An Overview of the Isochronets Architecture for High Speed Networks

Yechiam Yemini and ***Danilo Florissi***

Distributed Computing and Communications (DCC) Lab
450 Computer Science Bldg., Columbia University, NYC, NY 10027

Technical Report CUCS-007-95

{Yemini,Florissi}@cs.columbia.edu

Abstract

This paper overviews a novel switching architecture for high-speed networks: Isochronets. Isochronets time-divide network bandwidth among routing trees. Traffic moves down a routing tree to the root during its time band. Network functions such as routing and flow control are entirely governed by band timers and require no processing of frame headers bits. Frame motions need not be delayed for switch processing, allowing Isochronets to scale over a large spectrum of transmission speeds and support all-optical implementations. The network functions as a media-access layer that can support multiple framing protocols simultaneously, handled by higher layers at the periphery. Internetworking is reduced to a simple media-layer bridging. Isochronets provide flexible quality of service control and multicasting through allocation of bands to routing trees. They can be tuned to span a spectrum of performance behaviors outperforming both circuit or packet switching.

1 Introduction

Recent advances in transmission technologies dramatically increase the bandwidth afforded by future networks. These quantitative changes give rise to significant qualitative changes in the applications supported by the network and in their service needs, in the relations between processing and communication speeds, in tradeoffs between bandwidth and complexity of network mechanisms, and in latency constraints on network control. New network technologies are required that can address these changes. The main goal of this section is to provide a description of High-Speed Network (HSN) problems addressed by Isochronets.

Support end-end transport of multi-protocol frames. Performance can be severely degraded if internal HSN operations are anchored on a single protocol suite. In such scenario, the adaptation between external protocols and the internal one may severely degrade end-end communication Quality of Service (QoS). Multi-protocol frames must be supported without severe performance degradation.

Support tunable guaranteed QoS of end-end transport. Application needs range from cheap best-effort service (such as electronic mail delivery) to expensive guaranteed QoS (such as embedded parallel processor communications). HSN applications may depend critically on QoS guarantees. It is essential to support such on-demand flexibility by providing means to control, finely tune and strictly guarantee QoS.

Support asynchronous, synchronous and isochronous traffic streams. Future integrated network services include an array of synchronization needs: asynchronous (e.g., distributed applications), synchronous (e.g., embedded parallel processors), and isochronous (e.g., voice) streams. These must be supported uniformly by the infrastructure.

Minimize processing and queueing latency in the transport path. Traditional networks sought to maximize utilization of communication capacity via sophisticated processing and queueing to overcome contention for resources. HSN may stretch the processing capacity (and costs) of network elements to its limit. Processing by network elements need be minimized and operations simplified, possibly trading-off communication bandwidth for processing bandwidth. Network queueing delays are unpredictable and must happen under controlled supervision, with the option of being avoided if necessary.

Support speed-scalability for link bandwidth ranging from hundreds of megabits to terabits per second. Link speeds have been increasing steadily in recent years and, as a consequence, HSNs are likely to encompass a range of transmission rates. HSN operations must possess the flexibility to span heterogeneous link speeds.

Support efficient internetworking with existing stacks and media. Current networks with associated hardware media and protocol stacks will continue to inhabit future internetworks. HSNs must be inter-operable directly with such legacy networks.

Support scalability for large size networks (ranging from ten to hundred million end nodes). HSNs will bring multiple multimedia integrated communication services to the subscriber premises (very likely homes). The potential size of such networks demands a scaleable HSN architecture.

Support efficient interfacing with end nodes. The interface can be a major bottleneck in end-end QoS performance and thus a the HSN architecture must enable the implementation of efficient interfaces.

Support efficient bandwidth sharing and flexible adaptation to service demands. Bandwidth sharing is the driving force to lower cost services. A HSN must enforce sharing when QoS can be compromised for lower cost services.

Handling large latency-bandwidth products. Traditional networks often involve global feedback control. The time-scale over which the behaviors that need be controlled occur (e.g., recovery from loss) is typically of round-trip latency order. When latency exceeds the time scale for control, feedback may be inadequate. HSN will need to replace global feedback with open-loop or local control. A particular example of concern is admission control.

Isochronets divide network bandwidth among routes, that is, at each time, routing trees to particular destinations are enabled. The network configures itself over time to build such routes. Switching is accomplished entirely based on time and on the current configuration and is independent of frame contents. Additionally, Isochronets signal sources when new routes become available so that the latter can schedule traffic motions accordingly. Isochronets also provide periods when certain sources have priority to the tree root. Sources can synchronize their QoS demanding services to these periods.

Isochronets fit the requirements set forth for HSNs. First, Isochronets switch without looking inside frames. Consequently, there is no processing inside the network that needs to scale with transmission speeds, multiple protocol frames can coexist in the same physical network without adaptation, and multiple link speeds and media (electronic or optical) can coexist without complicating network operations. Second, sources are signaled when routes become available. These signals may be propagated to applications which can then implement multiple synchronization services. Third, the fact that there is no frame adaptation and that media access is based on network supplied signals simplifies the design of the network interface. The interface functionality is to flow frames to and form the network and forward signaling information, which may be used to avoid buffering. Fourth, Isochronets support priority services that can be mapped and flexibly tuned to the diverse QoS needs of end applications. Fifth, trees multiplex network bandwidth among sources, thus promoting multiplexing of non-QoS demanding sources. Sixth, interconnection of Isochronets with existing stacks and media is simple, implemented using bridge-level functionality. Seventh, Isochronets can be clustered to provide more flexible tree allocation mechanisms within and between clusters. The clustering concept in Isochronets can be exploited to support scalability in terms of number of nodes. Finally, all control mechanisms in Isochronets rely on a single local mechanism responsible for the allocation of bandwidth to trees. This local mechanism completely avoids end-end feedback delays in controlling the network.

This paper is organized as follows. Section 2 describes Isochronets operations. Section 3 reviews related work. Section 4 exemplifies two Isochronets realizations. Section 5 describes control mechanisms in necessary to operate Isochronets. Section 6 evaluates the performance behavior. Finally, Section 7 concludes.

2 Isochronets Operations

2.1 Routing on Trees

Consider the motion of a frame in a store-and-forward or Packet-Switched (PS) network [2, 15]. The frame follows a path to its destination on a routing tree maintained by routers. It experiences random processing and queueing delays at nodes on its way, due to contention traffic. This is depicted in Figure 1, where a point in the graph indicates the location and time of a typical frame moving from the Origin node to the Destination node through the indicated path in the underneath network topology.

PS networks permit arrival randomness to propagate into network nodes. Network resources are efficiently utilized at the cost of QoS degradation in the form of frame loss or random delays. To support QoS, the very sources of traffic randomness need be suppressed via global admission controls. Admission delays and reduced network utilization are traded-off against reduced contention. However, QoS can only be statistically guaranteed. The interruptions seen by a source depend on aggregated contention traffic of other random sources. Statistical QoS costs in increased admission delays, in reduced network utilization and in lower effective bandwidth seen by sources (for example, when leaky-buckets drip slowly [13]). In the limit, where contention is eliminated with high probability, the very value of store-and-forward with respect to a circuit-switched service becomes questionable.

Circuit-Switched (CS) networks [3, 15] seek to provide absolute QoS by global resource reservations. Once a source establishes a circuit, traffic can move uninterrupted. Contention is elimi-

nated in favor of long (larger than round-trip latency) admission delay and reduced bandwidth availability. Elimination of contention results in poor network utilization under random traffic.

