A Dynamic Connection Management Scheme for Guaranteed Performance Services in Packet-Switching Integrated Services Networks.†

Colin J. Parris and Domenico Ferrari.

Tenet Group
Computer Science Division, UC Berkeley and
International Computer Science Institute, Berkeley, CA 94720

ABSTRACT

With the demand for multimedia and computational science applications, guaranteed performance communication services have become a necessary feature of future high-speed networks. These communications services should possess a high level of sophistication so that they can easily adapt the network to the wide variety of applications soon to be seen, thereby allowing the network to increase its *availability* and *flexibility*. Availability is the ability of the network to accommodate as many real-time clients as possible without violating any client's performance guarantees, while flexibility is the ability to adapt to changing network state and client demands in order to maintain the performance guarantees and quality of service promised to the client. Flexibility also refers to the ability of the network to easily increase the variety of real-time services that it offers. It is our contention that availability and flexibility can be enhanced in a network by providing the network with the ability to modify the performance parameters and/or the route of any guaranteed performance connection in the network without violating the previously made performance contracts.

In this paper, we present a scheme for dynamically managing guaranteed performance service connections and experimental results to verify the correctness and usefulness of the scheme. The motivation for this scheme, Dynamic Connection Management (DCM), is discussed, and detailed descriptions of the DCM modification contracts and algorithms are provided. A survey of guaranteed performance services protocols, architectures, and routing algorithms are presented, together with their relevance to this work. A simulator has been built, and preliminary experiments and analyses were done on the scheme. The paper concludes with a summary and some topics for future work.

†This research was supported by the National Science Foundation and the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives, by AT&T Bell Laboratories, Hitachi, Ltd., Hitachi America, Ltd., Pacific Bell, the University of California under a MICRO grant, and the International Computer Science Institute. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing official policies, either expressed or implied, of the U.S. Government or any of the sponsoring organizations.

1. Introduction.

With the dramatic increase in multimedia and computational science applications, the development of guaranteed performance services (i.e., *real-time services*) to support these applications is essential. Multimedia is the current term used to encompass a variety of applications including, but not limited to, digital audio and video, data navigation, electronic whiteboards, computer-supported collaborative workstations (CSCW), and high-density animation and imaging systems. These applications exhibit a broad spectrum of traffic characteristics and desire an equally broad spectrum of performance requirements.

Computational science applications are applications that do large distributed computations usually involving several supercomputers (e.g. applications that simulate the climatic conditions of the earth, or the flow of air over an aircraft's wing). Each supercomputer simulates a specific part of the experiment and passes its result, usually a large data set, to the other supercomputers, which use these results to continue their simulations [Catl92]. To be effective these supercomputers must pass these large data sets among themselves within a specified time interval. The traffic characteristics and the performance requirements of these services are quite broad.

To accommodate the need for these real-time services, various protocols have been proposed that provide performance guarantees usually in terms of throughput, delay, delay jitter, and packet loss rate. Most of these protocols provide some subset of the performance spectrum specified above [AnHs90], [Zhan90], [Topo90], with only one providing the entire spectrum [FeVe90a]. As new applications are constantly being developed and the communication requirements of these applications are constantly changing, it is important that the real-time services provided by these networks be manageable in such a manner that the services can quickly and efficiently adapt to the needs of the applications. Management of these real-time services is an important problem to be addressed in future high-speed, integrated services networks.

While there are many desirable goals to be achieved by proper management, of greatest interest are those of *availability* and *flexibility*. Availability in this sense pertains to the ability of the network to accommodate as many real-time clients as possible without violating any of the clients' performance guarantees. Flexibility refers to the ability of a network to adapt to changing network state and client demands in order to maintain the performance guarantees and quality of service promised to the client. Flexibility also refers to the ability of the network to easily increase the variety of real-time services it offers.

In order to support increased availability and flexibility in a real-time integrated services packet-switching network, a Dynamic Connection Management (DCM) scheme has been developed. This scheme has been implemented in the context of the Tenet real-time protocol suite [FeBa92] and can be used to increase the availability and flexibility of real-time packet-switching networks.

DCM achieves the goals of availability and flexibility by providing the network with two powerful capabilities. The first capability is the ability to modify the traffic and performance parameters of a real-time connection without violating the client's contract. This modification can be made transparently to the client; however, in most cases (in accordance with the client wishes), the client is informed. The second capability is the ability to transparently reroute a real-time connection without violating the client's contract. These two capabilities allow the network to increase its availability and flexibility.

Network availability is increased as rerouting can adjust the network load in hindsight, thereby allowing clients a greater access to the real-time services. Essentially, rerouting can allow the network to rearrange its resources so that *resource fragmentation* is lessened. Network flexibility is increased as the network is able to utilize unused resources dynamically in order to increase or maintain the quality-of-service contracts made to clients. A simple example of this is seen in a network with a "lossy" link. This lossy link may be causing a violation of the client's reliability contract. In this case the network can reroute the connection to ensure that the client's contract is not violated.

The modification of parameters is especially important for clients desiring to modify their throughput, delay, or delay jitter on demand. An example of this can be seen with environmental scientists who need to browse large uncompressible still images. The throughput of a connection can be dynamically increased to accommodate the browsing of these large images. This ability is also useful in the management of multicast channels where the addition of new members with different quality of services may necessitate a movement of the multicast tree and a modification of the performance parameters on the links connecting the new member to the channel.

This paper presents a detailed description of the DCM scheme. In Section 2 the motivation for this research and a survey of related work is presented. The modification contracts and algorithms that comprise the scheme are described in detail in Section 3 along with an overview of the Tenet scheme. Preliminary simulations and their analyses are given in Section 4. Section 5 provides a summary and topics for future work.

2. Motivation and Related Work.

A. Motivation.

As mentioned in the previous section, providing the network with the ability to modify a connection's performance parameters and route will increase the network's availability and flexibility. A few examples of these availability and flexibility problems are given below .

An example of an availability problem is shown in Figure 1. In this diagram three real-time connections are present from the source **S** to the destination **D**, two connection along the lower path and one along the upper path. These connections, A and B, along the lower path require a bandwidth of 45 and 30 Mbps, respectively, while that along the upper path, connection C, requires 50 Mbps. For the sake of simplicity only one performance parameter is guaranteed (i.e. throughput performance); however, this example can be applied to other performance parameters such as buffer space and delay. The current network state seen in the example is often caused by the routing of connections in a load balanced manner or the successive connection establishment and teardowns. With the network in this state a client requests a connection from **S** to **D** with a bandwidth of 60 Mbps. While the network has an aggregate available bandwidth of 75 Mbps, it cannot accommodate the request as this aggregate bandwidth is *fragmented* across both paths. The solution enabled by providing the network with the ability to reroute the channel is shown in Figure 2.

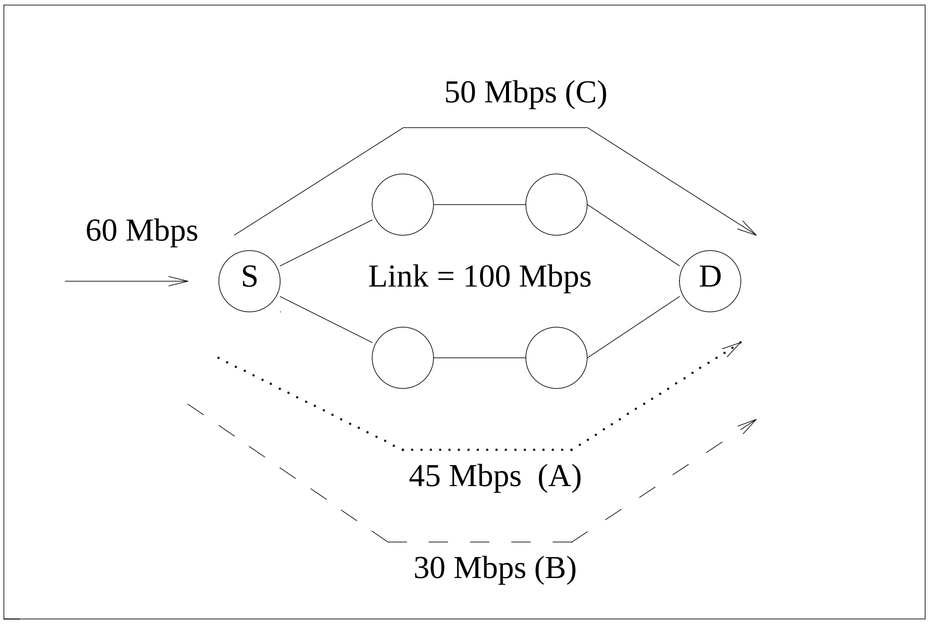


Figure 1.

In this solution connection A (the 45 Mbps connection) is dynamically rerouted to the upper path in a manner that is completely transparent to the client at both the source and the destination. After this rerouting, 70 Mbps are available along the lower path which can easily accommodate the new 60 Mbps connection.

Another example scenario that can be posed is that of a client wishing to increase its quality of service. The network state is as shown in Figure 1; however, there is no new connection requesting service (i.e. there is no 60 Mbps request at this time). The client responsible for connection B (the 30 Mbps connection) wishes to double its throughput. While there is ample aggregate bandwidth, the distribution of this available bandwidth will prevent connection B from acquiring the needed bandwidth. The solution in this case is to reroute connection A to the upper path and dynamically modify the channel parameters to permit the client to obtain the needed bandwidth.

The flexibility inherent in the network allows it to react to state changes to maintain service guarantees. An example of this can be seen in Figure 3, where a real-time connection between source **S** and destination **D** traverses a "lossy" or error prone link. This lossy link causes a large number of packet losses due to errors, thereby reducing the quality of service provided to the connection.[1] This link is shown with a large X in the figure.

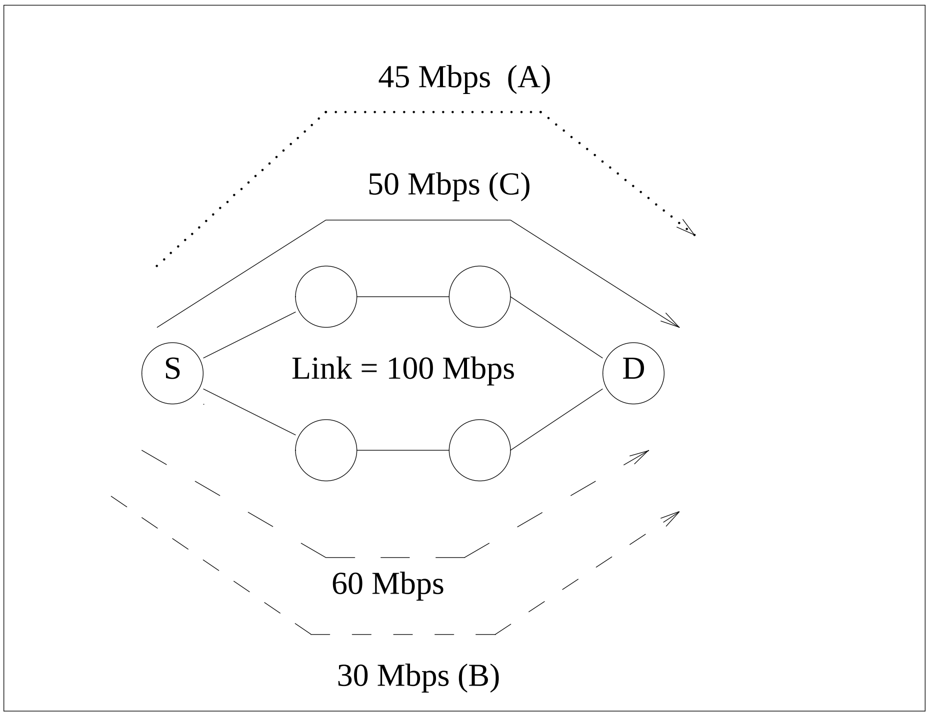


Figure 2.

While there is an alternate path that is available in the network, the network cannot utilize the path without asking the client to further disrupt the service and re-establish the channel along the new path. The network should be able to utilize this path dynamically without client participation. The solution provided by dynamic channel management is to transparently reroute the connection along the available lossless links, thereby maintaining the guarantees made to the client. This solution is shown in Figure 4.

There are many other examples that can be provided along the different (or multiple) dimensions of network resources. The previous examples have all been of unicast connections; however, multicast connections provide a rich environment for dynamic management. With multicast connections it is often the case that the addition of a new member to a multicast session already in progress produces a multicast tree that is not minimum-cost. This situation may occur even if the previous multicast tree was of minimum-cost and the branch connecting the new member to the tree was also minimum-cost. In these cases the main branches of the tree may be dynamically and transparently rerouted to produce the minimum-cost tree that includes the new member. In the case where different qualities of services are desired by the new members, connection parameters may have to be modified along main branches to ensure that these different qualities of services are met.

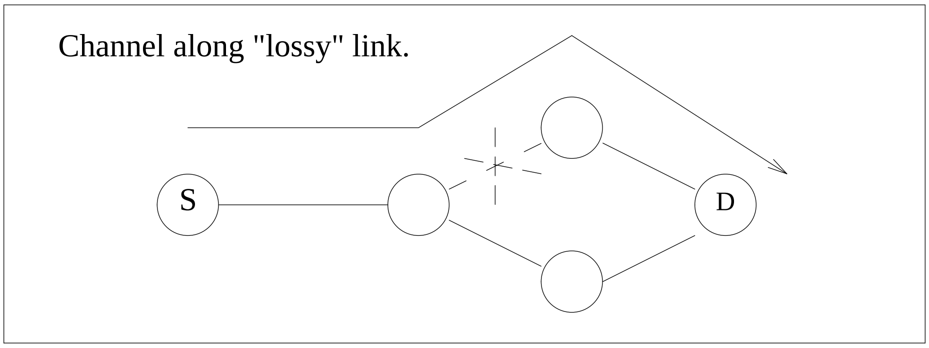


Figure 3.

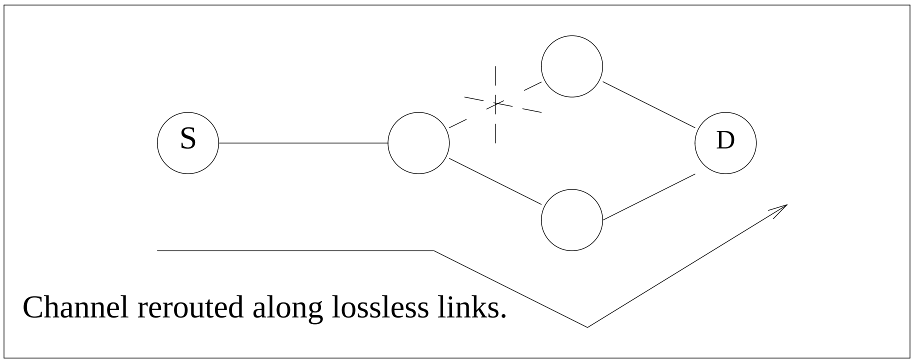


Figure 4.

B. Related Work.

There are a few guaranteed performance service schemes of which the authors are aware. Most of these services do not span the range of performance dimensions offered by the Tenet scheme[2] and very few support any type of network management. These guaranteed performance schemes are described below.

The Flow Protocol provides guarantees for average throughput bounds [Zhan90]. It does not provide guarantees for delay, delay jitter or loss bounds. While end-to-end delay bounds are not guaranteed, these delays can be somewhat reduced by overbooking throughput resources; however, the quantitative relationship between the percentage of resource overbooking and the delay reduction has yet to be determined. The Flow Protocol reserves resources and uses rate control to enforce a source's traffic characteristics. The *VirtualClock* scheduling discipline is used to ensure that the bandwidth reserved by a client is accessible to that client and to provide isolation from other clients. There are no features in this protocol for the dynamic modification of flows.

