論文 95-8-8-3

동적 계획법에 기초한 로컬 액세스 트리 네트워크의 토폴로지 설계 알고리즘

正會員 李 湧 振*, 金 泰 潤**

A Dynamic Programming Based Algorithm for the Topological Design of Local Access Tree Network

Yong Jin Lee*, Tai Yun Kim** Regular Members

要 約

본 연구는 로컬 액세스 트리 네트워크의 토폴로지 설계 문제를 다룬다. 이 문제는 종단 사용자 노드를 하나의 센터 노드에 최적으로 연결하는 방법을 구하는 것으로, 그래프 용어로는 용량 제한을 갖는 최소 신장 트리의 집합을 구하는 것이며 NP-complete 문제이다. 노드 수가 30개 이하인 경우 최적해를 제공하는 기존의 알고리즘은 대부분 분기 한계 기법을 사용한다. 본 연구에서는 동적 계획법에 기초한 새로운 2단계 알고리즘을 제시한다. 알고리즘은 1단계에서 동적 계획법을 이용하여 제약조건을 만족하는 가능해를 생성하고 2단계에서는 최적해를 찾는다. 제안한 알고리즘은 트래픽과 신뢰도의 두 가지 제약 조건을 동시에 고려할 수 있고 네트워크의 형상에 관계없이 적용될 수 있다.

ABSTRACT

This study deals with the topological design problem of local access tree network. The problem consists of finding the best way to link end user nodes to a center node. In graph-theoretical terms, it is to determine a set of CMST(Capacitated Minimal Spanning Tree) and a NP-complete combinatorial problem. In the case of the number of nodes less than thirty, most of the existing algorithms to produce the optimal solution use the branch and bound method. This paper presents new dynamic programming based algorithm which consists of two phases for this problem. In the first phase, the algorithm generates feasible solutions to satisfy the constraints and in the second phase, finds the optimal solution. The proposed algorithm can solve the problem with both traffic and reliability constraint and be applied to any network regardless of its configuration.

*중경산업대학교 컴퓨터과학과
**고려대학교 전산과학과
論文番號 : 94281-1010
接受日字 : 1994年 10月 10日

Ⅰ. 서 론

토폴로지는 네트워크내에서 링크의 연결 형태를 나타내는 용어로 성능, 전송 매체 그리고 프로토콜등과 상호 연관되어 있으며 연결 비용의 최소화가 무엇보다 중요한 문제이다. 또한 특정 토폴로지가 한번 선정되면 장기간에 걸쳐 네트워크 전반에 걸쳐 영향을 미치게 되므로 토폴로지의 선정 문제는 컴퓨터 네트워크의 기초를 이룬다고 할 수 있다.

현대적인 토폴로지의 구조는 단일형의 구조보다는 다양한 응용 분야에 대처하기 위한 혼합형 구조로 진행되고 있으며, 특히 광역 네트워크는 다양한 토폴로지를 갖는 네트워크의 인터네트워킹을 요구하고 있는 추세이다. 그러나 이러한 혼합형 네트워크의 토폴로지 설계 문제는 하나의 모델링으로 해결하기에는 고려해야 할 파라메타들이 매우 많은 복잡한 문제가 되기 때문에 문제를 여러 개의 부문제(subproblem)로 분할하고 각 부문제를 단계적으로 해결해 나가는 방법이 사용되고 있다. 즉 광역 네트워크의 구성을 기간망과 그에 접속된 다양한 형태의 로컬 액세스 네트워크(Local Access Network)의 혼합형으로 보고, 기간망 토폴로지 설계를 하나의 부문제로, 각각의 로컬 액세스 네트워크의 토폴로지 설계를 각각의 부문제로 나누어서 해결한다[10,14]. 여기서 로컬 액세스 네트워크의 개념은 근거리 통신망(Local Area Network)을 포괄하는 보다 광의의 개념으로 LAN을 하나의 단말 노드로 간주할 수 있다[11]. 대부분의 경우에 있어서 기간망 토폴로지는 신뢰도 향상을 위해 여러 개의 통신 경로를 두는 메쉬형의 분산형 토폴로지로 설계하고 로컬 액세스 네트워크는 기간망의 어느 한 노드에 접속되는 집중형 토폴로지로 설계하며 그 구조는 트리나 루프의 형태를 갖는 것이 일반적이다. 이 때 기간 노드에 있는 포트 1개가 처리할 수 있는 트래픽 용량이 제한(트래픽 조건)되거나 노드의 수가 제한(신뢰도 조건)되므로 로컬 액세스 네트워크의 토폴로지는 대개 제한된 수의 단말 노드들이 하나의 트리 또는 루프를 형성하면서 전체 단말 노드들을 담당하기 위해 여러 개의 트리 또는 루프가 기간 노드에 연결되는 구조를 갖게 된다. 이러한 최소 연결 비용을 갖는 트리의 집합을 구하는 문제를 로컬 액세스 트리 네트워크의 토폴로지 설계 문제라고 하고 그래프 용어로는 CMST(Capacitated Minimum Spanning Tree) 문제라고도 한다. 또한 최소 비용으로 루프의 집합을 구하는 문제를 로컬 액세스 루프 네트워크의 토폴로지 설계 문제 또는 MCLP(Minimum Cost Loop Problem)라고 한다. 이러한 집중형 토폴로지의 설계 기법은 대부분 LAN이나 분산형 토폴로지에도 적용 가능한 기본 알고리즘으로 알려져 있다[1,15]. 본 연구에서는 CMST 문제의 최적해를 구하는 데 촛점을 맞추며 이 문제는 NP-complete 문제로 알려져 있다[18].

CMST 문제를 해결하기 위한 알고리즘의 기법은 크게 휴리스틱 기법과 최적해 기법으로 나눌 수 있다. 먼저 휴리스틱 기법은 노드의 수가 많은 경우에 주로 이용되는 기법으로 부분 트리에 한번에 하나의 링크를 더하여 스패닝 트리를 구해 나가는 FOGA(First Order Greedy Algorithm)[2,6,13,14,20]와 FOGA의 해를 초기해로 이용하여 해를 개선해 나가는 SOGA(Second Order Greedy Algorithm)[17,21]등이 있다. 노드의 수가 비교적 적은 경우(대개 30개 이하)에는 최적해 기법을 이용할 수 있는 데 먼저 Chandy와 Russell[4]의 알고리즘은 분기 한계(branch and bound)기법을 사용한 것으로 제약 조건이 없는 최소 신장트리(Unconstrained Minimum Spanning Tree)의 해를 구한 후 이 값을 하한 값(lower bound)으로 놓고 알고리즘의 모든 해를 부분 집합(subset)으로 분할한다. 이 중에서 제약 조건을 만족하는 부분 집합(feasible subset)이 하한 값을 만족할 때 까지 분기(branching) 과정을 계속해 나가는 방법이다. Chandy와 Lo[3]에서는 30개 이하의 노드에서 적용 가능한 분기 한계 기법을 이용하였으나 수렴 속도가 느리고 매 단계에서 MST를 구해야 하는 단점이 있다.

