STATE UNIVERSITY OF NEW YORK AT

STONY BROOK

CEAS TECHNICAL REPORT 613

Performance Evaluation of the Manhattan Street Newtwork with Input Buffers

H.-Y. Huang and T.G. Robertazzi

December 17, 1991

Performance Evaluation of the Manhattan Street Network with Input Buffers

Thomas G. Robertazzi** Hsin-Yuan Huang
Dept. of Electrical Engineering,
SUNY at Stony Brook,
Stony Brook, NY 11794
tom@sbee.sunysb.edu
516-632-8412

September 23, 1991

Abstract

Simulation results for an 8x8 Manhattan Street Network with input buffers are presented. Performance data is presented for four different routing strategies, with six local traffic rates, and thirteen input buffer sizes. It is shown that routing strategy based on packet age yields the best performance. The effect of input buffer size on blocking probability is emphasized.

** Please address all correspondence to T. Robertazzi

300 Series Talk: Metropolitan Area Networks (MAN's) is best match.
Switching is the next best match.

1 Introduction

Modern communication plays a most important role in our everyday life. But modern communication does not mean pure voice exchange anymore. Rather things like computer data and video signal are also parts of today's sophisticated communication environment. In an environment like this, fiber optics are often used. Although fiber optics offer the potential of very large bandwidth, a limitation is the speed of electronic devices. Unless all-optic communication devices become realizable, this constraint will be with us for a long time.

To reduce the bottleneck imposed by electronic devices, besides increasing the operation speed of these devices, one can reduce the processing time for network operations through the use of a parallel switching fabric with synchronous operation using simple routing algorithms. The Manhattan Street Network (abbreviated as MSN) proposed by N.F. Maxemchuk [2, 3] is one such fabric. It was proposed originally for use as a metropolitan area network backbone.

The MSN is a two connected regular mesh network (Fig. 1). It can be naturally embedded on the surface of a torus. The N nodes in the same row or the same column are connected in a unidirectional loop. Adjacent rows and adjacent columns alternate direction like the streets of Manhattan.

Since each node in a MSN has two input links and two output links, at any given time slot at most two packets can arrive at the same node and leave for other nodes through the two output links. There is thus no internal blocking in the nodes, and storing packets is not necessary. Deflection routing is used in the MSN. That is if two packets in a node's packet buffers prefer the same output link, one packet is sent to the preferred link and the other is "deflected" to the non-preferred link. Because each of the two packets that can be in a node will prefer one or either output link, there are six different routing events [5]. Only one of them correspond to a deflection ("conflict"). Packets experiencing a deflection will increase their distance to the destination by at most 4 hops [3, 5], and different routing strategies yield different numbers of deflections.

New traffic can only enter a node if at least one of the node's two packet buffers is not occupied by a transiting packet. So if both packet buffers are occupied, local traffic is blocked. Even under light load there is a considerable chance of blocking, making the addition of input buffers necessary [5]. A properly sized input buffer serves to present an acceptable blocking probability to arriving packets while maximizing the probability that there is a packet in the input buffer head of line ready to be placed in an empty packet buffer. The purpose of this work is to evaluate different routing strategies operating with input buffers.

We note that there are other topologies with similar characteristics being proposed, e.g. HR^4-NET [1] with bidirectional links, and TAC [4] with a triangular mesh instead of a rectangular mesh.

2 Network Model

The network model used in this simulation is an 8x8 Manhattan Street Network. It is simulated for a half million time slots, and the data from the first 2000 slots are discarded as transient. It differs from simulation results for one million slots by less than 0.1%, so the result from this simulation is considered to be statistically stable. During the simulated period, more than 1.5 million packets were routed under very light load, and more than 7 million under super heavy load.

The number of packets generated in a time slot, are assumed to be Poisson distributed, with arrival rates of 0.05, 0.1, 0.15, 0.2, 0.22, 0.5 per slot time. Thus the arrival process models asynchronous arrivals well. These values are chosen to represent the network load from very light, moderate, to super heavy. The value of 0.22 is chosen because this network model has a similar parameterization as that used in [5], where the arrival rate of 0.22 is about equal to the network saturation throughput. The packets generated at the node are assumed to be statistically independent of each other, and are lost if blocked, with no retransmission.

One of the essential goals of this simulation is to find the effect of adding input buffers at each node, and to determine the right buffer size to satisfy the blocking probability constraint. For each arrival rate, 13 different buffer size ranging from 0 to 20 (packets) are examined. Since the MSN network operates synchronously, but the local traffic is generated asynchronously, each packet arriving at the input buffer will suffer from a "synchronization delay". To simplify this problem, all packets arriving during a time slot are assumed to have a "birth time" of the previous time slot, e.g. packets generated during time t1 to t2 (t2-t1=one slot), are assumed to be one slot old at t2. Under this assumption, the synchronization delay will be higher than the actual value, but other statistics will not change.

