## Chapter 10

# Traffic Management in Isochronets Networks

Kelvin Lopes Dias[1], José Augusto Suruagy Monteiro[2] and Danilo Florissi[3]
*1 Universidade Federal de Pernambuco-UFPE,Centro de Informática*
*P.O. Box 7851, CEP: 50732-930, Recife,Pernambuco – Brazil*
*E-mail: kld@cin.ufpe.br*
*2 Universidade Salvador – UNIFACS, Engineering Departament*
*E-mail: suruagy@unifacs.br*
*3 Columbia University, Departament of Computer Science*
*Distributed Computing and Communications Lab*
*450 Computer Science Bldg, New York, NY, USA 10027*
*E-mail: df@cs.columbia.edu*

**Key words:** Quality of Service, Isochronets, RDMA, Leaky Bucket, Virtual Clock

**Abstract:** This paper presents an evaluation of a traffic management mechanism for high speed networks called RDMA (Route Division Multiple Access), developed as part of the Isochronets, a novel architecture for high speed networks. RDMA enables routes to designated destinations during time intervals (green-bands) available periodically. We evaluated the support RDMA offers to applications quality of service (QoS) represented by delay and jitter and its network congestion prevention effectiveness. RDMA is compared to two other well-known mechanisms: Virtual Clock and Leaky Bucket. RDMA showed itself to be satisfactory as an access control mechanism for applications with strict QoS requirements. In RDMA, memory resource needs at network elements is minimized; traffic sources or network peripheral elements are responsible for the use of a large quantity of memory.

## 1. INTRODUCTION

The multiservice structure of the broadband networks will carry a mix of traffic types ranging from traditional data communications to high quality video transmission, like: telemedicine, video on demand, and virtual reality.

J. N. de Souza et al. (eds.), *Managing QoS in Multimedia Networks and Services*

Fundamental for the success of this service integration is the concept of traffic management, which comprises a set of functions which aim at satisfying Quality of Service (QoS) requirements for applications, especially, during periods of network congestion.

Access control mechanisms, part of the traffic management framework, prevent high speed network congestion by monitoring and enforcing traffic contract established during call admission. Basically, these mechanisms can act in two ways: policing and shaping the traffic. In the first case, a policer can either drop packets violating contracts or mark them as low priority which may lead to discarding in congested nodes. In the second case, the mechanism can delay the packets at the sources to comply with the contract. Discarding can occur due to buffer scarcity [11]. The access control can be implemented at the traffic sources [8] and at network switches [10]. Besides the essential function performed by these mechanisms, they can degrade the QoS, for example, increasing end-to-end delay when acting as shaper, or discarding packets when acting as a policer.

In this paper, we analyse a new access control mechanism called RDMA (Route Division Multiple Access) proposed as part of a new architecture for high speed networks, Isochronets[3], from the view point of shaping. Our objective is to evaluate the support RDMA offers to application performance requirements and its network congestion prevention effectiveness. The results obtained for RDMA are compared to two other access control mechanisms: Leaky Bucket [12] and Virtual Clock [13]. This paper is structured as follows: section 2 describes Isochronets/RDMA, as well as, Leaky Bucket and Virtual Clock. In section 3 performance evaluation results of these mechanisms are presented. Finally, section 4 presents the conclusions and proposals for future works.

# 2. MECHANISMS

## 2.1 Isochronets and Route Division Multiple Access (RDMA)

Isochronets are a novel switching architecture that coordinates traffic motions in the network over time, similar to how green waves coordinate traffic motion in roads. In the latter scenario, coordination is accomplished through synchronized traffic lights to create routes for uninterrupted and collision-free motions from source to destinations. In Isochronets, coordination is accomplished by enabling a set of routes by means of periodic configuration of network switches. Switches set up routing trees, that is, routes from all nodes to a given destination. A frame is switched by following its path through an enabled tree.

*Figure 1* ilustrates the idea. 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.

Multiple simultaneous routing trees can schedule transmission 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 2* shows two non-interfering routing trees.

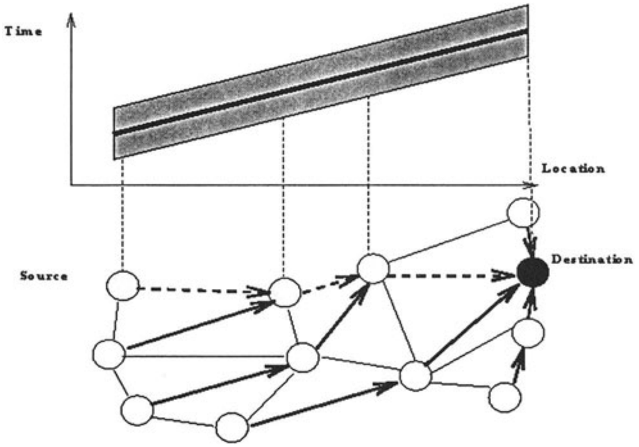

*Figure 1.* Motion Via Green Band.

