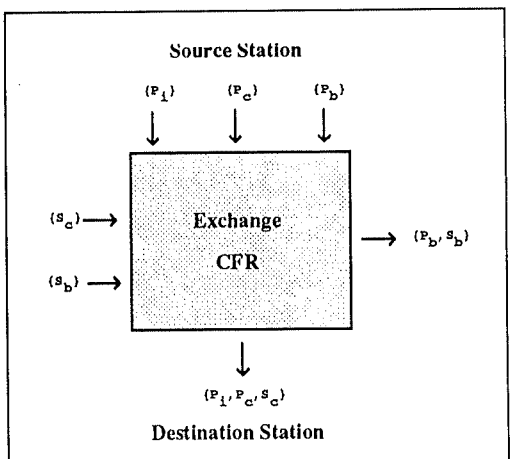

Figure D.3:
CFR Contention Elements

background traffic destined for other stations.

The appropriate substitutions are made in order to parametrize the CFR contention elements to represent either the source or destination exchanges of the upper layer contention model of Figure 11.2. At the source exchange, for example, P_c and S_c encompass all of the traffic through the source ramp, and so:

$$P_c = \{SP_{DP}, SP_{DC}, SP_{SR}\};$$
$$S_c = \{SC_{DP}, SC_{DC}, SC_{SR}\};$$
$$P_b = \{SP_{SC}\}; \text{ and}$$
$$S_b = \{SC_{SC}\}.$$

D.2.2 Hierarchical Decomposition

Hierarchical modelling techniques facilitate the separation of the primary and secondary contention elements. The primary inputs contend for the transmission capacity available at a particular transmitting station. This contention is represented by an intermediate level model (Figure D.4) consisting of a single stage, multiple class, queueing centre. The customers of this open service centre correspond to individual packets and their class dependent arrival processes are determined by the primary contention elements. The queueing delay experienced by the customers corresponds to the jitter component introduced by these contention elements.

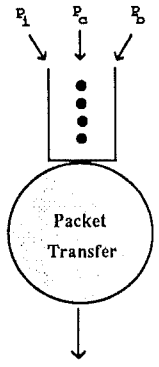

Figure D.4: Intermediate Model

The service time experienced by each class of arrivals is derived from a lower level model that characterizes the transfer of a single packet to a selected recipient. A separate instance of this model is parametrized to determine the service time distribution of each recipient or traffic class. The secondary contention elements, introduced at other ring stations, increase the effective transfer delay experienced by individual packets. The increase in the delay experienced by a particular symbol stream corresponds to the jitter component induced by the secondary contention elements.

The service discipline of the intermediate model is dependent on the queue management strategy implemented at the exchange interface of the transmitting portal or ramp. The operation of an expedited transfer function can be modelled

through the division of each service class into high priority and low priority sub-classes and the use of a nonpreemptive priority queueing model. Given a set of class-dependent arrival processes and service time distributions, the intermediate model can be analyzed to determine the queue length and residence time distributions. Since there are a number of known techniques for the solution of multiple class and priority servers[1], the remainder of this section concentrates on the development, analysis, and validation of the lower level model.

D.2.3 Packet Transfer Model

The transfer model, illustrated in Figure D.5, is an extension of the basic CFR model incorporating additional delay centres that account for the secondary contention elements. In the absence of secondary traffic, the first ring slot following an alignment interval is always empty. However, in the presence of S_c and/or S_b contention elements some fraction of the slots will be filled by other stations on the ring. The delay experienced as a result of this *slot contention* is modelled by the empty slot delay centre of the transfer model.

Similarly, in the presence of S_c and/or P_c contention traffic, a packet may be transmitted before the receiving station has completed the unloading of a previously received packet. In this case, the recipient will exert back-pressure on the transmitter by rejecting the packet. The transmitter will delay for a fixed retry interval and then retry the transmission. In the transfer model, a feedback path is used to model this *receiver contention* effect. A fraction of the transmission stage customers, representing rejected transmissions, return to the empty slot stage

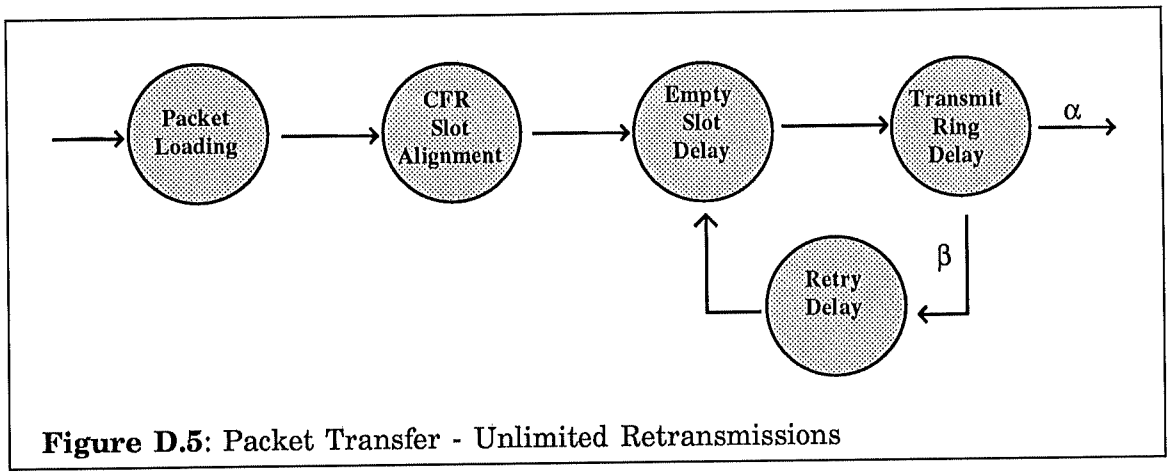

Figure D.5: Packet Transfer - Unlimited Retransmissions

[1]Introductions to multi-class server analysis can be found in [Lazowska 84] and [Kleinrock 76].

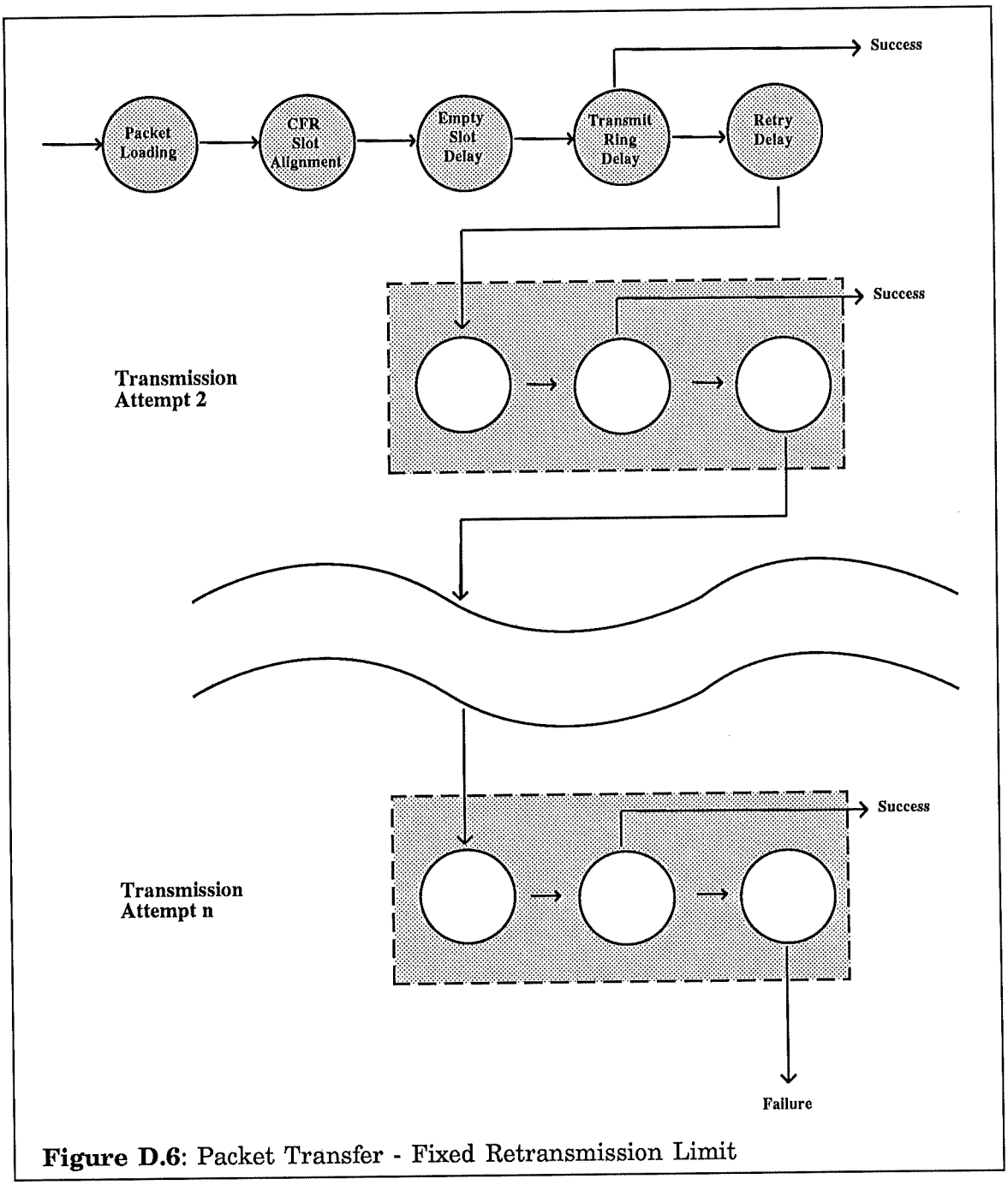

Figure D.6: Packet Transfer - Fixed Retransmission Limit

via the retry delay centre. In practice, the CFR implementation imposes a maximum limit on the number of transmission attempts and the packet transfer is aborted if this limit is exceeded. This arrangement can be modelled through replication of the appropriate delay centres as illustrated in Figure D.6.

D.2.4 Slot Contention: Analysis and Measurements

Slot contention delays are analogous to the collision delays of the slotted Aloha transmission protocol.[1] The probability of being delayed during a given slot interval can be evaluated by treating each slot as a *vulnerable period* during which a packet waiting to be transmitted can collide with a full packet generated by another ring station.

The delay experienced by a given transmission is quantized into an integral number of vulnerable periods[2] corresponding to the number of consecutive collisions. This delay can be represented by a train of impulses beginning at the origin and separated in time by the vulnerable period.[3] For n >= 0, the area of the n^{th} impulse is given by:

$$P[0\ \text{delays}] = (1 - P_0[\text{collision}]) \qquad \text{(D.2)}$$

and

$$P[n\ \text{delays}] = (1 - P_n[\text{collision}]) \{ \prod_{i=0}^{n-1} (P_i[\text{collision}]) \}$$

where:

P_n[collision] is the probability of being delayed by a collision during the n^{th} period.[4]

If the aggregate traffic on the ring[5] is represented by a single Poisson process, with an arrival rate of A packets per vulnerable period, then for all n:[6]

---

[1]Descriptions of Aloha analysis techniques and references to the original works can be found in [Schwartz 77] and [Kleinrock 76]. In contrast to Aloha, slot collisions are non-destructive and do not induce additional traffic.

[2]For a single slot ring the vulnerable period is one ring revolution.

[3]In the absence of other contention elements, convolution with the basic CFR delay leads to a train of pulses that are: one ring revolution wide; contiguous with each other; and have the same area as the corresponding impulses.

[4]One of the participants in each collision is successful and is not delayed.

[5]The aggregate traffic includes retry attempts arising from receiver contention.

[6]The Poisson approximation is somewhat optimistic as the stations delayed by an empty slot *collision* are persistent, i.e., they remain active and contend for the next slot. Similarly, the approximation (which is based on an infinite number of traffic sources) implies an unbounded number of collisions. In practice, the number of active sources is finite, and the worst case delay is bounded in accordance with Equation A.9 of Appendix A.

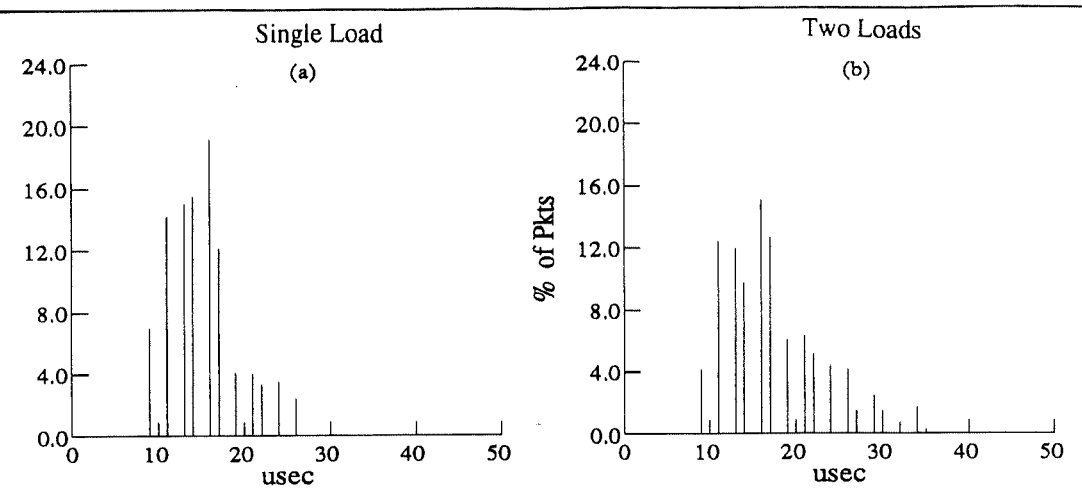

Figure D.7: Empty Slot Contention

$$P_n[\text{collision}] = 1 - e^{-A} \qquad \text{(D.3a)}$$

and so:

$$P[\text{n delays}] = (\, e^{-A} \,)(\, 1 - e^{-A} \,)^n \qquad \text{(D.3b)}$$

The experimental programme included a number of single hop experiments that measured the effect of empty slot contention on the impulse response observed by the delay probe. Figure D.7(a) and Figure D.7(b) are the jitter spectra obtained in the presence of an S_b contention element generated at one and two portals, respectively. Through comparison to the contention-free spectrum (Figure D.2(b)) it can be seen that some fraction of the impulse transmissions were delayed by one ring revolution and, in the presence of two contention sources, a smaller fraction of the packets were delayed by two ring revolutions.

