

링 구조의 데이터 통신망에서의 라우팅 방안 Routing Algorithms on a Ring-type Data Network

주운기

선문대학교 지식정보산업공학과

충남 아산시 탕정면 갈산리 100, ugjoo@sunmoon.ac.kr

Abstract

This paper considers a routing problem on a RPR(Resilient Packet Ring). The RPR is one of the ring-type data telecommunication network. Our major problem is to find an optimal routing algorithm for a given data traffic on the network under no splitting the traffic service, where the maximum load of a link is minimized. This paper characterizes the Minmax problem and develops two heuristic algorithms. By using the numerical comparison, we show that our heuristic algorithm is valuable for efficient routing the data traffic on a RPR.

1. 서론

인터넷과 같은 데이터 망에서 서비스 제공자는 가입자에게 서비스를 제공하기 위해 다수의 링크를 통합하여 하나의 POP(Point Of Presence)와 인터페이스를 하는데, 통합을 위해 사용 가능한 방법으로는 모든 노드를 직접 점대점 방식으로 연결하는 POS(Packet Over SONET) 방식, GE(Gigabit Ethernet)이나 FE(Fast Ethernet)에 대해 L2 스위치를 이용하여 노드를 연결하는 방법, 그리고, 링으로 노드를 연결하는 방법이 있다[1].

링으로 노드를 연결하는 방법에는 LAN 영역의 토큰 링, FDDI(Fiber Distributed Data Interface), 메트로 및 백본망의 SONET(Synchronous Optical NETwork), RPR(Resilient Packet Ring) 등이 있다. 토큰 링이나 FDDI는 송신 노드에서 한번 전송된 패킷은 수신 노드에서 수신된 후에도 송신 노드에서 제거될 때 까지 링 자원을 계속 점유하는 문제가 있다. 한편 SONET은 높은 신뢰성을 보장하지만, 음성 트래픽 전송에 최적화 되어 있어서 데이터와 같이 버스티한 트래픽 전송에는 비효율적이다. 반면, RPR은 이러한 문제점을 개선하고자 서로 반대 방향으로 데이터를 전달하는 2개의 링으로 구성되며, 주요 특징으로는 레질런시(resiliency), 공간 재사용, 노드 간 공평성 및 우선순위 서비스(priority) 등이 있다. RPR 관련 표준화는 IEEE 802.17 Resilient Packet Ring Working Group에서 수행하고 있는데, 시스코를 중심으로 결성된 Gandalf 진영의 Gandalf

draft와 노텔을 중심으로 결성된 Alladin 진영의 Alladin draft가 제출되어 경합을 벌여왔다. 이들 두 draft는 2002년 2월에 Darwin draft 0.1로 통합되어 제안되었고, 2004년 6월 현재 RPR의 표준 규격이 완성되었다[7]. RPR의 성능 상 이점으로는 이중 링의 사용으로 인한 대역폭의 효율적인 활용이 가능하고, 장애 시 복구가 용이하다는 장점이 있다. 그리고, plug-and-play 형태로 링 상의 노드 수가 총 64개가 될 때까지 새로운 노드의 추가나 제거가 용이하며, 다양한 유형(우선순위)의 데이터 서비스가 가능하고 새로운 데이터 서비스의 수용도 용이하다는 장점이 있다[7]. 그러나, 링 상의 대역폭을 여러 노드가 공동으로 이용하므로 효율적인 대역폭 할당 및 라우팅 방안이 필요하다.

링에서의 라우팅 문제는 트래픽 분할의 허용 여부에 따라 두 가지 유형의 문제로 구분할 수 있다. 분할 허용 문제(splitting problem)은 주어진 트래픽을 링 상의 두 개의 방향(시계 방향과 반 시계 방향)으로 각각 나눠서 서비스를 할 수 있는 경우의 문제를 말하고, 분할 불가 문제는(unsplit problem) 주어진 트래픽에 대해 링의 두 방향 중 하나의 방향을 통해서만 서비스 하는 경우를 말한다. 그리고, 이러한 분할 허용 트래픽 할당 문제는 요구된 트래픽이 정수 단위로만 분할되어야 하는 경우와 실수 값으로 분할될 수 있는 경우의 문제로 더 세분화시키기도 한다.

