論文

Flood Search Algorithm with MFDL Path in Circuit-Switched Networks

Young Chul Park* Sang Chul Lee** Chong Kwan Un*** *Regular Members*

회선 교환망에서 MFDL 경로를 이용한 Flood Search 알고리즘

正會員 朴英澈* 正會員 李相哲** 正會員 殷鍾官***

Abstract

Flood search algorithm is known to be an effective routing mechanism for tactical application, since it provides high degree of survivability and robustness. But, it is known that it has significant drawbacks with respect to the network efficiency [1]. We consider a tactical circuit-switched grid network with a maximum of four links and two priority classes of voice traffic. Using the minimum first-derivative length (MFDL) path, we improve the blocking probability performance of a circuit-switched network without increasing the call set-up time and processor loading of the algorithm.

要約

Flooding 탐색 알고리즘은 고도의 생존성과 신뢰성이 있으므로 전술응용을 위한 효과적 라우팅 메카니즘이라고 알려져 있다. 그러나 망 효율성 측면에서는 큰 결점을 갖고 있다.

본 논문에서는 음성트래픽에 대하여 최대 4개의 링크와 2개의 우선순위클래스를 갖는 전술 회선교환 격자망을 MFDL 경로기법을 사용하여 호설정시간 증가 및 알고리즘의 프로세서로딩없이 회선교환망의 블로킹 확률에 대한 성능개선 방법을 제안하였다.

Ⅰ. Introduction

Circuit-Switched networks for military tactical application exhibit properties inherently different from those of strategic or commericial systems. By its nature, the tactical network is highly mobile and is constantly in a state of change. Neither the subscribers nor the communications assets remain fixed for long periods of time. The

---

*國防科學研究所
Agency for Defence Development
**通信網研究所
Telecommunication Network Laboratories
***韓國科學技術院 電氣및 電子工學科
Dept. of Electrical and Electronics Eng., KAIST

論文番號：93－39

networks are frquently subject to stress situations such as dynamic traffic, damage, and jamming. In addition, tactical circuit-switched networks must accommodate numerous levels of precedence with a preemption capability for high-priority subscribers. The highest priority class of subscribers must be guaranteed no blockage given the existence of a physical path between them. For such a military network, the survivability and the robustness of the flood search algorithm (also called saturation routing or flooding algorithm) is highly desirable. To satisfy these military network requirements, the flooding algorithm is used for many military tactical communication networks [1]-[3]. This technique has a great advantage of not having to maintain any routing tables at the intermediate nodes. Flooding algorithm is also known to be an effective mechanism for routing and forwarding packets in packet switching networks. The **ARPANET**, the oldest of the packet switching networks, exchanges its routing information using a form of flooding [4]-[6]. Recently, several new architectures to promote high speed packet switching have been proposed [7]-[8]. The premise of these networks is that a great deal of the routing computation is done ahead of time so that when the actual data packets start traversing the network, intermmediate nodes do not have consult routing tables that would have slowed down their operation. It is known that the algorithm offers many potential advantages, but also gives several weaknesses [2]. For circuit-switched networks, it results in undesirable effect, particularly with respect to the network efficiency. An avalanche effect may occur, reducing the overall network efficiency [9]. In addition, in an equally loaded network the effect of high usage / overflow routing is lost and increases the probability of blocking on the shortest routes [10]. Recently, Miller et al. compared flooding algorithm with restricted flood search and hybrid routing methods in terms of the switching network's connectivity, throughput, grade of service, and routing delay [3]. It is concluded that the flood search gives the most reliable performance in all aspects under possible network outage conditions in the battlefield environment. These algorithms do not deal with optimization of performance such as the minimization of average end-to-end blocking probability (EEBP) in circuit-switched networks. In this paper, with the same or less level of call set-up time and processor loading of the flood search algorithm, we propose a modified flooding algorithm, thereby improving the blocking performance in the networks. Finding the optimal solution that minimizes the average (EEBP), P_B, with constraints on the link blocking probability is complicated and time consuming. Therefore, in the proposed approach, instead of calculating P_B, we compare the gradient of the objective function at each node, and choose the shortes path so that the link length is minimum, that is, determine the minimum first-derivative length (MFDL) path between the source node S and the destination node D. The link length here is defined to be the marginal increase of the objective function with respect to the marginal increase of the amount of traffic flow of the link.

In the next Section we introduce the modified flooding algorithm in detail and consider the swtich processor load and the call set-up time of the algorithm proposed. In Section 3, numerical results of the blocking performance for the algorithm are presented. Finally, we give conclusions in Section 4.

Ⅱ. Modeling and analysis

Let the nodes of an n-node network be represented by the integers 1,2,...,n and let a link from node i to node j be represented by (i, j). Each link has a fixed number of channels, each of which is able to carry one call at a time. A link is said to be busy if all of its channels are busy for transmission of calls. In general, there may be several paths between a (S,D) pair or commodity. Now we denote

D_{ij} : A monotonically increasing cost function.
F_{ij} : Total flow on link (i,j) (in calls/s).
x_p : Flow of path p (I calls/s).
w : A Source-destination (S,D) node pair, i.e.,
W_{ij} : A set of all (S,D) node pairs, w's
P_w : A set of all directed paths connecting source and destination nodes of (S,D) pair w.
r_w : Arrival rate of traffic entering the network at node i and destined for node j (in calls/s).