Table D.1 compares the measured collision densities with the values derived using the Poisson approximation. For these experiments, the assumption of Poisson arrivals is somewhat questionable, given the limited number of traffic sources and their deterministic arrival patterns. An alternative analysis which takes account of the actual traffic environment of the experiments,

No. Of Traffic Sources (s)	Arrivals/ Vulnerable Period A = (s)*(s)		Successful Transmission P[n retries]			
			0	1	2	>3
1	.185	Measured	.83	.17	---	---
		Deterministic	.815	.185	---	---
		Poisson	.83	.14	.02	.01
2	.37	Measured	.66	.26	.08	---
		Deterministic	.665	.300	.035	---
		Poisson	.690	.215	.065	.030

Table D.1: Empty Slot Delay Density

has been developed and the results of this *deterministic* analysis[1] are included in Table D.1. Given the limited accuracy of the measurements, the experimental results are consistent with the predicted values.

D.2.5 Receiver Contention: Analysis and Measurements

The jitter induced during packet transfer can be decomposed into distinct components associated with each of the exit points of Figure D.6. For each transmission attempt it is necessary to determine the probability of the *success* exit being taken and the cumulative jitter up to the exit point. The cumulative jitter is represented by the convolution of all of the retry, empty slot, and transmission delays not accounted for within the basic CFR model. One immediate observation is that each of these constituent delays is bounded, and so, the worst case jitter, induced by the maximum number of retries, is also bounded.

The Aloha analogy can be applied to the analysis of receiver contention in order to determine: P[m retries], the probability of a successful outcome at the m^{th} transmission attempt; and P[>max retries], the residual probability of failure following the maximum number of transmission attempts.

---

[1]Let a represent the frequency of each periodic traffic source, measured in arrivals per vulnerable period. In this analysis, A, the aggregate traffic from all sources, is assumed to be less than one.

Arrivals from contending sources may cause a random probe to be delayed during its first vulnerable period. In the event of an initial collision the probe will contend for subsequent slots until it is successful. If collisions at these slots are resolved fairly, then the probe is only delayed by half of any additional collisions.

When one source is active only one collision is possible and so:

$$P_0[\text{collision}] = P[\text{arrival}] = a$$

and

$$P_n[\text{collision}] = 0 \quad (n > 0)$$

When two sources are active, either one or both of the sources may be involved in an initial collision. In the event of simultaneous arrivals, the *losing* source will remain active and participate in a second collision during the next slot. In the case of a lone arrival, the probability of a second collision is dependent on the arrival rate of the remaining source. On this basis:

$$P_0[\text{collision}] = 2 \times P[\text{each arrival}] - P[\text{two arrivals}] = 2a - a^2$$

$$P_1[\text{collision}] = (a^2 + a(1 - a^2)) / 2$$

and

$$P_n[\text{collision}] = 0 \quad (n > 1)$$

The analysis is considerably simplified in cases where the overall ring utilization is relatively low. In the absence of empty slot contention delays, the intervals between successive transmission attempts are statically determined by the retry and transmission delays. The jitter arising from receiver contention can be represented by a train of impulses beginning at the origin and separated by the retry period. The area of each impulse corresponds to P[m retries].[1]

The receiver contention *vulnerable period* corresponds to the unloading interval following the reception of a packet. When this interval is less than the time between retries, P[m retries] corresponds to P[n delays] of Equation D.2. When the vulnerable period is greater than the time between retries, the retry attempts are lumped into *collision groups*.[2] In this case, P[n delays] is the probability of a successful outcome within the n^{th} collision group. This probability can be pro-rated to determine the corresponding value of P[m retries] for each of the transmission attempts within the group.[3]

The experimental programme included a number of single hop experiments that measured the effect of receiver contention on the impulse response observed by the delay probe.[4] In each of the experiments, two synthetic portals generated periodic contention traffic destined for a target receiver. Since the effects of contention are dependent on the packet unloading characteristics of the receiver, the experiments were performed using two different targets: a synthetic portal; and an ISDN ramp. The resultant jitter spectra are presented in Figure D.8[5] and Figure D.9, respectively.

---

[1]Convolution of these impulses with the basic CFR delay leads to a train of pulses that are: one ring revolution wide; periodically spaced in accordance with the retry period; and have the same area as their generating impulses.

[2]The number of retries in each group corresponds to the number of transmission attempts per vulnerable period.

[3]Clearly, P[0 retries] = P[0 delays].

[4]The probe was configured to support a maximum of eight transmission attempts with a single slot delay between successive attempts. In the absence of empty slot contention the total period between the initiation of successive attempts will be two ring revolutions. At the source CFR this configuration corresponds to 16.4 usec between transmission attempts and a worst case jitter of 114.8 usec.

[5]Figure D.8(a) is the basic response in the absence of contention traffic.

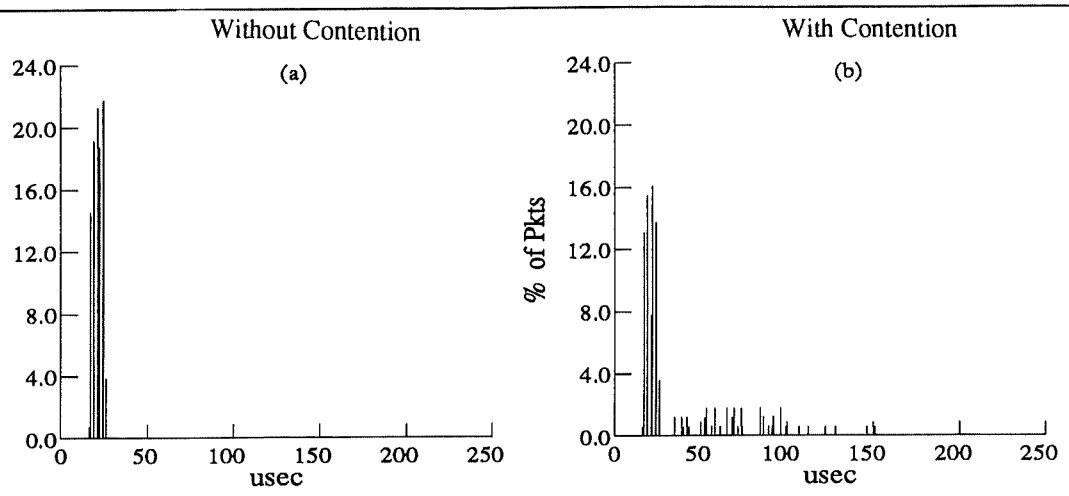

Figure D.8: Receiver Contention at a Portal

Contention at Portals

In the portal experiment, the packet unloading time at the target receiver was measured and found to be seven ring revolutions. The retry interval of the probe is only two revolutions and so the experimental configuration can be modelled using two collision groups with 3.5 transmission attempts per group. The deterministic and Poisson approximations were used to predict P[n delays] and the results pro-rated to derive P[m retries] and P[>max retries].

No. Of Traffic Sources (s)	Arrivals/ Vulnerable Period A = (s)*(a)		Successful Transmission P[n retries]								Failure P[Max Retries]
			0	1	2	3	4	5	6	7	
2	.32	Measured	.70	.05	.07	.08	.06	.02	.01	.01	---
		Deterministic	.710	.074	.074	.074	.042	.009	.009	.009	---
		Poisson	.726	.057	.057	.057	.037	.015	.015	.015	.021

Table D.2: Retransmissions to a Portal

The analytical and experimental results are given in Table D.2. For the most part, the measurements are consistent with the expected values and the two collision groups are easily identified. One apparent discrepancy is that the measured values are somewhat lower than expected at n = 1 and somewhat higher than expected at n = 4. This observation is explained by the presence of empty slot contention within the measured system.[1]

[1]Consider the case where the receiver is idle and the probe experiences slot contention during its first transmission attempt. The offending traffic source will seize the receiver until the 3rd or 4th transmission attempt.

Contention at Ramps

The packet unloading time at the ISDN ramp is estimated to be twenty-six ring revolutions. This interval is sufficiently long that all of the retries are contained within a single collision group. The experimental results and the predicted values for three different levels of contention traffic are presented in Table D.3.

No. Of Traffic Sources (s)	Arrivals/ Vulnerable Period A = (s)*(a)		Successful Transmission P[n retries]								Failure P[Max Retries]
			0	1	2	3	4	5	6	7	
2	.166	Measured	.86	.01	.01	.01	.01	.01	.01	.01	.07
		Deterministic	.840	.012	.012	.012	.012	.012	.012	.012	.078
		Poisson	.847	.010	.010	.010	.010	.010	.010	.010	.083
2	.338	Measured	.76	.03	.02	.03	.02	.01	.02	.01	.11
		Deterministic	.691	.021	.021	.021	.021	.021	.021	.021	.159
		Poisson	.713	.016	.016	.016	.016	.016	.016	.016	.176
2	.666	Measured	.76	.03	.05	.04	.02	.01	.02	.02	.07
		Deterministic	.445	.034	.034	.034	.034	.034	.034	.034	.317
		Poisson	.514	.019	.019	.019	.019	.019	.019	.019	.351

Table D.3: Retransmissions to a Ramp

For A = .166 and A = .338, the results are consistent with the expected values, however, there is considerable discrepancy at A = .666. The results suggest that the ramp's packet unloading interval is load dependent. If the interval decreases at certain loads, then the effective value of A declines and P_n[collision] increases accordingly. The experimental results suggest that the interval drops below 14 ring revolutions and a second collision groups comes into play.

Contention Under Heavy Load

In the experiments described above, the synthetic portals generated single packet arrivals on a periodic basis. In many applications, such as the Universe Portals, arrivals from the client network are actually bulk arrivals corresponding to multi-packet blocks. The traffic generated by the portal will consist of bursts of packets, separated by idle periods.

A second traffic source, contending for the same receiver as the portal will observe a bi-modal jitter distribution. During idle periods the jitter will be governed by the basic jitter components of the CFR, however, during bursts the traffic source must contend with the load generated by the portal. The jitter experienced during bursts is aggravated by the following considerations:

- During each burst the portal attempts successive packet transmissions as quickly as possible. A single

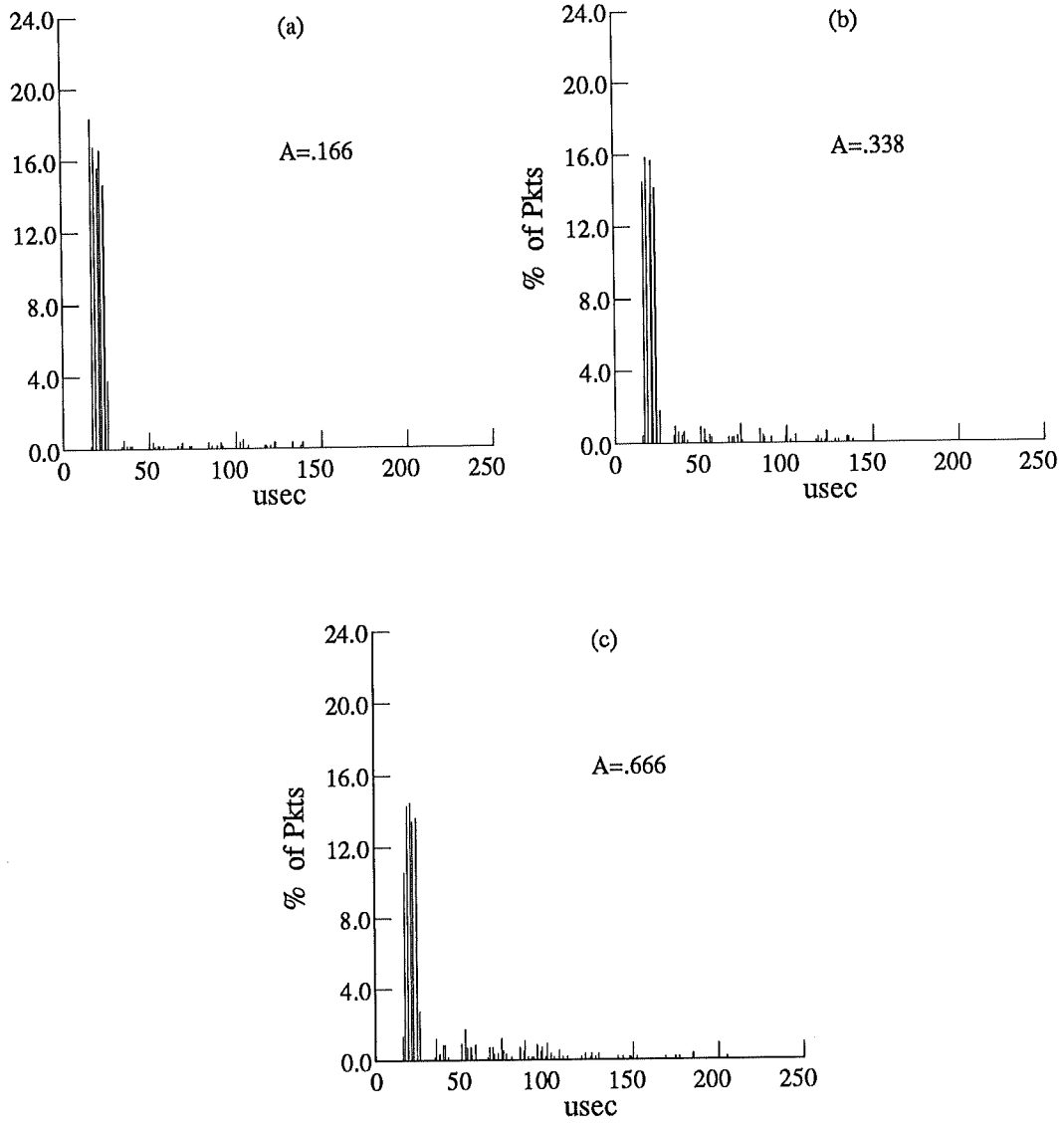

Figure D.9: Receiver Contention at a Ramp

portal may generate a burst load that exceeds the capacity of the receiver;