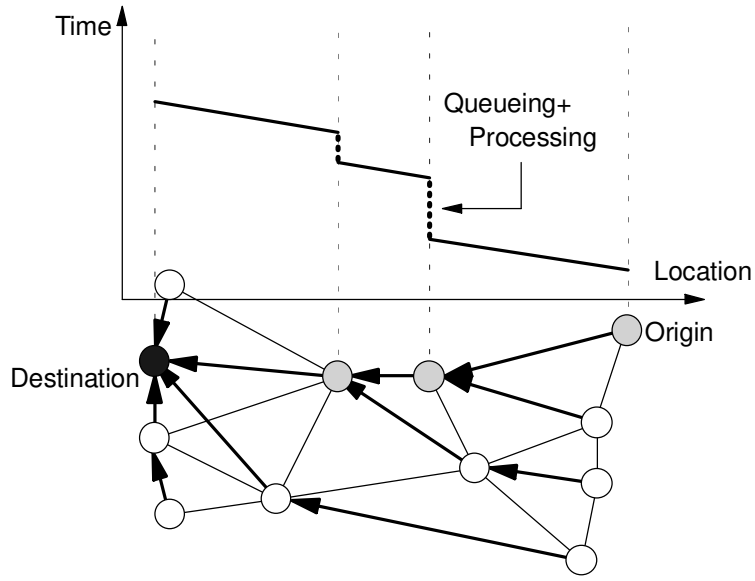

Figure 1: Internet routing on trees

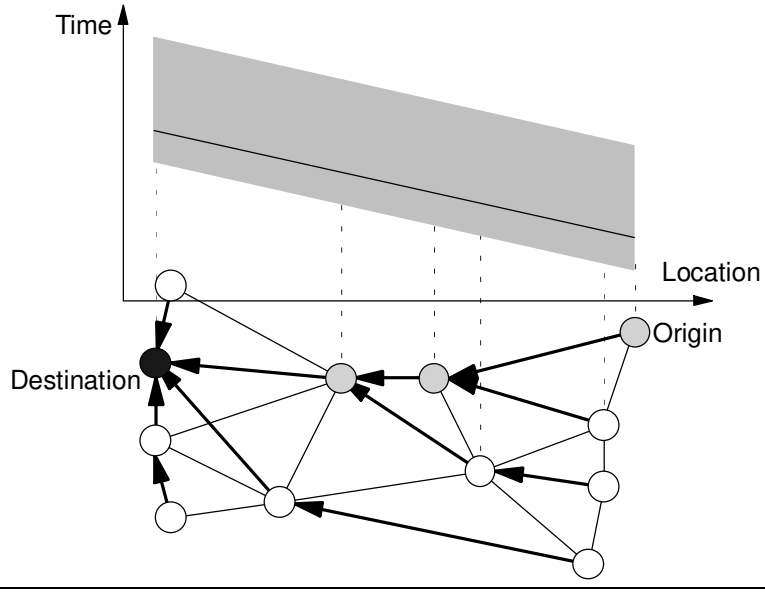

Figure 2: Green-band

2.2 Motion Via Green Bands

Isochronets seek to provide flexible control of contention to accomplish desired QoS. The basic construct used to schedule traffic motion is a time band (green band) assigned to a routing tree (Figure 2). During the green band (shaded), a frame transmitted by a source will propagate

down the routing tree to the destination root. If no other traffic contends for the tree, it will move uninterrupted, as depicted by the straight line.

The green band is maintained by switching nodes through timers synchronized to reflect latency along tree links. Synchronization is per band size, which is large compared to frame transmission time. It can thus be accomplished through relatively simple mechanisms. Furthermore, synchronization errors can be easily contained. Routing along a green-band is accomplished by configuration of switch resources to schedule frames on incoming tree links to the respective outgoing tree link. A source sends frames by scheduling transmissions to the green bands of its destination.

In similarity to CS or burst switched networks green-bands allocate reserved network resources. However, the units to which resources are allocated are neither point-point connections, nor traffic bursts, but routes. Routes represent long-lived entities and, thus, processing and scheduling complexities can be resolved over time scale much longer than latency.

2.3 Route Division Multiple Access (RDMA)

Route Division Multiple Access (RDMA) is employed in Isochronets to assign bands (or bandwidth slices) to routing trees. The main role of RDMA is to resolve contention for the outgoing tree link when frames arrive simultaneously to a switching node. The allocation of synchronized time bands to routing trees and resolution of frame collisions are the primitive constructs used by RDMA to control traffic motions and QoS.

It is important to comment at this point that the allocation of routing trees is per se an interesting and complex problem. Nevertheless, its more general study is left for future publications. In the context of this work, it is assumed that trees are allocated as in traditional networks, based on some optimization criteria such as, for example, minimal distance between nodes. The tree allocation problem needs to be solved only once off-line with respect to Isochronets operations.

Bands need not occupy the same width throughout the network. Indeed, one can view a green band as a resource which is distributed by a node to its up-stream sons (as long as the bands allocated to sons are scheduled within the band of the parent). In particular, if the bands allocated to two sons do not overlap, their traffic does not contend. By controlling band overlaps, switches can fine-tune the level of contention and statistical QoS seen by traffic.

When operating through the simplified bands described in Section 2.2, frames avoid node processing delays but still incur random queueing delays when contenting for outgoing links. These *contention bands* are only one sort of band provided by RDMA. *Priority bands* are allocated to sources requiring absolute QoS guarantees, similar to a circuit service. Traffic from a priority-source is given the right of way, by switches on its path, during its priority band. Unlike CS networks, however, priority sources do not own their bands. Contention traffic may access a priority band and utilize it whenever the priority source does not. During *broadcast bands*, the routing tree is reversed and the root can broadcast to any subset of nodes.

The collision resolution modes used in RDMA are distinguished in terms of the signs "-", "+", and "++". In RDMA- all the colliding frames but one are discarded. In RDMA+, when collision occurs during a band, one frame proceeds and the others are buffered. RDMA++ is similar to RDMA+, but additionally stores frames beyond band termination, rescheduling them during the next band.

One may view these mechanisms to schedule traffic motions via band allocations as a media-access technique. The entire network is viewed as a routing medium consisting of routing trees.

Bandwidth is time- and space-divided among these routes. Sources need access respective trees during their band times, seeing the network as a time-divided medium.

2.4 Further Remarks

In this section, a few observations are made regarding Isochronets. Multiple simultaneous routing trees can schedule transmissions in parallel (have simultaneous green bands), depending on the network topology. For an extreme example consider a fully connected network: all trees to all nodes can be simultaneously active without interference. In more realistic examples, significant parallelism can be accomplished. Figure 3 shows two non-interfering routing trees.

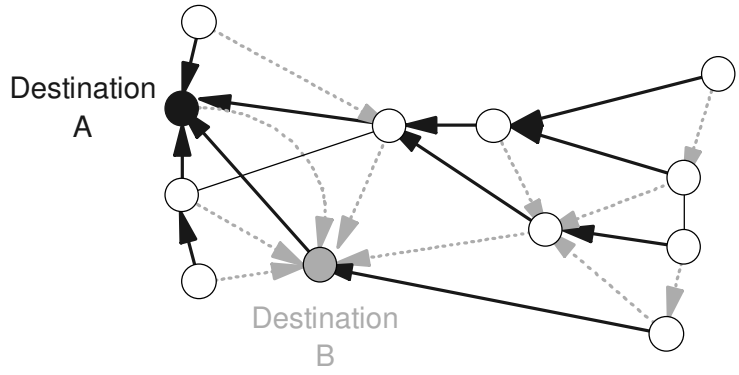

Figure 3: Multiple non-interfering trees

All stack layers above the media-access layer are delegated to interfaces at the network periphery. A typical stack organization for Isochronets is depicted in Figure 4.

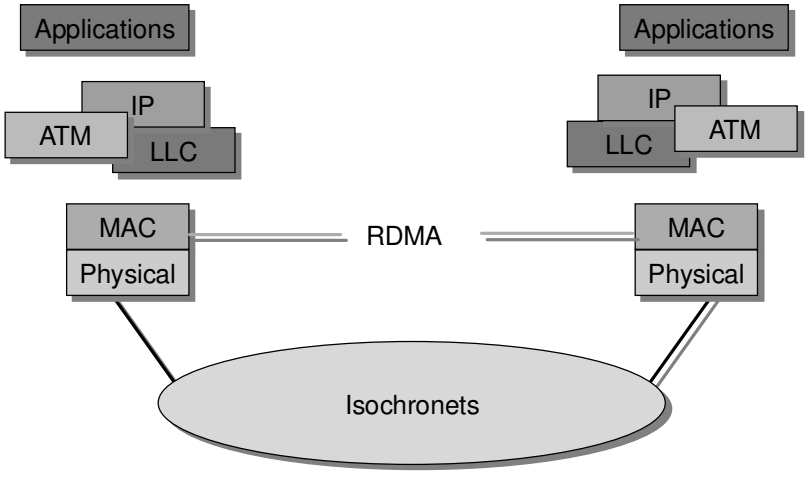

Figure 4: Multiple protocol stacks in Isochronets

Finally interconnection of Isochronets can be accomplished via media-layer bridges using extensions of current well-understood technologies. Conversions need only handle physical layer interfaces and media-access control. Above the media access layer, interconnection becomes transparent. Contrast this with the problem of internetworking two distinct high-speed network architectures via higher-layer gateways.

