

분할이 허용된 SONET 링의 루팅 해법들에 대한 비교 분석*

명영수** · 김후곤***

Experimental Evaluation of Routing Algorithms for Optimal
Load Balancing on SONET Rings with Demand Splitting*

Young-Soo Myung** · Hu-Gon Kim***

■ Abstract ■

In the ring loading problem with demand splitting, traffic demands are given for each pair of nodes in an undirected ring network and a flow is routed in either of the two directions, clockwise and counter-clockwise. The load of a link is the sum of the flows routed through the link and the objective of the problem is to minimize the maximum link load on the ring. Recently, four different optimal algorithms have been developed for solving the problem. In this paper, we compare those algorithms in a single framework and conduct a computational study to evaluate the practical performance of those algorithms.

Keyword : 분할이 허용된 SONET링의 루팅, 계산분석

1. 서 론

대용량, 고속 전송을 특징으로 하는 광통신망이

광범위하게 구축됨에 따라 통신망의 생존도는 매우 중요한 요소가 되고 있다. 즉 광케이블이나 광전송장비에 장애가 발생하게 되면, 대규모의 통신

논문접수일 : 2001년 4월 10일 논문게재확정일 : 2001년 8월 4일

* 이 논문은 2000년도 단국대학교 대학연구비에 의하여 연구되었음.

** 330-714 충남 천안시 안서동 산 29, 단국대학교 경상학부 Email : myung@anse.dankook.ac.kr

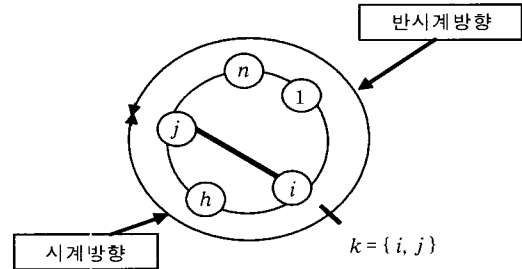
*** 경성대학교 경영학부

장애를 초래하므로 이는 개인이나 기업뿐만 아니라 국가적으로 매우 심각한 사태를 발생시킬 수 있다. 이러한 문제점을 고려하여 동기식 광전송망(SONET, Synchronous Optical Network) 기술이 제안되고 있다. SONET은 광전송에 필요한 신호 형식 및 이의 전송과 관련된 표준을 규정해 놓은 것으로, 이를 이용한 광통신망은 광전송 장비의 장애를 자동적으로 감지하여 단시간 내에 복구가 가능하다. 특히 동기식 광전송 장비인 ADM(Add-Drop Multiplexer)을 이용하면, 자동적인 망의 복구가 가능한 링(SHR, Self-Healing Ring)을 구성할 수 있는데 이는 현재 가장 많이 이용되는 광통신망 방식이다.

SHR은 모든 링에서 전송이 단방향으로만 이루어지는 형태(USHR, Unidirectional SHR)와 시계방향과 반시계방향의 양방향으로 가능한 형태(BSHR, Bidirectional SHR)가 있다. USHR 및 BSHR은 이를 구성하는 링크들이 모두 동일한 용량을 가져야 한다는 기술적 제약이 따르게 된다. 따라서 USHR의 경우는 신호를 보내는 방법, 즉 루팅(routing) 방법이 고정되어 있으므로 통신 수요가 주어지면, 수요를 처리하기 위하여 필요한 링의 용량이 바로 결정되나 BSHR의 경우는 각각의 통신 수요를 처리하는 루팅 방법에 따라 필요한 링의 용량이 달라지게 된다. 따라서 BSHR의 설계를 위해서는 링의 용량을 최소화하기 위한 최적의 루팅 방법을 구하는 링의 용량결정문제(RLP, Ring Loading Problem)가 중요한 과제이다[2, 11].

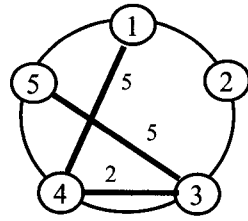
BSHR에서의 링의 용량결정문제는 다음과 같이 설명할 수 있다. 노드의 집합 $V = \{1, 2, \dots, n\}$ 와 링크의 집합 $L = \{(1, 2), (2, 3), \dots, (n-1, n), (n, 1)\}$ 로 이루어진 방향성이 없는 링 모양의 그래프 $R = (V, L)$ 이 주어져 있다고 가정하자. 링크 $(i, i+1)$ 과 링크 i 는 동일한 링크를 지칭하는 것으로 정의한다. 다만 링크 $(n, 1)$ 이 링크 n 으로 표시되는 것은 예외로 인정한다. K 는 두 노드, 즉 노드 쌍들의 인덱스 집합이다. 각 노드 쌍 $k \in K$ 에 대하여, r_k 의 통신 수요가 존재하고, $o(k)$ 와

