UNIVERSITY AT STONY BROOK

CEAS Technical Report 798

Dynamic Decentralized Routing Algorithms

Seong-Hwan Kim and Thomas G. Robertazzi

Feb. 19, 2002

Dynamic Rerouting Algorithms for Packet Switched Networks

Seong-Hwan Kim [1] ,and Thomas G. Robertazzi [2]

Department of Electrical and Computer Engineering,
University at Stony Brook,
Stony Brook, NY 11794

Abstract

Several dynamic rerouting algorithms are presented to recover from network congestion. Dynamic spatial rerouting algorithms are different from the connection rerouting algorithms which have been presented to date. Here, several decentralized rerouting algorithms are presented which reroute traffic when congestion occurs in a node, instead of blocking the traffic immediately or lowering source rates. In addition, the methods of reducing control signals are presented. The presented dynamic rerouting algorithms are single threshold detection rerouting, double threshold detection rerouting, group controlled rerouting, and adaptive Markov modeled predictive rerouting.

Key words: Dynamic rerouting, congestion detection rerouting, single thresholds detection rerouting, double threshold detection rerouting, group controlled rerouting, adaptive Markov modeled predictive rerouting.

1 Introduction

A great many studies have been presented involving recovery from congestion in networks. Rerouting a path is one of the methods of congestion recovery. If a packet is caught in a congested node, it will take an alternate route or wait in previous node until the congestion is alleviated (or do something else). For example, in a hierarchical cellular network, a call in a microcell can be rerouted to a macrocell when congestion or overflow occurs in the microcell, or vice versa. Here, several decentralized rerouting algorithms are presented

[1] E-mail: shkim@ece.sunysb.edu
[2] E-mail: tom@ece.sunysb.edu

which reroute traffic when congestion occurs in a node, instead of blocking the traffic immediately or lowering source rates. Decentralized control improves routing processing time and performs a simpler control [13], because each node is responsible for just few other nearby nodes. Decentralized control can stabilize traffic flow and equalize traffic load throughout the network.

In this study, single threshold detection rerouting, double threshold detection rerouting, group controlled rerouting, and adaptive Markov modeled predictive rerouting algorithms are presented. The first two algorithms presented can increase system performance by detecting the buildup of congestion (pre-congestion) with simple mechanisms. The last two algorithms presented can reduce the amount of control traffic by inferring congestion based on the group of control signals or a small number of samples. Also, these algorithms can maximize system throughput by spreading traffic instead of controlling source rate. Last but not least, we go beyond earlier proposals involving the use of alternating paths upon failure [13] and discuss congestion avoidance rerouting. It is required to note that rerouting is the distribution[7] of traffic all around the network. This study is applicable to both virtual circuit service (packet following virtual circuits) as well as datagram service.

It is worthwhile to review work on congestion recovery. Some of the studies presented to date perform control through the source rate or the intermediate node source rate rather than through rerouting [20]. Network state advertisement packets broadcasting using a flooding protocol has also been studied [11][14]. Rerouting traffic has been studied for failed link recovery [6][9][10][16]. For instance, the Flexible Rerouting Protocol (FRP) [17] triggers upon virtual path (VP) failure and chooses an alternate VP. In other words, it does not consider returning the traffic to the original (or preferred) path even though the congestion may be alleviated.

In our study, the network path rerouting methods are presented based on packet-switched networks. Generally, congestion can occur in time and space level [1]. In time, congestion occurs in the packet (or cell) level, burst level, or the call level. Examples of congestion control in time are Selective Cell Discard, Dynamic UPC, Loss Feedback, Disconnection, and so on. In space, congestion occurs in a node or at several nodes. In this paper, space level congestion with dynamic rerouting is presented and simulated.

At present, congestion control in virtual circuit subnets [2], Dynamic Alternating Routing [4] and multipath routing incorporating congestion feedback mechanism [14] suggests a rerouting method by avoiding the congested node. But the algorithm presented in [2] requires all the nodes between the source and destination nodes to redraw the whole new network topology without congested nodes. In the case of Dynamic Alternating Rerouting [4], it takes two paths for every pair of customer source and destination (these could be

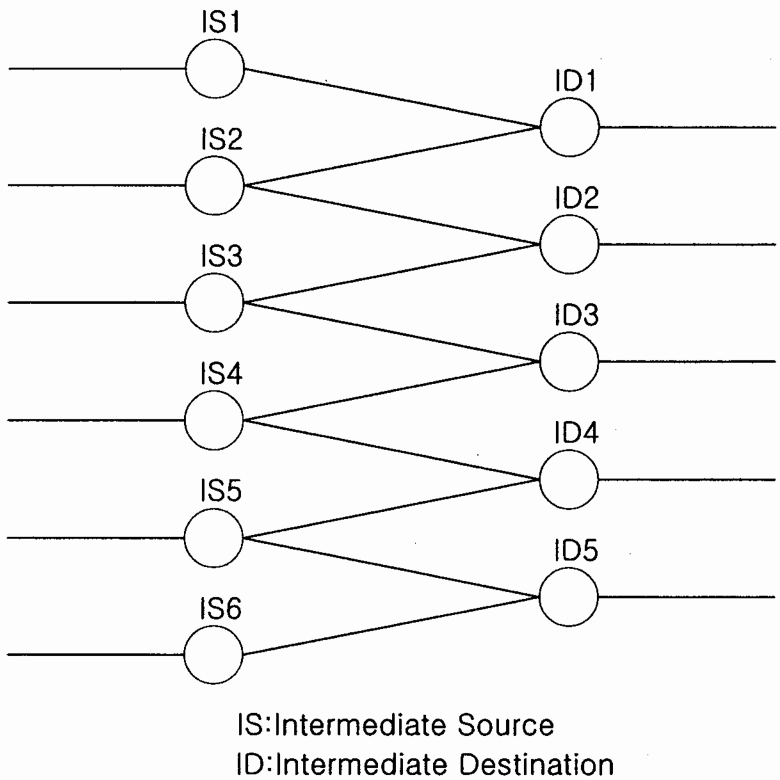

Fig. 1. Simulation network model

the shortest two paths between the locations). If there is no idle path along the preferred path, the call is assigned to another path. If there is no idle path in the second path, the call is blocked. But this algorithm requires continuous feedback traffic which indicates ACK (Acknowledgment) and node status. Also the multipath routing [14] incorporating a congestion feedback mechanism depends on continuous network congestion status feedback. In addition to large amount of feedback control traffic, all of three rerouting algorithms trigger an alarm after detecting congestion.

Other solutions to improve this problem include Random Early Detection (RED) [12] and Explicit Congestion Notification (ECN) [19] routers. These algorithms slow down the sources based on the early notification of congestion. But, these methods require a great deal of control traffic, and control the source rate instead of rerouting. Also, these methods can not alleviate burst traffic, and can have high average queue lengths, thus possibly being ineffective [20].

This article organized as follow. Several rerouting algorithms are discussed in section II. The simulation model is presented in section III and simulation parameters appear in section IV. A comparison of each algorithm's performance is presented in section V. The conclusion appears in section VI.

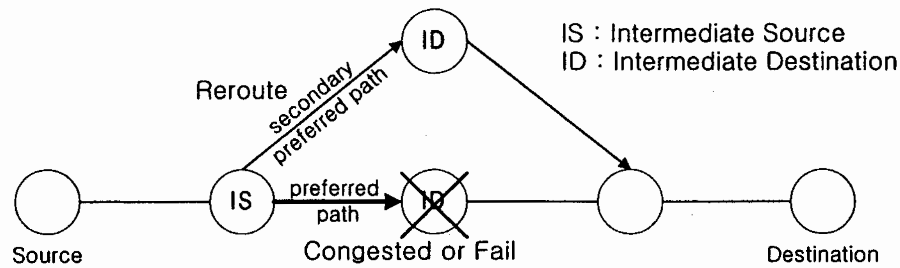

Fig. 2. Basic rerouting method

2 Rerouting Algorithms

In this study, several rerouting algorithms are presented. The rerouting mechanism is similar to deflection routing [8][15][18] which is simple to implement in VLSI hardware for high speed networks. Under generally known discussed methods, when congestion occurs, traffic is rerouted around the congested node by the following strategies. There is local rerouting which reroutes traffic just around the congested node, local-to-end rerouting and end-to-end rerouting which reroutes traffic from the local node or the source to the destination [10]. In this paper, the first two techniques are considered.

