

WDM방식을 기반으로 한 광 네트워크상에서 최대 EDPs(Edge Disjoint Paths)을 이용한 라우팅 및 파장할당 알고리즘

추 현 승* · 정 성 태** · 이 성 창***

요 약

본 논문에서 파장 분할 다중화(WDM) 방식을 이용한 광 네트워크상의 라우팅과 파장할당 알고리즘을 고찰해 보겠다. 선형 프로그래밍(Linear Programming)과 그래프 컬러링(Graph Coloring)의 조합으로 이루어진 기존의 RWA기법들은 복잡하며, 발견적 방법(Heuristic Method) 사용이 요구된다. 이와 같은 방법은 실행시간이 길며, 최악의 경우에는 실행이 불가능하여 결과를 얻지 못한다. RWA를 해결하기 위한 다른 방법은 최대 EDPs(Edge Disjoint Paths)를 얻기 위해 greedy algorithm을 적용하는 것이다. 이것은 실행시간이 짧지만 파장의 수를 최적으로 사용하지 못한다. 본 논문에서 최대의 EDPs를 얻기 위해서 최대 흐름 기법(Maximum Flow Technique)을 이용한 새로운 알고리즘을 제안한다. 그리고 제안한 알고리즘과 기존에 제시된 최대 EDPs 알고리즘을 비교해 보겠다.

Routing and Wavelength Assignment in Optical WDM Networks with Maximum Quantity of Edge Disjoint Paths

Hyunseung Choo* · Sung-Taek Chung** · Sungchang Lee***

ABSTRACT

In the present paper routing and wavelength assignment (RWA) in optical WDM networks is considered. Previous techniques based on the combination of integer linear programming and graph coloring are complex and require extensive use of heuristics. Such methods are mostly slow and sometimes impossible to get results due to infeasibility. An alternative approach applied to RWA employs on the greedy algorithm for obtaining the maximum edge disjoint paths. Even though this approach is fast, it produces a solution for any connection request, which is very far from the optimal utilization of wavelengths. We propose a novel algorithm, which is based on the maximum flow technique to obtain the maximum quantity of edge, disjoint paths. Here we compare the offered method with previous maximum edge disjoint paths algorithms applied to the RWA.

키워드 : RWA, 광 네트워크, Edge Disjoint Paths, Maximum Flow Technique

1. 서 론

광 네트워크는 광 기술과 광 구성요소를 기반으로 하는 대용량 통신 네트워크이다. 광 구성요소는 파장 기반 서비스[1] 뿐만 아니라 파장 레벨(Wavelength Level)에서 라우팅(Routing), 통합(Grooming), 복구(Restoration)를 제공한다. 광 네트워크의 핵심기술 중 하나인 파장분할 다중화(WDM)와 광 크로스 커넥터(OXC)을 기반으로 하는 새로운 기술들은 세계 정보망 뿐만 아니라 전국 백본망에서 인터넷 트래

픽 요구를 지원하는데 큰 역할을 하리라 기대하고 있다. 광 네트워크 기술에서 활발하게 진행되고 있는 연구분야 중 효율적인 라우팅과 선택된 경로에 대한 파장할당 문제는 네트워크 성능에 큰 영향을 미치는 중요한 분야이다[2]. RWA 알고리즘의 목적은 파장의 수를 최소화하고[3-5], 고정된 파장 수에 동적 연결 요구에 대한 블로킹 확률(Blocking Probability)을 줄이고[6, 7], 네트워크 비용을 줄이는 것[8, 9]이다.

가상 토폴로지, 파장분할 다중화(WDM)에 기반 한 광 네트워크상의 RWA문제는 NP-complete문제이므로 많은 발견적 해결 방법(Heuristic approach)[3-9]이 제안되었다. RWA는 여러 가지 조건 하에서 연구된다. 그 중 하나는 경로를 어떻게 설정하고 해제하느냐의 문제이다. 본 논문에서는 광

* 종신회원 : 성균관대학교 정보통신공학부 교수

** 정 회 원 : 한국산업기술대학교 컴퓨터공학과 교수

*** 정 회 원 : 한국항공대학교 통신정보공학과 교수

논문접수 : 2003년 9월 2일, 심사완료 : 2004년 7월 19일

경로가 한번 설정되면 해제되는데 긴 시간이 걸리는 정적 경로 할당을 가정한다. 정적 경로 할당에서는 광 경로를 미리 알고 있으며 다른 파장이 할당된 각 링크로 이루어진 광 경로에 동일한 파장을 할당해야 한다. 정적 경로 할당은 망 경로가 랜덤하게 설정되고 해제되는 동적 경로 할당보다 현재 기술에서 실질적으로 구현 가능하다.

