

EDP들의 참조 테이블을 이용한 실용적인 경로 설정 및 파장 할당 알고리즘

(A Practical RWA Algorithm-based on Lookup Table for Edge Disjoint Paths)

김명희[†] 방영철^{**} 정민영^{***} 이태진^{***} 추현승^{****}
(Myung Hee Kim) (Young-Cheol Bang) (Min Young Chung) (Tae-Jin Lee) (Hyunseung Choo)

요약 경로 설정 및 파장 할당(Routing and Wavelength Assignment) 문제는 파장 할당 전송 기법(Wavelength Division Multiplexing)을 기반으로 하는 광 네트워크에서 중요한 분야이다. 경로 설정 및 파장 할당 문제는 선형 프로그래밍(Linear Programming)과 그래프 컬러링(Graph Coloring)의 조합으로 해결되거나, 경로 선택에 기반한 그래프 알고리즘에 의해서 해결된다. 이와 같은 방법은 복잡하며 휴리스틱 알고리즘(heuristic algorithm)이 요구된다. 본 논문에서는 각각의 연결 요구에 대한 최대한의 EDPs(Edge Disjoint Paths)를 구함으로써 경로 설정 및 파장 할당 문제를 효율적으로 해결하는 새로운 방법을 제안한다. 이렇게 구한 EDPs는 참조 테이블에 저장되고 가중치 매트릭스의 갱신에 사용된다. 각 연결 요구에 대한 경로는 순서대로 가중치 매트릭스를 참조하여 결정한다. 본 논문에서 제안하는 알고리즘이 현재 실질적으로 효율적이라 알려진 BGAforEDP(Bounded Greedy Approach for EDP)보다 약 50%의 수행 시간 내에 같거나 작은 파장 수를 사용하여 연결 요구를 만족시켜주고 있음을 시뮬레이션을 통해서 확인할 수 있다.

키워드 : 경로 설정 및 파장 할당 문제, 광 네트워크, EDP, 참조 테이블, 가중치 매트릭스

Abstract Routing and wavelength assignment(RWA) problem is an important issue in optical transport networks based on wavelength division multiplexing(WDM) technique. It is typically solved using a combination of linear programming and graph coloring, or path selection based graph algorithms. Such methods are either complex or make extensive use of heuristics. In this paper we propose a novel and efficient approach which basically obtains the maximum edge disjoint paths (EDPs) for each source-destination demand pair. And those EDPs obtained are stored in Lookup Table and used for the update of weight matrix. Routes are determined in order by the weight matrix for the demand set. The comprehensive computer simulation shows that the proposed algorithm uses similar or fewer wavelengths with significantly less execution time than bounded greedy approach (BGA) for EDP which is currently known to be effective in practice.

Key words : RWA, optical network, EDP, Lookup Table, weight matrix

1. 서론

· 이 논문은 두뇌한국21(BK21)사업과 ITRC 프로젝트의 지원을 받았음

† 비회원 : 코어세스 연구원

orangemh@shinbiro.com

** 비회원 : 한국산업기술대학교 교수

ybang@kpu.ac.kr

*** 비회원 : 성균관대학교 정보통신공학부 교수

mychung@ece.skku.ac.kr

tjlee@ece.skku.ac.kr

**** 종신회원 : 성균관대학교 정보통신공학부 교수

(corresponding author)

choo@ece.skku.ac.kr

논문접수 : 2003년 4월 21일

심사완료 : 2003년 12월 15일

광 네트워크가 대량의 연결 요구를 만족시키기 위한 가장 확실한 방법임은 명백하다. 경로 설정 및 파장 할당 문제(Routing and Wavelength Assignment) 알고리즘은 파장 할당 전송 기법(Wavelength Division Multiplexing) 기술과 광 교환기 시스템으로 구성된 광 네트워크를 운영하기 위해서는 필수 불가결한 요소이며, 현재 아주 활발하게 연구되고 있는 분야이다[1]. 경로 설정 및 파장 할당 문제가 광 네트워크의 가격과 성능에 많은 영향을 끼치기 때문에 여러 가지 최적화 방법이 개발되고 있다. 이에 사용되는 알고리즘의 목적은 필요한 파장의 수를 줄이거나[2,3], 고정된 파장 수에 동적

연결 요구에 대한 블로킹 확률(Blocking Probability)을 줄이는 것[4,5], 그리고 네트워크 비용을 줄이는 것[6,7] 등을 포함한다.

경로 설정 및 파장 할당 문제는 여러 가지 조건 하에서 연구된다. 그 중 하나의 조건은 경로를 어떻게 설정하고 해제하는지는 문제이다. 본 논문에서는 광 경로가 한번 설정하면 해제되는 데 긴 시간이 걸리는 정적 경로 할당을 가정한다. 정적 경로 할당은 망 경로가 랜덤하게 설정되고 해제되는 동적 경로 할당보다 현재의 기술에서 실질적으로 구현 가능하기 때문이다. 또 다른 조건은 파장 변환기(Wavelength Converter)의 사용 유무이다. 여기에서는 토폴로지 상에 파장 변환기가 없는 경우에 대하여 연구한다. 그러나 제한하는 알고리즘은 파장 변환기가 존재하는 경우에도 쉽게 적용될 수 있을 것으로 예상된다. 본 연구의 목적은 사용하는 파장의 수는 줄이거나 유지하면서 이에 걸리는 수행 시간을 줄이는 것이다.