Gavish[12]는 CMST 문제를 혼합 정수 계획법(mixed integer programming)을 이용하여 정식화하고 이를 풀기 위해서 Bender의 분해 절차(decomposition procedure)를 사용하였으나 노드 수가 10개 정도인 작은 문제에서도 결과가 좋지 않았다.

[9]에서는 lagrangean relaxation과 subgradient optimization에 기초한 새로운 relaxation을 제시하고 이를 이용하여 노드 수가 30개 이하인 경우에 있어서 해를 구한 바 있다.

[8]에서는 노드 사이의 거리가 대칭(symmetric)인 경우에 CMST 문제를 정수 계획법으로 정식화하고 LP(Linear Programming) relaxation을 이용하여

해를 구하였다.

또한 Kershenbaum 등[16]은 분기 한계 기법을 사용하여 노드를 분할하고 분기 규칙으로 깊이 우선 탐색(Depth First Rule)를 사용하였으나 실행 시간은 문제의 크기가 커짐에 따라 지수적으로 증가한다. 지금까지 살펴본 바와 같이 기존의 최적해 기법은 주로 분기 한계 기법을 사용하였고, 트래픽 조건이나 신뢰도 조건 중 어느 하나를 제약 조건으로 사용하였으며 거리 행렬을 대칭 행렬로 가정한 연구가 대부분이다.

본 연구에서는 새로운 접근 방식으로 동적 계획법(dynamic programming)을 이용하여 가능해를 생성하는 1단계와 이를 기초로 최적해를 구하는 2단계로 구성되는 알고리즘을 제시하고자 한다. 이 알고리즘은 트래픽 및 신뢰도의 두 가지 제약 조건을 동시에 고려할 수 있으며 거리 행렬의 형상에 관계없이 적용 가능하다. 전체 4장으로 구성되는 본 연구는 Ⅱ장에서 알고리즘의 모델링 및 해법을, Ⅲ장에서 성능 평가 및 분석을 그리고 Ⅳ장에서는 결론을 제시한다.

Ⅱ. CMST 문제의 모델링 및 알고리즘

2.1 용어 및 기호의 정의

n : 전체 노드수 (센터 노드 제외)

0 : 센터 노드 번호

N′: 0을 제외한 노드 번호의 집합 {1,2,...,n}

N_j: 노드 j를 제외한 노드 번호의 집합 {1,2,...,j-1,j+1,...,n} = N′ - {j}

d_{ij} : 노드 i와 노드 j 사이의 거리(시간,비용등)

D : 거리(시간,비용) 매트릭스

D′ : 최단 연결 거리 매트릭스(shortest connected distance matrix)

S : k개의 노드로 구성되는 N_j의 부분 집합(cardinality of S = k)

Q : 단일 서브트리에 둘 수 있는 최대 트래픽

W_j : 각 노드에서의 트래픽 요구량(j=1,2,...,n)

$f_k(j,S)$: Q 범위내에서 센터 노드 0에서 임의의 노드 j까지 k 개의 중간 경유 노드 집합 S를 거치는 최소 트리 비용(거리)

(이 때 k는 집합 S에 포함된 노드의 갯수를 나타낸다)

P_m: 순서를 무시한 임의의 동일한 원소로 구성된 {j} ∪ S의 집합

$f_k'(P_m)$: 집합 P_m에 대응하는 $f_k(j, S)$중에서 최소값

R_m : $f_k'(P_m)$에 대응하는 순서가 고려된 최적 정책의 노드 집합

$f(R_m)$: R_m에 대응되는 비용(= $f_k'(P_m)$)

R: 최적 서브트리의 집합

F : 최종 최적값

2.2 기본 가정

(1) 센터 노드의 수는 하나이며 그 용량에는 제한이 없다.

(2) 하나의 노드에서의 트래픽은 단일 서브트리가 담당할 수 있는 최대 트래픽(Q)을 초과할 수 없다. ($\mathrm{Max}\ W_i \le Q$)
$1 \le i \le n$

(3) 각 노드에서의 트래픽은 한번에 처리된다(No Split).

(4) 전체 노드의 트래픽의 총합은 Q를 초과한다.
$(\sum_{i=1}^{n} W_i > Q)$

2.3 동적 계획법에 기초한 모델링

CMST 문제를 해결하기 위한 알고리즘은 기존의 TSP(Traveling Salesman Problem)를 해결하기 위한 동적 계획법[5]을 수정하여 가능 서브트리(feasible subtree)를 생성하는 단계와 최적 트리의 조합을 얻는 매칭 단계의 2단계로 구성된다.

2.3.1 서브 트리 생성 단계 (subtree generation phase)

동적 계획법으로 정식화하기 위해 먼저 단계 변수(stage variable)와 상태 변수(state variable)를 정의한다. 단계 변수 k (k=1,2,...)는 센터 노드 0에서 임의의 노드 j까지 서브트리를 형성하기 위한 노드의 갯수이고, 상태 변수 j와 S는 각각 연결하려는 노드와 그 노드를 연결하기 위해 서브트리에 포함된 노드의 집합이다. 그러면, 최적성의 원리를 이용하여 다음의 순환 관계식(recurrence relation)을 얻는다.

$$\text{if } \sum_{q \in S} W_q + W_j \le Q$$

$$f_k(j,S) = \mathrm{Min}_{q \in S,\ q \neq j} \{ f_{k-1}(q, S-\{q\}) + d_{qj} \} \quad \text{for } k=1$$

$$f_k(j,S)=\mathrm{Min}[f_{k-1}(q,S-\{q\})+d_{qj}], \quad \text{for } k \geq 2$$
$$q\in S, q\neq j$$
$$\Sigma f_1(q,\{j\})-(k-1)d_{0j}]$$
$$q\in S, q\neq j$$
$$(k=1,2,\ldots ; S\subseteq N_j) \qquad (1)$$
$$\text{else } f_k(j,S) = \infty$$

식 (1)에서 $f_k(j,S)$는 노드 j를 연결하기 위해 k 개의 노드가 포함된 서브트리의 비용중에서 최소 비용을 뜻하며, Q를 초과하는 노드의 연결에 대해서는 $f_k(j,S)$를 ∞로 놓음으로써 서브트리를 형성할 수 없도록 하였다.

