DETERMINISTIC AND ADAPTIVE ROUTING POLICIES

IN PACKET-SWITCHED COMPUTER NETWORKS*

Mario Gerla
Network Analysis Corporation
Glen Cove, New York 11542

Summary

Efficient routing policies are required for both design and operation of distributed, packet-switched computer networks. In the network design process, deterministic policies, which are time invariant, easy to analyze and simple to represent, are generally used. For the routing of packets in a real network, on the other hand, adaptive policies, which can adjust to traffic fluctuations and component failures are implemented. Therefore, a computer network is typically optimized with respect to a deterministic policy and then operated with an adaptive policy.

In the first part of this paper the two policies are compared using analytical and simulation methods, and it is shown that, in uniform traffic situations, the flow patterns generated are very similar. This result justifies the use of deterministic policy in the design phase.

In the second part of the paper, the inadequacy of traditional adaptive policies in handling particular traffic situations is pointed out. As a solution, a new centralized adaptive policy is proposed, which combines the positive features of both deterministic and adaptive routing. The feasibility of such centralized policy implementation on a real network, and its efficiency with respect to the traditional implementation is discussed.

1. Introduction

In a packet-switched network, messages are segmented into packets of appropriate size and each packet is transmitted independently. Packet transmission is performed using the store and forward (S/F) technique: a packet traveling from source n_i to destination n_j is "stored" in queue at any intermediate node n_k , while awaiting transmission, and is sent "forward" to n_l , the next node in the route from n_i to n_j , when channel (k,l) permits.¹

The topology of a distributed, packet-switched computer network generally provides, for reliability and channel cost economy reasons^{2,3}, two or more alternate routes between each source-destination pair. A routing problem, therefore, must be solved, i.e., routes must be selected so as to accommodate the external traffic requirements and optimize network performance (usually measured in terms of packet or message delay). The solution to the routing problem is a routing

policy, i.e., a set of rules assigned to each node in the network. Following such rules, and according to the destination of the incoming packet, the node selects the output channel for packet forwarding.

Routing policies can be divided into two main classes: deterministic and adaptive. A deterministic policy is time invariant and provides the optimal routing for a network in steady state (no load fluctuations, no failures). Such a policy is easy to evaluate analytically and therefore is used for all design problems requiring repeated routing computation. An adaptive policy is time varying and can adjust to load fluctuations and failures; for such a reason it is more attractive for real network operations than a deterministic policy. On the other hand, its performance can be evaluated only by means of time consuming computer simulations⁴; therefore, it is rarely used in the network design phase.

For the validity of a network design obtained with the deterministic policy assumption, it is important to compare the performance (throughput, delay, etc.) of deterministic and adaptive policies. In the first part of this paper, after a brief description of the routing policies presently used and a more general routing policy classification, analytical and simulation results are compared, and it is shown that the flow patterns generated by the two policies are very similar. Therefore, the deterministic assumption during network design is robust.

The real network experience has shown that traditional adaptive policies are inadequate in certain traffic situations, in particular when heavy traffic between two nodes requires the simultaneous use of alternate routes⁵. Consequently, various modifications and improvements have been proposed⁶. Here, in the last part of the paper, a "centralized" adaptive policy is proposed as a possible solution to the problem.

2. Present Routing Policies

Various types of deterministic and adaptive policies can be found in the S/F network literature^{1,4}. Here, we briefly outline the policies that are currently used for the design and the operation of the ARPA computer network, a S/F network that connects more than thirty computer sites in the United States^{7,8,9}.

2.1 Deterministic Policy

Several assumptions on network status and traffic characteristics are required in order to make the computation of the policy mathematically tractable¹. Among the assumptions,

*This work was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC 15-73-C-0135

we mention:

Poisson external arrivals at nodes, with steady rate,

packet length exponentially distributed,

infinite storage at nodes,

network components perfectly reliable.