본 논문은 다음과 같이 구성되어 있다. 먼저 경로 설정 및 파장 할당 문제에 대한 알고리즘의 일반적인 두 방법인 *lpsolver*에 기초한 라운딩 휴리스틱 방법(Rounding Heuristic method)[3]과 BGAforEDP(Bounded Greedy Algorithm for Edge Disjoint Path)[2]을 설명한다. 3장에서 EDPs을 기초로 하는 DB_MAX_EDP(Demand Based Maximum EDP)와 LTB_RWA(Lookup Table Based RWA) 알고리즘을 제안한다. 4장에서는 제안한 알고리즘인 LTB_RWA와 BGAforEDP 알고리즘에 대한 성능평가를 서술한다. 이러한 성능평가 결과는 제안한 알고리즘이 기존의 방법에 비하여 수행 시간이나 사용하는 파장의 수에 있어서 더 나은 성능을 지니고 있음을 보여준다. 마지막으로 전체적인 결론을 5장에서 서술한다.

2. 관련 연구

경로 설정 및 파장 할당 문제를 해결하는 데는 기본적으로 두 가지 접근법이 있다. 그 하나는 경로 설정 및 파장 할당 문제에 대한 최적의 해결책을 얻기 위해서 *lpsolver*를 사용하는 방법인데, *lpsolver*는 선형 프로그램(linear program)을 다중항목 흐름(multicommodity flow) 개념을 이용하여 해결하는 프로그램이다. 나머지 하나는 사용하는 파장의 수를 줄이기 위해서 경로 설정을 기초로 하는 그래프 알고리즘을 사용하는 방법이다. 본 장에서는 각각의 방법에 대한 대표적인 알고리즘에 대해 논한다.

2.1 라운딩 휴리스틱 알고리즘과 그래프 컬러링을 이용하는 방법

이 방법에서는 경로 설정 및 파장 할당 문제를 2가지 단계로 해결하고 있는데 이는 경로 할당 단계와 파장

할당 단계로 나뉘어진다. 라운딩 휴리스틱 알고리즘은 경로 할당에 사용되고 그래프 컬러링은 파장 할당 단계에서 사용한다[3].

경로 할당 단계인 라운딩 휴리스틱 알고리즘은 다음의 세 단계로 이루어져 있다. 일 단계인 정수 제약이 없는 다중항목 흐름 문제를 해결, 이 단계 경로 탐색 그리고 마지막으로 램덤하게 경로 설정하기가 그것이다. 일 단계에서는 일반적인 *lpsolver*를 이용하여 각 요구에 대한 네트워크에서의 링크 흐름을 얻는다. 여기에서 사용하는 *lpsolver*에서는 소수를 허용하기 위해서 0-1 흐름에 대한 제약을 없애고 사용한다. 이 단계에서는 연결 요구 i 에 대한 링크 흐름에서 그 i 의 흐름이 가능한 경로들의 집합인 Q 를 얻는다. 연결 요구의 출발점에서 나가는 흐름이 없을 때까지 가능한 경로와 그 경로의 가중치를 찾아 Q 에 저장하게 되는 것이다. 마지막 단계에서는 $|Q|$ 의 면을 가지고 각 면이 나올 확률이 각 경로에 대한 가중치인 주사위를 던지게 된다. 여기에서 나온 면에 해당하는 경로가 연결 요구 i 에 할당된다. 모든 연결 요구에 이 알고리즘을 적용함으로써 라운딩 휴리스틱 알고리즘은 각각의 연결 요구 i 에 대한 경로를 얻게 한다.

파장 할당 단계에서는 그래프 컬러링을 사용하는데, 이를 위하여 먼저 그래프 $G_R(V_R, E_R)$ 를 생성하게 된다. 이 그래프는 경로 할당 단계에서 얻은 각 경로가 노드가 되고 각 경로가 같은 링크를 지날 경우에 그 경로에 의해 만들어진 노드 사이에 방향성이 없는 에지가 존재하게 되는 그래프이다. 그래프 GR 을 얻은 후에는, 그래프에서 이웃하는 노드들이 같은 색을 가지지 않게 각 노드에 색을 칠하는 그래프 컬러링 방법을 사용한다. 최종적으로 사용된 색의 수가 이 알고리즘을 사용하였을 때 사용하는 파장의 수가 된다.

2.2 BGAforEDP에 기초한 알고리즘

이 알고리즘은 최단거리 알고리즘에 기초한 간단한 EDP 기법을 사용한다[2]. V_B 와 E_B 가 각각 노드의 집합과 에지의 집합을 가리키는 네트워크의 그래프를 $G_B(V_B, E_B)$ 라 한다. 그리고 τ 는 연결 요구 집합을 의미한다. s_i 를 i 번째 연결 요구의 시작점으로 하고 t_i 를 i 번째 연결 요구의 도착점이라고 할 때 $\tau = \{(s_1, t_1), \dots, (s_k, t_k)\}$ 로 표현된다.