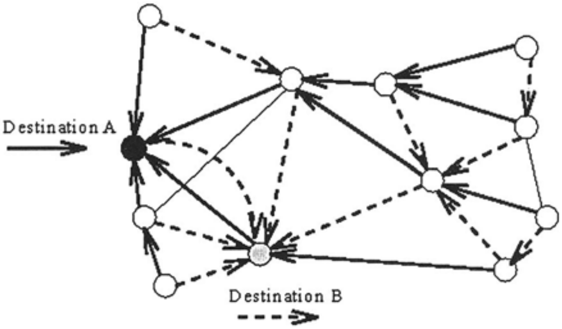

*Figure 2.* Multiple Non-interfering Trees.

The Isochronets coordination mechanism is called *Route Division Multiple Access* (RDMA) because it divides network bandwith among routing trees. RDMA is employed in Isochronets to assign bands (or bandwidth slices) to routing trees, that is, network switches are configured periodically to build different routing trees. Each configured tree lasts a time band (green band) after which the next time band and corresponding tree replaces it. A sequence of bands building trees that cover each destination once and only once form a *cycle*, which is repeated periodically.

The green band is maintened by switching nodes through timers synchronized to reflect latency along tree links. This problem of band synchronization is solved by manipulating the link propagation delays in the network. There is a device in each link that can be set to increase the link delay. The protocol compute the real link delay and then increases it using the device to make final delay a multiple of the clock cycle period. Because of this manipulation, if something is sent at time $t$ within the cycle from a given node, it reaches the next node still at time $t$ but many cycles later. This fact trivializes synchronization because all that is necessary after link delays are set is to start and end each band exactly at the same moment within each cycle. To solve the *clock synchronization problem*, any of the traditional protocols such as the Network Time Protocol [5] may be used.

*Figure 3* and *Figure 4* show an example of RDMA operation. The hypothetical clock with 12 units cycle (*Figure 3*), presents a contention band allocation (band that may be shared by all sources simultaneously) for a routing tree with only two sources. *Figure 4* depicts the band allocation at the intermediary node *C*. *C* divides the band among its descendents *A* and *B*. The band sizes allocated to *A* and *B* have different widths and only overlap partially. When contention occurs, the collision resolution mode used is designated in terms of the signs: –, +, and ++. In RDMA–, only one of the colliding frames proceeds, while the others are discarded. In RDMA+, one colliding frame proceeds while the others are buffered, but only up to the band duration. RDMA++ operates similarly to RDMA+, but also stores frames beyond band termination, rescheduling them during the next band.

There is also a possibility to use priority bands. As an example, the last quarter of clock cycle (*Figure 3*), from 9 to 12, could be used by a priority source (e.g., video source). Traffic from a priority-source is given the right of way, by switches on its path, during its priority band. This band will not overlap with the contention band for non-real time traffic, however, priority sources do not own their bands, that is, contention traffic may access a priority band and utilize it whenever the priority source does not.

RDMA dimensioning needs to define only the band size for the flow to be controlled. This choice takes into account the mean transmission rate required by the source. The band size may represent a percentage of RDMA basic cycle, defined as 125μs [3]. The flow band size will have to take into account the output link capacity, that is, the proportion of the cycle allocated will define the proportion of output link capacity allocated. For example, if a link has 150Mb/s capacity and a source requires a mean rate of 75Mb/s, the allocated band for this source may be 75/150 = 50% of the 125μs RDMA cycle, that is, the band size will be 62,5μs.

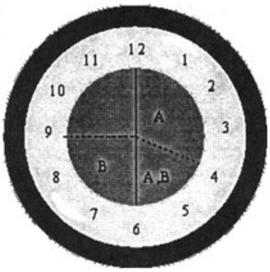

*Figure 3.* RDMA Allocation Example.