- The CFR slot protocol does not ensure the fair resolution of receiver contention collisions;[1] and
- The deterministic retry policy permits quasi-stable operating states in which the system may discriminate

[1]The empty slot protocol ensures the fair allocation of overall ring bandwidth, however, it does not partition the bandwidth available at a given receiver. Receiver contention is not unique to the CFR, however, few networks provide for the detection of collisions let alone their fair resolution. The response mechanism embedded within the CFR protocol supports the detection of collisions and the implementation of a retry strategy. Other networks, such as Ethernet, Orwell, and QPSX, lack this response mechanism and rely on upper layer protocols to detect and recover from collisions. This policy can lead to a serious degradation in service as P_0[collision] increases.

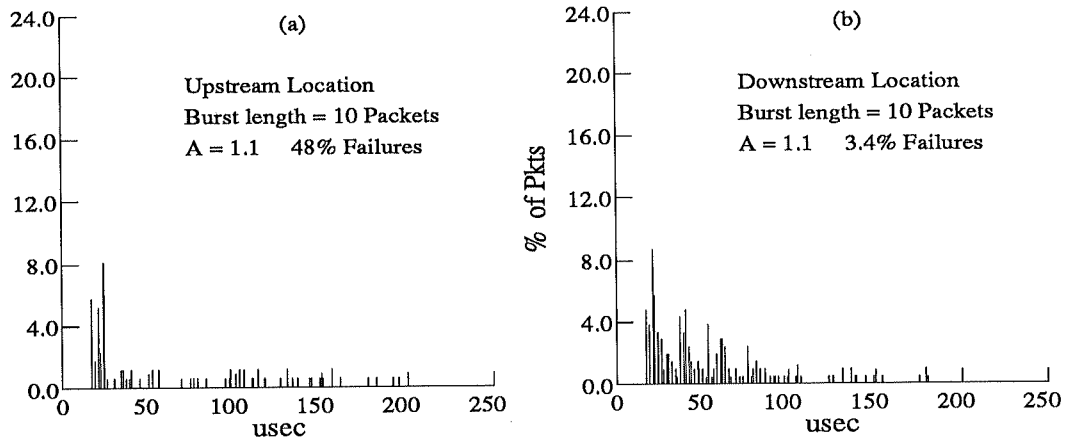

Figure D.10: Receiver Contention Under Burst Load

amongst traffic sources on the basis of their relative positions in the ring.[1]

A consequence of these considerations is that, for the duration of a burst, a single contention source can prevent other traffic sources from accessing a receiver.[2]

Figure D.10 illustrates the impact of relative ring location on receiver contention jitter. In both cases, two synthetic portals were used to generate burst arrivals of 10 packets each. Figure D.10(a) is the jitter spectrum obtained when the probe was positioned immediately upstream from the receiver, i.e., between the contention sources and the receiver. Figure D.10(b) is the complementary result with the probe positioned downstream from the receiver.

The most significant effect of unfair collision resolution is that it dramatically increases the residual probability of a transmission failing after the allowed number of transmission attempts. Experiments were conducted to investigate this effect and Table D.4 summarizes the results obtained for a number of different burst sizes.

---

[1]Following a successful transmission, a receiving station is busy for some number of slot intervals during packet unloading. In the absence of empty slot contention, the station just downstream of the receiver will have the first opportunity to seize the first slot that will be eligible for reception.

[2]Consider the case where a portal is *tuned* so that the intervals between packet transmission match the packet unloading delay at the receiver. The initiation of successive transmissions will be timed to seize the first slot that will be eligible for reception. A traffic source that is between the receiver and the portal may seize this slot before it arrives at the portal. However, a traffic source that is between the portal and the receiver, will always find the slot busy and will be prevented from accessing the receiver.

In the present exchange environment, two techniques can be used to limit the impact of bursty contention elements:

- The retry mechanism within the CFR hardware can be supplemented by a software retry strategy. The software strategy comes into play following the maximum number of automatic transmission attempts. The software approach effectively increases the total number of transmission attempts and introduces some variability into the delay interval between retries. The presence of non-deterministic delays may trigger the collapse of quasi-stable states; and
- Traffic sources can be instructed to *throttle* their transmissions to specific receivers. This approach imposes an artificial limit on the *duty cycle* of packet transmissions within individual bursts. The bandwidth available at individual receivers can be allocated by an exchange management service that assigns appropriate throttle values to each traffic source.[1]

A more sophisticated collision resolution mechanism is a desirable objective for future exchange switch fabrics. One problem with the slotted ring protocol is that successful transmitters are not aware that collisions are taking place. The present response field could be extended to advise a successful transmitter that a collision has recently been observed. The transmitters accessing a receiver could use this information to dynamically adjust their transmission *throttles* to suit the overall load on the receiver. This simple mechanism should reduce the frequency of collisions, however it does not ensure their fair resolution. A more elaborate scheme, described in [Hopper 78], allows each receiver to control the order of transmissions arising from different sources.

Burst Length (Packets)	Aggregate Offered Load (Arrivals/ Vulnerable Period	% Failure Upstream Of Receiver	% Failure Downstream From Receiver
1	0.332	0.3	0.0
2	0.462	1.1	0.5
4	0.713	9.2	7.1
6	0.870	32.2	20.9
10	1.1	48.0	3.4
147	1.85	84.4	8.0

Table D.4: Upstream vs Downstream Position

[1]Both of the techniques suggested here can be combined with a queue management discipline that supports the interleaved transmission of packets associated with different receivers.

Appendix E
ISDN Ramp Performance: Analysis and Results

This appendix presents an analysis of the ISDN ramp performance observed by a particular symbol stream.[1] The first section of this appendix develops and validates a basic performance model that predicts the basic impulse response of the peer ramp halves.[2] The second section describes how contention elements, arising from parallel symbols streams, affect the observed performance of the ramp halves.

E.1 Basic Ramp Performance

The transmit half components of a *source* ramp and the receive half components of a *destination* ramp jointly support the directional flow of packets along an inter-site channel. In this section, independent models are used to analyze the delay and jitter induced at each of the peer ramp halves.

In order to validate the basic performance models, the peer ramp halves were joined by a *null* transmission channel. This configuration can be modelled through the concatenation of the ramp half models. A variety of basic delay, jitter, and throughput experiments were performed. The experimental observations are presented and contrasted with the predicted values.

---

[1]Chapter 11 of this dissertation describes an upper level model of a two hop path through an exchange configuration. The lower level models developed in this appendix can be substituted into the Source Ramp and Destination Ramp *stages* of the upper level model. In order to permit this substitution, the domains of the ramp models are restricted to the corresponding logical *stages* of the upper layer model. In particular, aspects of ramp performance that are dependent on the exchange CFRs are modelled within the CFR stages and are excluded from the ramp model's.

[2]Where appropriate, this appendix includes results derived from the experimental programme described in Appendix C. The design of the ISDN ramp is described in Chapter 8.

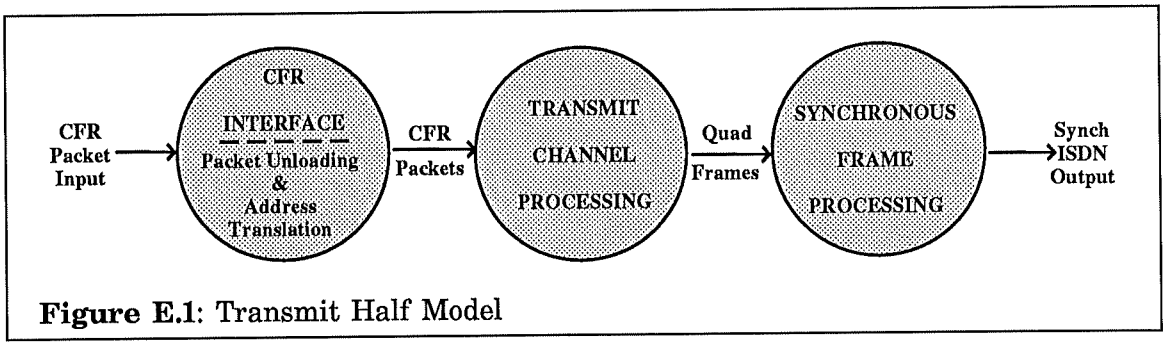

Figure E.1: Transmit Half Model

E.1.1 Ramp Half Models

Transmit Half Model

The transmit half of the ramp encodes the octets of CFR packets into synchronous ISDN frames. This process is modelled by the three stage pipelined network illustrated in Figure E.1. Arriving packets pass through the CFR interface stage and are presented to the channel stage where packets are segmented and quad frames are assembled. The frames processing stage models the separation of quad frames into individual ISDN frames and their subsequent transmission on the ISDN.[1]

There is a simple one-to-one mapping between the packets arriving at and departing from the CFR interface stage. This stage is a single pipeline element that accounts for: the unloading of packets from the CFR station; window field address translation; and channel assignment. Using an experiment similar to that used in the determination of the packet loading delay at the destination CFR, the maximum throughput of this stage was found to be 4.4 KPPS[2]. Since this stage is not internally pipelined, its basic delay is the inverse of the throughput, i.e., 227 usec.

A detailed model of the channel stage is presented in Figure E.2. The frame assembly queueing centre models the transfer of CFR packets between the transputers of the ramp and the mapping of packets onto the generated quad frames.[3] This centre operates synchronously with respect to the downstream ramp

---

[1]There is some incongruity between the arriving and departing *customers* of this model. For the purposes of delay analysis a CFR packet *departs* the model when the ISDN frame containing the last packet octet is transmitted.

[2]Kilopackets per second.

[3]Each quad frame consists of 120 octets.

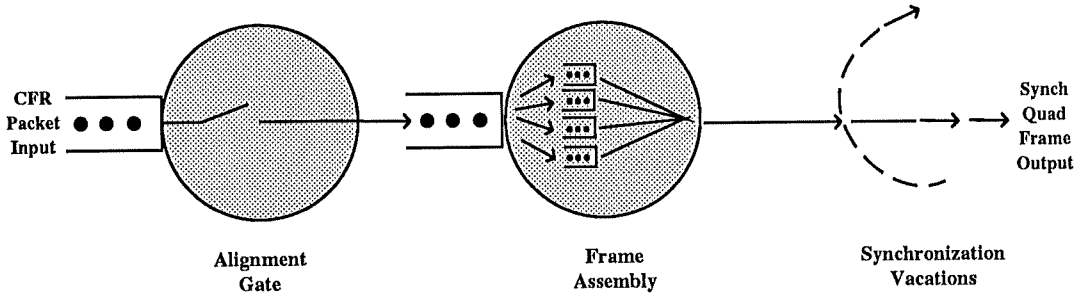

Figure E.2: Transmit Channel Stage

components and a single quad frame is assembled during each 500 usec cycle. In the absence of other traffic, the segments of an arriving packet[1] are spread across some integral number of quad frames, q, determined by the number of timeslots allocated to the channel. For example, packets arriving at a single timeslot channel (q = 10) will be subject to a basic delay of 5000 usec whilst packets arriving at channels of 10 or more timeslots (q = 1) will be processed in a single cycle.