또 다른 조건은 파장 변환기(Wavelength Converter)의 사용 유무이다. RWA문제의 물리적 제약 조건은 광 링크에서 파장 채널의 간격, 광학 송수신기의 수, 그리고 파장 변환기의 수이다. 이와 같은 제약조건은 사용 가능한 파장의 수와 출발점에서 도착점까지 연결의 수를 결정한다. 각 파장은 단일 채널에 할당된다고 가정한다. 그리고 문제를 간단하게 하기 위해서 각 채널에 대한 파장의 세분성(Granularity)은 고려하지 않는다. 본 논문에서는 정적인 파장 할당과 토폴로지에 파장 변환기가 없는 경우로 가정하였다. 제안하는 알고리즘은 파장 변환기가 존재하는 경우에도 쉽게 적용될 수 있을 것으로 예상된다. 이 연구의 목적은 사용하는 파장의 수를 줄이거나, 기존에 제시된 알고리즘과 거의 같은 파장의 수를 유지하면서 실행시간을 짧게 하는 것이다. 이것은 최대의 EDPs 기법을 통해서 해결될 수 있다. 일반 운용 컴퓨터에서 시뮬레이션 한 결과 제안된 알고리즘이 기존의 다른 알고리즘보다 실행시간 측면에서 상당히 좋은 성능을 지니고 있음을 알 수 있다.

본 논문은 다음과 같이 구성되어 있다. 먼저 RWA 알고리즘의 일반적인 3가지 방법인 *Ipsolver*를 이용한 라운딩 휴리스틱 방법(Rounding Heuristic Method)[3], BGAforEDP (Bounded Greedy Approach for Edge Disjoint Paths)[4], 그리고 DB_MAX and LTB(Demand Based Maximum EDP and Lookup Table Based RWA)[5] 알고리즘에 대해서 설명한다. 3장에서는 최대 EDPs를 기반으로 하는 새로운 알고리즘을 제안한다. 4장에서는 파장의 수, 실행 시간 측면에서 알고리즘의 성능 평가를 언급한다. 마지막으로 전체적인 결론을 5장에서 서술한다.

2. 관련 연구

RWA문제를 해결하기 위해서 기본적으로 두 가지 접근법이 있다. 하나는 RWA문제에 대한 최적의 해결책을 얻기 위해서 *Ipsolver*를 사용하는 방법인데, *Ipsolver*[10]는 선형 프로그래밍(Linear Programming)을 다중항목 흐름(Multicommodity Flows)개념을 이용하여 푸는 프로그램이다. 나머지 하나는 사용하는 파장의 수를 줄이기 위해서 경로 설정을 기초로 하는 그래프 알고리즘을 사용하는 방법이다. 본 장에서는 *Ipsolver*를 기반으로 하는 대표적인 알고리즘과 그래프 알고리즘에 기반한 최근에 제안된 두 가지 알고리즘에 대해 논한다.

2.1 경로 설정 및 파장할당 문제

본 논문은 파장 분할 다중화 방식을 이용한 광 네트워크

상의 경로 설정 및 파장 할당 문제에 초점을 맞추고 있다. 광 경로 설정은 망 구조의 기본 요소이므로 효율적인 경로 설정 및 할당은 매우 중요하다. 연결 요구에 광 경로를 설정하고 그 링크에 파장을 할당하는 것은 중요하다. 그리고 특정한 성능 측정 기준을 최적화하기 위해서 가능한 여러 가지 경로 중에서 최적화 된 경로를 선택해서 할당해야 한다. 이것을 경로 설정 및 파장 할당 문제(Routing and wavelength assignment problem)라 한다.

광 경로를 설정 할 때 경로가 선택되어지면 그 경로에 걸쳐진 모든 링크에 하나의 파장이 할당 되어야 한다. 만약 경로를 위한 사용 가능한 파장이 없다면 그 연결은 차단된다. 경로 설정 및 파장 할당 문제는 출발점에서 도착점까지 트래픽에 대한 가정과 성능 측정 기준에 따라 다르게 접근하고, 일반적으로 트래픽에 두 가지로 분류한다. 정적 트래픽에서는 출발점에서 도착점까지 연결요구의 집합이 미리 주어진다. 동적 트래픽에서는 연결 요구가 랜덤하게 주어지고 해제된다. 경로 설정 및 파장 할당은 NP-complete문제로 알려져 있으며 경로 설정과 파장 할당 문제 또한 각각 NP-complete 문제이다. 동적 트래픽의 경우 경로 설정 및 파장 할당의 목적은 광 경로를 설정하고 파장의 수와 블록킹 확률을 줄이는데 있다. 즉 고정된 파장의 수로 연결요청의 수를 최적화하는 것이다.