BGAforEDP는 다음 3개의 파라미터를 가지고 동작하는데, 초기 그래프, 연결될 요구 집합, 적절한 d 가 그것이다. 여기에서 d 는 $\max(\text{diam}(G), \sqrt{|E|})$ 이다. d 는 할당되는 경로의 홉(hop) 수를 제한하기 위하여 사용된다. 먼저 BGAforEDP 알고리즘은 연결 요구 집합에서 랜덤하게 하나의 연결 요구 하나를 선택하고, 그에 대한 최단 경로 P_i 를 찾는다. 만약 P_i 의 길이가 제한 길이인

d 보다 짧다면 (i, P_i) 를 설정 경로 집합 P 에 포함시키고, 경로가 설정된 연결 요구를 현재 경로가 설정된 연결 요구 집합인 $a(G_B, \tau)$ 에 포함시킨다. 그리고 P_i 에서 사용하는 에지들을 다른 연결 요구에 할당하지 않기 위해서 G_B 에서 삭제한다. 만약 연결된 경로 P_i 의 길이가 d 보다 크다면, 연결 요구 i 는 경로를 할당 받지 못한다. 이와 같은 과정으로 τ 에 있는 연결 요구에 경로 설정이 가능한지를 모두 체크한다. 알고리즘이 끝난 후에 $a(G_B, \tau)$ 는 같은 파장이 할당될 수 있는 연결 요구들의 집합을 가지고 있게 된다. $a(G_B, \tau)$ 를 τ 에서 삭제함으로써 경로 설정이 아직 되지 않은 연결 요구 집합인 τ' 을 구한다. 만약 이 τ' 집합이 공집합이 아니라면 아직 경로 할당을 받지 못한 연결 요구가 있다는 의미이므로, 다른 파장에 경로 할당 가능한 연결 요구들의 집합을 찾기 위해서 BGAforEDP 알고리즘을 초기의 그래프 G_B 와 τ' 을 파라미터로 하여 다시 실행한다. 경로 설정이 되지 않은 집합인 τ' 이 공집합이 될 때까지 이를 반복 시행한다. 전체적으로 할당된 파장의 수가 이 알고리즘의 결과 값이 된다.

논문 [2]에서 저자는 라운딩 휴리스틱 알고리즘과 BGAforEDP를 기초로 한 알고리즘의 사용하는 파장의 수와 실행 시간에 대하여 성능 평가를 수행하였다. 그 결과는 두 방법이 사용하는 파장의 수는 거의 같지만 라운딩 휴리스틱 알고리즘이 BGAforEDP를 기초로 한 알고리즘보다 항상 더 오랜 시간이 걸린다는 것을 보여 주고 있다. 따라서 본 논문에서는 제안하는 방식과 BGAforEDP에 기초로 한 알고리즘에 대한 성능 비교를 하기로 한다.

3. 제안 알고리즘

3.1 기본 개념

본 절에서는 BGAforEDP를 기반으로 하는 알고리즘의 예를 먼저 살펴보기로 하겠다. 그림 1(a)에서 예제 토폴로지를 보여주고 있고, 연결 요구 집합은 $\{(1,4), (1,4) (2,3)\}$ 이다. 여기에서 집합은 중복되는 연결 요구

를 포함하고 있으며 나열되는 순서 또한 지켜진다. 그림 1(a)에서 d 는 4이다. 먼저 파장 1에서 첫 번째 연결 요구인 (1,4)에 대한 최단 경로는 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ 이다. 이렇게 경로 할당을 할 경우 나머지 두 연결 요구 (1,4)와 (2,3)는 첫 번째 요구 (1,4)와 같은 파장 1에서 경로 할당을 할 수가 없게 된다. 파장 2에서 두 번째 연결 요구 (1,4)에 대한 최단 경로 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ 를 할당하고, 마지막 요구 (2,3)에는 파장 3에서 최단 경로 $2 \rightarrow 3$ 을 할당해야 한다. 보는 바와 같이 본 예제에서 BGAforEDP 기반의 알고리즘을 사용할 경우 3개의 파장을 사용하게 된다. 그러나 그림 1(c)에서 보는 바와 같이 연결 요구 (1,4)는 두 개의 EDP를 가지고 있다. $1 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 4$ 그리고 $1 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 4$ 가 그것이다. 이 EDPs를 사용하면 하나의 파장으로 모든 연결 요구를 만족시킬 수 있다. 즉 첫 번째 연결 요구 (1,4)에는 $1 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 4$ 를, 두 번째에는 $1 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 4$ 를, 그리고 마지막 요구에는 $2 \rightarrow 3$ 을 할당하면 된다. 이와 같이 연결 요구에 대한 EDPs를 이용한다면 좀 더 효율적으로 경로 설정을 할 수 있을 것이므로, 이러한 개념을 LTB_RWA 알고리즘에 사용한다.