The frame assembly queue is preceded by a *gate* server that delays arriving packets until the beginning of a frame cycle. This server, which induces an evenly distributed 500 usec jitter component, models the alignment delay arising from the asynchronous nature of packet arrivals. Similarly, frame assembly is followed by a *vacation* server that models the periodic transmission of the channel synchronization sequence. This server, which normally passes quad frames without additional delay, takes periodic vacations that cause frame assembly cycles to be skipped. In order to model the transmission of the synchronization sequence, during one out of every 100 quad frames, the vacation server couples a lumped jitter of 500 usec into 1% of the quad frame intervals. The fraction of CFR packets affected by this jitter is a function of q, the number of frame assembly cycles during which each packet is exposed to vacation delays.[2]

The frame processing stage is represented by a series of five delay centres. The first three centres model the forwarding of quad frames between the transputers of the ramp and the separation of quad frames into ISDN frames. Although these centres are clocked at 500 usec intervals, the transfer of each 120 octet frame over the transputer link leads to an overlap in the frame residence time at two of these stages. At the link transmission rate of 3.2 Mbps the overlap interval is

---

[1]Each packet is segmented into 40 octets.

[2]An individual packet is never exposed to more than one 500 usec vacation period.

approximately 300 usec and the total delay across the first three stages is (3 * 500) - 300 usec. The final two stages contribute a further delay component of 250 usec arising from the transfer of ISDN frames into the elastic buffer of the megacard and the shifting of the buffer contents onto the ISDN.

In summary, the model's response to a packet impulse has a basic delay component of 227 + 1450 + q * 500 usec. The basic delay at a 30 timeslot channel can be represented by a unit area impulse that is displaced from the origin by 2177 usec. The superimposed jitter has a 500 usec evenly distributed component and a 500 usec lumped component. The distributed component is equivalent to a 500 usec pulse of unit area and the lumped component can be represented by two impulses: one of .99 units area located at the origin and a second of .01 units area that is displaced by 500 usec. The total transmit half jitter is determined by the convolution of these two components and the overall response is the convolution of this result with the basic delay.[1]

Receive Half Model

The receive half model, illustrated in Figure E.3, is a mirror image of the transmit half with the operation of the three stages reversed so that arriving ISDN frames are mapped into departing CFR packets. For the purpose of basic analysis the delay through the network is measured from the arrival of an ISDN frame carrying the trailing segment of a packet until the departure of the re-assembled packet.

The frame processing stage models the collection of incoming ISDN frames into quad frames and their subsequent transfer to the mapper transputer of the ramp. This stage has a processing delay of 125 usec, an octet deskewing delay of 0, 1, 2, or 3 ISDN frame intervals, and a transputer link delay of 300 usec. The deskewing delay is fixed when a channel is initialized and may be due to timeslot skew within the ISDN or misalignment between the quad frame generation process at the transmit half and the collection process of the receive half.

The channel stage models the operation of the quad frame deskewing buffer and the frame unpacking cycle at the receive half mapper. In the absence of substantial timeslot skew within the ISDN, the quad frame containing the last octet of the packets will be delayed for two quad frame intervals before its contents are unpacked. The additional delay attributable to the frame unpacking process is believed to be proportional to the number of timeslots carrying packet data.

---

[1]For the purposes of this discussion, convolution with an impulse of unit area is equivalent to simple displacement along the time axis.

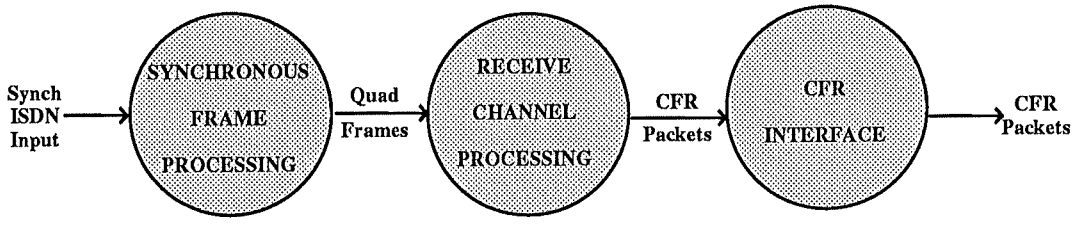

Figure E.3: Receive Half Model

Estimates of the unpacking delay can be derived from the observation that the mapper requires a complete quad frame interval to unpack a fully populated frame.

The CFR interface stage only accounts for the 100 usec packet transfer time delay between the mapper and interface transputers of the ramp. Although the loading of packets into the CFR station is implemented within the ramp itself, the resultant delay is incorporated into the destination CFR model.

In summary, the model's response to a packet impulse has a basic delay composed of a 1525 usec fixed delay, a channel dependent deskewing delay, and a channel dependent unpacking delay. Since there is no variation in the delay at the receive half model, the response can be represented by a unit area impulse that is displaced along the time axis by an amount corresponding to the basic delay.

E.1.2 Delay and Jitter Experiments

For experimental purposes the ISDN interfaces of the peer ramps were locally connected to each other by short lengths of co-axial cable. There is no significant delay or jitter across this *null* transmission medium and, in effect, the transmit ramp half directly clocks ISDN frames into the peer receive half. The delay probe was used to characterize the ramp performance by transmitting single packet impulses between stations attached to peer CFRs. The following paragraphs contrast the expected performance, derived from the ramp models, with the experimental results.

The performance of the inter-ramp channel can be modelled by the simple concatenation of the ramp half queueing networks. The impulse response of this concatenated model is the convolution of the component responses. Since there is no jitter at the receive half, this convolution operation corresponds to the simple displacement of the transmit half result by the basic delay of the receive half. To facilitate comparison with delay probe measurements, the predicted response must

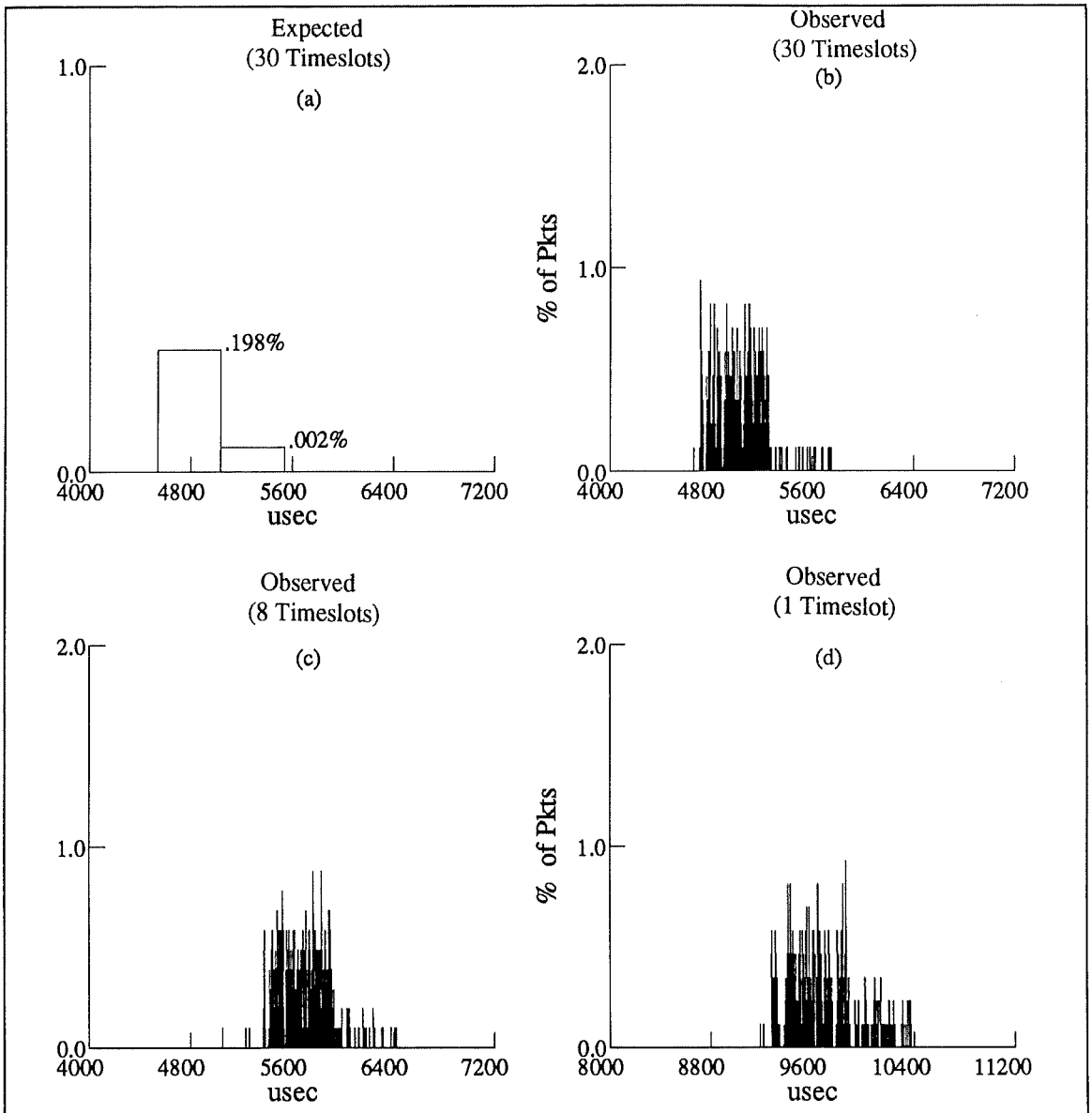

Figure E.4: Basic Ramp Response

be adjusted to compensate for the external CFR delays that are included in the measurements. The CFR jitter components are relatively insignificant[1] and so this adjustment corresponds to the further displacement of ramp response by the relevant basic delays.

[1]8-11 usec at the CFRs versus 500-1000 usec at the transmit half.

Figure E.4(a) depicts the expected ramp result for a 30 timeslot channel.[1] The delay histograms of Figure E.4 (b)-(d) have been plotted from delay probe measurements made using 30, 8, and 1 timeslot channels.[2] Note that the overall shape of these jitter spectra is consistent with the expected density, i.e., the vast majority of the samples are distributed within a 500 usec wide cluster that corresponds to the alignment jitter predicted by the transmit half model. It is expected that packets travelling along narrower channels will experience longer basic delays and this effect is demonstrated by the variation in the initial displacements of the jitter spectra. In Figure E.4(d), the initial cluster is displaced by 4664 usec which is consistent with the predicted displacement of 4535 usec.

A small fraction of the samples are distributed over a second quad frame interval that is displaced by an additional 500 usec. This cluster corresponds to the fraction of packets that are delayed by synchronization vacations. Narrow channels are expected to suffer from increased exposure to vacation periods and this prediction is confirmed by the noticeable difference in the fraction of packets falling within this cluster.

Analysis of the results reveals that, for wide channels (i.e., q = 1), the fraction of packets falling within the second cluster is almost double the 1% fraction anticipated by the original vacation analysis. This observation led to a revised model of behaviour in which the alignment gate remains open at the frame assembly cycle *following* a synchronization vacation. Packets arriving *during* the vacation period are subject to a 500 usec delay at the gate. These packets will depart the ramp before the next vacation and so a given packet experiences a maximum of one delay. The revised model, which is consistent with the experimental results, indicates that a total of (q + 1) % of all packets are subject to vacation-related delays.[3]

The experimental results for a variety of channel widths are summarized in Table E.1. These results are largely as expected, however, the basic delay and

---

[1]The transmit half convolution result is further displaced by a total of 2358 usec corresponding to: 16 usec of basic delay at the source CFR; 275 usec of basic delay at the destination CFR; 1525 usec of fixed basic delay at the receive half; 375 usec of deskewing delay at the receive half; and 167 usec of unpacking delay at the receive half.

[2]The vertical scales of Figure E.4(a) and Figure E.4 (b)-(d) are different.

[3]The revised model triggered an investigation by the author of the ramp software. The portion of the program corresponding to the gate was examined and the source of the additional delay isolated. As a result of the modelling process this jitter component may be removed from future releases of the software.

Channel / usec	Width (Timeslots) 1	2	8	15	30
Basic Delay	9192	6800	5055	4739	4664
Average Delay	9665	7158	5687	5257	5019
95% Jitter	898	709	559	745	535
99% Jitter	1095	962	952	1199	957
Maximum Jitter	1187	1084	1370	1279	1087

Table E.1: Channel Delay and Jitter

maximum jitter entries suggest the presence of delay and jitter components that are not accounted for within the basic ramp model. The discrepancy is partly explained by a transient receive half delay that arises from the processing of incoming synchronization sequences. Further anomalies may be accounted for by variations in the process scheduling of the ramp transputers.[1]

E.1.3 Throughput Experiments

The throughput across a particular channel can be predicted by bottleneck analysis of the ramp half models. Many stages of the ramp models are synchronized to the primary rate ISDN interface and this design restricts the maximum throughput to 6 KPPS. In practice, the throughput is further constrained at either the transmit half channel stage or one of the CFR interface stages.

The packet impulse response of the transmit channel stage implies that its maximum throughput is limited to one packet during every q frame assembly intervals. Under more substantial loads, queues develop within the channel stage allowing the segments of more than one packet to be processed during a single frame interval. For this reason, q, the average number of quad frames per packet,

[1]In particular, a binary choice of process schedules may account for the distinct *notch* that occurs near the leading edge of the ramp jitter spectra.

