Int. Conf. on Computer and Information Technology (ICCIT2001)
Dhaka University, 28-29 December 2001

A NEW ADAPTIVE HYBRID ROUTING STRATEGY FOR DATAGRAM SERVICE

Mohammad Shahidul Hasan, Mosaddek Hossain Kamal, M. Lutfar Rahman
Network and Algorithm Research Lab
Department of Computer Science, University of Dhaka, Dhaka 1000, Bangladesh
Shasan@ssl-idt.net, shahidul_hasan@yahoo.com

Abstract

In this paper, a new bounded-hop-count routing strategy for datagram service has been developed. The hybrid routing technique for datagram is obtained by combining store-and-forward and deflection routing. It shows good adaptability to change in the network and does not require any specific measures of the network dynamics to be computed. Under the technique for datagram, packets are routed on a suboptimal route towards the destination instead of discarding them or wasting time in buffer because of congestion, link failure etc. This strategy is very much suitable for networks with limited or no buffer space at all, especially optical networks. Two algorithms are proposed to select a suboptimal or deflected route. Simulations are performed for both the proposed strategies which shows better results in terms of packet delivery, discarding, average delay when compared with the simple store-and-forward routing.

Keywords: Store-and-forward routing, deflection routing

1. INTRODUCTION

Routing is defined as the network function that computes a path through the network from a source node to a given destination node such that the path chosen optimizes some criteria. Therefore, routing messages between source and destination through a network is a crucial task that directly affects the performance of the network. Example of criteria can be minimization of overall delay, the use of expensive transmission bearers, maximization of network utilization, throughput etc. If routing decisions are made based on inaccurate information stored in routing tables, retransmissions caused by higher layer end-to-end transport protocols generally occur as a result of unexpected delay or lost packets. This places additional load on the network further consuming transmission resources [1]. In all-optical networks, messages are transmitted from a source to the destination without converting it from optical to electrical form to get the maximum benefit of the optical transmission [2]. Therefore, nodes in an optical network generally have limited buffer or no buffer at all. Routing must have the capability to resolve the contention when two or more packets select the same preferred link.

2. PRELIMINARIES

2.1 Store And Forward Routing

In *store-and-forward* packet switching network, buffers are used to temporarily store the packets that can not be immediately forwarded. Nearly all WANs (Wide Area Network) except those using satellites, have store and forward subnets [3].

2.2 Deflection Routing

On the other hand, *deflection* or *Hot-potato* [4] routing algorithm is a packet routing algorithm, in which nodes are unable to buffer packets in transit: any packet that arrives at a node, other than its destination, must immediately be forwarded to another node. The node must decide what to do with the packets that can not be relayed on the preferred links immediately. Such a packet can be buffered until the link becomes available in case of store-and-forward approach or it can be relayed on another available link along a sub-optimal path to the destination and the packet is said to be deflected [5]. This scheme has been used in parallel machines as well as high-speed communication networks of regular topology like hypercube, mesh etc. Deflection routing algorithms are well suited for optical networks because it is difficult to buffer optical messages.

3. PREVIOUS WORKS

The first deflection routing was proposed by Baran [6]. Borodin and Hopcroft suggested an algorithm for hypercube [7], Prager [8] showed that the *Borodin-Hopcroft algorithm* terminates in n steps on 2^n nodes hypercube for a special class of permutations. Several subsequent works considered permutation routing and randomized algorithms were proposed for d-dimensional meshes (or tori). The goal of earlier works was to present a simple algorithms for hypercubes and meshes which routes k packets with any combination of origins and destinations, in $d_{max} + 2*(k-1)$ steps, where d_{max} is the maximal source to destination distance. The permutations algorithms of [9] and later are based on sorting. In a sense, they simulate *store-and-forward* routing algorithms in a deflection manner.

A bounded-hop-count deflection scheme was proposed in "*A bounded-hop-count deflection scheme for Manhattan-street network (MSN)*" [10]. The proposed routing scheme was intended for bi-directional networks – four incoming and four outgoing links per switch. Under light and moderate loads, their proposed scheme performed no worse than the other local schemes. In particular, when there is no contention, packets propagate along the shortest route to their destinations.