For each (S,D) pair w, the input traffic arrival process is assumed stationary with rate r_w. Assuming that D_{ij} is a function of the link flow F_{ij} only, an optimal routing problem can be written as

$$\textit{minimize} \sum_{(ij)} D_{ij}(F_{ij}) \tag{1a}$$

$$= \textit{minimize} \sum_{(ij)} D_{ij} \sum_{\substack{\textit{all path } p \\ \textit{containing}(ij)}} x_p \tag{1b}$$

$$\textit{subjet to} \sum_{p \in P_w} x_p = r_w \text{, for all } w \in W \tag{1c}$$

$$x_p \geq 0\text{, for } \textit{all } p \in P_w,\ w \in W \tag{1d}$$

where F_{ij}' on link (i,j) is the sum of all path flows traversing the link. Thus, the problem is formulated in terms of unknown path flows $x = \{x_p | p \in P_w,\ w \in W\}$.Optimal solutions of the routing problem (see (1)) may be characterized as done by Bertsekas and Gallager [11]. We assume that each D_{ij} is a differentiable function of F_{ij}'. Now, taking the partial derivative on D with respect to x_p, we have

$$\frac{\partial D(x)}{\partial x_p} = \sum_{\substack{\textit{all links}(i,j) \\ \textit{on path } p}} D'_{ij} \tag{2}$$

where the first-derivatives D'_{ij} are evaluated at the total flows corresponding to x. From (2) it is seen that $\partial D / \partial x_p$ is the length of path p when the length of each link (i,j) is taken to be the first-derivative length (FDL) of path p [6]. The minimum first-derivative length (MFDL) here is defined to be the minimum of FLD's.

Considering a call routing procedure of the modified flood search algorithm, there are two major functions as in the original flood search algorithm. One is a subscriber location function which searches the network for the called subscriber and the other is a route search function which searches the network for an available route to connect the calling to the called switches. In the original flooding algorithm, the route search need not have to be concerned with the probability of blocking through the switch matrices [1]; but in the modified flooding algorithm the FDL or the gradient of the objective function represented by blocking probabilities (see (4) and (5)) need to know the probability of blocking.

The network model used in this study is a circuit-switched grid network with a maximum of four links. All the links which have the same transmission bandwidth are capable of transmission in both directions. The common channel signaling (CCS) method is used between nodes, and the EUROCOM recommended message format is assumed [1]. The trunk group contains 30 speech channels with a 32 kbit/s common channel. A call is considered as a basic unit of circuit-switched traffic, and each call originating from S is destined to D. The call arrival process is assumed to be Poisson with rate λ_i on the ith link and the call holding time is exponentially distributed with mean $1/\mu$. The call connection and disconnection times are assumed to be negligible, and link blocking probabilities are assumed to be statistically independent. Blocked calls are also assumed to be cleared and do not return. The link blocking probability on the ith link is given by the Erlang B formula,

$$B(a_i, \mathrm{N}) = \frac{a_i^{\mathrm{N}} / \mathrm{N}!}{\sum_{j=0}^{\mathrm{N}} a_i^j / j!} \tag{3}$$

where a_i is the offered load to link i (in Erlangs) or the Erlang rate on link i, defined as $a_i = \lambda_i / \mu$, and N is the number of channels. Since there may exist more than one path between each (S,D) node commodity and a path may consist of more than one link, it is necessary to define a path blocking probability and an average EEBP for the traffic of one (S,D) node pair [12]. That is, the path blocking probability of the k-th path of (S, D) traffic, $P_{PB}(\mathrm{S,D})_k$, is given by

$$P_{PB}(\mathrm{S,D})_k = 1 - \prod_{i \in L(\mathrm{S,D})_k} (1 - B(a_i, \mathrm{N})) \tag{4}$$

where $L(\mathrm{S,D})_k$ is the set of all the links on the k-path of (S,D) traffic. With an assumption of small link blocking probabilities, we can get another expression of the path blocking probability by canceling the product terms of the B(a_i,N)'s in (4).

$$P_{PB}(\mathrm{S,D})_k = 1 - \sum_{i \in L(\mathrm{S,D})_k} B(a_i, \mathrm{N}) \tag{5}$$

Now, the average EEBP of (S,D) traffic, $P_B(\mathrm{S,D})_k$, is

$$P_B(\mathrm{S,D}) = \sum_{k=1}^{K(\mathrm{S,D})} q(\mathrm{S,D})_k \sum_{i \in L(\mathrm{S,D})_k} B(a_i, \mathrm{N}) \tag{6}$$

where $\sum_{k=1}^{K(\mathrm{S,D})} q(\mathrm{S,D})_k = 1$, $q(\mathrm{S,D})_k$ is the proportion of (S,D) traffic that uses the k-th path of (S,D), and $K(\mathrm{S,D})$ is the total number of paths for the (S,D) traffic. Again, we define the average EEBP, P_B, as the weighted sum of the EEBP of all the (S,D) node pairs, and we have

$$P_B = \sum_{j=1}^{M} \frac{\lambda(\mathrm{S}_j, \mathrm{D}_j)}{\gamma} P_B(\mathrm{S}_j, \mathrm{D}_j)$$

$$= \sum_{j=1}^{M} \frac{\lambda(\mathrm{S}_j, \mathrm{D}_j)}{\gamma} \sum_{k=1}^{K(\mathrm{S}_j, \mathrm{D}_j)} q(\mathrm{S}_j, \mathrm{D}_j)_k \sum_{i \in L(\mathrm{S}_j, \mathrm{D}_j)_k} B(a_i, \mathrm{N})$$

