Journal of Mobile Communication 2 (2): 30-38, 2008
ISSN: 1990-794X

Congestion Control in ATM Network using Multi-Source Virtual Dynamic Routing

[1]A. Subramani and [2]A. Krishnan
[1]Department of M.C.A, K.S.R. College of Engineering,
[2]K.S. Rangasamy College of Technolgy, Tiruchengode-637 215,
Namakkal District, Tamilnadu, India

Abstract: The ATM forum has proposed the Private Network-Network Interface (PNNI) specification to facilitate routing in ATM networks, pre-path selection (or) routing algorithms that guarantee a user's multiple Quality of Service (QoS) requirements remain an open issue. In this study, we consider Multi-Source Virtual dynamic QoS Routing in an integrated-services environment where some of the offered traffic streams are dynamic. While there has been work on routing performance in the presence of purely stationary traffic, the presence of dynamic traffic has received very little attention. We consider here two routing schemes: Multi-Source Virutal Dynamic Routing (MSVDR), Hierarchical Based Least Loaded Routing with Periodic update (HBLLR). Based on simulation, we have found that while MSVDR performs and adapts better than Backward Hierarchal Routing Algorithm (BHRA) in the case of no trunk reservation, it is other way around when there is moderately high trunk reservation (even for less frequent routing updates). Further, we found that dynamics of a traffic class can result in dynamic flow blocking behavior even on stationary traffic, especially at a low or no trunk reservation level. This proposed Algorithm (MSVDR) reduce the Reservation Time, Minimize the Queue length, Fast route recovery, Maximum link utilization and Good throughput compare to the other Routing Algorithms.

Key words: PNNI, routing, queue management, QoS, LLR

INTRODUCTION

One of the architectural features in the ABR specification is the Virtual Source/Virtual Destination (VS/VD) option. This option allows a switch to divide an end-to-end ABR connection into separately controlled ABR segments by acting like a destination on one segment and like a source on the other. The coupling in the VS/VD switch between the two ABR control segments is implementation specific. The VS/VD option allows a switch to divide an ABR connection into separately controlled ABR Segments. On one segment, the switch behaves as a destination end system, i.e., it receives data and turns around Resource Management (RM) cells (Which carry rate feedback) to the source end system. On the other segment the switch behaves as a source end system, i.e., it control the transmission rate of every Virtual Circuit (VC) and schedules the sending of data and RM cells. Its call such switches a "VS/VD" switch. In effect, the end-to-end control is replaced by segment-by-segment control.

The segment-by-segment control is that the segments have shorter feedback loops, which can potentially improve performance because feedback is given faster to the sources whenever new traffic bursts are seen. The VS/VD option requires the implementation of per-VC queuing and scheduling at the switch.

The basic idea of our methods is that we maintain, for each node, a set of edges, called switches that are guaranteed to make substantial progress. Then every time the node chooses the "lightest" switches to relay a packet.

This way, we show that our algorithm has good performance in terms of both path length and maximum load. In addition, we show that the switches can be dynamically maintained by using only local information. Specifically, we can guarantee the following properties of our algorithm.

It uses only short paths: The number of hops of the path used is at most four times as many as the number of hops of the shortest path algorithm.

Corresponding Author: A. Subramani, Department of M.C.A, K.S. Rangasamy College of Engineering, Tiruchengode-637 215, Namakkal District, Tamilnadu, India

It balances the load: The maximum total size of packets passed on any node is at most 3 times as much as the optimum.

It is localized and scales well to large networks: Each node only needs information in its local neighborhood to make routing decision and as a consequence, our algorithm handles dynamic change and mobility efficiently as only a node's neighborhood is affected.

It is online: The routing decision of a packet depends only on the previously routed packets, i.e., the current state of the network. It doesn't need to know the packets in the future.

In addition to providing rigorous analysis, we have also implemented the algorithm and studied the performance by simulation. The good performance of our algorithm is supported by the simulation results as well. For example, even for random traffic pattern, under which we would expect the shortest path routing works well, the maximum load created by our algorithm is only about 20% of the shortest path routing. We also compare the number of hops in the path produced by our algorithm to that in the shortest path and show that the path length is only increased by a small fraction.

