

광 전달망에서 계획 연결 요구의 시간적 비공유 경로를 이용한 RWA

준회원 안 현 기*, 정회원 이 태 진*, 정 민 영**, 추 현 승***

Time Disjoint Paths (TDP) - RWA on Scheduled Lightpath Demands in Optical Transport Networks

Hyun Gi Ahn* *Associate Members*,
Tae-Jin Lee*, Min Young Chung**, Hyunseung Choo*** *Regular Members*

요 약

광전달망(Optical Transport Network)에서 사용자의 서비스 요구는 종종 주기적인 특성을 보이므로 이러한 계획 연결 요구에 대하여 시간 중복성을 RWA문제에 이용할 수 있다. 계획 연결 요구의 RWA 문제는 조합 최적화 방법(Combinatorial Optimal Solution)과 그래프 컬러링(Graph Coloring)의 조합으로 해결되거나, 경로 선택에 기반한 순차적(sequential) RWA(sRWA) 의해서 해결된다. 이와 같은 방법은 복잡하거나 많은 연산을 필요로 하므로 본 논문에서는 그룹화를 통해 계획 연결 요구들의 시간 중복성을 이용하여 효과적으로 경로 설정 및 파장 할당을 하는 알고리즘을 제안한다. 본 논문에서 제안하는 알고리즘이 현재 실질적으로 효율적이라 알려진 sRWA 보다 같거나 작은 파장 수를 사용하여 54% 이상의 계산량의 성능 향상을 보인다.

Key Words : Routing and wavelength assignment (RWA), Time disjoint paths (TDP), Scheduled Lightpath Demand (SLD)

ABSTRACT

In optical networks, traffic demands often demonstrate periodic nature for which time-overlapping property can be utilized in routing and wavelength assignment (RWA). A RWA problem for scheduled lightpath demands (SLDs) has been solved by combinatorial optimal solution (COS) and graph coloring, or heuristic sequential RWA (sRWA). Such methods are very complex and incurs large computational overhead. In this paper, we propose an efficient RWA algorithm to utilize the time disjoint property as well as space disjoint property through fast grouping of SLDs. The computer simulation shows that our proposed algorithm indeed achieves up to 54% faster computation with similar number of wavelengths than the existing heuristic sRWA algorithm.

I. 서 론

광 전달망(Optical Transport Network)^[1]에 의해

서 서비스 되는 네트워크를 가상 광 개인 망(Optical Virtual Private Network)이라 한다. 광 개인망에서는 사용자의 서비스 요구에 따라 연결 요구를

* 성균관대학교 정보통신공학부 네트워크시스템 연구실 (puppybit, tjlee)@ece.skku.ac.kr

** 성균관대학교 정보통신공학부 네트워크기술 연구실 (mychung)@ece.skku.ac.kr

*** 성균관대학교 정보통신공학부 통신망 연구실 (choo)@ece.skku.ac.kr

논문번호 : KICS2005-01-006, 접수일자 : 2005년 1월 3일

※ 본 논문은 BK21과 정보통신부 지원에 의하여 연구되었음.

세가지로 분류 할 수 있다. 정적 요구(Static Lightpath Demands), 계획 연결 요구(Scheduled Lightpath Demands), 동적 연결 요구(Dynamic Lightpath Demands)들이 바로 이것이다. 광 개인망을 이용하는 소비자는 최소한의 접속 요구와 네트워크 용량을 확보하기 위해서 정적인 연결 요구들을 요청할 수 있다. 그리고 특정한 링크(link)에서 어떤 시간에 네트워크의 용량을 늘리기 위해 몇 개의 계획 연결 요구들을 요청한다. 예를 들면 업무시간에 회사 본부와 생산 기지 사이, 또는 한 밤중에 데이터 센터들 간에 데이터 백업이 필요할 경우 등이다. 마지막으로 예상하지 못했던 트래픽으로부터 동적인 연결 요구를 요청하기도 한다.

광 전달망이 실질적으로 구성이 된다면 정적 연결 요구와 계획 연결 요구가 주요한 연결 요구가 될 것이다. 그 이유는 광 전달망의 트래픽(traffic)이 자연스러운 주기적인 현상으로 예측 가능하게 될 것이기 때문이다^{[2],[3]}. 그림 1은 이러한 현상을 잘 보여준다. 이 그림에서는 일정한 기간을 사이에 두고 유사한 주기적 패턴을 관찰할 수 있다. 물론 반드시 네트워크 종단간의 트래픽의 양을 나타낸 것이 아니며 일반적인 상업용 네트워크의 트래픽 양과는 다를 수 있다. 하지만 네트워크 사용자간의 트래픽 밀집 현상을 보여준다.(업무 시간에 트래픽이 더 많다.)