A greedy deflection routing algorithm for *2*-dimensional $n \times n$ mesh was presented in *" Randomized Greedy Hot-Potato Routing"* [11]. For any permutation problem or random destination problem, it ensures that each packet reaches its destination in asymptotically optimal expected $O(n)$ steps and all packets reach their destinations in $O(n \ln n)$ steps with high probability, an improvement over previously known deterministic upper bound of $O(n^2)$ for greedy algorithm.

A lot of research has been done on deflection routing for regular topologies like hypercube, mesh etc. and on optical networks that do not have the ability to store packets temporarily at nodes.

4. THE NEW HYBRID DATAGRAM PACKET ROUTING STRATEGY

The proposed hybrid routing strategy is a combination of store-and-forward and deflection routing that uses the knowledge of current utilization or load of links. Under normal operation of the network, this routing strategy behaves just like pure store-and-forward scheme and a datagram packet follows the optimal or the shortest path from a source to the destination. The processing of a packet is the same as the processing in store-and-forward routing in a congestion free environment. However, the proposed hybrid strategy shows adaptation in abnormal situations, for instance, link failure, congestion etc. A datagram packet is deflected only when a link on the optimal path is congested or failed. After deflection, the packet may follow the optimal path from the next hop to the destination.

On arrival of a packet at a router, routing table look-up is performed depending on the destination of the packet to determine the outgoing link L on which it will be forwarded. When the output queue of the link L is full and a new packet arrives that chooses link L then in pure store-and-forward routing, the packet is discarded generally. In this proposal, instead of discarding, the packet is given a chance to go to the network again on a deflected link, if possible.

Each datagram packet contains a hop count field in the header. Whenever a router forwards the packet it decrements the field value. The initial value is set by the source. In the proposed strategy, a datagram packet is allowed to be deflected as long as its hop count field value is valid or non-zero and there is no restriction on the number of deflections that a datagram packet can experience.

In this proposal, routing decision is made independently for each datagram packet. This allows simple routing algorithm and switch design. Selection of a deflected link at a node is crucial that affects the performance of the overall routing strategy. In the proposed strategy, the load of a link is used to reflect the utilization of the link. For choosing a deflected link, the load of links must be estimated. Since, in a high-speed (Gbps) network, the processing time of a deflected packet must be very small, only local information is used to determine the load of a link. As the estimation of load of a link, simply the current queue length of the link is used to keep the proposed strategy simple. So, it does not require any information to exchange among the nodes for load computation of links. Under this assumption, a link is underloaded if the number of packets in the queue of that link is less than the half of the maximum length of the queue. Such a link can be chosen as a deflection link. In the following sections, two algorithms are proposed to select a deflection link. Any one of these two can be used.

4.1 Proposal 1: Deflection On Queue Length

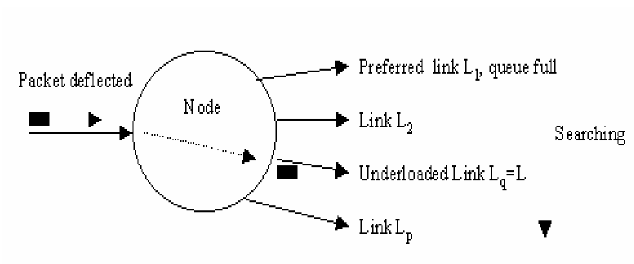

Figure 1: Deflection On Queue Length.

The algorithm for routing a packet in Deflection on queue length scheme

1. Consult routing table
2. Let L_1 be the preferred link on the best route
3. If the queue for L_1 is full then
4.   If there is a underloaded link L then
5.   Put the packet in the queue of link L
6.   Else
7.   Discard the packet
8.   End of if structure
9. Else
10.   Put the packet in the queue of the link L_1
11. End of if structure

4.2 Proposal 2: Deflection On Queue Length With Suboptimal Next Hop

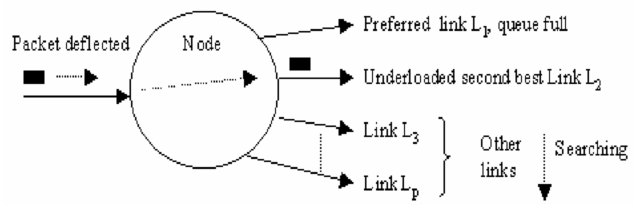

Figure 2: Deflection On Queue Length With Suboptimal Next Hop.