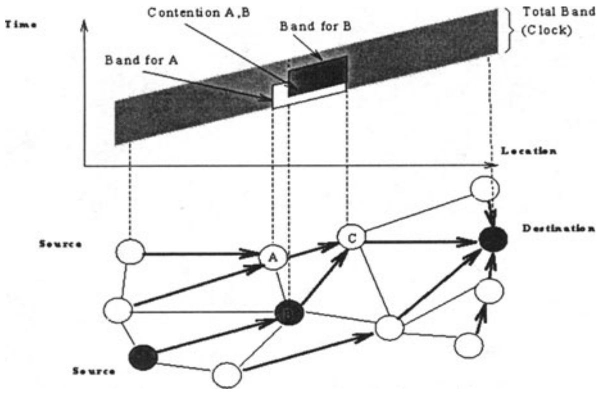

*Figure 4.* RDMA: Contention Band.

## 2.2 Leaky Bucket

This mechanism was proposed initially in [12] and it is used as the access control mechanism for ATM networks. Leaky Bucket consists of a token/credit generator, a pool of tokens/credits, and a buffer (this buffer can be optional). Credits are generated every $\lambda$ seconds and stored in the pool of maximum size $B$ credits. An arriving cell (packet with fixed size used in ATM networks) uses one credit from the pool and leaves the system immediately. An arriving cell that finds the credit pool empty joins the queue if the buffer is not full. When the queue is not empty (which means that the credit pool is empty) and a credit is generated, one cell takes the credit and leaves immediately.

## 2.3 Virtual Clock

Virtual Clock mechanism was designed as part of network architecture called Flow Network[13]. The resource requirements for each source are expressed in terms of two parameters: an average rate (AR) and an average interval (AI) to check the flow conformance. The network access is controlled by a mark called *virtualclock* given to the packets. Each data flow has a *virtualclock* which ticks at every packet that may access the network for that flow; the tick step is equal to the mean inter-packet gap (assuming a constant packet size). Network access is controlled in the following way: upon transmiting each AR x AI packets, if a flow sends packets according to its specified average rate, its *virtualclock* reading should be in the vicinity of the real time. Otherwise, the flow will be penalized by delaying or discarding its packets.

# 3. EXPERIMENTS

## 3.1 Delay and Jitter at Network Access for Real Time Traffic

The model used in this study is presented in *Figure 5*. An individual source generates ATM cells with data packed in 48 byte payloads. The link capacity to access the network is 150Mb/s. The following metrics were considered: access delay, defined as the difference between the time when the cell enters the network (source transmits cell) and the time when the cell was generated; jitter, defined as the difference between the network access delay of two consecutive cells; and, finally, the mean occupation in the infinite buffer [1] of *Figure 5*.

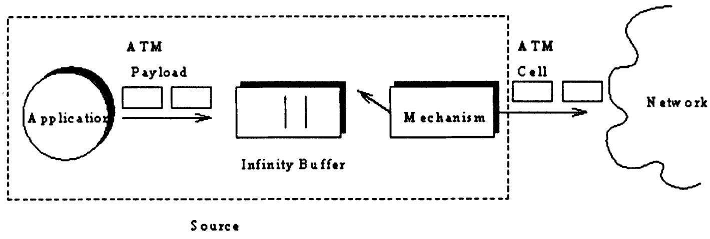

*Figure 5.* Source Model.

---

[1] The term "infinite buffer" means that the buffer space is sufficient to avoid losses due to the shaping effect caused by each one of the three simulated mechanisms.

### 3.1.1 Voice Source

For the voice source we utilized the ON-OFF model according to the conclusions in [2], with the following parameters: peak rate=167 cells/s, mean rate = 60 cells/s, burstiness(peak rate/mean rate) =2.78, and mean burst length = 60 cells. In RDMA dimensioning, the band size for this traffic was set to $1.9 \times 10^{-2}\mu s$ for a corresponding 125μs RDMA cycle. Leaky Bucket used a credit pool equal to 60 (mean burst length) and the credit rate corresponding to the voice mean transmission rate of 60 cells/s. For Virtual Clock the mean rate, 60 cells/s, was its rate parameter (AR) and the time between checks corresponded to the source average transmission time (ON state): 360ms or $\beta^{-1}$ [2]. The parameters have been set to provide a fair comparison of these mechanisms.