The main purpose of these rerouting algorithm is to improve the system's overall performance. The system performance parameters used are the amount of control traffic, probability of packet loss and throughput. The parameters are obtained through the following performance measures.

$$\begin{aligned} Pr[\text{Packet loss}] &= \frac{\text{Total number of lost packets}}{\text{Total number of lost packets} + \text{Total number of served packets}} \\ &= \frac{\text{Total number of lost packets}}{\text{Total number of generated packets}} \end{aligned} \quad (1)$$

$$\text{Throughput} = \frac{\text{Total number of served packets}}{\text{Total number of lost packets} + \text{Total number of served packets}} \quad (2)$$

$$\text{Control Traffic} = \frac{\text{Total number of control traffic packets}}{\text{Total number of served packets}} \quad (3)$$

In the following, an intermediate source (or destination) is an intermediate node on a communication path that locally serves as a source or destination (see Fig. 1, 2).

2.1 *Congestion Detection Rerouting*

The congestion detection rerouting algorithm is based on the well known Backpressure algorithm. This Backpressure algorithm is used to improve network performance [8][17][20]. The concept of Backpressure is, when one closes a pipe (i.e. congestion or failure occurs), then fluid will be backed up the pipe to the point of source. That means no more flow is allowable. In networks, when a node detects congestion, it stops acknowledging packets involved in congestion. That is, the receiver forces the sender to stop transmitting. When the receiver empties its buffer, a "GO" signal is sent to the sender. In this study, if a node (intermediate source) detects congestion, it alternately reroutes the traffic to another link (or secondary preferred path). If the other path also does not have enough space, it will be lost.

However, this algorithm can not alleviate congestion in a node. Once congestion occurs, a heavy volume of traffic is already concentrated in the congested node. The solution of this problem is rerouting traffic before the congestion occurs, as is done in the following policies. Congestion detection rerouting algorithm results are presented in this paper as a reference algorithm.

2.2 *Single Threshold Detection Rerouting*

If the number of packets in the input buffer of intermediate destination (ID) node reaches a certain threshold, then the ID node will send a rerouting control packet which is an RTNP (Reaching Threshold Notification Packet) to the intermediate source (IS) node. Then the IS selects an alternate route, so rerouted traffic will be sent to the second path. Single threshold detection rerouting is different from window flow control and credit based control, because the source rate is not changed, instead traffic is spread to other less loaded nodes.

2.3 *Double Threshold Detection Rerouting*

Each ID input queue has two thresholds. If the number of packets is below the low threshold, the IS continues to use the path. If the number of packets is between the low and the high threshold, the IS will choose randomly either ID's which are connected to the IS (one of them is a preferred path and the other is a secondary preferred path). If the amount of traffic is above the high threshold, the IS will reroute the traffic to the secondary preferred path. That

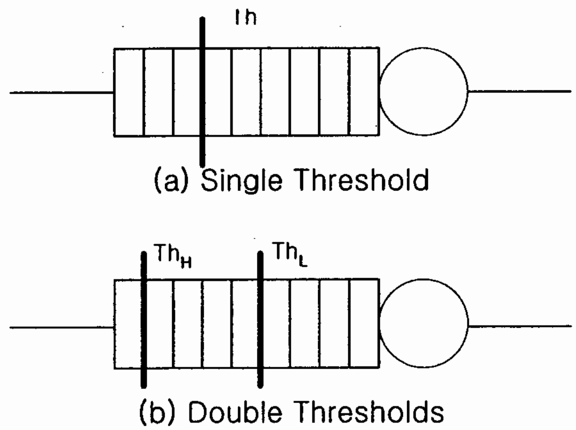

Fig. 3. Single and double threshold queue model

is,

$$Rerouting = \begin{cases} N_q > Th_{high}, & reroute = 1 \\ Th_{low} < N_q < Th_{high}, & reroute = random(0,1) \\ N_q < Th_{low}, & reroute = 0 \end{cases}$$

Here, N_q is the number of packets in the system, and Th_{high}, Th_{low} are threshold high and threshold low respectively.

2.4 Group Controlled Rerouting

In order to reduce the amount of control traffic, one can use group controlled rerouting (GCR). The single threshold rerouting model is used for GCR. Each ID accumulates the RTNP's until the number of RTNP's reaches some arbitrary limit, at this point the ID sends back the rerouting notification to IS. By doing this, a considerable amount of control traffic is reduced, but, as a trade off, the throughput and packet loss probability are degraded.

2.5 Adaptive Markov Modeled Predictive Rerouting

The RTNP's from each ID in the single threshold detection rerouting network are adaptively modeled by a 1-dimensional binary Markov model for short periods of time (the so called learning time). Specifically, a brief record of RTNP's are used to calculate the transition probabilities of the Markov source model of Fig. 4. Then, each IS generates its own rerouting control signal using the Markov source model and *without* control signals from the each ID.

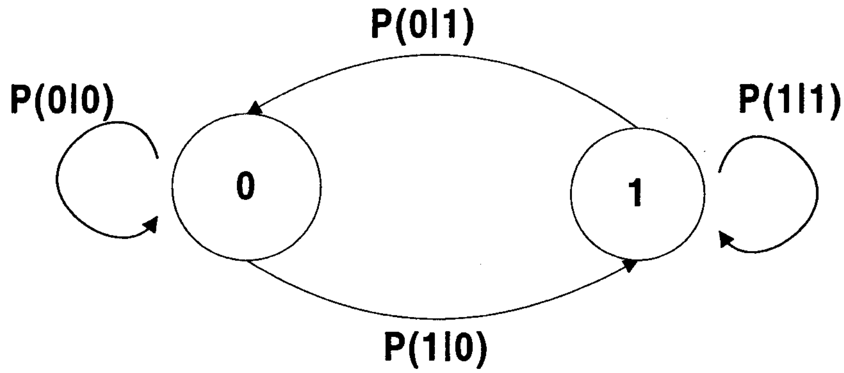

Fig. 4. Markov source for adaptive Markov modeled predictive rerouting

However, in general, it is unknown whether the RTNP source is Markov or not, or i.i.d. (independent, identically distributed), or stationary, or ergodic. We assume it to be Markov.

It is easy to see that the adaptive Markov modeled predictive rerouting and Group control rerouting require a smaller amount of control traffic than the congestion detecting rerouting algorithm and the threshold detection rerouting algorithms (see Fig. 15).

Finally, Fig. 5 shows the rerouting control signal generated from the each ID when the single threshold detecting algorithm and the adaptive Markov rerouting algorithm are used separately. The figure shows a similarity between the rerouting control signal from the single threshold detecting algorithm and the random control signal from the adaptive Markov source in comparing (a) and (b) for ID1 and (c) and (d) for ID2.

3 Simulation Model

In Fig. 1 which is our simulation model, the bottlenecks occurs in the ID's. There are 6 IS's and 5 ID's. Two IS's (IS1, IS2) send traffic to an ID (ID1), also see table 1. For this type network, heavy traffic can be offered in order to generate congestion. If ID1 does not have enough idle buffer space, the traffic coming from IS2 whose preferred path is ID1 will be rerouted to ID2. In our network topology, traffic generated from IS2 will be served in either ID1 or ID2, or will be lost. This type of network topology is required to prevent an "avalanche" (i.e. ripple effect) of rerouted traffic in the network. That is, this type of network prevents the direct path from using an alternate path which is far from the direct path.

Each IS has a routing function which depends on the control signal from each ID. Each ID generates control signals indicating the status of the ID's input