$$= \frac{1}{\gamma} \sum_{i=1}^{l} \lambda_i \, B(a_i, \mathrm{N})$$

$$= \frac{1}{\gamma / \mu} \sum_{i=1}^{l} a_i \, B(a_i, \mathrm{N}) \tag{7}$$

where $\gamma = \sum_{j=1}^{M} \lambda(\mathrm{S}_j, \mathrm{D}_j)$, M is the total number of (S,D) node pair, and l is the total number of links in the networks. Here, the term γ / μ is fixed for a given problem and does not play a part in the minimization. We hence will exclude it in (8). We assume that there are two classes of voice traffic, high-priority voice v_1 and low-priority voice v_2. High-priority ovice v_1 can preempt low-priority voice v_2. Letting a_{pi} be the offered load for circuit-switched voice of priority p, $p = 1,2$, then the high-and low-priority voice blocking probabilities are given as $B_{1i} = B(a_{1i}, \mathrm{N})$ and $B_{2i} = B(a_{1i} + a_{2i}, \mathrm{N})$, respectively. Now, we can express the objective functions, $f_1(\mathbf{a}_1)$ for v_1 and $f_2(\mathbf{a}_2)$ for v_2, respectively, that is,

$$P_{B1} \triangleq f_1(\mathbf{a}_1) = \sum_{i=1}^{l} a_{1i} B_{1i} \text{ for } \mathrm{v}_1 \tag{8}$$

$$P_{B2} \triangleq f_2(\mathbf{a}_2) = \sum_{i=1}^{l} (a_{1i} B_{1i} + a_{2i} B_{2i}) \text{ for } \mathrm{v}_2 \tag{9}$$

where $\mathbf{a}_1 = [a_{11}, a_{12}, \ldots, a_{1l}]$, and $\mathbf{a}_2 = [a_{21}, a_{22}, \ldots, a_{2l}]$, . Now, we can derive the gradient of the objective function with respect to the link flow as

$$\nabla f_1(\mathbf{a}_1) =$$

$$\frac{\partial f_1(\mathbf{a}_1)}{\partial a_{11}} \hat{\mathrm{e}}_1 + \frac{\partial f_1(\mathbf{a}_1)}{\partial a_{12}} \hat{\mathrm{e}}_2 + \cdots + \frac{\partial f_1(\mathbf{a}_1)}{\partial a_{1l}} \hat{\mathrm{e}}_l \tag{10}$$

$$\nabla f_2(\mathbf{a}_2) =$$

$$\frac{\partial f_2(\mathbf{a}_2)}{\partial a_{21}} \hat{\mathrm{e}}_1 + \frac{\partial f_2(\mathbf{a}_2)}{\partial a_{22}} \hat{\mathrm{e}}_2 + \cdots + \frac{\partial f_2(\mathbf{a}_2)}{\partial a_{2l}} \hat{\mathrm{e}}_l \tag{11}$$

where $\hat{\mathrm{e}}_l$ is a $1 \times l$ vector whose elements are all zero except for its lth element being equal to one. Thus, at link l, the first-derivative length (FDL) for v_1 is given by $\partial f_1(\mathbf{a}_1) / \partial a_{1i} = a_{1i} \cdot \partial B_{1i} / \partial a_{1i} + B_{1i}$ which is a function of N and a_{1i} only. Similarly, the FDL for v_2 is given by $\partial f_2(\mathbf{a}_2) / \partial a_{2i} = (a_{1i} + a_{2i}) \cdot \partial B_{2i} \, a_{2i} + B_{2i}$ which is a function of N, a_{1i}, and a_{2i}

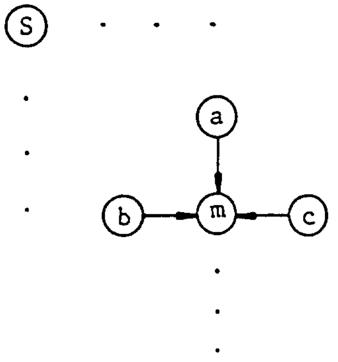


Fig 1. Reception of the forward search-messages at intermediate node m
S=source node, (a,b,c)=neighbor nodes of m.

only. In Fig. 1, node m receives the first search message from node a, after which the second search message from node b (and the third search message from node c, if any) will arrive at node m within Δ time. Then, node m calculates FDL's and finds the shortes path MFDL from the source node S to node m. That is,

$$\mathrm{MFDL}_{s,m}=\mathit{min}[\mathrm{FDL}_{a,m}+\mathrm{MFDL}_{s,a},\ \mathrm{FDL}_{b,m}+\mathrm{MFDL}_{s,b},\ \mathrm{FDL}_{c,m}+\mathrm{MFDL}_{S,c}]$$

where *min* stands for minimum. The search message with the above $\mathrm{MFDL}_{S,m}$ information will then be retransmitted sequentially to other nodes. This search message propagation scheme is similar to the flood search algorithm [1] except for the decision rule described below.