주어진 링의 부하(load)는 링 내의 가장 큰 부하를 가진 링크 용량으로 결정된다. 따라서, 링의 용량 증대는 망의 설치 비용 증대를 유발하므로 가능한 한 최대 부하를 가진 링크의 부하 값을 작게하기 위한 라우팅 방안이 필요하다. 이를 최소-최대(Min-max) 라우팅 문제라고 한다. 즉, 최소-최대 라우팅 문제는 주어진 링에서 서비스가 요구되는 트래픽에 대해, 최대 링크 부하를 갖는 링크의 부하 값을 최소화 시키는 라우팅 방안을 찾는 문제이다.

분할 불가 최소-최대 문제에 대해, Cosares and Saneiee[2]은 SONET에서의 최소-최대 라우팅을 위한 발견적 해법을 제안하였다. 이는 분할 불가 문제에 대한 것으로, 같은 문제에 대해 Dell'Amico et al.[3]는 분지한계법(branch-and-bound method)를 제안하였다.

분할허용 문제에 대해서는 Myung et al[6]과 Wan and Yang[8]이 SONET에서의 최적 라우팅 방안을 제안하였다. SONET은 두 인접 장치 간에 두

개의 광섬유를 설치하는 경우와 4개의 광섬유(fiber)를 설치하는 경우를 구분할 수 있다. SONET 할당에 대한 기존 연구의 대부분은 2-광섬유 문제를 다루었고, 4-광섬유 문제는 Wan and Yang[8]이 다루었다. 본 논문은 양 방향 분할불가 링 망에서의 트래픽 라우팅 문제를 다룬다.

2. 문제 설명

링 구조의 통신망은 N개의 노드와 인접 노드 간에만 직접적인 가지가 존재하는 네트워크 표현된다. 주어진 링에 대해, 임의의 장치를 1번이라고 칭하고, 시계 방향으로 N개의 모든 장치에 번호를 부여하였다고 가정하자. 이 경우 인접한 두 노드 i와 i+1 간에는 시계 방향의 가지 (i,i+1)와 반시계 방향의 가지 (i+1, i)가 존재한다.

주어진 망을 이용하고자 하는 K개 유형의 트래픽을 고려하자. 각 트래픽 k는 서비스 요구량이 d_k 이고, 송신처가 O_k 이고 수신지는 D_k 인 것으로 알려져 있다고 하자. 여기서, 각 트래픽의 구분은 송신처와 수신처 (O_k, D_k)가 다른 지 여부로 구분되며, 트래픽의 방향성이 중요한 역할을 하므로 트래픽 (O_k, D_k)와 트래픽 (D_k, O_k)은 다른 트래픽으로 간주하며, 데이터 통신망의 경우는 트래픽은 일반적으로 bps(Bytes Per Second)로 표시하므로 d_k 는 실수(real number)라고 가정한다.

각 트래픽은 시계방향 링과 반시계 방향의 링의 2중 하나를 통해 서비스할 수 있고, 총 K개의 트래픽 유형이 존재하므로 총 2^K 가지의 라우팅 방안이 존재한다. 이중 최선의 라우팅을 찾는 것이 본 연구의 목적이다.