Our research for the nodes in a narrow strip is closely related to the on-line load-balancing problems on related machine model. Azar's (1998) paper and Borodin and El-Yaniv's (1998) book contain excellent survey of this subject. Load-balancing routing in general can be formulated as the unsplittable flow problem where we aim to minimize the maximum node congestion. This is a well-known Nphard problem that can be approximated to a factor of O(log n/loglog n) (Raghavan and Thomsen, 1985; Raghavan, 1988). In all the previous work, one either ignores the length of the path or uses only shortest paths for regular networks such as meshes. Another related problem is the on-line virtual circuit routing problem, which has also been studied extensively (Serge, 1995; Azar, 1998; Borodin and EI-Yaniv, 1998).

On the other hand, energy-aware routing algorithms, which try to maximize the network survivability, have attracted a lot of interest (Bambos, 1998; Xu *et al*., 2001). The energy aware metrics, such as "maximize time to partition" and "minimize maximum node cost", were first proposed by Singh *et al.* (1998). Chang and Leandros (2000a, b), used a flow augmentation algorithm and a flow redirection algorithm to balance the energy consumption on different nodes. Their method, however, requires a full knowledge of traffic demands and does not handle node insertion and deletion. Extensions along this approach were addressed in Vikram *et al.* (2002) and Zussman and Anderion (2003). Li *et al.* (2000) studied the online power-aware routing which minimizes the earliest time when a packet can not be sent. They proved that any online algorithm has unbounded competitive ratio and provided algorithms with zone-based heuristics. Yan *et al.* (2001) proposed a method that uses the geographical locations of wireless nodes for energy aware routing. Xu *et al.* (2001) proposed an algorithm GAF which is designed to reduce the energy consumption by turning off unnecessary nodes. In Wei and Jangwon (2002) and Nishant and Samir (2002), the traditional energy-unaware routing protocols such as DSR or AODV were re-visited to take into account the energy-aware metric. All of the energy-aware protocols mentioned above are heuristics and do not provide any guarantee on the performance.

Lee (1995) proposed a call-by-call source routing strategy that makes use of rule based fallbacks. This strategy provides a flexible platform on which routing can be done efficiently subject to performance, resource and priority constraints. The fallback routing algorithm sequentially computes paths based on a predetermined fallback sequence of routing instances, until an acceptable one is available or the call will be blocked. The proposed routing architecture uses hierarchical source routing with optional crankbacks. A variety of traffic-dependent QoS-related topology state parameters are advertised to support call-level QoS matching. Topology information at each hierarchical level is aggregated to trade-off fine-grain QoS matching for scalability in very large networks. A large portion of the routing architecture designed in this study has been integrated into the PNNI specification.

VIRTUAL SOURCE/VIRTUAL DESTINATION SWTICH

The VS/VD switch implements the source and the destination end system functionality in addition to the normal switch functionality. Therefore, like any source and destination end system, it requires per VC queues to control the rates of individual VCs. The switch queue structure is now more similar to the source destination structure where we have per VC queues feeding into the per-class queues before each link. This switch queue structure and a unidirectional VC operating on it is shown in Fig. 1.

The VS/VD switch has two parts. The part known as the Virtual Destination (VD) forwards the data cells from

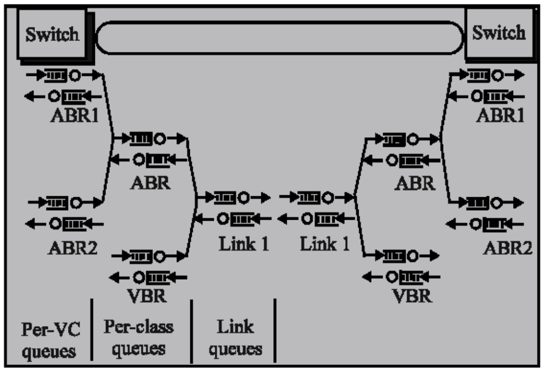

Fig. 1: Per-VC, per-class and per-link queues in a VS/VD switch

the first segment ("previous loop") to the per VC queue at the Virtual Source (VS) of the second segment ("next loop"). The other part or the Virtual Source (of the second segment) sends out the data cells and generates FRM cells as specified in the source end system rules. The switch also needs to implement the switch congestion control algorithm and calculate the allocation for VCs depending upon its bottleneck rate.