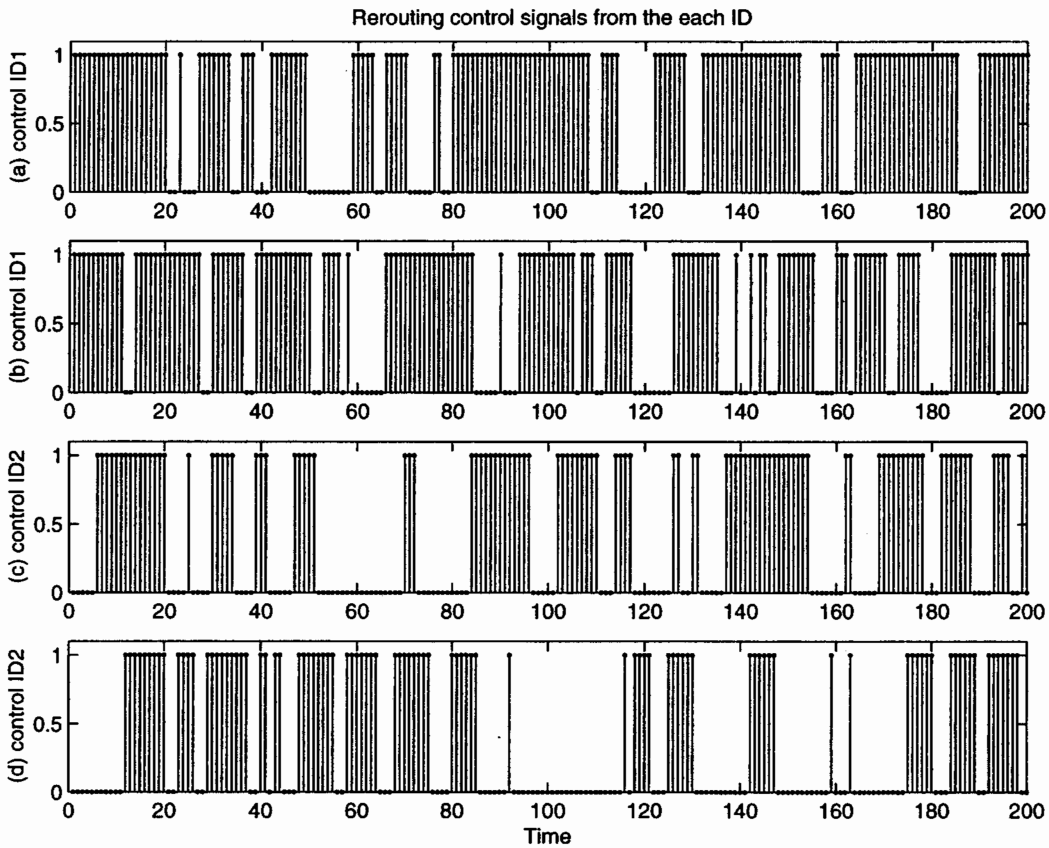

Fig. 5. Rerouting control signals (a) from ID1 by the single threshold rerouting (b) for ID1 by the adaptive Markov rerouting (c) from ID2 by the single threshold rerouting (d) for ID2 by the adaptive Markov rerouting

Table 1

Preferred paths in network (BR = see Equation (13))

For BR=5/6		For BR=3/4	
IS	Preferred destination	IS	Preferred destination
IS1	ID1	IS1	ID1
IS2	ID1	IS2	ID1
IS3	ID2	IS3	ID2
IS4	ID3	IS4	ID3
IS5	ID4		
IS6	ID5		

buffer, and follows the rerouting algorithms. If moderate traffic is offered to the network, rerouting incidents rarely occur. By using 6 IS's and 5 ID's, it is easy to create congestion and bottlenecks. Each link has a transmission speed of 155.52 Mbps (SONET OC-3) and has a deterministic service time

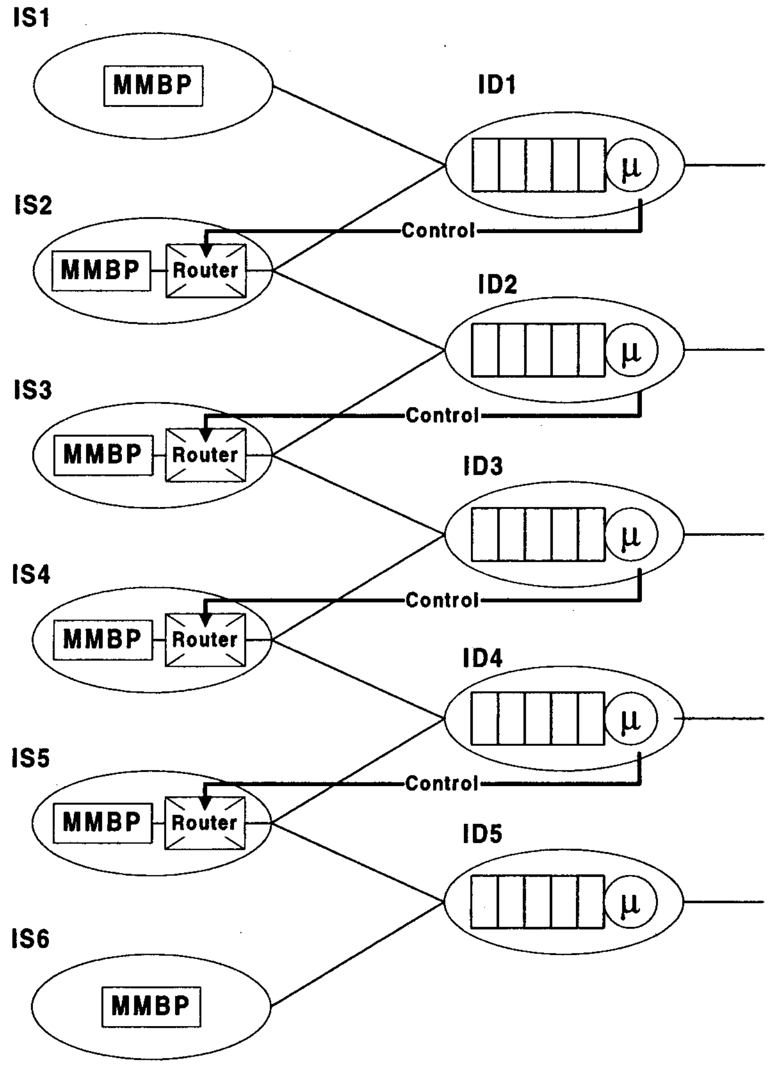

Fig. 6. Simulation Model

during which a single packet length processed by each server. Packet size is set to 53 bytes (or 424 bits) yielding a discrete event slot length of 53 bytes. Processing time is ignored. Control signals coming from the ID's have a unit packet delay. The model is described in Fig 6. Matlab and Simulink were used for the simulation.

3.1 Intermediate Source (IS)

Each IS has an Markov Modulated Bernoulli Process (MMBP) source generator and a router which routes the packet to the ID's. A MMBP is the discrete-time analog of a Markov Modulated Poisson Process. The MMPP has been used widely to model voice-data network traffic. Also it is appropri-

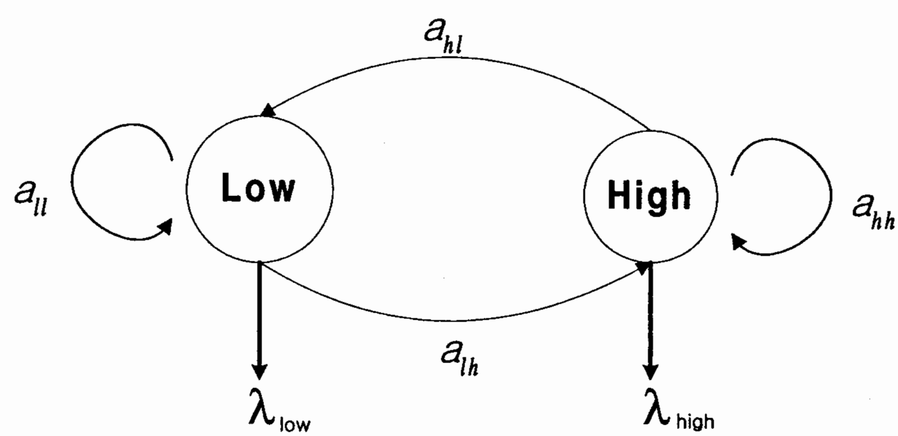

Fig. 7. Markov modulated Bernoulli process for intermediate source

ate to represent actual traffic which shows correlations over time, such as ATM multimedia traffic [3][5]. Time in MMBPs is discretized into fixed length slots. That is, a single slot is assumed to be 53 bytes in length. A two state MMBP source is generated. The transition matrix of the state transition diagram is (also see Fig. 7),

$$Q = \begin{bmatrix} a_{ll} & a_{lh} \\ a_{hl} & a_{hh} \end{bmatrix} = \begin{bmatrix} a_{ll} & 1 - a_{ll} \\ 1 - a_{hh} & a_{hh} \end{bmatrix} \tag{4}$$

$$\underline{\pi} = \underline{\pi} Q \tag{5}$$

Here $\underline{\pi} = [\pi_{low}, \pi_{high}]$ are equilibrium state probabilities. When the arrival process is in the π_{low} state, it generates a packet with probability of γ and when in the π_{high} state, it generates a packet with probability of ζ. Also, $\pi_{low} + \pi_{high} = 1$. To obtain each probability,

$$[\pi_{low}, \pi_{high}] \begin{bmatrix} a_{ll} & 1 - a_{ll} \\ 1 - a_{hh} & a_{hh} \end{bmatrix} = [\pi_{low}, \pi_{high}] \tag{6}$$

then,

$$\pi_{low} = \frac{a_{hh} - 1}{a_{hh} + a_{ll} - 2}, \qquad \pi_{high} = \frac{a_{ll} - 1}{a_{hh} + a_{ll} - 2} \tag{7}$$

For the high state, the expected interarrival time is,

$$E[\text{Interarrival time of high-state}] = \sum_{i=1}^{\infty} i(1-\gamma)^{i-1}\gamma = \frac{1}{\gamma} \quad (8)$$

For the low state, the expected interarrival time is,

$$E[\text{Interarrival time of low-state}] = \sum_{i=1}^{\infty} i(1-\zeta)^{i-1}\zeta = \frac{1}{\zeta} \quad (9)$$

The packet arrival rate λ will be,

$$\lambda = \pi_{low}\lambda_{low} + \pi_{high}\lambda_{high} \quad (10)$$

Here, λ_{low} is the arrival rate for the low-state and λ_{high} is the arrival rate for the high-state. In the simulation,

$$\begin{aligned} a_{ll} &= \Pr(low|low) = 0.7 \\ a_{lh} &= \Pr(high|low) = 0.3 \\ a_{hl} &= \Pr(low|high) = 0.1 \\ a_{hh} &= \Pr(high|high) = 0.9 \end{aligned} \quad (11)$$

are used, then $\pi_{low} = 0.25$, $\pi_{high} = 0.75$.