경계 조건(boundary condition)은 센터 노드 0에서 중간 노드를 거치지 않고 직접 각 노드 j를 연결하는 비용을 의미하며 그 식은 (2)와 같다.

$$f_0(j,-) = d_{0j} \qquad (2)$$

가능해의 집합을 얻기 위해서는 먼저 f_0를 사용하여 $\sum_{q\in S} W_q + W_j \leq Q$를 만족하는 모든 (j,S)에 대해 $f_1(j,S)$를 계산한다. 다음으로 f_1을 사용하여 $f_2(j,S)$를 계산한다. 이런 식으로 계속해 나가면 모든 (j,S)에 대해 $\sum_{q\in S} W_q + W_j > Q$인 단계에 도달된다. 그러한 단계 k를 L이라고 하면, 당연히 $\forall(j,S)$, $f_k(j,S) = \infty$이고 이는 더 이상 트리가 확장될 수 없음을 의미한다. 따라서 이전의 k(0,1,2,....,L-1) 단계에서 얻어진 트리들이 가능해의 집합이 된다. 식 (1)에서 $\sum_{q\in S, q\neq j} f_1(q,\{j\}) - (k-1)d_{0j}$는 트리가 분기되는 경우를 고려한 식이다. 이를 예를 들어 설명한다.

〔예제 1〕 노드 수가 4, 센터 노드 번호가 0, 연결 대상 노드가 2라 하고 대상 그래프가 완전 연결 그래프라고 하자.

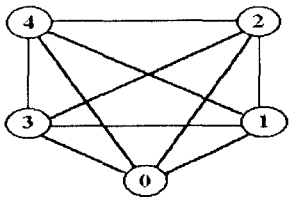

그림 1. 예제 그래프

이 경우의 순환 관계식은 식 (1)에 의해 다음과 같다.

k = 0: $f_0(2,-) = d_{02}$

k = 1: $f_1(2,\{1\}) = f_0(1,-)+d_{12}$, $f_1(2,\{3\}) = f_0(3,-)+d_{32}$
$f_1(2,\{4\}) = f_0(4,-)+d_{42}$

k = 2: $f_2(2,\{1,3\}) = \mathrm{Min}[f_1(1,\{3\})+d_{12},$
$f_1(3,\{1\})+d_{32}$, $f_1(1,\{2\})+f_1(3,\{2\})-d_{02}]$
$f_2(2,\{1,4\}) = \mathrm{Min}[f_1(1,\{4\})+d_{12},$
$f_1(4,\{1\})+d_{42}$, $f_1(1,\{2\})+f_1(4,\{2\})-d_{02}]$
$f_2(2,\{3,4\}) = \mathrm{Min}[f_1(3,\{4\})+d_{32},$
$f_1(4,\{3\})+d_{42}$, $f_1(3,\{2\})+f_1(4,\{2\})-d_{02}]$

$f_2(2,\{3,4\})$인 경우의 가능한 연결은 그림 2와 같다.

그림 2의 〈c〉에서 $f_1(3,\{2\}) + f_1(4,\{2\}) = f_0(2,-) + d_{23} + f_0(2,-) + d_{24} = 2f_0(2,-) + d_{23} + d_{24} = 2d_{02} + d_{23} + d_{24}$이므로 d_{02}가 두 번 포함된다. 따라서 〈c〉의 비용은 $f_1(3,\{2\}) + f_1(4,\{2\}) - d_{02}$이다.

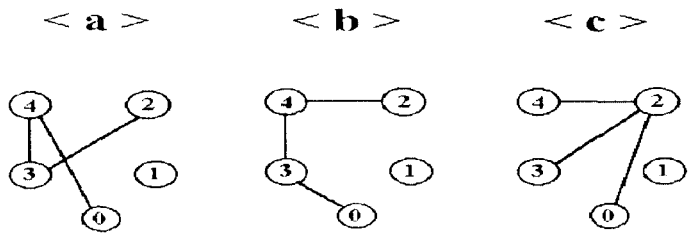

그림 2. $f_2(2,\{3,4\})$인 경우의 가능한 연결

k = 3: $f_3(2,\{1,3,4\}) = \mathrm{Min}[f_2(1,\{3,4\}) + d_{12}, f_2(3,\{1,4\}) + d_{32}, f_2(4,\{1,3\}) + d_{42}, f_1(1,\{2\}) + f_1(3,\{2\}) + f_1(4,\{2\}) - 2d_{02}]$

이 경우의 가능한 연결은 그림 3과 같다.

그림 3의 〈a〉,〈b〉,〈c〉에서 타원으로 표시된 부분의 최적 연결은 이미 k = 2의 단계에서 구해졌으므로 단지 노드 2에 이르기 위한 S만을 고려하면 된다. 예를 들어 그림 3의 〈a〉의 비용은 $f_2(1,\{3,4\}) + d_{12}$이다. 그런데 k = 2의 단계에서 $f_2(1,\{3,4\}) = \mathrm{Min}[f_1(3,\{4\}) + d_{31}, f_1(4,\{3\}) + d_{41}, f_1(3,\{1\}) + f_1(4,\{1\}) - d_{01}]$의 값을 구했으므로 그것의 최적 연결은 그림 4중에서 하나가 되었을 것이다.