The destination node of each packet is assumed to be chosen equally likely among all nodes and to be independent between successive packets, excluding the source node.

3 Routing Strategy

As mentioned earlier when packets are routed at a node it can encounter one of the six different routing events, with one of them involving a deflection. A routing strategy must be used which minimizes the impact of such conflicts.

Four different routing strategies are implemented in the simulation, namely Standard deflection, Age deflection, Conflict deflection and Predictive deflection. When there is no deflection situation, packets are routed according to their preferred route, independent of the routing strategy used. But when deflections become necessary, these four different routing strategies react differently. These strategies are chosen for study because they yielded the best performance of those studied in [5].

In *standard deflection*, the conflicting packets are routed randomly, so each packet has the same chance of getting the preferred link. This is the simplest strategy of all. It was used by Maxemchuck in the original MSN [2, 3].

Age deflection routes conflicting packets according to the packet age (defined to be the time that the packet has spent in the network but excluding the input buffer). The older the packet is, the higher the priority. If two conflicting packets are of the same age, the node chooses randomly among the two.

Conflict deflection is similar to age routing, except, instead of using the age as a priority indicator, the number of deflections a packet has experienced is used as the indicator. Higher priority is given to packets which have experienced more deflections.

Predictive deflection routes conflicting packet according to "predicted" source-destination distance. This distance is initialized at first as the minimum distance between source and destination, but every time packet experiences a deflection, this value is increased by 4. The conflicting packet with a higher value is sent over the preferred link.

4 Results and Discussion

The data gathered from this simulation can be characterized as those related to the network (excluding input buffer), and those related to the input buffer. Since different input buffer sizes will not change the relative network performances of different routing strategies, the discussion will focus on data acquired for an input buffer size of 5 (packets). Also an arrival rate (λ) of 0.25 and above results in a saturated network with an unacceptably large blocking probability. Thus for $\lambda > 0.22$, only data acquired for $\lambda = 0.5$ will be included, as a reference, in the discussion.

4.1 Network Delay

The network delay is defined to be the time packet spends, traversing the network from source to destination, excluding the time spent in the input buffer. This delay depends on two factors, one is the source-destination distance, the other is the number of deflections encountered. For obvious reasons, this delay should be as small as possible, and if possible the variance should also be small, so that packet delay will not vary much. This prevents the occurrence of extremely old packets, and makes the reordering of packet, if necessary, at the destination easier.

Since source-destination distance is fixed for any given source-destination pair, the only factor that would lengthen the network delay is the number of deflections. A single deflection due to a conflict can increase the distance by as much as four. For an 8x8 MSN under uniform destination distribution assumption, the average distance is only 5.02, so one deflection could increase the network delay by 80%. Thus it appears that the conflict routing strategy would yield the best delay distribution. This would be true if there are many deflections, and the probability of having more than one deflection is high, because in this way it is more likely that two packets will have distinct priorities. But as can be seen from Table 2, even under super heavy load, the average number of deflections is only about one, and percentile of more than one deflection is less than 29% for super heavy load, and less than 2% for very light load. Note also that to get to half (52% here) of the other nodes from any node, there is no preferred output link [2, 3]. Concluding from the above facts, conflict routing strategy might not be necessarily the best strategy. In fact age deflection is more likely to result in two packets in a node having distinct ages. Also note that the packet age implicitly contains conflict information. The result of the simulation proves this intuition.

The problem with using packet age as a priority indicator is that, this value is the packet's current age, not the "life-span" of the packet. On the other hand, Predictive deflection depends only on the packet's life-span, regardless of the time it entered the network. Thus long delay packets tend to be less likely to occur. One problem with our implementation of Predictive strategy is that, the 4 added to each deflection, only approximately represents the overhead of that particular deflection, thus making the algorithm less effective.

From Table 1 and Fig. 2, the Age and Predictive deflection strategies yield better performance, with Conflict deflection falling not far behind. Age deflection always has the smallest mean delay, but the difference is less than 5%. As for delay variance, both Age and Predictive deflection are relatively close to each other, but Conflict deflection can get p to about 21% worse than those two. This can be seen from the heavier tail for Conflict deflection in Fig. 2. Standard deflection has the poorest performance of all the strategies, as should be expected. Note that this ranking of performance does not take into account the header overhead to implement priorities.