Rate calculation in a VS/VD Switch: Figure 2 shows the rate calculations in a VS/VD switch. Specifically, the segment starting at Link2 ("next loop") returns an ER value, ER_2 in the BRM and the FRM of the first segment ("previous loop") is turned around with an ER value of ER_1. The MSVDR algorithm for the port to Link2 calculates a rate (VAL_2) as:

VAL_2 = Function { Input Rate, VC's Current Rate}. The rate calculations at the VS and VD are as follows:

- Destination algorithm for the previous loop:

$$ER_1 = \text{Min}\ \{ER_1, Val_2, ACR_2\}$$

- Source Algorithm for the next loop: Optionally,

$$ER_2 = \text{Min}\ \{ER_2, VAL_2\} ACR_2 = \text{FN}\ \{ER_2, ACR_2\}$$

PROPOSED MODEL

Changes to PNNI packet formats: The level of congestion experienced by a node (high, low, or no congestion) shall be flooded to all peer nodes and be sent to its neighbor nodes. For the purposes of flooding the indication throughout the peer group, a Congestion State Indication (CSI) is introduced in the nodal information group type of PTSE, as defined in this Section. For the

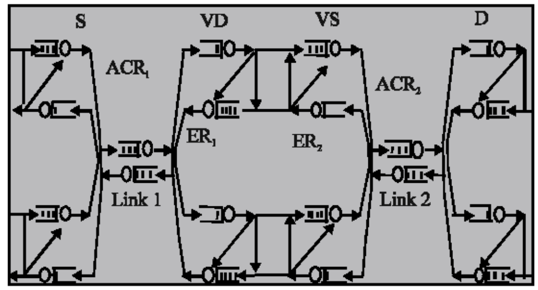

Fig. 2: Rate Calculation in VS/VD Switches

purpose of indicating the congestion to a neighbor node, a CSI is introduced in the PNNI packet header of the PTSP, PTSE Acknowledgement packet and Hello packet, as defined in this study. The modified PNNI Nodal flags and PNNI packet header flag are shown in the Table 1 and 2.

Motivations:

- Reduced Queue Waiting Time in Switches.
- Avoid Retransmission of the Same packets.
- Reduced the delay time.
- Using Multicast Methodology.
- Fast Recovery.

Proposed model features:

- Virtual Source/Destination Resource Management. Cell (RM Cell).
- Doubly Finite Queue Management.
- Share memory Concepts.
- Cell priority based Scheduling.
- Hierarchical based least loaded routing Algorithms.
- PNNI protocol Used.
- Geographical Extension.
- Cell loss Loss Ratio Reduced.
- Cell Life Management for Redirect.
- Efficient Bandwidth Allocation.
- Cell Rate based Explicit Congestion Notifications.

Phase 2 - on-demand path selection: When a connection setup request arrives, it is the responsibility of the source node to choose a path that is most likely able to support the required QoS. The source node makes a routing decision based on its local knowledge of the network topology. As a result of the PNNI configuration, the knowledge base of a node contains full information about its own peer group, aggregated information about its parent group, more aggregated information about its grandparent group and so forth. Therefore, source routing

Table 1: Nodal flags

Bit ID:	Bit 8 (MSB)	Bit 7	Bit 6	Bit 5	Bit 4	Bits 3,2	Bit 1
Bit Name:	"I am Leader" bit	"Restricted Transit" bit	"Nodal Representation" bit	"Restricted Branching" bit	Non-transit for PGL election	Congestion State Indications	Reserverd
Description:	Value 0: "I am not PGL"	Value 0: "I am a transit node"	Value 0: "Simple node representation" branch points"	Value 0: "Can support additional	Value 0: "Normal Operation"	Value 00: "Normal flow no Congestion"	
	Value 1: "I am PGL"	Value 1: "I am a restricted transit node"	Value 1: "Complex node representation" points"	Value 1: "Cannot support additional branch this node for PGL election"	Value 1: " No connectivity through priority cell add primary Queue, Low priority cell in secondary queue" Value 11: "High Congestion Call Routing Algorithm" Value 10: "Low priory cell discard from secondary queue"	Value 01: "Low Congestion add high	