$d(k)$ ($o(k) < d(k)$)는 노드 쌍 k 의 공급 노드와 수요 노드를 각각 표시하는 것으로 정의한다. $o(k)$ 와 $d(k)$ 사이의 흐름은 시계 방향과 반시계 방향의 양방향으로 보내질 수 있다. 다시 말해서 흐름이 $\{o(k), o(k)+1, \dots, d(k)-1, d(k)\}$ 방향으로 움직이면 시계 방향이라고 부르고, 흐름이 $\{o(k), o(k)-1, \dots, 1, n, \dots, d(k)+1, d(k)\}$ 방향으로 움직이면 반시계 방향이라고 부른다 (<그림 1> 참조). 링을 구성할 때는 모든 링크에 동일한 용량의 링크를 설치하여야 되기 때문에 링 구성에 필요한 용량은 링크 중에서 가장 많은 흐름이 통과하는 링크의 흐름량에 의하여 결정되게 된다. 링에서의 링크별 흐름의 양은 각 노드 쌍간의 흐름 수요를 어떻게 분산해서 루팅(routing)하는가에 의하여 결정되므로 링의 용량결정문제는 링의 용량이 최소가 되게 해주는 루팅 방법을 찾는 문제가 된다.



<그림 1> 링에서의 두 가지 루팅방법

RLP의 예가 <그림 2>에 나타나 있다. <그림 2>는 5개의 노드로 구성된 링에서 3개의 노드 쌍이 통신 수요를 가진 예를 보여 주고 있다. <그림 2>의 (a)는 5개의 노드와 링크로 구성된 링과 노드 쌍에 주어진 수요를 나타내고 있다. 왼쪽의 그림에서 노드 쌍을 연결하는 굵은 직선은 통신 수요가 존재하는 노드 쌍 {1,4}, {3,4}, {3,5}를 나타내고, 굵은 직선 위의 숫자는 이들 노드 쌍의 통신 수요를 나타내고 있으며, 동일한 내용이 오른쪽의 표에도 표시되어 있다. <그림 2> (b)의 표는 노드 쌍 {1,4}의 수요를 시계 방향으로, 노드 쌍 {3,5}의



k	$\{o(k), d(k)\}$	r_k
1	{1,4}	5
2	{3,5}	5
3	{3,4}	2

(a)

k	루팅방법	링크				
		(1,2)	(2,3)	(3,4)	(4,5)	(5,1)
{1,4}	시계	5	5	5		
{3,5}	반시계	5	5			5
{3,4}	양쪽	1	1	1	1	1
		11	11	6	1	6

링의 필요 용량=11

(b)

〈그림 2〉 루팅 방법에 따른 링의 필요용량

수요는 반시계 방향으로, 그리고 노드 쌍 {3,4}의 수요는 양방향으로 1씩 보내는 루팅 방법을 선택하는 경우에 필요한 링의 용량이 어떻게 계산되는지를 보여주고 있다. 표에서 음영으로 처리된 칸은 각 노드 쌍에 대해서 시계 방향에 속한 링크들을 표현한다.

양방향으로 루팅이 가능한 링에서는 두 노드간의 수요를 분할해서 동시에 양방향으로 전송할 수 있는 경우와 양방향 중 시계 방향이나 반시계 방향의 어느 한 방향으로만 전송해야 하는 경우로 나누어진다. 이러한 조건에 따라 링의 용량결정문제도 분할 루팅이 가능한 경우와 가능하지 않은 경우로 구분한다. 전자의 경우인 분할 루팅이 허용되는 링에서의 용량결정문제를 RLPW(RLP With demand splitting)로, 후자의 경우인 분할 루팅이 허용되지 않는 링에서의 용량결정문제를 RLPWO(RLP WithOut demand splitting)로 부르기로 한다. 분할이 허용되는 경우도 전송은 꼭 정수 단위 이어야 되는 경우도 있다. 왜냐하면 링크의 용량이 단위 용량의 정수 배로서만 설치될 수 있는 경우를 고려하기 때문이다. 분할이 정수 단위로만 허용

되는 문제를 RLPWI(RLP With Integer demand splitting)로 부르기로 한다. RLPWO는 NP-hard 문제임이 증명되었고[3, 6], RLPW와 RLPWI는 다항시간(polynomial time) 내에 풀 수 있는 해법이 존재한다[6, 7, 9, 10].

RLPW는 선형계획문제로, RLPWO는 0-1 정수 계획문제로, RLPWI는 일반적인 정수계획문제로 각각 표현할 수 있다[3, 6, 7]. 또한 RLPW는 RLPWO와 RLPWI의 완화문제(relaxation)가 되는 속성 때문에 RLPW의 최적해는 RLPWO와 RLPWI의 최적해의 하한(lower bound)이 된다. 실제로 RLPWO의 최적해를 구하기 위해서는 분단탐색법(branch and bound method)이 필요하며 분단탐색법의 성공적인 수행을 위해서는 좋은 하한이 필요하다. 따라서 RLPWO의 성공적 해결은 RLPW를 푸는 효율적 해법의 존재여부에 달려 있다고 할 수 있다. RLPWI의 경우는 정수계획문제이나 다항시간 내에 풀 수 있는 해법이 개발되었다[7, 9, 10]. RLPWI의 해법들은 서로 다른 RLPW의 해법을 일부 변형하여 만들어졌기 때문에 이들 해법의 효율성은 각 해법들이 근거한 RLPW의 해법의 효율성과 밀

〈표 1〉 RLP에 대한 기존의 해법들

연구	대상문제	RLPW	RLPWI	RLPWO
Cosares and Saniee [3]		휴리스틱		휴리스틱
Vachani 등 [10]		$O(n^3)$	$O(n^3)$	
Myung 등 [6]		$O(n K)$		휴리스틱
Schrijver 등 [9]		$O(n^2 K)$	pseudo-polynomial	휴리스틱
Dell'Amico 등 [4]		$O(n^3)$		
Myung [1, 7]			$O(n K)$	분단탐색법

접하게 연관되어 있다.