Under such assumptions whose validity is discussed in Section 4, the average packet delay T can be expressed as a function of average flow rates in the channels, as follows¹:

$$T = \frac{1}{\gamma} \sum_{i=1}^{NA} \frac{f_i}{C_i - f_i} \quad (1)$$

where: NA = number of channels
 γ = total external input rate [packets/sec]
 f_i = average rate on channel i [bits/sec]
 C_i = capacity of channel i [bits/sec]

More detailed traffic models lead to more elaborate expressions of T ¹⁰: however, the conceptual difficulty of the routing problem remains the same.

The deterministic policy is defined as the policy that minimizes the delay T , for a given external requirement matrix. It can be shown that the deterministic routing problem is a convex multicommodity flow problem whose solution, if it exists, is unique. Efficient algorithms for both optimal² and suboptimal⁹ solutions are available. Typically, the routing policy for a 30 node network is obtained after 5 to 10 iterations of the routing algorithm, and requires 2 to 3 seconds CPU time on a large computer.

A deterministic policy can be described by a set of routing tables $RT^{(i)}$, $i = 1, \dots, N$, where N is the number of nodes in the network. The table for node i , $RT^{(i)}$, is an $N \times A_i$ matrix, where A_i is the set of nodes adjacent to node i . The entry $RT^{(i)}(k, \ell)$ represents the fraction of incoming traffic directed to destination k , which is transmitted to neighbor node ℓ . Clearly, for a given destination k :

$$0 \leq RT^{(i)}(k, \ell) \leq 1$$

$$\sum_{\ell \in A_i} RT^{(i)}(k, \ell) = 1 \quad (2)$$

Thus, packets with same destination k can be distributed among different routes. Routing tables corresponding to optimal deterministic policies have the very important property of not allowing loops: i.e., a packet which travels through the network according to routing table rules will never visit the same node twice².

The routing table representation makes the deterministic policy directly implement-

able on a real network. However, the unrealistic assumption of a network perfectly reliable has so far limited the use of deterministic policies to network analysis and design.

2.2 Adaptive Policy

The adaptive policy here described contains the basic concepts of the policy presently used in the ARPA computer network¹¹.

Each node i ($i=1, \dots, N$) has an $N \times A_i$ matrix $DT^{(i)}$, called Delay Table whose entry $DT^{(i)}(k, \ell)$ is the estimated minimal delay from node i to destination k , if ℓ is chosen as the next node in the route to k . From $DT^{(i)}$, an N -dimensional vector $MDV^{(i)}$ (the Minimum Delay Vector) is obtained as follows:

$$MDV^{(i)}(k) \triangleq \min_{\ell \in A_i} DT^{(i)}(k, \ell) \quad (3)$$

$MDV^{(i)}(k)$ represents the minimum estimated delay from i to k . The adaptive policy attempts to send packets on minimum delay routes; therefore, packets directed to k are sent to neighbor ℓ_s such that:

$$DT^{(i)}(k, \ell_s) = \min_{\ell \in A_i} DT^{(i)}(k, \ell) \quad (4)$$

Each node periodically updates its own delay table using the delays measured on its output lines and delay information received from neighbor nodes. In fact, every half second, each node transmits asynchronously to its neighbors the vector MDV . Upon reception of vectors $MDV^{(\ell)}$, $\ell \in A_i$, node i updates $DT^{(i)}$ as follows:

$$DT^{(i)}(k, \ell) \triangleq d(i, \ell) + MDV^{(\ell)}(k) \quad (5)$$

where $d(i, \ell)$ is the measured delay (queueing + transmission) on channel (i, ℓ) . Notice that, between subsequent updates, the adaptive policy uses only one route per destination (while a deterministic policy can use more than one route simultaneously).