The Stream Protocol, Version II (ST-II) [Topo90] is an experimental internetwork-layer stream protocol developed for the Internet. ST-II decouples control from data delivery, as it uses the ST Control Message Protocol (SCMP) to establish and teardown connections. Connections can be unicast or multicast. Connections are established by sending a SCMP message to each hop along the route which determines if resources are available for the connection. The ST-II agents in the intermediate and destination nodes utilize the parameter flow specifications[3] to determine the availability of resources. Some of the parameters included in the specification are minimum allowable/deliverable bandwidth, maximum allowable/deliverable delay, delay variance, and so on. The specification does not detail any of the resource

[1] Or violating the performance guarantees if such guarantees were made.

[2] An overview of the Tenet Scheme is provided in Section 3.1.

reservation policies. The design of these policies is left to the implementor. The intermediate nodes modify the flow specification to reflect the available resources they have reserved. The destination can accept or reject a request for a connection by examining the modified flow specification and sending its response back to the source. If the source and destination are in agreement, the connection is established and data transport can begin. The SCMP specifications include a CHANGE message that can be used to modify parameters but no formal descriptions are given of this "CHANGE" functionality, and no implementations have been reported in the literature.

The Session Reservation Protocol (SRP) provides throughput and delay guarantees by reserving resources at each node [AnHS90]. The client is required to specify its traffic parameters using the DASH resource model. SRP creates sessions at each node, which represent reservations of a part of the capacity of a resource. These reservations are sufficient to guarantee the local throughput and delay requirements of the client. These sessions are then linked together along a fixed route to form an end-to-end session, which provides the end-to-end performance guarantees. The end-to-end session establishment and philosophy is similar to that of the Tenet scheme, but there are a few differences. As mentioned before, SRP uses a different traffic model and a different admission control policy, and the control and delivery functions are incorporated within the same protocol. Also, SRP aligns itself closely with IP in that it seeks to achieve performance guarantees for IP-based communication without changing the IP protocol and does not provide guaranteed delay jitter bounds, overflow bounds, or any statistical guarantees. SRP does not employ any mechanisms for the dynamic management of sessions.

The Asynchronous Time Sharing (ATS) approach provides a fixed menu of classes-of-services [LaPa91]. There are 4 classes of services. Three of the classes, Class I, II and III, transport user traffic while the fourth transports network management traffic. Class I traffic is characterized by zero percent contention[4] loss and an end-to-end delay distribution with a narrow spread. Class II traffic is characterized by *e* % contention packet loss and an upper bound, *n*, on the average number of consecutively lost packets. Its end-to-end delay distribution has a larger spread than the Class I distribution. The end-to-end delay of Class I packets is less than that of the Class II packets. Class III traffic is characterized by zero percent end-to-end packet loss. This is achieved by retransmissions. Real-time connections can only obtain performance guarantees corresponding to those of their class. The service discipline of ATS, MAgnet Real-time Scheduling (MARS), has its activity divided into cycles. A cycle is the period during which a fixed number of cells can be transmitted. This cycle is divided into four subcycles, each corresponding to a class-of-service and to the allocation of the link to this class-of-service. The duration of each subcycle is determined dynamically by the MARS scheduler according to the traffic load and the mix. While there is an extensive monitoring capability in the ATS approach, to our knowledge there is no ability to manipulate real-time connections.

The Multipoint Congram-oriented High-performance Internet Protocol (MCHIP) [PaTu90], is a guaranteed performance service that also provides multicast capabilities and may utilize resource servers to monitor and record channels established and resources available. To the best of our knowledge, its channel establishment procedure and resource reservation policies have not yet been published.

Clark, Shenker, and Zhang propose an Integrated Services Packet Network (ISPN) architecture that will support guaranteed and "predicted" performance services [ClSh92]. In their context, guaranteed bounds on throughput and delay can be provided to a client if the client specifies its maximum sending rate. For this rate the network will inform the client of the offered delay bounds; if the delay bounds is insufficient, the client must request a higher sending rate so that the delay requirement can be satisified. For predicted service, the client characterizes its traffic and the delay and loss rate it can tolerate. The admission control scheme determines if there are sufficient resources and stability in the network to accommodate the client. Due to network load fluctuations, predicted-service clients experience fluctuations in their performance specifications, to which they can choose to adapt to accommodate these perturbations. The network naturally does its best to ensure that there is high stability in its load. Clark et al. propose a *unified scheduling algorithm* capable of supporting both guaranteed and predicted traffic. The ISPN architecture

---

[3] There are 15 flow parameters, with the specification permitting applications to add extra fields.

[4] Contention packet loss represents packets that are discarded due to buffer overflow and packets whose end-to-end delay is greater than the maximum set limits.

does not address any of the network management issues, and provides no capability for channel modification.

Switched Multi-Megabit Service (SMDS) is a connectionless network service being provided by telecommunication carriers to offer high-speed data services between a customer's LANs, computer systems, and high performance workstations [DiKe90]. SMDS provides for the transport of a large variable-length Service Data Unit (SDU). Source address validation, multiple address assignment, source/destination address screening, and an addressing scheme consistent with the North American Numbering Plan are some of the features supported by SMDS. Currently SMDS defines two interfaces, based on DS1 (1.544 Mbps) and DS3 (44.736 Mbps), respectively, for access to a network supporting the service. While SMDS does not provide *hard* guarantees, it specifies an average end-to-end delay bound, for data traversing two DS3 access paths, of 20 msec for at least 95% of the packets[5]. For data traversing a DS3 and DS1 access path the average end-to-end delay is specified to be less than 75 msec for at least 95% of the packets, while for two DS1 access paths the average end-to-end delay is less than 130 msec. It should be noted that all of the performance specifications stated above apply to a single Local Access Transport Area (LATA). SMDS does not, to our knowledge, contain any features that permit the manipulation of the routes or of the performance of the services it supports.

3. Dynamic Connection Management (DCM).

The goal of DCM is to increase the availability and flexibility of real-time networks while maintaining the integrity of all of the performance contracts made with clients with real-time channels.[6] One method of achieving this goal is to be able to dynamically modify the performance and traffic parameters of the real-time channel and to be able to modify the route traversed by the channel. It is also our desire to be able to do these modifications in a manner that is non-disruptive to the client. In most cases the modifications will be transparent to the client unless the network deems it necessary for the client to know. This need to know is dictated by the DCM policy that is in effect.

DCM is comprised of the *DCM scheme* and the *DCM policies*. The DCM scheme is the collection of algorithms that permit the network to dynamically modify channel parameters and routes. The modification of a channel is a procedural abstraction whereby a real-time channel with the new performance parameters (referred to as the alternate channel) is established, the client's traffic is moved from the current real time channel (referred to as the primary channel) to the alternate channel, and then the primary channel is removed. The movement of traffic from the primary to the alternate channel is referred to as the *transition* from the primary to the alternate channel. This modification applies to both the performance and traffic parameters and the route. The DCM policies are *rules* that determine if a real-time channel is to be modified and the new values of parameters and/or the new route. These rules may examine the network or client state data to determine if modification should take place. The DCM policies are usually implemented as management applications.

The DCM scheme essentially consists of two algorithms:

- A *channel administration* algorithm, which determines if a real-time channel can be accepted along a specified alternate route and, if so, reserves the resources along the alternate route so that all of the client's traffic and performance requirements are met. These reserved resources also include the buffers to be used in the transition from the primary to the alternate channel. In addition, the algorithm ensures the correct handling of resources on any links that are common between the primary and alternate routes.
- A *transition* algorithm, which ensures that the transition from the primary to the alternate channel does not violate the client's modification contract. This is especially important in the case of the agreed-upon delay and delay jitter bounds.

A third algorithm in the DCM scheme, but in a supportive rather than essential capacity, is:

---

[5] Based on a maximum packet size of 9228 bytes.

[6] In this paper a *real-time connection* is referred to as a *channel*.

- A *routing* algorithm, which determines the most probable route that would satisfy the client's traffic and performance requirements. This algorithm considers the resources currently reserved for the primary channel when determining the alternate route.

These algorithms are to be encorporated into the Tenet scheme, and can be considered as extensions to it. As DCM is viewed as an extension the Tenet scheme, an overview of the Tenet scheme will now be provided.

3.1. The Tenet Scheme.

The Tenet scheme presents a method by which a network can provide real-time services, in the form of *real-time channels,* to the clients. A *real-time channel* is a communication abstraction that defines a communication service with guaranteed traffic and performance parameters in a packet-switched network. The performance guarantees in the Tenet scheme are those of bounds on *throughput*, *delay* , *delay jitter* , and *loss rate due to buffer overflows*. In [FeVe90] it has been demonstrated that a connection-oriented network architecture with admission control and resource reservation is needed in order to offer a real-time service. After satisfying these needs, packets from different real-time channels can still interact with each other, at the switches, in such a manner as to adversely affect the performance guarantees given to the clients. This adverse interaction can be eliminated by using an appropriate scheduling discipline at the switches. The Tenet scheme can be decomposed into three major areas: the service definition, the channel establishment, and the service discipline.

A. Service Definition.

The service definition is the interface that the network offers to its clients (in our case the clients are transport layer connections). With this interface the clients specify their traffic characteristics and their performance requirements [Ferr90]. In specifying their traffic characteristics, clients provide the peak and average rates of their traffic streams together with an indication of their burstiness using the following parameters:

- $X_{\min}$ - the minimum packet inter-arrival time.
- X_{ave} - the average packet inter-arrival time.
- I - the averaging interval.
- $S_{\max}$ - the maximum packet size.

The first three parameters belong to the set of positive real numbers, while the fourth parameter belongs to the set of positive integers. The performance requirements available to the client are:

- D - the maximum delay permissible from the source to the destination.
- J - the maximum delay jitter.[7]
- Z - the probability that the delay of the packet is smaller than the delay bound, D.
- W - the buffer overflow probability.

The delay violation probability specification allows the client to indicate if a *deterministic* or *statistical* channel is desired. If the value of Z is one, this indicates that the delay of the packet will always be less than the delay bound D. This is considered a *hard* or deterministic guarantee. If the value of Z is less than one, the channel is considered a *statistical* channel.

The *throughput* guarantee is obtained from the traffic characteristics of the client as the network, after accepting the channel, agrees to absorb the load produced by the client. The *delay*, *delay jitter*, and *loss rate due to buffer overflows* guarantees are obtained from the performance specifications as the network agrees to deliver packets within the specified performance parameters.

---

[7] In this case jitter is defined as the difference between the delays experienced by any two packets on the same connection.

B. Channel Establishment.

In the Tenet scheme a channel must be established in the channel establishment phase before the data can be transferred. During this phase, a real-time client specifies its traffic characteristics and its end-to-end performance requirements to the network. The network determines an appropriate route and divides these end-to-end traffic and performance requirements into local requirements for each node. These local requirements are then translated into resources that need to be reserved at each node.

In the Tenet scheme the resources reserved are bandwidth, buffer space, and schedulability. The total bandwidth reserved on any link for deterministic channels cannot exceed the bandwidth of that link. The buffer space needed to prevent buffer overflows for each connection is calculated on the basis of the maximum residence time of a packet in the switch and the maximum packet arrival rate. The schedulability resources also need to be reserved. Even without over-reservation of bandwidth resources it is possible to exceed the schedulability resource, as shown in [FeVe90]. A simple example of this is that of two packets with delay bounds of 5 units and service times of 3 units, which cannot be scheduled so that the delay bounds of both packets are met.

Resource reservations are attempted during a round-trip establishment communication. During the forward pass of the establishment communication, resources are reserved that give the greatest level of local performance (i.e., they provide a channel with the lowest delay, adequate bandwidth, and adequate buffers). During the reverse pass, the client's end-to-end requirements and the network's end-to-end availability are compared. If the possible end-to-end performance is greater than the requirements, the network relaxes its reservations to match the requirements of the client, thereby increasing its availability for future requests.

During this channel establishment, call admission and resource reservation algorithms are used to determine if the needed resources can be reserved and, if available, to reserve the resources, thereby ensuring that the guarantees made to the clients can be met. Resources are only reserved inasmuch as they do not cause the violation of the guarantees made to the other clients. These algorithms are based on the service disciplines at each switch; therefore, after the establishment phase, *a priori* end-to-end performance guarantees can be made to the client. The Tenet scheme is applicable to a heterogeneous internetworks running a variety of scheduling mechanisms[Ferr92].

C. Service discipline.

The service discipline is of fundamental importance in the Tenet scheme as it ensures that the traffic and performance guarantees made to the client are satisified at run-time. Both the call admission and the resource reservation algorithms are based on it. The service discipline is used to ensure that interactions between channels in the same switch and interactions of traffic between switches can be controlled in such a manner that clients' performance guarantees are not violated. The service discipline controls the flow of packets into the switches and the order in which the packets are scheduled to be sent on the outgoing link. This flow control or regulation is essential to ensure the throughput guarantees, while the scheduling ensures that the other performance guarantees are met. The Tenet scheme is general in that real-time guarantees can be provided with any starvation-free service discipline; however, some disciplines allow a much more efficient utilization of the network. A thorough description of various such disciplines can be found in [ZhKe91].

The scheduling discipline used in DCM is *Rate-Controlled Static Priority* (RCSP) queuing [ZhFe92]. It should be noted that the DCM channel administration algorithm can be adapted to any service discipline that can be used by the Tenet scheme; the choice of RCSP was based on the fact that RCSP queuing is an instance of a class of service disciplines that separate the functions of rate-control and scheduling. We believe that this class of service disciplines will allow the network to offer the greatest range of real-time services while simplifying admission control and implementation. This separation is especially useful as it decouples bandwidth allocation from delay allocation. A RCSP server is comprised of two components: a rate controller and a static priority scheduler. The rate controller shapes the input traffic from each channel into the specified traffic pattern by assigning an eligibility time to each packet; the static-priority scheduler orders the transmission of the eligible packets from all channels. A rate controller is a set of traffic regulators, each associated with a channel traversing the switch. The regulators in RCSP can be either *rate-jitter* or *delay-jitter* controlling regulators. A rate-jitter controlling regulator controls rate jitter by partially reconstructing the traffic pattern, while the delay-jitter controlling regulator controls the delay jitter by fully reconstructing the traffic pattern. The regulator achieves this by assigning to each packet an eligibility time and holding the packet until it becomes eligibile before passing it to the static-priority scheduler. In controlling the channels admitted into the switch and shaping traffic patterns to conform to the specified traffic requirements, the channels can obtain their previously specified throughput requirements.

The static-priority scheduler consists of a fixed number of prioritized real-time packet queues and a single non-real-time queue. Associated with each priority queue is a delay bound; the packets in the topmost priority queue (denoted as priority level one) have the lowest delay bound. A channel is assigned a priority level during the channel establishment phase, and that level is usually maintained for the duration of the session. By restricting the number of channels at each priority level using the admission tests, the queuing time of each packet at a priority level is guaranteed to be less than or equal to the delay bound associated with that level.

The scheduler services the packets as follows: the next packet transmitted is always the packet at the head of the highest priority non-empty queue, and non-real-time packets are transmitted only when there are no real-time packets in the priority queues. The transmission of a lower-priority packet is not preempted by the arrival of a higher-priority packet. A thorough description of this service discipline can be found in [ZhFe92].