BGAforEDP를 기반으로 하는 알고리즘은 BGAforEDP 알고리즘을 실행할 때마다 각각의 연결 요구에 대하여 최단 경로를 구하려고 시도를 하게 되고, 파장의 수만큼 BGAforEDP 알고리즘이 실행하게 된다. 그렇기 때문에 전체적인 알고리즘으로 볼 때는 연결 요구의 개수의 몇 배에 해당하는 만큼 최단 경로를 구하는 루틴을 실행하게 된다. 이는 같은 연결 요구가 있는 경우에도 최단 경로를 구하는 루틴을 매번 실행해야 하는 시간적인 낭비가 생긴다. 만약 한 번 요구된 연결 요구에 대하여 구해진 최단 경로를 저장해 둔다면 이를 다시 계산하지 않고 바로 경로 할당을 할 수가 있으므로 시간을 절약할 수 있을 것이다. LTB_RWA 알고리즘에서는 이러한 개념을 이용하여 알고리즘 수행 시간을 줄이는 데 사용하고자 한다.

3.2 DB_MAX_EDP(Demand-based Maximum EDP)

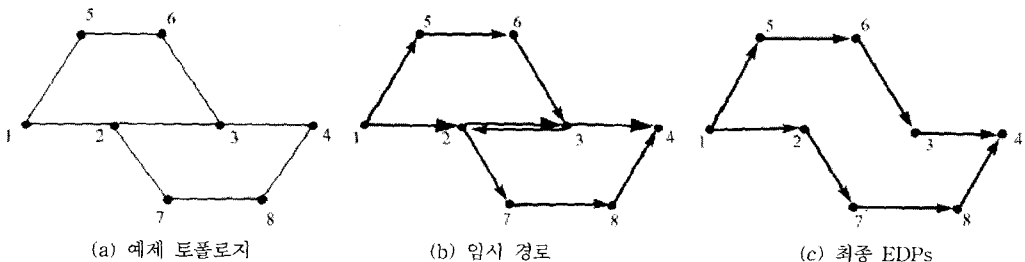


그림 1 연결 요구 (1,4)에 대한 EDPs를 찾는 과정

네트워크의 그래프를 $G=(V,E)$ 라 하자. 여기에서 V 는 노드들의 집합이고 E 는 에지들의 집합을 말한다. 연결 요구는 랜덤하게 발생한다. 먼저 참조 테이블을 만들게 되는데 이는 연결 요구 집합에 있는 각 연결 요구에 대한 최대한의 EDPs를 가지고 있는 테이블이다. 본 논문에서도 적절한 d 값을 사용하는데 이는 BGAforEDP를 기반으로 하는 알고리즘과 마찬가지로 $\max(\text{diam}(G), \sqrt{|E|})$ 을 의미한다(그림 2에서 라인 1). d 에 대한 자세한 내용은 [9]에 서술되어 있다. 참조 테이블에는 각 경로에 대한 홉수가 d 보다 작은 EDP가 저장되어 있다(그림 2에서 라인 7-8).

본 논문에서는 각 연결 요구에 대한 최대한의 EDPs를 구하기 위해서 DPSP(Disjoint Pathset Selection Protocol)[10] 개념을 사용한다. DPSP의 목적이 본 논문과 다르기는 하지만 EDPs를 선형 시간에 구하는데 이를 사용할 수는 있다. [10]에서 정방향 에지(forward edge)는 할당된 경로에 포함되어 있으면서 출발점에서 종착점 방향으로 향하고 있는 에지를 말한다. 역방향 에지(backward edge)는 할당된 경로에 포함되어 있으면서 정방향 에지의 방향과 반대되는 방향을 가지는 에지를 의미한다. 단, 출발점이나 종착점에 연결되어 있는 에지는 역방향 에지에 속하지 않는다. 본 논문에서는 최단 경로를 찾을 때 할당되지 않은 정방향 에지나 또는 역방향 에지들에 대해서만 경로를 찾게 된다. 최대한의 EDPs를 얻을 때까지 최단 경로 알고리즘을 실행하는데, 하나의 경로를 구한 후에 그 경로에 역방향 에지가 존재하면 그 에지를 지운다(그림 2에서 라인 3-6). 이러한 과정에서 역방향 에지를 동시에 지나는 경로들에서 독립된 EDPs를 얻을 수 있다. 이것이 최대한의 EDPs를 얻는 기본적인 절차로서 이는 출발점에 연결되어 있는 에지의 수와 종착점에 연결되어 있는 에지의 수 중에 작은 쪽의 개수만큼 EDPs를 얻을 때에야 종료하게 된다. 예를 들어 그림 1(a)에서 연결 요구 (1,4)는 최단 경로 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ 를 가지고, $1 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 4$ 가 다시 발견되게 된다(그림 1(b)). (3,2)가 역방향 에지이기 때문에 그 에지를 제거한다. 그러면 $1 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 4$ 와 $1 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 4$, 2개의 EDPs를 얻을 수 있다(그림 1(c) 참조).