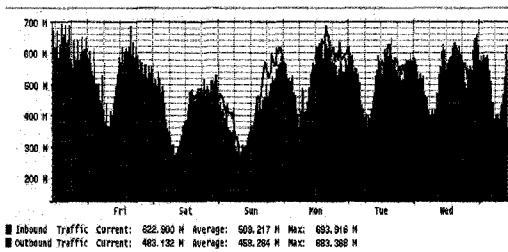


그림 1. 2003년 4월 3일부터 10일 동안의Abilene 백본 네트워크: 뉴욕-워싱턴 간 링크의 트래픽^[3]

RWA(Routing and Wavelength Assignment) 문제는 WDM(Wavelength Division Multiplexing)을 기반으로 하는 광 네트워크에서 중요한 분야이다^[4]. RWA가 광 네트워크의 가격과 성능에 많은 영향을 끼치고 있기 때문에 다양한 최적화 방법이 제안되고 있다. RWA 알고리즘의 목적은 필요한 파장의 수를 줄이거나, 고정된 파장 수에 동적 연결 요구에 대한 충돌 확률(Blocking Probability)을 줄이는 것, 그리고 네트워크 비용을 줄이는 것 등을 포함한다.

계획 연결 요구는 시작 지점(source node), 도착 지점(destination node) 외에 서비스 시작 시간(setup time), 종료 시간(teardown time)을 가지고 있다. 그러므로 서비스의 시작 시간과 종료시간을 미리 알 수 있는 계획 연결 요구는 서로 시간적으로 중복되지 않은 연결 요구들의 특성을 이용할 수 있다. 만약 두 계획 연결 요구가 시간적으로 겹치지 않는다면 경로의 겹침에 상관없이 두 계획 연결 요구는 같은 파장으로 할당 할 수 있다. 그래서 본 논문에서는 이러한 계획 연결 요구의 시간적 비공유 현상을 최대한 많이 이용하기 위해 계획 연결 요구들을 서비스 시간에 따라 그룹화 하는 휴리스틱 RWA 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 계획 연결 요구에 대한 관련 연구를 소개하며 3장에서는 새로운 알고리즘에 대해서 소개를 한다. 그리고 4장에서는 본 논문에서 제안하는 알고리즘과 기존의 다양한 알고리즘들과 비교하여 성능평가를 한다. 마지막으로 5장에서 결론을 맺는다.

II. 관련 연구

휴리스틱 RWA 알고리즘 중에서 가장 대표적인 것은 BGAforEDP 알고리즘이다^{[5][6]}. 이 알고리즘은 최단거리 알고리즘에 기초한 간단한 EDP 기법을 사용한다^[6]. V_B 와 E_B 가 각각 노드의 집합과 엣지(edge)의 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 $\max(\text{diam}(G), \sqrt{|\tau|})$ 이다^[8]. 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 엣지(edge)들을 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 공집합이 아니라면 다른 파장에 할당 가능한 연결 요구들의 집합을 찾기 위해서 BGAforEDP 알고리즘을 초기의 그래프 GB와 τ' 을 파라미터로 하여 다시 실행한다. τ' 이 공집합이 될 때까지 이를 반복 시행한다. 전체적으로 할당된 파장의 수가 이 알고리즘의 결과값이 된다.

BGAforEDP 알고리즘은 계획 연결 요구에 적합하지 않은 알고리즘이다. 왜냐하면 계획 연결 요구는 시작 지점(source node)과 도착 지점(destination node) 외에도 연결 요구를 서비스하는 시작 시간과 종료 시간이 존재하기 때문이다⁴⁾. 따라서 [3]에서는 이와 같은 계획 연결 요구에 적합한 조합 최적화 알고리즘과 이에 대해서 시간 비용을 줄이기 위한 휴리스틱 알고리즘을 제안하였다. 조합 최적화 방법과 그래프 컬러링⁷⁾ 방법은 매우 복잡하고 시간 비용이 크다. 특히 연결 요구가 많아지면 그 비용은 기하급수적으로 늘어난다. 예를 들어 3개의 최단 거리 경로에 30개의 연결 요구를 고려한다면 무려 3^{30} 회의 연산이 필요하다. [3]에서는 브랜치와 바운드(Branch & Bound) 검색(search) 알고리즘⁹⁾^[10]과 타부검색(Tabu Search) 알고리즘¹¹⁾^[12]을 이용해 계산량을 줄였지만 여전히 비교적 많은 시간 비용을 필요로 한다. 또한 계획 연결 요구에 적합한 First Fit(FF)⁵⁾ 알고리즘에 기반한 sRWA 휴리스틱 알고리즘은 시간적 비용은 비교적 적게 들지만 구현하기가 용이하지 않다. 본 논문에서는 계획된 연결 요구에 적절하게 구현하기 간편하면서 시간 비용이 적은 휴리스틱 알고리즘을 제안한다.