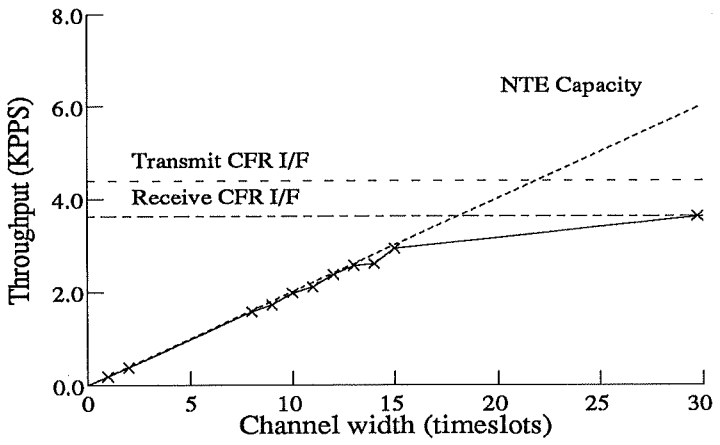

Figure E.5: Ramp/Channel Throughput

need not be an integer and the predicted throughput is directly proportional to the number of timeslots allocated to the channel.

For sufficiently wide channels, the CFR interface stage limits the throughput of the transmit ramp half to 4.4 KPPS. This bottleneck is only exposed when a transmit half supports channels to multiple peer ramps. In basic configurations the ceiling is masked by the 3.6 KPPS limit on CFR transmissions at the receive half. Synthetic portals have been used to measure the maximum ramp throughput for a variety of channel widths. The results are presented in Figure E.5 together with the throughput bounds derived in the preceding paragraphs. The analysis and experiments have only considered unidirectional channel traffic. Although the receive half bottleneck is modelled at the destination CFR, both it and the transmit half constraint arise from the limitations of the CFR interface.[1] Clearly the present interface, which is shared by the two ramp halves, imposes a severe restriction on bidirectional throughput. This bottleneck could be relieved by replacing the shared component by dedicated half duplex interfaces.

E.2 Contention Analysis

This section presents a model for the analysis of the contention effects present at the ISDN ramps and reports on the contention experiments that have been performed. Although the detailed analysis of the model is beyond the scope of this

[1]Recent modifications to the ramp interface software may yield a considerable improvement in unidirectional performance.

dissertation, the modelling process and the experimental observations have provided useful insights concerning ramp performance and behaviour.

E.2.1 The Contention Model

Ramp contention is modelled through the identification of the individual contention elements present at each ramp half. The impact of each of the contention elements is discussed and the appropriate extensions to the basic models are described. An important consequence of contention traffic is the appearance of packet queues that can induce large variations in the delay through an exchange configuration. A principal objective of the modelling process is the identification of the points within the ramp halves at which packet queues may develop.

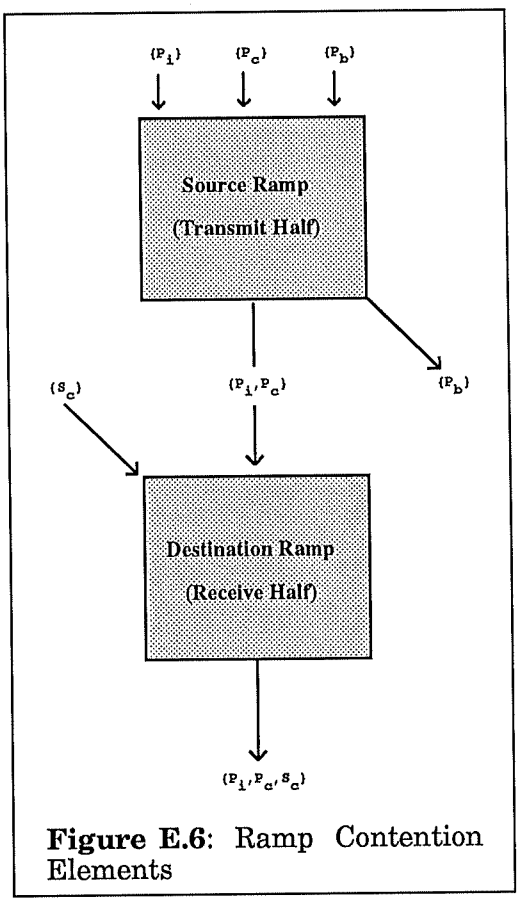

Figure E.6: Ramp Contention Elements

Contention Elements

Figure E.6 illustrates the contention elements that may be encountered along a path between a particular pair of ramp halves.[1] The primary inputs, which arise at the CFR station of the transmit half are: P_i, the traffic of interest; P_c, representing parallel contention traffic mapped to the same channel as P_i; and P_b, the background traffic that is present at the transmit half but mapped to other channels concurrently supported by the transmit half. The secondary input S_c, which is present at the receive half, represents traffic arriving on other channels concurrently supported by the receive half.[2]

---

[1]The following substitutions can me made in order to map the contention elements of the upper layer contention model described in Chapter 11 into the ramp contention elements described in this appendix:

P_c = $\{SP_{DP}, SP_{DC}, SC_{DP}, SC_{DC}\}$;
S_c = $\{DR_{DP}, DR_{DC}\}$; and
P_b = $\{SP_{SR}, SC_{SR}\}$.

[2]The special case where contention traffic flows along a parallel channel between the peer ramps can be represented by paired P_b and S_c elements.

Transmit Half Model

From a contention perspective, the principal transmit half activities are the multiplexing of incoming P_t, P_c, and P_b packets at the ramp's CFR interface and the queueing of P_t and P_c packets awaiting transmission over a shared channel. Packets arising from the P_b contention sources are marshalled onto parallel channels and have no performance impact downstream from the CFR interface.[1] The contention performance of the transmit half can be modelled through extensions to the basic model of Figure E.1. The CFR interface and frame processing stages of the basic model are directly applicable to the contention environment,[2] and so, the channel processing stage is the focal point of transmit half contention analysis.

Queueing effects, induced by the shared channel, are modelled within the frame assembly centre of the detailed channel processing model (Figure E.2). This stage contains a separate queueing centre that models the operation of each of the channels supported by the transmit half. Each channel can be modelled as a multiple-class G/D/1 queue with separate classes representing each of the traffic elements. The deterministic service characteristic reflects the synchronous nature of the channel and the service rate is fixed by the channel bandwidth. The description of the arrival processes and the detailed analysis of this model are beyond the scope of this dissertation.[3]

An important observation concerning the channel queue is that it is *gated* by the 500 usec alignment interval between successive quad frames. In some configurations the maximum number of arrivals during a frame interval will be less than the instantaneous channel capacity. In these cases all of the arrivals within a frame interval will be transmitted along the channel and arrive at the destination ramp in a single batch. The principal contention effect induced at the

---

[1]The ramp design should prevent traffic bound for a saturated channel from inducing internal back-pressure that interferes with the operation of parallel channels.

[2]The operation of the synchronous frame processing stage is traffic independent and, accordingly, contention independent. In contrast, multiplexing at the CFR interface generates back-pressure which, in turn, gives rise to receiver contention at the individual CFR sources. These contention effects are accounted for within the packet transfer model of the source CFR and are excluded from the transmit half contention model.

[3]The arrival processes are somewhat complex as arrivals at this stage will have been *serialized* by the CFR interface and then *batched* by the alignment gate guarding the channel. Given the class-dependent service time distributions at the CFR interface it should be possible to approximate the bulk arrival processes at the channel queue.

transmit half is the batching and relative positioning of transmitted packets. The ordering of packets within a frame is determined by the serialization of arrivals, imposed at the CFR interface, and the service discipline at the channel queue. Variations in batch size and packet ordering at the transmit half induce variations in the delay experienced at the receive half of the peer ramp.

Receive Half Model

The contention performance of the receive half can be modelled through extensions to the basic model of Figure E.3. The principal receive half operation is the unpacking of incoming synchronous frames carrying P_i, P_c, and S_c packets. This unpacking process is modelled within the channel processing stage of the model. Contention effects are also observed at the CFR interface stage where packets queues may develop. This stage is modelled as part of the destination CFR using the multiple class model described in Appendix D.[1]

The frame unpacking process may give rise to batched arrivals at the CFR interface which can, in turn, induce significant jitter components into the symbol stream of interest.[2] The size and ordering of each batch is, in part, determined by the channel processing stages within the peer transmit halves. The alignment gates within all of the peer halves are synchronized to the ISDN[3], thereby forming a single *distributed* gate that generates an aggregate batch of P_i, P_c, and S_c arrivals at the CFR interface of the receive half. The distributed gate is a consequence of the channel multiplexing present at the ISDN interface. The impact of the resultant batch arrivals can be reduced by:

- Introducing a priority queueing discipline that expedites the CFR processing of jitter sensitive traffic; and
- Improving the CFR interface software so that a larger number of packets can be transmitted during each frame interval.

---

[1]Arrivals at this stage have separate classes representing P_i, P_c, and S_c. In the present ramp implementation, the service time at the CFR is dominated by the limitations of the receive half's CFR software. The effect is so severe that, for many configurations, contention effects at the destination CFR can be ignored and a single distribution can be used to represent the service time experienced by all three classes of traffic.

[2]Consider a receive half supporting one channel, carrying the P_i traffic, and 29 other channels carrying S_c traffic. An arriving synchronous frame may contain the final octets of packets being transmitted on any or all of the channels. This frame may generate a thirty packet bulk arrival at the CFR interface and, depending on its relative positioning within the batch, a P_i packet could experience an S_c induced jitter of up to 29 CFR transmission periods.

[3]The channel and frame processing stages of all of the ramps halves are synchronously drive by the ISDN.

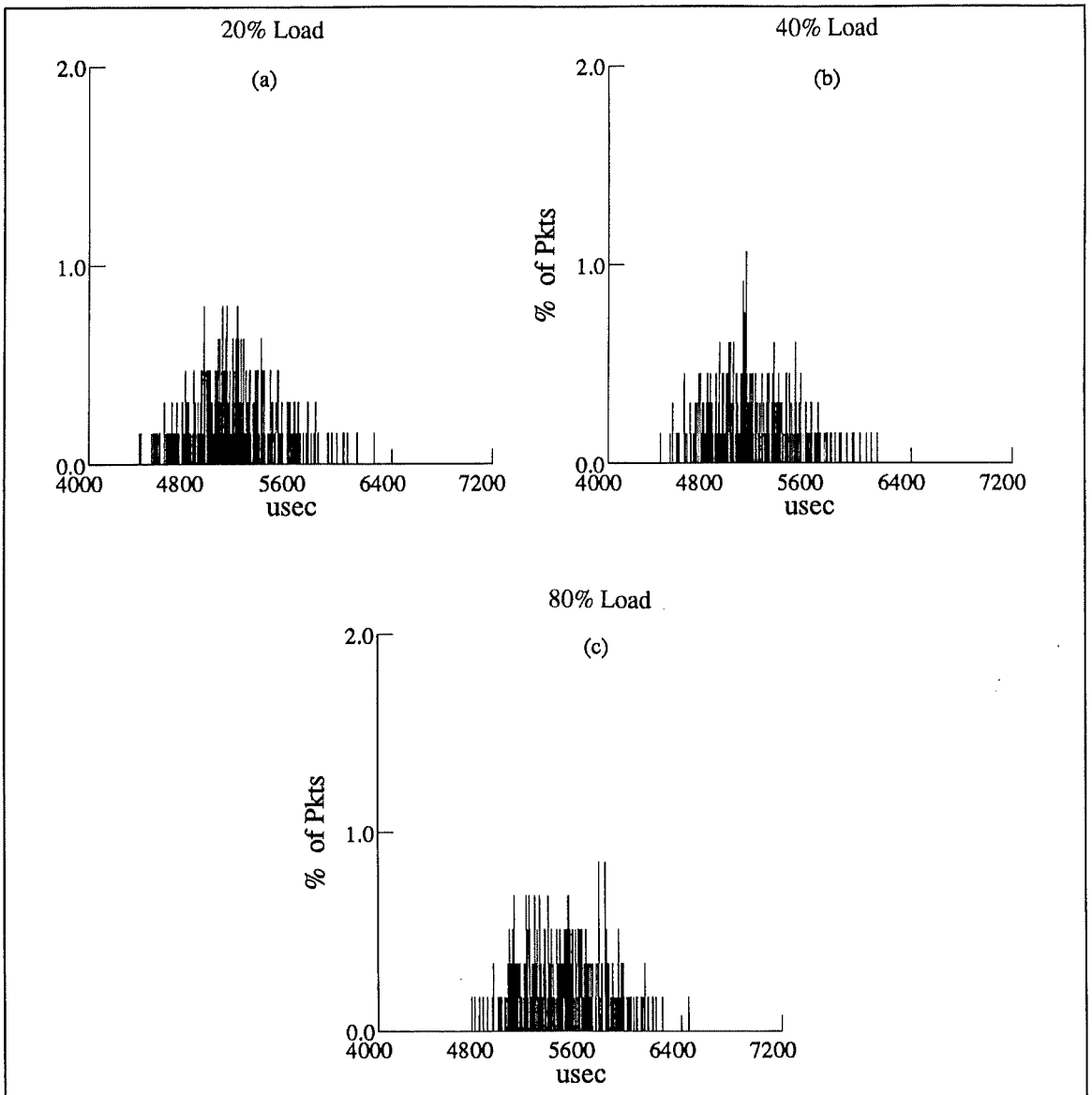

Figure E.7: Ramp Contention Response - Periodic Load

E.2.2 Experimental Results

The experimental programme included a number of experiments that investigated the effect of P_c contention on the impulse response observed by the delay probe.[1] The contention loads were generated by synthetic portals and experiments were performed using periodic and burst arrival patterns.