D. Implementation.

The Tenet scheme has been implemented as the Tenet protocol suite. These protocols decouple the control and data delivery algorithms. There are two control protocols, two transport layer data delivery protocols and one network layer data delivery protocol. The control protocols are the Real-Time Channel Administration Protocol (RCAP) [BaMa91] and the Real-Time Control Message Protocol (RTCMP). The two transport layer data delivery protocols are the Real-Time Message Transport Protocol (RTMP) and the Continuous Time Message Transport Protocol (CMTP) [WoMo91]. The network layer data delivery protocol is the Real-Time Internet Protocol (RTIP).

3.2. DCM Modification Contract.

A client's request for modification is governed by a request/response paradigm that is depicted in Figure 5. In this figure the client makes a modification request (or the DCM policy determines that a route modification is needed and submitts a request) to the network at time T1, and the network attempts to satisify this request and returns a response based on the current network load. The response indicates if the request is *accepted or denied*. This response occurs some time later, such as time T2, at which point the client, if the request was client-initiated, can begin sending with the new traffic characteristics and expecting that the performances guaranteed for packets on both the primary and alternate channels are met. This is to say that packets traversing the primary channel will meet the performance guarantees corresponding to the primary channel, and packets traversing the alternate channel will have their alternate channel performance guarantees met. It should be noted that the primary and alternate channels exist simultaneously for a short interval of time, and then the primary is removed. This interval during which both the primary and alternate channels exist is called the transition interval. After this interval (i.e. at time T3), only the alternate channel will exist.

In DCM there are contractual obligations made to the client that determine the extent of the disruption that will be experienced by the client due to the transition from the primary to the alternate channel. There are two types of DCM modification contracts:

(1) *No performance guarantees will be violated* during or after the transition from the primary to the alternate channel.

(2) *A bounded number*[8] *of performance violations can occur* during or after the transition from the primary to the alternate channel.

There are four types of performance violations that can occur:

- Throughput Violations. - A throughput violation occurs when a client, sending traffic with the specified characteristics, experiences packet overflows due to the unavailbility of bandwidth.
- Delay Bound Violations. - A delay bound violation occurs when a client, sending traffic with the specified characteristics, has one or more packets that exceed the delay bound **D** at the destination node.
- Delay Jitter Bound Violations. - A delay jitter bound violation occurs when a client, sending traffic with the specified characteristics, has a packet that exceeds its delay jitter bound **J** at the destination node.
- Packet Ordering Violations. - A packet ordering violation occurs when a client, sending traffic with the specified characteristics, receives packets out of sequence. The out-of-order sequence could be as short as two packets.

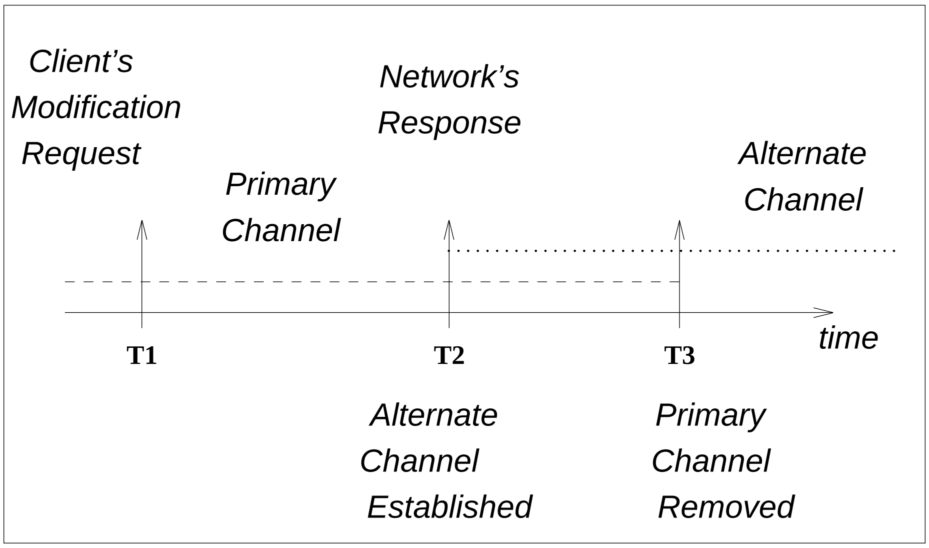


Figure 5.

These violations may occur singly, or multiple violations can occur simultaneously. The first type of DCM modification contract ensures that none of the four performance violations occur during or after the transition to the alternate channel. This contract may be explicitly requested by the client before channel parameter modification or implicitly demanded by the network during channel route modification. The client will seldom request a route modification as the client is usually not aware of the route. Route modification is usually done by the network and must be totally transparent to the client; hence there must be no performance violations. Reroutings may be done directly for network administrative or management purposes, or indirectly due to a client's performance parameter modification request. Administrative or management reroutings may be due to reclamation of resources by a subnetwork, to preventive maintenace on "lossy" nodes, or because of run-time maintenance being performed on "lossy" links in the network. Other management purposes such as the addition of a new member to a multicast channel can also cause a rerouting. Channels may be rerouted indirectly during performance parameter modification in the event that the resources needed for the alternate channel are best acquired along another route.

Performance violations are a result of the relationship between the traffic and performance parameters of the primary and alternate channels. If the modification contract chosen is that which ensures no

[8] The number of performance violations is specified as a packet count; however, it will be converted into a unit relevant to the client (i.e., frames, messages, etc.).

performance violations (i.e. contract (1)), then constraints must be placed on the alternate channel performance parameters so that none of the above mentioned performance violations occurs. Table 1 provides the constraints.

Table 1 - DCM Modification Contract (1)	
No Performance Guarantees Violated.	
Performance Violations	Constraints
Throughput	None
Delay	$D_{alternate} \geq D_{primary}$
Delay Jitter	$D_{alternate} \geq D_{primary}$
Packet Ordering	$D_{alternate} \geq D_{primary}$

During performance parameter modifications of a primary channel the constraints presented in Table 1 must be met if performance guarantees are not to be violated. In the case of throughput performance violations, there are no constraints on the alternate channel parameters, whereas constraints are needed to ensure no performance violations on delay, delay jitter, and packet ordering. If the delay bound requested by the primary route is greater than that of the alternate route, then situations can arise where delay, delay jitter, and packet ordering violations will occur. An example of this type of situation is described in detail in Sections 3.4.C and 3.4.D. If the delay bound requested by the primary route is less than or equal to that of the alternate route, then the DCM algorithms can ensure that no performance violations will occur.

While DCM Contract (1) constrains the ranges of parameter modifications so as to ensure no contract violations, some clients can tolerate the violation of these guarantees during a transition provided that the effect is bounded. The performance violations that can occur are those of delay, delay jitter, and packet ordering, and the violation bound is an upper bound on the number of packets exceeding their required delay or delay jitter bounds or the number of packets that are out of sequence after the client has received an alternate channel acceptance response from the network. This provision is useful, as we believe most clients to expect a slight disruption in service upon modification, and to be ready to accept it as long as it is bounded.

In the case of DCM Contract (2) (i.e. performance violations are bounded) we remove the parameter constraints and specify an upper bound on the number of packets that will exceed the delay or delay jitter bounds of the alternate channel or be out-of-sequence after the client has being notified of alternate channel acceptance from the network. This violation bound is the same for all performance violations, as it is expressed in terms of the maximum number of primary channel packets that are present during the transition from the primary to the alternate channel. The bound will be made known to the client before the channel modification is attempted, and the client can then decide if the modification is worth attempting. When $D_{alternate} < D_{primary}$, where $D_{alternate}$ is the delay bound specified for the alternate channel and $D_{primary}$ is the delay bound specified for the primary channel, then the network guarantees that at most $\lceil d_{primary}/X_{\min_{primary}} \rceil$ packets will exceed their delay bounds, where $d_{primary}$ is the sum of the transmission, the propagation, and the maximum queuing delays of the links along the primary route. Obviously $D_{primary}$ will always be larger that $d_{primary}$, and $D_{alternate}$ will always be larger that $d_{alternate}$[9]. Intuitively, this is a high upper bound, corresponding to the case in which all the packets remaining in the primary channel would be declared late; however, we can relax this bound after the alternate channel is established, when we know the actual value of $d_{alternate}$. The upper bound would then be $\left\lceil \frac{d_{primary} - d_{alternate}}{X_{\min_{primary}}} \right\rceil$ packets.

[9] $d_{alternate}$ is the sum of the transmission, the propagation, and the maximum queuing delays of the links along the alternate route.

3.3. The DCM Channel Administration Algorithm.

The DCM scheme has the same procedural format as the Tenet scheme, as it is an extension of this scheme. Channel establishment is accomplished with a round-trip communication message that contains the end-to-end information and the local network state that would enable an acceptance decision and, if appropriate, a reservation of the needed resources. The Tenet channel administration and resource reservation algorithms are responsible for channel establishment. In the DCM scheme, the Tenet channel administration algorithms have been supplemented by the DCM channel administration algorithm. In this paper, the DCM algorithms are described only for deterministic real-time[10] services.

The goal of the DCM channel administration algorithm is to establish an alternate route, conforming to the specified traffic and performance parameters, between a source and a destination in such a manner that the client's contract is not violated. This alternate route is established in the presence of a primary route on which the client is currently active. The establishment entails the decision as to the acceptance or rejection of the client's request subject to resource availability; in the presence of resource availability, the algorithm must reserve the appropriate resources such that an *a priori* guarantee is made.

The establishment of an alternate route begins with the acquisition of a route made by the DCM routing algorithm. After the alternate route is obtained, the end-to-end traffic and performance requirements are translated into local resource demands by the channel administration algorithms, and local channel admission tests are conducted to determine if this channel can be accepted at each node. If the channel can be accepted, the needed resources are reserved. As can be seen by the above procedure, the DCM algorithms can be implemented by modifying the RCAP protocol. This process continues along each node until the destination node is reached or an intermediate node rejects the channel. Initially, the local maximum possible resources are presented to the channel on this forward pass of the round-trip establishment communication. On the reverse pass, the destination node examines the clients performance requirements and the accumulated local resources and, if possible, reduces or *relaxes* the resources reserved in the nodes along the path so that only the necessary resources are given to the client.

When a node receives a channel establishment request, it performs the following tests:

a) the *bandwidth and scheduler resource test*;

b) the *buffer space resource test*.

In should be noted that passing a test will cause the resources to be immediately reserved until a message is received that forces their release. If both tests are successful, the node forwards the establishment message to the next node along the forward direction of the route after appending some of the parameter information from its tests. This parameter information is used by the destination node in the final acceptance tests to do the relaxation of resources. This relaxation of resources enables an equitable distribution of the resources along the path. This is needed as the the local channel establishment algorithms attempt to acquire as much of the resources as possible on the forward pass.

In applying these tests to establish an alternate route, one of two scenarios can be expected to occur. In the first scenario the alternate route and the primary route have no common links (i.e., they are totally disjoint) while in the second scenario there are common links[11]. In the second scenario the primary route and the alternate route may be identical. The impact of these scenarios on the tests are presented below.

A. Scenario 1: Completely Disjoint Alternate Route.

In this scenario the alternate route is completely disjoint from the primary route. This scenario is similar to that associated with the channel establishment of the primary route, with the exception of the implementation concern to be mentioned in Section 3.3.C. When the local node receives an establishment request, it determines if enough bandwidth and scheduler resources can be reserved to ensure the throughput and delay bound guarantees. These resources are reserved according to the result of Test 1 given below.

---

[10] These deterministic services have a deterministic delay bound and a buffer overflow probability of one.

[11] Common nodes do not require any special treatment because of the assumption by RCSP of infinite computational power. However, the DCM algorithms can be easily modified to support computational power as a limited resource.

Admission Control Tests.

Test 1: For a request with the traffic specification ($X_{\min}, X_{ave}, I, S_{\max}$) and a delay bound requirement d_k (where $d_1, d_2, d_3, .., d_n$ are the delays associated with each of the n priority levels in the switch), if the following condition holds

$$\sum_{j=1}^{j=c_k} \left\lceil \frac{d_k}{X_{\min_j}} \right\rceil * S_{\max_j} + \left\lceil \frac{d_k}{X_{\min}} \right\rceil * S_{\max} + S_{maxP} \le d_k * L \qquad (1)$$

for all priority levels k, $k+1$,.., n, then the alternate channel can be accepted. c_k is the current number of channels at and above priority level k, $X_{\min_j}$ is the minimum interarrival time of packets corresponding to the jth connection at that priority level , S_{maxP} is the largest packet size that can be transmitted over the link, and L is the link speed.

The first term in equation 1 uses the previously accepted channels to calculate the maximum number of bits that can be transmitted in d_k time for channels whose priority level is less than or equal to level k. The second term calculates the maximum number of bits that the new channel can transmit during the d_k interval, while the third term illustrates the worst case scenario where a packet has just had its transmission initiated. To ensure that all previous guarantees can be met and the new channel can be accepted under the worst possible conditions, the maximum amount of data that can be sent from the current clients and the new client, in the presence of an occupied transmission line, must be less than or equal to the capacity of the link. This condition has been stated and proved in [FeZh92].

The following local condition must be met to ensure that enough buffers are reserved so that the performance guarantees are not violated.

Test 2: For a request with the traffic specification ($X_{\min}, X_{ave}, I, S_{\max}$) whose delay bounds are $d_{i,s}$ where i refers to the priority level and s refers to the node along the path, the alternate channel can be accepted if

$$R_{bu} + \left\lceil \frac{d_{i,s-1}}{X_{\min}} \right\rceil * S_{\max} + \left\lceil \frac{d_{i,s}}{X_{\min}} \right\rceil * S_{\max} \le B \qquad (2)$$

($s = 1, .., p$; $d_{i,0} = 0$)

where R_{bu} is the current buffer space available (in bits) and B is the maximum buffer space (in bits) allotted in that node to that output link.

Note that the buffer space depends on the delay bound in the previous switch. This situation exists because the longest time a packet can stay in the regulator in switch s is $d_{i,s-1}$ (this occurs when the previous switch, $s-1$, has no other packets in its scheduler so it forwards this packet immediately), and the longest time that a packet can stay in the scheduler is $d_{i,s}$. Therefore the buffer space that is needed to ensure no overflows must consider both of these delays. It should be noted that the clients' delay jitter guarantees are entirely dependent on buffers as delay jitter guarantees are enforced by the regulators, which hold packets until their eligible time before passing them to the scheduler. To guarantee delay jitter bounds, the local delay in the last switch must be less than or equal to the delay jitter bound required by the client. The delays in the intermediate switches can then be relaxed as long as their sum is less than or equal to the needed delay bound.