III. 제안하는 RWA 알고리즘

계획 요구에서는 정적 연결 요구와는 달리 서비스를 시작하는 시간과 종료하는 시간이 있다. 정적 연결 요구에서는 같은 경로를 거치는 두 연결 요구는 하나의 파장으로 할당할 수 없으나 계획 연결 요구에서는 서비스 시간이 서로 중복되지 않으면 서로 같은 파장을 할당할 수 있다. 그림 2에서 계획 연결 요구 1과 2의 최단거리를 구해보자. 요구 1의 최단거리와 연결 요구2의 최단거리는 서로 겹치는 구간이 존재한다. 정적 요구라면 서로 같은 파장을 할당할 수 없지만 계획 연결 요구인 두 연결 요구는 서비스 시간이 중복되지 않기 때문에 같은 파장을 할당할 수 있다. 본 논문에서는 이와 같이 계획 연결 요구들의 시간적 비공유 경로들을 가능한

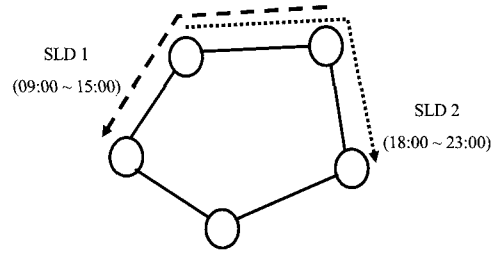


그림 2. 시간적 비공유 경로

```

/* G(V,E) : network, λ : wavelength number, Δ : set of
SLDs, ΔT : sets of time-disjoint SLDs
P(G, Δ) : set of the assigned shortest paths of demands,
a(G, Δ) : set of assigned SLD */

Input : G, Δ
Output : λ
01: Algorithm TDP-RWA(G, Δ)
02: Greedy-TDP-Selector(G, Δ)
03: d = max(diam(G), √|E|)
04: λ = 0
05: while(Δ ≠ ∅)
06:     λ = λ + 1
07:     RWAforTDP(G, ΔT, d)
08:     Assign λ to all paths in P(G, Δ)
09:     Δ = Δ - a(G, Δ)
    
```

그림 3. TDP-RWA 알고리즘

많이 그룹화하여 파장 할당 및 경로 설정을 한다.

TDP-RWA 알고리즘은 시간적 비공유 연결요구를 그룹화 하는 부분과 경로 설정 및 파장 할당하는 두 부분으로 구성된다. 그림 3은 TDP-RWA 알고리즘의 의사 코드이다.

$G(V, E)$ 를 V 개의 노드와 E 개의 링크를 가진 네트워크라 하자. $\lambda, \Delta, \Delta_T, P(G, \Delta), a(G, \Delta)$ 는 각각 파장, 연결 요구 집합, 그룹화된 연결 요구 집합, 연결 요구의 최단거리들이 지나가는 간선들(edge)의 집합, 파장이 할당 되어진 경로들의 집합을 나타낸다. 먼저 TDP-Selector 함수에 의해서 연결 요구들을 그룹화 하고 결과로써 Δ_T 를 생성한다 (라인 2). 그리고 적절한 d 값을 사용하여 불필요하게 긴 경로를 제한하는 효과를 갖도록 $\max(\text{diam}(G), \sqrt{|E|})$ 을 사용한다(라인 3)⁸⁾. 라인 5부터 9까지는 Δ 가 공집합이 될 때까지 코드를 반복한다. 라인 6에서는 새롭게 파장을 할당하기 위해 λ 를 증가시킨다. RWAforTDP 함수는 계획 연결 요구들에 대한 적절한 경로를 설정해 현재의 파장이 할당 될 최단거리 경로에 의한 간선들의 집합 $P(G, \Delta)$ 와 파장 할당에 쓰일 계획 연결 요구들의 집합 $a(G, \Delta)$ 을 반환한다(라인

7). 이어서 $P(G, \Delta)$ 집합의 간선들에 파장 할당을 하고 라인 9에서 파장 할당에 쓰인 계획 연결 요구들을 전체 연결 요구 집합 Δ 에서 제거한다(라인 8).

그룹화 단계에서(TDP-Selector) 시간에 중복되지 않는 SLD (Scheduled Lightpath demand)들의 집합을 만든다. 먼저 최대 크기의 SLD 집합을 만든 후에 집합 Δ 에서 남아있는 SLD들로 다른 집합을 만든다. 이런 과정을 모든 SLD들을 그룹화 할 때까지 계속 반복한다. 자세한 알고리즘의 의사코드는 다음과 같다.