*Table 1.* Voice Source

| Mechanism | Mean Delay | Max Jitter | Mean Jitter | MeanOccupation |
|---|---|---|---|---|
| RDMA | 0.0001s | 0.000116 s | 0.000009 s | 0.5 cell |
| L. Bucket C=1.2 | 1.0058 s | 0.0138 s | 0.00523 s | 65.57 cells |
| L. Bucket C=1.4 | 0.2930 s | 0.0119 s | 0.00242 s | 22.29 cells |
| V. Clock | 0.2814 s | 0.6501 s | 0.0335 s | 27.22 cells |
| V. Clock C=1.2 | 0.1930 s | 0.6501 s | 0.0144 s | 19.03 cells |

As done in [9], an overdimensioning factor C > 1 is added to the Leaky Bucket mechanism. Under this scheme, the Leaky Bucket credit rate is equal to C times the negotiated mean traffic rate. We used the same idea for Virtual Clock rate parameter, AR. Simulation results are showed on *Table 1* and they are accurate within 6% with a 95% confidence level using batch means technique.

RDMA has the smallest delay while Leaky Bucket has the largest. RDMA delay is exclusively due to waiting time for the band to the destination. The maximal delay suffered by RDMA will be the inactive band duration (cycle duration minus band duration), queueing of previous cells waiting for the same band, and the cell transmission time. It is important to emphasize that only cells to the same destination are queued waiting for a band. Leaky Bucket delay is essentially caused by absence of tokens in the credit pool (a similar result was obtained in [7] when it was verified a serious delay due to the Leaky Bucket shaping process at an user end station).

RDMA can satisfy the requirement of end-to-end delay, typically varying from 0 to 150 ms [6]. It is important to say that both Leaky Bucket and Virtual Clock could have a smaller delay than RDMA. However, their parameters must be overdimensioned, which will cause congestion inside the network and significantly degrade the QoS of other connections. We have choosen to dimension all mechanisms based on the average rate promised by the source. Jitter and occupation for RDMA was smaller as well. For RDMA, the biggest jitter occurs when the second cell arrives right when the band closes and is forced to wait until the next active band to enter the network. RDMA fixes the cycle size at 125µs to satisfy the sampling rate of voice sources. In the voice experiments, the cell inter arrival time is 1/60 cells per second=16ms and the periodicity of active band is aproximately 125µs. RDMA services at least one voice samples per cycle.

### 3.1.2 Video Source

We assume that video traffic is produced by encoding subsequent video frames at a rate of 30 frames/s. The amount of information (number of bits) in a frame is generated according to a first order autoregressive model as proposed in [4], representing a videophone application, with mean rate equal to 3.9Mb/s (0.52 bits/pixel) and a peak rate equal to 10.68 Mb/s (1.41 bits/pixel). With this model, the number of bits/pixel for frame *n*, denoted by $\lambda(n)$, is given as follows: $\lambda(n) = a\lambda(n-1) + b\omega(\eta)$, where *a* and *b* are constants and $\omega(\eta)$ is a white Gaussian noise component with mean $\eta$ and variance 1. Here we use the following values: $a \approx 0.8781$; $b \approx 0.1108$ and $\eta \approx 0.572$. The number of pixel/frame is assumed to be 250,000 and the expected value $E(\lambda)$, of $\lambda(n)$ is 0.52 bits/pixel.

RDMA band size needs to accomodate a mean rate equal to 3.9Mbps/(8bits×48 bytes) = 10,156 cells/s. For a 150Mb/s link capacity, the band occupies 2.6% of the basic RDMA cycle, that is, 3.25µs from a 125µs cycle. Leaky Bucket had credit rate equal to 10,156 cells/s and the credit pool size configured to the mean and maximal burst size, that is, (0.52×250,000)/(8×48) = 339 and (1.41×250,000)/(8×48) = 918 cells, respectively. Virtual Clock used the video mean rate as the AR parameter and a intercheck time equal to 33ms (video frame periodicity).

*Table 2* shows the results obtained for this experiment, which are accurate within 8% for a 95% confidence level using batch means. The delay suffered by RDMA is only greater than the one by Leaky Bucket for *C=1.4* and the one by Virtual Clock for *C=1.2*. The maximal jitter introduced by the mechanisms is related to the parameter used to dimension them. For example, the maximal jitter for Leaky Bucket corresponds to the inter-credit arrival time, while for Virtual Clock it is due to the maximal penalty delay that a video cell may suffer. In RDMA, again, jitter value is due to the inactive band size.

Mean occupation was relatively high for all mechanisms, but mainly for Leaky Bucket with C = 1.15. This value was substantially bigger than the mean video frame (339 cells). Virtual Clock had the smallest occupation because we imposed a limit in the maximal time that a cell could be delayed, that is, 33 ms (video frame periodicity). RDMA had a satisfactory occupation compared to the others, even using only a conservative dimensioning, that is, sizing the band size for the promised video mean rate.