B. Scenario 2: Partially Disjoint or Identical Alternate Route.

In this scenario the alternate route is not completely disjoint from the primary route, but contains one or more links that are part of the primary route. It is also possible that all links are common links. As in the previous scenario, when the local node receives an establishment request it determines if enough bandwidth and scheduler resources can be reserved to ensure satisfaction of the throughput, delay or delay jitter guarantees. If the link is not a common link, Tests 1 and 2 should be used to determine the local acceptance of the channel. If the link is a common link, then there are two possible actions to be taken in order to accommodate the alternate channel over this link. These actions arise from the fact that we are attempting to route two channels simultaneously over the same link. Although the primary channel resources will be released very quickly, for a very short period of time the sum of resources of the two channels may be needed on the common link as there may be packets in the primary and the alternate channel simultaneously on their way to the destination. This situation occurs when the local delay of the primary channel up to the common link is greater than the local delay along the alternate channel up to the same point. This is not an out-of-sequence problem but the sudden appearance of primary channel packets that were previously buffered in the network, together with the expected alternate channel packets, requires that excess bandwidth and scheduling resources be available to handle this "burst". We are essentially accommodating two channels at this point. While this may not be a problem as enough resources may be available[12], there are cases where multiple channels or a very large (i.e., resource demanding) channel are being rerouted, and the additional resources needed are not available. Special actions must be performed for channel establishment on a common link. These actions are as follows:

- If there are sufficient resources to accommodate both channels, then the alternate channel is created along this common link using Tests 1 and 2 in the same manner given above. Since the transition interval is small, it is to be hoped that these additional resources will not adversely affect the network in any significant manner.
- If there are insufficient resources to accommodate both channels, then Tests 3 and 4, given below, need to be executed in order to reserve resources for the larger of the two channels. These tests are a modification of Tests 1 and 2, and take into consideration resources that are already reserved for the primary channel. This ensures no duplication of resources, thereby giving the alternate channel a better probability of acceptance. If the primary channel has acquired resources that are greater than those of the alternate channel, these tests need not be applied. If Test 3 and 4 are successful, we have guaranteed that enough resources are available for the higher performance channel but not for both of the channels. Packets can now possibly by sequentialized along the routes so that packets from the primary channel encounter all of the common links at least $X_{\min_{alternate}}$ time units before packets from the alternate channel. This sequentialization is achieved by doing the appropriate delay relaxation at the destination node[13]. The delay relaxation function must ensure that the accumulated delay on the primary channel before any common point is less than or equal to the accumulated delay on the alternate channel at that point, and the local maximum queuing delay along the common link must be same on the primary and the alternate path.

If either of these actions can be done successfully, then the channel can be accepted, otherwise it is rejected.

Modified Admission Control Tests.

These modified tests are only applied to the common links if any or all of the performance requirements of the alternate channel are *greater* than those of the primary channel. When the conditions below hold, the performance needs of the alternate channel are *greater* than those of the primary channel.

- Throughput: $\frac{S_{\max_a}}{X_{\min_a}} > \frac{S_{\max_p}}{X_{\min_p}}$
- Delay: $D_a < D_p$
- Delay Jitter: $J_a < J_p$

The subscript p denotes the primary channel, while a denotes the alternate channel.

---

[12] Especially as these resources are only needed for a short period of time.

[13] It should be noted that this sequentialization occurs because delay jitter control is applied to all real-time channels. Delay jitter is applied as it reduces the buffer reservations needed in the network; however, the client is not informed that this jitter control is present.

To ensure that there are no performance violations due to the channel modification during a transition on a common link, it is necessary to retain ample resources on the primary channel such that the primary channel traffic can meet its obligations. It is easy to envision a situation where a smaller delay bound is required for an alternate channel with lower throughput. In this case the excess bandwidth is released on this common link and the additional delay resources are obtained. However, primary channel packets that have not yet reached this common link before the transition is completed will encounter reduced bandwidth reservations and most likely violate their performance contracts. This situation can be prevented by a judicious choice of parameters upon which resource reservation at this common link will be based. In Test 3 below, an adjustment will be made to virtually remove the resources currently reserved for the primary channel and to acquire the resources needed for the alternate channel. After this adjustment the resources must be available to ensure that the primary channel packets present on the primary route during the transition will not violate their contracts, and that the new alternate channel packets will have sufficient resources to meet their obligations. This is accomplished by choosing the appropriate $X_{\min}$ and $S_{\max}$ parameters from among those of the primary and alternate channels, and ensuring that on that common link the local delays corresponding to the alternate channel are always less than or equal to those of the primary channel. In the event that any performance index for the alternate channel is greater than the same index of the primary channel, the values of $X_{\min}$ and $S_{\max}$ to be used in the admission test, and the local delay conditions[14] are given by:

$$X_{\min} = \min\ (\ X_{\min_a}\ ,\ X_{\min_p}\ ) \tag{3}$$

$$S_{\max} = \max\ (\ S_{\max_a}\ ,\ S_{\max_p}\ ) \tag{4}$$

$$d_a^c \le d_p^c \tag{5}$$

If all of the performance indices of the alternate channel are less than those of the primary channel, no admission test need be applied, as sufficient resources have already been reserved for the alternate channel. As the resources reserved by Test 3 below, ensure that the primary channel packets as well as the alternate channel packets meet their obligations, they may be in excess of those needed for the alternate channel. These excess resources are only present during the transitional period, and will need to be recovered by the network upon the release of the primary channel for use by other clients.

Test 3: For an alternate channel request with the traffic specification ($X_{\min_a}, X_{ave_a}, I_a, S_{\max_a}$) and a delay bound requirement d_{k_a}, and a primary channel with specification of ($X_{\min}, X_{ave}, I, S_{\max}$) and a delay bound d_m, first the resources are adjusted to "virtually" remove the primary channel and then Test 1 is applied:

Adjustment:

$$R_{ba,sh_k} = \sum_{j=1}^{j=c_k} \left\lceil \frac{d_k}{X_{\min_j}} \right\rceil * S_{\max_j} - \left\lceil \frac{d_k}{X_{\min}} \right\rceil * S_{\max} \quad \text{for } k = m\ ,\ m+1\ ,..,\ n$$

where k is the priority level, and c_k is the current number of channels at and above priority level k.

If the condition given below can be met:

$$R_{ba,sh_k} + \left\lceil \frac{d_k}{X_{\min}} \right\rceil * S_{\max} \le d_k * L \quad \text{for } k = k_a\ ,\ k_{a+1}\ ,..,\ n\ (\ k_a < m\ )\ ,.$$

where $X_{\min}$ and $S_{\max}$ are as given by equations (3) and (4) above, then the alternate channel can be accepted provided the condition specified by equation (5) holds.

---

[14] The superscript *c* indicates the local delay in the switch corresponding to the common link *c*.

The buffer resource test modification is of the same form as that of the bandwidth and scheduling test above, but an adjustment must be made to ensure that there is no duplication of previously reserved resources. Again, the reserved resources are adjusted and the test condition applied to the adjusted resources. As before, the values of $X_{\min}$ and $S_{\max}$ and the delay condition provided by equations (3), (4), and (5) are used to in this test.

Test 4: For an alternate channel request with the traffic specification ($X_{\min_a}, X_{ave_a}, I_a, S_{\max_a}$ *) and a delay bound requirement* d_{k_a}*, and a primary channel with a traffic specification of (* $X_{\min}, X_{ave,} I, S_{\max}$ *) and a delay requirement* d_m*, the adjustment is:*

$$R_{bu_{adj}} = R_{bu} - \left\lceil \frac{d_{i,s-1}}{X_{\min}} \right\rceil * S_{\max} - \left\lceil \frac{d_{i,s}}{X_{\min}} \right\rceil * S_{\max} \ .$$

The condition that needs to be satisfied is :

$$R_{bu_{adj}} + [\left\lceil \frac{d'_{i,s-1}}{X_{\min}} \right\rceil + \left\lceil \frac{d'_{i,s}}{X_{\min}} \right\rceil] * S_{\max} \leq B, \quad (s = 1 ,.., p \ ; d_{i,\,0} = 0 \ )$$

where R_{bu} *is the current buffer space in use (in bits), B is the maximum buffer size (in bits) at that node allocated to the output link,* $d_{i,s}$ *is the delay at switch s along the primary path , and* $d'_{i,s}$ *is the delay at switch s along the alternate path.*

Note also that this modified test is only performed if the performance indices of the alternate channel are greater than those of the primary channel. In the case where the performance requirements are less restrictive, no resources are released during the establishment phase; rather, the excess resources are reclaimed during the teardown of the primary channel.

C. Switching from the Primary Channel to the Alternate Channel.

To make the transition from the primary route to the alternate route "invisible" to the client, two concerns must be addressed.

(1) How is the channel interface to the client preserved ?

(2) How is the switching from the primary to the alternate channel accomplished?

While both of these concerns are implementation issues, a look at the framework of their solution will provide insight into some of the problems addressed by the transition algorithm.

First, the channel interface to the client must be preserved to allow for an "invisible" transition. This interface is usually in the form of a unique identifier (i.e. source host IP address and source local channel ID). This unique identifier must be preserved at the source and a similar identifier must be preserved at the destination. After the transition, the unique identifier will refer to the alternate channel, and the client’s packets will be sent along this alternate route, while at the destination the receiver will continue to receive its packets from its usual receiver abstraction (e.g., its usual, unique UNIX socket).

The second concern involves the switching from the primary to the alternate path. The transfer to the alternate path, while maintaining the same unique source and destination channel identifiers, is done during the alternate channel establishment. On the forward pass, each intermediate node creates an instance of a channel structure which stores the state of the channel in that node, and an entry into a virtual circuit routing table that indicates the outgoing virtual circuit identifier and the outgoing link. Along the alternate route, the node preceding the destination configures its virtual circuit routing table so that it points to the same entry as the node preceding the destination in the primary path. In this manner both the primary and alternate channels point to the same destination virtual circuit entry. Note that this action is taken regardless of the conditions of the routes (i.e. whether the alternate and primary routes are completely disjoint, partially disjoint , or identical), as new channel structures and new table entries are required in all conditions. On the reverse pass the switching between the primary and alternate channel takes place when the virtual circuit table entry corresponding to the source is changed to point to the alternate route. This change is accomplished by modifying the outgoing virtual circuit identifier and the outgoing link identifier. The previous outgoing virtual circuit and outgoing link identifiers are maintained and used to teardown the primary channel. A more thorough description of these implementation details can be found in [PaZF92].

3.4. The DCM Transition Algorithm.

The DCM transition algorithm ensures that the transition from the primary to the alternate channel does not violate the DCM modification contract.

A. Throughput Bounds.

The throughput bounds should never be violated, as a modification of these bounds is usually done with the full participation of the client. Therefore, the client is aware of the modification and is notified immediately when the alternate channel is available. As the switch occurs from the primary to the alternate route, the regulator at the source controlling the flow of traffic into the network has already reserved the appropriate resources for this new throughput. With the client's awareness of the new throughput bounds and the regulator's enforcing of this bounds, violations cannot occur unless the client is misbehaving. We assume that the client will have some mechanism to perform an immediate switching from the primary to the alternate channel. This mechanism may involve local client buffers to handle the packet overflow that may occur if the throughput bounds of the alternate are less than those of the primary channel.

B. Buffer Overflow Bounds.

As we are concerned with deterministic channels, packet losses due to buffer overflows will not occur. Buffer overflow bounds are explicitly linked to the delay and delay jitter bounds in that sufficient buffers will always be maintained at each switch to accommodate for the maximum residence time in that switch. The buffers maintained along the primary and alternate routes reflect the maximum delay along each route, respectively. Therefore, as the maximum possible buffers are reserved on both routes, during the transition there is no possibility that a packet will be lost due to buffer overflow.

C. Delay Bounds.

To ensure that a given delay bound is achieved, we must consider the two instances when $D_{primary} > D_{alternate}$ and when $D_{primary} \le D_{alternate}$. If $D_{primary} > D_{alternate}$, then channel modification may not be possible, unless the client is willing to tolerate bounded performance violations. A simple example will illustrate this point. Consider the scenario shown in Figure 6: the client had initially requested a channel with a $D_{primary}$ of 70 msec and had received a channel whose actual end-to-end delay $d_{primary}$ was 60 msec. On requesting an alternate channel whose new delay bound, $D_{alternate}$, was 30 msec, an alternate channel with an actual end-to-end delay, $d_{alternate}$, of 20 msec was provided. These actual delays $d_{alternate}$ and $d_{primary}$ are the maximum delays that packets corresponding to these channels experience. These delays will always be less than or equal to the delay bounds guaranteed to the clients. If we assume that the client's traffic for both the primary and the alternate channel is characterized by $X_{\min} = X_{ave} = 20\ msec$, then a traffic flow starting at time 0 has packets leaving at time 0, 20 , 40, 60, .. msec. If we assume that all packets experience the maximum delay possible along their route then, packet 1, leaving at time 0 along the primary route will arrive at the destination at time 60. If we assume that the switch from the primary to the alternate route takes place in the interval of time between 1 and 19, then packet 2 will arrive at the destination at time 40 msec and will be due at time 50 msec. There will be no way that we can pass packet 1 to the destination before packet 2 without violating the deadline of packet 2. Hence in this case we can meet the delay bound guarantees if the packets are delivered out-of-sequence (thereby violating the packet ordering guarantee) or we can deliver the packets in sequence but exceed the delay bound (thereby violating the delay bound guarantee).

If the client chooses the bounded performance violation contract (i.e. DCM contract (2)), then a performance violation bound equal to 3 packets[15] will be presented to the client. Therefore, at most 3 packets will be delayed or out-of-sequence depending on the manner in which packets are delivered to the destination. After the alternate channel establishment, a more restrictive performance violation bound can be determined using the value of $d_{alternate}$. In this case the new bound is 2 packets, and by going through the example, it can be seen that, if maximum delays are experienced, only 2 packets will be out-of-sequence.

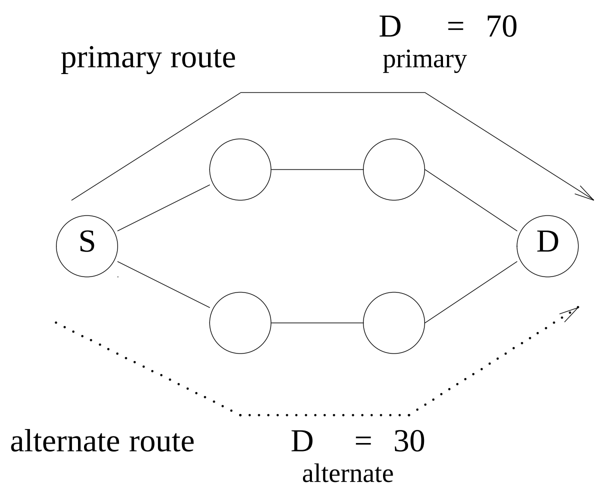


Figure 6.

Given the above mentioned scenario, it is possible that the actual delay of the primary channel may be such that $d_{primary} \le D_{alternate}$ even if $D_{primary} > D_{alternate}$, thereby making it possible to transition from the primary to the alternate channel without violating performance contracts. Therefore this first case of $D_{alternate} < D_{primary}$ may or may not cause performance violations. The values of $d_{primary}$ and $D_{alternate}$ will need to be examined in order to determine the feasibility of this channel modification. If $D_{primary} \le D_{alternate}$, it is possible to accommodate a channel modification in all cases regardless of the actual delay values experienced along the routes, provided that a packet resequencing mechanism exists at the destination node.

As all real-time services guarantee in-order packet delivery, it may be necessary to resequence the packets arriving at the destination during the transition interval. A simple example of packets arriving out of sequence can be found in this scenario: a packet is sent by the client just before the transition from the primary to the alternate route, and this packet experiences a maximal delay $d_{primary}$; the next subsequent packet which will travel along the alternate route experiences a maximal delay $d_{alternate}$ along that route; if $d_{primary} > d_{alternate}$, then the packet traveling along the alternate route will arrive at the destination before the packet traveling along the primary route and will be passed to the destination process out of sequence. This situation may arise due to the fact that the alternate route utilization is lower, or that the alternate route is shorter (i.e. it has less hops). As the propagation delay in the network act as a buffer, there must be some additional buffering at the destination to compensate for a possible reduction delay along the alternate route during a transition interval. This additional buffering is to be used to buffer packets on the alternate channel until the packet that had been sent on the primary channel has arrived. The buffered packets from the

[15] $\left\lceil \frac{d_{primary}}{X_{\min}} \right\rceil = \left\lceil \frac{60}{20} \right\rceil = 3$

alternate channels will be held until the outstanding packets on the primary channel have arrived or just prior to the violation of the alternate channel delay bound, $D_{alternate}$. In this manner no delay bound would be violated, as the outstanding primary route packets would have arrived and the other buffered packets would be passed up, in sequence, to the destination. In the event that the outstanding primary route packets did not arrive (this situation would only arise if packets were lost due to network errors or failures), the buffered alternate route packets would then be passed up to the destination just before the expiration of their respective deadlines.