참조 테이블은 각 연결 요구에 대한 최대 개수의 EDPs를 가지고 있다. 이는 토폴로지가 바뀌기 전에는 다른 연결 요구 집합들에도 사용된다. 참조 테이블을 얻는 동안 각 에지에 대한 가중치 또한 계산하게 된다. $G(V, E)$ 에서의 에지 가중치는 각 요구에 대한 EDP가 에지를 지나갈 때 그 에지의 가중치를 1씩 증가하여 얻게 된다(그림 2에서 라인 9). 에지 가중치 매트릭스는 각 연결 요구 집합에 대하여 따로 계산된다. 이 합

수에 대한 수도(pseudo) 코드를 다음 그림 2에 보이고 있다.

Algorithm DB_MAX_EDP(G,D)

INPUT: G is a network topology and D is a demand set.

OUTPUT: Lookup Table and weight matrix of edges.

Begin

1. $d = \max(\text{diam}(G), \sqrt{|E|})$
2. For $i=1$ to $|D|$
3. Repeat
4. Find shortest path P which passes only unassigned forward edges along with backward edges.
5. Remove backward edge(s) used in the path
6. Until (si, ti) has max. EDPs
7. For each P_j ($j=1..#$ of EDPs)
8. If (the number of hops of $P_j > d$) ignore P_j
9. Increment 1 the weight of used edges
10. Store all EDPs of (si, ti) to Lookup Table
11. Endfor

End

그림 2 DB_MAX_EDP Algorithm

3.3 LTB_RWA(Lookup Table-based RWA)

DB_MAX_EDP 함수를 사용하여 참조 테이블과 에지 가중치 매트릭스를 얻은 후(그림 3에서 라인 3), 각 연결 요구는 차례대로 경로 할당을 하게 된다. 먼저 임시 그래프 G^* 생성하고(그림 3에서 라인 6), 그 임시 그래프에 할당될 파장의 번호를 설정한다(그림 3에서 라인 7). 먼저 연결 요구에 대하여 참조 테이블에 저장해 둔 EDPs 중에 가장 짧은 경로를 할당한다. 이때 만약 같은 길이의 경로가 있다면 각 EDP의 에지 가중치 중에 가장 작은 에지 가중치를 체크한다. 이를 W_i 라 하자. W_i 중에 가장 작은 값을 W 라 하고 W 와 같은 가중치를 가지는 경로를 그 연결 요구에 할당하게 된다(그림 3에서 라인 9-10). 경로를 할당한 후에는 연결 요구를 D 에서 제거하고(그림 3에서 라인 11), 할당한 경로의 에지를 G^* 에서 지운다(그림 3에서 라인 12). 모든 연결 요구에 대하여 경로 할당 시도를 해 본 후에 연결되지 않은 요구가 있다면, 그 연결 되지 않은 경로들에 대하여 초기의 G 에서 증가된 파장을 할당하면서 경로 할당을 하게 된다(그림 3에 있는 알고리즘 LTB_RWA (G)에서 라인 5의 while문을 통해서 구현된다).

이 알고리즘에 대한 수도(pseudo) 코드가 아래 그림 3에 나와 있다. DB_MAX_EDP(G, D) 함수는 참조 테이블과 에지 가중치 매트릭스를 리턴해 준다. 위에서 언급한 바와 같이 참조 테이블은 이전에 할당된 적이 있는 연결 요구에 대하여는 이전에 구해진 EDP를 다시 사용한다. 반면 새로 요구된 연결 요구 집합에 대하여 에지 가중치 매트릭스를 매번 구한다. 라인 9,10에서의 $P(s_i, t_j, i)$ 는 연결 요구 (s_j, t_j)에 대하여 참조 테이블에 저장되어 있는 i 번째 EDP를 의미한다.

Algorithm LTB_RWA (G, D)

INPUT: G is a network topology and D is a demand set.
 OUTPUT: Total number of used wavelength λ for D.

```

Begin
  1. While (the same G) do
  2.   Retrieve a new D
  3.   DB_MAX_EDP (G, D)
  4.    $\lambda = 0$ 
  5.   While (D ≠ ∅) do
  6.      $G^* = G$ 
  7.      $\lambda = \lambda + 1$ 
  8.     For j = 1 to |D| do
  9.       If P( $s_j, t_j, i$ ) with the smallest W is available
          in  $G^*$  then
 10.        Assign P( $s_j, t_j, i$ ) and  $\lambda$  to ( $s_j, t_j$ )
 11.         $D = D - (s_j, t_j)$ 
 12.        Delete the edge in P( $s_j, t_j, i$ ) from  $G^*$ 
 13.      Endif
 14.    Endfor
 15.  Endwhile
 16.  Return  $\lambda$ 
 17. Endwhile
End
    
```