링의 용량결정문제는 현실적인 중요성 때문에 많은 연구가 이루어져 왔다. 이에 대한 요약표가 <표 1>에 나타나 있다. RLPWO의 경우를 살펴보면 Cosares와 Saniee[3] 및 Myung 등[6]이 휴리스틱을 제시하였다. RLPW에 대해서는 Cosares와 Saniee[3]가 휴리스틱을, Vachani 등[10]과 Dell'Amico 등[4]이 각각 $O(n^3)$ 최적해법을, Schrijver 등[9]이 $O(n^2|K|)$ 최적해법을 개발하였으나, 최근에 발표된 Myung 등[6]의 최적해법이 $O(n|K|)$ 만에 문제를 풀 수 있어 이론적인 계산시간에서는 가장 빠른 해법으로 나타나 있다. RLPWI에 대해서는 Vachani 등[10]이 $O(n^3)$ 에 계산되는 최적해법을 제시하였고, Schrijver 등[9]은 노드간의 통신 수요가 1로 주어지는 특수한 경우에 최적해를 $O(n^2|K|)$ 에 계산할 수 있는 해법을 제시하였는데 이를 이용하면 일반적인 RLPWI를 pseudo-polynomial 해법을 만들 수 있다. Myung[1, 7]은 Myung 등[6]의 분할조건이 없는 문제의 최적해법을 이용하여 같은 시간, 즉 $O(n|K|)$ 계산시간 내에 최적해를 구할 수 있는 해법을 제시하였다.

여러 가지 링의 용량결정문제들 중에서 RLPW는 이 문제 자체로도 응용성이 높지만, 앞서 언급한 대로 이 문제가 RLPWO와 RLPWI를 푸는 해법에도 사용된다는 점에서 그 중요성이 더욱 크게 인식되어 왔다. 이미 살펴본 대로 RLPW에 대해서는 Vachani 등의 해법, Schrijver 등의 해법, Myung 등의 해법 및, Dell'Amico 등의 해법이 발표되었는데, 흥미로운 것은 상당수의 해법들이 서로 독립적으로 개발되었다는 것이다. 이 중에서

Vachani 등의 해법은 Schrijver 등의 해법과 기본적으로 동일한 해법이다. Schrijver 등의 해법은 $O(n^2|K|)$, Vachani 등의 해법은 $O(n^3)$ 의 계산시간이 소요되는 것으로 나타나 있으나, 링의 용량결정문제는 간단한 변환과정을 거쳐 수요가 존재하지 않는 노드를 제거하여 $|K| \geq n/2$ 이 되도록 표현할 수 있기 때문에 Schrijver 등의 해법도 이론적 계산시간은 실제로 $O(n^3)$ 이다.

이처럼 RLPW의 해법에 대한 연구들이 서로 독립적으로 이루어졌기 때문에 개발된 해법들이 실제 문제를 풀 때는 어떠한 성능을 보이는지에 대해서는 비교가 이루어지지 못하였다. 본 논문에서는 위의 해법들 중에서 구조가 서로 다른 Vachani 등의 해법, Dell'Amico 등의 해법 및, Myung 등의 해법의 특성을 비교 분석한다. 이론적인 계산의 효율성 면에서는 언급한 대로 Myung 등의 해법이 가장 우수한 것으로 나타나 있으나, 실제 문제를 푸는 계산실험에서는 해법간에 어떠한 성능의 차이를 보이는지 알아보기로 한다. 이를 위해서 다양한 테스트 문제에 각 해법의 컴퓨터 code를 실행시킨 계산시간을 비교하기로 한다. 본 논문은 다음과 같이 구성되어 있다. 2장에서는 사용된 기호를 정의하고 문제의 특성을 분석하며, 3장에서는 세 가지 해법의 특성을 비교 분석하고, 4장에서 계산실험을 통하여 해법의 계산능력을 비교 분석한다.

2. 용어의 정의 및 문제의 특성

각 노드 쌍 $k \in K$ 에 대해 공급 노드 $o(k)$ 로부터 수요 노드 $d(k)$ 까지의 시계 방향으로의 경로

상에 존재하는 링크의 집합을 L_k^+ 로, 반시계 방향으로의 경로 상에 존재하는 링크의 집합을 L_k^- 로 정의한다. 즉, $L_k^+ = \{(i, i+1) \in L \mid o(k) \leq i < d(k)\}$ 이고 $L_k^- = L \setminus L_k^+$ 이다. 따라서 링크 $n = (n, 1)$ 은 항상 모든 $k \in K$ 에 대하여 L_k^- 에 포함된다. 또한 모든 링크 $l \in L$ 에 대해 링크 l 을 시계 방향의 경로에 포함하는 노드 쌍들의 집합을 K_l^+ 로, 반시계 방향의 경로에 포함하는 노드 쌍들의 집합을 K_l^- 로 정의한다. 즉, $K_l^+ = \{k \in K \mid l \in L_k^+\}$ 이고 $K_l^- = \{k \in K \mid l \in L_k^-\}$ 이다. 또한 $K_l^- = K \setminus K_l^+$ 임을 쉽게 알 수 있다.