The delay tables reflect network congestion and node or link failure. However, disconnected nodes cannot be detected rapidly enough from delay vectors. Consequently, additional information is circulated to help detect disconnections. Each node keeps an N -dimensional vector MHV (Minimum Hop Vector), whose entry $MHV(k)$ represents the minimum estimated number of line hops to node k . The Hop Vector is transmitted periodically to the neighbors. Upon reception of neighbors' Hop Vectors, node i updates $MHV^{(i)}$ as follows:

$$MHV^{(i)}(k) = 1 + \min_{\ell \in A_i} MHV^{(\ell)}(k) \quad (6)$$

The entries of the MHV 's so propagated converge to the minimum hop numbers. If an entry does not converge (more precisely, if an entry ever becomes larger than N), the corresponding destination is disconnected.

3. A General Routing Policy Classification

In order to properly define and compare routing policies, it is important to establish a general routing policy classification on the basis of:

(i) network status information (traffic load and topology) as a function of time	(a) time invariant (deterministic) (b) varies with time (adaptive)
(ii) locality of the information	(a) global network information available (global) (b) local, or incomplete information available (local)
(iii) routing computation	(a) routing computation performed by only one processor (centralized) (b) routing computation performed in parallel by each node in the network (distributed)
(iv) objective	(a) minimization of total average delay (system optimizing) (b) minimization of individual source-destination delays until competitive equilibrium is reached (user optimizing)

According to the above classification, the deterministic policy defined in Section 2 is in fact: deterministic, global, centralized, system optimizing. Similarly, the traditional adaptive policy is: adaptive, local, distributed, user optimizing. Another possible policy, developed later in this paper, is the "centralized adaptive policy", defined as follows: adaptive, global, centralized, system optimizing.

4. Deterministic and Adaptive Policy Performance

The performance of a routing policy is usually defined as the average packet delay T produced by the policy under a given (generally uniform) external traffic requirement. For a deterministic policy the delay T is evaluated analytically, as from Equation (1), while, for an adaptive policy, computer simulation is required. The comparison of the two policies has been carried out by many researchers, for different network configurations 1, 4, 12. Generally, a very good agreement between the respective delays was found. Figure 1 shows the results obtained by Fultz⁴

for a 19 node network example. The delay of a nearly optimum deterministic policy (referred to as "lower envelope") and the delay of a specific adaptive algorithm are compared, for a wide range of traffic load RE ($RE = 1$ corresponds to an external requirement of 1 Kbs/node pair). The difference between the two curves is less than 10%, almost up to saturation.

FIGURE 1.

Performance of the Adaptive Routing Algorithm

As mentioned in Section 3, deterministic and adaptive policy differ under many aspects. The above results provide a global comparison. In some cases a separate analysis of the individual aspects is also possible: for instance, the analytical expression for the deterministic delay is based upon several simplifying assumptions (infinite storage etc.) while the adaptive delay, obtained from simulation, reflects the real network and traffic characteristics. The validity of the assumptions can be easily tested by simulating the deterministic policy under real network conditions and comparing analytical and simulation delays. Figure 2 shows the two delays for a 19 node network, for a wide range of

traffic load RE^4 : the agreement is excellent.

FIGURE 2.

Performance of a 19 Node Network Under Deterministic Routing Policy

Another interesting aspect to investigate is the system optimizing nature of the deterministic policy, as compared to the user optimizing nature of the adaptive policy. In order to evaluate such an effect, a deterministic, user optimizing policy is considered and the delay compared to that of the system optimizing policy. The mathematical analysis of the user optimizing policy is easily carried out after recognizing that such a policy is equivalent to a system optimizing policy which minimizes the following objective function P^2 :

$$P = \sum_{i=1}^{NA} \int \frac{df_i}{C_i - f_i} = \log \prod_{i=1}^{NA} \left(\frac{1}{C_i - f_i} \right) \quad (7)$$

The deterministic algorithm applies also to the objective function in Equation (7); user optimizing flow pattern and delay can therefore be evaluated. The results based on a 21 node network show a difference of less than 1% between system and user optimizing delays²: again an excellent agreement!