그림 3 LTB_RWA Algorithm

3.4 예제

그림 4(a)는 네트워크 토폴로지의 그래프를 보여준다. 연결 요구 집합은 {(1,8), (4,6), (1,8), (6,9)} 이다. diam(G)는 4이고 $\sqrt{|E|}$ 는 4.123이므로 d 는 4.1이다. 먼저 참조 테이블과 가중치 매트릭스를 DB_MAX_EDP (G,D) 함수를 통하여 구한다. 이 함수에서는 각 연결 요구에 대하여 최대한의 EDP를 얻는다. 첫 번째와 세 번째 연결 요구인 (1,8)에 대하여 1→2→7→6→8과 1→4→3→5→8이 얻어진다. 두 번째 요구 (4,6)은 4→6, 4→2→7→6, 그리고 4→3→5→6을, 마지막 연결 요구인 (6,9)는 6→8→9와 6→5→9를 EDP로 가지게 된다. 위에서 구한 7개의 경로들은 모두 d 보다 짧은 길이를 가진다. 가중치 매트릭스를 위해 각 연결 요구에 대해 얻은 EDPs에 있는 에지들의 가중치를 1 증가한다. 이때 연결 요구 (1,8)는 2번의 연결 요구가 있기 때문에 (1,8)이 가지는

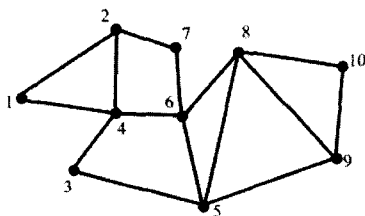
EDPs들에 대한 가중치를 각각 1씩 증가하게 되므로 에지 1→2의 경우 가중치가 2가 되게 된다. 연결 요구의 개수가 4이고 각 요구에 대한 경로들의 에지가 서로 겹치지 않기 때문에 가중치의 최대값은 4이다.

본 예제의 가중치 매트릭스를 그림 4(b)에서 보여주고 있다.

첫 번째 연결 요구 (1,8)은 같은 길이를 가진 2개의 EDPs를 가지고 있기 때문에 가중치 매트릭스를 보고 경로를 선택해야 한다. 이 두개의 EDPs의 가중치의 가장 작은 값은 2이다. 따라서 우리는 랜덤하게 경로 1→2→7→6→8을 선택한다. 두 번째 요구 (4,6)은 참조 테이블에 저장되어 있는 EDPs 중에서 가장 길이가 짧은 경로 4→6을 선택하여 할당한다. 세 번째 (1,8)은 첫 번째 연결 요구 (1,8)이 하나의 경로를 사용하고 있기 때문에 다른 하나의 경로 1→4→3→5→8를 할당한다. 마지막 연결 요구 (6,9)는 에지 (6,8)이 첫 번째 연결 요구에 할당되어 있으므로 다른 경로 6→5→9를 선택한다. 위에서 보는 바와 같이 필요한 파장의 수는 1이다. BGAfor EDP에 기반한 알고리즘을 사용하는 경우에는 2개의 파장이 필요하다. 즉 파장 1에서는 1→4→6→8과 4→3→5→6, 파장 2에서는 1→4→6→8 그리고 6→5→9가 할당된다.

4. 성능 평가

본 장에서는 제안하는 LTB_RWA 알고리즘과 BGAforEDP에 기반한 Greedy_EDP_RWA 알고리즘[2]의 성능평가를 하여 비교한다. 이때 비교 척도는 사용되는 평균 파장의 수와 수행 시간으로 하였다. LTB_RWA 알고리즘은 같은 토폴로지에서는 참조 테이블을 재사용하기 때문에 수행 시간에 대한 성능평가에서는 NSF-NET을 사용한다. 그러나 알고리즘을 여러 토폴로지에서 평가하기 위하여 파장의 수를 측정하는 경우에는 랜덤하게 생성한 그래프를 사용한다. 사용하는 파장의 수를 구하기 위해 실행할 시뮬레이션을 위해서 랜덤 그래



(a) 예제 토폴로지

	1	2	3	4	5	6	7	8	9	10
1		2		2						
2	2			1			3			
3				3	3					
4	2	1	3			1				
5			3			2		2	1	
6				1	2		3	3		
7		3					3			
8					2	3			1	
9					1			1		
10										

(b) 가중치 매트릭스

그림 4 예제 토폴로지와 가중치 매트릭스

프와 연결 요구 집합을 생성한다. 먼저 그래프에 존재하는 노드의 수(N)와 각 노드 쌍들 간에 에지의 존재 확률(P_e)에 의해서 랜덤 그래프를 생성하여야 한다. 랜덤 그래프는 토폴로지이므로 항상 연결이 되어 있어야 하기 때문에 이를 위해 N 노드를 가지는 스패닝 트리(spanning tree)를 먼저 생성한다. 이 스패닝 트리에서 $N-1$ 개의 에지를 연결하였으므로 P_e 를 다시 계산하여 얻는다. 그런 다음 새로 계산한 P_e 를 이용하여 나머지 에지들의 유무를 결정하여 랜덤 그래프를 최종적으로 생성한다. 연결 요구 집합은 $N \times N$ 매트릭스로 생성하게 되는데, 같은 연결 요구는 MAX 값 이하로 생성할 수 있고 하나의 연결 요구는 P_1 의 확률로 발생하게 하였다. 본 논문에서는 각각의 N , P_e , 그리고 P_1 에 대하여 1000번의 시뮬레이션을 시행하여 평균적으로 사용되는 파장의 수를 구하였다. 표 1은 그래프 G 와 랜덤하게 발생시킨 연결 요구에 대하여 사용하는 평균 파장의 수를 보여주고 있다. 제한하는 방법이 MAX = 1인 경우를 제외한 다른 환경에서 BGAforEDP에 기반한 Greedy_EDP_RWA 방식이 사용하는 파장의 수 이하의 파장 수를 사용함을 볼 수 있다. MAX가 1인 경우에는 한 연결 요구에 대한 중복을 허용하지 않기 때문에 현실적이지 않다.