각 노드 쌍 $k \in K$ 에 대해 변수 x_k 를 이용하여 $o(k)$ 와 $d(k)$ 사이의 수요 중 시계 방향으로 루팅되는 흐름의 양을 표시하기로 하자. 그리고 $X = \{x \in R^{|K|} \mid 0 \leq x_k \leq r_k, \forall k \in K\}$ 를 정의하면 임의의 $x \in X$ 는 주어진 수요를 만족하는 루팅 방법에 대응되는 벡터가 된다. 주어진 루팅방법 $x \in X$ 에 대해,

$$g(x, l) = \sum_{k \in K_l^+} x_k + \sum_{k \in K_l^-} (r_k - x_k), \quad l \in L$$

이라고 정의하면 $g(x, l)$ 은 x 에 따라 통신 수요를 루팅할 때 링크 l 을 통과하는 흐름의 합을 의미한다. 각각의 링크 $l \in L$ 에 대해 용량이 c_l 로 주어져 있다고 가정하자. 그러면 링크 l 에서의 흐름의 합은 링크의 용량을 초과할 수 없으므로 모든 링크 $l \in L$ 에 대해서 $g(x, l) \leq c_l$ 이 만족되어야 한다.

링의 용량결정문제는 주어진 수요를 만족하기 위하여 필요한 링크의 용량 중 최대용량을 최소화하는 루팅 방법을 결정하는 문제이므로 RLPW는 다음과 같이 정식화 된다.

$$\begin{aligned} (RLPW) \quad z = \min \max_{l \in L} c_l \\ \text{s.t. } g(x, l) \leq c_l, \quad l \in L \\ x \in X. \end{aligned}$$

(RLPW)는 다음과 같이 선형계획모형으로도 표현할 수 있다.

$$\begin{aligned} z = \min t \\ \text{s.t. } t \geq g(x, l), \quad l \in L \\ x \in X. \end{aligned}$$

이제부터는 (RLPW)의 해법들에 이용된 (RLPW)의 특성을 분석해 보기로 하자. 임의의 두 링크 i 와 j 는 링에서 컷(cut)을 구성하는 데 이러한 컷 $\{i, j\}$ 에 대하여 D_{ij} 를 다음과 같이 정의하기로 하자.

$$D_{ij} = \sum \{r_k \mid i \in L_k^+ \text{이고 } j \in L_k^-, \text{ 또는 } i \in L_k^- \text{이고 } j \in L_k^+\}.$$

즉 D_{ij} 는 컷 $\{i, j\}$ 를 통과해야만 만족시킬 수 있는 노드 쌍간의 수요량의 합이며, D_{ij} 를 이용하여 (RLPW)가 실행가능하기 위한 필요충분조건을 다음과 같이 유도할 수 있다.

정리 1 (RLPW)의 실행가능해가 존재하기 위한 필요충분조건은 모든 컷 $\{i, j\}$ 에 대해서 $c_i + c_j \geq D_{ij}$ (컷-조건이라고 부르기로 한다)가 성립하는 것이다.

위 정리의 필요조건은 임의의 루팅 방법 $x \in X$ 와 임의의 두 링크 i 와 j 에 대해서

$$\begin{aligned} D_{ij} = g(x, i) + g(x, j) - 2 \sum_{k \in K_i^+ \cap K_j^+} x_k \\ - 2 \sum_{k \in K_i^- \cap K_j^-} (r_k - x_k) \end{aligned} \quad (1)$$

이므로 $g(x, i) + g(x, j) \geq D_{ij}$ 가 성립한다는 사실에서 쉽게 알 수 있다. 충분조건은 Okamura와 Seymour의 정리 [8]의 특수한 경우로 성립하게 되는데, Vachani 등 [10]과 Schrijver 등[9]은 별도의 단순화한 증명을 소개하였다.

정리 1로부터 우리는 다음과 같은 사실을 알

수 있다. 링 내의 모든 컷들의 D_{ij} 값 중 최대값을 D_{\max} 라고 하자. 그러면 (1)의 관찰에 의해서 어떠한 실행가능한 루팅 방법 $x \in X$ 에 대하여도 $z = \min \max_{l \in L} c_l \geq \min \max_{l \in L} g(x, l) \geq D_{\max}/2$ 가 성립함을 알 수 있다. 또한 우리가 모든 링크 $l \in L$ 에 대해서 $c_l = D_{\max}/2$ 로 정하면 $\{c_l\}$ 은 정리 1의 충분조건을 만족하므로 $z \leq D_{\max}/2$ 도 성립되어, 결국 $z = D_{\max}/2$ 가 된다. 따라서 우리는 D_{\max} 를 계산함으로써 (RLPW)의 최적해의 목적함수 값을 구할 수 있게 되는데 이러한 사실이 Vachani 등의 해법과 Dell'Amico 등의 해법의 근본을 이루게 된다.