An upper bound on the amount of buffer space needed to guarantee a violation-free transition from the primary to the alternate route is given by the following equation:

$$TB = \left\lceil \frac{d_{primary}}{X_{\min_{alternate}}} \right\rceil * S_{\max_{alternate}} - \left\lceil \frac{d_{alternate}}{X_{\min_{alternate}}} \right\rceil * S_{\max_{alternate}} \text{ if } \quad d_{primary} > d_{alternate}$$

$$TB = 0 \text{ if } \; d_{primary} \le d_{alternate}$$

where TB is the size of the required transitions buffers.

The transition buffers attempt to buffer all of the packets from the alternate channel that could arrive at the destination before the last packet along the primary channel has been delivered. The first term of the above equation is the buffer space needed to buffer all of those alternate channel packets, while the second term is the amount of buffers that have been reserved for use by the alternate channel along this route. Therefore the difference between the amount of buffer space needed and that reserved is the transition buffers required for this channel.

The maximum duration that these buffers will be held is given by:

$$duration = d_{estb} + d_{primary} + d_{processing}$$

where d_{estb} is the time taken to complete the reverse pass of the channel establishment message, and $d_{processing}$ is the time it takes the client to process the packets in the buffers. d_{estb} is included in the duration as it is the channel establishment message that causes the destination node to acquire these buffers for the transition. In the worst case, if just prior to switching to the alternate channel a packet was sent out on the primary channel, then this packet will take $d_{primary}$ time units to get to the destination. The final component of the duration $d_{processing}$, is the time it will take the client to empty all of the transition buffers.

In the case of bounded performance violations, the transition buffers will be reserved, and alternate channel packets will be held for as long as possible (so as to allow the maximum number of primary channel packets to arrive) without violating their delay bound. After the first alternate channel packet is passed to the destination, all of the remaining primary channel packets arriving there are discarded. Since the performance violation bound is the maximum number of primary channel packets present in the network during the transition, the number of packets discarded will always be less than or equal to this violation bound.

D. Delay Jitter Bounds.

The channel administration algorithm makes the delay at the last switch equal to the delay jitter bound. In this situation delay jitter controlling regulators are employed to completely reconstruct the traffic pattern; therefore, the only delay jitter that can be experienced is that due to the queuing at the last node. Therefore, if we bound the queuing time at the last node, we can bound the jitter. If $J_{alternate} \neq J_{primary}$, contract violations can be avoided by maintaining a delay jitter equal to $\min(J_{primary}\ , J_{alternate})$, as this value will ensure that the channel with the more stringent delay jitter bound meets the requirement, and the other, less stringent channel will be provided with much better than demanded performance. Additional transition buffers will be used to ensure that performance guarantees are not violated during the transition. All out-of-sequence packets along the alternate route will be buffered at the destination and passed up to the client at the appropriate time. In the previously discussed delay bound case, upon arrival of all packets on the primary route the buffered out-of-sequence packets are all passed up to the client immediately. However, to preserve the exact traffic pattern in the delay jitter bound case, the out-of-sequence packets will be passed up to the scheduler at the appropriate times. This appropriate time, t_i , for packet i is $t = src_i + D_i - J_i$ where src_i is the source timestamp on the packet, and D_i and J_i are the delay and delay jitter bounds on the channel. As mentioned previously, if DCM Contract (1) is desired, the delay bound conditions must be met to ensure that there are no delay jitter performance violations. With bounded performance violations, the bound for the delay jitter is the same as that for the delay.

3.5. The DCM Routing Algorithm.

The DCM routing algorithm is designed to find an *optimal* route based on the constraints imposed by the traffic characteristics, the performance and administrative requirements, and the source/destination pair. The definition of *optimal* is given in Section 3.5.2. While this algorithm is not essential to the function of DCM, it is important for the efficient operation of the DCM channel administration algorithm.

The DCM channel administration algorithm obtains a route for the specified source/destination pair by requesting a route from the DCM routing algorithm that obeys the specified routing constraints. In requesting this route the values of various traffic, performance, and administrative parameter sets are required by the routing algorithm. These parameter values are used to ensure that the network can guarantee the performance and administrative specifications of the client as well as the traffic characteristics along the route obtained.

The routing algorithm takes the following parameter sets:

- the primary channel parameter set,
- the alternate channel parameter set,
- and the administrative parameter set.

The *primary channel parameter set* is a parameter set that contains the traffic characteristics[16], the performance characteristics [17], and the channel path (specified by the links and nodes of the primary channel). In the case where a primary route is desired, the channel path is unspecified and will be filled in by the routing algorithm. If an alternate route is desired, the path information is completely filled in. The *alternate parameter set* is identical to the primary parameter set, but contains the alternate channel's request information. The *administrative parameter set* contains a route type parameter and an array of network link identifiers. The route type parameter indicates if the new route requested is the route of a primary channel or that of an alternate channel. If the new route requested is that of an alternate channel, then the type may specify whether the route should be completely or partially disjoint from the route of the primary channel. In the event that the parameter indicates a partially disjoint route, the links to be avoided are specified in the array of link identifiers. The routing algorithm uses these traffic, performance, and administrative parameters to determine an optimal route from the source to the destination host. It then passes this route to the channel establishment protocol which attempts to set up the channel using the channel administration algorithm.

The routing algorithm calculates an optimal route based on the queueing, transmission, and propagation delays of the links. It essentially determines a shortest path that meets the D requirement. This is accomplished by creating a directed graph and applying a constrained shortest path algorithm to this graph. The weighted edges of the graph represent the maximum total delay (i.e. the sum of the queueing, transmission, and propagation delays) on the link, while the nodes represent switches and hosts. The route type parameter causes the algorithm to adjust the edge weights of the graph appropriately. The route type parameter can be chosen from the set $\{0, 1, 2, 3\}$.

If the route type is 0, the routing algorithm assumes that a primary route is needed and ignores the alternate channel parameter set and the link array. It uses the source/destination fields of the primary channel parameter set and determines a "minimal" route that meets the end-to-end delay performance requirements from the source to the destination. It also ensures that the traffic characteristics can be accommodated along this route by using the traffic characteristics to determine the maximum queueing delay

---

[16] $X_{\min}, X_{ave}, I, S_{\max}$

[17] D, J, W, Z

experienced at a node. This maximum queueing delay is used in determining the maximum delay experienced along this link. If the route type is 1, the algorithm determines an alternate route that is completely disjoint from the primary route. This route is determined by removing[18] the links along the old route from the graph and then applying the constrained shortest-path routing algorithm. With a route type parameter of 2, the algorithm assumes that a partially disjoint route is needed. The links that are to be avoided in the alternate route are obtained from the link array and are removed from the graph. The links that are not removed have their resources adjusted so that the resources consumed by the primary channel are *virtually removed* before the optimal route is computed. This virtual removal is done in the same manner as the bandwidth and scheduler adjustments made by the administration algorithm. This ensures that resource duplication does not prohibit the use of routes that would normally be available. A type 4 route is an alternate route with no restrictions. In this case the primary route resources are virtually removed, and the optimal route that meets the traffic and performance requirements is computed. The constrained shortest-path routing algorithm used by DCM is explained in the section 3.5.2.

3.5.1. Routing Techniques.

In this section an overview of the real-time routing problem will be presented, along with a description of the DCM routing algorithm.

To maximize the utilization of high-speed, real-time networks and reduce congestion in these networks, real-time routing is very important. While datagram routing techniques can be used, virtual circuit routing is better, as performance guarantees can best be made on a fixed path. Also if the network is ATM based, the small cell size prohibits the use of datagram routing, instead virtual circuits routes are set up and small virtual circuit identifiers are used to determine the outgoing links. These small virtual circuit identifiers are prefered, as a network with a large number of hosts would need a large address space that could not be easily supported within a small cell. As we wish to maximize the "run-time" efficiency and reduce the congestion of the network, we consider only dynamic or adaptive routing schemes. Dynamic routing is a network routing technique that routes client channels on routes determined by using the current state of the network, thereby allowing the network to respond quickly and correctly to changes in network loading and facilitating a high utilization of the network’s resources. As fixed routes are to be used, many current fixed route or virtual circuit routing techniques must be examined for suitability of use in a real-time context. Virtual circuit routing techniques are employed in both circuit switching and packet switching networks. The next two sections present a survey of many of the important routing techniques used in these networks, and provide an analysis of their usefulness in our real-time environment.

A. Circuit Switched Routing Techniques.

Dynamic routing techniques used in circuit switched networks can usually be divided into two major categories: *time-dependent* and *state-dependent*. In time-dependent routing, preplanned routing patterns, which were computed offline, are entered at fixed times during the day to allocate network capacity for previously forecasted traffic demands. In state-dependent routing, the routing patterns are automatically varied according to instantaneous traffic demands and network status information to respond to traffic variations. Network status information includes such information as the number of successful calls and the occupancies of the trunk groups. The analysis of the information and the selection of routes can be done in a distributed or centralized manner. Usually the collection of load information and the route selection is done periodically, with the selected routing patterns valid for the entire period. Circuit switch routing techniques are almost always applied exclusively in telecommunications, where the backbone network[19] is fully connected and routes are considered as
direct or alternate. Direct routes are one link routes; alternate routes are usually limited to two link routes. Performance studies have shown that alternate routes with more than two links reduce call acceptance rates. Also with these networks there is associated a trunk reservation scheme that limits the effect of

[18] Actually the delay values of these links are set to ∞.

[19] This is the lowest level network in the hierarchy of networks that comprise the AT&T telecommunications system.

overflow traffic (i.e. calls that utilize the two-link alternate routes) on calls that require their direct routes. The network management facilities also include Automatic Call Blocking (ACB) whereby the network limits access to certain portions of itself in order to reduce the degradation of service to current users.[20] The main circuit switching routing techniques in use are summarized below.

Aggregated Least Busy Alternative (ALBA) is a distributed, state-dependent, dynamic routing technique for fully-connected networks [MiGi91] . We assume that the network has N nodes and a link with C circuits between every node pair; routes are restricted to have at most two links. The direct link is the single link route between the source and the destination. The alternate route is any of the possible $N-2$ two link routes between the source and the destination. In ALBA, upon arrival of a call, local information on the state of the links of all possible routes is used to determine the route of a call. In ALBA(k) the (C+1) states of each link, which represent the number of occupied circuits on that link, are lumped into k aggregates $(A_0, A_1, A_2, \cdots, A_{k-1})$ in order of increasing occupancy level. The largest aggregate A_{k-1} is the set of states with r or less idle circuits; we say that A_{k-1} comprises the set of reserved states and r is the trunk reservation parameter. Routing occurs in the following manner: the arriving call is attempted on the direct route; if a circuit is available, the call is carried along this route; if no circuit is available, the call is attempted on an aggregated-least-busy-alternative two link route. An ALBA two link alternate route for a source/destination pair s, d is one that minimizes max (A_{si}, A_{id}), where i is the intermediate node and A_{si} and A_{id} are the aggregate states of links (s,i) and (i,d) respectively. Any of the $N-2$ alternate routes can be chosen with ties broken randomly. The arriving call is accepted on a two link alternate route if it does not leave either link in aggregate A_{k-1}, else the call is blocked and lost.

State- and Time-Dependent Routing (STR) was developed by Nippon Telephone and Telegraph (NTT) and combines a learning automaton (the state-dependent portion) and a time-varying method (the time-dependent portion). In STR each node has a route selection list that is updated periodically by a central processor. These route selections are determined by that processor based on the network topology and link size, and the predicted traffic load between these nodes. When an incoming call arrives, it is routed using the route selection list. The call is first offered to the direct route and overflows to an alternate route, specified in the route selection list, if it is blocked. The alternate route is chosen using a learning automaton routing method. In this method a call overflowing its direct route is offered an alternate route. If the alternate route is not congested (i.e. if the number of idle trunks on either of the links does not exceed a set threshold), the call is given the route, and any subsequent call attempts the same alternate route. If the alternate route is congested, another alternate route is attempted by the call. Congestion status on the second link is relayed by using the call-completion signal or the trunk release signal sent back to the source node. This isolated scheme has several variations depending on whether the alternative routing is of the single-overflow type or of the multiple-overflow type. In the single-overflow type only one alternate route is considered if a direct route is not available, whereas in the multiple-overflow type a specified number of alternate route are considered if the direct route is unavailable. The first available alternate route is chosen for the call. The value of the congestion threshold can be varied to produce other scheme variations. In STR, usually the single-overflow scheme is used with a set non-zero value for the congestion threshold. Performance evaluations for combinations of these variants are given in [MaYa90].

Dynamic Non-Hierarchical Routing (DNHR) [Ash90a] was developed by AT&T and is a centralized, hybrid time-dependent and state-dependent route selection technique, where the time-dependent factor is determined by forecasted traffic patterns and the state-dependent factor is a response to network load variations. In DNHR, direct primary route and two-link alternate routes are used to carry traffic between source and destination DNHR tandem switches[21]. The time-dependent routing capability allows prespecified routing patterns to change as frequently as every hour in response to forecasted traffic patterns. In DNHR a call is first offered to the direct route and, if no circuits on this route are available, the call is offered to the alternate route set. If blocking occurs on the second link of an alternate two link route, a control message is sent back to the source node so that this blocked call can be routed on another route in the

[20] These situations are called focused overload situations. An example of this situation occurred during the last California earthquake, where many people tried to call California at the same time.

[21] These switches form the highest-level of the telecommunications network hierarchy and are usually intercity exchanges.

alternate route set (this technique is commonly referred to as *crankback*). These alternate routes (there is a maximum of 14 for each source/destination pair) are examined sequentially until the call is accepted. If the call cannot be accepted by any of the alternate paths, then the call fails. DNHR also includes a state-dependent routing ability, courtesy the NEtwork Management Operations System (NEMOS), which searches for idle trunk capacity on an individual call basis using the trunk reservation parameters as idle thresholds. If idle links are found, they can be incorporated into the alternate route set to increase the offering to an arriving call. This dynamic routing is only used in the case of network link failures and other unusual scenarios.

DNHR has also been extended to provide state-dependent routing based on trunk status information. This new routing technique is called Trunk Status Map Routing (TSMR) [Ash90b]. The TSMR concept involves having an update of the number of idle trunks in each of the DNHR trunk groups sent to the centralized network database every T seconds. The database determines a new ordered routing sequence based on the number of idle trunks and returns this sequence to each switch. The new ordered sequence is used for the next T seconds until the next update.

Dynamically Controlled Routing (DCR) [CaGa80] is a state-dependent routing method, developed by Bell-Northern Research, which uses centralized routing to determine the best routes, depending on the occupancy of the trunk groups. Each call is first offered the primary route. If this call is blocked, it is then offered to an alternate two-link route. The alternate route is selected on a probability basis and the number of call attempts on alternate routes can be set at some threshold value. A central processor computes route probabilities at fixed intervals based on the residual capacities of the links. These probabilities are usually computed every 10 seconds.