The service time of each server is fixed at one slot time because of packet based service processing. The buffer size is determined as in [20], that is the buffer size B is proportional to μT (μ is the service time, T is the minimum round trip time and $T =$ (Fixed propagation delay + $\frac{1}{\mu}$)).

$$\begin{aligned} B &= \alpha\mu T \\ &= \alpha\mu(\text{Fixed propagation delay} + \frac{1}{\mu}) \end{aligned} \quad (12)$$

Here, $\alpha = 10$ is used. The fixed propagation delay is ignored. Thus the buffer in each node is $B = 10$.

3.2 Intermediate Destination (ID)

Each ID (Intermediate destination) has a queue and a server which has a deterministic service time. Our simulation model is instrumentated to record the node status. Rerouted traffic may be sent to the standby path (which is reserved by the IS) or to another path which is connected to the IS as a secondary preferred path. server busy/idle indicator, service time generator, and programmable module for different rerouting algorithm.

In the network topology, each ID may or may not be located next to each other. The latter case is for preventing a propagation of congestion around a congested node, so each IS may choose a secondary path which is independent of the preferred path.

3.3 Simulation

A discrete event simulation was performed for a SONET rate of 155.52Mbps. Fig. 8 depicts the reliable simulation time. After 50000 slots, which is 0.1363 sec, the probability of packet loss converges to a stable value. In our simulation, after running the network simulator for 0.2726 sec which is 100,000 discrete event slots, data was collected.

4 Simulation Parameters

In simulating the rerouting network, several parameters are obtained. To observe the differences in performance, we use a bottleneck ratio (BR).

$$\text{Bottleneck ratio (BR)} = \frac{\text{number of destination nodes}}{\text{number of source nodes}} \tag{13}$$

Note that for our topology the maximum bottleneck ratio is one. If the bottleneck ratio is decreased, more congestion occurs. In our simulation, two cases were simulated. One is 3 ID's accepting traffic coming from 4 IS's, and the other is 5 ID's accepting traffic coming from 6 IS's. The latter is more stable than the former. In the simulations, the bottleneck ratios are assigned as 5/6=0.8333 and 3/4=0.75.

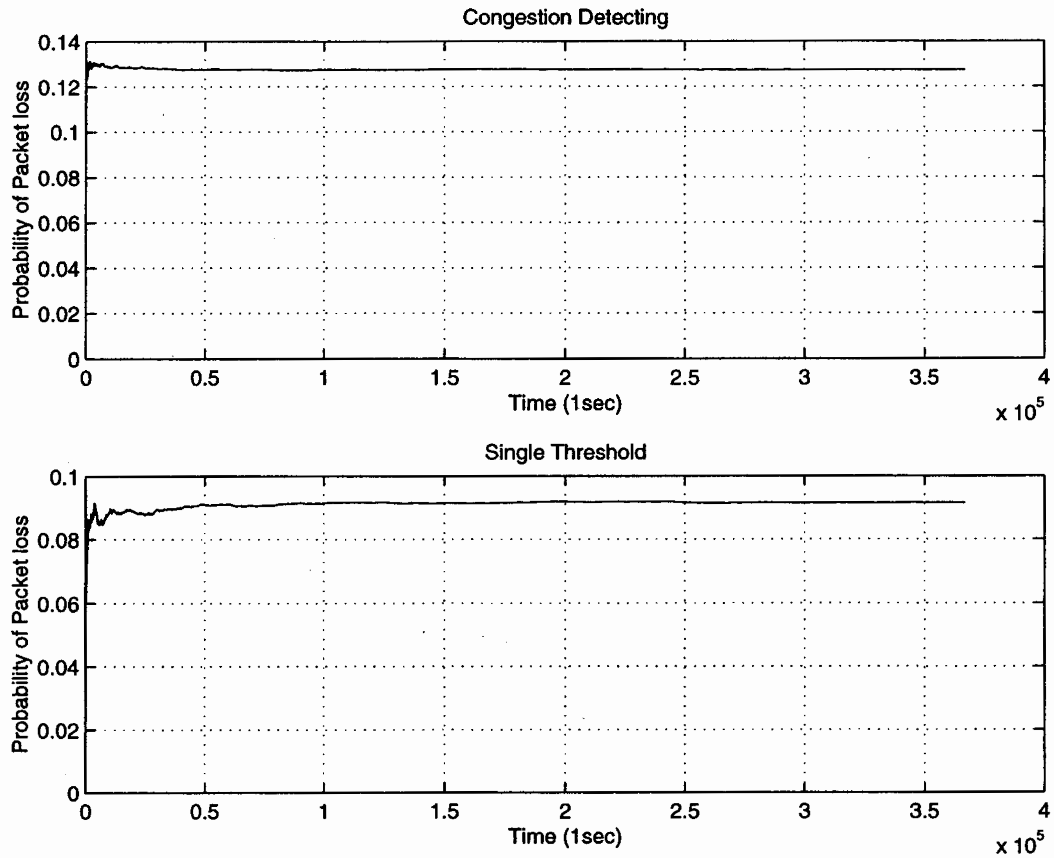

Fig. 8. Probability of packet loss variation when congestion detection and single threshold rerouting are used

4.1 Threshold for the Single Threshold Rerouting Algorithm

In order to obtain a good choice of threshold for single threshold detection rerouting, control traffic volume and throughput are considered (see Fig. 9 which is plotted twice with two simulations for 100,000 slots ($\approx$ 0.3sec) with the same parameters, also see the simulation model). With a different load (116Mbps), the result is the same as for the 128.3 Mbps case which is done in Fig. 9.

From Fig. 9, the throughput reaches some limit as threshold is increased, but the amount of control traffic decreases. Choosing a threshold of 9 is unstable because it is easy to overflow, thus, for our network, a good choice of threshold will be 8 which has a maximum throughput and a reasonable control traffic volume.

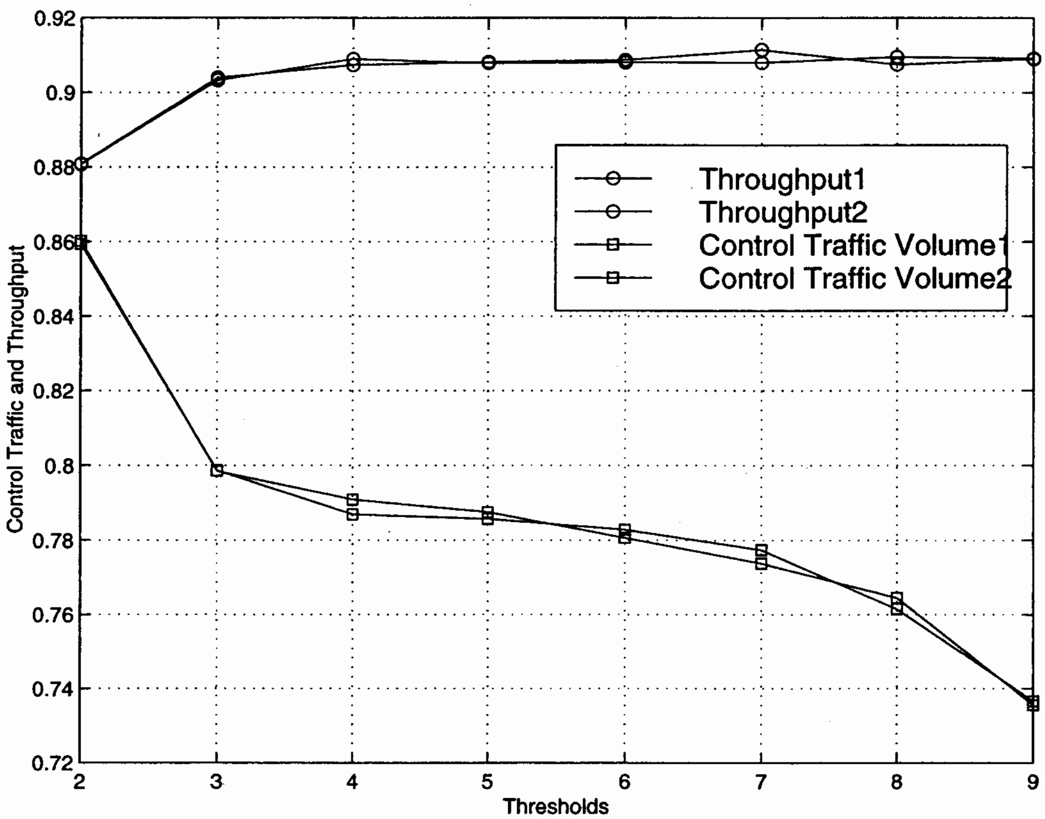

Fig. 9. Performance comparison for the single threshold detection rerouting varying threshold (Offered load to each IS is 128.3Mbps and buffer length is 10)