3. 해법들의 비교 분석

3절에서는 (RLPW)의 최적해를 구하는 Vachani 등의 해법, Dell'Amico 등의 해법 및, Myung 등의 해법을 비교 분석하기로 한다. 본 절의 의도는 상세한 해법의 소개는 아니고, 2절에서 소개한 내용을 이용하여 해법들의 특성을 하나의 체계 속에서 비교하는 것이다. 따라서 각 해법의 자세한 과정은 저자들의 논문[4, 6, 10]을 참조하기 바란다.

Vachani 등의 해법은 우선 D_{\max} 를 계산함으로써 (RLPW)의 최적해의 목적함수 값을 구하고 이 목적함수 값을 구현할 수 있는 루팅 방법을 찾는 방식으로 설계되었다.

Vachani 등의 해법

(단계 1) D_{\max} 를 계산한 다음에 모든 링크 $l \in L$ 에 대해서 $c_l = D_{\max}/2$ 로 초기용량을 정한다.

(단계 2) 모든 컷 $\{i, j\}$ 에 대해서 $c_i + c_j \geq D_{ij}$ 를 만족시키는 범위 안에서 각 링크의 용량을 최대로 감소시킨다.

(단계 3) 정해진 용량을 만족시키는 루팅 방법을 구한다.

단계 1은 D_{\max} 의 정의에서 $O(n^2)$ 의 계산으로 수행할 수 있음을 쉽게 알 수 있다. 이 해법에서 가장 많은 계산을 요하는 단계는 단계 2인데 Vachani 등 [10]은 다음과 같은 반복과정을 통해서 단계 2를 $O(n^3)$ 시간 내에 수행할 수 있음을 보였다: 이제까지의 반복과정에서 고려하지 않은 컷들 중에서 $\Delta_{ij} = c_i + c_j - D_{ij}$ 가 최소인 컷을 선택하고 컷을 이루는 링크 중 용량 조정이 이루어지지 않았던 링크의 용량을 조정한다.

Dell'Amico 등의 해법은 Vachani 등의 해법과 비슷하나 루팅 방법과 용량을 조정하는 방법에서 차이가 난다.

Dell'Amico 등의 해법

(단계 1) D_{\max} 를 계산한 다음에 모든 링크 $l \in L$ 에 대해서 $c_l = D_{\max}/2$ 로 초기용량을 정한다.

(단계 2) 각 노드 쌍 $k \in K$ 에 대해서 노드 쌍 k 의 수요를 흘려 보낸 뒤에 남은 링크의 용량과 남아있는 수요가 모든 컷 $\{i, j\}$ 에 대해서 여전히 컷-조건을 만족시킬 수 있도록 노드 쌍 k 의 루팅 방법을 선택한다.

Dell'Amico 등 [4]은 단계 2의 반복과정에서 조건을 만족시키는 루팅 방법이 항상 존재하며 단계 2를 $O(n^3)$ 시간 내에 수행할 수 있음을 보였다. Dell'Amico 등과 Vachani 등의 해법의 차이는 전자가 노드 쌍별로 반복과정을 통해서 루팅 방법을 정해 나가는데 반해서 후자는 컷별로 반복과정을 통해서 용량을 사전에 조정한 뒤에 루팅 방법을 정해 나간다는 것이다.

Myung 등의 해법은 D_{\max} 나 링크의 용량을 이용하지 않는다는 점에서 Dell'Amico 등과 Vachani 등의 해법과는 근본적으로 차이가 난다.

Myung 등의 해법

(단계 1) K 에 속한 노드 쌍의 순서를 다음과 같이

재배열한다. k_i 를 K 에 속한 i 번째 노드 쌍이라고 할 때 $o(k_i) < o(k_j)$ 이면 $k_i < k_j$ 되게 하고, $o(k_i) = o(k_j)$ 인 경우에는 $d(k_i) > d(k_j)$ 이면 $k_i < k_j$ 되도록 한다.

(단계 2) 우선 모든 노드 쌍간의 수요를 시계 방향으로 루팅한다.

(단계 3) 각 노드 쌍 $k \in K$ 에 대해 배열된 순서로 노드 쌍 k 의 수요 중 일부 또는 전부를 링의 최대 용량이 감소되도록 반시계방향으로 다시 루팅을 변경한다.

단계 1의 재배열은 $O(|K| \log |K|)$ 또는 $O(|K|)$ 계산으로, 단계 2의 초기해의 설정은 $O(|K|)$ 계산으로 가능하다. 어떤 노드 쌍 $k \in K$ 에 대한 단계 3의 반복과정에서 어떤 노드 쌍 $k \in K$ 의 최대 흐름의 링크들이 모두 노드 쌍 k 에 대해 시계 방향의 경로에 속하면 현재 시계 방향으로 루팅되어 있는 노드 쌍 k 의 수요를 반시계 방향으로 재루팅함으로써 링의 용량을 감소시킬 수 있다. Myung 등 [6]은 단계 3의 반복과정이 $O(n|K|)$ 시간 내에 간단히 수행될 수 있음을 보였고 이러한 방법으로 정해진 루팅 방법은 링크의 용량을 D_{\max} 이

내로 사용함으로써 최적해임을 보였다.