4.2 Throughput and Utilization

Table 3 lists the throughput and link utilization of different deflection strategies and different arrival rates. As with previous work [5], the throughput saturates around 0.22. At heavier loads, Age deflection yields higher throughput. At light loads, all strategies performs approximately the same. This higher throughput is another consequence of lower average packet age. Standard deflection, while having the lowest throughput, has the highest link utilization under heavy loads. This implies that packets spend more time in the network, as can be seen from the network delay statistics. Note, from Little's law,

$$\overline{Network\ Delay} = \frac{2 \times \overline{Link\ Utilization}}{\overline{Throughput}}$$

A maximum throughput of 0.22 seems to be low, but considering the speed the network is able to operate, even 0.22 could mean a high local traffic rate, e.g. if the network operates at 100 Mbps, 0.22 corresponds to 22 Mbps of throughput. This is already 14 times the operating speed of T1 channel. Also note that this maximum throughput corresponds to a network with uniform traffic. In actuality, this is unlikely to be the case. If only a few nodes are heavily loaded, and all others are lightly loaded, then for the heavily loaded node, the actual throughput is not limited to 0.22. This result is discussed in section 5.2.

4.3 Input Buffer Statistics

There are a number of different statistics related to input buffer, that are important to the design of a network.

Shown in Table 4 are the input buffer delay statistics. As mentioned earlier, due to the assumptions, the delay acquired in the simulation is going to be larger than the actual value. However the difference is within one time slot and for most practical purposes this will be appreciable.

From the statistics, one should be concerned with not only with average delay but also with the delay density tails (Fig. 3). Note from Table 1 and 4, that the maximum delay (with age routing and $\lambda = 0.2$) for the network is 26, while for the input buffer it is 45. Thus the input buffer delay might be much larger than network delay in some cases. Thus even if achieving the desired blocking probability by using large input buffers is possible, the excessive buffer delay might be far from satisfactory.

For the input buffer delay, again Age deflection performs the best in every respect. This difference is more evident as the traffic load increases, especially near saturation. When $\lambda = 0.22$ the improvement of mean delay is at least 21% and the improvement of variance of delay is at least 29% over other strategies.

To include both the network delay and input buffer delay, the total delay experienced by a packet while traversing through buffer and network is also calculated. If delay is a great concern for the network designer, then it is this delay that has to be carefully examined. Table 5 shows the statistics. Fig. 4 shows the total delay density.

Table 7 thru 10 lists probability of blocking for various arrival rates and buffer sizes. When there is no input buffer, even under light load, there is nearly a 10% of chance a packet is blocked and this blocking grows as the load increases. A blocking probability as high as this are unacceptable in most cases. But as can be seen from the Tables, even with a buffer size of one packet, this probability drops dramatically. As is known from queuing theory, as long as one doesn't overload the buffer, by increasing the buffer size this probability can be made as low as desired. For this network model, it is apparent that even under moderately heavy load (e.g. $\lambda = 0.2$), a large buffer size is necessary to keep blocking probability at an acceptable level.

The tables presented here can be used as a reference, to determine the proper buffer size, when designing a 8x8 MSN network.

5 Variations

Variations of the routing strategy and/or input buffer mechanism can be made to try to further enhance the performance.

5.1 Combined Strategies

One of the variations is to use the number of deflections as a tie breaker in the age deflection strategy (Agecon deflection strategy). Simulation results show very little difference in performance (Table 11). The density and distribution curve are almost identical (Fig. 5). Considering the increased complexity and overhead, there seems little justification for this variation.

Another variation is the opposite of the above case, that is use age as a tie breaker in conflict deflection (Conage deflection strategy). This time simulation results show a small improvement over the pure conflict deflection strategy, but again one that falls short of the performance of the pure age deflection strategy (Fig. 6).

The third variation is a combination of Predictive and Age routing strategies. Here the Predictive deflection strategy is used to resolve conflicts, and packet age is used as the tie breaker (Adaage deflection strategy). This strategy also does not result in a significant improvement. A comparison of the network delay of the previous 3 variations is shown on Fig. 7.

5.2 Multiple Inputs from Buffer

If one allows two packets waiting in the input buffer to enter the network, when both of the packet buffers in that node are not occupied, this will enhance the throughput and lower the blocking probability. But from the simulation data, the probability of both packet buffers being available (Age routing) drops from ≈ 0.7 when $\lambda = 0.05$ to ≈ 0.015 when $\lambda = 0.2$. This means that this scheme is not likely to show much significant improvement for moderate to high arrival rates because most packet buffers will be occupied and there will be little chance for an input buffer to insert two packets into a node. However we have assumed uniform traffic throughout the network. If most of the nodes are under light load, and only few nodes are heavily loaded, then this scheme could make a significant difference.