3 Related Work

3.1 Circuit-Switching (CS), Packet-Switching (PS), and In-between

If the band associated with a routing tree consists of priority-bands only, that tree is operated in an optimized circuit-switched (CS) mode [15]. That is, each source is allocated a circuit (priority band) to the tree root. The form of CS supported by Isochronets is superior to traditional CS as circuits only get priority over band usage but do not own it. In a situation where the entire band has been allocated to priority bands serving real-time isochronous traffic, non-real-time data traffic may take advantage of underutilized parts of the band.

Consider now an Isochronet operating in RDMA++ contention resolution mode. If the entire band is allocated to contention traffic, frames moving down the tree will be stored and forwarded as in an ordinary packet-switched (PS) networks [15]. The form of PS supported by Isochronets is advantageous to traditional PS in a few ways. First, Isochronets support virtual cut-through mechanisms as frames arriving to a free switch will continue without store-and-forward delay. Second, no headers are processed in Isochronet switches. Third, buffered frames are aggregated into larger units and transmitted at once, improving the efficiency of buffer retrieval. Fourth, contention happens only among frames to the same destination (and not among non correlated traffic).

Isochronets, it may be argued, could potentially under-perform PS networks due to the time-division of bandwidth among routes. In situations where significant traffic bursts are randomly generated at different routes with other routes empty, the bandwidth committed to unused routes will be underutilized while the routes serving a burst may have insufficient bandwidth to handle it. A PS network would have permitted the traffic burst to move into the network and utilize its entire band without pre-allocation. Typically, however, admission-control policies will prevent large bursts from entering the network. Such mechanisms as leaky-bucket [13] reduce the effective bandwidth available to any given source. A PS network governed by admission policies which limit source bandwidth, presents no advantage over an Isochronet which limits the bandwidth to sources through pre-allocation to routes.

In summary, at the two extremes, Isochronets compare favorably with CS or PS networks. In-between, Isochronets can be operated to span a spectrum of switching techniques of superior performance characteristics to both, as evidenced by the report of performance (Section 4).

3.2 Other High-speed Architecture Proposals

We now compare Isochronets with non-traditional high-speed switching techniques including wavelength division multiplexing (WDM) [1, 4, 5], Highball [10], and Linear Lightwave Networks (LLN) [14]. WDM networks, like Isochronets, provide dedicated access to destinations via appropriate allocation of wavelength. Routing is accomplished by configuring nodes to switch wavelength to provide source-destination connectivity. Contention among simultaneous transmissions to the same destination must, in similarity to Isochronets, be resolved at switches. WDM networks too may be configured to support circuit-like services and multicasting. In similarity to Isochronets, WDM provide media-access layer networking. One can view Isochronets as a time-domain allocation of bandwidth among destinations, of similarity to the frequency-domain allocation used by WDM networks. The two architectures are orthogonal rather than competing alternatives. The main advantage of Isochronets over WDM is their independence of the transmission

medium technologies. Also, optical tuning of switches at incoming traffic rates is beyond the current state of the art. To cope with this limitation, current implementations of WDM use dedicated wavelengths between node pairs. Packets may only be sent directly to a node's peer. At the peer, packets need to be processed in order to determine the destination route. Isochronets do not require such processing and switch routing configurations over sufficiently long time periods to permit use of optical switches and, thus, all-optical networks.

The Highball network proposal [10] bears some similarity to Isochronets. Nodes schedule traffic bursts by configuring the switches to support uninterrupted motion similar to train motions through intersections. Nodes broadcast requests to all other nodes, specifying their data transmission needs to all possible destinations. This information is then used to compute a train schedule at each node and establish time intervals during which output links are dedicated to specific input links. The scheduling problems are NP-complete and are thus solved through heuristics. Additionally, the schedules computed by different nodes must be consistent and nodes must maintain fine synchronization on time scales much shorter than used by Isochronets. Highball networks are geared to serve traffic that can tolerate the latency delays between requests to transmit and their granting. Regulating traffic motions through switch configurations is similar to the approach taken by Isochronets. However, this is where the similarity ends. Trying to switch configurations to match the structure of bursty demands is in contrast with the Isochronet solution of switching routes, independent of immediate demand patterns. The complexity of burst scheduling, the need for fine synchronization, and other derivatives of the approach do not arise in Isochronets. Isochronets do not require non-conflicting global schedules. Instead, they settle for contention resolution by local switches and myopic scheduling by sources. Nor are Isochronets restricted to serve the kind of traffic targeted by Highball networks.

LLN communicate using wavebands. When two nodes want to communicate, the same waveband can be assigned to both only if their paths are disjoint. When combined, different wavebands cannot be separated at switches. Thus, the assignment of different wavebands for connections becomes even more complicated than in WDM, since it is necessary to make sure that all combined wavelength in a given link do not interfere with the new wavelength that is being assigned. The scheduling of wavebands to incoming calls is NP-complete and thus heuristics are used to solve the problem. This technique also divides bandwidth in the waveband domain, in similarity to WDM. Nevertheless, no processing or buffering is necessary at intermediate switches in similarity to Isochronets. It is designed to serve applications that can tolerate the long call set up delay to find a proper wavelength for the call. No such delays are incurred in Isochronets. Also, since the perfect schedule is not attainable, the bandwidth may be underutilized.

4 Isochronet Switch Architectures

There could be many potential Isochronet switch implementations. This section reviews briefly the organization of two such implementations: an electronic (Section 4.1) and an all-optical (Section 4.2). The electronic version operating at gigabits per second was implemented by the authors at Columbia University. The description of this implementation together with a more detailed description of the optical implementation proposed in this section can be found in [7].

4.1 Electronic Organization

This section describes the electronic RDMA+ Isochronet switch implementation. The switch architecture is depicted in Figure 5. Input fiber lines feed the input line cards that convert serial

optical signals into parallel electronic signals and store them in internal FIFO buffers while contention for the switching fabric is being resolved. There is no protocol processing at the interfaces to the internal buffers, thus simplifying their implementation. Fiber rates are on the order of gigabits per second.

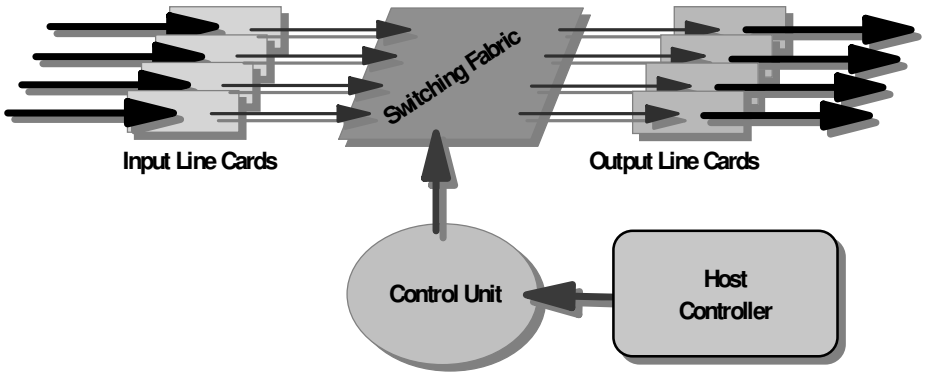

Figure 5: Overall electronic design

The control unit contains a Configuration Table (CT) with configuration information, including the current enabled routing trees, used in implementing input to output line connections and in resolving contentions for outgoing lines. The CT together with information on which incoming lines are active are input to the decision engine at the control unit that grants access to the switching fabric and that issues control signals to switch the fabric.

The converted parallel electronic bits plus control unit signals are input to the switching fabric that provides the required input/output connection. The switching fabric outputs are connected to the output line cards where they will be converted into sequential optical signals for transmission. The output cards may also need to delay the signals before transmission, as it will be explained later.

The CT is periodically stored from the host controller, a general purpose machine executing control and management software. The primary function of such software is to compute the allocation and switching of bands. Configuration and switching of bands, execution of protocols for band synchronization and allocation, and other control and management functions processed by the host are relatively slow and can be entirely accomplished by software.

The hosts in the network exchange control information using the Ethernet line card connected to their buses. This separated Ethernet channel simplifies the implementation and achieves more flexibility in the prototype.

Isochronet switches thus separate high-speed transmission path and access arbitration functions, handled by trunk interfaces and switching fabric, from network control and management functions, handled by slower-speed logic. This separation allows Isochronets to scale favorably for a broad spectrum of trunk speeds without requiring changes of the network control mechanisms.