The decision rule for the modified flood search algorithm is as follows. For simplicity, we explain the rule by taking an example shown in Fig. 2,

(a) At first cycle : Nodes 7,10,12,15 receive the first search message from the source node S and FDL's at each node (node 7,10,12,15), i.e., $\mathrm{FDL}_{S,7}$, $\mathrm{FDL}_{S,10}$, $\mathrm{FDL}_{S,12}$, and $\mathrm{FDL}_{S,15}$, are calculated. Next, the shortest paths from the source node S to nodes 7,10,12,15, i.e., $\mathrm{MFDL}_{S,7}$, $\mathrm{MFDL}_{S,10}$, $\mathrm{MFDL}_{S,12}$, and $\mathrm{MFDL}_{S,15}$ are determined. If none of these four nodes are destination nodes, this MFDL information (in this case, MFDL=FDL) is sent to all adjacent nodes.

(b) At second cycle : Nodes 3,6,8,9,14,16 receive the first and second (and third, and so forth, if any) search message(s) and FDL's are calculated at each node. Next, the MFDL's are determined. For example, at node 6 we have $\mathrm{MFDL}_{S,6}=\mathit{min}[\mathrm{FDL}_{6,7}+\mathrm{MFDL}_{S,7},\ \mathrm{FDL}_{6,10}+\mathrm{MFDL}_{S,10}]$. In this case, node 6 receives the first search message and receives the second search message and will not receive the third message within Δ time (on the order of millisecond). Also, at node 9 we have $\mathrm{MFDL}_{S,9}=\mathrm{FDL}_{9,10}+{}_{S,10}$. In this case, node 9 receives only the first search message and will not receive the second search message within Δ

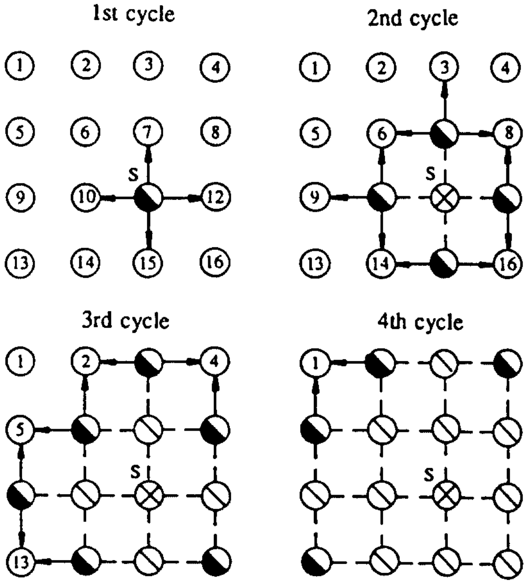


Fig 2. Forward search-message propagation scheme of a call originated at source node S

⊗ source node

⦸ intermediate node over which a message has passed

◐ intermediate node in progress of transmitting a search-message

ⓝ intermediate node n in progress of receiving a search-message.

time. If none of these six nodes are destination nodes, this MFDL information is sent to all adjacent nodes.

(c) At third cycle : Nodes 2,4,5,13 receive the first and second (and third, and so forth, if any) search message(s) and FDL's are calculated at each node. Next, the MFDL's are determined. For example, at node 2 we have $\mathrm{MFDL}_{S,2} = \boldsymbol{min} [\mathrm{FDL}_{2,3} + \mathrm{MFDL}_{S,3},\ \mathrm{FDL}_{2,6} + \mathrm{MFDL}_{S,6}]$. If none of these four nodes are destination nodes, this MFDL information is sent to all adjacent nodes.

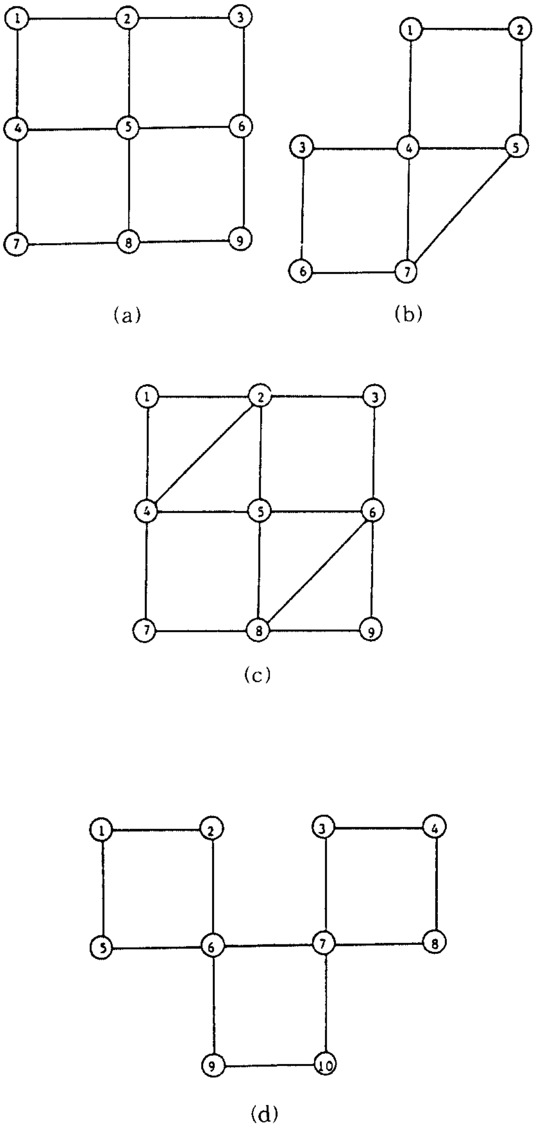
(d) At fourth cycle : Node 1 does the same job as in the third cycle. At node 1 we have $\mathrm{MFDL}_{S,1} = \boldsymbol{min} [\mathrm{FDL}_{1,2} + \mathrm{MFDL}_{S,2},\ \mathrm{FDL}_{1,5} + \mathrm{MFDL}_{S,5}]$.

