Isochronets: a High-Speed Network Switching Architecture

Yechiam Yemini and Danilo Florissi

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

Technical Report CUCS-050-92

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 Networks (HSN) problems addressed by Isochronets.

Reducing network processing. Traditional networks sought to maximize utilization of communication capacity via sophisticated processing. 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.

Handling significant diversity of quality of service (QOS) needs. HSN will require unified support of a broad spectrum of QOS demands by diverse traffic, in contrast with current networks, built to support relatively narrow traffic models. HSN applications may depend critically on QOS guarantees. HSN require means to control, finely tune and strictly guarantee QOS.

Handling transient dynamics of random traffic. Traditional networks multiplex large number of uncorrelated random traffic sources. They depend on the laws of large numbers to operate in statistical steady-state regimes. HSN elements, in contrast, may operate at aggregate bandwidth of a similar order as sources, losing the advantages of the laws of large numbers. These sources, furthermore, may involve isochronous highly correlated traffic. HSN must handle operations in dynamic transient traffic regimes.

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. The network need protect itself against large traffic bursts and ensure smooth motion of admitted traffic. Additionally, HSN must handle traditional network challenges: interconnection of multiple networks and protocol stacks, scalability with respect to size and speed, and handling heterogeneous networks.

Isochronets seek to address these HSN design goals. In what follows we describe their operations (sec. 2) and architectures (sec. 3), control mechanisms (sec. 4), performance behavior (sec. 5), and conclusions (sec. 6).

2 Isochronets operations

2.1 Routing on trees

Consider the motion of a frame in a store-and-forward network. 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.

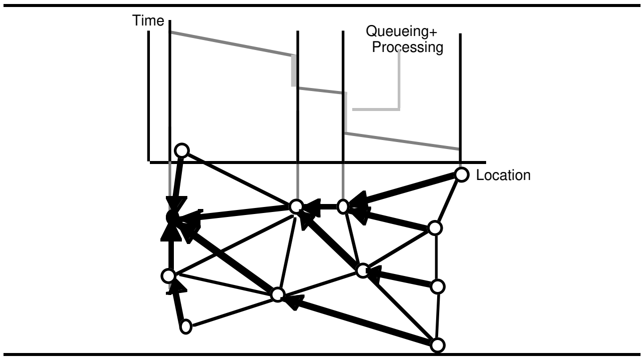

Figure 1: Internet routing on trees

Store-and-forward networks permit arrival randomness to propagate into network nodes. Network resources are efficiently utilized at the cost of QOS. 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 [1]). In the limit, where contention is eliminated with high probability, the very value of store-and-forward vs. a circuit-switched service becomes questionable.

Circuit-switched networks seek to provide absolute QOS by global resource reservations. Once a source establishes a circuit, traffic can move uninterrupted. Contention is eliminated 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.

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.

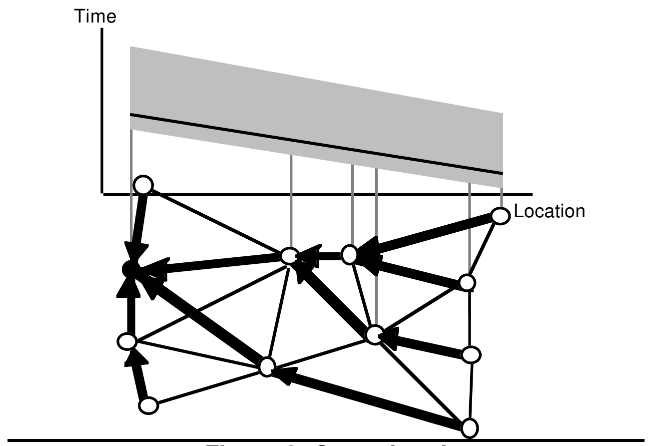

Figure 2: Green-band

In similarity to circuit-switched, 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)

Frames arriving simultaneously to a switching node contend for the outgoing tree link. The allocation of synchronized time bands to routing trees and resolution of frame collisions are the primitive constructs used by Isochronets to control traffic motions and QOS.

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.

Isochronets use priority bands and broadcast bands in addition to contention bands. 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 circuit-switched 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 a broadcast band, the routing tree is reversed and the root can broadcast to any subset of nodes.

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, much like TDMA. We call this technique, accordingly, Route Division Multiple Access (RDMA).