이제 트리가 분기되는 경우를 보면, 그림 3의 〈d〉에서 $f_1(1,\{2\}) + f_1(3,\{2\}) + f_1(4,\{2\}) = f_0(2,-) + d_{21} + f_0(2,-) + d_{23} + f_0(2,-) + d_{24} = 3f_0(2,-) + d_{21} + d_{23} + d_{24} = 3d_{02} + d_{21} + d_{23} + d_{24}$이므로 d_{02}가 3번 포함된다. 따라서 〈d〉의 비용은 $f_1(1,\{2\}) + f_1(3,\{2\}) + f_1(4,\{2\}) - 2d_{02}$가 됨을 알 수 있다. 이상으로 부터 단계 k에서 노드 j에 이르기 위한 분기 비용은 $\sum_{q\in S, q\neq j} f_1(q,\{j\}) - (k-1)d_{0j}$가 된다. ■

2.3.2 매칭 단계(matching phase)

서브트리 생성 단계의 각 단계 k에서, $f_k(j,S)$는 $\{j\} \cup S$가 순서는 다르나 동일한 노드의 집합으로 구성된 트리 비용이므로 그들 중에서 최소값인 $f'_k(P_m)$를 구한다. 즉, 단계 k에서 집합 P_m에 대한 값 $f'_k(P_m)$는

$$\begin{cases} f'_0(P_m) = f_0(j,-), & k = 0 \\ f'_k(P_m) = \mathrm{Min}[f_k(j,S)], & k = 1,2,\ldots,L-1 \end{cases} \quad (3)$$

$$\forall (j,S) \text{ such that } P_m - \{\{j\} \cup S\} = \emptyset$$

이 된다. 위 식의 값은 각 단계에서 노드의 순서는 다르지만 동일한 노드로 구성된 집합의 트리 비용이 되고 $f'_k(P_m)$에 대응하는 최적 정책(optimal policy)의 노드 집합은 R_m가 된다. 즉 R_m은 센터 노드를 루트로 하는 트리상에 나열된 노드의 집합이다. 물론 R_m에는 {0}을 포함시키지 않는다.

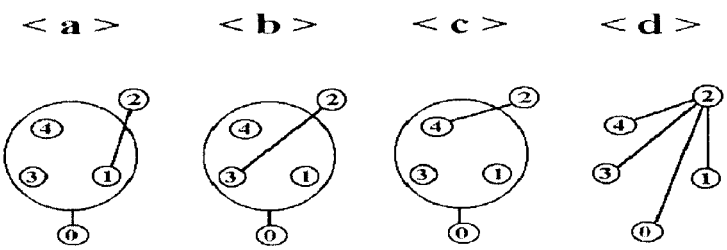

그림 3. $f_3(2,\{1,3,4\})$인 경우의 가능한 연결

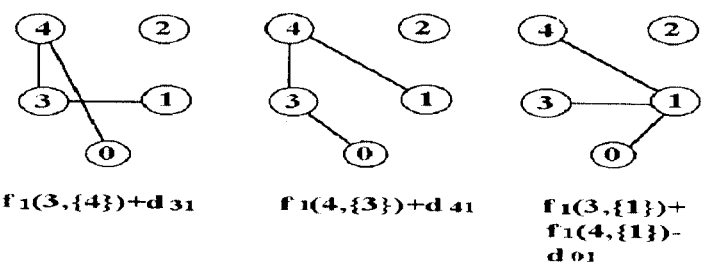

그림 4. $f_2(1,\{3,4\})$인 경우의 가능한 연결

마지막으로 최적해(optimal solution)을 얻기 위해서는 n개의 노드가 중복이 없이 루트를 0으로 하는 임의의 트리 R_m에 포함되어야 하므로, 식 (3)에서 얻은 트리들의 값인 $f'_k(P_m)$를 $f(R_m)$로 치환한 후에 최종 최적값을 구한다.

집합 N'의 분할 조건을 만족하는 R_m을 원소로 하는 집합류는 여러 개가 될 수 있다. 따라서 최적값 F는 대응되는 여러 개의 비용 $f(R_m)$의 합 중에서 최소값이다. 이는 다음과 같이 정리화 될 수 있다.

[Lemma 1] 전체 집합 $N'=\{1,2,\ldots,n\}$의 분할(partition)중의 하나인 τ를 $\tau=\{A_0, A_1, \ldots, A_L\}$라고 하자. N'의 부분 집합들의 집합류(collection)의 하나를 C, 집합류 C의 원소인 부분 집합을 $S_0, S_1, \ldots, S_L$라고 하자. 여기서 임의의 두 개의 집합 $S_i \cap S_j = \emptyset$ $(\forall i,j, i \neq j)$이고 동시에 $\cup_{i=0}^{L} S_i = N'$이라고 하자. 이제 S_i (i=0,1,2,...,L)는 i+1개의 원소로 구성된 집합류로, 그것의 원소로 S_{ij}(j=1,...,i+1)라는 집합을 갖는다고 하자. 이 때 S_{ij}는 A_i의 원소들과 동일한 원소로 구성되고 나열 순서가 서로 다른 동적 계획법에 의해 생성된 집합으로 $A_i = \cup_{j=1}^{i+1} S_{ij} = \cap_{j=1}^{i+1} S_{ij}$의 성질을 만족한다. 이제, 집합 S_{ij}에 대응하는 비용 $f(S_{ij})$가 주어지고 집합 S_{ij}의 합집합이 N'이 되는 최소 비용을 Fc라고 하면

$$F_c = \underset{\substack{i\in\{0,1,2,\ldots,L\} \\ j\in\{1,2,\ldots,i+1\}}}{Min}\left[\ \sum f(S_{ij})\ \right] = \sum_{i=0}^{L}\left[\ \underset{1\le j\le i+1}{Min}\left[\ f(S_{ij})\ \right]\ \right]$$

이 성립한다.

[증 명] 1) L = 0일 때,

$F_c = \underset{i=0,\, j=1}{\text{Min}}[\Sigma f(S_{ij})] = [\text{Min}\ \{f(S_{01})\}]$

$= \text{Min}[f(S_{01})]$이므로 성립

```
PHASE 1 : subtree generation
STEP 1 : Initialization( For k = 0 )
         compute boundary condition,
          f0(j,-) = d0j, for each j, j ∈ N'
STEP 2 : For k=1,2,3,....
         (1) Feasibility Check and Recurrence Relation Computation
         if Σ_{q∈S} Wq + Wj ≤ Q,

          fk(j,S) = Min{fk-1(q,S-{q})+dqj},                          for k = 1
                    q∈S,q≠j
          fk(j,S) = Min{fk-1(q,S-{q})+dqj, Σ f1(q,{j})-(k-1)dpj}, for k ≥ 2
                    q∈S,q≠j                q∈S,q≠j

                (k=1,2,... : S⊆Nj)
         else = ∞

         (2) End condition check
         if ∀(j,S), fk(j,S)=∞ let L=k, Go to PHASE 2
         else                 Repeat STEP 2

PHASE 2 : Matching
STEP 1 : For k = 0, For each j, j ∈ N',
         Find f'0(Pm) = f0(j,-)
STEP 2 : For k=1,2,...,L-1,
         For j such that j < Min [q], let Pm = {j} ∪ S
                              q∈s
         Find f'k(Pm) = Min{fk(j,S)}
                        ∀(j,S) such that Pm - {j ∪ S} = ∅
STEP 3 : Find F = Min[Σ f(Rm)],
                  ∀ Rm such that ∪Rm = N' and Ri∩Rj = ∅ (i ≠ j)
STEP 4 : Find optimal subtrees(R) corresponding to F value
```