*Table 2.* Video Source

| Mechanism | Mean Delay | Max Jitter | Mean Jitter | MeanOccupation |
|---|---|---|---|---|
| RDMA | 0.00787 s | 0.0001217 s | 0.000041 s | 125.65 cells |
| L. Bucket C=1.15 | 0.16115 s | 0.000086 s | 0.000010 s | 1,282 cells |
| L. Bucket C=1.15 Credits=918 | 0.13198 s | 0.000086 s | 0.000008 s | 947.34 cells |
| L. Bucket C=1.4 | 0.0166 s | 0.00007 s | 0.000002 s | 210.79 cells |
| V. Clock | 0.00904 s | 0.03300 s | 0.000102 s | 88.22 cells |
| V. Clock C=1.2 | 0.00456 s | 0.03300 s | 0.000027 s | 45.19 cells |

## 3.2 Congestion Control and Delay for MMPP Traffic

The primary aim of this study is to understand RDMA congestion prevention effectiveness. The congestion degree is indicated *qualitatively* by means of number of cell losses and memory requirements for buffers present at the network internal switches. The accesss control mechanisms are used at peripheral nodes of the network topology proposed for the simulations. Besides that, we investigated the delay caused by the mechanisms at entrance of the network due to the shaping process.

The network topology used in this study is formed by local area networks and wide area networks, interconnected by a high-speed backbone. The corresponding access control mechanisms (RDMA, Leaky Bucket and Virtual Clock) are implemented inside the peripheral nodes. The resources, such as: bandwidth and available memory in switching elements are the same for simulations of all mechanisms.

In order to model the aggregated traffic generated by each one of the peripheral networks, we used a two-state Markov Modulated Poisson Process (MMPP) [1]. Each ON-OFF source that belongs to a MMPP has the following characteristics: peak rate = 10Mb/s, mean rate = 1Mb/s, burstiness= 10, mean burst length = 100 cells, and cell length = 424 bits. The MMPP is a reasonable model for this experiment because the sources generate periods of overload, even within the traffic contract.

The RDMA dimensioning uses a band size corresponding to the average rate specified by the aggregate traffic. The credit rate of Leaky Bucket is configurated with the mean rate of the aggregated traffic multiplied by the factor $C = 1.4$, and the credit pool size corresponds to the MMPP burst mean length. Virtual Clock uses the mean aggregate rate as its rate parameter and the period between checks are presented by graphs. We limit the maximum period that the flow should wait when penalized by Virtual Clock to 0.1 ms.

### 3.2.1 Results for Internal Memory Utilization

For this simulation we utilize 10 batches each one with $6.8 \times 10^7$ cells. In our experiments, the flow could travel in two ways along the backbone: one routing tree or two routing trees, according to Isochronets philosophy. In the first case, the sources $LAN_1$ and $LAN_2$, transmit data to the destination $LAN_4$. Both flows contend for the same output link at $C_3$ node, which leads to the $C_4$ node and, finally, to the corresponding peripheral node belonging to the backbone that do the direct connection with destination network (*Figure 6*). According to the Isochronets philosophy this corresponds to a routing tree. In the second configuration, the bands at $LAN_1$ and $LAN_2$ do not overlap in time and the destinations are $LAN_3$ and $LAN_4$, respectively. In this case we have two routing trees (*Figure 7*)

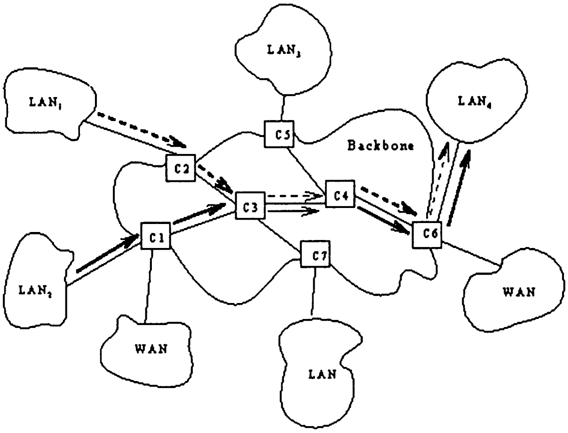

*Figure 6.* One Routing Tree.

The curves obtained with the experiments consider two load regions for the traffic accessing the backbone: an underload region with the traffic within the contract for the mean rate specified by source, and an overload region where the source generates traffic above the contract. The mechanisms are configured to a load of 50% of link capacity and such configuration remains unchanged for all loads used to plot the curves. We used a backbone with 150Mb/s internal link capacity.