The System for Test Adaptive Routing (STAR) [GaCh87] is a state-dependent routing technique developed by the Centre National d'Etudes des Telecommunications (CNET) for dynamic routing in the French Telecom network. In this method, trunk group occupancy information is periodically collected at a central processor. The route selection sequences are updated at each local switch in order of decreasing route residual capacity. The residual capacity of a link is defined as the minimum residual capacity of all trunk groups belonging to the route. Incoming calls are offered first to the direct path and, if blocked, are offered to the alternate paths in sequential order. Crankback may used to detect blocked calls and do multiple alternate route attempts. Initially the STAR prototypes were implemented with the update period varying from 1 to 2 minutes due to technical constraints; however, the production switches are expected to have an update period on the order of 10 seconds.

Real-Time Network Routing (RTNR) is a decentralized routing technique developed by AT&T to provide routing for future dynamic class-of-service networks, which provide connections for voice, data, and wideband services on a shared transport network [AsCF91]. With RTNR, the source switch attempts to route an incoming call onto a direct trunk. If this direct trunk is not available, the switch attempts to find an available two-link route by first querying the destination switch[22] for the busy-idle status of all trunk groups connected to it. The source switch then compares its own trunk group busy-idle status information to that obtained from the destination switch in order to obtain the least loaded two-link path over which the call can be routed. This least loaded two-link path is obtained by comparing load threshold bit maps of the trunks corresponding to the source and destination switches. To obtain a specific class-of-service route, these load threshold bitmaps can be overlayed by an *allowed−via−switch* bit map which indicates routes that possess this class of service. Of these allowed links the least loaded two-link path is selected. It should be noted that the class-of-service routing provided by RTNR usually refers to bandwidth-based services.

Dynamic Alternative Routing (DAR) is a decentralized (isolated) learning-automaton routing method developed by British Telecom [StSo87]. In a DAR network an incoming call is offered the direct route. If it is blocked, it overflows to the currently selected two-link alternate route. If the call is also blocked at this alternate route, the call is refused, and a new two-link alternate route is selected at random from possible two-link routes for subsequent calls.

---

[22] This query is done over the Common Channel Signaling (CCS) Network .

B. Disadvantages of Circuit Switched Routing Techniques.

In the previous sections we have reviewed some of the more prominent circuit switched routing techniques. The major disadvantages of these techniques are as follows: only one type of service guarantees is considered (i.e. guarantees are made implicitly by the use of dedicated circuits and by the performance-oriented design of the telecommunication system[23], and the value of the bandwidth and delay parameter in these circuits cannot be changed); fully connected networks are assumed; in some cases the network traffic can be predicted (i.e. in time dependent routing techniques), and this prediction is utilized by the routing technique. Another drawback of some of the schemes is that centralized control is used.

These disadvantages reduce the usefulness of these techniques. In our environment many different classes of services are to be provided, hence a routing technique that optimizes routing for a single class of service is not useful. Also, the routing techniques surveyed cannot determine routes that provide a broad range of delay and jitter guarantees. The assumption of a fully connected network is not appropriate for the types of network topologies we deal with. These routing techniques are limited to two-link routes. As the topologies that will be encountered are most likely to be relatively sparse, optimizing for two-link routes is not appropriate. The traffic loads that will be seen on these networks will be very dynamic and highly unpredictable. The unpredictability is due to the lack of a convincing model for the multimedia traffic that will be present on these networks. The rate of growth of multimedia applications and the possible combinations of applications will provide an environment that will render historical call data obsolete; prediction will be almost impossible. While there are some useful ideas to be derived from circuit switching routing techniques, the disadvantages present in these techniques do not allow us to utilize them directly.

C. Packet Switched Virtual Circuit Routing.

In this section we present some relevant packet switching virtual circuit routing techniques. Although there are many virtual circuit routing techniques, three of them that are particularly suitable for examination. They are the *plaNet* routing algorithm, the scheme used by *Codex*, and a multicast routing scheme proposed by Kompella et al.

The *plaNet* network is a high-speed packet switching network designed to support multiple classes of service [AhCh91]. The objective of the source routing technique used in plaNet is to minimize call blocking while ensuring low end-to-end delay. In plaNet only bandwidth is guaranteed by reservation, but delay is minimized as much as possible. The routing algorithm assumes that queueing delay is not going to be a major issue in high-speed networks, as switches will be very fast and hence the focus of the routing algorithm is to minimize call blocking. Minimization of call blocking is achieved by favoring the shortest path between source and destination, and by load balancing. By favoring the shortest path, less links are used; hence the call blocking effect of this call on other calls in the network is reduced. By load balancing among shortest paths, the load is distributed evenly among the links, and, when this is correctly done, this also tends to reduce call blocking. Results are provided in [AhCh91] to verify that these objectives are achieved. The routing algorithm used is a modified shortest path algorithm where the link weights are an increasing function of the link load (this promotes load balancing). This algorithm is further constrained to obtain the minimum number of hops between the source and the destination, thereby reducing the effect of this channel on other channels in the network. Link weights take into consideration the bandwidth of the channel requested and the current load on the link; they are used to make saturated links unavailable and to discriminate among paths with equal number of hops. Among the minimum-hop paths, the path with the lowest total weight is chosen.

The Codex routing scheme permits the routing of virtual circuits and the rerouting of these virtual circuits in response to link congestion levels [HuSo86]. A virtual circuit is established between a source and a destination by sending a route message to the destination node, which then uses a source[24] routing shortest-path technique to establish a path from the destination to the source. Thus, the destination node is

---

[23] The bandwidth given by a circuit is dedicated to the client for the duration of the call and the delay experienced in a phone call is bounded.

[24] In this case the source responsible for the routing is actually the destination node.

responsible for routing and rerouting the virtual circuit. This routing algorithm is based on the total cost of all of the links in the network. A link's cost is a function of the excess capacity of the link, the data rate of the connection, the propagation delay, the total weighted traffic, and the priority of the connection. Therefore the total network cost will be lower if connections with high data rates or high priorities are routed along routes containing only a few hops. Routes for new paths are selected to minimize the increase in network cost. This is accomplished by assigning to each link the increment in the link's cost if the new connection were accepted, and then running a shortest path algorithm to determine the path with the least total cost. This path is the selected route. Rerouting decisions are made by scanning the routes terminating at a node. For each route terminating at a node, computations are made to determine what the traffic characteristics of the network would be if the route were deleted. Using those characteristics, the cost of the current route and the cost of the shortest route are computed. If the latter is less than the former, a secondary path is established. To protect against the possibility that many reroutings will overload a previously lightly-loaded link, only a fraction of the paths are considered for rerouting at any time, and a maximum amount of rerouted traffic can be established on any link over an interval.

Kompella et al. propose a multicast routing algorithm that determines a least-cost tree, spanning from the source to all of the destinations, subject to the constraint that all source/destination routes have a client-specified delay bound [KoPa92]. In their algorithm, they initially construct a closure graph on the set containing the source and all of the destination nodes. This graph is constructed from the cost and delay weights assigned to each edge (the edges represent links and the nodes represent packet-switching nodes) and is a complete graph where each edge connecting two nodes represents the minimum cost path where the delay along the path is less than the client-specified delay. The delay along the path is the sum of the propagation and transmission delays. Prim's spanning tree algorithm is then applied to this graph to determine a minimum-cost spanning tree. Heuristic functions are provided for choosing edges in the spanning tree. This tree is expanded into the edges that represent the constrained cheapest paths. As it is possible that the resulting graph is not a tree because the edges in the closure graph represent paths and expanding them may cause loops, a subsequent operation is performed to remove these loops. The result is a minimum spanning tree where each source/destination path is constrained by a specified delay bound. While this is a multicast algorithm, it can easily be applied to the unicast case, hence it merits analysis here.

The plaNet routing technique contains several useful ideas: for example, blocking probability is minimized; routes can be determined that support different class of service; and the topology considered is similar to those encountered in our environment. However, the routing technique only takes into consideration the bandwidth resources, and not the delay or buffer resources. In the Codex algorithm bandwidth resources are taken into consideration, and rerouting is examined; however, the delay and buffer resources are not addressed. The problem of achieving a minimum-cost route subject to delay constrains is exactly that we wish to solve, and Kompella et al. propose a useful solution which in essence is similar to the solution we describe below. However, Kompella's solution is very costly in terms of computation power because of the expansion of the closure graph and the loop-removal operation. Hence, in large networks this method would not be efficient. With these disadvantages in the circuit and packet switching routing techniques, it was necessary to extract the essence of the advantages from these techniques and use them to develop a suitable set of criteria from which a DCM routing algorithm can be developed.

3.5.2. The DCM Routing Algorithm.

In this section we will provide the criteria on the basis of which the DCM routing algorithm was designed, and present the actual algorithm

The routing algorithm should seek to :

- Maximize throughput.
- Balance the load.
- Obtain routes sufficiently fast.
- Maximize the probability that the route provided will be successfully established (i.e. the route will be established with the traffic and performance specifications given by the client).

Our algorithm seeks to maximize throughput by minimizing the number of intermediate nodes encountered along the path from the source to the destination host. By minimizing the hop count, there will be less contention for resources among channel requests; hence, call blocking will be decreased, with a corresponding rise in throughput. The load balancing criterion also reduces call blocking by distributing the load more evenly throughout the network. A fast algorithm is necessary to reduce the channel establishment time; however, speed should not be traded off for a high success rate. There is no advantage in obtaining a route quickly if a channel cannot be established along the route, and several retries are necessary before establishment. An important criterion is that of maximizing the probability that channel establishment is successful along the route provided by the routing algorithm. This maximization is achieved by using the traffic and performance requirements of the client and the most recent network state information to determine a route.

The routing algorithm calculates a minimal-cost route, in accordance with the criteria presented above, where the cost of the route is the sum of the costs of the links comprising the route. The cost of a link is a delay value, which is the sum of the minimum queueing delay offered by the starting node to a real-time channel with these traffic characteristics, the transmission delay, and the propagation delay along the output link. While the transmission and propagation delay and are fixed costs, the queuing delay experienced in the RCSP scheduler is variable, and is dependent on the current channel resource reservations on the corresponding output link and the traffic characteristics of this new channel. A detailed description of the equations used to obtain this variable value is given in [ZhFe92].

The algorithm proceeds in the following steps:

1. A directed graph is created in which the nodes correspond to switches and hosts in the network and the edges to the links connecting these switches and hosts. The weights attributed to each edge represent the link costs. The link cost are computed just prior to applying the algorithm thereby using the most recent link information obtained from routing update messages.
2. A constrained modified Bellman-Ford[25] algorithm is then applied to this graph to determine a possible route.

- For the delay bound case the algorithm proceeds as follows:
    - A. Consecutive searches are performed on all 1, 2, .., N-2-hops path from the source to the destination node until the delay condition $\sum_{l(s,d)} d_l \le D$ is satisfied, where D is the delay bound of the channel, d_l is the weight of link l, and (s, l) is the path from the source s to the destination d. A *constraint* is placed on the number of possible searches by stopping at the *hoplevel*[26] at which the delay bound condition is satisified.
    - B. At this *hoplevel*, the path is chosen with the minimum cost (i.e. min $\sum_{l(s,d)} d_l$) that meets the delay condition.
- For the delay jitter bound J case, the following steps are performed:
    - A. The minimum queuing delay offered by the link preceeding the destination node, d_{q_n}, is first examined. If $d_{q_n} \le J$, then the algorithm proceeds, else the channel cannot be accepted as the delay jitter bound condition cannot be satisified.
    - B. Consecutive searches are performed on all 1, 2, .., N-2-hops path from the source to the destination node until $\sum_{l=0}^{n-1} d_l + d_{p_n} + d_{t_n} \le D - d_{q_n}$ where D is the delay bound of the channel, d_{p_n} is the propagation delay associated with the link n, d_{t_n} is the transmission delay associated with link n, and there are n links along the path.
    - C. At this hoplevel, the path with the minimum cost that meets the delay condition is chosen.

[25] The fundamental Bellman-Ford algorithm [Bell58] searches for the shortest paths between a specified source and destination node starting from all possible one-hop paths and continuing until the N-2-hop paths are examined.

[26] This *hoplevel* is the number of hops from the source to the destination node.

This algorithm limits its search space, thus reducing its computation time, and provides a minimum-hop route that maximizes throughput. Throughput is maximized by minimizing the number of intermediate nodes encountered along the path thereby reducing call blocking. It also increases the probability that channel establishment will be successful as it determines the queuing delays based on the traffic characteristics of the channel and the most recent resource reservation information.

Another modification that has been made to the fundamental algorithm is the introduction of a constraint factor, α, which is used to constrain the search space to at most α hoplevels. This constraint is intended to be used during periods of network congestion, so that a route's impact on the other routes in the network can be reduced. The α factor is dependent on the current network load and the minimum possible hop count between the source and the destination. Initially α has been set to ∞ so that the entire search space is examined, but further work needs to be done to determine a suitable algorithm or set of heuristics for α.

Another routing algorithm that can be used for real-time routing is Dijkstra's shortest-path algorithm [Dijk59]. In this algorithm, the directed graph is constructed in the same manner as above. If the resultant route has a cost that is less than the delay bound, D, then the route is accepted and given to the administration algorithm. If, on the other hand the cost of the resultant route is greater than the delay bound, the client's request is denied. Simulation are needed to determine the improvement provided by the constrained Bellman-Ford algorithm described above over this straightforward applications of Dijkstra's shortest-path algorithm.

3.5.3. Routing Updates.

Routing updates are currently done on a per-channel-establishment basis. This can be accomplished by at least two methods. In the first method, the source can broadcast a route update message to each node in the network using reverse path broadcasting. This route update message will contain the traffic characteristics, the performance parameters, and the route of the channel. Each node will then use this information to calculate the incremental load on each of the links comprising the route, and update its local link-state tables. The second method is to have every node broadcast the load values of its links to all other nodes after it sends the reverse channel establishment message to the previous node on the new channels route; each receiving node updates its local link-state table. This broadcast is done along a minimum spanning tree. The tradeoff between the first and second methods is bandwidth savings in the first method versus computational power savings in the second method. Currently, the second method has been implemented and is being used in the simulation experiments; however, both schemes will be developed to determine the more efficient method. The use of periodic and incremental routing update is also being considered.

4. Further Aspects of DCM.

In the previous sections we provided an in-depth description and analysis of the DCM algorithms. There are other aspects of the DCM scheme that were not addressed previously and will be discussed in this sections. These aspects refer to the granularity of control of the algorithms and the responsiveness of the scheme.

A. Granularity of Control.

The DCM algorithms can be applied to modify the performance parameters or the route of an entire channel or a segment of the channel. The smallest segment of a channel that can be modified is a single link. Control can be applied at the link (or *local*) level or at the route (or *global*) level. We make no distinction between a modification affecting a single link and modification affecting any subset of a channel's links (excluding the entire route); both of these are called *local* modifications. The modification of an entire channel is referred to as a *global* modification. The DCM algorithms can be utilized for both local and global modifications, as segments of a channel are themselves real-time channels whose source is the node at which the segment begins and destination the node at which the segment ends. The traffic characteristics of this segment are the same as those of the "parent" channel, and the performance requirements are derivatives of those of the parent. The throughput requirements of the channel segments are the same as the parent's, but the delay bound is the sum of the local queuing bound in the switches, and the transmission and propagation delays of the links comprising the segment, while the delay jitter bound is the queuing delay of the last switch along the path of the segment. The DCM contracts are equally applicable to channel segments, since the DCM algorithms can be used without changes.

