Adaptive Loop-free Routing Technique in Store-and-forward Communication Networks

Tohru Nishigaki,* Chiaki Ikeda* and Hiroshi Ikegaya*

Adaptive routing techniques for store-and-forward communication networks sometimes cause a looping or ping-pong effect. This effect leads to degradation of rapidity and assurance of message delivery.

A new Adaptive Loop-free Routing technique is developed employing an Adaptive Loop-free Routing Algorithm. The technique is loop-free and adapts to changes in traffic loads and network topology.

Experimental simulation reveals high performance in typical network configurations with various traffic loads. Performance is compared with representative adaptive routing techniques.

This technique is expected to increase the efficiency of operational store-and-forward communication networks.

1. Introduction

Messages are transmitted in store-and-forward communication networks by being relayed along some source-destination paths determined by routing algorithms. For the many algorithms available, adaptive routing algorithms are considered the most promising for use in operational networks [1]. Adaptive routing algorithms are adaptive to variations in both traffic loads and network topology. However, a critical drawback is the possibility of messages being trapped in loops, thereby degrading both performance and delivery assurance. While a Last M Nodes Visited (LMNV) algorithm has been proposed to avoid this effect [2], it cannot prevent loops of at least M+2 nodes.

A new adaptive routing algorithm is proposed which is free from looping. In the algorithm, nodes are classified into several sets according to the minimum number of intermediate nodes to the destination node. The minimization of the time for a message to travel from the current set to the next set is the objective. The algorithm never permits transmission to farther sets or messages to remain in the same set for extended periods of time. Thus, looping is avoided when this algorithm is applied.

2. The Routing Problem

To begin with, messages are considered present at node A and addressed to some node Z. Each node belongs to one of the sets defined below.

$$\Omega(Z, n) \triangleq \left\{ \text{nodes} \,\middle|\, \begin{array}{l} \text{the shortest path length** from} \\ \text{the node to } Z \text{ is } n \text{ hops} \end{array} \right\} \quad (1)$$

Accordingly, any node becomes a member of as many different sets as the number of destination nodes in the network.

Assuming that node A belongs to $\Omega(Z, n)$, its subset $\omega(Z, n, A)$ is defined as follows:

$$\omega(Z, n, A) \triangleq \left\{ \text{nodes} \,\middle|\, \begin{array}{l} \text{belong to } \Omega(Z, n) \text{ and} \\ \text{connected to } A \end{array} \right\} \quad (2)$$

Unless h is $n+1$ or n or $n-1$,

$$\omega(Z, h, A) = \{\phi\} \quad (3)$$

Next, the time interval between $\tau(n-1)$ and $\tau(n)$ is denoted as $d(n)$, where $\tau(n-1)$ and $\tau(n)$ represent the instant a message reaches a node belonging to $\Omega(Z, n-1)$ and $\Omega(Z, n)$ for the first time, respectively. The delay D from node A to node Z is given as follows:

$$D = d(n) + d(n-1) + \cdots + d(m) + \cdots + d(1) \quad (4)$$

It is apparent that no algorithm always gives the optimal route minimizing D. This is because $d(m)$'s are unknown variables depending upon future states. The error of $d(n)$ is, however, comparatively small; and the routing algorithm presented is employed to achieve $d(n) \to \min$ instead of $D \to \min$.

It is noteworthy that looping can occur if messages pass through nodes in $\Omega(Z, n)$ three or more times, or travel to $\Omega(Z, k)(k > n)$.

3. Adaptive Loop-free Routing Algorithm

An Adaptive Loop-free Routing Algorithm (referred to as ALF hereafter) is presented in relation to messages at node A and destined for node Z. To begin with, $Q(i)$ is defined as follows:

$$Q(i) \triangleq \min_{S \in \omega(Z, n+i, A)} \{\text{delay to go from node } A \text{ to node } S\} \quad (i = -1, 0, 1) \quad (5)$$

where the delay is given by the output line queue length for node S. Either $Q(-1)$ or $Q(0)$ will be the delay to the

---

*Systems Development Laboratory, Hitachi, Ltd., 5030, Totsuka-machi, Totsuka-ku, Yokohama 244, Japan.