The interconnection fabric is a fully connected mash, that is, each input is connected to all outputs. The conception is performed using multiplexers controlled by the control unit, as depicted in Figure 6. When a new band begins, the control unit enables the multiplexers connected to active output lines, that is, output lines participating in a routing tree. When input line cards receive information, their busy lines become active. On the basis of which input lines are active, on

which one has priority, and on the current band configuration, the control unit sends control bits to the registers connected to the control lines of the multiplexers. The registers keep the configuration of the switch until the status of any input line changes or the current band ends.

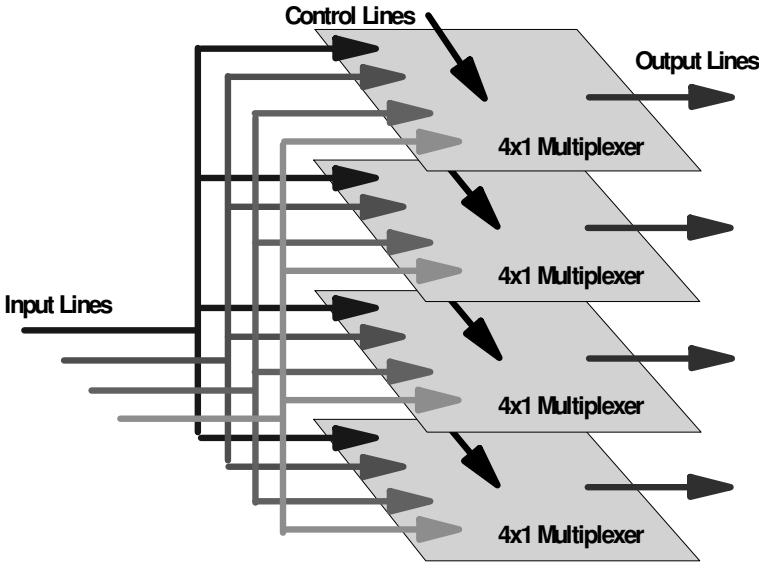

Figure 6: Switching fabric

There are other possibilities to implement the interconnection fabric. One choice would have been a time-divided bus [18]. If n trees can simultaneously cross the switch, the bandwidth supported by the switching fabric is at least n times larger than the respective trunk bandwidth. Such design is easier to scale, since all that is necessary to increase the bandwidth in the bus is to provide new buses in parallel. Nevertheless, the design is more complex than the interconnection network presented. Yet another possibility would be a multistage interconnection network [16, 17], which would have provided less interconnection complexity. Nevertheless, such implementation would increase the internal delay in the switch.

4.2 All-Optical Organization

This section describes the all-optical RDMA- Isochronet design. An all-optical realization of Isochronets must avoid buffering at intermediate switches. The design proposed in this section uses wavelength division multiplexing (WDM) to allocate one wavelength for each band. Another possibility would be to time-divide within the same wavelengths, but this solution is left for future work.

The architecture for a single tree per band is depicted in Figure 7. Each wavelength is depicted using a different gray scale. Incoming wavelengths are first fed into a selector (explained later) and then multiplexed through a single optical broadcast link (the interconnection fabric) connecting all source and destination links. At each output link, a slowly-tunable receiver picks the wavelength of the trees sharing the link. The receiver is directly connected to a slowly-tunable transmitter that regenerates the wavelength in its output link.

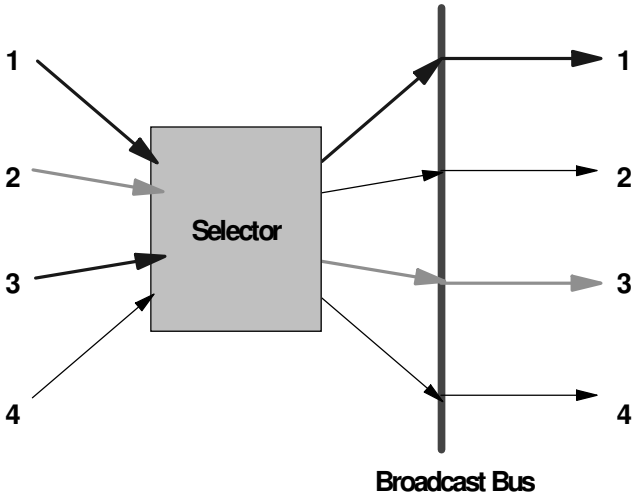

Figure 7: All-optical switch implementation: one tree per band.

Contention in the all-optical implementation is resolved by discarding one of the frames. Contention occurs when two or more of the input links is pouring frames using the same wavelength. In this circumstance, one of the sources must be enabled while the others must be stopped, otherwise all data may be lost. This functionality is achieved through the selector, the only electronic component in this architecture. Its function is to detect incoming signals from the links and immediately grant access to one of them, shutting the others, usually based on priority information derived form the CT in the control unit.

The number of bands necessary in this implementation is, in the worst case, equal to the number of nodes in the network. This number can be improved by reusing bands in multiple broadcast buses, the equivalent to multiple trees per band in the optical domain.

The optical transmitters and optical receivers in this solution are fixed for the duration of the band. They need to be tuned only when the band ends to configure the new trees. However, this can be accomplished very slowly between bands, at speeds independent of transmissions.

It is important to notice that even though the all-optical implementation uses RDMA-, all bands are opened all the time, avoiding synchronization of bands.

Nevertheless, frame loss may occur in this scheme during contention bands. It is best that an all-optical implementation uses slots in each link to decrease the probability of frame loss. In the following, the frame-loss probability for the slotted implementation when arrivals are Poisson is analyzed. Also, an extension to the basic scheme to reduce the frame-loss probability is suggested.

Let $\frac{\lambda}{n}$ be the input rate (as a percentage of the peak rate 1) of each input link to a particular switch, and n be the number of input links to the switch. The probability of no transmission from a source link during a slot is $1 - \frac{\lambda}{n}$. Thus, the average successful transmission rate during a slot is $1 - \left(1 - \frac{\lambda}{n}\right)^n$ (that is, if at least one source transmits). As $n \to \infty$, the rate becomes $1 - e^{-\lambda}$. The expected success probability is $\frac{\left(1 - e^{-\lambda}\right)}{\lambda}$. Finally, the expected loss probability is $1 - \frac{\left(1 - e^{-\lambda}\right)}{\lambda}$. When $\lambda \to 1$ (loaded system), the expected loss is e^{-1} (less than 37%).

It is possible to improve the performance of this scheme. Multiple copies of the same frame may be sent, thus decreasing the loss probability for the frame. If each frame is repeated m times, the loss probability becomes e^{-m}. Thus, m may be computed from the maximum loss rate r that can be tolerated in the system: $r \le e^{-m}$ so that $m \ge -\ln r$. For example, $m = 4$ insures less than 2% loss rate when the system is heavily loaded and the number of input sources is big.

To complete the design using the analysis above, a filter is placed at the traffic sources (before the traffic enters the network), which disturbs the input traffic frames interarrival times to the network and makes them exponentially distributed (thus generating a Poisson arrival process to the network). Each source sends m copies of the same frame, where m is computed from the tolerated loss rate.

5 Network Control

5.1 Clocks

Clock management is central to Isochronets. A switch must maintain clocks to allocate bands on each of its links. The first problem to consider is that of selecting clock periods for band repetitions. Let U indicate the shortest clock unit used in band allocation. Let P denote the periodicity of the clock measured in U units. For example, let U= 1μs and P =125U; that is, after 125μs the clock returns to 0. Time may then be indicated in terms of period counters similar to seconds, minutes, hours etc. For example, the time <12, 3>, with the above U and P, means 3 periods (125μs long) plus 12μs.

Typically, allocations of green bands on a link will be repeated periodically. The periodicity may vary with the type of traffic served. Low duty traffic such as file transfers may use periods of long duration, whereas interactive voice or video traffic may use much shorter periods. Traffic may also vary in terms of typical frame sizes. A telephone call may use very short frames while an image transfer task may utilize very long frames. Obviously, the green band should exceed the size of a typical frame used. Consider the choices of U and P above over a 2.4 Gb/s link. During a period of P = 125μs, some 300 Kb can be transmitted. If the link is equally shared among 3–6 trees, this means that each tree can be allocated an average of 50–100Kb. This average over time may result from allocations of varying sizes over periods of different duration. Voice and video traffic using ATM cells [3] may be allotted some 115–230 cells in a green band repeated every 125μs. Client-server traffic using IP frames of 10 Kb may see a green-band of some 50–100 frames per period of 10 P = 1.25 ms. Additionally, since link speeds may vary greatly, Isochronets may wish to use different periodicity over links. For example, a link of 155 Mb/s may use a period of 16 P = 2 ms. Arrivals over this link will be buffered and delivered to higher speed links.