라우팅 문제의 모형화를 위해 다음의 기호를 먼저 정의한다.

x_k : 트래픽 k를 시계방향 링으로 서비스하면 1의 값을 가지고, 반시계 방향의 링으로 서비스를 하면 0 값을 갖는 0-1 변수, $k=1,2,\dots, K$

$L = \{1,2,\dots, N\}$: 주어진 링의 인접노드 간의 가지 집합으로, 두 인접 노드 장치 i과 i+1 간에 존재하는 시계 방향 가지 (i,i+1)와 반 시계 방향 가지인 (i+1,i)을 동일한 링크 l라고 표시

L_k^+ : 트래픽 k에 대한 시계 방향 링을 구성하는 링크 집합

$L_k^- = L - L_k^+$: 트래픽 k에 대한 반 시계 방향 링을 구성하는 링크 집합

K_l^+ : 링크 l을 시계 방향으로 이용할 수 있는 트래픽 집합

$K_l^- = K - K_l^+$: 링크 l을 반 시계 방향으로 이용할 수 있는 트래픽 집합

z : 링 용량

양 방향으로 서비스가 가능한 링 망에서의 최적 라우팅 방안을 찾기 위한 문제는 다음과 같이 모형 [P1]으로 모형화 할 수 있다.

$$\text{Min } z \quad [P1]$$

$$\text{s.t. } z \geq \sum_{k \in K_l^+} d_k x_k, \quad l=1,2,\dots, L$$

$$z \geq \sum_{k \in K_l^+} d_k (1 - x_k), \quad l=1,2,\dots, L$$

$$x_k = 0 \text{ 또는 } 1, \quad k=1,2,\dots, K$$

여기서, 목적함수 식은 각 링크의 부하 중 최대값인 링 부하 값 z 를 최소화함을 나타낸다. 첫 번째 제약식의 오른쪽 항은 가지 l에서의 시계 방향 부하량을 표시하고, 두 번째 제약식의 오른쪽 항은 반시계 방향의 부하량을 나타낸다. 그리고, 모형 [P1]의 마지막 제약식은 주어진 트래픽 d_k 은 시계 방향 링이나 반 시계 방향의 링의 둘 중 하나로만 서비스되어야 함을 나타낸다. 만약, 두 방향으로 일부분씩 나눠서 서비스가 가능한 분할허용 문제를 위해서는 세 번째 제약식의 x_k 가 0과 1사이의 임의의 실수 값을 가지도록 변경하면 된다.

3. 라우팅 해법

분할이 허용되지 않는 문제는 NP-complete임이 밝혀졌다[8]. 따라서, 발견적 해법(heuristic algorithm)을 개발하는 것이 현실적이다.

발견적 해법 1 (H1) : 각 트래픽 k에 대해, 만약 $2(D_k - O_k) > N$ 이거나 $N > 2(O_k - D_k) > 0$ 이면, $x_k=0$ 으로 하고, 그렇지 않으면, $x_k=1$ 로 둔다.

발견적 해법 H1은 주어진 각 트래픽에 대해 시계 방향 링과 반 시계 방향 링 중 경로 상 노드 장치 수가 적은 경로를 통해 서비스를 해주는 방안이다. 이 라우팅 방안은 간단하게 구현할 수 있고 빠른 라우팅을 실현할 수 있으므로 트래픽의 변화가 많은 환경에서 유용하게 사용될 수 있을 것으로 예측된다. 그러나, H1은 요구되는 트래픽 양인 d_k 를 고려하지 않는 것이므로 전체 링 차원의 부하 평균화 및 자원의 효율적 활용 면에서는 양호한 가능성이 많을 것으로 예상할 수 있다. 따라서, 트래픽을 고려하는 라우팅 방안이 필요하다.

앞 장에서 본 바와 같이 링의 부하는 트래픽 요구량 $\{d_k\}$ 와 라우팅 방안 $\{x_k\}$ 로 결정된다. 모형 [P1]의 최적 라우팅을 하기위해 분할 허용 문제를 고려하는데, 이를 모형 [P2]라고 하자. 모형 [P2]는 모형 [P1]에서 제약식을 확장한 형태이므로, 모형 [P2]의 최적해는 모형 [P1]의 최적해에 대한 하한값(lower bound)을 형성한다. 그리고, 모형 [P2]는 선형계획법(Linear Programming) 문제이므로, 최적해를 용이하게 구할 수 있으나, 라우팅 문제는 링 설계 문제의 일부 문제일 뿐이고, 라우팅의 필요성은 트래픽의 변화나 링의 변화에 따라 빈번하게 요구되므로 효율적인 해법의 필요성은 매우 크다고 할 수 있다. 모형 [P2]는 다음과 같은 성질을 가지고 있다.