**Here, all lines are assumed to have the same capacity.

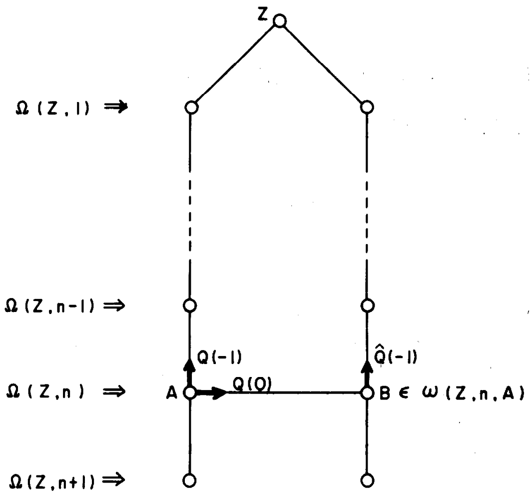

Fig. 1 Routing based on eq. (6); min $\{Q(-1), Q(0)+\hat{Q}(-1)\}$.

next node where the message should be sent. The former corresponds to the direct path and the latter, the roundabout path. The selection is made via eq. (6) and depicted in Fig. 1.

$$\min\{Q(-1), Q(0)+\hat{Q}(-1)\} \quad (6)$$

$Q(-1)$ and $Q(0)+\hat{Q}(-1)$ represent the travel time to $\Omega(Z, n-1)$ via the direct path and the roundabout path, respectively. $\hat{Q}(-1)$ indicates an estimated value of $Q(-1; B)$ which is given by replacing A with B in eq. (5) $(i=-1)$, where B is the neighbor node corresponding to $Q(0)$. In case, however, $\omega(Z, n, A)=\{\phi\}$, the direct path is chosen immediately.

It should be noted that the direct path is selected for a message arriving via a roundabout path. This is because a roundabout path is assumed to be followed by a direct path in eq. (6). As a result, the number of nodes passed through from a node in $\Omega(Z, n)$ to Z is $2n$ at most, and looping is completely avoided.

The message taking a roundabout path carries current $Q(-1)$ to the neighbor node. Utilizing that, the estimation value $\hat{Q}(-1)$ is updated at regular intervals Δt as follows:

$$\begin{cases}\hat{Q}_j(-1)=(1-\theta)\cdot Q_j(-1; B)+\theta\cdot\hat{Q}_{j-1}(-1) \\ \hat{Q}_0(-1)=0 \quad (\theta: \text{const.}, 0<\theta<1)(j=1, 2, 3\cdots\cdot)\end{cases} \quad (7)$$

where $\hat{Q}_j(-1)$ and $\hat{Q}_{j-1}(-1)$ represent the estimation values at time t_j and $t_{j-1}(=t_j-\Delta t)$, respectively. $Q_j(-1; B)$ represents the $Q(-1; B)$ brought by the last roundabout message from node B during $t_{j-1}\sim t_j$. Only when no roundabout message has arrived for more than Δt, is node A informed of $Q_j(-1; B)$ by a single message. Even in such cases, it can be omitted when $Q(-1; B)=0$; that is, node A updates $\hat{Q}_j(-1)$ regarding $Q_j(-1; B)$ as zero if no information has come during $t_{j-1}\sim t_j$. This simply aims at reducing the traffic overhead due to $Q(-1)$ exchanges. The overhead is much smaller than that for exchanging update vectors reflecting overall network traffic conditions.

The algorithm described above is expected to attain satisfactorily small $d(n)$, although it may not always be the minimal one.

Another advantage of the algorithm is its adaptability to variations in network topology resulting from line or node failures. Knowledge of the whole network topology is unnecessary; and each node need only know which sets it, and its neighbor nodes, belong to. They can be learned dynamically whenever the network topology changes. The learning algorithm is the same as the one being employed in the ARPA network to detect "disconnections" in the network [3]. It differs only in that it is not executed periodically but only when any change occurs.

4. Simulation Experiments

The performance of ALF was compared with those of representative and actually implemented adaptive routing algorithms like Shortest Queue plus Bias ($SQ+B$) [4] and Periodic Updating (PU) [3]. Three network configurations shown in Fig. 2 were investigated on a detailed simulator.

In all these configurations, nodes are assumed to be connected by full duplex lines each with a capacity of 4800 bits/sec. Messages with a fixed length of 500 bits enter them in a Poisson manner.

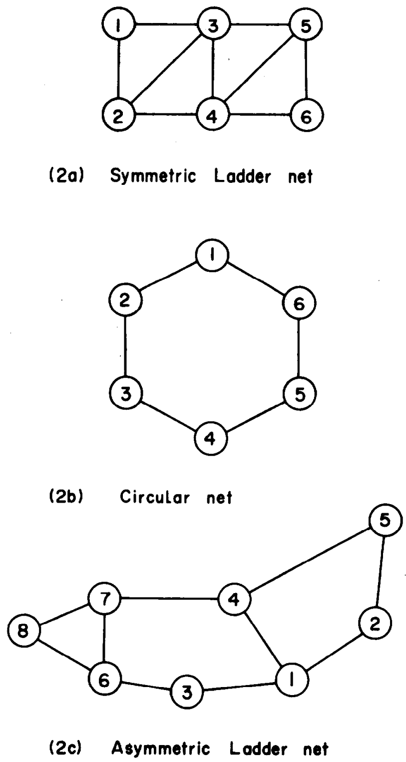

Fig. 2 Network configurations investigated.