Therefore, Isochronets may wish, in general, to allocate bands over links to reflect the frame-sizes and periodicity required by different traffic sources as well as link speeds. Discussion of this general case, however, is beyond the scope of this introductory paper. Henceforth, it is assumed that all traffic requires the same periodic repetition of green bands and all links have the same speed. The concepts and protocols discussed below generalize to cases of varying periodicity and link speeds.

5.2 Band Synchronization

This section considers the problem of synchronization in the electronic implementation of Isochronets (see reference [18] for an alternative protocol). The all-optical implementation discussed has all bands opened all the time, thus the synchronization problem is eliminated.

There are two kinds of synchronization necessary for Isochronets operations: clock synchronization and band synchronization. To solve the clock synchronization problem, any of the traditional protocols such as the Network Time Protocol [11] may be used. The band synchronization mechanisms are discussed in this section.

Synchronization must ascertain that the bands on incoming links must be strictly contained (when propagation delay is added) within the band time of outgoing link (we call this the band constraint) and, additionally, ensure the following overlap constraints: the intervals of different trees on the same link do not intersect. The goal of band synchronization is to establish band initialization values that satisfy both the band constraints and the overlap constraints for all links. The latency delay parameter in each link can be tuned to meet the band constraints by the switching node at which the link is incident. These ideas are depicted in Figure 8. The top right link is shared by two trees depicted using two shades of gray. The respective band initiation and termination times are marked at each node in their respective clocks. The clock size is equal to the cycle period.

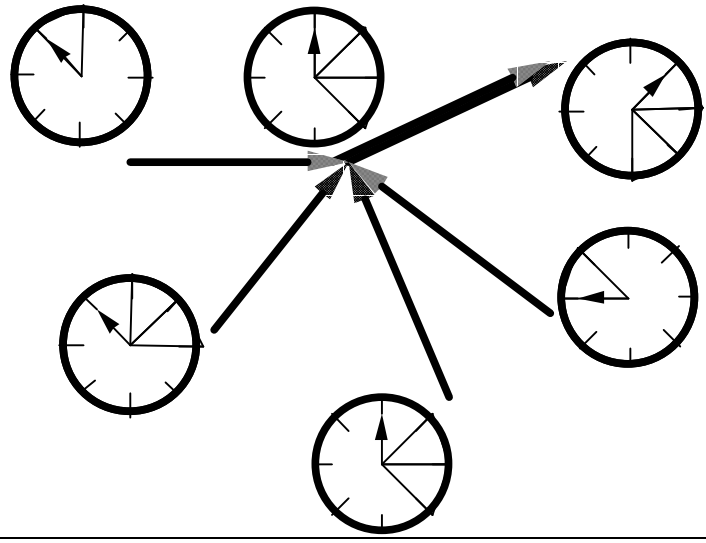

Figure 8: Band allocation and synchronization

The synchronization problem can be stated as follows (see reference [6] for a formal definition of the problem). Given a network, a collection that contains sets of trees that participate in the same band (that is, trees that do not interfere). The goal is to find: (1) for each node in the network, and for each band, the initiation and termination times of the band; and (2) delays in each link in the network that participates in some tree. The solution is restricted so as to satisfy the band and overlap constraints. The minimization criterion is to avoid wasting bandwidth.

Notice that the link delay is a parameter of the problem. The idea is to incorporate a hardware delay component after each outgoing switch port in Figure 5. The delay component is implemented as a FIFO buffer and its purpose is to introduce an artificial tunable delay in the link.

The following is an optimal solution: make all the link delays equal to 0 modulo the cycle period and all the band initiation and termination times the same at all the nodes. It is thus necessary to make sure that the link delay is 0 for all links in the network. To accomplish this, it is assumed that the delay element may be set to any value between 0 and C (the cycle period). Protocol 1 ensures that the link delays at each node is 0 at all times.

Protocol 1: Given two nodes A and B, the protocol sets the delay in the link between A and B (l(A,B)) to 0. The delay element at the output of A to B is d(A,B). All operations are modulo C, the clock period (see Figure 8).

1. A->B: Request For Delay (RFD) message for link l(A,B).
2. B->A: Delay Response (DR); B marks time T at which DR is sent.
3. A marks arrival time R of DR. A measures the offset S=R-T.
4. If d(A,B) > S, set d(A,B) to d(A,B)-S. Otherwise, set d(A,B) to d(A,B)+S.

5.3 Band Allocation

The goal of band allocation protocols is to establish appropriate band duration. The allocation must satisfy the band and the overlap constraints (Reference [18] discusses an alternative protocol and reference [6] contains a formal definition of the problem).

First observe that, since all the trees are spanning, all the nodes must know where each band is allocated in the cycle. Thus, in order to allocate bands, it is necessary to communicate the allocation to all the nodes in the network.

The band allocation problem can be solved in a manner similar to band synchronization, by setting the link delays to 0 (modulo the cycle period) and the band initiation and termination values to be the same at each node. To complete band allocation, it is necessary to set what the band initiation and termination times should be. There are many solutions to this problem. One solution is to allocate bands according to traffic demands, which can be easily pursued: a band of size X on a link with bandwidth B allocates $\frac{XB}{C}$ bandwidth to the band. Other solutions may dynamically adapt the size of the bands according to demand. Random data traffic persisting over periods longer than the band period can be statistically averaged and then result in local re-allocation of incoming bands. Thus, for example, wider band may be allocated to trees showing a greater degree of activity by taking band from other links (or trees) producing lesser activities. The problem can be cast in terms similar to the management of memory references in computers. Indeed, the band associated with a complete clock cycle (for band period) may be viewed as a memory to be shared. A least recently used (LRU) algorithm may be used to shift band allocations among sources. The study of such algorithms is left for future work.

6 Preliminary Performance Evaluation

This section studies the performance of Isochronets using both analytical tools (Section 6.1) and simulations (Section 6.2).

6.1 Analysis

This section studies a performance model of RDMA. Consider RDMA serving ATM cell [3] traffic generated from Poisson sources. Sources sending traffic to a given destination compete for

the use of a shared band subject to RDMA contention resolution. For simplicity, assume that the same band is provided to all sources (in the more general case, the band can be divided to a few sub-bands where arrival rates will vary, depending on source access provided).

In the study of RDMA- and RDMA+, the assumption is that arrivals are Poisson within a band, since there is no buffering beyond the band limits within the network. For RDMA-, one can consider the band as a shared service mechanism. Cell arrivals to the band represent a renewal process. During transmission of cell, arrivals of other cells will be discarded by the RDMA- mechanism at switching nodes. One can use, therefore, Type-I Counter models [8] to represent RDMA-. In other words, a cell arrival may be considered as a counter mechanism which is blocked by a successful cell transmission. The process of interest is the arrivals of cells whose transmissions are successful. This process is, again, renewal process whose interarrivals are defined as the sum of two independent random variables representing cell interarrivals and cell transmission times. With time measured in cell-transmission duration units, a traffic arrival rate to a given band of λ (cells per cell transmission period), the distribution of successful interarrivals is given, therefore, by the convolution of the interarrival and cell duration distributions:

$$F_{RDMA-}(t) = \Pr[\text{Interarrival of successful cell} \leq t] = 1 - e^{-\lambda(t-1)}.$$

One can compute the average rate of successful cell transmissions (from the expectation of $F_{RDMA-}(t)$) to be $S_{RDMA-} = \frac{\lambda}{\lambda + 1}$. Thus, the expected cell loss rate is given by $L_{RDMA-} = \lambda - \frac{\lambda}{\lambda + 1}$. The percentage loss amounts to:

$$LP_{RDMA-} = 1 - \frac{1}{\lambda + 1}.$$

When the load is low, the loss rate is almost 0. It approaches 50% when the load reaches saturation ($\lambda = 1$), giving a very impressive result for a system without buffering. The cell delay will be just the transmission and propagation delay, since no queueing is incurred in this system. Thus,

$$W_{RDMA-} = 1.$$

W_{RDMA-} measures the average queueing delay seen by a cell between arrival and departure from a switch. In addition to this queueing delay, a cell sees a latency delay through the network. So the average delay seen by a cell is given by:

$$T_{RDMA-} = 1 + L,$$

where L represents the average latency[1].