The algorithm for routing a packet in Deflection on queue length with suboptimal next hop scheme

1. Consult routing table
2. Let L_1 be the preferred link on the optimal route
3. If the queue for L_1 is full then
4.   Consult routing table
5.   Let L_2 be the link on the suboptimal route
6.   If the link L_2 is not underloaded then
7.   If there is an underloaded link L then
8.   Put the packet in the queue of link L
9.   Else
10.   Discard the packet
11.   End of if structure
12.   Else
13.   Put the packet in the queue of the link L_2
14.   End of if structure
15. Else
16.   Put the packet in the queue of the link L_1
17. End of if structure

5. ANALYSIS

5.1 Complexity Of Deflection On Queue Length

Let a node has average connectivity degree *p*. If a packet can not be forwarded on the preferred link, a scanning is performed on other *p-1* available links. Finding an underloaded link may require checking the load of all *p-1* links at best. So, the complexity of this algorithm is *O(p)*.

5.2 Complexity Of Deflection On Queue Length With Suboptimal Next Hop

In the best case, when a packet can be forwarded on the suboptimal link, it requires checking only the status of the suboptimal link. However, if the suboptimal link is not underloaded, a searching is initiated among the other links. For average connectivity *p*, it will take at most *p-2* checking to find an underloaded link. So, it has complexity *O(p)*.

6. SIMULATION AND RESULT

Simulations were performed for the proposed strategies using widely used C programming language. Results of pure *store-and-forward*, *deflection on queue length*, *deflection on queue length with suboptimal next hop* strategies are observed in terms of delivered, discarded packets and average delay for datagram service. The number of delivered packets is observed for various hop count N and N/2 where N is the number of nodes in the network. 20000 to 100000 packets were generated with random destinations. Number of delivered and discarded packet, average delays are observed for different (1, 5, 10) queue length of nodes in the network.

6.1 Packet Delivery

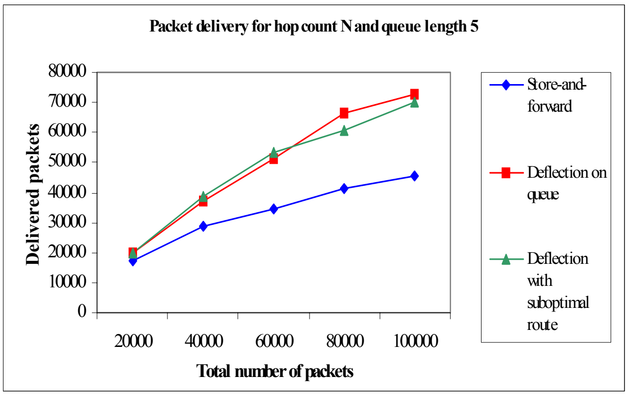

Figure 3: Packet Delivery In Store-And-Forward, Deflection On Queue Length And Deflection On Queue Length With Suboptimal Route For Hop Count N And Queue Length 5.

Figure 3 shows that more packets are delivered under the proposed strategies than the store-and-forward routing.

6.2 Packet Discarding

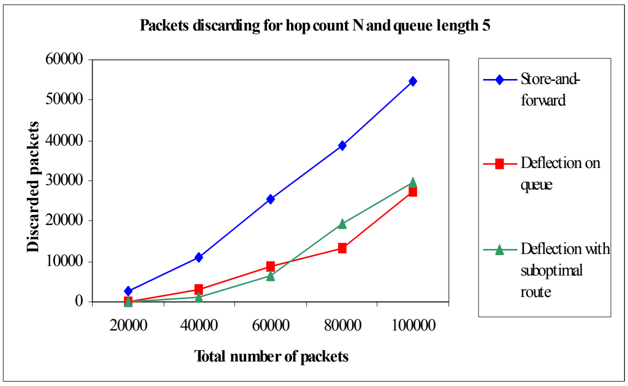

Figure 4: Packet Discarding In Store-And-Forward, Deflection On Queue Length And Deflection On Queue Length With Suboptimal Route For Hop Count N And Queue Length 5.

From figure 4, it can be understood that, smaller number of packets is discarded under deflection routing schemes.