2.2절~2.4절까지는 절에서는 기존에 제시된 경로 설정 및 파장 할당 알고리즘에 대해 간단하게 서술한다. 본 논문에서 우리는 잘 알려진 세 가지의 경로 설정 및 파장 할당 알고리즘들을 평가해 보겠다.

2.2 라운딩 휴리스틱 알고리즘과 그래프 컬러링을 이용하는 방법

이 방법에서는 RWA 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. ① 정수 제약이 없는 다중항목 흐름 문제를 해결 ② 경로 탐색 ③ 랜덤하게 경로 설정하기. 스텝 1에서는 일반적인 *Ipsolver*를 이용하여 각 요구에 대한 네트워크에서의 링크 흐름을 얻는다. 여기에서 사용하는 *Ipsolver*에서는 소수를 허용하기 위해서 0~1 흐름에 대한 제약을 없애고 사용한다. 스텝 2에서는 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 나오게 되는 면을 가진 경로가 연결 요구 i 에 할당된다. 모든 연결 요구에 이 알고리즘을 적용함으로써 라운딩 휴리스틱 알고리즘은 각각의 연결 요구 i 에 대한 경로를 얻게 된다.

파장 할당 단계에서는 그래프 $G_R(V_R, E_R)$ 를 생성하게 된다. 그래프 G_R 을 얻은 후에는, 그래프에서 이웃 하는 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.

2.3 BGAforEDP에 기초한 알고리즘

이 알고리즘은 최단거리 알고리즘에 기초한 간단한 EDP 기법을 사용한다[4]. 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: $\tau = \{(s_1, t_1), \dots, (s_k, t_k)\}$.

BGAforEDP는 다음 3개의 파라미터를 가지고 동작한다: 초기 그래프, 연결될 요구 집합, 적절한 d . 여기에서 d 는 $\max(\text{diam}(G), \sqrt{|E|})$ 이다[11]. 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 $\alpha(G_B, \tau)$ 에 포함시킨다. 그리고 P_i 에서 사용하는 에지들을 G_B 에서 삭제한다. 만약 연결된 경로 P_i 의 길이가 d 보다 크다면, 연결 요구 i 는 경로를 할당 받지 못한다. 이와 같은 과정이 연결요구가 모두 만족이 될 때까지 반복된다. 알고리즘이 끝난 후에 $\alpha(G_B, \tau)$ 는 같은 파장이 할당될 수 있는 연결 요구들의 집합을 가지고 있게 된다. $\alpha(G_B, \tau)$ 를 τ 에서 삭제함으로써 경로 설정이 아직 되지 않은 연결 요구 집합인 τ' 을 구한다. 만약 이 공집합이 아니라면 다른 파장에 할당 가능한 연결 요구들의 집합을 찾기 위해서 BGAforEDP 알고리즘을 초기의 그래프 G_B 와 τ' 을 파라미터로 하여 다시 실행한다. τ' 이 공집합이 될 때까지 이를 반복 시행한다. 전체적으로 할당된 파장의 수가 이 알고리즘의 결과값이 된다.

이 논문에서 성능평가 결과, 라운딩 휴리스틱 알고리즘과 BGAforEDP에 기초한 알고리즘 두 방법의 파장수는 거의 같지만 수행시간은 BGAforEDP가 우수하다. 따라서 본 논문에서는 제안하는 방식과 BGAforEDP에 기초로 한 알고리즘에 대한 성능 비교를 하기로 한다.

2.4 Lookup Table을 이용한 RWA 알고리즘(LTB_RWA)

LTB_RWA 알고리즘은 최대 흐름 기법(Maximum Flow Technique)을 기초로 한다. G_L 은 노드와 에지의 집합으로 이루어진 네트워크 모델을 나타낸다. 그리고 τ 은 앞에서 정의한 바와 같이 연결 요구 집합을 나타낸다.

이 알고리즘은 랜덤하게 그래프를 생성하고, 각 노드 쌍들 간에 연결 요구가 있을 확률과 에지가 존재할 확률을 설정하여 실행한다. 연결 요구는 균등 분포로 랜덤하게 출발점과 도착점을 선택한다. LTB_RWA 알고리즘은 각 연결 요구에 대하여 최대 EDPs를 가지는 Lookup Table을 만든다.