In the case of RDMA+, cells are lost only when they are queued beyond band termination. We want to compute the number of queued elements at the end of a slot. If we consider the band to be large enough, the problem reduces to finding the average queue size in a M/D/1 queueing system, which is simply [9] $q = \frac{\lambda}{1 - \lambda} - \frac{\lambda^2}{2(1 - \lambda)}$. The loss rate is just the mean queue size found at the end of a band divided by the number of packets sent during a band (λB, where B is the band size). Thus,

$$LP_{RDMA+} = \frac{2 - \lambda}{2B(1 - \lambda)}.$$

We expect the mean cell delay in this system to be the same as the one for M/D/1 system in equilibrium, if the band is large enough. Then, we may write:

---

[1] The same observation is valid for all queueing delays in this section, and T may be obtained from W by adding L in all cases.

$$W_{RDMA+} = \frac{\lambda}{2(1-\lambda)}.$$

Let us analyze operations under RDMA++ discipline. The band may be viewed as a service mechanism with periodic vacations. This can be modeled as an M/D/1 queue with periodic vacations. The solution of such models is generally very difficult (see, for example, [12] for a discussion on the subject). For the mean queueing delay, we approximate the solution by the same method to compute the mean queueing time for M/G/1 systems with vacations [2]. In RDMA++ the vacation periods are generated only due to the ending of a band. With the vacation period between bands of duration V (cells), the queueing delay of a contention band using RDMA++ may be approximated by:

$$W_{RDMA++} = \frac{\lambda}{2(1-\lambda)} + \frac{V}{2} \cdot \frac{V}{V+B} \cdot \frac{1}{1-\lambda}.$$

The calculation of this formula is as follows. We compute the mean residual service time for the busy and vacationing periods. For the busy period, the calculation is the same and gives the queueing delay for an M/D/1 type of system [2]. For the idle period, the mean residual service time is the mean vacation period ($\frac{V}{2}$) times the probability of being on vacation ($\frac{V}{V+B}$). We then divide the mean residual service time by the idle period ($1-\lambda$), to obtain the mean waiting time [2]. This last term can also be interpreted as the penalty in the delay incurred by the burst of cells generated during the vacations, present when the band begins.

If the band is divided (in part or in whole) among priority bands devoted to certain sources, the average delay will not change as long as the network resolves contention in a work-conserving manner (for example, pre-emptive resume or non-pre-emptive priority mechanisms). This is where the shared CS (SCS) greatly improves on classical time-division CS. Indeed, suppose traffic to the band is divided among n sources using traditional CS. Suppose, further, that traffic is uniformly generated by each source at a rate of $\frac{\lambda}{n}$. The utilization of a given circuit remains $\rho = \frac{\lambda}{n} \cdot n = \lambda$. However, the vacation time increases and circuit bandwidth available decreases to result in:

$$W_{CS} = \frac{n\lambda}{2(1-\lambda)} + \frac{V + B(n-1)/n}{2} \cdot \frac{V + B(n-1)/n}{V+B} \cdot \frac{1}{1-\lambda}.$$

In other words, the queueing delay increases by a factor of n with additional delay in waiting for the circuit band. Therefore, the SCS allocation of priority bands by Isochronets greatly outperforms traditional CS, while providing sources so desiring the same performance guarantees as CS does.

6.2 Simulation Studies

In this section we provide a performance evaluation of Isochronets obtained through simulation studies. The topology studied is depicted in Figure 9. It is a symmetric configuration that allows the overlapping of 3 non-interfering trees. An example of 3 non-interfering trees to destinations 1, 6 and 8 is depicted in Figure 10. These destinations thus can share a band. Two additional bands are sufficient to serve the 6 trees of the other destination nodes.

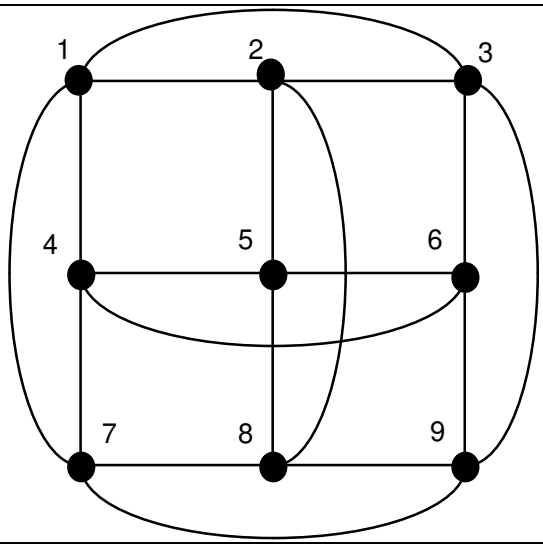

Figure 9: Simulated network topology

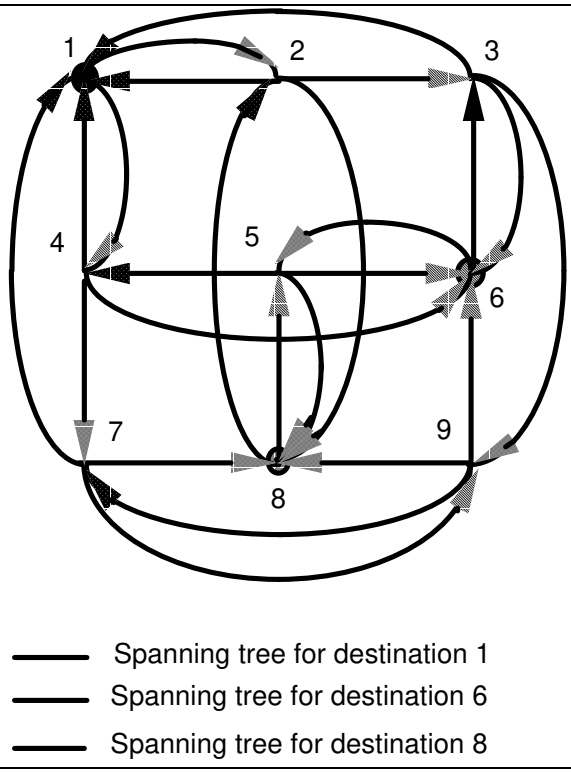

Figure 10: Allocation of trees in one band

The simulation model works as follows. Each node generates ATM cells according to a Poisson process. Destinations are assigned to cells according to a uniform distribution. The link speed used is 2.4 Gb/s, resulting in 177 ns transmission time per cell. The clock period is 125 μs. The propagation delay in each link is negligible (equivalent to 1 cell transmission delay). The bands to all destinations are of the same size, since the traffic is uniformly distributed. Each cell waits for

the proper destination band at the source nodes and then moves through the network down the respective tree. Our goal is to give a broad comparison of PS, CS, and RDMA++.

The PS simulation uses the same trees allocated for RDMA++ (see Figure 10) and cut-through. Processing delays at nodes are included. The CPU at each switch was assumed to operate at the same rate of one input link. This means, for example, that if 50 instructions are necessary to process each ATM cell, we are simulating a 283 MIPS machine for PS, an unrealistic assumption. The CS simulation queues cells for that circuit until the circuit becomes available.

The simulation was run for periods where 10,000 ATM cells per input node were generated. After each of these periods, all statistics were saved and reset. Two RDMA++ experiments were conducted. In the first, all traffic had the same priority (RDMA++<c>). In the second, priority traffic was generated as follows. Each band was equally partitioned to priority sub-bands, one for each input node (RDMA++<p>).

Figure 11 depicts the mean packet delay (in μs) for the experiments we have conducted. The input traffic load is given as a percentage of the 2.4 Gb/s maximum input rate at each node. As it can be seen, PS has a steady performance until the input load 50% saturates the CPU capability at the nodes with network delays growing unbounded. CS has a similar behavior, the unstable point being 30%. RDMA++ has a stable performance. Both RDMA++<c> and RDMA++<p> have the same mean packet delay characteristics, as expected from queueing analysis [9], and thus overlap in the figure. The "Pr." curve plots the mean delay for priority traffic generated for the RDMA++<p> experiment. Priority was assigned randomly to ATM cells according to an uniform distribution. Priority traffic was scheduled to access the network during its priority band, thus not incurring admission delays. The delay incurred by the priority traffic is thus only the propagation delay and contention with other cells scheduled at the beginning of the priority band.