따라서 본 논문에서 제안하는 방식이 현실적인 조건 하에서 기존의 방법과 유사한 성능을 낸다고 결론지을 수 있다.

수행 시간에 대하여는 NSFNET에서 위와 같은 방식

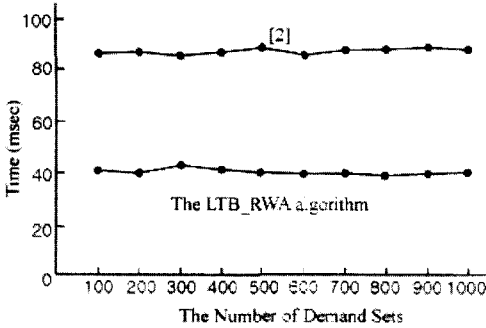
으로 얻은 연결 요구 집합에 대하여 참조 테이블과 가중치 매트릭스를 얻는데 걸리는 시간(DB_MAX_EDP 함수의 수행시간)과 라우팅 시간을 측정하였다. 100개부터 1000개까지의 연결 요구 집합을 따로 발생시켜서 평균 수행 시간을 구한다. 그림 5는 MAX가 5일 경우의 수행 시간을 보여주고 있다(그림 5(a)는 $P_1=0.6$ 일 경우, 그림 5(b)는 $P_1=0.8$ 일 경우이다.) 그래프는 LTB_RWA 알고리즘이 BGAforEDP에 기반한 Greedy_EDP_RWA 보다 약 2배 가량 빠른 것을 보여주고 있다.

5. 결론

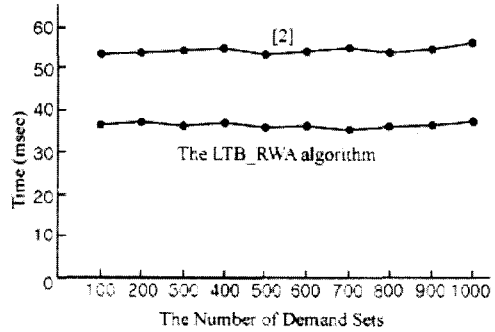
본 논문에서는 광 네트워크에서의 경로 설정 및 파장 할당 문제 해결을 위하여 최대의 EDPs에 기반한 알고리즘을 제안하였다. 연구 목적은 다른 알고리즘보다 더 빠른 시간 안에 필요한 파장의 수를 비슷하거나 적게 사용하는 것이고, 정적 경로 할당과 파장 변환기가 없는 경우에서 문제 해결을 시도하였다. 제안한 알고리즘은 DB_MAX_EDP 함수를 통하여 각 연결 요구에 대한 최대한의 EDPs를 가지고 있는 참조 테이블과 가중치 매트릭스를 구한다. 참조 테이블과 가중치 매트릭스를 저장해야 하기 때문에 메모리의 사용이 필요한 단점이 있기는 하지만 이는 그리 큰 공간을 필요로 하지 않기 때문에 무시할 수 있다. LTB_RWA는 여기에서 구한 참조 테이블과 가중치 매트릭스에 기반하여 경로를 할당한다. 기존에 구한 참조 테이블은 네트워크 토폴로지가 바뀌지 않을 경우 재사용할 수 있다. 시뮬레이션 결

표 1 Greedy_EDP_RWA와 LTB_RWA의 성능 비교

N	P_e	P_1	MAX = 1		MAX = 3		MAX = 5	
			[2]	LTB_RWA	[2]	LTB_RWA	[2]	LTB_RWA
10	0.6	0.4	2.02	2.09	5.41	5.38	9.04	8.83
10	0.6	0.6	2.69	2.78	7.86	7.66	13.27	12.78
10	0.6	0.8	3.50	3.54	10.68	10.34	17.69	16.97
10	0.8	0.4	1.22	1.33	3.62	3.56	5.86	5.71
10	0.8	0.6	1.90	1.94	5.23	5.11	8.68	8.31
10	0.8	0.8	2.15	2.17	6.91	6.68	11.52	11.01
14	0.6	0.4	2.03	2.06	5.36	5.23	8.99	8.60
14	0.6	0.6	2.72	2.79	7.86	7.53	13.18	12.43
14	0.6	0.8	3.40	3.42	10.59	10.05	17.60	16.52
14	0.8	0.4	1.18	1.31	3.68	3.64	6.00	5.79
14	0.8	0.6	1.98	1.99	5.38	5.21	8.96	8.49
14	0.8	0.8	2.19	2.23	7.16	6.84	11.89	11.21
18	0.6	0.4	2.02	2.03	5.34	5.21	8.82	8.38
18	0.6	0.6	2.77	2.84	7.93	7.53	13.09	12.26
18	0.6	0.8	3.38	3.0	10.46	9.85	17.57	16.34
18	0.8	0.4	1.13	1.21	3.81	3.74	6.16	5.93
18	0.8	0.6	2.0	2.0	5.44	5.27	9.13	8.64
18	0.8	0.8	2.23	2.24	7.24	6.98	12.17	11.42