During performance comparison, various adjusted parameters were used in routing algorithms such as D_p bias, Periodic Update Rate (treated by Fultz [1]), θ and Δt. Thus, it was possible to achieve the best message delay for each configuration under moderate loads.

The "average message delay" from the time a message enters the source until it arrives at the destination is shown in Figs. 3–5 as a function of load (=total amount of message input rate in K bits/sec) for the three routing algorithms. Here, loads are assumed to be "uniform"; that is, all entries except diagonal ones (=0) are in the 6×6 traffic matrix. Therefore, the message traffic between each source-destination node pair is the same. Additional information is presented in Tables 1–3. "Average hops" represents the average number of lines encountered in a source-destination path. "Average time in queue" represents the average time interval when a message is waiting or being transmitted on an output line queue. The increased "average hops" accompanied by a correspondingly decreased "average time in queue" indicates an increased effectiveness of the roundabout path. However, increases in both suggest looping or inadequate roundabout paths.

Tables 1–3 also show performance comparison results where the network is partially loaded. As for a Symmetric Ladder net, the situation is such that the (4, 3) entry in the traffic matrix corresponding to the traffic from node

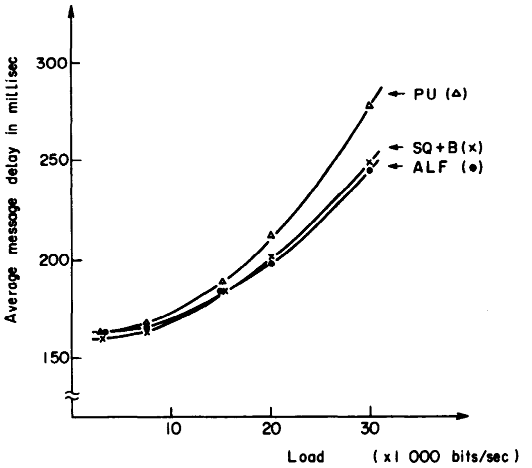

Fig. 3 Average message delay as a function of uniform loads for a Symmetric Ladder net.

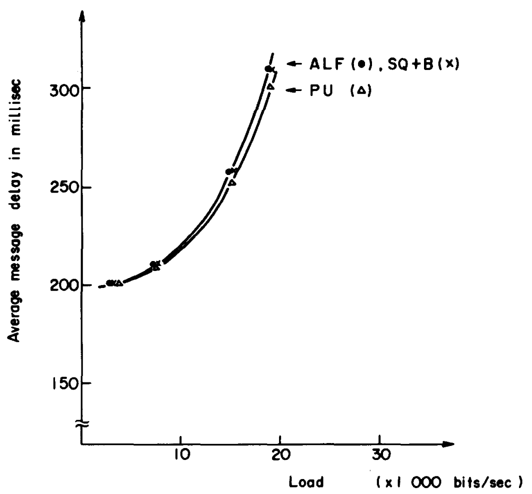

Fig. 4 Average message delay as a function of uniform loads for a circuler net.

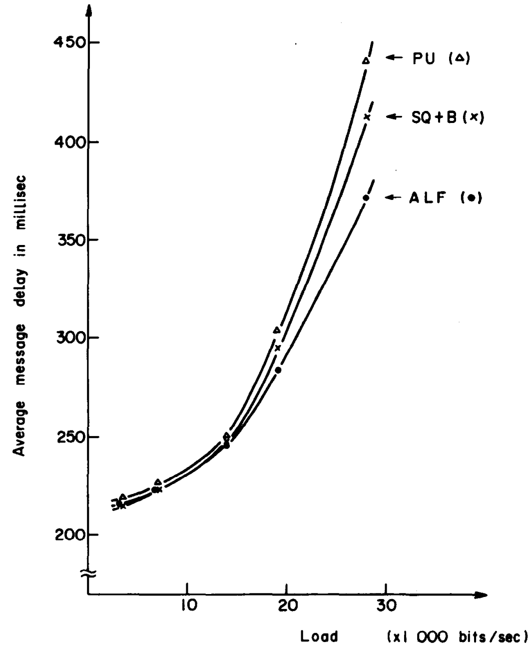

Fig. 5 Average message delay as a function of uniform loads for an Asymmetric Ladder net.

Table 1 Performance comparison of the three algorithms for a Symmetric Ladder net with uniform/partial load.