위의 세 해법의 설명에서 나타난 것처럼 이론적인 계산시간은 Dell'Amico 등과 Vachani 등의 해법은 $O(n^3)$ 으로 동일하고 Myung 등의 해법은 $O(n|K|)$ 이다. 따라서 수요가 발생하는 노드 쌍이 많지 않을 경우에는 Myung 등의 해법이 이론적인 계산시간에서 가장 우수하나, 수요가 모든 노드 쌍에 존재한다면 $|K| = \binom{n}{2}$ 이므로 Myung 등의 해법도 $O(n^3)$ 의 소요시간이 필요하다. 이론적인 계산시간과는 별개로 실제 문제를 풀 때 각 해법의 계산능력이 어떻게 나타날지도 흥미로운 내용이며, 특히 현실적인 응용 면에서는 더욱 관심이 가는 중요한 사항이다. 다음 장에서는 여기서 소개된 세 가지 해법에 대해서 다양한 테스트 문제를 통하여 각 해법의 실질적 계산능력을 비교하려고 한다.

4. 계산 결과의 비교 분석

Vachani 등의 해법, Dell'Amico 등의 해법 및, Myung 등의 해법이 실제 계산에서는 어떠한 계산 결과를 산출하는 지 알아보기 위하여 다양한 테스트

〈표 2〉 노드 쌍간의 통신 수요 범위가 [5,100]인 경우의 계산 결과

문제크기		VACHANI			DAMICO			MYUNG		
n	K	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
20	50	0.0030	0.0000	0.0600	0.0030	0.0000	0.0600	0.0000	0.0000	0.0000
100		0.0025	0.0000	0.0500	0.0025	0.0000	0.0500	0.0000	0.0000	0.0000
150		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
40	100	0.0135	0.0000	0.0600	0.0190	0.0000	0.0600	0.0000	0.0000	0.0000
200		0.0175	0.0000	0.0600	0.0185	0.0000	0.0600	0.0025	0.0000	0.0500
400		0.0165	0.0000	0.0600	0.0360	0.0000	0.0600	0.0000	0.0000	0.0000
60	150	0.0520	0.0000	0.0600	0.0635	0.0500	0.1100	0.0000	0.0000	0.0000
300		0.0550	0.0500	0.1100	0.0910	0.0500	0.1100	0.0000	0.0000	0.0000
600		0.0765	0.0500	0.1100	0.1040	0.0500	0.1100	0.0025	0.0000	0.0500
80	200	0.1565	0.1000	0.1700	0.1705	0.1600	0.2200	0.0000	0.0000	0.0000
400		0.1600	0.1100	0.1700	0.2050	0.1600	0.2200	0.0025	0.0000	0.0500
800		0.1835	0.1600	0.2200	0.2780	0.2700	0.3300	0.0030	0.0000	0.0600
100	250	0.3545	0.3300	0.3900	0.3825	0.3300	0.3900	0.0025	0.0000	0.0500
500		0.3760	0.3300	0.3900	0.4475	0.4300	0.5000	0.0060	0.0000	0.0600
1000		0.4010	0.3800	0.4400	0.6005	0.5400	0.6100	0.0000	0.0000	0.0000

〈표 3〉 노드 쌍간의 통신 수요 범위가 [5,1000]인 경우의 계산 결과

문제크기		VACHANI			DAMICO			MYUNG		
n	K	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
20	50	0.0030	0.0000	0.0600	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
100		0.0025	0.0000	0.0500	0.0060	0.0000	0.0600	0.0000	0.0000	0.0000
150		0.0000	0.0000	0.0000	0.0025	0.0000	0.0500	0.0000	0.0000	0.0000
40	100	0.0170	0.0000	0.0600	0.0165	0.0000	0.0600	0.0000	0.0000	0.0000
200		0.0135	0.0000	0.0600	0.0215	0.0000	0.0600	0.0000	0.0000	0.0000
400		0.0165	0.0000	0.0600	0.0320	0.0000	0.0600	0.0000	0.0000	0.0000
60	150	0.0555	0.0500	0.0600	0.0590	0.0500	0.1100	0.0000	0.0000	0.0000
300		0.0555	0.0500	0.0600	0.0595	0.0500	0.1100	0.0030	0.0000	0.0600
600		0.0675	0.0500	0.1100	0.1070	0.0500	0.1100	0.0000	0.0000	0.0000
80	200	0.1525	0.1100	0.1700	0.1710	0.1600	0.2200	0.0030	0.0000	0.0600
400		0.1635	0.1600	0.1700	0.2060	0.1600	0.2200	0.0000	0.0000	0.0000
800		0.1785	0.1600	0.2200	0.2780	0.2700	0.3300	0.0025	0.0000	0.0500
100	250	0.3650	0.3300	0.3900	0.3845	0.3800	0.3900	0.0060	0.0000	0.0600
500		0.3775	0.3300	0.3900	0.4525	0.4300	0.5000	0.0000	0.0000	0.0000
1000		0.4065	0.3800	0.4400	0.5965	0.5500	0.6100	0.0085	0.0000	0.0600

트 문제에 각 해법의 컴퓨터 code를 실행시키기로 한다. 실험을 위한 데이터는 Dell'Amico 등[4]의 연구에서 제시한 것과 동일한 방법으로 생성하였다. 즉 노드 쌍간의 통신 수요는 네가지 범위인 [5,100], [5, 1000], [50, 100], [500, 1000]에서 임의로 선택하였다. 노드의 개수 n 은 20, 40, 60, 80, 100의 값을 가지고, 노드 쌍의 개수 $|K|$ 는 50에서

1000사이로 하였다.