그림 5. 최적 CMST 알고리즘

2) L = n일 때,

$F_c = Min[\Sigma f(S_{ij})] = Min\ [f(S_{01}) + f(S_{11}) + .. + f(S_{n1}),\ f(S_{01}) + f(S_{11}) + .. + f(S_{n2}),\ \ldots, f(S_{01}) + f(S_{12}) + .. + f(S_{n,n+1})] = Min[f(S_{01})] + Min[f(S_{11}), f(S_{12})] + \ldots + Min[f(S_{n1}), \ldots f(S_{n,n+1})] = \sum_{i=0}^{n} Min\ [f(S_{ij})]$이 성립한다고 가정하면

3) L = n+1일 때,

$F_c = Min[\Sigma f(S_{ij})] = Min\ [f(S_{01}) + f(S_{11}) + .. + f(S_{n+1,1}),\ f(S_{01}) + f(S_{11}) + .. + f(S_{n+1,2}),\ \ldots, f(S_{01}) + f(S_{12}) + .. + f(S_{n+1,n+2})]$

$= Min[f(S_{01})] + Min[f(S_{11}), f(S_{12})] + \ldots + Min[f(S_{n1}), \ldots, f(S_{n,n+1})] + Min[f(S_{n+1,1}) \ldots f(S_{n+1,n+2})] = \sum_{i=0}^{n} Min\ [f(S_{ij})] + Min[f(S_{n+1,1}), \ldots f(S_{n+1,n+2})]$

$= \sum_{i=0}^{n+1} Min\ [f(S_{ij})]$이므로 임의의 L에 대해 성립된다.

■

[Lemma 1]과 식 3에서 {j∪S}는 S_{ij}로, $f_k(j,S)$는 $f(S_{ij})$로 대응된다. 또한 $Min[f(S_{ij})]$는 $f'_k(P_m)$ (= $f(R_m)$)에 대응된다. 집합류 C가 여러 개 존재할 수 있으므로 F는

```
알고리즘: State Space Relaxation Algorithm
 STEP 1 : /* Initialization(k = 0) */
          π0 ← ∅;
          for j=1 to n{
          ψ0j ← j; f0(j,-) ← d0j; π0 ← π0 ∪ψ0j; /* ψ0j ∈ π0 */
          }
          M ← n;
 STEP 2 : /* Recurrence Relation */
          for k=1 to n-1{
            πk ← ∅;
            l ← 1; if( M == 1) { L ← k; break; }
            for p=1 to M-1{
              for i=1 to M-p{
                 temp ← ψk-1,p∪ψk-1,p+i;
                 if(Σ_{q∈temp} ≤Q && temp∉πk && cardinality of temp == (k+1)){

                       ψk,l ← temp; l++;
                       πk ← πk∪ψkl; /* ψkl∈πk */
                 } /* end of if */
               } /* end of i */
             } /* end of p */

          for ψkp∈πk (p=1,2,...,l),
              if(k == 1){
                 fk(j,S) ← Min [fk-1(q,S-{q})+dqj]; }
                           q∈S,q≠j

                 else{
                   fk(j,S) ← Min [fk-1(q,S-{q})+dqj, Σ f1(q,{j})-(k-1)dj0]; }
                             q∈S,q≠j                 q∈S,q≠j

            if(πk == ∅ || l == n-1){
                 L ← k; break;}

            M ← l;
          } /* end of k */
```

그림 6. State Space Relaxation Algorithm(PHASE 1)

$$F = \mathrm{Min}\{\Sigma f(R_m)\} \quad (4)$$
$$\forall R_m \text{ such that } \cup R_m = N' \text{ and } R_i \cap R_j = \emptyset\ (i \neq j)$$

이 되고, 구하는 최적 서브 트리의 집합은 식 (4)를 만족하는 트리들이 된다.

2.4 최적 CMST 알고리즘

이상의 모델링으로 부터 최적 CMST 알고리즘은 그림 5와 같이 표현된다.

이제 신뢰도와 트래픽 조건을 만족하는 문제를 고려해 보자. 임의의 멀티드롭 라인에 둘 수 있는 노드의 수를 제한함으로써 신뢰도를 만족시키는 문제는 각 노드에서의 트래픽을 $W_j = 1 (j=1,2,\ldots,N)$, 최대 노드의 수를 Q로 놓고 최적 CMST 알고리즘을 적용하면 해결된다. 다음으로 하나의 트리에 둘 수 있는 최대 노드의 수(v)를 만족하면서 동시에 트래픽 조건을 만족시키려면 최적 CMST 알고리즘의 PHASE 1의 STEP 2의 (2)를 다음과 같이 수정하면 된다.

(2) End condition check

if $\forall (j,S)$, $f_k(j,S)=\infty$ or k=v, let L=k, Go to PHASE 2

else Repeat STEP 2

즉, 단계 변수(k)가 하나의 트리에 연결되는 노드의 수보다 1이 크므로(단, 센터 노드는 제외) k가 최대 v가 되거나 모든 (j,S)에 대해 $f_k(j,S)=\infty$가 될 때 까지 STEP 2를 반복하면 된다.

2.5 상태 공간의 축소(state space relaxation)

식 (1)에서 순환 관계식 $f_k(j,S)$를 계산할 때, 제약 조건을 만족하지 못하는 (j,S)에 대해 다음의 알고리즘을 이용하면 상태 공간을 축소시킬 수 있다.

먼저, 다음의 용어를 추가로 정의한다.