A network with 3 nodes, located at (0,3), (4,1), (4,5), with node (0,0) being the upper-left corner node and (7,7) being the lower-right corner node, generating traffic at the rate of $\lambda = 0.9$, and all other nodes generating at

$\lambda = 0.1$, was simulated. Result shows the throughput at the heavily loaded nodes goes up by about 8.5%, from ≈ 0.81 *to* ≈ 0.89, and the probability of blocking drops more than 80%, from ≈ 0.096 *to* ≈ 0.016, while the penalty of increasing mean network delay is merely 1%. This performance boost decreases as the number of heavily loaded nodes increases. Since it possesses no packet overhead, and not much complexity in nodal structure, it appears attractive to use this structure for nodes with potentially heavy traffic.

6 Conclusions

An 8x8 Manhattan Street Network with input buffers was simulated, over a wide range of different parameters. The resulting data is valuable to the design of such a network. The most important result is in showing the necessity of using input buffers to improve network performance and what the input buffer size is needed to achieve a certain blocking probability.

From the simulation, it is found that Age deflection always gives a better performance in almost every respect, while Conflict and Predictive are not far behind. Standard deflection, although the simplest to implement has a much poorer performance.

We can also conclude that, under heavy loads, the input buffer delay might be excessive, and that the blocking probability could be much too high to be acceptable.

References

[1] F. Borgonovo and E. Cadorin, "HR^4-NET : A hierarchical random-routing reliable and reconfigurable network for metropolitan area," *Pro-*

ceedings of IEEE Infocom'87, pp. 320–326, 1987.

[2] N. F. Maxemchuk, "The manhattan street network," *Proceedings of IEEE Globecom'85*, pp. 255–261, 1985.

[3] N. F. Maxemchuk, "Regular mesh topologies in local and metropolitan area networks," *AT&T Technical Journal*, Vol. 64, No. 7, 1659–1685, Sept. 1985.

[4] G. E. Myers and M. El Zarki, "TAC-A triangular-arranged network," *Proceedings of IEEE Infocom'90*, pp. 481–487, 1990.

[5] T.G. Robertazzi and A.A. Lazar, "Deflection strategies for the manhattan street network," *Proceedings of the IEEE International Conference on Communication*, Denver CO, June 1991. A revised version appears in SUNY at Stony Brook College of Engineering and Applied Science Technical Report #603, April 24, 1991, available from T. G. Robertazzi.

λ	variable	Stadard	Age	Conflict	Predictive
0.05	Mean	5.480169	5.472759	5.482042	5.481855
	Variance	5.893224	4.889231	5.482283	4.722838
	Max	33	18	21	17
0.1	Mean	6.070309	6.025990	6.070733	6.074573
	Variance	9.600031	6.592798	7.704816	6.363454
	Max	50	20	23	22
0.15	Mean	6.882693	6.756026	6.871827	6.905921
	Variance	16.132692	8.906854	10.597297	8.771917
	Max	58	22	27	25
0.2	Mean	8.164686	7.796441	8.118445	8.178784
	Variance	29.772621	12.242683	14.903189	12.530347
	Max	94	26	27	27
0.22	Mean	8.654623	8.240180	8.580522	8.648850
	Variance	36.035183	13.672448	16.484674	13.929312
	Max	107	27	29	27
0.5	Mean	9.044296	8.688122	8.978495	9.026494
	Variance	41.532497	15.140995	17.848732	15.053483
	Max	106	28	33	29

Table 1: Network Delay (Buffer size 5)

λ	variable	Stadard	Age	Conflict	Predictive
0.05	Mean	0.116443	0.113907	0.116368	0.116372
	Variance	0.132661	0.110426	0.107832	0.111479
	% > 1 deflection	1.17	0.46	0.25	0.43
	Max	6	3	3	3
0.1	Mean	0.263371	0.252520	0.263283	0.264999
	Variance	0.345050	0.238391	0.228635	0.243858
	% > 1 deflection	4.54	2.33	1.68	2.33
	Max	11	4	4	4
0.15	Mean	0.466657	0.435183	0.464257	0.472490
	Variance	0.723456	0.400841	0.383461	0.417427
	% > 1 deflection	10.33	6.79	6.25	7.47
	Max	13	5	5	5
0.2	Mean	0.786944	0.695541	0.775687	0.790669
	Variance	1.525846	0.623886	0.614954	0.667678
	% > 1 deflection	19.38	16.01	18.21	19.46
	Max	22	5	5	5
0.22	Mean	0.909630	0.806063	0.891334	0.908183
	Variance	1.897459	0.717543	0.700007	0.757249
	% > 1 deflection	22.55	20.59	23.54	24.66
	Max	26	5	6	6
0.5	Mean	1.006897	0.918358	0.990687	1.002589
	Variance	2.225226	0.813188	0.775422	0.829996
	% > 1 deflection	24.91	25.42	28.30	28.93
	Max	25	6	6	6

Table 2: Deflections (Buffer size 5)