There are advantages and disadvantages to local and global control. Local control has the advantages that the routes obtained by the routing algorithm have a high success rate, as the routing information gathered within a small radius is usually up to date, and the channel establishment time is low due to the short distances traveled. These facts, coupled with the fact that call blocking (due to the interactions of several channel establishments) is less likely to occur over this short distance, result in quick channels establishments. However local routing algorithms do not posses the global knowledge that the source has and may not be aware of routing constraints (these constraints may be cost or security restrictions). Local control can also lead to congestion, as the local rerouting of multiple channels on a single link can result in the area around the link becoming congested, whereas global control will tend to spread the rerouted channels more evenly throughout the network.

It should be noted that a single segment of a channel may in fact be a subnetwork if the channel route composed of logical links. The extension of the DCM scheme to internetworks can be easily done and will be reported in a future paper.

B. Responsiveness of the DCM scheme.

The response time of the DCM scheme is the time interval between a client's modification request and the reply of the network. This interval is dependent on the client's request and the current network load. We have the following two cases:

- If the client requests a modification that reduces the level of the performance parameters, then the response is **acceptance**, and the client can modify its traffic characteristics to reflect this response.
- If the client requests a modification that increases the level of the performance parameters, then the response can be an **acceptance** or **rejection** of the request. If the response is an acceptance, then at the instant the response is passed to the client, the excess resources are available and the client can now begin to use these resources.

The response time should be as short as possible for client satisfaction. Response time is dependent on three factors: the efficiency of the routing algorithm, the channel establishment policy, and the speed of the channel establishment procedure. The efficiency of the routing algorithm is dependent on the correct representation of the network state (i.e. how recent is the routing update information) and the method by which this data is utilized to produce a route that has a high probability of being successfully used by the channel establishment procedure. The channel establishment policy determines the number of channel establishment retries that a request is allowed before a rejection response is given to the client. There is an obvious relationship between these two factors. If the routing algorithm is very efficient, few retries may be necessary; since the algorithm has supplied paths with a high probability of success and due to the fact that the routing information is up to date, those paths should be able to accept the channel. If updates are done infrequently the efficiency of the algorithm is most likely low and more retries should be allowed to increase the probability of finding a path. The speed of the channel establishment procedure is obviously a factor in the response time and attempts have been made to minimize this time by reducing the computational complexity of the admissions tests.

The shortest-path algorithm used by DCM have been widely applied and rigorously tested, and would not be enhanced by further work. The efficiency of the modified Bellman-Ford algorithm would however be enhanced by the use of the most current state information. This is the classical update problem, as the number of messages needed to provide all nodes in the network with a correct and consistent "view" of the network would produce significant overhead. A suitable update mechanism must be employed that reduces the resources consumed by update messages, yet provides useful, recent, and consistent state information. The mechanisms to be implemented and analyzed by simulation are Source-demand Routing

Update Mechanism [EsRe92], and incremental routing updates. The adoption of one of these mechanisms or the development of a more suitable mechanism should increase the efficiency of the routing algorithm.

The channel establishment policy is a heuristic that can only be enhanced by constant observation. However, the channel establishment procedure can be modified to exploit the *time value* of information in the network. The value of network state information decreases in time if it is not updated. At its inception, an update packet contains the exact state of the node from which it was generated; as it is sent to nodes further away from its source, the network state changes and the information provided in the packet is a less accurate description of the state of the source node. Beyond a certain amount of time the information is useless.

As a channel establishment packet moves from the source node towards the destination node, it encounters nodes that have received more recent network state information from the destination node. Thus the time value of the information pertaining to the route along which the establishment packet is traveling is increasing. This indicates that the nodes have more precise information on the state of the network along that route as the establishment packet moves towards the destination. If the establishment packet encounters a link which has insufficient resources to accommodate the request, it may return to the node preceding this unavailable link and request a new route from this node to the destination. This node, which has more recent information than the source of the packet, computes the route from itself to the destination using the traffic and performance parameters contained in the establishment packet. Each node that attempts to route the establishment packet because of a failed node knows the previously attempted paths and removes them from the graph before applying the shortest-path algorithm. Thus, looping is not possible. We refer to this routing technique as *hybrid routing* as it combines source routing with distributed routing. Hybrid routing can also be useful in the case of network failure, where a failure condition on that link renders it unavailable, and the establishment message can be routed along an available path.

5. Simulations and Results.

In this section we present some of the initial simulation experiments conducted on the DCM algorithms. In these initial experiments we sought to verify that the algorithms performed correctly. The algorithms should ensure that

- the delay bounds of all packets in the primary and alternate channels are met,
- the throughput of the alternate channel correctly reflects the modification (if a throughput modification has been made),
- there are no out-of-sequence packets or, in the case of a bounded-violation contract, the number of out-of-sequence packets does not exceed the violation bound.

To meet the criteria mentioned above, various mechanisms were used. To ensure that the delay bounds of all of the packets were met, histograms of the delays of all packets on the primary and alternate channel were used. Throughputs were verified by using a graph of *packets per interval* vs *simulated time*. In the simulator, the number of packets arriving at the destination during a 100 ms interval were recorded. This provided the value of throughput in each interval. Out-of-sequence packets were recorded by an out-of-sequence counter associated with each channel.

The above criteria were applied to a series of five experiments done to verify the validity of the DCM scheme. The first three experiments tested the global, "violation free" operation of DCM, the fourth experiment tested the local, "violation free" operation, and the fifth the global, bounded violation operation. The topology of the simulated network and its configuration are given in Figure 7.

In this network there are 8 nodes connected by 13 links. All of the links have the same maximum speed of 1 Mbps, with the exception of link(1,5)[28], link(5,1), link(6,5), and link(5,6) which have links speeds of 10 Mbps. The propagation delay on link(0,6), link(6,0), link(1,2), link(2,1), link(3,2), link(2,3), link(7,6) and link(6,7) is 20 ms, while that of the remaining links is 10 ms. The different link speeds and propagation delays were chosen to introduce a slight heterogeneity into the network.

[28] A link is characterized by the pair (s , d), where s refers to the source gateway and d refers to the destination gateway.

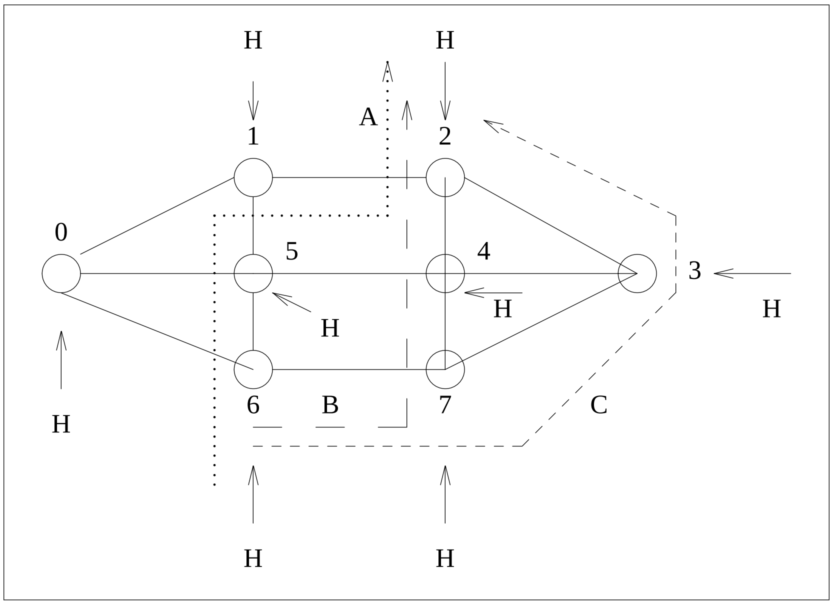
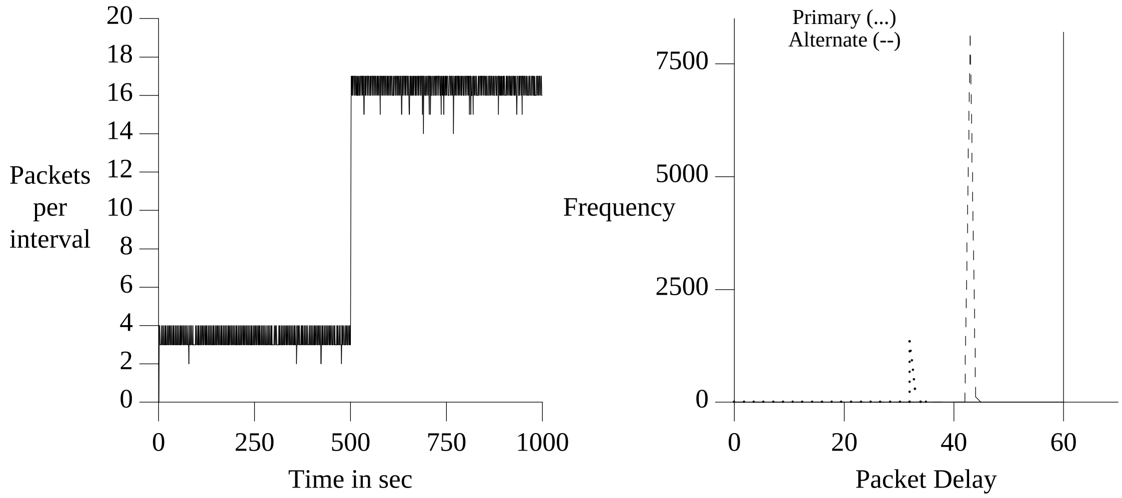


Figure 7.

In the first three experiments a single real-time channel had both its parameters and route modified. The initial parameters of the real-time channel under consideration were: $X_{\min} = 30\ ms$, $X_{ave} = 40\ ms$, $I = 1000\ ms$, $S_{\max} = 1000\ bytes$, and $D = 60\ ms$. The initial route, denoted by the gateways traversed, was 6->5->4->2. This channel was requested at time 100 ms (0.1s). There were 14 other real-time channels present in the network, with the aggregate link bandwidth reserved being 51% of the total bandwidth available. It should be noted that some of the links had bandwidth reservations as high as 80% of their total bandwidth.

In the first experiment, a modification request was made by the client at time 500 sec to change the parameters to $X_{\min} = 6\ ms$, $X_{ave} = 8\ ms$, $I = 1000\ ms$, $S_{\max} = 1000\ bytes$, and $D = 60\ ms$. This request is essentially a demand for five times the throughput reserved for the initial channel. To further test the algorithm, the experiment was designed so that the new route chosen by the routing algorithm was 6->7->4->2. The choice of this route exercised the "common" link part of the algorithm. The results of this experiment are shown in Graph 1 and Graph 2 of Figure 8. Graph 1 plots the throughput vs time, and Graph 2 is a histogram showing the delay distribution of the packets in the real-time channel before and after modification. As shown in Graph 1, the average throughput of the channel was increased at time 500 sec from 3.5 packets per interval to 16.5, an increase of 371%. Although a 400% increase was requested, the channel only used 371% of its excess capacity. The delay histogram shows that no packet exceeded its deadline of 60 msec. The average delay of the primary channel was 32.12 ms, and that of the alternate 43.02 ms. The difference between the delays on the primary and the alternate route is due to the increased propagation delay on link(6,7) and the extra queuing delay experienced by the packets along the alternate route. The delays on the alternate channel are larger than those on the primary, (see Graph 2) due to the larger throughput the client has available and is utilizing. There were no out-of-sequence packets delivered to the destination, and the correct alternate route was used.

In the second experiment a modification request was made by the client at time 500 sec to change the parameters to $X_{\min} = 60\ ms$, $X_{ave} = 80\ ms$, $I = 1000\ ms$, $S_{\max} = 1000\ bytes$, and $D = 60\ ms$. This request increased the throughput of the alternate channel by 50% of that of the primary channel. The route provided by the routing algorithm was the completely disjoint route 6->7->3->2. The results of this experiment are shown in Graph 3 and Graph 4. Graph 3 shows the 50% reduction in the throughput requested by the channel. The average throughput of the channel was decreased from 3.5 packets per interval to 1.5, a decrease of 57%. The delay histogram shows that no packet exceeded its deadline. The average delay of the primary channel was 32.81 ms, and that of the alternate 53.07 ms. The difference between the delays on the primary and the alternate route is due to the increased propagation delays on link(6,7) and link(2,3), as

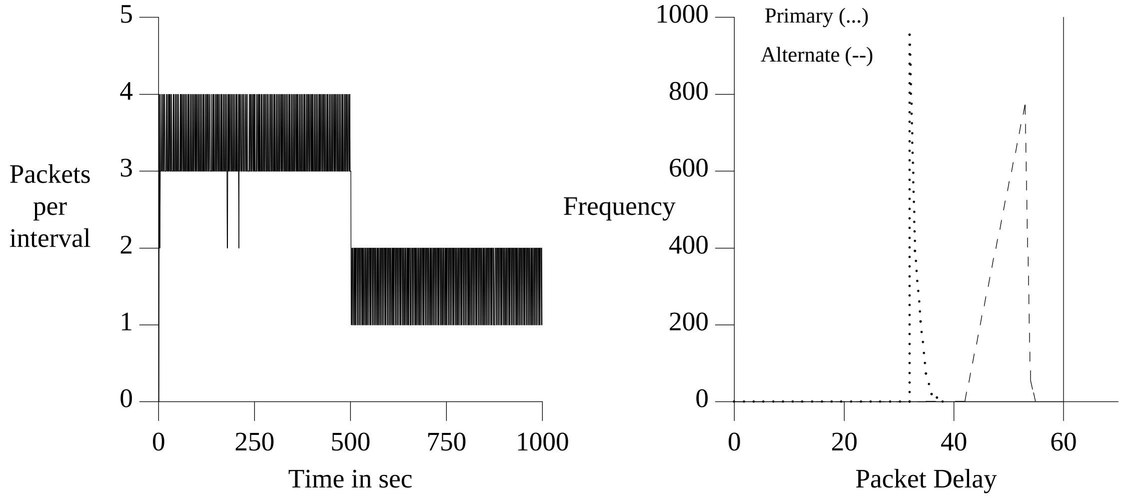


Graph 1.
Throughput Graph

Graph 2.
Delay Histogram.

Figure 8.

well as the extra queuing delay experienced by the packets along the alternate route. There were no out-of-sequence packets delivered to the destination, and the correct alternate route was used.

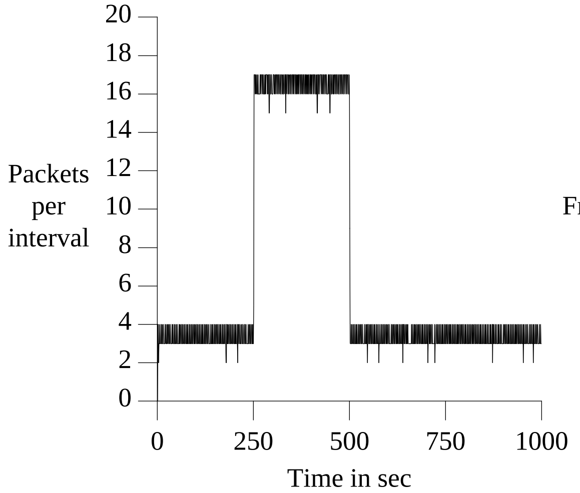


Graph 3.
Throughput Diagram.