π_k : 임의의 단계 k에서 원소가 N'의 부분 집합으로 구성되고 원소의 갯수가 k+1인 집합류(collection)

ψ_{kj} : π_k의 j번째 원소인 집합 ($\psi_{kj} \in \pi_k$)

M : cardinality of π_k

l : 제약 조건을 만족하는 π_k의 원소 갯수를 누적하기

[예제 2] Q = 8, $W_1 = 3$, $W_2 = 6$, $W_3 = 3$, $W_4 = 2$

d_{ij} = 노드 i와 노드 j 사이의 거리(i=j=0,1,2,3,4, 센터 노드: i = 0)

k = 0	$\psi_{01} = \{1\}, \psi_{02} = \{2\}, \psi_{03} = \{3\}, \psi_{04} = \{4\}$ $\pi_0 = \{\{1\},\{2\},\{3\},\{4\}\}$ $f_0(1,-) = d_{01}$, $f_0(2,-) = d_{02}$, $f_0(3,-) = d_{03}$, $f_0(4,-) = d_{04}$ M ← 4
k = 1	$\psi_{11} = \{1,3\}$, $\pi_1 = \{\{1,3\}\}$ $\psi_{12} = \{1,4\}$, $\pi_1 = \{\{1,3\},\{1,4\}\}$ $\psi_{13} = \{2,4\}$, $\pi_1 = \{\{1,3\},\{1,4\},\{2,4\}\}$ $\psi_{14} = \{3,4\}$, $\pi_1 = \{\{1,3\},\{1,4\},\{2,4\},\{3,4\}\}$ M ← 4 $f_1(1,\{3\}) = \mathrm{Min}[f_0(3,-) + d_{31}]$, $f_1(3,\{1\}) = \mathrm{Min}[f_0(1,-) + d_{13}]$, $f_1(1,\{4\}) = \mathrm{Min}[f_0(4,-) + d_{41}]$, $f_1(4,\{1\}) = \mathrm{Min}[f_0(1,-) + d_{14}]$, $f_1(2,\{4\}) = \mathrm{Min}[f_0(4,-) + d_{42}]$, $f_1(4,\{2\}) = \mathrm{Min}[f_0(2,-) + d_{24}]$, $f_1(3,\{4\}) = \mathrm{Min}[f_0(4,-) + d_{43}]$, $f_1(4,\{3\}) = \mathrm{Min}[f_0(3,-) + d_{34}]$
k = 2	$\psi_{21}=\{1,3,4\}$, $\pi_2=\{\{1,3,4\}\}$ $f_2(1,\{3,4\}) = \mathrm{Min}[f_1(3,\{4\})+d_{31},\ f_1(4,\{3\})+d_{41},\ f_1(3,\{1\})+f_1(4,\{1\})-d_{01}]$ $f_2(3,\{1,4\}) = \mathrm{Min}[f_1(1,\{4\})+d_{13},\ f_1(4,\{1\})+d_{13},\ f_1(1,\{3\})+f_1(4,\{3\})-d_{03}]$ $f_2(4,\{1,3\}) = \mathrm{Min}[f_1(1,\{3\})+d_{34},\ f_1(3,\{1\})+d_{34},\ f_1(1,\{4\})+f_1(3,\{4\})-d_{04}]$ M ← 1
k = 3	since M = 1, stop, L = 3, go to PHASE 2 ■

위한 임시 변수

그러면, 최적 CMST 알고리즘의 PHASE 1은 그림 6과 같이 변환될 수 있다.

위 알고리즘에서 집합 π_k의 원소들의 갯수는 k값보다 1이 큰 (k+1)개이다. 만일 위의 조건이 성립되지 않으면 집합 π_k에서 조건을 만족하지 못하는 원소, ϕ_{kj}는 제거된다.([예제 2] 참조)

2.6 알고리즘의 적용 방법

네트워크를 형상(configuration)에 따라 분류하면, 무방향 네트워크(undirected network), 방향 네트워크(directed network), 혼합 네트워크(mixed network)로 나눌 수가 있다. 또한 무방향 네트워크는 거리 매트릭스가 대칭(symmetric)인 경우와 비대칭(non-symmetric)인 경우로 나눌 수가 있다. 무방향 네트워크인 경우에는 최적 CMST 알고리즘을 그대로 적용하고 방향 네트워크 또는 혼합 네트워크인 경우에는 기존의 Shortest Path Algorithm[7]을 기초로 하여 주어진 거리 매트릭스(D)를 최단 연결 거리 매트릭스(D′)로 변환한다. 이 때, 임의의 노드(i, j)의 연결이 센터 노드를 경유하는 경우에는 해당 d_{ij}를 ∞로 놓은 후 최적 CMST 알고리즘을 적용한다.

2.7 계산 예

하나의 서브트리에 둘 수 있는 최대 트래픽(Q)이 3이고 거리 매트릭스(D)가 대칭으로 아래와 같이 주어지는 문제를 고려해 보자. 각 노드에서의 트래픽 요구량은 1이다($\forall i,\ W_i = 1$).

	0	1	2	3	4
0	0	3	3	5	10
1		0	6	4	8
		2	0	3	5
			3	0	7
				4	0

동적 계획법을 이용한 계산 과정은 아래와 같다. 최적 정책(optimal policy)은 노드 j를 연결하기 위한 트리 상의 전단계 노드이고, 이 경우 ( )안에 표기된 부분은 트리에서 분기된 노드를 표시한다.

Ⅲ. 성능 평가 및 분석

본 연구에서 제시한 동적 계획법에 기초한 알고리즘을

phase	step	$f_k(j,S)$	optimal policy
1	1 (k=0)	$f_0(1,-) = d_{01} = 3$ $f_0(2,-) = d_{02} = 3$ $f_0(3,-) = d_{03} = 5$ $f_0(4,-) = d_{04} = 10$	0 0 0 0
	2 (k=1)	$f_1(1,\{2\}) = f_0(2,-) + d_{21} = 3 + 6 = 9$ $f_1(1,\{3\}) = 9,\ f_1(1,\{4\}) = 18$ $f_1(2,\{1\}) = 9,\ f_1(2,\{3\}) = 8,\ f_1(2,\{4\}) = 15$ $f_1(3,\{1\}) = 7,\ f_1(3,\{2\}) = 6,\ f_1(3,\{4\}) = 17$ $f_1(4,\{1\}) = 11,\ f_1(4,\{2\}) = 8,\ f_1(4,\{3\}) = 12$	2 3,4 1,3,4 1,2,4 1,2,3
	(k=2)	$f_2(1,\{2,3\}) = \mathrm{Min}\{f_1(2,\{3\})+d_{21},\ f_1(3,\{2\})+d_{31},\ f_1(2,\{1\})+f_1(3,\{1\})-d_{01}\}$ $= \mathrm{Min}\{8+6,\ 6+4,\ 9+7-3\} = 10$ $f_2(1,\{2,4\}) = 16,\ f_2(1,\{3,4\}) = 15$ $f_2(2,\{1,3\}) = 9,\ f_2(2,\{1,4\}) = 14,\ f_2(2,\{3,4\}) = 11$ $f_2(3,\{1,2\}) = 12,\ f_2(3,\{1,4\}) = 16,\ f_2(3,\{2,4\}) = 15$ $f_2(4,\{1,2\}) = 14,\ f_2(4,\{1,3\}) = 14,\ f_2(4,\{2,3\}) = 13$	3 4,(3,4) 3,(1,4),(3,4) 1,1,4 1,3,2
	(k=3)	$\forall (j,S),\ f_k(j,S) = \infty,\ L = 3$	