4.2 Thresholds for the Double Threshold Rerouting Algorithm

Fig. 10 depicts the overall network's amount of control traffic and throughput when the double threshold varies as [High threshold, Low threshold] to find a good choice of thresholds (the simulation follows the simulation model section description). The amount of control traffic mostly depends on the low threshold with a small amount of control traffic variation, because more control packets are generated as the threshold is lowered. Thus the control traffic curve is a saw tooth like function as the thresholds are increased. The throughput is not sensitive to the threshold values. In our simulation, the high and the low threshold were chosen as [9 7] respectively as a good choice, because it shows a relatively small amount of control traffic. However, choosing [9 8] is the same as the single threshold case in terms of the controlling method because there is no gap between the low and high threshold. Here, the utilization in tables is obtained by Offered Traffic / Data Rate (155.52Mbps).

According to the simulation result, the double threshold mechanism improves throughput and packet loss probability by only less than 1% under the same condition as the single threshold mechanism (see table 2, the simulation model

Table 2
Performance comparison between the single threshold policy and the double threshold policy (when single threshold = 8, and double threshold = 7 and 9. BR=3/4)

Utilization	Offered Traffic (Mbps)	Control Traffic		P[Packet Loss]		Throughput	
		Single	Double	Single	Double	Single	Double
0.55	85	0.0590	0.1022	0	0	1	1
0.77	120	0.2751	0.4118	2.15e-6	0	1	1
0.93	143.9	0.6393	0.7593	0.0998	0.0987	0.9002	0.9013

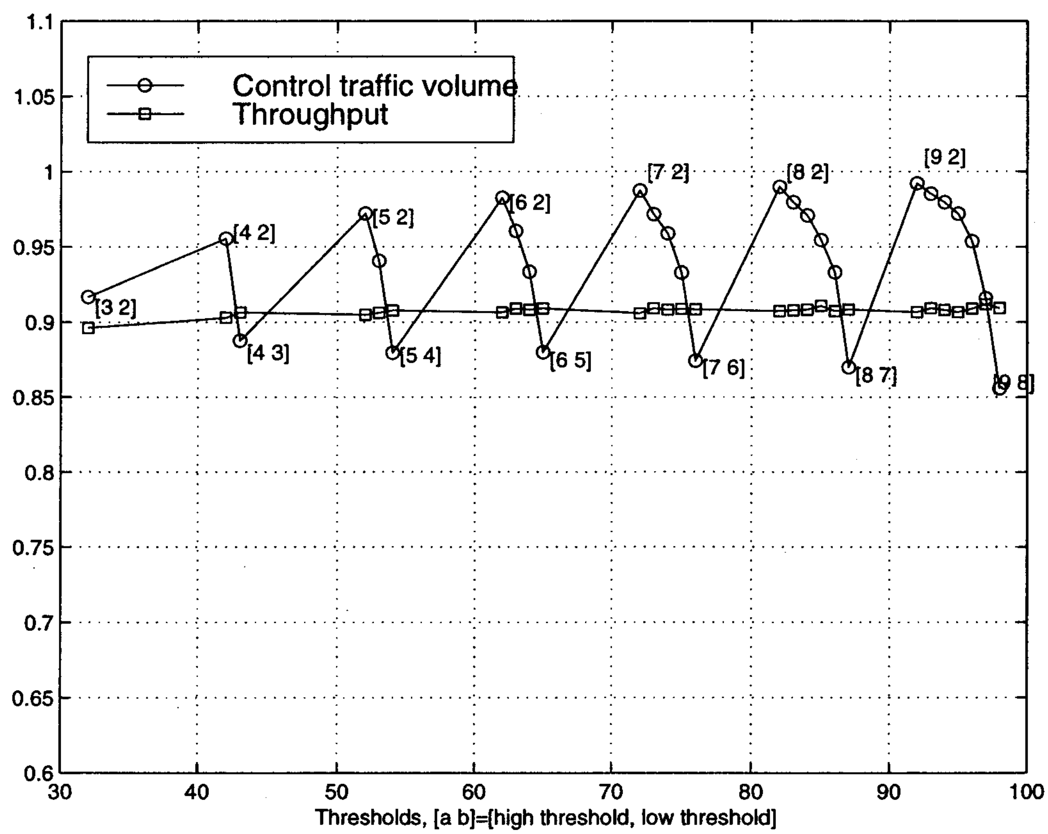

Fig. 10. Performance comparison for the double threshold detection rerouting varying thresholds (Offered load to each IS is 128.3Mbps and buffer length is 10), BR=3/4

follows the simulation model section description and uses 7 and 9 for the low and the high thresholds). The double threshold detection rerouting mechanism generates significantly more control traffic than the single threshold policy. Moreover, the throughput is not sensitive to the thresholds values. Thus, a more efficient algorithm is required for this double threshold policy, instead of the generic policy of section 2.3. According to [21], however, if the double threshold algorithm is used for source flow control instead of rerouting, it shows better performance than if used for on-off flow control.

Table 3
Choosing a threshold for group controlled rerouting : AL = Accumulating Limit, BR =5/6, AL=2

Utilization	Offered Traffic (Mbps)	Control Traffic			P[Packet Loss]		
		Th=3	Th=5	Th=7	Th=3	Th=5	Th=7
0.55	85	0.07262	0.0593	0.0507	0.0008	0.0020	0.0039
0.77	120	0.1035	0.0915	0.0891	0.063	0.063	0.0626
0.93	143.9	0.1827	0.1733	0.1697	0.1061	0.1027	0.1016

4.3 Accumulating Limit and Threshold for Group Controlled Rerouting

In table 3, the first simulation's threshold is set to 3, the second simulation's threshold is set to 5, the third simulation's threshold is set to 7. All of these simulations accumulate 1 more RTNP after detecting the first RTNP (i.e. AL=2, so 2 RTNP's generate a rerouting control signal). From the table 3, threshold 7 shows a slightly better performance in large offered traffic. The accumulation limit (AL) can't be set to a large number (above 3) because too much degradation occurs in the throughput and the probability of packet loss. For the simulation to follow, the group controlled rerouting (GCR) uses 2 as an accumulation limit (AL) and the threshold is set to 7. Also, AL=2 is the best choice based on a simulation which is not presented in this paper.

4.4 A Comparison of Markov and Uniform Random Rerouting

The randomness of the RTNP is compared with Markov and uniform random rerouting control which generates rerouting signals with probability of 0.5 (i.e. There is no preferred path) to show the validity of Markov modeling (see Fig. 11).

In Fig. 11, the Markov source shows a better performance than uniform random rerouting for higher data rates and the converse is true for lower data rates.