It is important to say that in the simulated model, RDMA is also at internal nodes according to the Isochronets operation. The models for Leaky Bucket and Virtual Clock do not limit the bandwidth inside the backbone, because these two mechanisms are present only at backbone periphery. In this way, all capacity of output links at internal nodes is available to simulate Leaky Bucket and Virtual Clock.

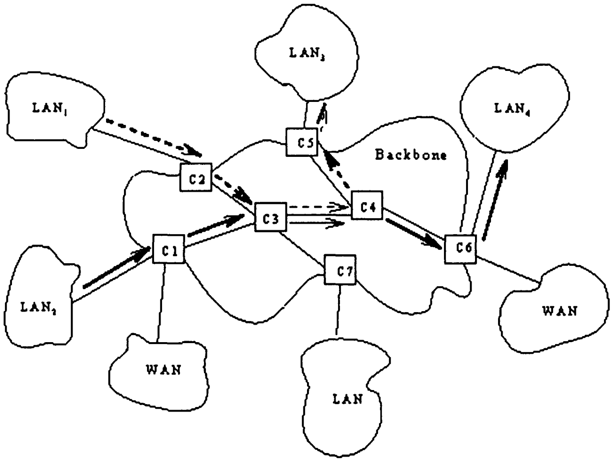

*Figure 7.* Two Routing Trees.

The internal switching elements are configured with a buffer of 20,000 cells. This value is sufficient to avoid losses at backbone peripheral nodes for loads bellow those used to configure the mechanisms parameters.

The *Figure 8* depicts the losses suffered by each mechanism for one of the peripheral nodes ($C_2$) that receive the flows. The losses occur after an offering load of 50% (used to configure the mechanisms). The average number of rejected cells is very high for RDMA and Leaky Bucket because this is an instability region, where the service rate of mechanisms is less than the arriving rate of flow cells.

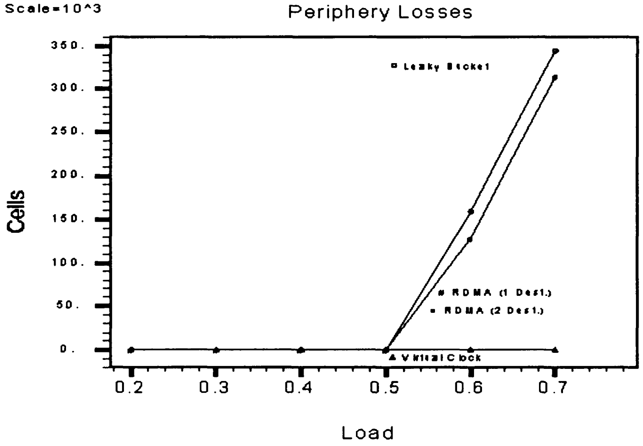

*Figure 8.* Loss at Backbone Periphery.

Virtual Clock is the only mechanism that, practically, does not suffer losses, even in the overload region. The possibility that a higher load (above traffic contract) enter the network could be dangerous to the other flows sharing the network when they contend at internal nodes. Therefore, Leaky Bucket and RDMA present a satisfactory result. However, Virtual Clock does not limit the extra traffic. This occurs because during periods where there is not flow check, all link capacity is available, and for virtual clock the system is stable.

The *Figure 9* depicts the results of average occupancy at backbone internal switch, $C_3$. This graph indicates the mechanisms effectiveness to avoid the network internal nodes be overloaded and, verifies if the mechanisms can offer a upper bound to that occupancy.

The lower average occupancy was obtained by RDMA when utilizing two routing trees, that is, two different destinations. For this configuration the active bands of RDMA at periphery are opened in intervals that do not overlap in time. Because of the opening of band at internal node is synchronized with the opening of the band at peripheral node, the average occupancy is nearly zero, even when the offered load is greater than that used to configure RDMA band size. The reason for this is that the excessive traffic is not allowed to access the backbone internal links, it is bounded at periphery. See *Figure 8*.