phase	step	$f'_k(P_m)$	R_m
2	1 (k=0)	$f'_0(\{1\}) = f_0(1,-) = 3$ $f'_0(\{2\}) = 3$ $f'_0(\{3\}) = 5$ $f'_0(\{4\}) = 10$	{1} {2} {3} {4}
	2 (k=1)	$f'_1(\{1,2\}) = \text{Min}[f_1(1,\{2\}), f_1(2,\{1\})] = 9$ $f'_1(\{1,3\}) = 7$ $f'_1(\{1,4\}) = 11$ $f'_1(\{2,3\}) = 6$ $f'_1(\{2,4\}) = 8$ $f'_1(\{3,4\}) = 12$	{1,2} {1,3} {1,4} {2,3} {2,4} {3,4}
	(k=2)	$f'_2(\{1,2,3\}) = \text{Min}[f_2(1,\{2,3\}), f_2(2,\{1,3\}), f_2(3,\{1,2\})] = \text{Min}[10, 9, 12] = 9$ $f'_2(\{1,2,4\}) = \text{Min}[16, 14, 14] = 14$ $f'_2(\{1,3,4\}) = \text{Min}[15, 16, 14] = 14$ $f'_2(\{2,3,4\}) = \text{Min}[11, 15, 13] = 11$	{1,3,2} {2,4,1} {1,3,4} {2,(3,4)}
	3	$f(\{1,3,2\}) + f(\{4\}) = 9 + 10 = 19$ $f(\{2,4,1\}) + f(\{3\}) = 14 + 5 = 19$ $f(\{1,3,4\}) + f(\{2\}) = 14 + 3 = 17$ $f(\{2,(3,4)\}) + f(\{1\}) = 11 + 3 = 14$ $f(\{1,2\}) + f(\{3,4\}) = 9 + 12 = 21$ $f(\{1,3\}) + f(\{2,4\}) = 7 + 8 = 15$ $f(\{1,4\}) + f(\{2,3\}) = 11 + 6 = 17$ $\therefore F = \text{Min}[19, 19, 17, 14, 21, 15, 17] = 14$	
	4	* optimal cmst : {0-1}, {0-2-(3,4)}	

평가하기 위해 먼저 각 단계 변수(k)에 대한 계산량을 [5]를 이용하여 계산하기로 한다. 각 k에 대한 계산량은 모든 노드에서의 트래픽이 1이고($W_j = 1, \forall j$), 하나의 서브트리에 둘 수 있는 최대 트래픽이 Q(= L)인 경우에 최대가 된다. 먼저 임의의 단계 k에서 $f_k(j, S)$는 $n\binom{n-1}{k}$개의 서로 다른 j, S쌍에 대해 계산되어야 한다. 그러한 계산은 k개의 덧셈과 (k-1)개의 비교를 요구하므로 단계 k = 1 ~ k = L에 대해 덧셈 횟수(number of additions)는 식 (5) 그리고 비교 횟수(number of comparisions)는 식 (6)으로 나타낼 수 있다.

$$\begin{aligned}\text{덧셈 횟수} &= n\sum_{k=1}^{L} k\binom{n-1}{k} \\ &= n(n-1)\sum_{k=1}^{L}\frac{(n-2)!}{(k-1)!(n-1-k)!} \\ &= n(n-1)\sum_{k=1}^{L}\binom{n-2}{k-1} \approx n(n-1)2^{L}\end{aligned} \tag{5}$$

$$\begin{aligned}\text{비교 횟수} &= n\sum_{k=1}^{L}(k-1)\binom{n-1}{k} \\ &= \text{덧셈 횟수} - n\sum_{k=1}^{L}\binom{n-1}{k} \\ &= n(n-1)\sum_{k=1}^{L}k\binom{n-2}{k-1} - n\sum_{k=1}^{L}\binom{n-1}{k} \\ &\approx n(n-1)2^{L} - n(2^{L}-1) \\ &\approx n(n-2)2^{L}\end{aligned} \tag{6}$$

식 (5)와 식(6)를 이용하여 노드 수가 30이고 Q가 최대 29일 때의 덧셈 횟수와 비교 횟수의 합에 근접하는 노드 수(n = 30 ~ n = 100)와 최대 Q에 대해 정리한 것이 그림 7에 나타나 있다. 그래프의 오른 쪽에 표시된 숫자는 각 노드 수에 대한 최대 Q이다.

그림 7에서 보듯이, 이론적인 계산량은 n=30, Q=29일 때와 n=100, Q=25일 때가 유사하지만 실제 실행 시

간은 매우 달라지게 된다. 그 이유는 필요한 메모리의 량이 각 단계 k에서 $f_k(j,S)$를 저장하기 위해 $n\binom{n-1}{k}$개의 기억 장소가 필요하기 때문이다. 즉, 노드 수가 커지는 경우에는 주기억 장치에 이전 계산 결과 값을 전부 유지할 수 없기 때문에 메모리 액세스 시간이 엄청나게 증가한다.

표 1은 본 연구에서 제시한 알고리즘의 평균 실행 시간에 대한 결과이다. 각 경우에 대해 문제를 10개씩 난수 발생(random number generation) 하고, 평균 실행 시간을 계산하였다. 실행 환경은 MS-DOS하의 IBM-486 PC이고 언어는 FORTRAN-77이다. 표 1에서 보듯이 본 연구에서 제시한 알고리즘은 전체 트래픽의 합이 Q에 비해 상대적으로 작은 경우에 효율이 좋다. 그 경우에는 L의 값이 작아지므로 당연히 PHASE 1의 계산량과 PHASE 2의 매칭 갯수가 작아지기 때문이다. 그림 7과 표 1로 부터 본 연구에서 제시한 알고리즘은 노드 트래픽과 Q에 따라 영향을 받음을 알 수 있고, 최대 노드 수가 30개 정도까지 효율적임을 알 수 있었다. 따라서 노드 수(n)가 30 이상인 경우에는 계산 시간이 지수적으로 급격하게 증가하기 때문에 휴리스틱 알고리즘을 이용하는 것이 바람직하다.