성질 1. 링 부하 z 는 총 시계방향 흐름량

$$\sum_{k=1}^K d_k x_k \text{에 대해 오목함수(convex function)이다.}$$

증명. Wang and Yang[8]의 증명을 참조바람.

성질1은 링의 부하 z 의 특성을 밝힌 것으로, z

가 $\sum_{k=1}^K d_k x_k$ 에 대해 오목 함수이므로 0에서부터 $\sum_{k=1}^K d_k$

까지의 값 중 z 값을 최소로 하는 $\{x_k\}$ 가 존재한다는 의미이다. 그러나, 특정한 총 시계 방향 흐름량을 구성하는 서비스 방안 $\{x_k\}$ 이 매우 많이 존재하므로 이러한 $\{x_k\}$ 를 직접적으로 찾는 것이 어렵다. 따라서, 다른 성질의 규명이 필요하다.

라우팅 방안을 위해 추가적으로 다음의 기호를 정의하자.

$R_l^+ = \sum_{k \in K_l^+} d_k x_k$: 가지 l에 부과된 시계 방향 트래픽의 합

$R_l^- = \sum_{k \in K_l^-} d_k (1 - x_k)$: 가지 l에 부과된 반 시계 방향 트래픽의 합

$$\delta_i = \max_{1 \leq l \leq N} \{R_l^+\} - \max_{1 \leq l \leq N} \{R_l^-\} = \max_{1 \leq l \leq N} \left\{ \sum_{k \in K_l^+} d_k x_k^i \right\} - \max_{1 \leq l \leq N} \left\{ \sum_{k \in K_l^-} d_k (1 - x_k^i) \right\}$$

여기서 x_k^i 는 i 번째 반복 상태의 x_k 값을 나타냄

$$\Delta_i = \min_{1 \leq j \leq K} \{\delta_i/2, d_j\}$$

성질 2. 모형 [P2]를 위한 모든 i 에 대해 $\delta_i \geq \delta_{i+1}$ 가 성립한다. 여기서, δ_{i+1} 를 위한 $\{x_k^{i+1}\}$ 는 $\Delta_i = \min\{\delta_i/2, d_j\}$ 인 트래픽 유형 j 에 대해서만 $x_j^{i+1} = x_j^i - \Delta_i/d_j$ 의 관계를 가지고 변경되고, 나머지는 $x_k^{i+1} = x_k^i$ 이다

$$\text{증명. } R_l^+ = \sum_{k \in K_l^+} d_k x_k^i \text{ 와 } R_l^- = \sum_{k \in K_l^-} d_k (1 - x_k^i)$$

및 $\Delta_i = \min\{\delta_i/2, d_j\}$ 에 대해, $\delta_i = \max_{l \in L_j^+} \{R_l^+\} - \max_{l \in L_j^-} \{R_l^-\}$ 이고, $\delta_{i+1} = \max_{l \in L_j^+} \{R_l^+\} - \Delta_i - \max_{l \in L_j^-} \{R_l^-\}$ 이다. 따라서, δ_i 와 δ_{i+1} 의 4가지의 대소관계에 대해 모두 $\delta_i \geq \delta_{i+1}$ 이 성립함을 보임으로써 증명한다.

성질 2를 이용하여 다음과 같은 발견적 해법 H2를 생각할 수 있다.

발견적 해법 H2

단계 1. <초기 가능해를 설정>

$i=1$ 로 두고, 모든 k 에 대해 $x_k^i=1$ 로 둔 후, 각 가지

에 대해 $R_l^+ = \sum_{k \in K_l^+} d_k$ 을 계산하고 $R_l^- = 0$ 으로 둔다.

단계 2. <개선 가능성 검토>

δ_i 값을 계산하여, 만약 $\delta_i > 0$ 이면 단계 3으로 가고, 그렇지 않으면 단계 4로 간다.