이는 최초의 토폴로지 G_L 이 바뀌기 전에는 다른 연결 요구 집합들에도 사용된다. Lookup Table을 얻는 동안 각 에지의 가중치 또한 계산한다. Lookup Table을 얻고 난 후, 연결 요구에 대하여 Lookup Table에 저장된 EDPs중 가장 짧은 경로를 할당하고 그래프에서 할당된 에지들을 모두 지운다. 만약 어떤 연결 요구에 대해 더 이상 사용 가능한 경로가 없다면 최초의 그래프에서 다른 파장을 할당하여 위와 같은 과정을 반복한다. 모든 연결 요구가 고려될 때까지 계속된다.

3. 본 논문에서 제안 알고리즘

네트워크 모델이 되는 그래프를 $G(V, E)$ 라고 하자. 여기서 V 는 노드의 집합이고 E 는 에지의 집합이다. 네트워크가 n 개의 호스트를 가질 때 $|V|=n$ 이다. 연결 요구 집합 D 는 $s_i, t_i \in V(i=1, \dots, k)$ 일 때, $\{(s_1, t_1), \dots, (s_k, t_k)\}$ 이다. 여기서 k 는 전체 연결 요구 개수이며, s_i 는 전송측, t_i 는 수신측이다. 연결 요구 (s_i, t_i) 에 대해 파장 λ_i 와 광 경로가 할당되어야 한다. 이때 광 경로를 이루는 에지들은 에지의 집합인 E 의 원소여야 한다. 동시에 동일한 파장이 할당된 광 경로가 있을 경우 같은 물리적 링크를 지날 수 없다는 것을 인지해야 한다. 즉 이때 각 광 경로의 에지들은 겹치지 않아야 한다. $\ell = \max_i \lambda_i$ 라고 하자. RWA 문제의 목적은 $L_i \cap L_j = \emptyset (\forall i, j \in \{1, \dots, k\})$ 일 때 ℓ 의 개수를 최소화하는 것이다. 이 조건이 거짓이면, 할당이 적합하지 않다고 말한다.

만약 $s_1 = s_2 = \dots, s_k = s$ 와 $t_1 = t_2 = \dots, t_k = t$ 이면, 문제의 최적화된 해결 법에서는 s 와 t 사이의 EDPs의 최대 개수가 k 인 것이다. 즉 k 를 s 와 t 사이의 에지 단절 집합(Edge Cut Set)의 개수인(Cardinality)인 $Card(s, t)$ 로 나눈다.

3.1 등 기

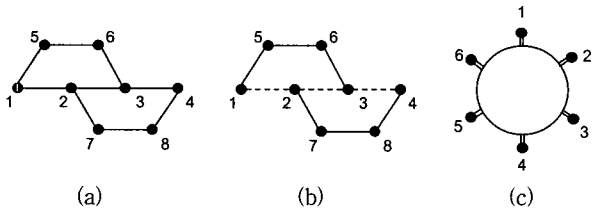
선형 프로그램과 그래프 컬러링에 의한 접근법[2, 3]은 매우 복잡하고, 발견적 방법이 많이 사용된다. RWA[4]에 적용된 EDPs기법을 고려해보자. 이 방법에 의한 해결 법은 자명할 수 있다.

$$\text{Optimal number of wavelengths} = \lceil \frac{k}{Card(s, t)} \rceil$$

(그림 1)(a)의 광 네트워크를 보자. 연결 요구에 대해 가장 짧은 경로를 할당한다. 연결 요구의 집합이 (1, 4), (4, 1), (2, 3)이라고 하자. 먼저 임의로 선택된 연결 요구는 (1, 4)이다. 노드 1과 4사이의 가장 짧은 경로는 (1, 2, 3, 4)이다. 그러므로 연결요구(1, 4)는 파장 $\lambda=1$, 광 경로는 (1, 2, 3, 4)이다. 그래프에서 이 경로를 지우면 연결되지 않는 그래프가 된다. (그림 1)(b)의 그래프에는 다른 연결 요구를 만족시킬 수 있는 경로가 없기 때문에 처음의 그래프로 돌아온다. 그리고 파장의 수를 2로 증가시킨다. 두 번째로 선택된 연결 요구는 (4, 1)이다. 가장 짧은 경로가 (4, 3, 2, 1)이다. 그러므로 연결 요구 (4, 1)은 경로 (4, 3, 2, 1)와 파장 $\lambda=2$ 을 할당 받는다. 연결 요구 (2, 3)을 만족하는 경로가 없기 때문에 파장을 $\lambda=3$

으로 증가시킨다. 그 결과 우리는 각 연결 요구에 대해 다른 파장을 사용한다. 즉 파장 3개를 사용한다. 하지만 연결 요구 (1,4)가 경로 (1,5,6,3,4)를, 연결 요구 (4,1)이 경로 (4,8,7,2,1), 연결 요구 (2,3)이 경로 (2,3)을 사용한다면 사용되는 최대 파장의 개수는 1이다.