4.5 Appropriate Learning Time for Markov Modeling

Modeling of each RTNP is required periodically for a short period of time and uses a very small amount of control traffic. In order to obtain an appropriate optimal (i.e. minimum) modeling time of each RTNP, several heuristic (or empirical) simulations were performed (see table 4, the single threshold is set

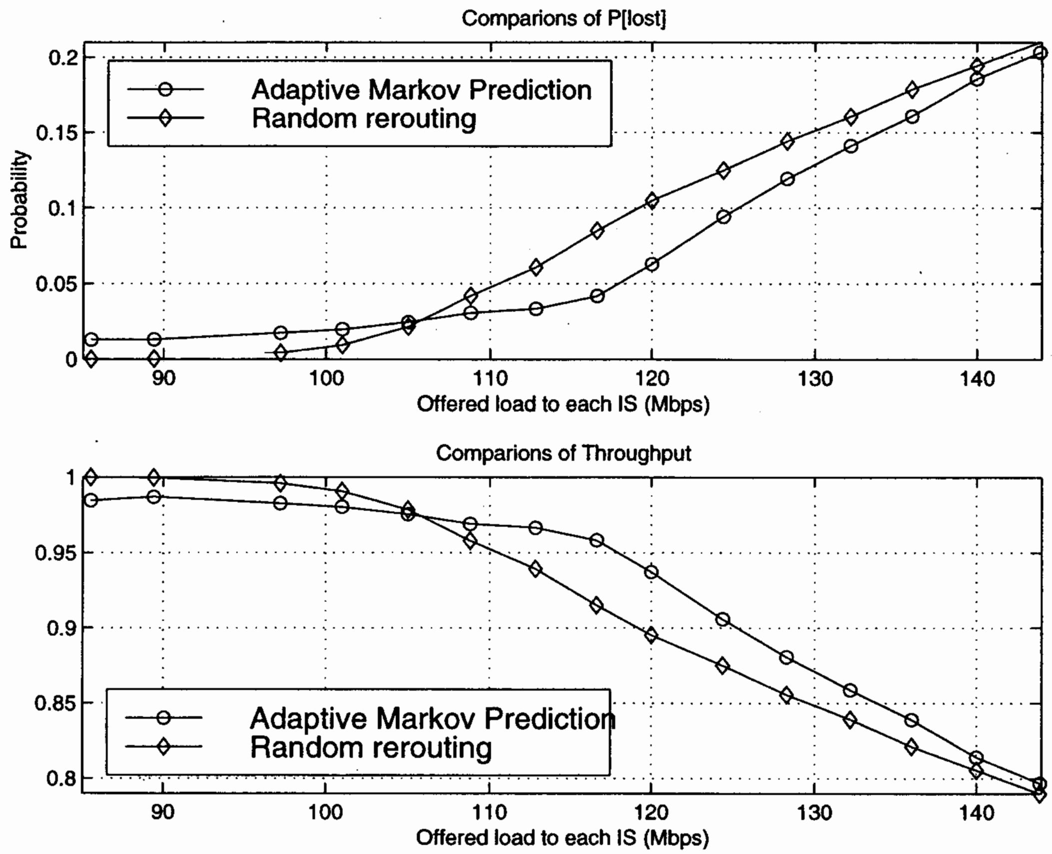

Fig. 11. Comparison of Markov and uniform random rerouting

Table 4
Optimal RTNP modeling time for each ID, when BR=3/4

Modeling time (seconds)	Markov model for ID1				Markov model for ID2			
	Pr(0|0)	Pr(1|0)	Pr(0|1)	Pr(1|1)	Pr(0|0)	Pr(1|0)	Pr(0|1)	Pr(1|1)
1	0.6703	0.3297	0.0886	0.9114	0.7922	0.2078	0.1527	0.8473
0.2726	0.67293	0.32707	0.08798	0.91202	0.79221	0.20777	0.15346	0.84656
0.0136	0.67558	0.32350	0.08991	0.91034	0.78913	0.21040	0.15459	0.84575
0.0027	0.68067	0.31513	0.09974	0.90157	0.79867	0.20133	0.16606	0.83394
0.0014	0.6310	0.3571	0.0745	0.9279	0.7642	0.2311	0.1736	0.8299

to 8, and the simulation parameters follows the simulation model). From the table 4, transition probabilities are stable for time greater than 0.0136 sec. In our simulation, the RTNP is modeled every 100,000 slots which is 0.2726 seconds and the modeling (learning) time is 10,000 slots which is 0.02726 sec.

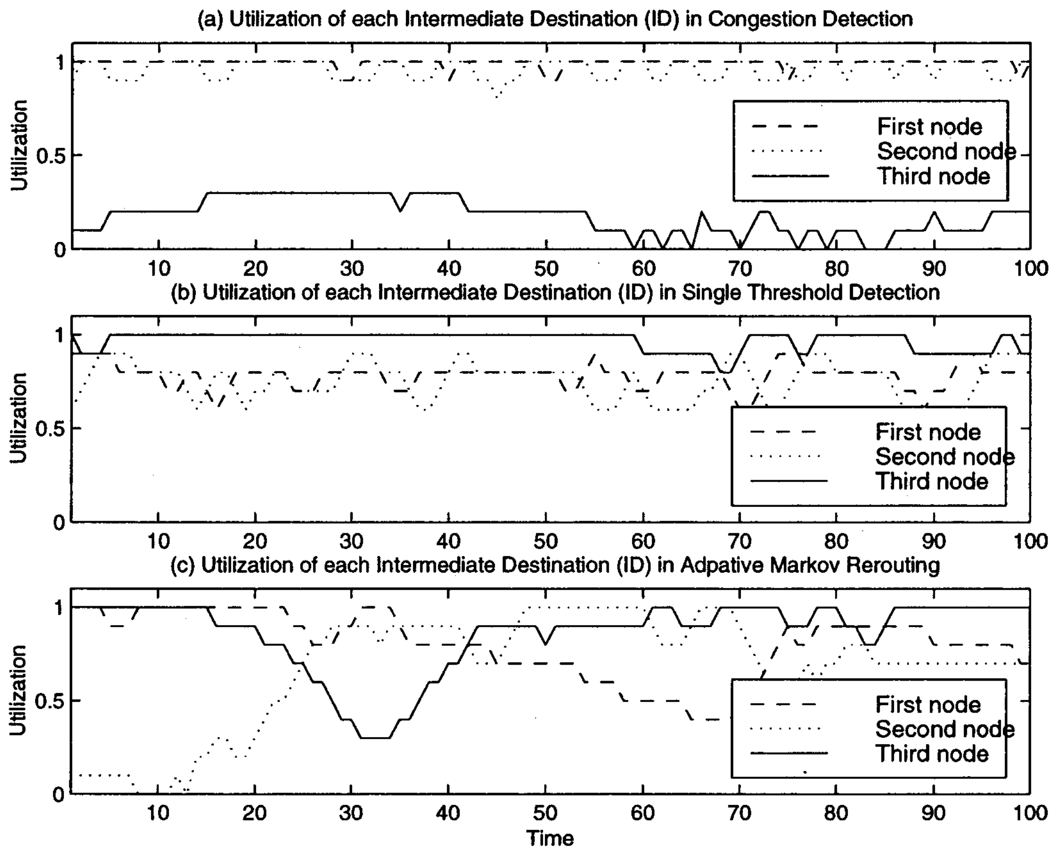

Fig. 12. Utilization of each ID when each IS generates 128.3Mbps load and BR=3/4, (a) Congestion Detection, (b) Single Threshold, (c) Adaptive Markov

5 Performance Comparison

Fig. 12 shows the utilization of each ID for the different algorithms when BR =3/4. For the congestion detection rerouting algorithm, ID1 (which has a severe bottleneck because two IS's send traffic to a single ID1) and ID2 are mostly utilized. This is because the rerouting signal is generated after detecting overflow. Also the packet loss occurs mostly in ID1 and ID2. For the single threshold algorithm, most ID's are fully utilized because threshold is set to 8 (i.e. 80 % of the buffer space) and the algorithm reroutes traffic to the less loaded node. That is, after filling up the buffer approximately 80%, the ID sends a rerouting control signal to the IS to spread traffic. However, due to the heavy offered load, the rerouted traffic is congested in the last node which is ID3 (when BR=3/4). Therefore there is possible packet loss in ID3. Finally, for adaptive Markov rerouting, the rerouted traffic is also congested in ID3 because it is using the single threshold rerouting control mechanism as a Markov model. Most of the ID's are fully utilized. But, every node has chance of packet loss. For adaptive Markov rerouting, the lack of a control signal leads to random fluctuation in the utilization curve.

Performance comparison curves are plotted in Figures 13, 14, 15, 16, and 17. The first three figures are plotted with BR=5/6, and the last two graph are plotted with BR=3/4. For packet loss probability (Fig. 13, 16) and throughput (Fig. 14), the single threshold and double threshold detection rerouting algorithm have the best performance. However, these two rerouting algorithms use the pre-congestion information which incurs a high cost in control traffic. When a large amounts of traffic is offered (from 140 Mbps to 155 Mbps), the performance of packet loss probability and throughput is degraded because of the system overflow due to feeding six 155.52 Mbps sources to five 155.52 Mbps outgoing paths (i.e. a bottleneck).

For control traffic (Fig. 15, 17), as offered load increases, the single threshold and double threshold detection algorithm require a considerably large amount of control traffic, but congestion detection, group control and the adaptive Markov modeled predictive rerouting require a relatively small amount of control traffic. However, as a trade-off, these three algorithms have a degraded performance in terms of throughput and packet loss probability. In particular, the double threshold policy with two thresholds generates a great deal of control traffic.