Table 2: PNNI packet flags

Bit ID:	Bit 8 (MSB)	Bit 7…1, 6 (LSB)	Bit 5..1 (LSB)
Bit Name:	"Resynch Congestion Indication" (RCI) bit	Congestion State Indication	Reserved
Description:	When the packet is type PTSP or PTSE Acknowledgement, set to zero when congestion is detected within the node and one when there is no congestion. Set to zero for other packet types.	When the packet is type PTSP, PTSE Acknowledgement, or Hello set to 00 normal flow indicate no congestion, 01 to indicate cell priority queue selection, 11 to indicate high congestion to select the alternate path with using MSVDR, 10 to indicate cell discard from secondary queue	

to a destination node outside the peer group of the source node is actually to find a path up the hierarchy to the level that source and destination nodes are in the same logical group.

Assume that a connection from the congestion node S to the destination node D is to be setup. The parent logical node of the congested node S and the destination node D at level i in the hierarchy is denoted as Si and Di. Let us consider the case where two QoS parameters need to be satisfied: bandwidth BW and segment-to-segment delay DY, which stand for the attribute and metric parameter types respectively. The Multi-Source Dynamic Routing (MSVDR) decisions can be made in the following steps:

Step 1: Calculate Shortest Path from congested Switch to Destination Node using Hierarchical Least Loaded Routing Algorithm.

Step 2: Send reroute information to destination with new source address, routing path and starting cell number.

Step 3: Send message to original source to select the second shortest path using OSPF Algorithm with out cross the congested link.

Step 4: Original source send message to destination with new route path and starting cell number.

Step 5: Destination rearranges the cells after receiving the cells from multi source.

Step 6: Destination send to the acknowledgement signal to respective sources.

Step 7: Read the Cell form Queue with using priority scheduling algorithm.

Step 8: Normal Data Flow.

Rerouting method: The rerouting mechanism is to provide fast recovery in the cases of link failures or topology changes that affect an ongoing session. Since ATM is a connection oriented technology, a new connection should be setup between the source and the destination if the current connection is broken. Link or node failures may occur in different places:

At level i, at which a logical group PGi(x) contains the ancestors of both the source and destination nodes. Rerouting is started from the very beginning in this case.