Here, node 1 is the final destination node, and therefore, the shortest path from S to node 1 using the MFDL informations is determined.

The processing time, timing delay, and call set-up time based on the modified flooding algorithm can be compared with those based on the original flood search algorithm [1] in the Appendix.

Ⅲ. Numerical Results

As mentioned, the performance measure used is average EEBP, i.e., P_B. The network topologies are depicted in Fig. 3 ; topology 1 (TOP1) has nine nodes, twelve links and three commodities, topology 2 (TOP2) has seven nodes, nine links and three commodities, topology 3 (TOP3) has nine nodes, fourteen links and three commodities, and topology 4 (TOP4) has ten nodes, twelve links and three commodities. The total external offered traffic of example topologies is shown in Table 1. In Fig. 4, end-to-end blocking performance versus the total external traffic for high-priority voice traffic, A_1 (in Erlangs), for topology TOP1 is plotted using a logarithmic scale. Three groups of curves correspond to the end-to-end blocking probability of (a) flood search, (b) modified flood search for low-priority voice, v_2, and (c) modified flood search for high-priority voice, v_1. It is seen that our modified flooding algorithm yields better blocking performance (curves b and c) than the flood search algorithm (curve a) under the same conditions. Also, the blocking poerformance for high-priority voice traffic with the modified



(a) (b) (c) (d)

Fig 3. Network topologies

(a)TOP1 (b)TOP2 (c)TOP3 (d)TOP4

Table 1. Experiment cases and offered traffics of example topologies

case	Topology	No. of Commodities (v_1/v_2)	No. of Channels	Total Offered Traffic (Erlangs)		
				Commodity 1	Commodity 2	Commodity 3
Case1	TOP1	3/3	30	$A_1(1,9)=VA1$ $A_2(1,8)=1$	$A_1(2,8)=1$ $A_2(2,8)=1$	$A_1(3,7)=2.5$ $A_2(3,6)=2.5$
				$A_1(1,9)=VA1$ $A_2(1,8)=1$	$A_1(2,8)=5$ $A_2(2,8)=5$	$A_1(3,7)=5$ $A_2(3,6)=5$
Case2	TOP2	3/3	30	$A_1(1,10)=VA1$ $A_2(1,9)=1$	$A_1(2,8)=1$ $A_2(2,8)=1$	$A_1(4,9)=2.5$ $A_2(4,10)=2.5$
	TOP3	3/3	30	$A_1(1,9)=VA1$ $A_2(1,8)=1$	$A_1(2,8)=1$ $A_2(2,8)=1$	$A_1(3,7)=2.5$ $A_2(3,6)=2.5$
Case3	TOP1	3/3	VA2	$A_1(1,9)=12$ $A_2(1,8)=12$	$A_1(2,8)=1$ $A_2(2,8)=1$	$A_1(3,7)=2.5$ $A_2(3,6)=2.5$
	TOP3	3/3	VA2	$A_1(1,9)=VA1$ $A_2(1,8)=1$	$A_1(2,8)=1$ $A_2(2,8)=1$	$A_1(3,7)=2.5$ $A_2(3,6)=2.5$
	TOP4	3/3	VA2	$A_1(1,9)=VA1$ $A_2(1,8)=1$	$A_1(2,8)=1$ $A_2(2,8)=1$	$A_1(3,7)=2.5$ $A_2(3,6)=2.5$
Note	v_1 = high-priority voice, v_2 = low-priority voice, A_p = total offered voice traffic of priority p (p=1,2) from source (S) to desination (D) node VA1 = 13~25 VA2 = 10~40					

flooding algorithm is superior to the others. We consider two different traffic load conditions which corresponds to the Case1 in Table 1, and in this case, N is fixed. It clearly shown that if we use small values of total offered traffic for each commodities, the blocking performance (solid line) is better than the other using large values (dashed line). This is because the FDL path is affected by two factors; the number of channels, N, and the total offered load of priority $p(p=1, 2), A_p$. The blocking performance for TOP2 and TOP3 in Case2 is shown in Fig. 5. One can see that the result is similar to that shown in Fig. 4. And it is seen that the blocking performance of TOP3 with the same offered load is better than that of TOP2. Considering the Case3, In Fig. 6, Log(P_B) versus the channel number for TOP1, TOP3 and TOP4 are plotted; (a) modified flood search for high-priority voice, v_1, (b) modified flood search for low-priority voice, v_2, and (c) flood search. Increasing the number of channers with total offered traffic fixed, the blocking performance becomes better, and the modified flood search for high-priority voice, (curve a) shows the best performance. Also, it is seen that the blocking performance of TOP3 with the same offered load is superior to others, this is because TOP3 has more links than others.