링 토폴로지는 중요한 네트워크 토폴로지중의 하나이다. (그림 1)(c)를 보자. 연결 요구 집합이 $\{(1, 2), (6, 2)\}$ 이다. 이 그래프의 직경(Diameter)은 3이다. 그러므로 두 번째 연결 요구인 (6,2)는 경로 (6,5,4,3,2)를 사용할 수 없다. 왜냐하면 이 경로의 길이는 직경을 넘기 때문이다. 연결 요구 (1,2)는 경로 (1,2)와 파장 $\lambda=1$ 을 사용하고, 연결 요구 (6,2)는 경로 (6,1,2)와 파장 $\lambda=2$ 을 사용한다. 그러나 EDPs를 사용하면 1개의 파장만으로 충분하다.



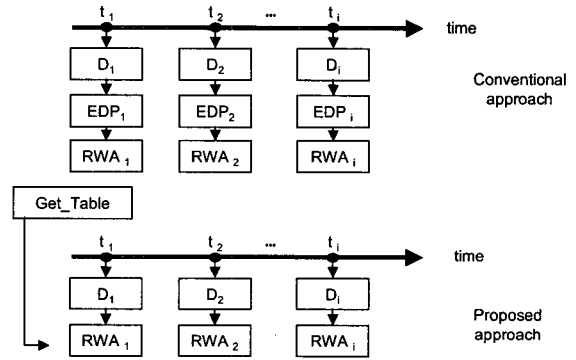
(그림 1) RWA를 위한 토폴로지 기본

3.2 경로설정 및 파장할당 방식

본 논문에서는 효율적인 파장 할당을 위한 EDPs를 구하기 위해 에지가 무 방향이며 수용량이 1인 그래프 G에서, 최대 흐름 문제를 (maximal flow problem)을 이용하여 푼다. 일반적으로 임의의 i와 j($i \neq j$)에 대해 (s_i, t_i) 와 (s_j, t_j) 는 다르다. 두 노드 사이의 트래픽을 처리하기 위한 최적화된 RWA 해결 법을 얻을 수 있다. 만약 이러한 해결법의 집합이 수용된다면 최적의 공용 해결 법을 찾을 수 있다. 만약 그렇지 않으면 경로상에 할당된 파장이 다른 파장으로 대체되어야 한다. 여기서 이러한 방식으로 최적의 해결 법을 얻을 필요는 없다. 더 나아가 임의의 i와 j에 대해 $\lambda_i \neq \lambda_j$ 일 때, 앞의 예제와 같이 파장 할당이 자명하지 않다.

네트워크 토폴로지는 고정되어 있기 때문에 연결 요구에 대한 사용 가능한 EDPs를 미리 계산할 수 있다. Get_Table(G)함수에서 EDPs를 구하고 이것을 사용한다. $s=1, 2, \dots, n-1; t=1, 2, \dots, n-1; i=1, 2, \dots, Card(s, t)$ 일 때, Get_Table(G) 함수의 결과값은 $P(s, t, i)$ 이다. 출발점 s와 도착점 t까지 모든 EDPs와 최대량은 그래프 G에서 최대 단일 흐름 해결법을 통해서 구할 수 있다. 여기서 하나의 흐름은 하나의 경로에 해당한다. 최대 흐름 해결 법에 대한 자세한 내용은 [12]에 나와있다. 본 논문에서는 각 연결 요구에 대한 최대한의 EDPs를 구하기 위해서 DPSP(Disjoint Pathset Selection Protocol)[13] 개념을 사용한다. DPSP의 목적은 [13]은 본 논문과 다르기는 하지만 EDPs를 선형 시간에 구하는데 이를 사용할 수는 있다. (그림 2)는 제안된 알고리즘과 기존에 제시된 알고리즘 기법을 나타낸다. 최대 단일 흐름 알고리즘을 올바르게 구현하면 실행시간이 $O(E \cdot Card(s, t))$ 이다.

네트워크 토폴로지는 링크와 호스트간의 장애나 네트워크 확장으로 변할 수 있다. 이런 상황일 때 링크가 겹치지 않는 경로인 $P(s_i, t_i, i)$ 를 갱신하고 RWA문제 해결 과정을 계속 진행한다.