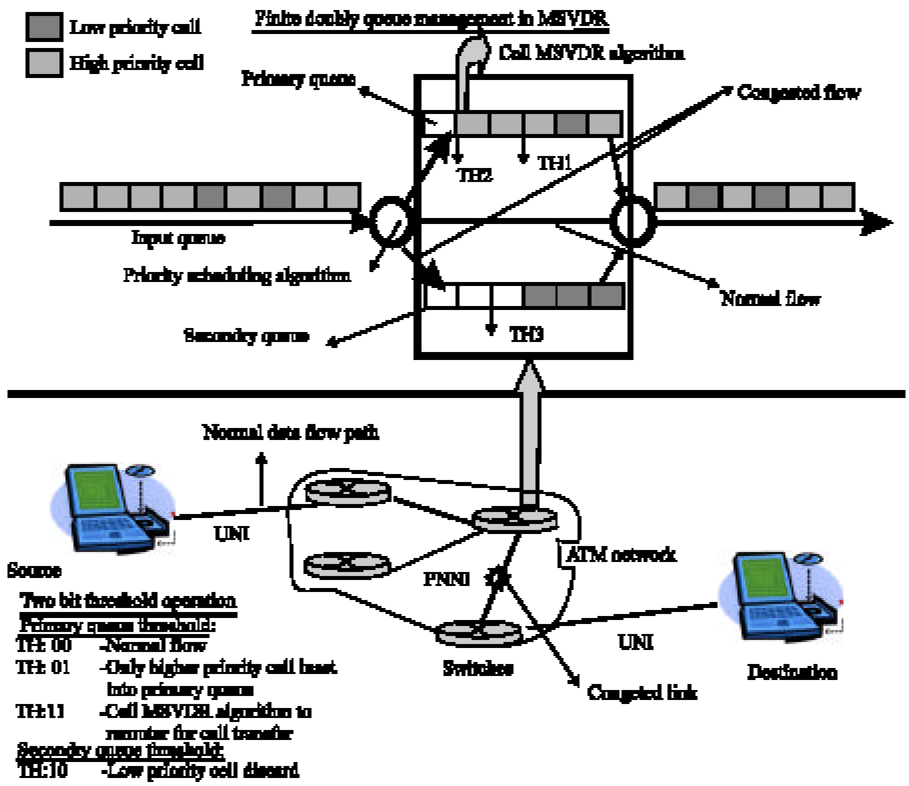

Fig. 3: Doubly finite queue management

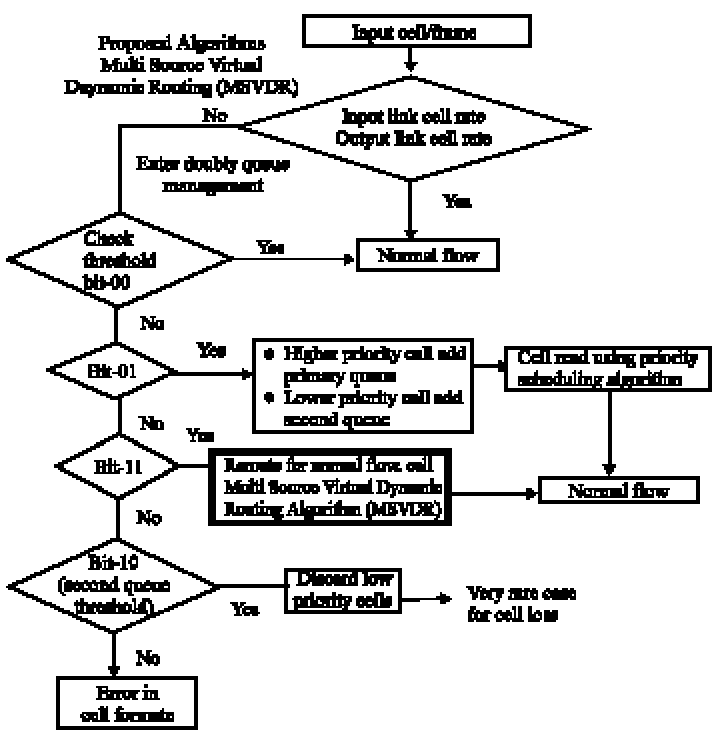

Fig. 4 : Flow diagram for doubly finite queue management

In the peer group which contains the source node at level 1. At the levels between level 1 and level i. wherever the node or link failure occurs, doubly queue finite queue maintain at the extreme state the rerouting starts from that particular level. The dead node or link is pruned from the set of pre-calculated paths. After that, phase 2 on-demand routing is performed again from the level at which this failure occurred down to level 1. The doubly finite queue management is shown in the Fig. 3. and also explain the maintenance of the finite queue flow diagram is given in the Fig. 4. Sampling data and simulation results are discussed in the next session.

HIERARCHICAL ROUTING

In hierarchical routing, a hierarchical topology is created by clustering the nodes into groups that form a logical node. These logical nodes are clustered to new groups to form a higher-level logical node and so on. Fig. 5 shows the actual physical network. The nodes, represented by the black dots, are clustered into groups of three or four as indicated by the red circles. Figure 6 shows the resulting aggregated topology, along with the second aggregation, in which the groups are clustered into new groups, in this case consisting of 3 first level groups, indicated by the blue boxes. Hierarchical routing proposed for QoS routing is somewhat similar to source routing, but each node has detailed state information only about the nodes in the same group and aggregated information about nodes in the other groups. Figure 7 shows the topology as seen by the node A2:1. Source node computes a path using the aggregated topology. When a node starts to transmit, first a control message is sent. When it reaches a border node which is a part of a group presented as a logical node in the path, it uses source routing to expand the path through the group. In the proposed hierarchical QoS routing schemes, source routing algorithms are used at each hierarchical level.