단계 3. <해의 개선>

(1) $\max_{1 \leq l \leq N} \{R_l^+\}$ 가 발생하는 가지 l_1 을 찾고, 가지 l_1 에서의 최대 시계 방향 흐름을 가지는 트래픽 유형 j 를 찾는다. 즉, $R_{l_1}^+ = \max_{1 \leq l \leq N} \{R_l^+\}$ 이고 $j = \arg \max_{k \in K_{l_1}^+} \{d_k x_k^i\}$ 임.

(2) 위에서 찾은 트래픽 j 에 대해, $\Delta_i = \min\{\delta_i/2, d_j\}$ 을 계산하여 $x_j^{i+1} = x_j^i - \Delta_i/d_j$ 로 두고, 나머지 트래픽에 대해서는 현 상태의 값을 유지한다.

(3) 각 가지 l에 대한 부하량을 다음과 같이 갱신한다: $l \in L_j^+$ 인 가지에 대해서는 $R_l^+ = R_l^+ - \Delta_i$ 로 하고, $l \notin L_j^+$ 인 가지에 대해서는 $R_l^- = R_l^- + \Delta_i$.

(4) $i=i+1$ 로 하고, 만약 $i \leq K$ 이면 단계 2를 수행하고, $i=K+1$ 이면 단계 4를 수행한다.

단계 4. 현 상태의 해인 $\{x_j^K\}$ 에 대해, 모든 값이 정수라면 현 상태를 원하는 해로 선택하고 끝낸다. 그렇지 않으면, 단계 5를 수행한다.

단계 5. <실행가능해를 구함>

(1) 0-1 값을 갖지 않는 변수들의 집합을 S라고 둔다: $S = \{x_j^K \mid 0 < x_j^K < 1\}$.

(2) S에 속하는 각 트래픽 유형 k 에 대해 차례로, 다음과 같이 λ_1 과 λ_2 를 계산하여 만약 $\lambda_1 < \lambda_2$ 라면 $x_k^K=0$ 로 두고, 그렇지 않으면 $x_k^K=1$ 로 두어 실행가능해를 구한다.

$$\lambda_1 = \max_{l \in L_k^+} \{R_l^+\} - d_k x_k^K, \quad \lambda_2 = \max_{l \in L_k^-} \{R_l^-\} + d_k x_k^K$$

$$\lambda_2 = \max_{l \in L_k^+} \{R_l^+\} + d_k (1 - x_k^K), \quad \lambda_1 = \max_{l \in L_k^-} \{R_l^-\} - d_k (1 - x_k^K)$$

여기서, 각 가지의 부하량은 $\lambda_1 < \lambda_2$ 인 경우는 $R_l^+ = R_l^+ - d_k x_k^K$, $l \in L_j^+$; $R_l^- = R_l^- + d_k x_k^K$, $l \notin L_j^+$ 가 되고, $\lambda_1 \geq \lambda_2$ 인 경우는 $R_l^+ = R_l^+ + d_k (1 - x_k^K)$, $l \in L_j^+$; $R_l^- = R_l^- - d_k (1 - x_k^K)$, $l \notin L_j^+$ 가 된다.

발견적 해법 H2는 단계 3에서 성질 2를 이용한 해의 개선을 한다. 현 단계의 해인 $\{x_k^i\}$ 에서 트래픽 j 에 대한 시계방향 트래픽 중 Δ_i 만큼을 반 시계 방향으로 서비스 해줌으로써 $\max_{1 \leq l \leq N} \{R_l^+\} = \max_{1 \leq l \leq N} \{R_l^-\}$ 이거나 트래픽 j 의 모든 요구량 d_j 를 반시계 방향으로 서비스를 하게한다. 이러한 단계 3은 $\delta_i > 0$ 일 때만 수행되므로, $\delta_i > \delta_{i+1}$ 가 되어 결국 $\delta_i=0$ 인 상태에

이르게 된다. 단계 5에서는 집합 S내의 각 트래픽 k에 대해 d_k 를 모두 시계 방향으로 서비스 시 증가하는 링 부하량 λ_1 과 d_k 를 모두 반시계 방향으로 서비스 시 증가하는 링 부하량 λ_2 중 링 부하의 증가가 적은 형태로 분할 불가 시의 라우팅 방안을 구한다.