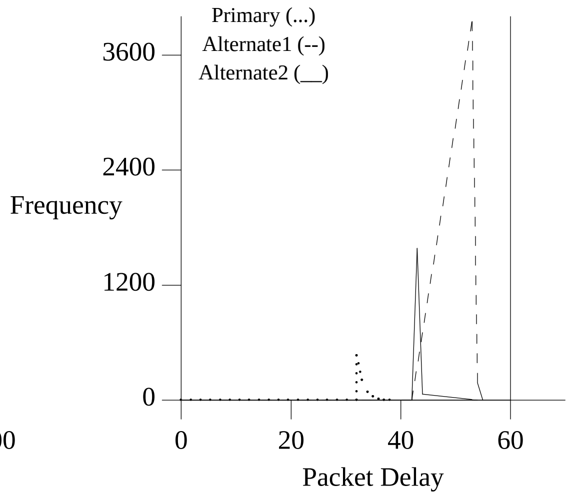
Graph 4.
Delay Histogram.

Figure 8 (cont'd).

In the third experiment two modification requests were made by the client at time 250 sec and 500 sec, respectively. The object of these modifications was to increase the channel's maximum throughput by a factor of 5 and then to reduce it back to its original value. Thus, the change in parameters were: $X_{\min} = 6\ ms$, $X_{ave} = 8\ ms$, $I = 1000\ ms$, $S_{\max} = 1000\ bytes$, and $D = 60\ ms$, followed by $X_{\min} = 30\ ms$, $X_{ave} = 80\ ms$, $I = 1000\ ms$, $S_{\max} = 1000\ bytes$, and $D = 60\ ms$. The routes provided by the routing algorithm were 6->7->3->2 and 6->7->4->2 for each modification, respectively. Graphs 5 and 6 display these results. The throughput diagram reflects both of the modifications made to the throughput of the channel. The average throughput increased to 16.5 packets per interval before returning to its original value of 3.5 packets after the interval. The histogram in Graph 6 shows the delay distributions of the packets along the original channel and after each modification; it indicates that no packet exceeded its 60 ms delay bound.



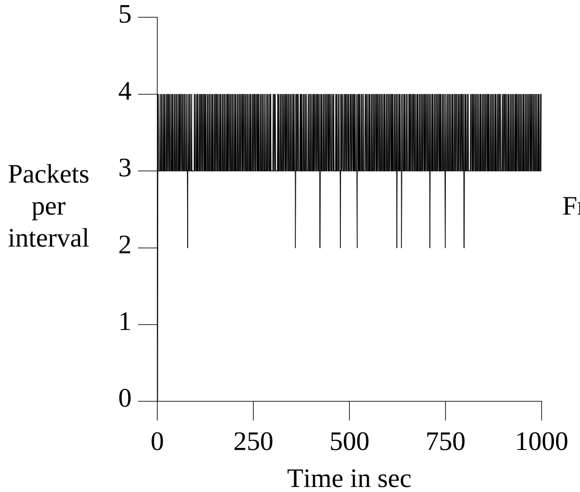
Graph 5.
Throughput.



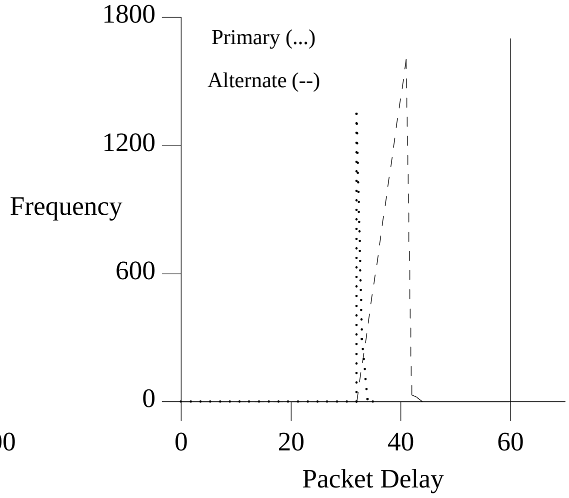
Graph 6.
Delay Histogram.

Figure 8 (cont'd).

The average delay of the primary channel was 32.70 ms, that of the first alternate 53.0 ms, and that of the second alternate 42.1 ms. The differences are due to the variations in propagation delays along the routes and the queuing delays in the gateways. Even in this experiment the destination received all real-time packets on this channel in the correct order, and the correct alternate route was used.



Graph 7.
Throughput.



Graph 8.
Delay Histogram.

Figure 8 (cont'd).

The fourth experiment has a local route modification at time 500 sec. There were no changes to the performance parameters. The initial local route 5->4->2 was modified to 5->1->2. Graphs 7 and 8 display the results. The throughput diagram reflects the fact that the throughput was unaffected during the transition, and the histogram, showing the delay distributions of the packets along the original channel and after the modification, indicates that no packet exceeded its 60 ms delay bound. The average delay of the primary channel was 32.80 ms, and that of the second alternate 40.9 ms. All packets were received in sequence at the destination.

In the fifth and final experiment, the bounded-violations operation of the DCM algorithms was tested. In the alternate channel all of the performance parameters were unchanged except of the delay bound parameter, $D_{alternate}$. The primary path parameters were: $X_{\min} = 30$, $X_{ave} = 40$, I = 1000, $S_{\max} = 1000$, $D_p = 90$. $D_{alternate}$ was made less than $D_{primary}$, and the pre- and post- establishment bounds on the violations were compared to the actual number of violations. This comparison is shown below in Table 2:

Table 2 - Bounded-Violation Experiment.

$D_{alternate}$	d_p	d_a	Bound(1)	Bound(2)	Actual packet count
80	85	65	3	1	1
60	85	55	3	2	1

where Bound(1) is $\left\lceil \frac{d_p}{X_{\min_{primary}}} \right\rceil$ and Bound(2) is $\left\lceil \frac{d_p - d_a}{X_{\min_{primary}}} \right\rceil$. As can be seen above, both the pre- and post-establishment violation bounds were correct.

6. Conclusion and Future Work.

In this paper we have presented a scheme for dynamically managing real-time channels. We provided the motivation for Dynamic Connection Management (DCM) and showed that the core of this motivation was to increase network *availability* and *flexibility*. A description of the DCM modification contracts was provided along with those of the DCM algorithms. The DCM contracts guarantee either a transition from the primary to alternate channel without any bound violations, or a transition where the number of packets involved in a performance violation is bounded. The three DCM algorithms: the Channel Administration algorithm, the Transition algorithm, and the Routing algorithm were then described in detail. The algorithms provided control at both the global and local level, and a *hybrid* routing scheme was developed to improve the responsiveness of DCM. A simulator was built and preliminary experiments and analysis were done to verify the correctness and usefulness of the global control, local control, violation-free and bounded-violation operation of the algorithms.

There is much future work to be conducted to realize a complete DCM scheme. The important aspects of this future work are the following:

- Multimedia workloads and internetwork topological characterizations are needed to generate a useful workload model and an appropriate topology for simulation experiments.
- The relationship between the routing algorithms and the delay relaxation function needs to be analyzed, to try to increase the number of channels established.
- The hybrid routing technique needs to be simulated and analyzed.
- A routing update mechanism needs to be developed and analyzed.
- A client/network interface needs to be specified which will exploit the new flexibility and availability provided by DCM.
- The DCM algorithms need to be expanded for use in an internetwork context.

Another important aspect of future work is the development of DCM applications (such as those given in the motivation section of the paper). DCM applications serve two major purposes: they exercise the DCM algorithms in a "real-world" environment and may suggest further enhancements to the algorithm, and they, in themselves, are useful network management applications of which very few have been seen in the literature. At present, three applications are being developed. The first two applications are unicast quality-of-service real-time channel management applications. The first is a throughput performance management application that dynamically provides changes in throughput to the clients. The throughput on a real-time channel can be decreased by the client at any time, whereas an increase in throughput is dependent on the client's new request and on the network's real-time load. This application can best be demonstrated as a *Still Image Sequence Browser* that is used to browse sequences of large uncompressible still images. These images are usually encountered, for example, by earth scientist in their analysis of large satellite pictures such as those of the earth's forest cover, cloud cover, and oceans, to name a few. The next unicast quality-of-service application is an error performance management application which performs rerouting of real-time channels based on their client-defined error rate thresholds and the error rates of the links along a channel's route. If the error rate of a link exceeds the rate threshold, that channels is a candidate for route modification. The last is a multicast management application. The multicast manager modifies channels parameters and routes based on the locations and performance characteristics of new receivers joining a multicast conferencing session. The manager also provides performance management in that it reroutes multicast tree segments based on the error rates of the links they traverse and on their error thresholds. As the development of these applications proceeds, they will be documented and made available in the literature.

References

[AhCh91] Ahmadi, H., Chen, J., Guerin R. "Dynamic Routing and Call Control in High-Speed Integrated Networks" *IBM Technical Report November 1991* **IBM Research Division, T.J. Watson Research Center. Yorktown Heights, NY.**

[AnHS90] Anderson, D., R. Herrtwich, and C. Shaefer, "SRP: A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet", *Technical Report TR-90-006,* **International Computer Science Institute, Berkeley, Feb. 1990.**

[AsCF91] Ash, G., Chen, J., Frey, A., Huang, B. "Real-Time Network Routing in a Dynamic Class-Of-Service Network" *International TeleTraffic Congress (ITC-13).* **Elsevier Science Publishers B.V. (North-Holland) IAC 1991.**

[Ash90a] Ash, G. "Design and Control of Networks with Dynamic Nonhierarchical Routing" *IEEE Communications Magazine,* **October 1990.**

[Ash90b] Ash, G. "Use of a Trunk Status Map for Real-Time DNHR" *International TeleTraffic Congress (ITC-11).* **Elsevier Science Publishers B.V. (North-Holland) IAC 1985.**

[BaMa91] Banerjea, A., and B. Mah, "The Real-Time Channel Administration Protocol", *Proc. Second International Workshop on Network and Operating System Support for Digital Audio and Video,* **Heidelberg, Germany, November 1991.**

[Bell58] Bellman , R., "On a routing problem" *Quarterly of Applied Mathematics,* **vol. 16 no.1 , pp 87-90, 1958.**

[Catl92] Catlett, C. "In Search of Gigabit Applications", *IEEE Communications Magazine,* **vol. 30, no. 4, April 1992.**

[ClSh92] Clark, D., Shenker , S., Zhang, L. "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism" *SIGCOMM '92.* **Proceeding of the SIGCOMM '92, pp 14-26, August 1992.**

[DeCH90] Deering, S., Cheriton , D. "Multicast Routing in Datagram Internetworks and Extended LANs" *ACM Transactions on Computer Systems.* **Vol 8. No. 2, pp 85-110, May 1990.**

[Dijk59] Dijkstra, E.W., "A note on two problems in connexion with graphs" *Numerische Mathematik,* **Vol. 1 pp 269-271 , 1959.**

[DiKe90] Dix , F., Kelly, M., Klessig, R. "Access to a Public Switched Multi-Megabit Data Service Offering" *Computer Communications Review.* **Vol 20.3, pp. 46-61, July 1990.**

[EsRe92] Estrin, D., Rekhter Y., Hotz, S. "Scalable Inter-Domain Routing Architecture" *SIGCOMM '92.* **Proceeding of the SIGCOMM '92, pp 40-52, August 1992.**

[FeBa92] Ferrari, D., Banerjea, A., Zhang, H. "Network Support for Multimedia. A Discussion of the Tenet Approach" *Technical Report TR-92-072,* **International Computer Science Institute, Berkeley, November 1992.**

[FePa92] Ferrari, D., Pasquale , J., Polyzos G. "Network Issues for Sequoia 2000" *Proceedings of COMPCOM 92,* **pp. 401-406.**

[Ferr90] Ferrari, D., "Client Requirements for Real-Time Communication Services" *Technical Report TR-90-007,* **International Computer Science Institute, Berkeley, June 1990.**

[Ferr91] Ferrari, D., "Design and Applications of a Delay Jitter Control Scheme for Packet-Switching Internetworks", *Proc. Second International Workshop on Network and Operating System Support for Digital Audio and Video,* **Heidelberg, Germany, November 1991.**

[Ferr92] Ferrari, D., "Real-Time Communication in an Internetwork" *Technical Report TR-92-001,* **International Computer Science Institute, Berkeley, Jan. 1992.**

[FeVe90] Ferrari, D., and D. C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-

Area Networks", *IEEE Journal on Selected Areas in Communications,* **vol. 8, no. 3, April 1990.**

[CaGa80] Cameron , W., Galloy , P., Graham , W. "Report on the Toronto Advance Routing Concept Trial" p.ul Proceedings of the First International Planning Symposium Paris , France , 1980.

[GaCh97] Gautier P., Chemouil, P. "A System for Testing Adaptive Traffic Routing in France" *Proceedings of Globecomm, 1987.* **Tokyo, Japan, 1987.**

[HuSo86] Humblet, P., Soloway, S. "Algorithms for Data Communication Networks - Part I, II" *Codex Corporation* **Cambridge Mass, 1986.**

[KoPa92] Kompella , V., Pasquale J., Polyzos , G. "Two Techniques for Multicasting for Multimedia Applications" *Proc. of Third International Workshop on Network and Operating System* **Support for Digital Audio and Video, San Diego, California, November 1992.**

[LaPa91] Lazar, A. A., and G. Pacifici, "Control of Resources in Broadband Networks with Quality of Service Guarantees", *IEEE Communications Magazine,* **vol. 29, no. 10, October 1991.**

[MaYa90] Mase, K., Yamamoto, H., "Advanced Network Traffic Control Methods" *IEEE Communications Magazine,* **October 1990.**

[MiGi91] Mitra, D., Gibbens, R., Huang B. "Analysis and Optimal Design of Aggregated-Least-Busy-Alternate Routing on Symmetric Loss Networks with Trunk Reservation" *Teletraffic and Datatraffic in a Period of Change.* **International TeleTraffic Congress (ITC-13). Elsevier Science Publishers B.V. (North-Holland) IAC 1991.**

[PaTu90] Parulkar, G., Turner , J. "Towards a Framework for High-Speed Communication in a Heterogeneous Network Environment" *IEEE Network Magazine* **March 1990.**

[PaZF92] Parris, C., Zhang , H., Ferrari, D. "A Mechanism for Dynamic Re-routing of Real-time Channels" *Technical Report TR-92-003,* **International Computer Science Institute, Berkeley, Jan. 1992.**

[RaSa92] Ravindran, K., Sankhla , M., Gupta, P. "Multicast Models and Routing Algorithms for High Speed Multi-Service Networks" *Technical Report - February 1992.* **Department of Computer and Information Sciences, Kansas State University, Manhattan, Kansas.**

[StSo87] Stacey, R., SongHurst D. "Dynamic Alternate Routing in the British Telecom Trunk Network" *ISS 1987* **Phoenix , Arizona, 1987.**

[Topo90] Topolcic, C., Editor, "Experimental Internet Stream Protocol, Version 2 (ST-II)", *Networking Working Group, RFC 1190,* **October 1990.**

[WoMo91] Wolfinger, B., and M. Moran, "A Continuous Media Data Transport Service for Real-Time Communication in High Speed Networks", *Proc. Second International Workshop on Network and Operating System Support* **for Digital Audio and Video, Heidelberg, Germany, November 1991.**

[ZhFe92] Zhang, H., Ferrari, D. "Rate-Controlled Static-Priority Queuing" *Technical Report TR-92-003,* **International Computer Science Institute, Berkeley, Jan. 1992.**