We designate the collision resolution mode used in terms of signs "-", "+", and "++". In RDMA- one of the colliding frames is discarded. In RDMA+, when collision occurs during a band, one is buffered and the other proceeds. RDMA++ stores frames beyond band termination, rescheduling them during the next band.

2.4 Circuit-switching, packet-switching 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 mode. That is, each source is allocated a circuit (priority band) to the tree root.

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 networks. The form of packet switching supported by Isochronets is advantageous to traditional packet switching 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 uncorrelated traffic).

Isochronets, it may be argued, could potentially underperform packet-switched 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 packet switched 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 [1] reduce the effective bandwidth available to any given source. A packet-switched 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 circuit or packet switched networks. In-between, Isochronets can be operated to span a spectrum of switching techniques of superior performance characteristics to both, as evidenced by the preliminary report of performance (sec. 5) and by more detailed studies to be reported [2].

Finally, this is useful point to also compare Isochronets with non-traditional high-speed switching techniques including WDM [3,4] and the Highball [5] proposal. 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 [5] 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.

2.5 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.

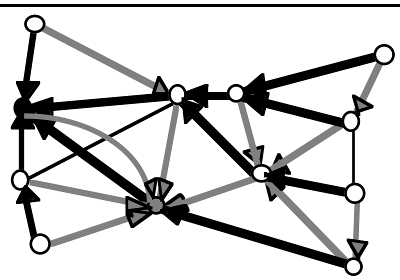

Figure 3: Multiple non-interfering trees

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.

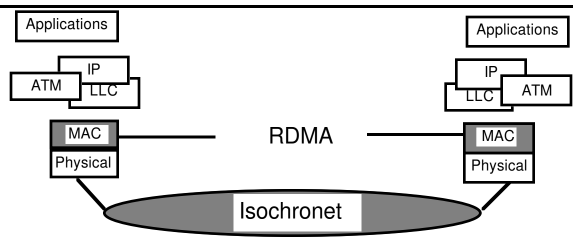

Figure 4: Multiple protocol stacks in Isochronets

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 Isochronets switch architecture

There could be many potential Isochronet switch implementations. In this section we review briefly the organization of one such classes of implementations.

The switch architecture is depicted in Figure 5. Trunk interfaces provide transmission/reception over trunks and serial/parallel conversion of communications. The switching fabric is as simple as a time-divided bus. 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.

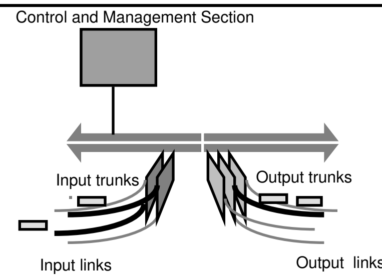

Figure 5: Electronic switch implementation

The switch control and management section (CMS) involves software running over a general purpose CPU. The primary function of the CMS is to configure the allocation and switching of bands using timers. Upon initiation of a band, the CMS configures the switching fabric and trunk cards to support appropriate communications from incoming tree trunks to the outgoing trunk. Trunk cards monitor the switching fabric and use a simple priority arbitration for access control. Arriving frames are transmitted via the switching fabric to the outgoing link.

During its priority band, an incoming trunk will gain pre-emptive access to the bus fabric. A pre-empted frame is retransmitted by the source trunk card when the priority transmission completes.

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 CMS logic. This separation permits Isochronets to scale favorably for a broad spectrum of trunk speeds without requiring changes of the network control mechanisms.

4 Network control

4.1 Clocks and band synchronization

Synchronization of bands and clock management are 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.

Consider the choices of U and P above over a 2.4Gb/s link. During a period of P=125µs, some 300kb 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. Additionally, since link speeds may vary greatly, Isochronets may wish to use different periodicity over links. For example, a link of 155Mb/s may use a period of 16P=2ms. Arrivals over this link will be buffered and delivered to higher speed links. Discussion of this general case, however, is beyond the scope of this introductory paper.

Band synchronization within Isochronets is a simpler problem than classical network clock synchronization [6]. Synchronization must only ascertain that the bands on incoming links must be strictly contained (when propagation delay is added) within the band time of the outgoing link (band constraints). The goal of a band synchronization protocol is to establish band initialization values that satisfy the band constraints for all links. The latency delay parameters in each link can be tuned to meet the band constraints by the switching node at which the link is incident.