(a) P₁ = 0.6



(b) P₁ = 0.8

그림 5 MAX = 5일 경우 평균 수행 시간

과에서 제안하는 알고리즘이 BGAforEDP에 기반한 알고리즘보다 50%의 수행 시간 내에 비슷한 파장을 사용하고 있음을 보여주고 있다. 앞으로는 컨버터가 고려된 광 네트워크에서의 RWA 문제나, 여러 광 경로를 묶어 주어 파장의 효율을 높이고자 하는 트래픽 그루밍에서의 RWA 문제에 대한 연구가 더 진행되어야 하겠다.

참고 문헌

- [1] H. Zang, J.P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, no.1, pp.47-60, Jan. 2000.
- [2] P. Manohar, D. Manjunath, and R.K. Shevgaonkar, "Routing and Wavelength Assignment in Optical Networks from Edge Disjoint Path Algorithms," *IEEE Communication Letters*, vol. 5, pp. 211-213, May 2002.
- [3] D. Banerjee and B. Mukherjee, "A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 903-908, June 1996.
- [4] Y. Zhang, K. Taira, H. Takagi, and S. K. Das, "An Efficient Heuristic for Routing and Wavelength Assignment in Optical WDM Networks," *IEEE International Conference on Communications*, vol. 5, pp. 2734-2739, 2002.
- [5] M.D. Swaminathan and K.N. Sivarajan, "Practical Routing and Wavelength Assignment Algorithms for All Optical Networks with Limited Wavelength Conversion," *IEEE International Conference on Communications*, vol. 5, pp. 2750-2755, 2002.
- [6] M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," *IEEE/ACM Trans. On Networking*, vol. 7, no. 5, pp. 767-778, Oct. 1999.
- [7] I. Chlamtac, A. Farago, and T. Zhang, "Light-path(wavelength) routing in large WDM networks," *IEEE J. Sel. Areas in Commun.* vol. 14, no. 5, pp. 909-913, June 1996.
- [8] <http://only1.snu.ac.kr/software/lpsolver>
- [9] J.M. Kleinberg. "Approximation algorithms for disjoint paths problems," Ph.D. dissertation, MIT, Cambridge, May 1996.
- [10] P. Papadimitratos, Z. J. Haas, E. G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," *MobiHoc 2002*.



김 명 희

2001년 성균관대학교 전기전자컴퓨터공학부(학사). 2003년 성균관대학교 전기전자컴퓨터공학부(석사). 2003년~현재 (주)코어 세스 주임 연구원. 관심분야는 Embedded system, Routing Protocol, Algorithm



방 영 철

1994년 University of Oklahoma, computer science(학사). 1997년 University of Oklahoma, computer science(석사) 2000년 University of Oklahoma, computer science(박사). 2000년~2002년 한국전자통신연구원(ETRI) 선임연구원. 2002년~현재 한국산업기술대학교 전임강사. 관심분야는 Routing protocols, optical networks, mobile computing, and parallel and distributed computing



정 민 영

1990년 한국과학기술원 전기 및 전자공학(학사). 1994년 한국과학기술원 전기 및 전자공학(석사). 1999년 한국과학기술원 전기 및 전자공학(박사). 1999년 1월~2002년 2월 한국전자통신연구원 선임연구원. 2002년 3월~현재 성균관대학교 정보통신공학부 조교수. 관심분야는 인터넷, 이동통신, Ad hoc 네트워킹 기술 등



이 태 진

1989년 연세대학교 전자공학과(학사)
1991년 연세대학교 전자공학과(석사)
1995년 University of Michigan, Ann Arbor, EECS(석사). 1999년 University of Texas at Austin, ECE(박사). 1999년 8월~2001년 2월 삼성전자 중앙연구소 책임연구원. 2001년 3월~현재 성균관대학교 정보통신공학부 조교수. 관심분야는 무선 개인/근거리 네트워크 및 통신시스템, 무선 인터넷, Ad-hoc 네트워크 등



주 현 승

1988년 성균관대학교 수학과(학사). 1990년 University of Texas at Dallas, computer science(석사). 1996년 University of Texas at Arlington, computer science(박사). 1997년~1998년 특허청 심사관사무관. 1998년~현재 성균관대학교 정보통신공학부 부교수. 관심분야는 Optical networking, mobile computing, grid computing