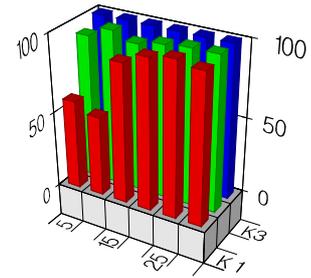
발견적 알고리즘 H2의 계산 복잡도를 결정하는 부분은 단계 3과 단계 5이다. 단계 3에서 각 트래픽 유형별 $O(N)$ 의 계산 복잡도를 가지고, K개의 트래픽 유형이 존재하므로 $O(N \cdot K)$ 의 계산 복잡도를 가진다. 그리고, 단계 5의 경우는 $O(2N)$ 인 트래픽 유형을 최대 K개 고려해야하므로 결국 발견적 해법 H2는 $O(N \cdot K)$ 의 계산 복잡도를 갖는다.

발견적 해법 H1과 H2간 비교를 위해 다양한 수치 예를 통한 평균적인 분석을 하였다. 이를 위해 링의 규모를 위한 N과 트래픽 규모에 따른 성능 평가를 위해 다음과 같은 다양한 조합의 (N,K) 값을 고려하였다 : (N, K) = (5,6), (5,8), (5,10), (10,12), (10,23), (10,45), (15,25), (15,50), (15, 105), (20,40), (20,95), (20,190), (25,60), (25,150), (25, 300), (30, 90), (30, 200), (30, 435). 각각의 (N,K)조합에 대해 10개의 문제 사례(instance)를 발생하였는데, 여기서 각 트래픽 k의 요구량 d_k 는 일양분포 $U[5, 100]$ 에서 임의로 추출하였고, 송신처와 수신처인 (O_k, D_k)도 일양분포 $U[1, N]$ 에서 임의로 추출한 값을 이용하였다. 두 가지 발견적 해법을 펜티엄 IV PC (1.0GHz, window XP)에서 Visual C/C++로 구현하여 적용한 결과는 (그림 1)~(그림 3)과 같다.

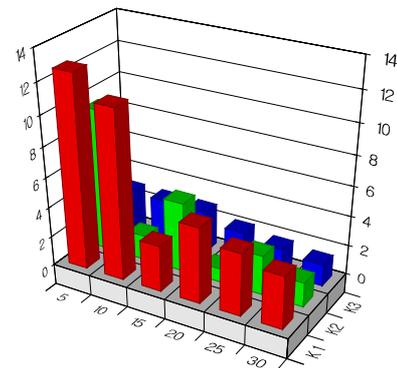
(그림 1)과 (그림 2)는 각각의 (N, K) 값에 따른 상대 편차값을 표시하였다. 상대편차는 문제 [P2]의 최적해는 문제 [P1]에 대한 최적해의 하한값(lower bound)을 형성하는 사실을 이용하여, 다음과 같은 방식으로 측정하였다.

상대적 편차 = (발견적 해법의 링 부하 - H2의 단계 3까지의 결과)/(H2의 단계 3까지의 결과)