[1]Experiments involving P_b contention traffic are described within Appendix D which reports on CFR contention experiments.

Periodic Loads

The histograms of Figure E.7 depict the contention response of a pair of peer ramps supporting a single 30 timeslot wide channel. The contention traffic was generated by two similar traffic sources which generated periodic packet arrivals. Figure E.7 (a), (b), and (c) represent experiments in which the total packet throughput corresponded to 20%, 40%, and 80% of the available ramp capacity.[1]

Through comparison to Figure E.4(b), it can be seen that the contention traffic causes the attenuation of the jitter spectra. This effect is consistent with the expectation that some fraction of the delay probe samples will be transmitted within the same frame alignment interval as packets generated by one or both of the traffic sources. These samples will be delayed by bulk arrival effects at the frame unpacking process of the receive half and, depending on their relative position within the frame, at the CFR interface.[2] Provided the overall traffic levels remain within the ramp capacity, queues do not develop and the overall jitter is constrained.

Burst Loads

In one set of experiments the arrival pattern generated by the contention source consisted of rapid bursts of packets separated by idle periods. The burst duty cycle was adjusted so that the average load offered by the contention source was equivalent to 25% of the available ramp capacity. Each burst results in a 41 msec period of 100% ramp utilization.

Figure E.8(a) is the jitter spectrum observed in the presence of a single burst load. The samples in the large cluster around 5400 usec correspond to the impulse packets transmitted during idle periods. The remaining samples, which are spread over a range of about 43 msec, correspond to the packets transmitted during bursts. The result clearly demonstrates that significant jitter components are induced when ramp capacity is saturated, even on a transient basis.

---

[1]In this configuration the maximum throughput supported by the peer ramps is limited by the CFR interface of the receive half to 3.6 KPPS.

[2]In the present ramp implementation the delay associated with frame unpacking varies with the number of timeslots in the frame that contain packet fragments. Although the software releases each packet as its final octets are assembled, the transputer's scheduling policy may allow the unpacking process to run to completion before activating the downstream process.

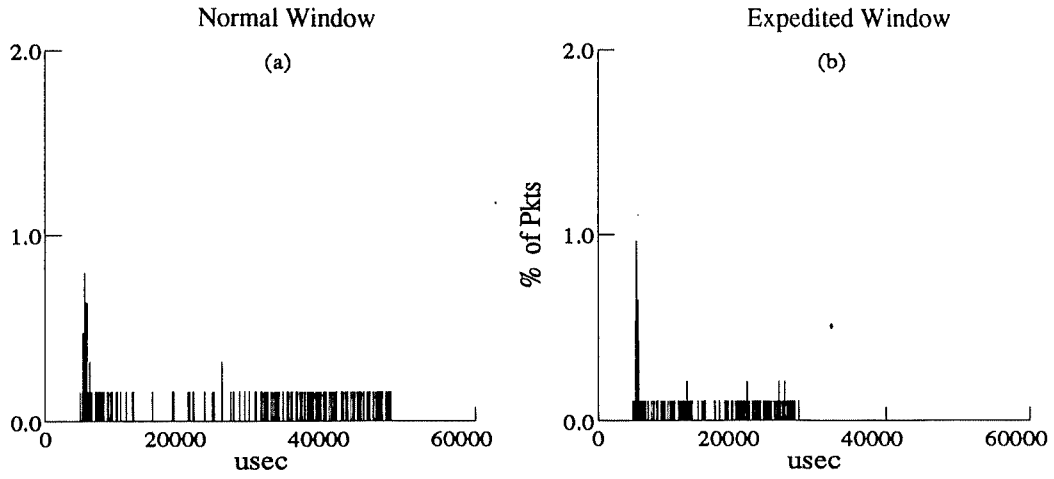

Figure E.8: Ramp Contention Response - Burst Load

The ramp's expedited transfer function can be used to insulate jitter-sensitive traffic from the transient effects of burst loads. Traffic is partitioned by assigning a distinguished address window to the expedited associations. Packets arriving on this window are granted priority over packets arriving on other windows bound to the shared channel. These packets will experience receiver contention at the source CFR but will not be delayed at the transient queues generated by the burst traffic.

An attempt to exercise this function was somewhat frustrated by the preliminary state of the ramp software. The experimental result, illustrated in Figure E.8(b), demonstrates a significant reduction in the jitter range but not the dramatic improvement that had been anticipated. The jitter spectrum should have been similar to the spectra observed in the presence of periodic loads. In the preliminary ramp software, an elastic buffer separated the CFR interface process from the downstream process that prioritized incoming packets. During packet bursts, this buffer expanded into a transient queue that unnecessarily delayed the priority traffic.[1]

[1]This problem has been addressed in the production release of the ramp software.

Appendix F
Exchange Protocol Suite

In the exchange environment the transmission of symbols between peer applications is supported by a layered stack of communication services as illustrated in Figure F.1. The lower layers of the stack, common to all exchange components, are based on the formatting of physical media to effect the exchange of CFR packets between peer systems. The Unison Data Link Layer (UDL), common to all systems, extends the lower layer services to provide the full functionality of the SI-service. This service is used by a variety of upper layer services to support portal and management applications.

F.1 Lower Layers

At the lower layers, communication within the local exchange is based on the use of CFR station nodes that support the direct exchange of CFR packets. The node interface defines a tangible boundary between the lower layers, which supports the exchange of CFR packets, and the UDL layer. In the exchange architecture the service provided at this boundary is extended, through the use of bridge mode controllers and ramps, to support communication between systems attached to different exchanges. In order to model these extensions, the specification of the lower layers is divorced from the electrical interface and transmission mechanisms associated with the CFR chip set.

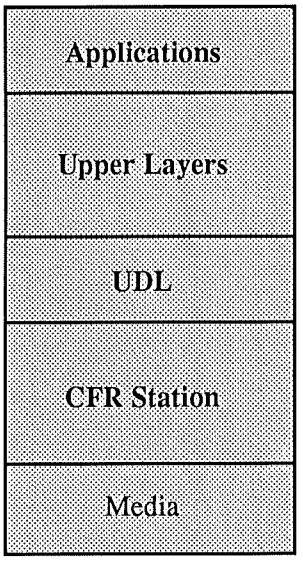

Figure F.1: Exchange Reference Model

Two distinct lower layers are identified in Figure F.2. The physical layer supports the exchange of digital symbols over a medium. A variety of media and transmission techniques can be

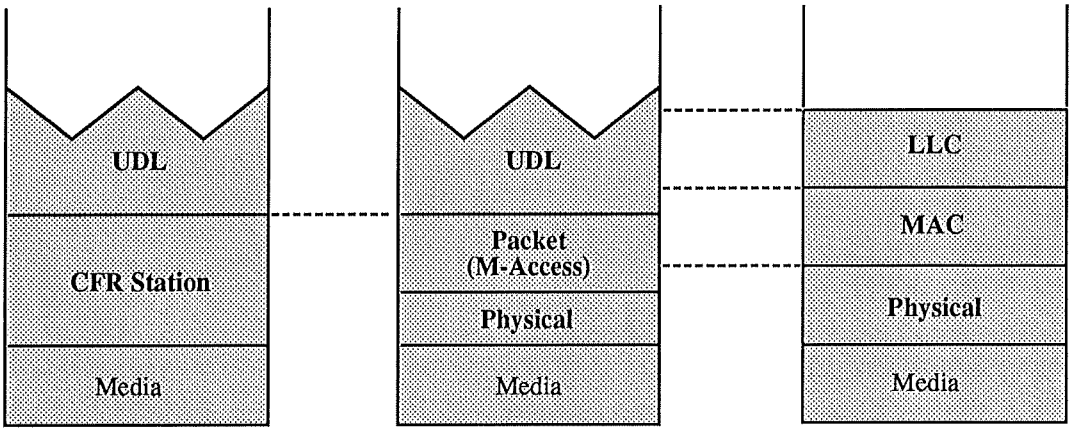

Figure F.2: Lower Layers - Exchange vs OSI

used to implement this service.[1] The packet layer supports the exchange of CFR packets over physical layer services and various encoding, multiplexing and switching techniques can be used within this layer. Peer ramps, operating within the packet layer, implement relaying in order to support the end-to-end exchange of symbols between packet layer uses.

An informal description of the packet layer service is provided in [Chambers 86b] and a more rigorous specification can be found in [Billington 87]. In both descriptions the service provided at the layer boundary is characterized in terms of *request* and *indication* primitives that model the presentation of outgoing and incoming packets. The specification of the packet layer in these terms provides a simple model that encompasses the in-band operation of CFRs, and ramps without being dependent on particular hardware implementations.

Relationship to IEEE and OSI Models

Figure F.2 illustrates the relationship between the exchange model of lower layer operation and the equivalent IEEE 802 model that is the basis of current LAN standardization work. The lower layers of the IEEE model consist of physical, Media Access and Control (MAC) and Logical Link Control (LLC) layers. MAC and LLC can be viewed as constituent sublayers of the ISO Data Link Layer specified in the OSI reference model.

The exchange packet layer, which is also referred to as the M-Access layer, directly support the UDL layer. M-Access provides a subset of the MAC functions and so

---

[1]For example, the local exchange medium is a parallel ring based on backplane interconnections. Similarly, the inter-ramp medium is based on common carrier services.

the line delineating its service boundary is drawn below the corresponding MAC boundary. Although packet layer functionality is restricted, relaying elements may be embedded within this layer and, consequently, its geographic range exceeds the local network limits of the MAC layer.[1]

The principal M-Access limitation is that the CFR packet format severely constrains the lengths of the upper layer address and data fields. To achieve an IEEE-compatible MAC service the packet layer must be augmented to support the segmentation and reassembly (SAR) of service data units (SDUs) that are longer than a single packet. This scheme has been adopted in the IEEE Metrolan project [IEEE 802.6] where the MAC layer has been separated into separate Access and SAR components. The advantage of this approach is that the presence of the SAR components within the MAC layer is transparent to the LLC users. The disadvantage, is that the IEEE service specifications limit the degree of multiplexing that can be performed at the MAC/LLC layer boundary.

F.2 The Unison Data Link Layer

Although the packet layer supports the sequenced and error-free exchange of ATM packets between peer systems, it does not support the multiplexing of packets arising from concurrent associations between upper layer users. The multiplexing function, which is taken to include the complementary demultiplexing function, is an important feature of the SI-service. In practice, it is convenient to combine the multiplexing and segmentation functions, and so, the Unison Data Link (UDL) service [Tennenhouse 86c] supports the exchange of multi-packet SDUs, referred to as *blocks*. UDL is logically divided into two sublayers: the lower sublayer supports the multiplexing function; and the upper sublayer supports segmentation. The SI-service boundary is located at the internal boundary between these sublayers.[2]

UDL lies at the heart of the exchange protocol suite. It is not aligned with the LLC sub-layer and the service provided to upper layer users is not compatible with the specification of the OSI data link layer. The distinctions are significant and deliberate. In some respects, for example, flow control, UDL has less functionality

---

[1]Recently the importance of low level relaying has become more generally accepted and work is under way within the IEEE P802 committee to define mechanisms for expanding the geographic range of the MAC layer.

[2]UDL-users can select segmentation functions on an association-specific basis. When segmentation is not selected, user primitives bypass the segmentation layer and are directly supported by the SI-service.

than its counterparts: many LLC features are not appropriate for inclusion in the lower layers of a site interconnection architecture. In other respects, for example multiplexing, UDL has greater functionality to ensure low-level support of features that are essential to the SI-service.

Link layer support of multiplexing is the single feature that most distinguishes UDL from other services, such as the connectionless variant of LLC. In the IEEE/OSI protocol suite, the data link layer identifies the SDUs associated with different local service access points (SAPs). However, these SAPs are only used to distinguish between different classes of upper layer protocols. Similarly, in the ARPA suite the IP layer multiplexing function is limited to protocol discrimination. The multiplexing of SDUs arising from different associations is accomplished within the upper layers and, consequently, the existence of distinct upper layer associations is transparent to the data link entities. Although this degree of transparency is prescribed by the OSI principle of layer independence, it is incompatible with the requirements of the SI-service.

The implementation of association-dependent multiplexing, within the data link layer, is essential to the transport of multi-service network traffic. If multiplexing and demultiplexing are restricted to the upper layers, then a traffic burst arising from a single association can easily back pressure other associations sharing the network point of attachment. If jitter is to be constrained, then the link layer must support the multiplexing of independent SDUs arising from different associations.

UDL multiplexing and segmentation are supported through the use of the standard encoding described in [Tennenhouse 86b]. A source UDL entity segments each service data unit (SDU) into a number of UDL packets that are presented to the packet layer for transfer to the peer destination entity. The data portion of each packet carries a 4 octet header together with an SDU segment of up to 28 octets. The header contains a port field, that supports the multiplexing function, and other fields that support the reassembly function.

At the source entity, the SDUs of a given association are processed sequentially and the order of SDU segments is preserved as they are presented to the packet layer. The UDL layer processes the SDUs of concurrent associations in parallel and it may interleave the presentation of segments arising from different associations. At the destination UDL entity, the port field, within the header of each packet, permits the interleaved reception of SDU segments arising from different associations, even in cases where the concurrent associations involve the

same pair of peer entities. The port field is used to immediately demultiplex packets belonging to different associations so that the reassembly and expedited transfer functions can be performed on an association-specific basis. Early demultiplexing ensures that the reassembly of an SDU arising from one association is not delayed by the reassembly of SDUs arising from other associations.