From the numerical results for a circuit-switched network with two priority classes of voice traffic presented above, we can conclude that the proposed algorithm yields a strictly

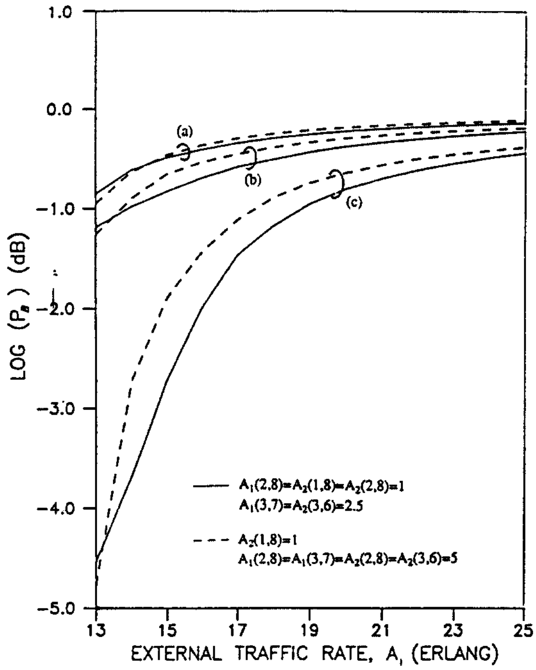


Fig 4. Log(P_B)versus external traffic rate, A_1 (in Erlangs), for TOP1
(a) flood search
(b) modified flood search for low-priority voice, v_2
(c) modified flood search for high-priority voice, v_1

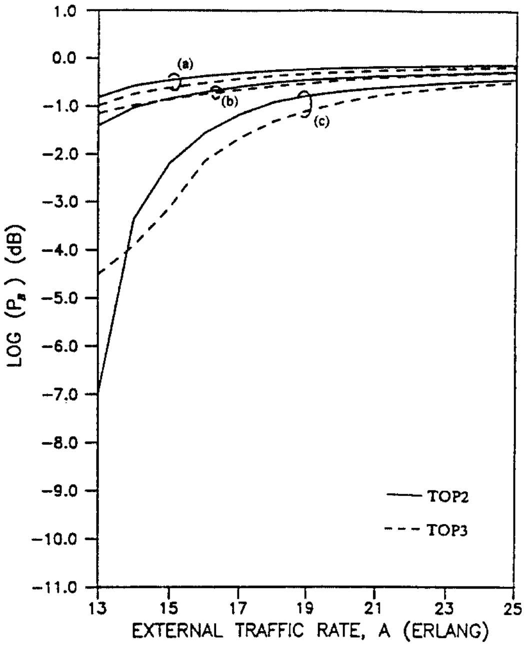


Fig 5. Log (P_B) versus external traffic rate, A_1 (in Erlags), for TOP2 and TOP3
(a) flood search
(b) modified flood search for low-priority voice, v_2
(c) modified flood search for high-priority voice, v_1

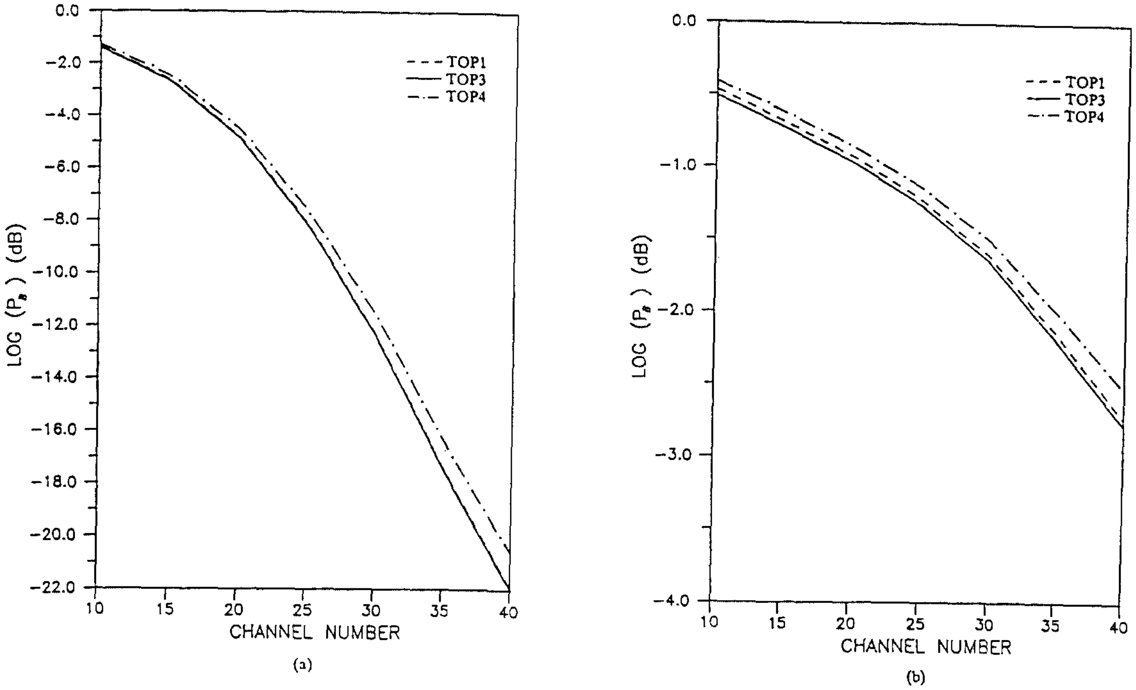
better blocking performance than the flooding algorithm. Specifically, the modified flooding for high-priority voice traffic shows excellent performance.

As for the processing and call set-up time, the total switch processing time per attmept of the modified flooding algorithm takes only 471 μs which is about one tenth that of the flood search algorithm, and the call set-up time for a 15 link-call of the modified flooding algorithm takes 582 ms which is less than that of the flood search algorithm. Thus, one can conclude that, using the modified flooding algorithm with MFDL path, one can improve the blocking performance of a circuit-switched network without increasing the call set-up time and processor loading of the algorithm.