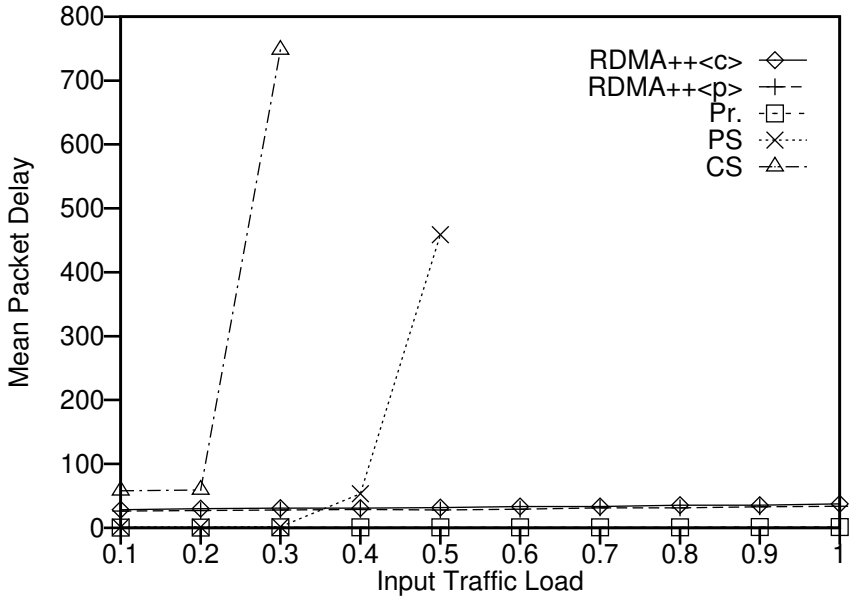

Figure 11: Mean network ATM cell delay for Poisson arrivals (in μs)

Figure 12 shows the network behavior when sources generate bursty traffic according to an on/off model, where the on and off periods are geometrically distributed in the number of cells. The mean on period is 10 ATM cells.

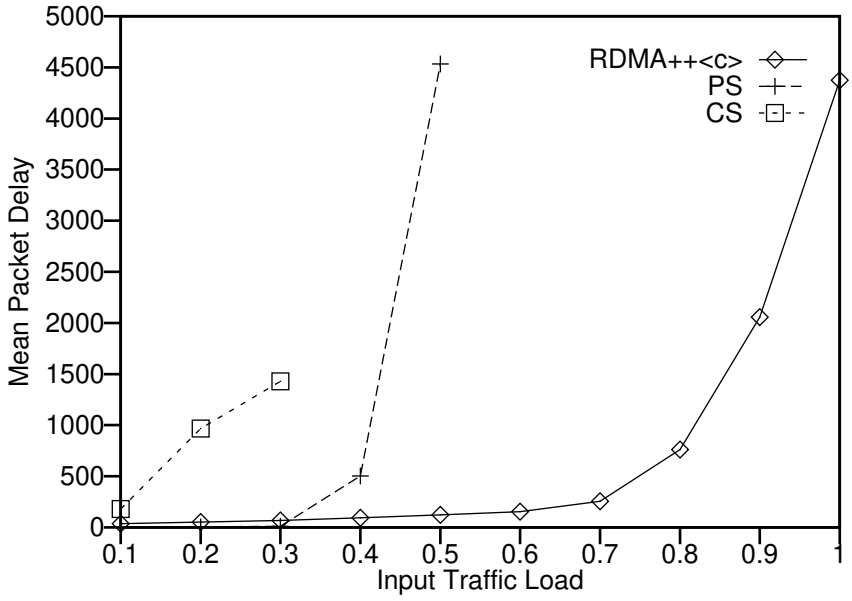

Figure 12: Mean network ATM cell delay for bursty arrivals (in µs)

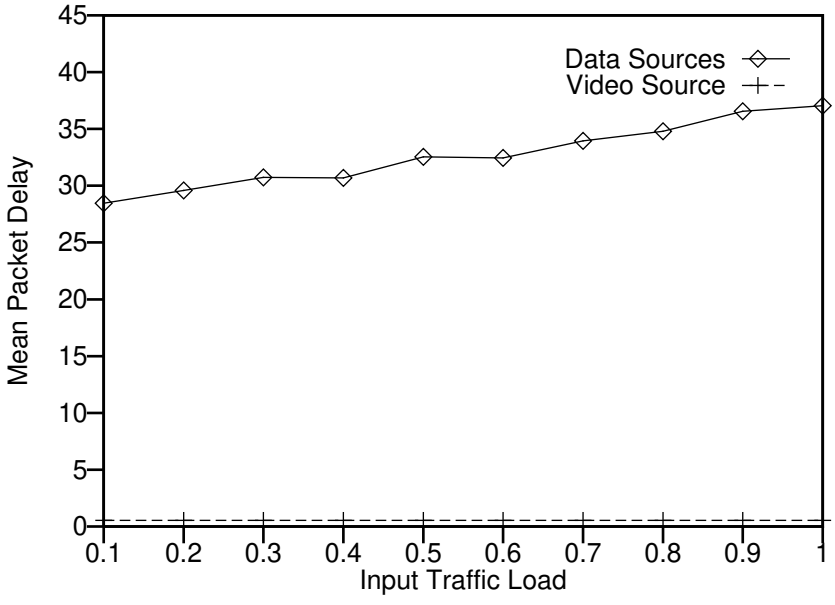

Figure 13: Mean network ATM cell delay when operating with video source

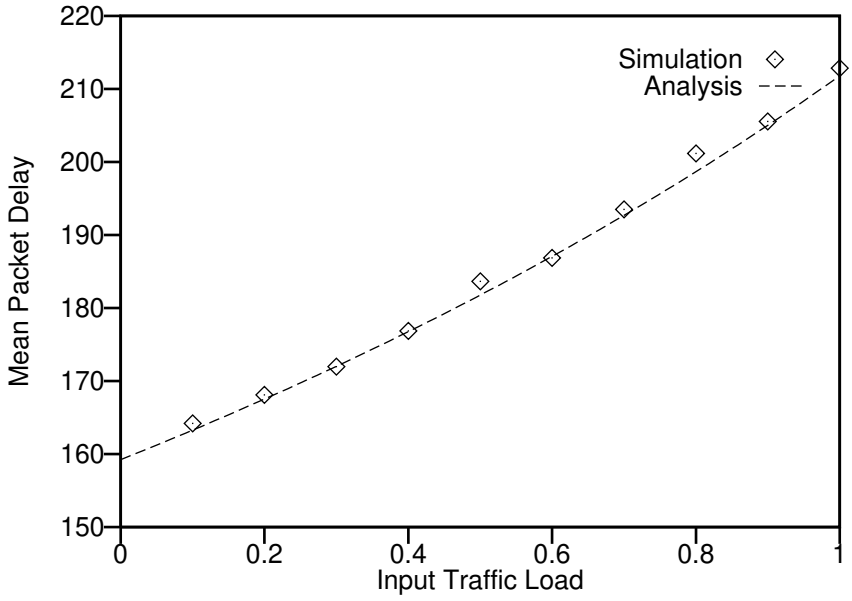

Figure 14: Simulation and analysis results for RDMA++ (time measured in cell transmission delay)

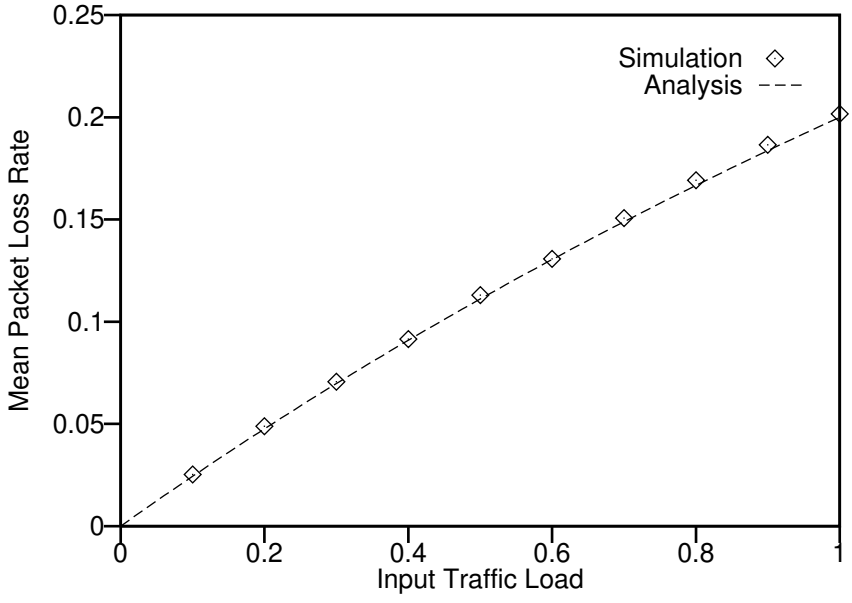

Figure 15: Simulation and analysis results for RDMA-

In Figure 13 we display a multimedia experiment. Source 9 sends motion picture frames to destination 1. All other sources send normal data traffic generated according to a Poisson process at the load specified in the x-axis. The video traffic is scheduled to be generated during the source's priority band, which is of size 10 cells during each 125μs cycle. As it can be seen, the

network provides high-quality service to the video source and normal traffic proceeds normally. The isolation of both traffic types can be accomplished by simple band tuning (in μs).