The port field value, transmitted in each packet header, uniquely identifies a single association within the context of the destination entity. When an association is established, the secretary stub operating within each of the peer systems, specifies a port field value for the reception of packets related to the association. The selected port values are exchanged as part of the end-to-end secretary interaction that takes places during association establishment.[1]

The low level support of multiplexing distinguishes UDL from other proposals such as the scheme described in [Ades 87a]. Although the UDL encoding resembles the *transfer sub-protocol* proposed in [Temple 84], transfer *channels* impose a block level acknowledgement function and are somewhat more transient than UDL associations. Temple suggests that peer stations use an in-band *exchange sub-protocol* to directly negotiate channel establishment. In contrast, UDL associations are negotiated out-of-band under the auspices of the exchange secretary service.

F.3 Upper Layers

In the exchange environment there is no requirement for the discrete network and transport layers described in the OSI model. Many of the functions normally provided at these layers are supported through the combination of: UDL multiplexing; the switching and relaying functionality embedded within the exchange packet layer; and the out-of-band exchange management services. Other network and transport functions, such as flow control, acknowledgement, and error recovery are not relevant to the site interconnection environment. Where appropriate, these functions are performed on an end-to-end basis by the client devices attached to local networks.

Above the UDL layer, application entities can directly exercise associations. Alternatively, an explicit session layer and a presentation encoding can be used to synchronize transactions and to support inter-operability between different

---

[1]The association establishment procedure is described in Chapter 10.

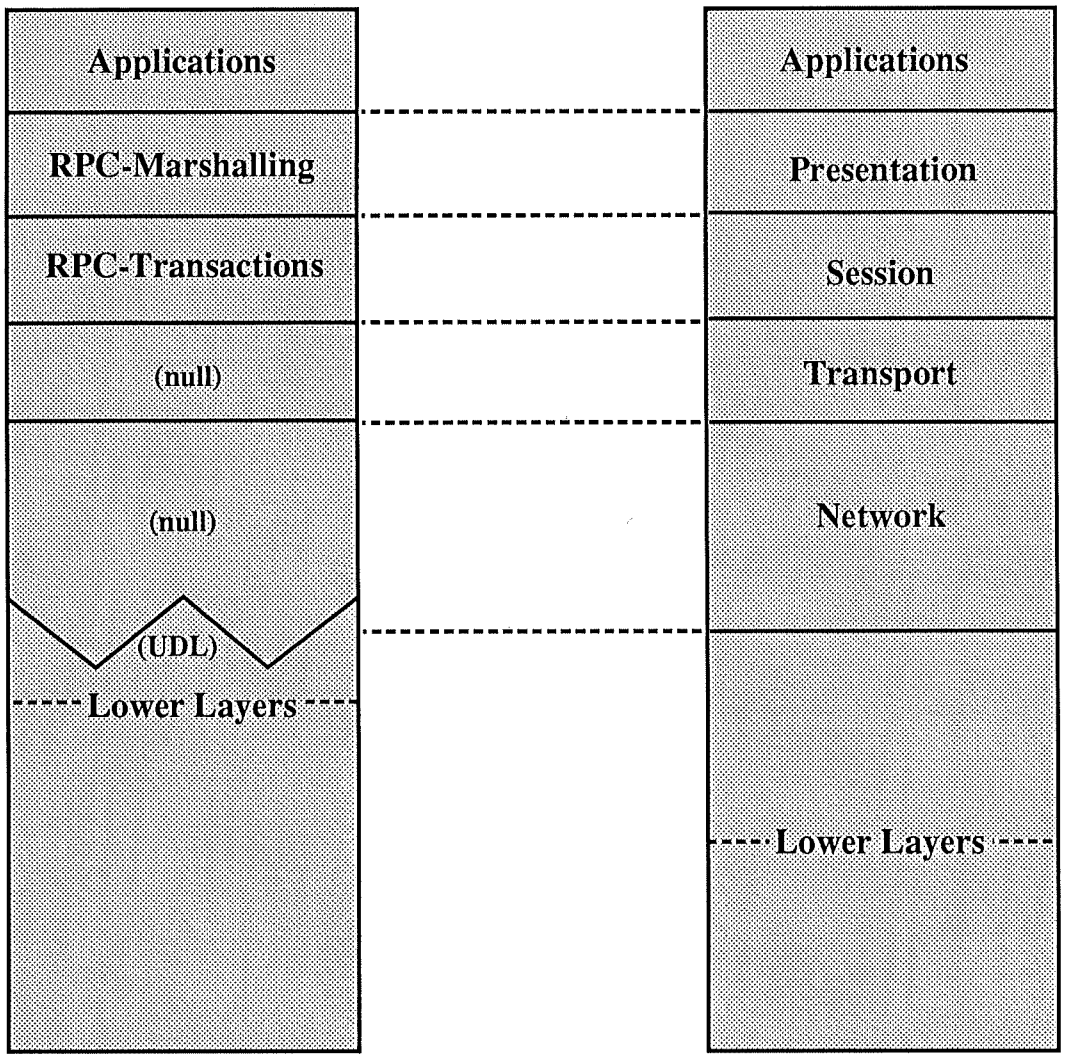

Figure F.3: Upper Layers - Exchange vs OSI

application layer implementations. Within Project Unison, the Unity [McAuley 87] Remote Procedure Call (RPC) mechanism has been adopted as the common session layer and presentation encoding.

Figure F.3 illustrates the relationship between the exchange upper layers and those of the OSI model. For modelling purposes, the RPC mechanism can be decomposed into two components:

- The *transaction* component is responsible for the binding between associations and module interfaces and for the operation of the RPC protocol. This component is equivalent to the OSI session layer and supports transaction level duplicate detection, acknowledgement, and error recovery; and
- The *marshalling* component supports a subset of the OSI presentation layer functions. It is responsible for the encoding of procedure call identifiers and

parameters into a standard implementation-independent syntax.[1]

F.4 Application Layer

Many aspects of application layer communication are specific to the communicating application processes. In the case of portals, the primary applications is the support of peer portal communication. A secondary application is access to exchange management services. These applications, function in parallel, and are supported by different upper layer protocol stacks operating over the common UDL layer.

Exchange management services, such as the secretary, are fundamentally transaction oriented and, accordingly, the applications that access these services rely on the RPC protocol to protect the integrity of their transactions.[2]

Since peer portal communication is the principal exchange function, different peer portal protocols have been developed to support the interconnection of different types of client networks. In most instances portal communication operates directly over the UDL layer without the functions, and overhead, associated with the RPC layer. This policy is appropriate for three reasons:

- Traffic between peer portals is not symmetric or transaction-oriented and so the RPC paradigm is not an appropriate model for peer portal communication;
- Portal entities do not, in general, support acknowledgement or error recovery and so there is no requirement for session layer synchronization functions; and
- The in-band data exchanged by peer portals is opaque to the portal applications and so there is no requirement for presentation encoding of the transferred data.

In some cases, peer portals may exchange both in-band and out-of-band information. In these applications portal communication may be split across several associations supporting the parallel operation of UDL-based in-band exchanges and RPC-based out-of-band transactions.

---

[1]There is no provision for the encoding negotiation function that is supported by the OSI presentation layer.

[2]The management transactions are defined in terms of remote procedure identifiers and corresponding sequences of parameters.

F.5 Summary

The services and protocols described in this appendix have been tailored to the support of the exchange site interconnection architecture. They are also appropriate to other multi-service network applications and form the basis of the CFR communications architecture used within CFR-based local area networks.[1] In particular, the UDL service and protocol have been adopted as the standard data link layer used in CFR environments. In the exchange application, UDL allows each portal to maintain concurrent associations with the exchange management services and with a number of peer portals. The low level multiplexing function ensures that a portal's interactions with exchange management have minimal impact on the in-band flow of user data.

[1]Further information concerning the relationship between the CFR protocol suite and the corresponding OSI and IEEE standards can be found in [Chambers 86a].

References

[Adams 85] Adams, C. J. and S. Ades
Voice Experiments in the Universe Network,
International Conference on Communications (ICC 85), Chicago, June 1985.

[Adams 87] Adams, C. J.
The CFR-CFR Portal: An Implementation Plan,
Project Unison Working Paper UR038,
Rutherford Appleton Laboratory, Oxford, September 1987.

[Ades 86] Ades, S., R. Want, and R. S. Calnan
Protocols for Real Time Voice Communication on a Packet Local Network,
International Conference on Communications (ICC 86), Toronto, June 1986.

[Ades 87a] Ades, S.
A High Speed Network Interface for Integrated Services,
Proc. IEEE Infocom '87, San Francisco, March 1987.

[Ades 87b] Ades, S.
An Architecture for Integrated Services on the Local Area Network,
PhD.Dissertation,
University of Cambridge Computer Laboratory TR 114, September 1987.

[Billington 87] Billington, J.
A High-level Petri Net Specification of The Cambridge Fast Ring M-Access Service,
University of Cambridge Computer Laboratory TR 121, December 1987.

[Boggs 80] Boggs, D. R. et al
PUP: An Internetwork Architecture,
IEEE Transactions on Communications, Vol 28 - No 4, April 1980.

[Broomell 83] Broomell, G. and J. R. Heath
Classification Categories and Historical Development of Circuit Switching Technologies,
ACM Computing Surveys, Vol 15 - No 2, June 1983.

[Budrikis 86] Budrikis, Z. L. et al,
QPSX: A Queued Packet and Synchronous Circuit Exchange,
Proc. 8th International Conference on Computer Communication, Munich, 1986.

[Bux 87] Bux, W., D. Grillo, and N. F. Maxemchuk (Ed.)
Interconnection of Local Area Networks,
IEEE J. on Selected Areas in Communications, Vol 5 - No 9, Dec. 1987.

[Calnan 87] Calnan, R. S.
ISLAND: A Distributed Multimedia System,
Proc. IEEE Globecom '87, Tokyo, November 1987.

[Calnan 88] Calnan, R. S.
The Integration of Voice Within a Digital Network,
PhD.Dissertation (To be Submitted),
University of Cambridge Computer Laboratory, September 1988.

[CCITT G.701] CCITT Recommendation G.701
Vocabulary of PCM and Digital Transmission Terms,
CCITT Red Book, Volume III - Fascicle 3, ITU, Geneva, 1984.

[CCITT G.703] CCITT Recommendation G.703
General Aspects of Interfaces,
CCITT Red Book, Volume III - Fascicle 3, ITU, Geneva, 1984.

[CCITT G.711] CCITT Recommendation G.711
Pulse Code Modulation (PCM) of Voice Frequencies,
CCITT Red Book, Volume III - Fascicle 3, ITU, Geneva, 1984.

[CCITT G.732] CCITT Recommendation G.732
Characteristics of Primary PCM Multiplex Equipment Operating at 2048 Kbit/s,
CCITT Red Book, Volume III - Fascicle 3, ITU, Geneva, 1984.

[CCITT I.320] CCITT Recommendation I.320
ISDN Protocol Reference Model,
CCITT Red Book, Volume III - Fascicle 5, ITU, Geneva, 1984.

[CCITT I.412] CCITT Recommendation I.412
ISDN User-Network Interfaces: Interface Structures and Access Capabilities,
CCITT Red Book, Volume III - Fascicle 5, ITU, Geneva, 1984.

[CCITT Q.920] CCITT Recommendation Q.920
ISDN User-Network Interface Data Link Layer: General Aspects,
CCITT Red Book, Volume VI - Fascicle 9, ITU, Geneva, 1984.

[CCITT Q.921] CCITT Recommendation Q.921
ISDN User-Network Interface Data Link Layer Specification,
CCITT Red Book, Volume VI - Fascicle 9, ITU, Geneva, 1984.

[CCITT Q.930] CCITT Recommendation Q.930
ISDN User-Network Interface Layer 3: General Aspects,
CCITT Red Book, Volume VI - Fascicle 9, ITU, Geneva, 1984.

[CCITT Q.931] CCITT Recommendation Q.931
ISDN User-Network Interface Layer 3 Specification,
CCITT Red Book, Volume VI - Fascicle 9, ITU, Geneva, 1984.

[Chambers 86a] Chambers, A. M. and D. Tennenhouse
Communication Architectures for the Cambridge Fast Ring,
Project Unison Working Paper UA004,
Acorn Computers Ltd, Cambridge, October 1986.

[Chambers 86b] Chambers, A. M.
CFR M-Access Service Definition,
Project Unison Working Paper UA010,
Acorn Computers Ltd, Cambridge, November 1986.

[Clark 86] Clark, P. F. et al
Unison - Communications Research for Office Applications,
IEE Electronics and Power, September 1986.

[Coudreuse 87] Coudreuse, J and M. Servel
Prelude: An Asynchronous Time-Division Switched Network,
International Conference on Communications (ICC 87), Seattle, June 1987.

[Dalal 81] Dalal, Y. K. and R. S. Printis
48-bit Absolute Internet and Ethernet Host Numbers,
Proc. 7th Data Communication Symposium
Computer Communication Review, Vol 11 - No 4, October 1981.