Ⅳ. Conclusions

Flood search algorithm for circuit-switched networks offers many advantages. This statement is supported by the increasing popularity of the algorithm for European and American tactical networks. But this algorithm has relatively poor blocking performance.

In this paper, we have proposed a modified flooding algorithm with MFDL path for a tactical circuit-switched grid network with a maximum of four links and two priority classes of voice traffic. We have studied the blocking performance of the modified flooding algorithm and the flood search algorithm. Also we have analyzed the signaling traffic of the algorithm. With the same or less cost of switch processor loading and call set-up



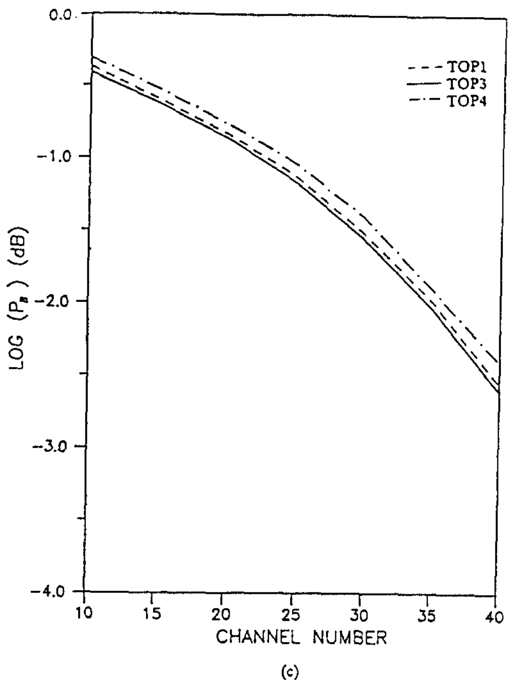


Fig 6. Log (P_B) versus the channel number for TOP1, TOP3 and TOP4
(a) modified flood search for high-priority voice, v_1
(b) modified flood search for low-priority voice, v_2
(c) flood search

time as compared to the flood search algorithm, the modified flooding algorithm provides the shortes path on the basis of blocking probability. Therefore, one can improve the blocking performance of a circuit-switched network. And our system provides the excellent blocking performance for high-priority voice traffic, which is very important for high-priority subscribers in the tactical network. Clearly, one can see that in military applications of the technique, our scheme can improve the network efficiency over the conventional flood search algorithm.

REFERENCES

1. G. Ludwig and R. ROY, "Saturation Routing

Network Limits," Proc. IEEE, vol 65 (9), pp 1353-1362. Sep. 1977.

2. V. O. K. Li and R. F. Chang, "Proposed Routing Algorithms for The U.S. Army Mobile Subscriber Equipment (MSE) Network" Milcom, paper 39.4. 1986.
3. L. E. Miller, R. H. French, J. S. Lee, and D. J. Torrieri, "MSE Routing Algorithm Comparison," Proc. IEEE Milcom, paper 2.2. 1986.
4. J. M. McQuillan et al, "Improvements in the Design and Performance of the ARPA Network," AFIPS Conference Proc. 41 pp. 741-745 (December 1972). Fall Joint Computer Conference.
5. J. M. McQuillan, I. Richer, and E. C. Rosen, "The New Routing Algorithm for the ARPANET," IEEE Trans. on communications COM-28(5) pp. 711-719 (May 1980).
6. A. Segall, "Distributed Network Protocols," IEEE Trans. on Information Theory, pp. 23-24 (January 1983).
7. I. Cidon and I. S. Gopal, "PARIS: An Approach to Intergrated High-Speed Private Networks," International Journal of Digital & Analog Cabled Systems 1(2) pp. 77-86 (April-June 1988).
8. I. Cidon, I. S. Gopal, and S. Kutten, "New Models and Algorithms for furture Networks," in proceedings of PODC'88, (1988).
9. C. Grandjean, "Traffic calculations in sqturation routin g with priorities," 7th Int. Teletraffic Congress, Stockholm, Sweden, June 1973.
10. Y. Nakagome, and Mori, H., "Flexible routing in the global communication network," 7th Int. Teletraffic Congress, Stockholm, Sweden, June 1973.
11. D. Bertsekas, and R. Gallager, "Data Networks,"(Prentice-Hall, 1987), Chap. 5.
12. T. -K. G. YUM: 'Circuit-switched routing in nonhierarchical networks,' Ph.D. dissertation, Dep. elec. Eng., Columbia Univ., New York, NY, (1985).

Appendix: Calculation of switch processor load, timing delay, and call set-up time for the modified flooding algorithm

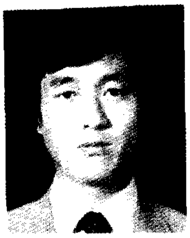
(a) Switch processor load for forward search pernetwork attmept: Assume that four interswitch trunk groups are connected to a switch and on the average three search messages are received per attempt. The node processor postulated is an Intel 80386, or equivalent, and processor resources required per network attempt are as follows (see [1] for details).