The above results show that throughput and delay evaluated with the deterministic policy assumption are very close to those obtained with an adaptive policy in a real network. It is, therefore, reasonable to assume deterministic routing for analysis and design purposes.

There are, however, some very special traffic situations in which the adaptive routing performance is considerably worse than the deterministic prediction. One such situation corresponds to heavy traffic between two nodes which requires the simultaneous use of more than one route. Assuming that two disjoint routes are available, the adaptive policy described in Section 2 can select only one route between subsequent updates. Within each update interval, one of the routes is alternately filled up to saturation, thus producing congestion, looping and, ultimately, performance degradation. Such a degradation has been observed both in simulation experiments and network traffic measurements⁵; consequently, various modifications and improvements have been proposed⁶. In the next section a centralized adaptive policy is presented as an alternative solution to the problem. The policy should combine the ability to adjust to load fluctuations and failures of the traditional adaptive policy, and the simultaneous use of alternate routes, proper to the deterministic policy.

5. A Centralized Adaptive Policy

As mentioned in Section 3, a centralized adaptive policy is defined as an adaptive, global, centralized, system optimizing policy. The policy here proposed is essentially a deterministic policy which is periodically updated according to load fluctuations and network failures. Thus, each node has a deterministic routing table (see Section 2); a Network Routing Center (NRC) collects network information, computes routing policy corrections according to such information, and transmits routing update messages to all nodes. For additional failure protection, the nodes are also equipped with Minimum Hop Tables. Following is a description of possible specifications for a centralized adaptive policy implementation.

5.1 Basic Operations

Each node transmits to NRC asynchronously, every .5 seconds, the following information:

- number of packets transmitted on each output channel in the past .5 seconds.
- for each destination k , the number of packets directed to k , which were received from external sources (Host computers, terminals, etc.) in the past .5 seconds.

Using the information received from the nodes, a computer, available at the NRC site, evaluates channel and external input rates averaged over the past 10 seconds. Every 10 seconds the above data is fed to a routing program resident in core. The routing program can be the Flow Deviation algorithm¹³: in such a case, the program computes, on the basis of channel traffic and external requirement, for each node-destination pair, the new route on which a fraction of the traffic, say α ($0 \leq \alpha \leq 1$), must be "deviated", in order to improve network performance. At the end of the computation, which typically requires 100 to 300 msec., NRC delivers to each node a routing update message which contains:

- a vector of "next nodes", one per destination, to which a fraction α of the incoming traffic must be deviated.
- the parameter α .

Upon reception of the routing message, each node updates its routing table². Since the Flow Deviation algorithm is an optimal routing algorithm, the centralized adaptive policy converges to the deterministic policy in steady network conditions.

5.2 Line and Node Failures

In order to avoid congestion when a failure occurs in the 10-second interval between routing updates, a very simple local adaptive policy is provided as a backup. For example, the minimum hop number policy, described in Paragraph 2.2, could be used. During normal network operation (no failures) the min hop policy remains inoperative. As soon as a node experiences the failure of an output line or neighbor node, (1) it modifies its min hop numbers, according to the failures, (2) it transmits immediately the min hop table to the neighbors, (3) it switches from centralized adaptive to min hop mode of operation, i.e., it routes the packets along min hop routes. Gradually, all nodes switch to min hop policy, until the NRC learns about the failure. New tables are then computed, which account for the failure, and the centralized adaptive policy is gradually restored. During the transient period following the failure, min hop and centralized adaptive policy co-exist in the network. It can be easily seen that such a situation is logically acceptable, and does not create severe performance degradation.

5.3 NRC Failures

The reliability of NRC can be improved by providing a backup NRC at another network site. When one NRC goes down, the backup takes over. Furthermore, if network failures isolate the NRC's so that they belong to two disconnected components, each component can be controlled by the respective NRC. Finally, if both NRC's go down, or if a network component becomes disconnected from both NRC's, packet routing is accomplished by the min hop policy.