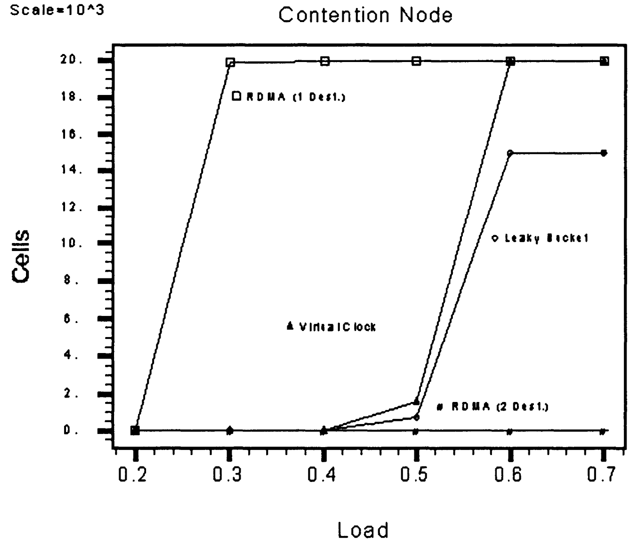

*Figure 9.* Contention Node Occupancy.

The average occupancy of Leaky Bucket is perceived only after the offered load becomes the same load used to its dimensioning. However, it does not grow indefinitely, being shaped as the load exceeds the contract.

Virtual Clock had a similar behavior to that of Leaky Bucket, but after a load of 50%, its mean number of discarded cells grows indefinitely as depicted in *Figure 10.*

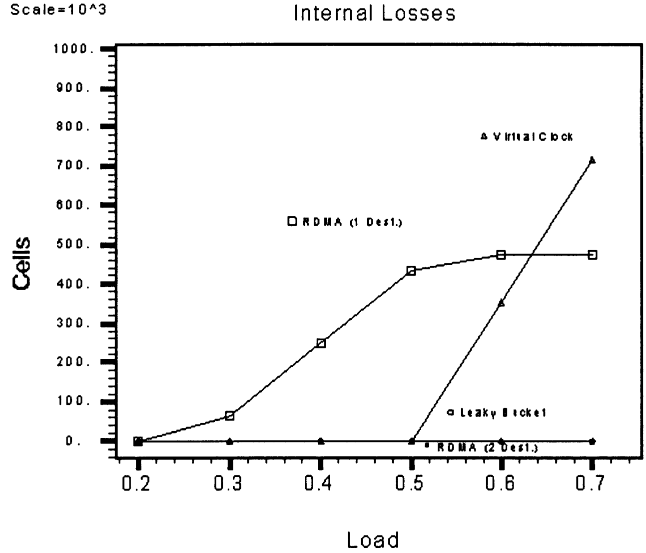

*Figure 10.* Contention Node Losses.

With only one routing tree (one destination for the two flows), the average occupancy of RDMA was very high, yet in the low load region. From 30%, the upper bound of internal node memory was reached and, consequently, losses occurred (See *Figure 10*). However, see that the internal RDMA node continues with only 50% of the total available cycle for this band, that is, half the link bandwidth, even now with only one routing tree. This implies that bands in peripheral nodes become active at the same time. This is an improbable situation in a RDMA routing design, that is, the sum of bands at peripheral nodes be greater than the bands at internal nodes. However, this situation is useful to show that, even using only one routing tree with overload, RDMA can limit the number of cells accessing the network as with Leaky Bucket. This can be verified in *Figure 10* where the losses for RDMA are smoothed from that load used to configure the mechanisms, that is, 50%.

### 3.2.2 Delay at Backbone Periphery

The experiment depicted in *Figure 11* represents the delay suffered by a MMPP flow in function of the load variation with a 95% confidence level. The MMPP is a reasonable model for this experiment because the sources have overload periods, even within the traffic contract. RDMA and the other mechanisms were dimensioned for a 50% load. For Virtual Clock we used two values for the maximal penalty time: 0.1ms (VC(0.0001s)) and 5ms (VC(0.005s)) and in both cases the time between checks (AI) was set to 0.1ms.

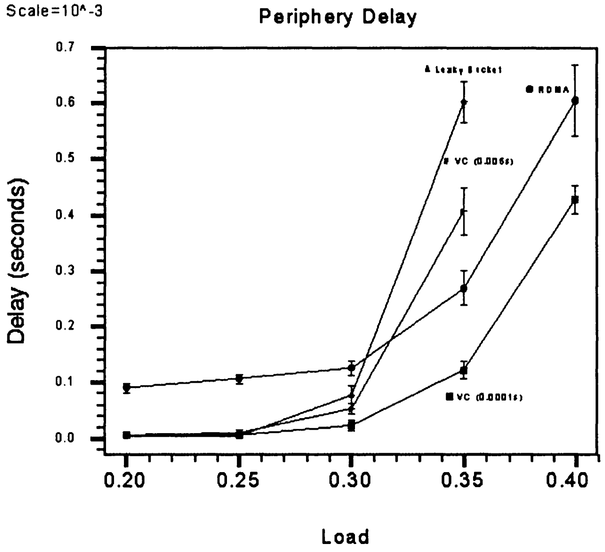

*Figure 11.* Delay for a MMPP Flow.