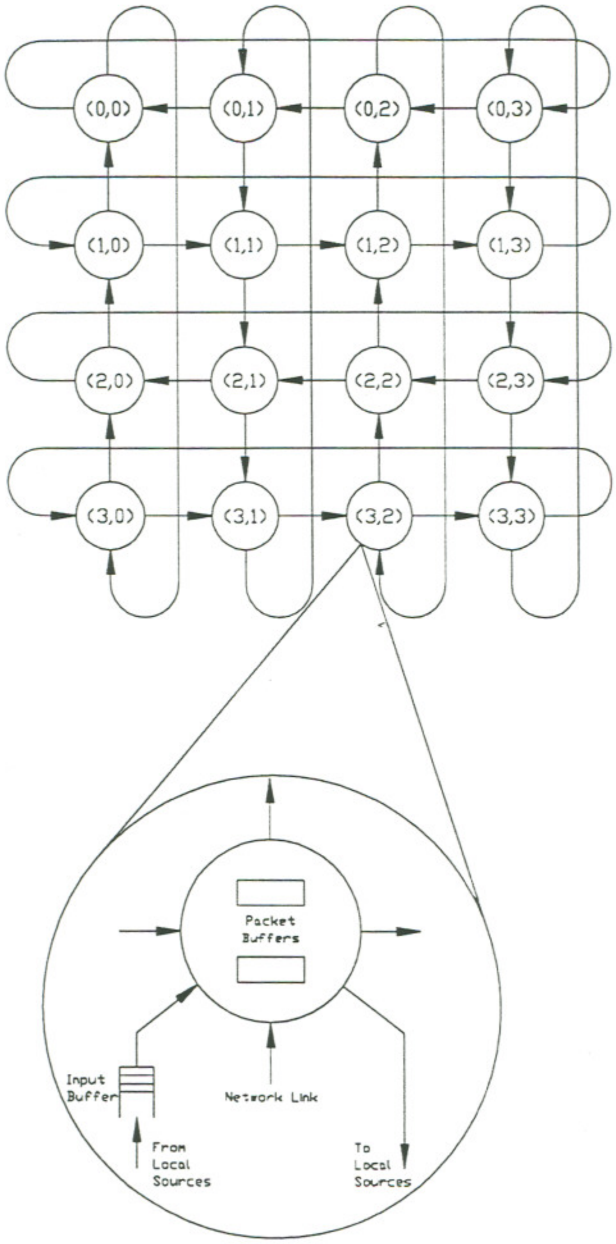

Figure 1: An example of MSN and its node structure used in the simulation

λ	variable	Stadard	Age	Conflict	Predictive
0.05	Throughput	0.050089	0.050092	0.050078	0.050078
	Utilization	0.137248	0.137073	0.137264	0.137261
0.1	Throughput	0.099911	0.099956	0.099953	0.099939
	Utilization	0.303246	0.301167	0.303397	0.303544
0.15	Throughput	0.150028	0.150070	0.150029	0.150047
	Utilization	0.516302	0.506941	0.515487	0.518107
0.2	Throughput	0.198804	0.199277	0.198886	0.198836
	Utilization	0.811594	0.776828	0.807327	0.813121
0.22	Throughput	0.211289	0.214873	0.211963	0.211251
	Utilization	0.914318	0.885298	0.909382	0.913545
0.5	Throughput	0.219465	0.228288	0.221032	0.219880
	Utilization	0.992460	0.991699	0.992270	0.992378

Table 3: Throughput and Utilization (Buffer size 5)

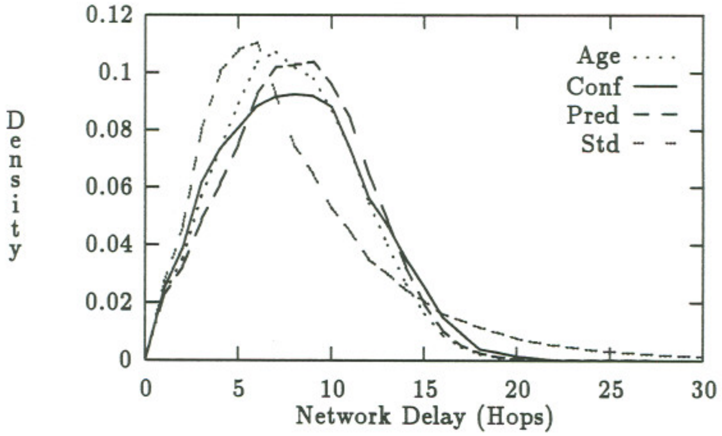

Figure 2: Network delay of Standard, Conflict, Age and Predictive deflection strategy (buffer size=5, *lambda* = 0.2)