Consider a routing-tree R, a switch A, an incoming link j from switch B and an outgoing link i. Suppose A wishes to set the band initiation time of link j (at B) so that it meets the band constraints. A and B use the primitive Protocol 1 (PSP) to set band initiation times on link j. PSP can be used to establish band initiation times throughout an entire routing tree. The root can use PSP to establish synchronized band initiation times at its sons. Each son can proceed to set band initialization time for its sons. The processing of PSPs can be accomplished in parallel with only a single sweep from the root to the leaves. Synchronization of initialization points can thus be accomplished very fast.

1. A->B: Request For Synchronization (RFS) message for tree R.
2. B->A: Synchronization Response (SR); B marks time T, at which SR is sent.
3. A marks arrival time of SR to link i. A measures offset O(i,j) from this marked time until desired band arrival to link i.
4. A->B: Initialize Band (IB) message for tree R with offset O(i,j).
5. B sets the band initialization for R on link j at the time B(j|R)=T+O(i,j).

Protocol 1: Primitive Synchronization Protocol (PSP) for band initialization times

4.2 Band allocation mechanisms

The goal of band allocation protocols is to establish appropriate band duration. A variety of mechanisms may be used to allocate green-bands to trees to optimize network performance. We provide a general discussion of a class of such mechanisms. The allocation must satisfy the band constraints of sec. 4.1 and, additionally, ensure the following overlap constraints: the intervals of different trees on the same link do not intersect. A cut-off value (δ) for band size needs to be set so that allocations under this size are reduced to 0.

One approach to the band allocation problem is to proceed and assign, initially, a maximal allocation of bands that meets the band and overlap constraints. This initial allocation could then be improved by shifting initialization times and band allocations. Protocol 2 provides a method for initial such allocation of bands. The protocol utilizes band-allocation-size (BAS) frames sent by a switch to its sons and provides a maximal band size accepted by the father.

It is easy to verify that the band allocations produced by this protocol satisfy the band and overlap constraints. The protocol requires only one sweep through a routing tree. In similarity to the Protocol 1, each routing tree can pursue band allocations independently and in parallel with other trees. The allocation of bands produced by this protocol may, however, be highly inefficient. In order to improve band allocations it is necessary to coordinate the choices of band allocations among interfering trees. A detailed study of such global coordinated band assignments is beyond the scope of this paper.

1. Tree R generates BAS frame with appropriate band size for each of its sons (typically, but not necessarily, the entire clock cycle) and sends it to them.
2. Upon receiving a BAS frame with a band size Δ(i|R) from its father on tree R down on link i, the i-switch sets the band size for the outgoing link i to D(i|R)=min{Δ(i|R), B(i|R')-B(i|R)} where R' is the successor of R on link i. If D(i|R)<d, then D(i|R) is set to 0.
3. The i-switch then computes for each incoming link j of the tree R a maximal band allocation size Δ(j|R) ≤D(i|R) (typically using equality) and transmits a BAS frame to the j-switch with the respective size.

Protocol 2: Initial band allocation

5 Preliminary Performance Evaluation

5.1 Analysis

In this section we consider a simple performance model of RDMA. A more comprehensive performance analysis is left to future publications. Consider RDMA serving ATM cell 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 study of RDMA-, we consider Poisson arrivals within a band. The band can be viewed 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 [7] to represent RDMA-. The process of interest is the arrival of cells whose transmissions are successful. With time measured in cell-transmission duration units, and 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:

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

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

$$LP_{RDMA-} = 1-1/(\lambda+1).$$

When the load is low, the loss rate is almost 0. It approaches 50% when the load reaches saturation (λ=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 (the same observation is valid for all queueing delays in this section).