[Day 87] Day, C., J. Giacopelli, and J. Hickey
Applications of Self-Routing Switches to
LATA Fiber Optic Networks,
Proc. IEEE Int. Switching Symposium (ISS 87), Phoenix, March 1987.

[IEEE 802.6] Draft IEEE Standard 802.6
Metropolitan Area Network (MAN), Media Access Control,
Revision F, IEEE P802.6/85-01, February 1986.

[Eguchi 87] Eguchi, Y., H. Ichikawa, and M. Yoshikawa
NTT's Improved Video Conferencing System,
Proc. IEEE Infocom '87, San Francisco, March 1987.

[Estrin 87] Estrin, D.
Interconnection Protocols for Interorganization Networks,
IEEE J. on Selected Areas in Communications, Vol 5 - No 9, Dec. 1987.

[Falconer 85a] Falconer, R. M., J. L. Adams, and G. M. Walley
A Simulation Study of the Cambridge Ring with Voice Traffic,
British Telecom Technology Journal, Vol 3 - No 2, April 1985.

[Falconer 85b] Falconer, R. M. and J. L. Adams
Orwell: A Protocol for an Integrated Services Local Network,
British Telecom Technology Journal, Vol 3 - No 4, October 1985.

[Gallagher 86] Gallagher, I. D.
Multi-Service Networks,
British Telecom Technology Journal, Vol 4 - No 1, January 1986.

[Garnett 83] Garnett, N. H.
Intelligent Network Interfaces,
PhD.Dissertation,
University of Cambridge Computer Laboratory TR 46, May 1983.

[Greaves 88] Greaves, D. and A. Hopper
The Cambridge Backbone Network,
Proc. European Fibre Optic Conference (EFOC/LAN 88), Amsterdam, June 1988.

[Griffiths 84] Griffiths, J. W. R. (Ed.)
Project Universe: Images,
Project Universe Report No.12,
Rutherford Appleton Laboratory, Oxford, 1984.

[Gruber 81] Gruber, J. G.
Delay Related Issues in Integrated Voice and Data Networks,
IEEE Transactions on Communications, Vol 29 - No 4, June 1981.

[Gruber 83a] Gruber, J. G. and L. Strawczynski
Judging Speech in Dynamically-Managed Voice Systems,
Telesis 1983, No 2, Bell Northern Research, Ottawa, 1983.

[Gruber 83b] Gruber, J. G. and H. L. Nguyen
Performance Requirements for Integrated Voice/Data Networks,
IEEE J. on Selected Areas in Communications, Vol 1 - No 6, Dec. 1983.

[Hall 79] Hall, R. D. and M. J. Snaith
Jitter Specification in a Digital Network,
Post Office Electrical Engineering Journal, Vol 72, July 1979.
(Note: POEEJ renamed British Telecommunications Engineering)

[Harita 87] Harita, B. R.
First Year Report and PhD. Thesis Proposal,
University of Cambridge Computer Laboratory, October 1987.

[Hawe 84] Hawe, W. R., A. Kirby, and A. Lauck
An Architecture for Transparently Interconnecting
IEEE 802 Local Area Networks,
Digital Equipment Corporation TR 322, November 1984.

[Hemrick 88] Hemrick, C. F., R. W. Klessig, and J. M. McRoberts
Switched Multi-Megabit Data Service and
Early Availability via MAN Technology,
IEEE Communications Magazine, Vol 26 - No 4, April 1988.

[Hopper 78] Hopper, A.
Local Area Computer Communication Networks,
PhD.Dissertation,
University of Cambridge Computer Laboratory TR 7, April 1978.

[Hopper 79] Hopper, A. and D. J. Wheeler
Binary Routing Networks,
IEEE Transactions on Computers, Vol 28 - No 10, October 1979.

[Hopper 86] Hopper, A. and R. M. Needham
The Cambridge Fast Ring Networking System (CFR),
University of Cambridge Computer Laboratory TR 90, June 1986.
To appear in: *IEEE Transactions on Communications, Fall 1988.*

[Hui 87] Hui, J. Y. and E. Arthurs
A Broadband Packet Switch for Integrated Transport,
IEEE J. on Selected Areas in Communications, Vol 5 - No 8, Oct. 1987.

[ISO 7498-1] International Standards Organization - Information Processing
Open Systems Interconnection - Basic Reference Model,
International Standard 7498-1, ISO.
Aligned with, CCITT Recommendation X.200

[ISO 7498-3] International Standards Organization - Information Processing
OSI Reference Model - Part 3: Naming and Addressing,
International Standard 7498-3, ISO.

[ISO 8348] International Standards Organization - Information Processing
Network Service Definition,
International Standard 8348, ISO.
Also, International Standards Organization - Information Processing
Addendum 1: Connectionless-mode Transmission,
International Standard 8348/Add.1, ISO.

[ISO 8473] International Standards Organization - Information Processing
Protocol for Providing the Connectionless-Mode Network Service (Internetwork Protocol),
International Standard 8473, ISO.

[ISO 8648] International Standards Organization - Information Processing
Internal Organization of the Network Layer,
International Standard 8648, ISO.

[ISO 9065] International Standards Organization - Information Processing
Message Oriented Text Interchange System (MOTIS) - Interpersonal Messaging System,
Draft International Standard DIS 9065, ISO.
Aligned with, CCITT Recommendation X.420

[ISO 9594-1] International Standards Organization - Information Processing
The Directory, Part 1: Overview of Concepts, Models, and Services,
Draft International Standard DIS 9594-1, ISO.
Aligned with, CCITT Recommendation X.500

[Kearsey 84] Kearsey, B. N. and R. W. McLintock
Jitter in Digital Telecommunication Networks,
British Telecommunications Engineering, Vol 3, July 1984.

[Key 87] Key, M. and M. Karimzadeh
Connection Control Protocols in a Fast Packet Switching Multi-Service Network Based on ATD Techniques,
British Telecom Technology Journal, Vol 5 - No 1, January 1987.

[Kim 83] Kim, B. G.
Characteristics of Arrival Statistics of Multiplexed Voice Packets,
IEEE J. on Selected Areas in Communications, Vol 1 - No 6, Dec. 1983.

[King 82] King, P. J. B. and I. Mitrani
Modelling the Cambridge Ring,
ACM Performance Evaluation Review, Vol 11 - No 4, 1982.

[Kleinrock 76] Kleinrock, L.
Queueing Systems, Volumes I and II, John Wiley & Sons, 1975 and 1976.

[Lampson 86] Lampson, B. W.
Designing a Global Name Service,
Proc. 5th ACM Symp. on Princ.of Distributed Comp., Calgary, Aug. 1986.

[Lazowska 84] Lazowska, E. D. et al
Quantitative System Performance, Prentice-Hall, 1984.

[Leslie 83] Leslie, I.
Extending the Local Area Network,
PhD.Dissertation,
University of Cambridge Computer Laboratory TR 43, February 1984.

[Leslie 84] Leslie, I. M. et al
The Architecture of the Universe Network,
Sigcomm '84, Computer Communication Review, Vol 14 - No 2, June 1984.

[Leslie 85] Leslie, I. M. and D. Tennenhouse
Unison Network Megastream Ramps: Internal Structure,
Computer Laboratory Working Paper,
University of Cambridge Computer Laboratory, May 1985.

[Littlewood 87] Littlewood, M. and J. L. Adams
Evolution Toward an ATD Multi-Service Network,
British Telecom Technology Journal, Vol 5 - No 2, April 1987.

[McAuley 87] McAuley, D.
Unity: An RPC Mechanism,
Project Unison Working Paper UC027,
University of Cambridge Computer Laboratory, March 1987.

[Metcalfe 76] Metcalfe, R. M. and D. R. Boggs
ETHERNET: Distributed Packet Switching for Local Computer Networks,
Communications of the ACM, Vol 19 - No 7, July 1976.

[Milway 86] Milway, D.
Binary Routing Networks,
PhD.Dissertation,
University of Cambridge Computer Laboratory TR 101, December 1986.

[Minzer 87] Minzer, S. E.
Broadband User-Network Interfaces to ISDN,
International Conference on Communications (ICC 87), Seattle, June 1987.

[Mollenauer 88] Mollenauer, J. F.
Standards for Metropolitan Area Networks,
IEEE Communications Magazine, Vol 26 - No 4, April 1988.

[Needham 82] Needham, R. M. and A. J. Herbert
The Cambridge Distributed Computing System, Addison-Wesley, 1982.

[Newman 88a] Newman, P. N.
A Broadband Packet Switch for Multi-Service Communications,
Proc. IEEE Infocom '87, New Orleans, March 1988.

[Newman 88b] Newman, P. N.
A Fast Packet Switch for the Integrated Services Backbone Network,
To appear in: *IEEE J. on Selected Areas in Communications*, Dec. 1987.

[Porter 85] Porter, J.
The Wheeler U-Mapper,
Project Unison Working Paper UC005,
University of Cambridge Computer Laboratory, November 1985.

[Postel 81] Postel, J.
Internet Protocol,
ARPA RFC 791,
Information Sciences Institute, USC, 1981.

[Postel 82] Postel, J.
Internet Multimedia Mail Document Format,
ARPA RFC 767,
Information Sciences Institute, USC, 1982.

[Rainforth 87] Rainforth, M. A.
The Megastream Interface System, Information Systems Group,
Rutherford Appleton Laboratory, Oxford, 1987.

[Richards 79] Richards, M. et al
Tripos: A Portable Operating System Design for Mini-Computers,
Software Practice and Experience, Vol 9 - No. 7, June 1979.

[Richardson 83] Richardson, M.
The Tripos Filing Machine, a Front End to a File Server,
Proc. 9th ACM Symposium on Operating Systems Principles,
ACM Operating Systems Review, Vol 17 - No 5, October 1987.

[Schwartz 77] Schwartz, M.
Computer Communication Network Design and Analysis,
Prentice-Hall, 1977.

[Svobodova 76] Svobodova, L.
Computer Performance Measurement and Evaluation Methods: Analysis and Applications, Elsevier, 1976.

[Sze 85] Sze, D. T. W.
A Metropolitan Network,
IEEE J. on Selected Areas in Communications, Vol 3 - No 6, Nov. 1985.

[Temple 84] Temple, S.
The Design of a Ring Communication Network,
PhD.Dissertation,
University of Cambridge Computer Laboratory TR 52, January 1984.

[Tennenhouse 84] Tennenhouse, D. and I. M. Leslie
Universe Tunnels - Service and Protocol Specifications,
Project Universe Working Paper UP/473.1,
University of Cambridge Computer Laboratory, August 1984.

[Tennenhouse 85] Tennenhouse, D.
The Unison Exchange Architecture,
Project Unison Working Paper UC010,
University of Cambridge Computer Laboratory, September 1985.

[Tennenhouse 86a] Tennenhouse, D.
Unison Secretary Service,
Project Unison Working Paper UC006,
University of Cambridge Computer Laboratory, April 1986.

[Tennenhouse 86b] Tennenhouse, D.
The Unison Data Link Protocol Specification,
Project Unison Working Paper UC022,
University of Cambridge Computer Laboratory, September 1986.

[Tennenhouse 86c] Tennenhouse, D.
The CFR Unison Data Link Service Specification,
Project Unison Working Paper UC021,
University of Cambridge Computer Laboratory, October 1986.

[Thomas 85] Thomas, R. H. et al
Diamond: A Multimedia Message System Built Upon a Distributed Architecture,
IEEE Computer, Vol 18 - No 12, December 1985.

[Thurber 74] Thurber, K. J.
Interconnection Networks - A Survey and Assessment,
Proc. 1974 National Computer Conference, AFIPS Press, Arlington, 1974.

[Turner 86] Turner, J. S.
Design of an Integrated Services *Packet* Network,
IEEE J. on Selected Areas in Communications, Vol 4 - No 8, Nov. 1986.

[Want 88] Want, R.
Reliable Management of Voice in a Distributed System,
PhD.Dissertation,
University of Cambridge Computer Laboratory TR 141, July 1988.

[Waters 82] Waters, A. G. and I. M. Leslie
Universe Bridges - Implementation Specification,
Project Universe Working Paper UP/95.3,
Rutherford Appleton Laboratory, Oxford, July 1982.

[Waters 84] Waters, A. G. (Ed.)
Project Universe: Satellite Bridge,
Project Universe Report No.17,
Rutherford Appleton Laboratory, Oxford, 1984.

[White 87] White, P. E. et al (Ed.)
Switching Systems for Broadband Networks,
IEEE J. on Selected Areas in Communications, Vol 5 - No 8, Oct. 1987.

[Wilkes 79] Wilkes, M. V.
The Cambridge Digital Communication Ring,
Local Area Communication Networking Symposium, May 1979.

[Wu 86] Wu, L. T. and N.-C. Huang
Synchronous Wideband Network -
An Interoffice Facility Hubbing Network,
Proc. IEEE Int. Zurich Seminar on Digital Communications, March 1986.

[Wu 87] Wu, L.T., S. H. Lee, and T. T. Lee
Dynamic TDM - A Packet Approach to Broadband Networking,
International Conference on Communications (ICC 87), Seattle, June 1987.

[Zafirovic 88] Zafirovic-Bukotic, M. and I. G. Niemegeers
Performance Modelling of the Cambridge Fast Ring Protocol,
Proc. IEEE Int. Zurich Seminar on Digital Communications, March 1988.