λ	variable	Stadard	Age	Conflict	Predictive
0.05	Mean	1.040431	1.040418	1.040146	1.040143
	Variance	0.041616	0.041586	0.041334	0.041357
	Max	5	5	5	5
0.1	Mean	1.137922	1.136043	1.138186	1.138347
	Variance	0.156854	0.154497	0.157367	0.157246
	Max	8	8	9	8
0.15	Mean	1.429247	1.408784	1.428483	1.433445
	Variance	0.655070	0.606522	0.652736	0.662427
	Max	17	15	18	17
0.2	Mean	3.379389	2.855462	3.310000	3.407402
	Variance	9.601339	6.319295	9.136863	9.771719
	Max	49	43	47	52
0.22	Mean	6.550269	5.208319	6.305882	6.540613
	Variance	33.192097	22.219179	31.241028	33.353951
	Max	72	63	76	72
0.5	Mean	20.111782	19.153788	19.931610	20.057301
	Variance	91.441757	84.873627	90.800804	91.682289
	Max	104	104	110	104

Table 4: Input Buffer Delay (Buffer size 5)

λ	variable	Stadard	Age	Conflict	Predictive
0.05	Mean	6.520599	6.513177	6.522189	6.521998
	Variance	5.934351	4.930926	5.523100	4.765042
	Max	34	19	22	18
0.1	Mean	7.208231	7.162034	7.208919	7.212921
	Variance	9.761969	6.750627	7.866966	6.523924
	Max	51	24	24	23
0.15	Mean	8.311937	8.164808	8.300310	8.339368
	Variance	16.831324	9.543645	11.291045	9.476191
	Max	60	27	29	29
0.2	Mean	11.544096	10.651903	11.428446	11.586188
	Variance	39.675812	18.776346	24.328972	22.600655
	Max	96	56	61	60
0.22	Mean	15.204889	13.448498	14.886402	15.189462
	Variance	69.499924	36.099545	47.969143	47.505905
	Max	112	72	89	82
0.5	Mean	29.156069	27.841921	28.910105	29.083805
	Variance	132.958176	100.060333	108.642334	106.766487
	Max	153	118	117	115

Table 5: Total Delay (Buffer size 5)

λ	variable	Stadard	Age	Conflict	Predictive
0.05	Mean	0.052114	0.052117	0.052088	0.052089
	Variance	0.052190	0.052198	0.052159	0.052165
0.1	Mean	0.113690	0.113554	0.113766	0.113765
	Variance	0.115045	0.114900	0.115107	0.115134
0.15	Mean	0.214427	0.211417	0.214314	0.215084
	Variance	0.228323	0.224300	0.228181	0.229118
0.2	Mean	0.671838	0.569027	0.658318	0.677519
	Variance	0.971483	0.780228	0.945273	0.980572
0.22	Mean	1.383999	1.119131	1.336630	1.381730
	Variance	2.134059	1.740271	2.072463	2.137222
0.5	Mean	4.413864	4.372610	4.405570	4.410249
	Variance	0.867137	0.941387	0.884607	0.876174

Table 6: Input buffer length (Buffer size 5)

Buffer Size	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.15$	$\lambda = 0.2$	$\lambda = 0.22$	$\lambda = 0.5$
0	0.036174	0.096158	0.175845	0.261764	0.295022	0.603921
1	0.025203	0.053676	0.093593	0.155366	0.187847	0.573671
2	0.000455	0.002538	0.010900	0.053801	0.094312	0.563529
3	0.000009	0.000111	0.001279	0.023647	0.063803	0.561722
4	0.000001	0.000004	0.000148	0.011232	0.048689	0.561467
5	0.000000	0.000001	0.000018	0.005448	0.039455	0.561187
6	0.000000	0.000000	0.000002	0.002696	0.033308	0.560960
7	0.000000	0.000000	0.000000	0.001355	0.028750	0.560836
8	0.000000	0.000000	0.000000	0.000645	0.025658	0.560990
9	0.000000	0.000000	0.000000	0.000322	0.023046	0.561061
10	0.000000	0.000000	0.000000	0.000158	0.020896	0.561043
15	0.000000	0.000000	0.000000	0.000007	0.014628	0.560858
20	0.000000	0.000000	0.000000	0.000001	0.010990	0.560843

Table 7: Probability of Blocking (Standard)

Buffer Size	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.15$	$\lambda = 0.2$	$\lambda = 0.22$	$\lambda = 0.5$
0	0.035925	0.095496	0.173714	0.257478	0.289798	0.594566
1	0.025200	0.053622	0.092871	0.151641	0.181765	0.559871
2	0.000459	0.002523	0.010514	0.047605	0.082189	0.547103
3	0.000008	0.000110	0.001151	0.017949	0.049174	0.544583
4	0.000000	0.000005	0.000136	0.007175	0.032854	0.543621
5	0.000000	0.000000	0.000014	0.002939	0.023098	0.543533
6	0.000000	0.000000	0.000001	0.001196	0.017002	0.543471
7	0.000000	0.000000	0.000000	0.000502	0.012555	0.543261
8	0.000000	0.000000	0.000000	0.000203	0.009446	0.543419
9	0.000000	0.000000	0.000000	0.000084	0.007271	0.543315
10	0.000000	0.000000	0.000000	0.000035	0.005568	0.542989
15	0.000000	0.000000	0.000000	0.000001	0.001386	0.543469
20	0.000000	0.000000	0.000000	0.000000	0.000394	0.543119