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, [8] for a discussion on the subject). For the mean queueing delay, we approximate the solution by the same method used to compute the mean queueing time for M/G/1 systems with vacations [9]. 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) and the band size B (cells), the queueing delay of a contention band using RDMA++ may be approximated by:

$$W_{RDMA++} = \lambda/[2(1-\lambda)]+[V/2][V/(V+B)][1/(1-\lambda)].$$

The calculation of this formula is as follows. We compute the mean residual service time for the busy and vacation periods. For the busy period, the calculation is the same as for an M/D/1 system [9]. For the vacation period, the mean residual service time is the mean vacation period (V/2) times the probability of being on vacation (V/(V+B)). We then divide the mean residual service time by the idle period ($1-\lambda$), to obtain the mean waiting time [9]. 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 circuit switching (SCS) greatly improves on classical time-division circuit switching (CS). Indeed, suppose traffic to the band is divided among n sources using traditional circuit switching. Suppose, further, that traffic is uniformly generated by each source at a rate of l/n. The utilization of a given circuit remains $\rho= (\lambda/n)n=\lambda$. However, the vacation time increases and circuit bandwidth available decreases to result in:

$$W_{CS} = n\lambda/[2(1-\lambda)]+[(V+B(n-1)/n)/2]\times [(V+B(n-1)/n)/(V+B)][1/(1-\lambda)].$$

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 circuit switching, while providing sources so desiring the same performance guarantees as circuit switching does.

5.2 Simulation studies

In this section we provide a preliminary 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 (the destinations of these trees can thus share one band).

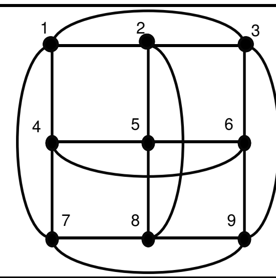

Figure 9: Simulated network topology

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 2.4Gb/s, resulting in 177ns transmission time per cell. The clock period is 125ms. 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 packet switching (PS), circuit switching (CS), and RDMA++.

The PS simulation uses the same trees allocated for RDMA++. Processing delays at nodes are included. The CPU at each switch is assumed to operate at the same rate of one input link. In other words, 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 each circuit until the circuit becomes available. Two RDMA++ experiments were conducted. In the first, all traffic has the same priority (RMDA++<c>). In the second, priority traffic is generated as follows. Each band is equally partitioned to priority sub-bands, one for each input node (RDMA++<p>).

Figure 10 depicts the mean packet delay (in μs) for the experiments we have conducted. The input traffic load is given as a percentage of the 2.4Gb/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 [10], and thus overlap in the figure. The "Pr." curve plots the mean delay for priority traffic generated for the RDMA++<p> experiment. Priority traffic was scheduled during its priority band (not incurring admission delays).

Figure 11 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 with mean 10 cells).

In Figure 12, we compare analysis and simulation results for RDMA++ mean packet delay. As it can be seen, the results are in good agreement. Figure 13 compares the simulation and analysis results for the mean packet loss rate in RDMA-.

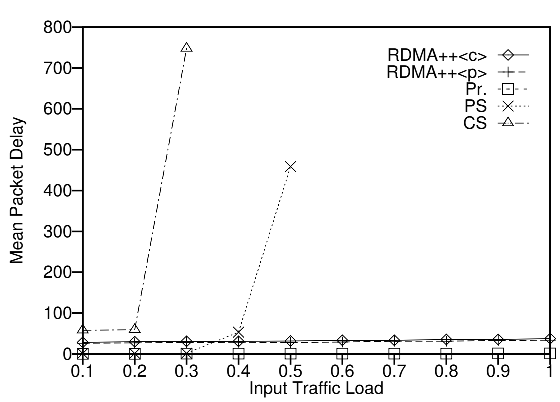

Figure 10: Mean network ATM cell delay for Poisson arrivals (in μs)

Finally, we display in Figure 14 the mean delay when we apply RDMA++ to the NSF T3 backbone network. We upgraded the link speeds[1] to 2.4Gb/s, and simulated a 100MIPs CPU at each node with 50 instructions 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.

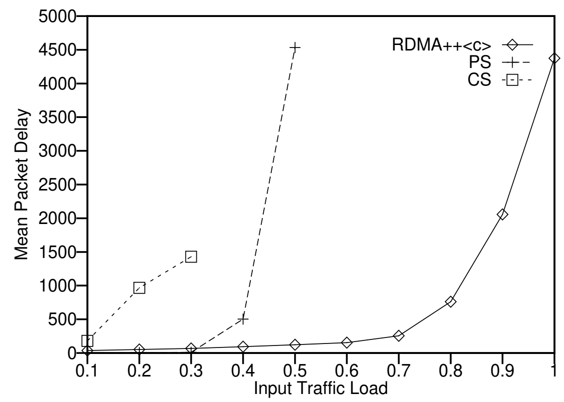

Figure 11: Mean network ATM cell delay for bursty arrivals (in μs)