It is interesting to observe that the RDMA delay is bigger than the others for low loads, but, as the offered load grows, the delay for Leaky Bucket and Virtual Clock surpass RDMA (except for Virtual Clock with 5ms penalty time). This behavior occurs with Leaky Bucket due to the absence of credits, and for Virtual Clock due to the growing of wait queue during penalty time as load increases.

# 4. CONCLUSIONS AND FUTURE WORK

In this article, we have analysed a new access control mechanism for high speed networks, called RDMA (Route Division Multiple Access). We have studied the shaping effect on applications QoS requirements described in terms of delay and delay variation (jitter). We also studied the effectiveness of this new mechanism to avoid network congestion inside the network. RDMA showed itself to be satisfactory as access control mechanism for applications with strict QoS requirements, represented by delay and jitter. The traffic shaping effect, common to these mechanisms, does not cause excessive delays and jitter. RDMA seems to provide adequate performance even when it allocates resources according to the mean transmission rate.

This may present some advantages over other mechanisms like Leaky Bucket, which need to know the application burst size to effectively control the traffic.

Network internal occupancy can be controlled by RDMA. This aspect depends on the route configuration too. For flows that do not share the same routing tree internal buffer occupancy of the network can be strongly reduced. For routing trees with shared bands the utilization can be higher, but controlled. The basic difference between contention using RDMA and the other mechanisms (Leaky Bucket and Virtual Clock) is that for RDMA contention will occur only when bands share the same routing tree (destination) whereas for the others, contention occur whenever the same output link is shared. Another important aspect of RDMA is that, memories at network switching nodes may be less required than at periphery.

In sumary, RDMA can be characterized as an effective access control mechanism for real time applications. It also permits a fine tunning of congestion degree inside the network. Future reseach work includes the analysis of RDMA with more realistic traffic models, like fractal models. Another important topic is to compare the effectiveness of ATM mechanisms for admission control of new connections (CAC - Connection Admission Control) with the RDMA allocation of time bands.

## REFERENCES

[1] A. Baiocchi, N. Melazzi, A. Roveri and R. Winkler. Modeling issues on an ATM Multiplexer within a Bursty Traffic Environment. INFOCOM'91, Pages 83-91, 1991.

[2] P. T. Brady. A Model for Generating ON-OFF Patterns in Two-Way Conversations. Bell System Technology Journal, 48:2445-2472, 1969.

[3] D. Florissi. Isochronets: A High Speed Switching Network Architecture. PhD. Thesis. Computer Science Department, Columbia University, 1995.

[4] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson and J. D. Robbins. Performance Models of Statistical Multiplexing in Packet Video Communications. IEEE Trans. Comm.,36(7): 834-844. July 1988.

[5] D. L. Mills. Internet Time Synchronization: The Network Time Protocol. IEEE Transactions on Communications, 39(10):1486-1493, August 1991.

[6] R. O. Onvural. Asynchrounous Transfer Mode Networks. Artech House, Second Edition, 1995.

[7] B. V. Patel and C. C. Bisdikian. End-Station Performance under Leaky Bucket Traffic Shapping. IEEE Network. 10(5):40-47, Sept/Oct.1996.

[8] S. Radhakrishnan, S. V. Raghavan and A. K. Agrawala. A Flexible Traffic Shaper for High Speed Networks: Design and Comparative Study with Leaky Bucket. Comp. Net. and ISDN Syst. 28(4): 453-469, Feb. 1996.

[9] E. P. Rathgeb, Modeling and Performance Comparison of Policing Mechanisms for ATM Networks. IEEE JSAC 9(3):325-334. April 1991.

[10]J. Rexford, F. Bonomi, A. Greenberg and A. Wong. Scalable Architectures for Integrated Traffic Shaping and Link Scheduling in High-Speed ATM Switches. IEEE JSAC, 15(5): 938-950. June 1997.

[11]M. Sidi, W.-Z. Liu , I. Cidon and I. Gopal. Congestion Control Through Input Rate Regulation. IEEE Trans. Comm. 41(3): 471-477, Mar. 1993.

[12]J. S. Turner. New Directions in Communications (or Wich Way in the Information Age?). IEEE Comm. Mag. 24(10): 8-15. Oct 1986.

[13]L. Zhang. Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks In Proc. of ACM SIGCOMM'90. Pag. 19-29, Sep. 1990.