The single threshold and double threshold detection rerouting algorithms show the best throughput and packet loss probability if the control traffic volume is not considered or if there is a designated control channel such as Signaling System 7 (SS7). Also adaptive Markov predictive rerouting shows a good performance with respect to a relatively high throughput and a very low amounts of control traffic.

6 Conclusion

This study presents several dynamic rerouting algorithms for the packet-switched networks, and also presents simulation results. One can see in the performance comparison section that there is a trade-off between information (i.e. control traffic) and performance. The policies with the best performance (single and double threshold detection rerouting) generate the most control traffic. One can reduce the control traffic (using group controlled rerouting and adaptive Markov modeled predictive rerouting) but performance suffers to some extent. An open research question is whether it is even possible to develop a congestion rerouting policy that matches the performance of the threshold policies without the corresponding large amount of control traffic.

References

[1] D.E. McDysan and D.L. Spohn, *ATM Theory and Application,* MacGraw-Hill, 1994.

[2] A.S. Tanenbaum, *Computer Networks,* 3rd ed., Prentice-Hall, 1996.

[3] T.G. Robertazzi, *Compter Networks and Systems : Queueing Theory and Performance Evaluation,* Springer-Verlag, 3rd ed., 2000.

[4] J. Walrand and P. Varaiya, *High-Performance Communication Networks,* 1st ed., Morgan Kaufman Publishers, 1996.

[5] Ng Chee Hock, *Queueing Modelling Fundamentals,* John Wiley and Sons, New York, 1997.

[6] P. Veitch, I. Hawker and G. Smith, "Administration of Restorable Virtual Path Mesh Networks," *IEEE Communication Magazine*, Dec. 1996.

[7] Y.A. Korilis, A.A. Lazar, and A. Orda, "Architecting Network Optima Using Stackelberg Routing Strategies," *IEEE/ACM Transactions on Networking,* vol.5, no.1, pp. 161-173, Feb. 1997.

[8] E. Leonardi and F. Neri, P. Torino, "Congestion Control in Asynchronous, High-Speed Wormhole Routing Networks," *IEEE Communication Magazine,* pp. 58-69, Nov. 1996.

[9] T. Wu, I. Korperoglu, B. Cheng, "Distributed Interactive Video System Design and Analysis," *IEEE Communication Magazine*, March, 1997.

[10] K.G. Shin and S. Han, "Fast Low-Cost Failure Recovery for Reliable Real-Time Multimedia Communication," *IEEE transactions on Network* , Nov. 1998.

[11] M. Sivabalan and H.T. Mouftah, "QUARTS-II: A Routing Simulator for ATM Networks," *IEEE Communication Magazine,* vol. 365, pp. 80 -87, May 1998.

[12] S. Floyd and V.Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking,* vol. 1, no. 4, Aug. 1993.

[13] E.Wong, A. Chan and T. Yum, "A Taxonomy of Rerouting in Circuit-Switched Networks," *IEEE Communication Magazine*, vol.37, no. 11, Nov. 1999.

[14] J. Sole-Pareta and I.F. Akyildiz, "Adaptive Multipath Routing of Connectionless Traffic in an ATM Network," *ICC '95*, vol.3, pp. 1626-1630.

[15] H. Huang, T.G. Robertazzi, A. Lazar, "A Comparion of Information Based Deflection Strategies," *Computer Networks and ISDN Systems, 27,* pp. 1399-1407, 1995.

[16] K. Mahesh, S. Lakshminarayanan, S. Raghaven, "An Efficient Rerouting Mechanism for Dynamic Network Resource Management," *IEEE International Conference on Multimedia Computing and Systems Proceedings,* vol 2. pp. 112-116, Jun 7-11, 1999.

[17] Y. Dubinsky and A. Segall, "A Flexible Rerouting Protocol in ATM Networks," *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 3 , pp. 1488 -1496.

[18] N.F. Maxemchuk, "The Manhattan Street Network," *Proceedings of IEEE Globecom '85*, pp. 255-261, Dec. 1985.

[19] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Compter Communication Review,* vol. 24, no. 5, Oct. 1995.

[20] C.M. Pazos and M. Gerla, "Using Back-Pressure to improve TCP Performance with Many Flows," *Proceedings of Infocom '99*, New York, March, 1999.

[21] Seong-Hwan Kim, "A Study on Congestion Control of Interconnections between ATM Network and LAN/MAN," *Master's Degree Thesis*, Kyung-Hee University, 1995.