Table 8: Probability of Blocking (Age)

Buffer Size	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.15$	$\lambda = 0.2$	$\lambda = 0.22$	$\lambda = 0.5$
0	0.036118	0.096113	0.175888	0.261277	0.294719	0.602897
1	0.025166	0.053761	0.093539	0.154900	0.187412	0.571429
2	0.000442	0.002559	0.010863	0.053088	0.092523	0.560790
3	0.000009	0.000116	0.001259	0.022891	0.061683	0.558591
4	0.000000	0.000007	0.000139	0.010531	0.045967	0.558251
5	0.000000	0.000000	0.000019	0.005094	0.036434	0.558042
6	0.000000	0.000000	0.000002	0.002332	0.030160	0.557918
7	0.000000	0.000000	0.000000	0.001179	0.025772	0.557932
8	0.000000	0.000000	0.000000	0.000508	0.022141	0.557795
9	0.000000	0.000000	0.000000	0.000255	0.019518	0.557787
10	0.000000	0.000000	0.000000	0.000127	0.017122	0.557839
15	0.000000	0.000000	0.000000	0.000003	0.010854	0.557779
20	0.000000	0.000000	0.000000	0.000000	0.007060	0.557736

Table 9: Probability of Blocking (Conflict)

Buffer Size	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.15$	$\lambda = 0.2$	$\lambda = 0.22$	$\lambda = 0.5$
0	0.036089	0.096201	0.176155	0.262357	0.296020	0.604417
1	0.025152	0.053754	0.093621	0.155948	0.188614	0.573466
2	0.000453	0.002551	0.010983	0.054490	0.094922	0.563177
3	0.000008	0.000119	0.001299	0.024157	0.064221	0.561123
4	0.000000	0.000006	0.000142	0.011523	0.048733	0.560528
5	0.000000	0.000000	0.000017	0.005519	0.039641	0.560398
6	0.000000	0.000000	0.000004	0.002801	0.033448	0.560337
7	0.000000	0.000000	0.000000	0.001359	0.028863	0.560152
8	0.000000	0.000000	0.000000	0.000668	0.025655	0.560203
9	0.000000	0.000000	0.000000	0.000328	0.022701	0.560309
10	0.000000	0.000000	0.000000	0.000173	0.020317	0.560149
15	0.000000	0.000000	0.000000	0.000007	0.014116	0.560393
20	0.000000	0.000000	0.000000	0.000000	0.010592	0.560430

Table 10: Probability of Blocking (Predictive)

Parameter	$\lambda = 0.1$	$\lambda = 0.22$	$\lambda = 0.5$
mean network delay	0.05%	0.31%	0.27%
variance network delay	0.02%	0.16%	0.30%
mean deflection	0.26%	0.78%	0.61%
variance deflection	0.38%	1.17%	0.90%
throughput	$\approx 0\%$	0.09%	0.26%
utilization	0.04%	0.22%	$\approx 0\%$
Prob. of Blocking	$\approx 0\%$	5.09%	0.22%

Table 11: Percentile difference of Age-Deflect deflection from Age deflection strategy

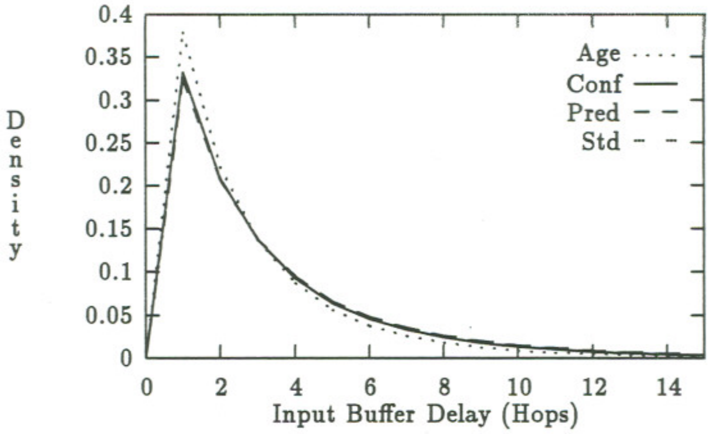

Figure 3: Input buffer delay of Standard, Conflict, Age and Predictive deflection strategy (buffer size=5, *lambda* = 0.2)