작성된 코드는 500Mhz CPU를 가진 펜티엄 Celeron PC를 이용하여 실행하였다. 모두 1200문제를 대상으로 계산실험이 이루어졌으며, <표 2>~<표 5>는 링의 구조와 수요가 발생하는 노드 쌍은 동일하나 서로 다른 통신 수요가 주어지는 문제에 대한 계산 결과를 보여주고 있다. 표들에 나

〈표 4〉 노드 쌍간의 통신 수요 범위가 [50,100]인 경우의 계산 결과

문제크기		VACHANI			DAMICO			MYUNG		
n	K	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
20	50	0.0000	0.0000	0.0000	0.0050	0.0000	0.0500	0.0000	0.0000	0.0000
100		0.0050	0.0000	0.0500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
150		0.0000	0.0000	0.0000	0.0140	0.0000	0.0600	0.0000	0.0000	0.0000
40	100	0.0165	0.0000	0.0600	0.0170	0.0000	0.0600	0.0000	0.0000	0.0000
200		0.0165	0.0000	0.0600	0.0220	0.0000	0.0600	0.0000	0.0000	0.0000
400		0.0110	0.0000	0.0600	0.0385	0.0000	0.0600	0.0000	0.0000	0.0000
60	150	0.0545	0.0500	0.0600	0.0550	0.0500	0.0600	0.0000	0.0000	0.0000
300		0.0545	0.0500	0.0600	0.0690	0.0500	0.1100	0.0000	0.0000	0.0000
600		0.0740	0.0500	0.1100	0.1100	0.1100	0.1100	0.0030	0.0000	0.0600
80	200	0.1535	0.1100	0.1700	0.1740	0.1600	0.2200	0.0000	0.0000	0.0000
400		0.1595	0.1100	0.2200	0.2140	0.1600	0.2800	0.0030	0.0000	0.0600
800		0.1755	0.1600	0.2200	0.2750	0.2700	0.2800	0.0090	0.0000	0.0600
100	250	0.3490	0.3300	0.3900	0.3825	0.3300	0.3900	0.0000	0.0000	0.0000
500		0.3740	0.3300	0.3900	0.4535	0.4300	0.5000	0.0050	0.0000	0.0500
1000		0.4030	0.3800	0.4400	0.5985	0.5500	0.6100	0.0025	0.0000	0.0500

〈표 5〉 노드 쌍간의 통신 수요 범위가 [500,1000]인 경우의 계산 결과

문제크기		VACHANI			DAMICO			MYUNG		
n	K	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
20	50	0.0000	0.0000	0.0000	0.0025	0.0000	0.0500	0.0000	0.0000	0.0000
100		0.0000	0.0000	0.0000	0.0030	0.0000	0.0600	0.0000	0.0000	0.0000
150		0.0030	0.0000	0.0600	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
40	100	0.0055	0.0000	0.0600	0.0115	0.0000	0.0600	0.0000	0.0000	0.0000
200		0.0105	0.0000	0.0600	0.0220	0.0000	0.0600	0.0000	0.0000	0.0000
400		0.0115	0.0000	0.0600	0.0365	0.0000	0.0600	0.0000	0.0000	0.0000
60	150	0.0550	0.0500	0.0600	0.0545	0.0500	0.0600	0.0000	0.0000	0.0000
300		0.0565	0.0500	0.0600	0.0820	0.0500	0.1100	0.0030	0.0000	0.0600
600		0.0575	0.0500	0.1100	0.1100	0.1100	0.1100	0.0000	0.0000	0.0000
80	200	0.1550	0.1100	0.1700	0.1635	0.1600	0.1700	0.0030	0.0000	0.0600
400		0.1580	0.1100	0.1700	0.2040	0.1600	0.2200	0.0055	0.0000	0.0600
800		0.1760	0.1600	0.2200	0.2750	0.2700	0.2800	0.0050	0.0000	0.0500
100	250	0.3680	0.3300	0.3900	0.3820	0.3300	0.3900	0.0000	0.0000	0.0000
500		0.3705	0.3300	0.3900	0.4525	0.4300	0.5000	0.0030	0.0000	0.0600
1000		0.4095	0.3800	0.4400	0.5855	0.5500	0.6100	0.0155	0.0000	0.0600

타난 모든 수치는 각각의 $(n, |K|)$ 에 대하여 20개의 문제를 실행한 뒤, 이에 대한 평균실행시간(Avg), 최소실행시간(Min), 최대실행시간(Max)을 구한 결과이다. 표들에서 VACHANI는 Vachani 등의 해법, DAMICO는 Dell'Amico 등의 해법, MYUNG은 Myung 등의 해법을 나타내고 있다.