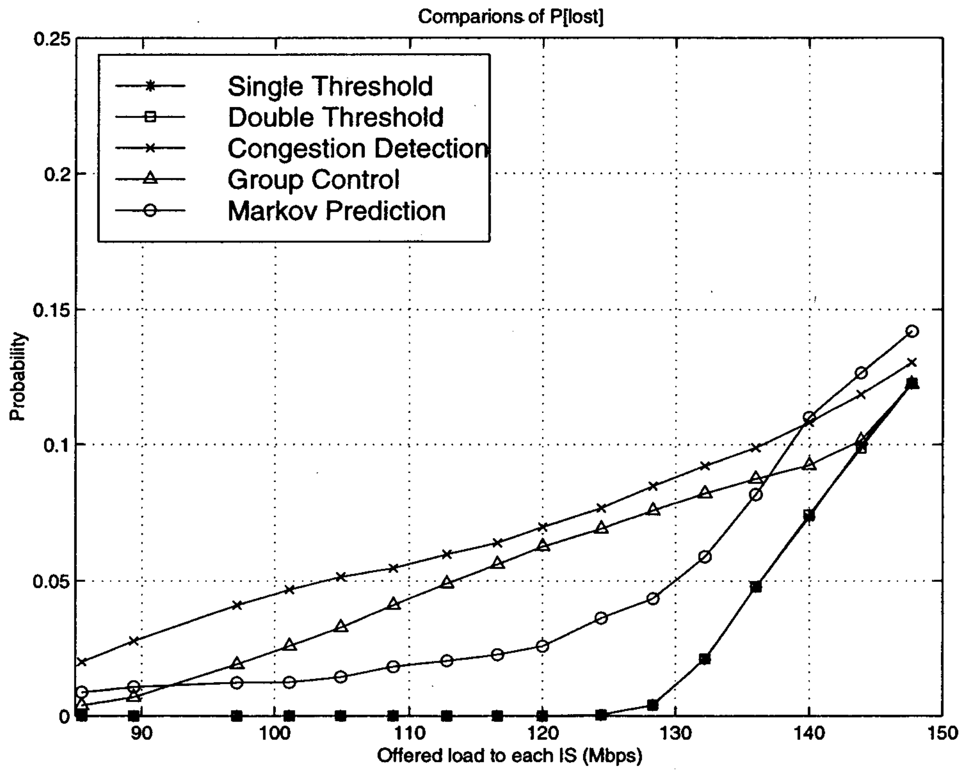

Fig. 13. Comparison of overall network's probability of packet loss, when BR=5/6

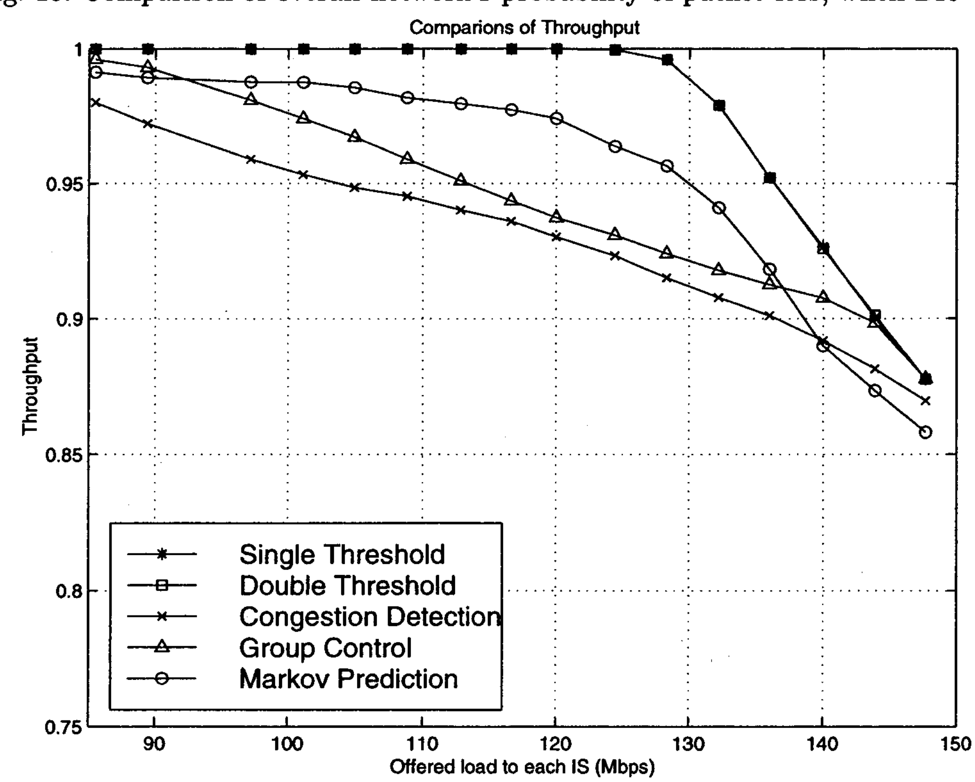

Fig. 14. Comparison of overall network's throughput, when BR=5/6

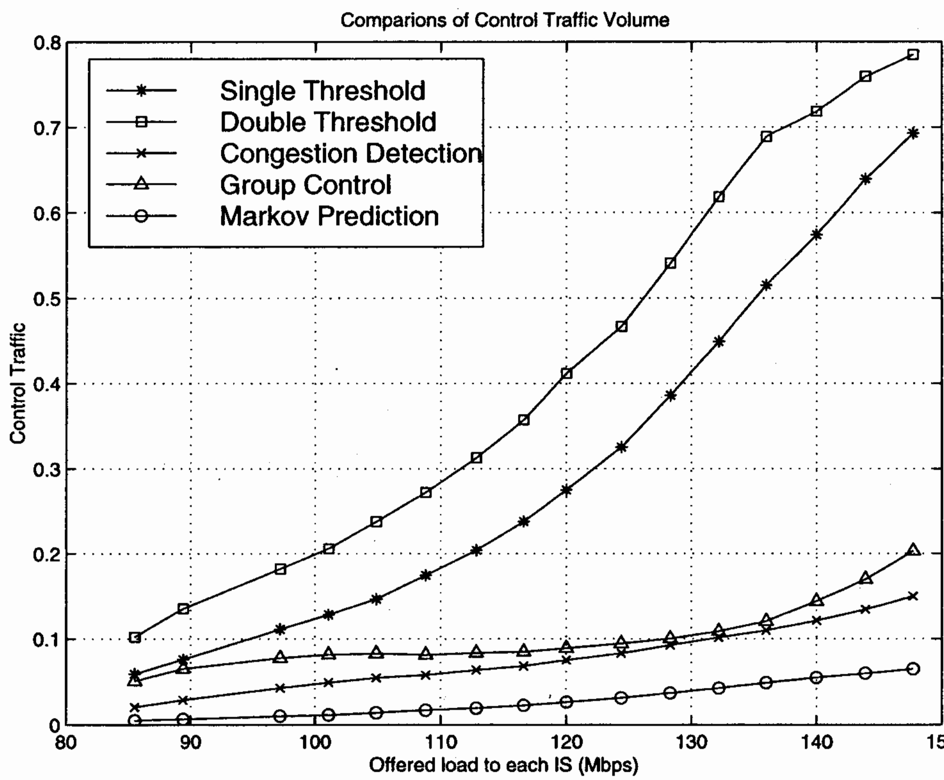

Fig. 15. Comparison of overall network's control traffic volume, when BR=5/6

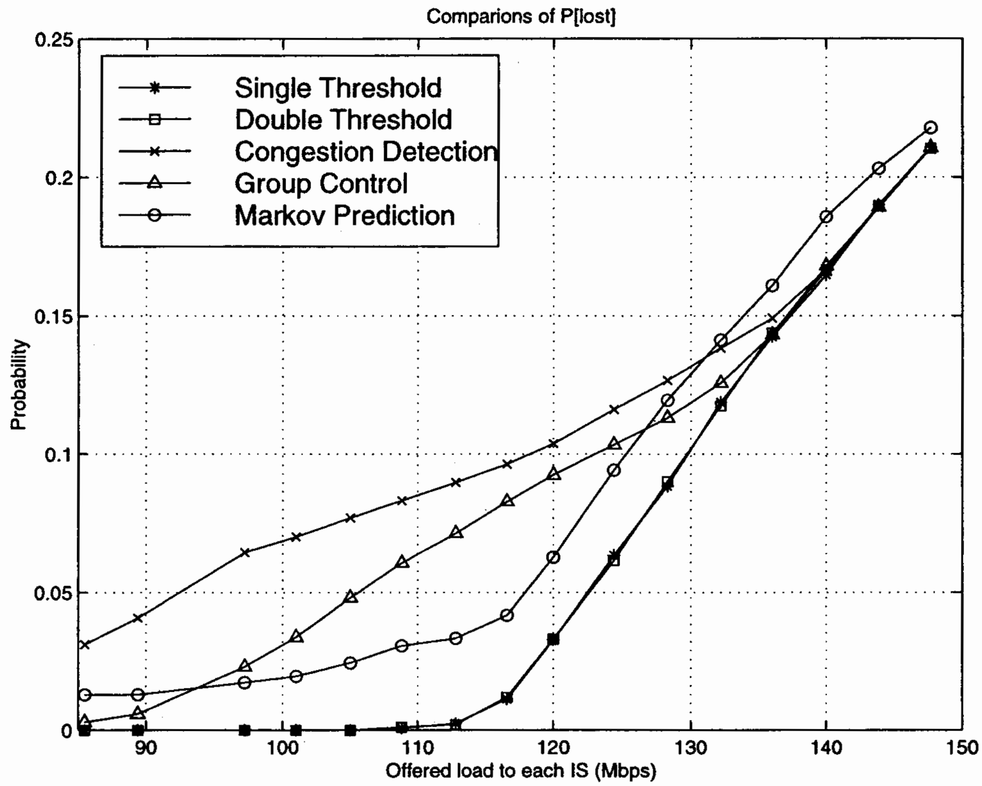

Fig. 16. Comparison of overall network's probability of packet loss, when BR=3/4

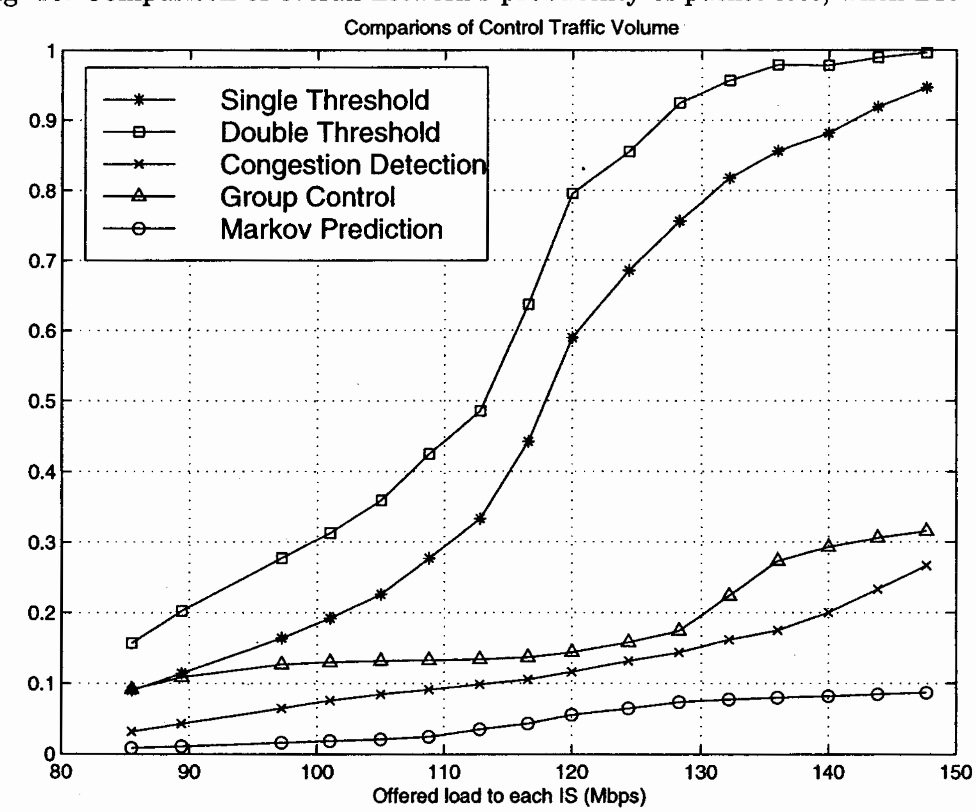

Fig. 17. Comparison of overall network's control traffic volume, when BR=3/4