(그림 2) 두 가지 접근법 비교

제안하는 방식의 의사코드는 다음과 같다.

```

Algorithm Max_EDP_RWA
Begin
Get_Table(G)
While (the same G) do
  Retrieve a new D
  λ = 0
  while (D ≠ 0) do
    λ = λ + 1
    for j = 1 to k do
      if ∃ empty path P(s_j, t_j, i) then
        Assign P(s_j, t_j, i) and λ to (s_j, t_j)
        D = D - (s_j, t_j)
      end if
    end for
  end while
end
End.
    
```

4. 성능 평가

본 장에서는 3장에 기술되어있는 제안한 방식과 [4,5]에 논의된 방식에 대해서 비교한다. 이때 비교의 척도는 사용되는 파장의 수와 수행 시간이다.

4.1 시뮬레이션을 위한 토폴로지

시뮬레이션을 위한 자세한 사항은 다음과 같다. Graph_Generation() 함수를 이용해서 임의의 그래프 생성한다. 임의의 그래프는 물리적 토폴로지를 나타낸다. 그래프에 존재하는 노드의 수(N)와 각 노드 쌍들 간에 에지의 존재 확률(P_e)에 의해서 랜덤 그래프를 생성한다. 랜덤 그래프는 네트워크 모델이므로 연결 되어 있어야 한다. 그러므로 랜덤 그래프는 적어도 하나의 스패닝 트리로 이루어져 있어야 한다. $N \geq 3$ 인 경우를 고려하여 스패닝 트리를 생성한다. 노드가 3인 트리는 유일하며 이것을 원시 트리라고 사용한다. P_e 를 조정하면 다음 랜덤하게 다른 에지들을 생성한다.

조정된 확률인 P_e^a 를 계산한다. $Prob(event)$ 은 사건이 일어날 확률을 나타내는 함수이다. e 를 두 노드간에 에지가 있을 확률이라고 가정하면

$$P_e = Prob(e \in \text{spanning tree}) + Prob(e \notin \text{spanning tree}) \cdot P_e^a,$$

$$P_e = \frac{n-1}{n(n-1)} + \left(1 - \frac{n-1}{n(n-1)}\right) P_e^a,$$

$$P_e^a = \frac{nP_e - 2}{(n-2)},$$

이다.

랜덤하게 그래프를 생성하는 수도코드는 아래와 같다. 여기서 A 는 사건 행렬(incident matrix)이다. r 은 간단한 변수이며, $random()$ 은 0과 1사이 값으로, 균등하게 분포된 임의의 값을 생성하는 함수이다.

```

Algorithm Graph_Generation
Begin
A12 = A21 = A23 = A32 = 1
for i = 4 to n do
r = (i - 1) × random() + 1
Ari = Air = 1
end for
for i = 1 to (n-1) do
for j = (i+1) to n do
if Pe > random() then Aij = Aji = 1
end
end
End
    
```

아래의 수도코드는 연결 요구 집합을 생성하기 위해 이용된다. 하나의 연결 요구는 P_e 의 확률로 발생한다. N_c 는 단일 연결 요구가 중복되는 정도를 나타내고, $random()$ 은 위에 설명한 바와 같다. (s, t) 한 쌍이 나올 수 있는 개수는 0과 N_c 사이의 값이다. 즉 중복될 수 있는 트래픽의 수를 가리킨다.

```

Algorithm Demand_set
Begin
for h = 1 to Nc do
for i = 1 to (n-1) do
for j = (i+1) to n do
if Pe > random() then generate new request (i, j)
end for j
end for i
end for h
End
    
```

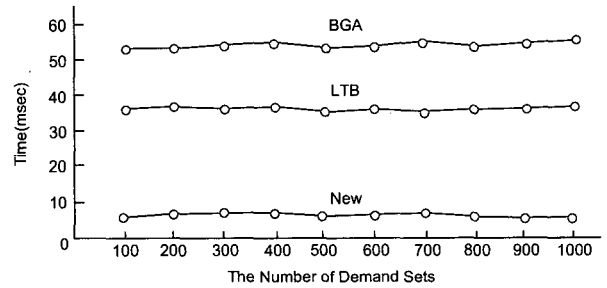
4.2 실험 결과

본 장에서는 논문에서 제안하는 알고리즘과 LTB_RWA 알고리즘[5], Greedy_EDP_RWA에 기반한 BGAforEDP 알고리즘[4]의 성능을 평가하고 비교한다. 비교의 척도는 파장의 수와 실행 시간이다. [5]에서 같은 네트워크 토폴로지에서 Lookup Table을 재사용하기 때문에 수행시간을 측정하기 위해 [5]의 NSFNET을 이용한다. 그러나 알고리즘을 여러

토폴로지에서 평가하기 위하여 파장의 수를 측정하는 경우에는 랜덤하게 생성한 그래프를 사용한다.