In Figure 14, we show the comparison[2] of analysis and simulation results for the RDMA++ mean packet delay. As it can be seen, the results are in good agreement. Figure 15 compares[3] the simulation and analysis results for the RDMA- mean packet loss rate.

Finally, we display in Figure 16 the mean delay when we apply RDMA to the NSF T3 backbone network[4]. We upgraded the link speeds to 2.4 Gb/s[5]. We assumed that the CPU speed for was 100 MIPs for the PS simulation and that some 50 instructions were necessary to process each ATM cell. It is important to notice that the topology of the NSF backbone is not suitable for RDMA since only two trees can coexist in each band. Nevertheless, the performance advantage of RDMA is clear in the figure.

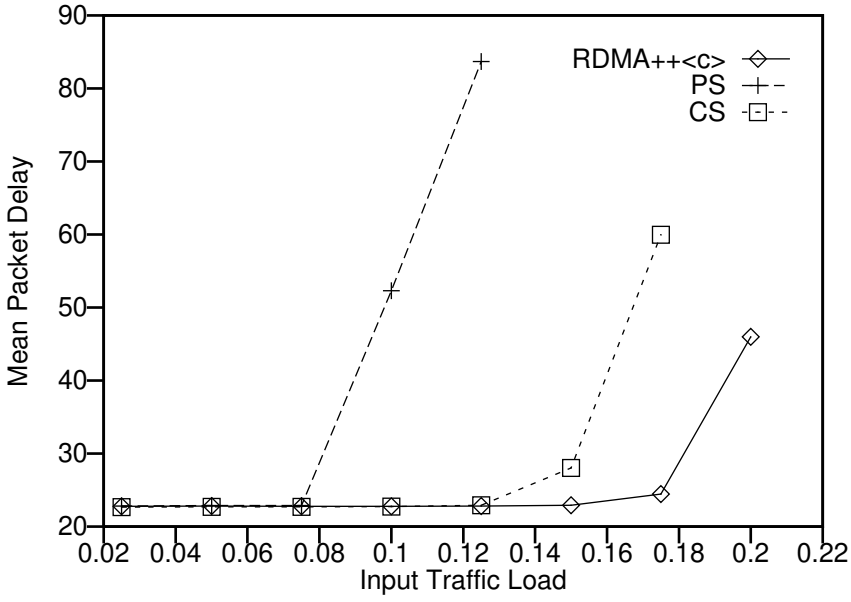

Figure 16: Mean network ATM cell delay for Poisson arrivals (in ms) in the NSF T3 backbone network

Another observation is in order for this experiment. When applied to wide area networks, the time incurred waiting for a particular band is negligible when compared to the propagation delays. For instance, the waiting time for the band in our NSF backbone simulation is at most 125μs (a complete cycle), but the cross-country propagation delay is of the order of 30 ms (240 times

[2] The final destinations in the simulation did not incur transfer delay. Thus, the maximum service rate in the simulation is 4 times the maximum input rate, since there are 4 incoming links at the destination node (see Figure 4.2). The input load in Figure 4.6 is a percentage of the 2.4Gb/s maximum input rate and, thus, λ in the formula for W_{RDMA++} should vary from 0% to 25%.

[3] Each tree is active only 1/3 of the cycle. Thus λ in the formula for LP_{RDMA-} should be scaled between 0 and 1/3 (thus the loss rate never reaches the 50% loss upper bound).

[4] The propagation delays are approximated since the exact measures are not available.

[5] We actually could not run this simulation on our SPARC server due to the huge state space necessary. We ran the simulation at T3 link-speeds and scaled the results to 2.4Gb/s link-speeds.

larger). Thus, the immediate admission seen by frames in a PS implementation is a negligible component of the total frame delay.

7 Conclusions

We now summarize the main characteristics of Isochronets. First, the network layer is reduced to a media-access layer, transparent to the framing layers which reside above it at its periphery. As a result: (1) no frame processing is required in the network; (2) there is no need for adaptation layers at network interfaces; (3) internetworking is reduced to media-layer bridging; (4) the network can adapt to the frame sizes and arrival statistics of sources.

Second, all network layer functions and control are accomplished through a single unifying mechanism: band allocation. This means that by controlling band timers: (1) all network functions: routing, switching, flow and admission controls are obtained; (2) a range of services and guarantees is provided: reserved circuits, contention-based bandwidth on demand, multicast; (3) a spectrum of performance behaviors can be obtained between shared circuit switching and PS with cut-through; (4) a single protocol to set, adjust and synchronize band timers can be used. Unification of these mechanisms permits simpler and more effective network control functions than traditional networks.

Third, network control functions are entirely separated from transmission activities. This leads to a few attractive features: (1) speed-elasticity: transmission speeds can be arbitrarily faster than control speeds; (2) distance-elasticity: the network can extend over local, metropolitan and wide areas; (3) all control decisions accomplished at traffic motion times are local; (4) bandwidth-heterogeneity: the network can incorporate links of different transmission speeds; (5) all-optical Isochronets implementations are feasible.

These features render Isochronets attractive candidates for high-speed network architecture.

8 References

[1] Acampora, A.S. and Karol, M.J., "An overview of light-wave packet networks," IEEE Network Magazine, vol. 3, 29-41, January 1989.

[2] Bertsekas, D. and Gallager, R., *Data networks*, Second Edition. Prentice Hall, 1992.

[3] Boudec, J.Y.L., "Asynchronous Transfer Mode: a tutorial," *Computer Networks and ISDN Systems*, vol. 24, no. 4, May 1992.

[4] Brackett, C.A., "Dense wavelength division multiplexing networks: principles and applications," *IEEE Journal of Selected Areas in Communications*, vol. 8, no. 6, 948-964, August 1991.

[5] Dono, N.R., Green, P.E., Liu, K., Ramaswami, R., and Tong, F., "A wavelength division multi-access network for computer communications," *IEEE Journal on Selected Areas in Communications*, August 1990.

[6] Florissi, D. "Isochronets: a high-speed network switching architecture (thesis proposal)," Tech. Rep. CUCS-020-93, Computer Science Department, Columbia University, 1993.

[7] Florissi, D., and Yemini, Y. "The gigabit per second Isochronet switch," Tech. Rep. CUCS-010-95, Computer Science Department, Columbia University, 1995.

[8] Karlin, S. and Taylor, H.M., *A first course in stochastic processes*, Second Edition. Academic Press, 1975.

[9] Kleinrock, L., *Queueing systems*, vol. I. Wiley, 1975.

[10] Mills, D.L., Boncelet, C.G., Elias, J.G., Schragger, P.A., and Jackson, A.W., "Highball: a high speed, reserved-access, wide area network," Tech. Rep. 90-9-1, Electronic Engineering Department, University of Delaware, September 1990.

[11] Mills, D.L., "Internet time synchronization: the Network Time Protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, 1482-1493, August 1991.

[12] Ott, T.J., "The single-server queue with independent GI/G and M/G input streams," *Adv. Appl. Prob.*, vol. 19, 266-286, 1987.

[13] Sidi, M., Liu, W., Cidon, I., and Gopal, I., "Congestion control through input rate regulation," in *Proceedings of GLOBECOM,* IEEE, Dallas, Texas, USA, May 1989, pp. 1764-1768.

[14] Stern, T.E., "Linear lightwave networks: how far can they go?," in *Proceedings of GLOBECOM,* IEEE, San Diego, California, USA, 1990.

[15] Tanenbaum, A.S., *Computer networks*, Second Edition. Prentice Hall, 1988.

[16] Tobagi, F.A., "Fast packet switching architectures for broadband integrated services digital networks," in *Proceedings of the IEEE*, vol. 78, no. 1, 133-167, January 1980.

[17] Turner, J.S., "Design of a broadcast packet switching network," in *IEEE Transactions on Communications*, vol. 36, no. 6, 734-743, June 1988.

[18] Yemini, Y. and Florissi, D., "Isochronets: a high-speed network switching architecture," in *Proceedings of INFOCOM,* IEEE, San Francisco, California, USA, April 1993.