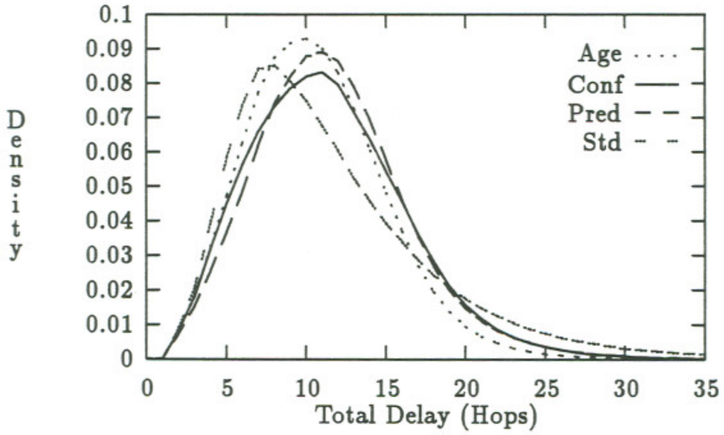

Figure 4: Total delay of Standard, Conflict,Age and Predictive deflection strategy (buffer size=5, *lambda* = 0.2)

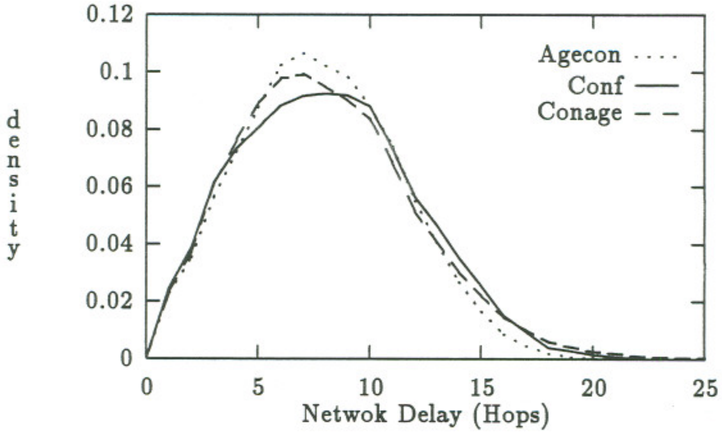

Figure 5: Network delay of Agecon, Conflict and Conage deflection strategy (buffer size=5, *lambda* = 0.2)

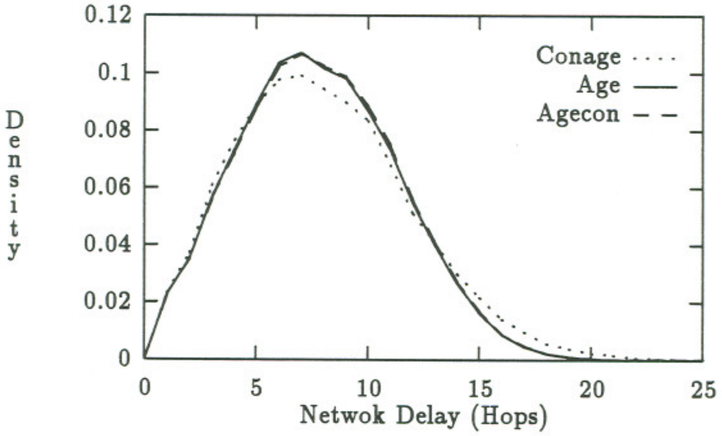

Figure 6: Network delay of Conage, Age and Agecon deflection strategy (buffer size=5, *lambda* = 0.2)

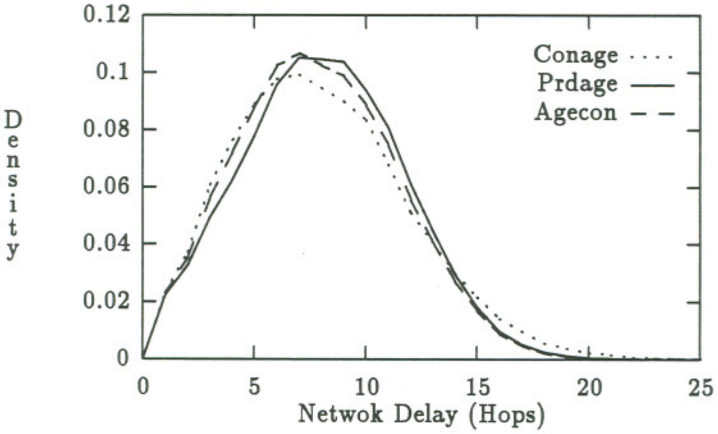

Figure 7: Network delay of Conage, Agecon and Adaage deflection strategy (buffer size=5, *lambda* = 0.2)