사용한 파장의 수를 비교하기 위한 시뮬레이션에서, 위에 소개한 알고리즘에 따라 랜덤 그래프와 연결요구 집합을 생성한다. 본 논문에서는 각각 N , P_e , 그리고 P_e 에 대하여 1000번의 시뮬레이션을 수행하여 평균적으로 사용되는 파장의 수를 구하였다. <표 1>은 그래프 G 와 랜덤하게 발생시킨 연결 요구에 대하여 사용하는 평균 파장 수를 보여주고 있다. 제안된 알고리즘(<표 1>에서 New로 명시)에서 사용되는 파장의 수는 Greedy_EDP_RWA에 기반한 BGAforEDP(BGA로 명시)와 LTB_RWA(LTB로 명시)에서 사용하는 파장의 수와 거의 비슷하다.

세 가지 알고리즘을 위와 같은 방식으로 얻은 연결 요구 집합에 대해 NSFNET에서 실행 시간을 측정하였다. 100개 부터 1,000개까지의 연결 요구 집합을 따로 발생시켜서 평균 수행 시간을 구한다. (그림 3)는 $P_e = 0.6$ $N_c = 5$ 일 때 평균 실행 시간을 보여준다. 다른 P_e 값에 대해서도 비슷한 경향을 보인다. 결과적으로 제안된 알고리즘은 N_c 가 1일 때 파장의 수는 거의 같지만, N_c 가 커지면 다른 알고리즘보다 나은 성능을 나타낸다. 게다가 제안된 알고리즘은 다른 알고리즘에 비해 실행 시간이 매우 짧다.



(그림 3) 평균 실행시간(NSFNET, $P_e = 0.6$, $N_c = 5$)

<표 1> 성능 비교 : 사용한 파장의 수

N	P _e	P _e	N _c = 1			N _c = 3			N _c = 5		
			BGA	LTB	New	BGA	LTB	New	BGA	LTB	New
10	0.6	0.4	2.02	20.9	2.1	5.44	5.38	5.39	8.81	9.04	8.63
10	0.6	0.6	2.69	2.78	2.75	8.1	7.66	7.86	13.22	13.27	12.74
10	0.6	0.8	3.49	3.54	3.53	10.55	10.34	10.2	17.34	17.69	16.7
10	0.8	0.4	1.25	1.33	1.34	3.6	3.56	3.57	5.88	5.86	5.74
10	0.8	0.6	1.92	1.94	1.94	5.24	5.11	5.11	8.6	8.68	8.26
10	0.8	0.8	2.15	2.17	2.18	6.8	6.68	6.6	11.44	11.52	10.93
14	0.6	0.4	1.99	2.06	2.05	5.36	5.23	5.2	8.95	8.99	8.58
14	0.6	0.6	2.74	2.79	2.8	8.03	7.53	7.68	13.19	13.18	12.5
14	0.6	0.8	3.44	3.42	3.46	10.37	10.05	9.88	17.34	17.60	16.21
14	0.8	0.4	1.22	1.31	1.34	3.68	3.64	3.6	6.05	6.0	5.82
14	0.8	0.6	1.97	1.99	1.98	5.33	5.21	5.19	8.95	8.96	8.55
14	0.8	0.8	2.2	2.23	2.27	7.10	6.84	6.79	11.87	11.89	11.16
18	0.6	0.4	2.06	2.03	2.06	5.45	5.21	5.33	8.9	8.82	8.42
18	0.6	0.6	2.67	2.84	2.73	7.86	7.53	7.5	13.08	13.09	12.24
18	0.6	0.8	3.38	3.0	3.38	10.48	9.85	9.81	17.88	17.57	16.61
18	0.8	0.4	1.15	1.21	1.26	3.73	3.74	3.67	6.09	6.16	5.86
18	0.8	0.6	1.97	2.0	1.98	5.47	5.27	5.26	9.12	9.13	8.58
18	0.8	0.8	2.28	2.24	2.35	7.27	6.98	6.91	12.2	12.17	11.4