(1a) Uniform load ; 30 000 bits/sec			
	SQ+B	PU	ALF
average hops	1.75	1.89	1.82
average time in queue (millisec)	142	147	135
(1b) Partial load ; traffic from 4 to 3/others = 16/1			
	SQ+B	PU	ALF
overall average message delay (millisec)	159	168	154
overall average hops	1.35	1.44	1.41
average message delay(millisec) (traffic from 4 to 3)	145	151	136
average hops (traffic from 4 to 3)	1.10	1.20	1.21
average time in 4→3 queue (millisec)	135	130	119

Table 2 Performance comparison of the three algorithms for a Circular net with uniform/partial load.

(2a) Uniform load ; 18 750 bits/sec			
	SQ+B	PU	ALF
average hops	1.80	1.81	1.80
average time in queue (millisec)	172	166	173
(2b) Partial load ; traffic from 2 to 1 / others = 16/1			
	SQ+B	PU	ALF
overall average message delay (millisec)	200	181	189
overall average hops	1.57	1.55	1.54
average message delay (millisec) (traffic from 4 to 1)	367	341	355
average hops (traffic from 4 to 1)	3.06	3.01	3.00
average time in 2→1 queue (millisec)	161	144	156

Table 3 Performance comparison of the three algorithms for an Asymmetric Ladder net with uniform/partial load.

(3a) Uniform load; 28 000 bits/sec			
	SQ+B	PU	ALF
average hops	2.18	2.39	2.15
average time in queue (millisec)	189	184	173
(3b) Partial load ; traffic from {6 to 3, 7 to 4} / others = 16/1			
	SQ+B	PU	ALF
overall average message delay (millisec)	266	241	213
overall average hops	1.98	1.81	1.67
average message delay (millisec) (traffic from 8 to 2)	586	544	499
average hops (traffic from 8 to 2)	4.55	4.37	4.20
average time in 6→3 queue (millisec)	181	164	158
average time in 7→4 queue (millisec)	184	177	162

4 to node 3 is 4 messages/sec; and all others, except diagonal entries (=0), are 0.25 messages/sec. Likewise, the traffic from node 2 to node 1 in a Circular net and the traffic from node 6 to node 3 and node 7 to node 4 in an Asymmetric Ladder net is 16 times larger than other traffic of 0.25 messages/sec, when they are partially loaded.

Algorithms to minimize $d(n)$ instead of D like ALF are expected to be effective on networks where nodes in the same Ω are generally connected with each other. This efficiency of ALF as compared with PU is exemplified in Fig. 3, when a Symmetric Ladder net is heavily loaded. In addition, Tables 1 and 3 give related information on a heavily loaded Symmetric/Asymmetric Ladder net, where ALF is smaller than PU in both "average hops" and "average time in queue", suggesting the advantages of a loop-free technique.

On the contrary, PU's advantage of utilizing global traffic conditions yields a significant difference on a partially loaded Circular net, as shown in Table 2. PU's superiority to ALF and SQ+B in "overall average message delay" could be explained by the fact that any node can know the congestion of line 2→1 for PU; whereas, only node 2 is cognizant for ALF or SQ+B. It is noteworthy, however, that the "average hops" of the traffic from node 4 to node 1 indicates the appearance of loops for SQ+B. Additionally, the figures in Table 3, related to the traffic from node 8 to node 2 on a partially loaded Asymmetric Ladder net, also exemplify the loops for SQ+B. In this situation, the increase in "average hops" as well as "average time in the 6→3 and 7→4 queues" indicates loops such as 6→7→6 may frequently appear. ALF is superior to SQ+B in this regard and is able to avoid loops, thus allowing moderate performance realization.

5. Conclusion

Adaptive loop-free routing technique (ALF) attains assured and rapid message delivery as the result of looping path avoidance. Simulation results revealed that it performs most effectively in symmetric/asymmetric ladder network configurations. This is particularly true when compared with such representative routing techniques as Shortest Queue plus Bias or Periodic Updating.

The ALF technique has been implemented on the HITAC-10II minicomputer (16 K words) for further investigation. It is a strong candidate for use in operational store-and-forward communication networks.

References

1. FULTZ, G. L. Adaptive Routing Techniques for Message Switching Computer-Communication Networks, *UCLA-ENG-7252, UCLA*, (July 1972).
2. PICKHOLTZ, R. L. et al. Improvements in Routing in a Packet-Switched Network, *Proc. of ICCC'74*, T 3230, (1974), 249–252.
3. HEART, F. E. et al. The Interface Message Processor for the ARPA Computer Network, *Proc. of SJCC*, (1970), 551–567.
4. ITO, T. et al. Packet Switching Subnet of JIPNET System, *Proc. of 2nd UJCC*, (1975), 459–463.