- Receive forward search message (for maximum 3 messages)
  [(50 instructions) × (0.3μs/istruction)] × (3 attmepts) = 45μs
- Search in progress table lookup
  -full search for first message received
  [(80 ns/entry) × (200 entries)] + [(100instructions/update) × (0.3μs/instructions)] = 46μs
  -partial search for subsequent message (for maximum 2 messages) (80 ns/entry) × (100 entries) × (2 messages) = 16μs
- Trunk group status search for transmitting order
  (100 instructions/trunk group) × (0.3μs/instruction) × (4 trunk groups) = 120μs
- Retransmission of message (for maximum 5 messages)
  (120 instructions/entry) × (0.3μs/instruction) × (5 messages) = 180μs
- Local directory table lookup (80% of 1000 entries)
  (80 ns/entry) × (800 entries) = 64μs

Thus, the total switch processor load (or processing tkme) per attmept takes only 471μs, while it takes 4950μs for the flood search algorithm.[1]

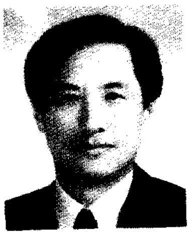
(b) Timing delay: Define an average call to be the time required for a call to travers 3 links and a maximum call to be the time required for a call to travers 15 links. The total timing delay is the time that takes a 160-bit forward search message to find the called subscriber, T_f, plus the time required for a 64-bit backward routing message to go back to the calling subscriber, T_b. Neglecting the transmission propagation time for a forward search 32 kbit /s commmon channel, $T_f = T_s + T_r + T_h$ =3 ms + 1 ms + 5 ms=9 ms=26 ms, while it take 28 ms for the flood search algorithm. The queueing delay is the time that buffers would have to store messages to wait for the processor to receive or transmit them.

(c) Call set-up time (or call processing time): Since the buffers are working simultaneously, only the delay due to one buffer would be necessary to calculate. An average delay is [(processor load /2)·(traffic /(1-traffic))] over processor load, where traffic means percentage of effective occupied time. With 7 traffic and 471 μs processor load, the average delay is calculated to be 1.2[1]. The queueing delay, T_q, is average delay multipled by the quantity given by (T_s for forward search + T_s for backward route). Therefore, $T_q = 1.2 \cdot (6\text{ ms} + 3\text{ ms}) = 10.8$ ms. For an average call with 3 links and with queueing delay, the call set-up time is $l \cdot (\mathrm{T} + T_q) = 3 \cdot (28\text{ ms} + 10.8\text{ ms}) = 117$ ms where l is the total number of links. for a 15-link call, the call set-up time takes 582 ms, while it takes 708 ms for the flood search algorithm.



▲Young Chul Park

Young Chul Park was born in Inchon, Korea, in 1953. He received the B.S. and M.S. degrees in electrical engineering from Sogang University, Seoul, Korea, in 1975 and 1981, respectively. Since 1976, he has joined the Agency for Defense Development (ADD) in Korea and spent the first five years as a design engineer of radar system and image processing system. During a 1982-83 sabbatical leave, he worked with AMMRC at Watertown, in Massachusetts, U.S.A. He returned to ADD in August 1983 and worked with Dr. S.C. Lee on the development of military communication systems as a senior research engineer. He received the Ph. D. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), in 1991. Recently, he joined ADD again, where he is currently project manager of communication network division.



▲Sang C. Lee

Sang C. Lee was born in Seoul, Korea, on February 20, 1948. He received the B.S. degree from Seoul Naional University, the M.S. degree from Virginia Polytechnic Institute and State University, and the Ph.D. degree from Duke University, in electrical engineering in 1971, 1973 and 1976, respectively.

From 1976 to 1979, he was with Western Union Spacecom, USA, as a senior member of research staff. Between 1979 and 1982, he was a principal engineer and manager at Computer Science Corp., USA, mainly involved in military communication network area. In 1982, he returned Korea and worked at Agency for Defense Development, Korea, as a menager. In 1991, he joined Korea Telecom(KT) Research Center, as a vice-president, and in 1992, became the first executive manager of KT Telecommunication Networks Research Lab. His research interests are in the fields of telecommunication information networking architecture and personal communication network. He led research planning of national G7 project Broadband-ISDN and currently is responsible for telecommunication network engineering part of HAN/B-ISDN.

▲Chong Kwan Un

Chong Kwan Un (S'63-M'64-SM'81-F'87) was born in Seoul, Korea, in 1940. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Delaware, Newark, in 1964, 1966, and 1969, respectively.

From 1969 to 1973 he was an Assistant Professor of Electrical Engineering at the University of Maine, Portland, where he taught communications and did research on synchronization problems. In May 1973 he joined the staff of the Telecommunication Sciences Center, SRI International, Menol Park, CA, where he did research on voice digitization and bandwidth compression systems. Since June 1977 he has been with Korea Advanced Institute of Science and Technology (KAIST), where he is a Professor of Electrical Engineering, teaching and doing research in the areas of digital communications and digital signal processing. He has authored or coauthored over 250 papers on speech coding and processing, data communications, packet communications, BISDN and adaptive filtering. Also, he holds seven patents granted. From February 1982 to June 1983 he served as Dean of Engineering at KAIST.

Dr. Un is a Fellow of IEEE. He received a number of awards, including the 1976 Leonard G. Abraham Prize Paper Award from the IEEE Communications Society, the National Order of Merits from the Government of Korea, and Achievement Awards from the Korea Institute of Telematics and Electronics (KITE), the Korea Institute of Communications Sciences (KICS), and the Acoustical Society of Korea (ASK). He was President of the ASK from 1988 to 1989. He is a member of Tau Beta Pi and Eta Kappa Nu.