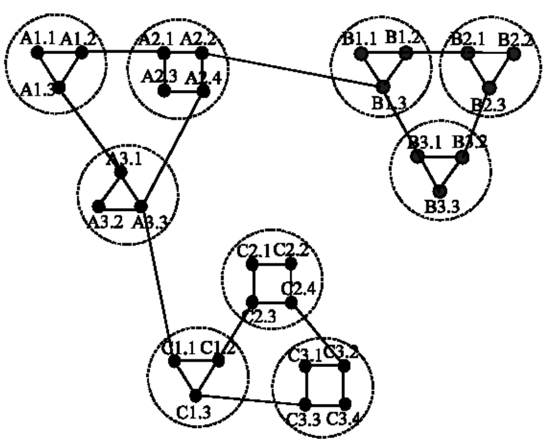

Fig. 5: Clustering in hierarchical topology

The size of the aggregated state information is logarithmic in the size of complete global information. Hierarchical routing solves the scalability issues of source routing and is regarded as the most promising scalable QoS routing approach. Negative effects of the aggregation include additional imprecision. For instance, there might be multiple paths through a logical node, one of which has lots of available bandwidth but larger hop count or longer delay, while another may have shorter delay, but does not have similar bandwidth. Requests

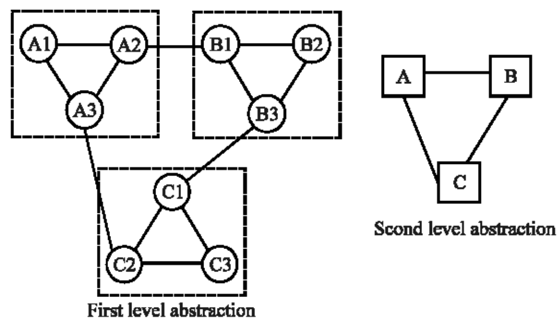

Fig. 6: The first and second level aggregation of the topology

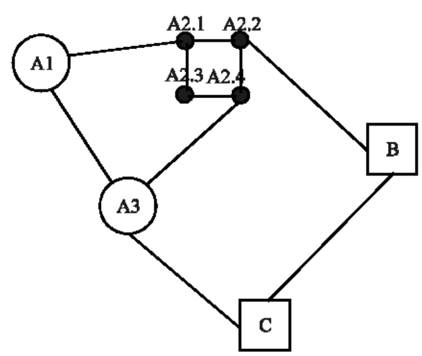

Fig. 7: The topology as seen by the node A2:1

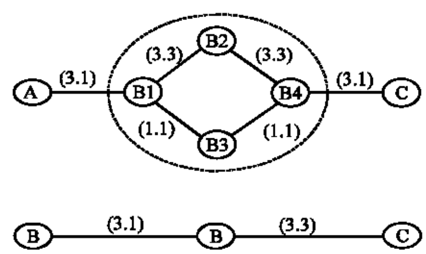

Fig. 8: Difficulties in aggregating link information. The link state informationin the figure is in the form (bandwidth, delay)