5.4 Centralized Versus Local Adaptive Policy

Following are a few points of comparison between centralized adaptive policy (CAP) and

local adaptive policy (LAP).

- Overhead traffic, due to transmission of network status and routing information, is approximately the same.
- Processor overhead, corresponding to routing and updating, is of the same order of magnitude for both systems (except that in LAP the routing computation is distributed among the nodes, while in CAP it is performed by NRC).
- In the transient period following each failure, CAP is less efficient than LAP. However, failures typically occur at the average rate of one every two hours, in a thirty node network. Therefore, the effect of failures on average CAP performance is not substantial.
- CAP has no loops and makes efficient use of alternate paths. LAP can generate loops and tends to use only one path at a time.
- CAP can perform a better and more selective flow control on the external input rates, than LAP, as it has the global knowledge of all external traffic requirements.

6. Conclusions

Characteristics and performance of two typical deterministic and adaptive routing policies have been compared. It has been recognized that the deterministic policy is more attractive for network analysis and design, while the adaptive one is more desirable for a real network implementation. As for the respective performances, it has been shown that, at steady state and with reasonably uniform external traffic requirements, the two policies can accommodate the same traffic load with approximately the same average packet delay. This fact justifies the use of deterministic policies during the design phase, and of adaptive policies during the operational phase.

The inefficiency of traditional adaptive policies in sending heavy source to destination traffic on more than one route has suggested, as a possible solution, the development of a "centralized" adaptive policy. This policy combines the steady state efficiency of a deterministic policy to the flexibility of a traditional adaptive policy. A description of the centralized policy and some specifications for an eventual implementation have been provided. The policy seems attractive: however, further investigation is required in order to determine under what network conditions (nature of the external traffic, component failure rates, nodal processing speed, etc.) such an implementation is more advantageous than the traditional adaptive policy implementation.

References

1. Kleinrock, L., Communication Nets: Stochastic Message Flow and Delay, McGraw-Hill, New York, 1964.
2. Gerla, M., The Design of Store-and-Forward Networks for Computer Communications,

School of Engineering and Applied Science,
University of California, Los Angeles.
PhD Dissertation, January 1973.

3. Frank, H. and W. Chou, "Topological Optimization of Computer Networks," IEEE Proceedings, November 1972, pp. 1385-1396.
4. Fultz, G. L., Adaptive Routing Techniques for Message Switching Computer-Communication Networks, School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7252, July 1972.
5. BBN Report #2161, August 1971.
6. Frank, H., R. E. Kahn, and L. Kleinrock, "Computer-Communication Network Design--Experience with Theory and Practice," AFIPS Conference Proceedings, 40:255-270, SJCC, Atlantic City, New Jersey, 1972.
7. Roberts, L. G., and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, 36:543-599, SJCC, Atlantic City, New Jersey, 1970.
8. Heart, F. E., R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The Interface Message Processor for the ARPA Computer Network," AFIPS Conference Proceedings, 36:551-567, SJCC, Atlantic City, New Jersey, 1970.
9. Chou, W. and H. Frank, "Routing Strategies for Computer Network Design," Symposium on Computer Communications Networks and Teletraffic, Polytechnic Institute of Brooklyn, April 4-6, 1972.
10. Kleinrock, L., "Analytic and Simulation Methods in Computer Network Design," AFIPS Conference Proceedings, 36:569-579, SJCC, Atlantic City, New Jersey, 1970.
11. BBN Report #1783, March 1969.
12. "Analysis and Optimization of Store-and-Forward Computer Networks," N.A.C. Fourth Semiannual Technical Report for the ARPA Project, Defense Documentation Center, December, 1971.
13. Fratta, L., M. Gerla, and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design," Networks, Vol. 3, No. 2, 1973, pp. 97-133.