6.3 Average Delay

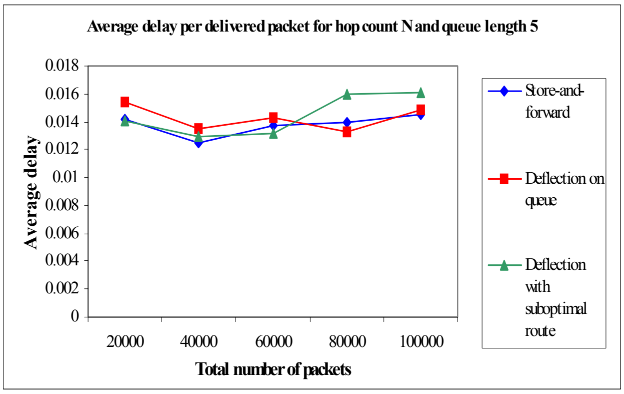

Figure 5: Average Delay In Store-And-Forward, Deflection On Queue Length And Deflection On Queue Length With Suboptimal Route For Hop Count N And Queue Length 5.

Figure 5 shows the average delays under three strategies are very close to each other. However, deflection with suboptimal next hop produces better performance under heavy load.

For the datagram proposals, both *deflection on queue length* and *deflection on queue length with suboptimal route* give higher packet delivery and lower number of discarded packets than *store-and-forward*. Moreover, these two strategies show better performance as the queue size increases.

7. CONCLUSION

In the proposal for datagram of this paper, packets are forwarded on a suboptimal route in case of congestion on the preferred link. Moreover, packets that are generally discarded because of output buffer overflow of the preferred link, are allowed to remain in the network by forwarding on an alternate link. So, the loss probability is minimized. However, because of deflection, packets may not travel the shortest paths. Therefore, the average delay for a packet may increase a little. But, in case of high connectivity and high transmission rate network this delay can be kept in tolerable range. It can be perceived from the simulation results that, the proposed hybrid routing strategies can deliver more packets than store-and-forward for the same time duration. The number of discarded packets under the proposed routing techniques is lower than store-and-forward even for very small amount of buffer. The performances of *deflection on queue length* and *deflection on queue length with suboptimal route* in terms of packet delivery, discard and average delay are close. However, deflection on queue length with suboptimal route requires more memory to store the next hop.

REFERENCE

1. Peter John Shoubridge, Adaptive Strategies for Routing in Dynamic Networks., Bachelor of Engineering in Electronic Engineering, School of Physics and Electronic Systems Engineering, Faculty of Information Technology, University of South Australia, Thesis for Degree of Doctor of Philosophy, December, 1996
2. Thierry Chick, Pierre Fraigniaud, "An extended comparison of slotted and unslotted deflection routing.", Research Report N° RR97-07, Ecole Normale Superieure de Lyon, 46 Allee d'Italie, 69364 Lyon Cedex 07, France, March 20, 1997.
3. Andrew S. Tanenbaum, Computer Networks, Third Edition, Prentice-Hall of India, 1998
4. Emmanouel A. Varvarigos, Member, IEEE, and Jonathan P. Lang, Student Member, IEEE, "A Virtual Circuit Deflection Protocol", *IEEE ACM Transactions on Networking*, Vol-7, No-3, June, 1999.
5. Cesur Baransel, Wlodek Doboseiwicz, Pawel Gburzynski, "Routing in Multihop Packet Switching Network: Gb/s Challenge.", *IEEE Network*, Vol. 9, No. 7, May-June, 95
6. P. Baran., "On distributed Communications Networks", *IEEE Transactions on Communications*, 12:1 9, 1964.
7. A. Borodin and J. E. Hopcroft, "Routing, Merging, and Sorting on Parallel Models of Computation", *Journals of Computer and System Science*, 30:130-145, 1985.
8. R. Prager, "An algorithm for routing in hypercube networks", Master's thesis, 1986
9. I. Newman and A. Schuster, "Hot-potato algorithms for permutation routing", Technical Report LPCR #9201, CS dept, Technion, November, 1992
10. Wlodek Dobosiwicz and Pawel Gburzynski, "A bounded-hop-count deflection scheme for Manhattan-street network", December 22, 1994
11. Costas Busch, Maurice Herlihy, and Roger Wattenhofer, "Randomized Greedy Hot-Potato Routing", Computer Science Department, Brown University, Providence, RI 02912.