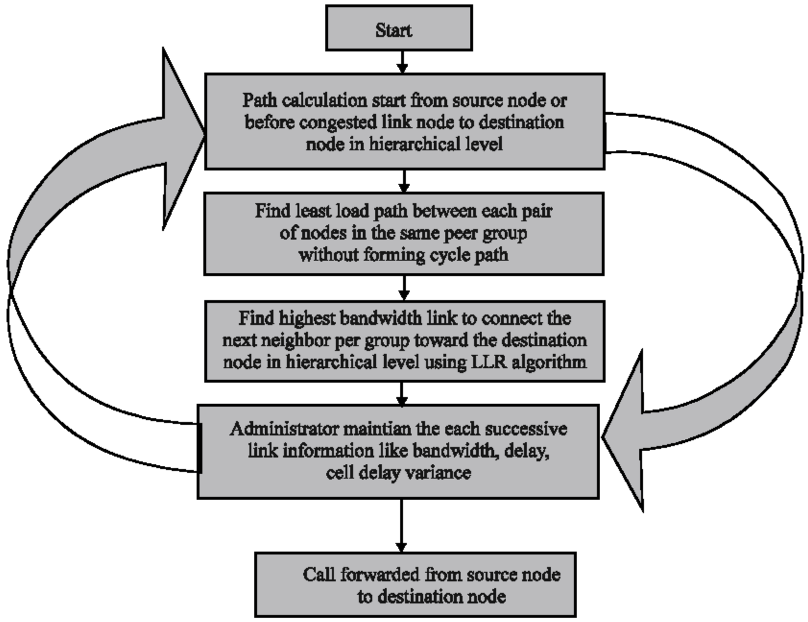

Fig. 9: HBLLR algorithm flow chart

with either requirements for a small delay or large bandwidth can be routed through the logical node, but both requirements cannot be satisfied at the same time. Figure 8 shows an example of this kind of a situation. What should node A useas the values of delay and bandwidth on link BC? The information in the figure includes maximum bandwidth and minimum delay, but as mentioned they cannot be attained at the same time. The maximum bandwidth available from B to C is 3 and the minimum delay is 1+1+1 = 3. But these are through different paths inside logical node B. Along the path through B2 the values of bandwidth and delays are and through B3 they are (1; 3). Which one should be used? If a connection from node A to node C has both bandwidth and delay constraints, the aggregated values in the figure are misleading. A typical example of hierarchical routing protocol is Private Network-Network Interface, PNNI, that has been used for ATM networks.

Algorithm for Hierarchical Based Least Loaded Routing (HBLLR)

HBLLR algorithm: HBLLR algorithm explain in the following steps and flow diagram is shown in the Fig. 9.

Step 1: We first try to setup the connection along the direct link. If Link capacity Call bandwidth, accept the call in the direct link; otherwise go to step 2.

Step 2: Select the 2-link alternative path Sk which has the largest free bandwidth.

Step 3: If there are more than one such path, pick one randomly, remaining path reserved for congestion cases.

Step 4: If there is no path is available Cell maintain in doubly finite queues.

Step 5: If the candidate set Sk is empty and queue has dramatically increase for long time congestion then only block the call.

RESULTS AND DISCUSSION

Table 3 described the logical group nodes queue input and output buffer sizes for maintain the doubly finite queue in the every ATM switches. Same logical group nodes have the same environments. Whenever, the primary queue size is full then only immediately call the MSVDR algorithm other wise the input data store in the primary queue for congestion time.

Table 4 Describe the traffic and QoS parameters for various traffic load for non real time. This parameters are modify at that time of data transfer from source to destination and also congested switch to destination whenever the link is congested.

Minimum rout recovery time: Figure 10 describe the alternate routing recovery time is minimum compare to other routing algorithms. The queue length is increate at that time of congestion occur, the doubly finite queue management maintain the queue, very critical condition

Table 3: Characteristics of logical group nodes

Logical group node	A	B	C	D
Output queue size	1000	750	500	750
Queuing delay (ms)	0.607	0.556	0.420	0.813

Table 4: Traffic characteristics in the simulation

Traffic	Call duration	PBR (Mbps)	CTD (s)	CDV (s)	CLR
Data	0.1-0.2 s	2-10	-	-	$1.0e^{-4}$
VBR	.05-2 s	5-20	$5.0e^{-3}$	$2.0e^{-2}$	$1.0e^{-5}$

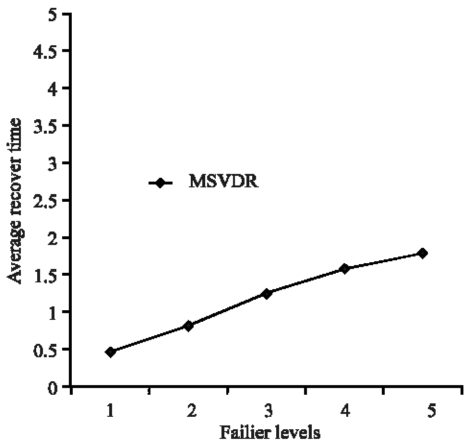

Fig. 10 : Comparison of Routing recovery time

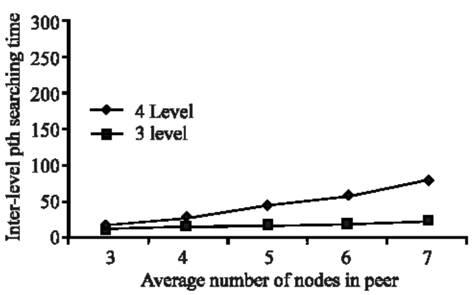

Fig. 11 : Path calculation of next Neighbor hop searching

they find out the alternate rout for data transfer. The numbers of node increase the level the rout recovery time in automatically increases.

Minimum time for next hop search: Figure 11 describe the path selection on that time of on demand, i.e., congestion occur, the path selection time also reduce for level-to-level hierarchical model. The low level model the path selection time is most probably same for all groups.