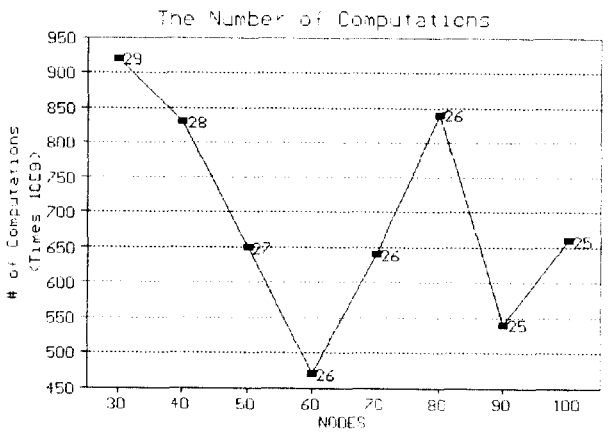

그림 7. 노드 수와 최대 Q와의 관계(노드 수 = 30, Q = 29 기준)

Ⅳ. 결 론

본 연구에서는 로컬 액세스 네트워크를 설계하는 데 있어서 발생되는 문제의 하나로 트래픽 및 신뢰도의 제약 조건을 만족하면서 전체 라인 비용을 최소로 하는 트리 형태의 토폴로지를 설계하기 위한 새로운 알고리즘을 제시하였다. 제안한 알고리즘은 동적 계획법에 기초하여 가능해를 생성하는 서브트리 생성 단계와 이를 토대로 최적해를 찾는 매칭 단계의 2단계로 구성되는 알고리즘이다. 아울러 서브트리 생성 단계에서의 계산량을 감소시키기 위한 state space relaxation 루틴도 제시되었다. 제안한 알고리즘은 기존의 최적해 알고리즘과는 달리 트래픽 및 신뢰도의 두 가지 제약 조건을 동시에 고려할 수 있으며 노드 사이의 비용 매트릭스의 형상에 관계없이 적용 가능하다. 실험 결과, 노드의 수가 30개 이하인 경우에 적용 가능함을 알 수 있었다. 따라서 본 연구는 노드 수가 비교적 적은 네트워크에서 최소 비용의 트리 토폴로지를 구하는 문제에 적용될 수 있다. 그러나 노드의 수가 커지는 경우에는 메모리 요구량과 계산 시간이 지수적으로 증가하게 되므로 휴리스틱 기법을 이용하는 것이 바람직하다. 앞으로의 연구에서는 이 문제에 대한 새로운 접근 방식이 기대된다.

표 1. 평균 실행 시간 (단위:초, IBM-486 PC)

경 우	n = 10	n = 20	n = 30
$Q=\frac{1}{2}\sum_{i=1}^{n} W_i$	0.32	174.00	200.00
$Q=\frac{1}{3}\sum_{i=1}^{n} W_i$	0.33	24.23	134.00
$Q=\frac{1}{6}\sum_{i=1}^{n} W_i$	0.27	25.10	60.10

참고문헌

1. V.Ahuja, Design and Analysis of Computer Communication Networks, pp.117-145, McGraw-Hill, 1985.
2. K.Altinkemer and B.Gavish, "Heuristics with constant error gurantee for the design of tree networks," Mgt. Sci., Vol. 34, No. 3, pp.331-341, 1988.
3. K.M. Chandy and T.Lo, "The Capacitated

Minimum Spanning Tree," Networks, pp.173-181, 1973.
4. K.M. Chandy and R.A.Russell, "The Design of Multipoint Linkages in a Teleprocessing Tree Network," IEEE Trans. on Computers, Vol.C-21, No. 10, pp.1062-1066, 1972.
5. R.E.Dreyfus and R.M.Law, The Art and Theory of Dynamic Programming, Academic Press, pp.69-75, 1977.
6. L.R.Esau and Williams, "On teleprocessing system design, part Ⅱ," IBM syst. J.,Vol. 5,No. 3, pp.142-147, 1966.
7. R.W.Floyd, "Algorithm 97, Shortest Path," Comm. ACM 5, 1962.
8. B.Gavish, "Augumented Based Algotithms for Centralized Network Design," IEEE Trans. on. Comm., Vol. COM-33, No. 12, pp.1247-1257, 1985.
9. B.Gavish, "Formulations and algorithms for the capacitated minimal directed tree problem," Journal of ACM 30, pp.118-132, 1983.
10. B.Gavish, "Topological Design of Telecommunication Networks-Local Access Design Methods," Annals of Operations Research 33, pp.17-71, 1991.
11. B.Gavish, "Topological Design of Communication Networks-The Overall Design Problem," European J. of Operations Research 58, pp.149-172, 1992.
12. B.Gavish, "Topological Design of Centralized Computer Networks-Formulations and Algorithms," Networks, Vol. 12, pp.355-377, 1982.
13. B.Gavish and K. Altinkemer, "Parallel Savings Heuristics for the Topological Design of Local Access Tree Networks," Proceedings IEEE-INFOCOM' 86, pp.139-139, 1986.
14. V.M.Grout and P.W.Sanders, "Communication Network Optimization," Computer Communication, vol. 11, pp.281-287, 1988.
15. A.Kershenbaum, Telecommunications Network Design Algorithm, McGraw-Hill, 1993.
16. A.Kershenbaum and R.R.Boorstyn, "Centralized Tele Processing Network Design," Networks, Vol. 13, pp.279-293, 1983.
17. A.Kershenbaum, R. Boorstyn and R.Oppenheim, "Second-Order Greedy Algorithms for Centralized Teleprocessing Network Design," IEEE Trans. on Comm., Vol. COM-28, No. 10, pp.1835-1838, 1980.
18. C.H.Papadimitriou, "The Complexity of the Capacitated Tree Problem," Networks, Vol. 8, pp.217-230, 1978.
19. R.Sharma, "Design of an Economical Multidrop Network Topology with Capacity constraints," IEEE Trans. on Comm., Vol. COM-31, No. 4, pp.590-591, 1983.
20. R.Signorile and A.Kershenbaum, "Multipoint Line Optimization using fuzzy set partitioning," Proceedings of GLOBECOM-86, pp.1571-1576, 1986.
21. 이 용진, 김 태윤, "근거리 통신망에서의 CMST 알고리즘," 정보과학회 논문지, 제21권, 제6호, pp.1097-1106, 1994.

李 湧 振(Yong Jin Lee) 정회원

현재 : 중경산업대학교 컴퓨터과학과 교수
한국통신학회 논문지 제19권 제7호 참조

金 泰 潤(Tai Yun Kim) 정회원

현재 : 고려대학교 전산과학과 교수
한국통신학회 논문지 제20권 제2호 참조