5. 결 론

본 논문에서 광 네트워크상의 RWA 문제를 풀기 위한 최
대량 EDPs에 기반한 해결 법을 제시하였다. 연구 목적은 다
른 알고리즘보다 더 빠른 시간 안에 필요한 파장의 수를 최
소화하는 것이다. 그리고 정적 경로 할당과 파장 변환기가
없는 경우를 가정하여 문제해결을 시도하였다. 제안된 알고
리즘은 각 연결 요구인 연결 요구 집합에 최대 EDPs를 얻
고 테이블에 저장한다. 연결 요구는 테이블에 참조하여 경
로를 할당한다. 기존에 구한 Lookup Table은 네트워크 토폴
로지가 바뀌지 않을 경우 재사용할 수 있다. 일반 운영 컴퓨
터에서 시뮬레이션 한 결과 제안된 알고리즘은 중복 개수
(N_c)가 1에 가까울 때 다른 알고리즘에서 사용한 파장의 수
와 거의 비슷하나, 중복 개수(N_c)가 클 경우에는 좋은 성능
을 나타내었다. 더 나아가 제안된 알고리즘은 다른 알고리
즘보다 아주 짧은 시간 내에 거의 비슷한 파장을 사용하
면서 RWA문제를 해결할 수 있다.

참 고 문 헌

[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] R. Ramaswami and K. Sivarajan, "Routing and Wavelength Assignment in All-Optical Network," IEEE/ACM Trans. Networking, Vol.3, No.5, pp.489-500, 1995.

[3] D. Banerjee and B. Mukherjee, "Practical Approach for Routing and Wavelength Assignment in Large Wavelength Routed Optical Networks," IEEE JSAC, Vol.14, No.5, pp. 903-908, 1996.

[4] P. Manohar, D. Manjunath and R. K. Shevgaonkar, "Routing and Wavelength Assignment in Optical Network From Edge Disjoint Path Algorithms," IEEE Communications Letters, Vol.6, No.5, pp.211-213, May, 2002.

[5] M. H. Kim, H. Choo, B. H. Yae and J. H. Lee, "A Practical RWA Algorithm Based on Lookup Table for Edge Disjoint Paths," Springer-Verlag Lecture Notes in Computer Science, Vol.2713, pp.138-147, June, 2003.

[6] 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.

[7] 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.

[8] 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.

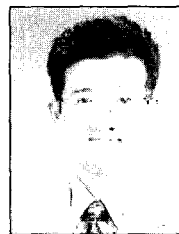
[9] I. Chlamtac, A. Farago and T. Zhang, "Lightpath (wavelength) routing in large WDM networks," IEEE J. Sel. Areas in Communs, Vol.14, No.5, pp.909-913, June, 1996.

[10] <http://only1.snu.ac.kr/software/lpsolver/>.

[11] J. M. Kleinberg, "Approximation algorithms for disjoint paths problems," Ph.D. dissertation, MIT, Cambridge, May, 1996.

[12] L. R. Ford and D. R. Fulkerson, "Flows in Network, Princeton University Press," Princeton, 1962.

[13] P. Papadimitratos, Z. J. Haas and E. G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," MobiHoc, 2002.



추 현 승

e-mail : choo@ece.skku.ac.kr

1988년 성균관대학교 수학과(학사)

1990년 University of Texas at Dallas,
컴퓨터공학(석사)

1996년 University of Texas at
Arlington, 컴퓨터공학(박사)

1997년 특허청 심사관(사무관)

1998년~현재 성균관대학교 정보통신공학부 부교수

관심분야 : 광네트워크, 이동컴퓨팅, 라우팅 프로토콜, 그리드
컴퓨팅 등



정 성 택

e-mail : unitaek@kpu.ac.kr

1992년 한국과학기술원 전기 및 전자공학과
(학사)

1995년 한국과학기술원 정보 및 통신공학과
(석사)

2000년 한국과학기술원 전기 및 전자공학과
(박사)

1998년~2000년 메디슨 MRI 연구소, 선임연구원

2000년~2004년 메디너스 MRI연구소장

2004년~현재 한국산업기술대학교 컴퓨터공학과 전임강사

관심분야 : Image Computing System, 영상처리, 디지털신호처리 등



이 성 창

e-mail : sclee@hau.ac.kr

1983년 경북대학교 전자공학과(학사)

1985년 한국과학기술원 전기 및 전자공학과
(공학석사)

1991년 미국 Texas A&M University
전자공학과(공학박사)

1985년~1987년 한국과학기술원 시스템공학센터 연구원.

1992년~1993년 한국전자통신연구원 선임연구원.

1993년~현재 한국항공대학교 통신정보공학과 교수

관심분야 : 광대역통합망(BcN), 디지털 홈, 유비쿼터스 네트워크 등