Minimizing queue delay time: Figure 12 describe the delay time for finite queue is minimize, the loss of the data is automatically reduced, then avoid the same cell retransmission.

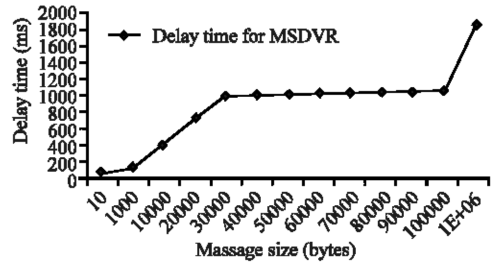

Fig. 12: Delay time for Queue in MSVDR

CONCLUTION

With the emergence of integrated services networks, QoS based dynamic routing schemes are very essential for maintaining better Quality of Service (QoS) in the network. However, the performance characteristics of these schemes are not known, especially in the presence of dynamic traffic loads. In this research, we have attempted to address this area in a multi-source environment. Using the NETSLIM simulator, we have studied the effect of various dynamic routing schemes on the flow blocking probabilities of different service classes. Some of the main observations of this work are summarized here. Due to dynamic routing, the dynamic traffic load on one node pair can affect the flow blocking probabilities of not only that node pair but also other node pairs in higher load situation in the case of no or low trunk reservation. The performance of Multi-Source Virtual Dynamic Routing (MSVDR) scheme is consistently better compared to other schemes. Additional studies are ongoing to understand possibly other implications of network traffic dynamics on dynamic routing schemes and the role network control plays.

REFERENCES

Bambos, N., 1998. Toward power-sensitive network architectures in wireless communications: Concepts, issues and design aspects. IEEE. Personal Commun., 5: 50-59.

Borodin, A. and R. El-Yaniv, 1998. Online Computation and Competitive Analysis. Cambrdige University Press.

Chang, J.H. and L. Tassiulas, 2000. Energy conserving routing in wireless ad-hoc networks. In Proc. Infocom, pp: 22-31.

Chang, J.H. and L. Tassiulas, 2000. Fast approximation algorithms for maximum lifetime routing in wireless ad-hoc networks. In Networking, LNCS 1815, pp: 702-713.

Lee, W. *et al.*, 1995. Routing Subject to Quality of Service Constraints in Integrated Communication Networks. IEEE. Network Mag., 9: 46-55.

Nishant Gupta and Samir Ranjan Das, 2002. Energy-aware ondemand routing for mobile ad hoc networks. In Proc. IWDC, pp: 164-173.

Qun Li, Javed Aslam and Daniela Rus, 2001. Online power-aware routing in wireless ad-hoc networks. In Proc. ACM. Mobicom, pp: 97-107.

Raghavan, P. and C.D. Thompson, 1985. Provably good routing ingraphs: Regular arrays. In Proc. 17th Ann. ACM Symp. Theory Comput., pp: 79-87.

Raghavan, P., 1988. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. J. Comp. Sys. Sci., pp: 130-143.

Serge A. Plotkin, 1995. Competitive routing of virtual circuits inATM networks. IEEE J. Selected Areas in Commun., 13: 1128-1136.

Suresh Singh, Mike Woo and C.S. Raghavendra, 1998. Poweraware routing in mobile ad hoc networks. In Proc. ACM/IEEE MobiCom, pp: 181-190.

Wei Yu and Jangwon Lee, 2002. DSR-based energy-aware routing protocols in ad hoc networks. In Proceeging of the 2002 International Conference on Wireless Networks.

Yu, Y., R. Govindan and D. Estrin, 2001. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor net-works. Technical report tr-01-0023, University of California, Los Angeles, Department of Computer Science.

Ya Xu, John S. Heidemann and Deborah Estrin, 2001. Geography informed energy conservation for ad hoc routing. In Proc. ACM Mobicom, pp: 70-84.

Yossi Azar, 1998. On-line load balancing, in On-line Algorithms: The State of the Art, A. Fiat and G. Woeginger (Eds.). LNCS 1442, Springer, pp: 178-195.

Zussman, G. and A. Segall, 2003. Energy efficient routing in ad hoc disaster recovery networks. In IEEE Infocom.