각각의 N 값에 대한 3가지 유형의 K값(K1, K2, K3)의 조합에 대해 발생한 문제를 각 발견적 해법에 적용하여 링 부하를 계산한 결과, (그림 1)의 발견적 해법 H1의 평균 상대 편차는 93.44%이었고, (N, K)가 증대됨에 따라 H1의 상대 편차도 증대되는 경향이 있음을 확인할 수 있다. 그러나, 발견적 해법 H2의 경우는 (그림 2)에서 보이는 바와 같이 (N, K)의 증가에도 불구하고 상대 편차가 증가하지 않고, 평균적으로 4.15%의 상대 편차를 나타내었다. 계산 시간 측면에서는 (그림 3)에서 보는 바와 같이 H2의 경우는 (N, K)가 증가함에 따라 계산 시간이 H1에 비해 크다는 것을 알 수 있다. 그러나, (N, K)가 큰 경우라 하더라도 계산 시간이 0.3초 정도로 크지 않으므로, 발견적 해법 H2는 링의 라우팅 방안 설계를 위해 유용한 방안으로 판단할 수 있다. 계산 시간은 $O(n \cdot K)$ 에 비례하는 것을 앞에서 밝혔으나, (그림 3)은 (N,K)에 따라 지수적인 증가를 하는 모습을 보이고 있다. 그러나, 이는 횡축에 해당되는 (N,K) 값의 K 값이 등 비율로 증가하지 않고 지수적으로 증가된 데에 오는 착시 현상인 것으로 판단된다.



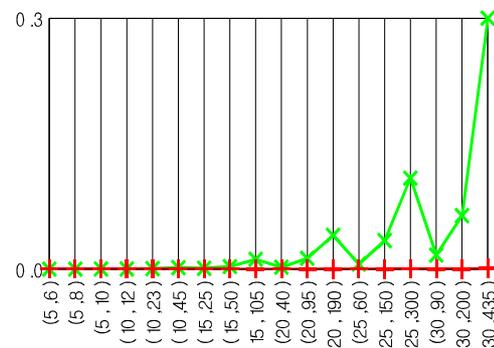
<그림 1> H1의 상대 편차



<그림 2> 해법 H2의 상대편차

4. 결론

본 논문은 RPR을 위한 라우팅 방식에 대한 것으로, 주어진 링 망의 이용률은 요구되는 트래픽을 어떻게 할당하는가에 매우 큰 영향을 받는다. 더구나, RPR의 성능 향상을 위한 주요 요소 중 하나인 공평성 할당 방식과 관련해서도, 적절한 트래픽 할당은 RPR 규격에서 권고하는 공평성 알고리즘



의 activation 기회를 줄여주는 효과가 있으므로, 트래픽 할당 문제는 링 망의 성능에 큰 영향

<그림 3> H1과 H2의 계산 시간 비교

을 미친다.

본 논문은 라우팅을 위한 척도로써 링 상의 할

당된 부하 중 최대 부하를 최소화하는 MinMax 기준에 대해 다루었고, 2 가지의 발견적 해법과 수치예를 통한 비교를 하였다. 이 비교 결과는 링 구조의 망에서의 효율적인 트래픽 할당과 링의 효율적 활용을 위해 본 논문의 H2 해법이 유용하게 이용될 수 있을 것으로 판단된다.

참고문헌

- [1] 남홍순 et al., "RPR 기술동향 및 향후 전망", 전자통신동향분석, 2002, 제17권 제5호, pp.55-67.
- [2] S. Cosares and I. Saniee, "An Optimization Problem related to Balancing Loads on SONET Rings", Telecomm. Sys., Vol.3, 1994, pp.165-181.
- [3] M. Dell'Amico et al., "Exact Solution of the SONET Ring Loading Problem", Oper. Res. Letters, Vol.25, 1999, pp.119-129.
- [4] V. Gambiroza et al., "High Performance Fair Bandwidth Allocation for Resilient Packet Rings", Proceedings of the 15th Specialist Seminar on Traffic Engineering and Traffic Management, 2002.
- [5] L. Massoulie and J. Roberts, "Bandwidth Sharing : Objectives and Algorithms", INFOCOM'99, 1999, pp.1395-1403.
- [6] Y.-S. Myung et al., "Optimal Load Balancing on SONET Bidirectional Rings", Oper.Res., Vol.45, 1997, pp.148-152.
- [7] RPR Alliance, A Summary and Overview of the IEEE 802.17 Resilient Packet Ring Standard, 2004.6.24.
- [8] P.-J. Wan and Y. Yang, "Load-Balanced Routing in Counter Rotated SONET Rings", Networks, Vol.35, 2000, pp.279-286.