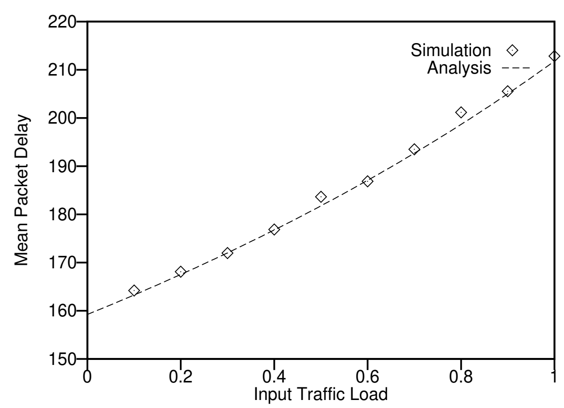

Figure 12: Simulation and analysis results for RDMA++ (time measured in cell transmission delay)

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 30ms (240 times larger). Thus, the immediate admission seen by frames in a packet switched implementation is a negligible component of the total frame delay.

[1] We actually could not run the simulation at 2.4Gb/s link speed due to the huge state space necessary. We ran the simulation at T3 link speed and scaled-up the results accordingly.

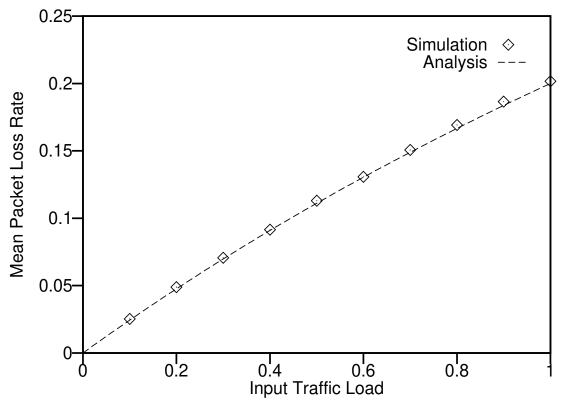

Figure 13: Simulation and analysis results for RDMA-

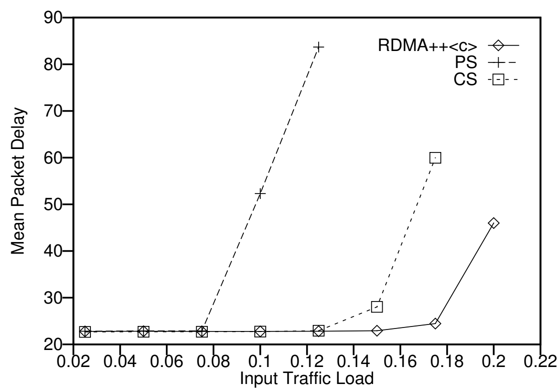

Figure 14: Mean network ATM cell delay for Poisson arrivals (in ms) in the NSF T3 backbone network

6 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 packet switching 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.

Acknowledgments

We would like to express our appreciation to Melik Isbara and to Prof. Hong Liu for many helpful suggestions on Isochronets and early drafts of this paper. This research was supported in part by NSF Project NCR-91-06127, by New York State CAT Grant 91053, and by Brazilian Research Council Grant 204544/89.0.

References

[1] Sidi, M., Liu, W., Cidon, I., and Gopal, I., “Congestion control through input rate regulation,” in Proc. GLOBECOM, IEEE, Dallas, Texas, USA, November 1989, pp. 1764-1768.

[2] Yemini, Y. and Florissi, D., “Isochronets performance characterization,” in preparation.

[3] Acampora, A.S. and Karol, M.J., “An overview of lightwave packet networks,” *IEEE Network Magazine*, vol. 3, 29-41, January 1989.

[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] 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, 1990.

[6] Mills, D.L., “Internet time synchronization: the network time protocol,” *IEEE Transactions on Communications*, vol. 39, no. 10, 1482-1493, August 1991.

[7] Karlin, S. and Taylor, H.M., *A First Course in Stochastic Processes*, Second Ed. Academic Press, 1975.

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

[9] Bertsekas, D. and Gallager, R., *Data Networks*, Second Ed. Prentice Hall, 1992.

[10] Kleinrock, L., *Queueing Systems*, vol. I. Wiley, 1975.