표들에 나타난 결과를 보면 세 가지 해법 모두에서 주로 $(n, |K|)$ 에 비례하여 계산시간이 증가하며, 통신 수요 범위의 변화는 계산 시계에 크게 영향을 주지 않음을 알 수 있다. 그리고 각각의 경우에 일관성 있게 Myung 등의 해법이, Dell'Amico 등과 Vachani 등의 해법보다 수 십 배에서 수 백 배까지 빠르게 수행되었으며 Vachani 등의 해법이 Dell'Amico 등의 해법보다 약간 적은 시간을 소비하였음을 알 수 있다. 특히 최소실행시간(Min)을 살펴보면, Dell'Amico 등과 Vachani 등의 해법은 문제의 성격에 상관없이 어느 정도 시간을 소비하는데 반해서, Myung 등의 해법은 빠른 시간 내에 완료하는 경우가 많다는 것을 알 수 있다. 이것은 전자의 두 해법이 우선 D_{max} 를 계산하여야 되고 문제의 구조가 특별해도 해를 구하기 위해서는 동일한 절차를 수행해야 되나, Myung 등

의 해법은 재루팅에서 링크 용량의 갱신이 많이 일어나지 않으면 전체 계산시간이 감소하기 때문이다. 결론적으로 Myung 등의 해법은 이론적인 계산 시간에서 $O(n|K|)$ 로 $O(n^3)$ 의 시간이 필요한 Dell'Amico 등과 Vachani 등의 해법보다 우수할 뿐만 아니라 실제 계산 시간에 있어서도 매우 우수한 성능을 보인다는 것을 알 수 있다.

5. 결 론

링의 용량결정문제는 현실적인 중요성 때문에 많은 연구가 이루어져 왔다. 이 중에서 수요의 분할 처리가 허용되는 경우에 링의 용량을 최소화하는 루팅 방법을 찾는 RLPW의 해법은 동기식 광통신망(SONET)에서 유용하게 사용될 수 있는 Self-healing Ring(SHR)중 양방향으로의 운용이 가능한 SHR(Bidirectional SHR : BSHR)의 설계문제에 직접 적용되며, 수요의 분할 처리가 허용되지 않거나 허용되더라도 정수 단위로만 가능한 문제에 대해서도 완화문제의 해법으로서 유용하게 사용될 수 있다. 이러한 응용성 때문에 RLPW에 대해서 최근에 Vachani 등의 해법, Schrijver 등의 해법,

Myung 등의 해법 및, Dell'Amico 등의 해법 등이 동시 다발적으로 개발되었으나, 서로 독립적으로 연구가 이루어진 관계로 이론적인 계산의 효율성 외에는 실질적인 성능의 비교는 이루어지지 못하였다.

본 연구에서는 위에서 언급된 네 가지 해법 중, 기본적으로 구조가 서로 다른 Vachani 등의 해법, Dell'Amico 등의 해법 및, Myung 등의 해법의 특성을 비교 분석하였고 각 해법의 컴퓨터 code를 만들어 다양한 테스트 문제를 풀어 보았다. 계산 실험의 결과 이론적인 계산 시간에서 가장 우수한 Myung 등의 해법이 실제 계산에서도 Dell'Amico 등과 Vachani 등의 해법보다 매우 우수한 성능을 보이는 것으로 나타났다. 본 연구의 대상이 되는 문제는 실제 SONET 링을 설계할 때 반복적으로 풀어야 되는 개발도구로 이용되므로, 본 연구에서의 결과대로 Myung 등의 해법을 이용함으로써 전체적으로 많은 계산상의 효율을 가져올 수 있을 것으로 기대된다.

참 고 문 헌

- [1] 명영수, "정수단위로만 루팅이 허용되는 SONET링의 용량결정문제", 『한국경영과학회지』, 제25권, 제1호(1998), pp.49-62.
- [2] Cosares, S., D.N. Deutsch, I. Saniee and O.J. Wasem, "SONET toolkit : a decision support system for designing robust and cost-effective fiber-optic networks," *Interfaces*, Vol.25, (1995), pp.20-40.
- [3] Cosares, S. and I. Saniee, "An optimization problem related to balancing loads on SONET rings," *Telecommunication Systems*, Vol.3, (1994), pp.165-181.
- [4] Dell'Amico, M., M. Labbe, and F. Maffioli, "Exact Solution of the SONET Ring Loading Problem," *Operations Research Letters*, Vol.26, (1999), pp.119-129.
- [5] Frank, A., T. Nishizeki, N. Saito, H. Suzuki and E. Tardos, "Algorithms for routing around a rectangle," *Discrete Applied Mathematics*, Vol.40, (1992), pp.363-78.
- [6] Myung, Y.-S., H.-G. Kim and D.-W. Tcha, "Optimal load balancing on SONET bidirectional rings," *Operations Research*, Vol.45, (1997), pp.148-152.
- [7] Myung, Y.-S., "An Efficient Algorithm for the Ring Loading Problem with Integer Demand Splitting," *SIAM J. Discrete Math.*, Vol.14, (2001), pp.291-298.
- [8] Okamura, H. and P.D. Seymour, "Multicommodity flows in planar graphs", *Journal of Combinatorial Theory Series B* Vol.31, (1981), pp.75-81.
- [9] Schrijver, A., P. Seymour and P. Winkler, "The ring loading problem," *SIAM J. Discrete Math.*, Vol.11, (1998), pp.1-14.
- [10] Vachani, R., A. Shulman and P. Kubat, "Multicommodity flows in ring networks," *INFORMS Journal on Computing* Vol.8, (1996), pp.235-242.
- [11] Wu, T.H., *Fiber network service survivability*, Artech House, Boston, MA, 1992.