연결 요구 집합인 Δ 는 원소들의 종료시간(tear-down time) ω 에 대하여 오름차순으로 정렬되어 있다. Δ 는 출발 노드(s_i), 도착 노드(t_i), 시작 시간(μ), 종료 시간(ω)을 갖는 계획 연결 요구 δ_i 를 원소로 갖는다. 위 두 가지를 입력으로 다음의 결과값을 얻는다. Δ_{T_k} 는 그룹화된 계획 연결 요구들의 하위 그룹 집합이다. 그러므로 Δ_{T_1} 는 첫 번째 하위 그룹 집합이 된다. 각각의 하위 그룹 집합은 그룹화된 계획 연결 요구 $\delta_{k|\Delta_{T_k}}$ 를 원소로 갖는다. 예를 들어 $\delta_{4,3}$ 은 4번째 하위 그룹의 3번째 계획 연결 요구이다. 먼저 변수 j 를 1로 설정하면 Δ_{T_1} 이 되고 공집합이 된다(라인 2~3). 라인 4~14은 Δ 가 공집합이 될 때까지 알고리즘을 반복한다. 변수 i 를 1로 초기화 한 후 Δ 의 i 번째 원소를 δ_x 와 δ_k 로 설정한다(라인 5~7). δ_k 가 Δ 의 마지막 연결 요구와 같아질 때까지 라인 8~13을 반복한다. Δ 의 $i+1$ 번째 원소를 δ_k 라 하고 δ_x 의 종료 시간 ω_x 와 δ_k 의 시작 시간 μ_k 을 비교한다. 만약 μ_k 가 ω_x 보다 크거나 같으면 이것은 두 계획 연결 요구가 겹치지 않는다는 의미이므로 하위 그룹 Δ_{T_j} 의 원소가 된다. 그리고 시간 중복의 비교 대상을 δ_k 로 설정하여 나머지 계획 요구들의 시작 시간과 계속 비교해 나간다. 이렇게 하여 모든 계획 연결 요구들을 비교하여 그룹화 한 다음 라인 15에서 이 그룹화된 연결 요구들을 Δ 에서 제거한다. Δ 에 원소들이 남아 있다면 라인 16에서 j 를 하나 증가시켜 라인 4부터 다시 알고리즘을 시작한다.

그림 5를 통해 TDP-Selector의 동작 과정을 알아 보자. 그림 4의 표에서 계획 연결 요구들을 종료 시간에 내림차순으로 정렬을 하였다. 먼저 연결 요구 δ_1 이 첫 번째 하위 그룹의 원소가 되고 연결 요구 δ_1 의 종료시간과 δ_2 의 시작시간을 비교한다. δ_2 의 시작 시간이 δ_1 의 종료 시간보다 앞서므로 다시 δ_3 의 시작 시간과 비교한다. δ_3 의 시작시간은 δ_1 의 종료 시간보다 크거나 같으므로 δ_3 을 첫 번째 하위 그룹으로 할당하고 비교 기준 연결 요구를 δ_3 으로

```

Input:  $G(V, E)$ 
 $\Delta = \{\delta_1, \delta_2, \delta_3, \dots, \delta_n\}$ : set of SLDs, which is sorted in an increasing order of  $\omega$  of  $\delta_i$ 
 $\delta_i: [s_i, t_i, \mu_i, \omega_i]$  ( $s_i$ : source,  $t_i$ : destination,  $\mu_i$ : setup time,  $\omega_i$ : teardown time)
Output:  $\Delta_T = \{\Delta_{T_1} = \{\delta_{1,1}, \delta_{1,2}, \dots, \delta_{1,|\Delta_{T_1}|}\}, \Delta_{T_2} = \{\delta_{2,1}, \delta_{2,2}, \dots, \delta_{2,|\Delta_{T_2}|}\}, \dots, \Delta_{T_k} = \{\delta_{k,1}, \delta_{k,2}, \dots, \delta_{k,|\Delta_{T_k}|}\}$ : set of grouped sets of time-disjoint SLDs

01: TDP-Selector( $G, \Delta$ )
02:  $j = 1$ 
03:  $\Delta_{T_j} = \emptyset$ 
04: while( $\Delta \neq \emptyset$ )
05:      $i = 1$ 
06:      $\delta_x = i_{th}$  element in  $\Delta$ 
07:      $\delta_z = i_{th}$  element in  $\Delta$ 
08:      $\Delta_{T_j} = \Delta_{T_j} \cup \{\delta_x\}$ 
09:     While( $\delta_x \neq$  last element in  $\Delta$ )
10:          $i = i + 1$ 
11:          $\delta_x = i_{th}$  element in  $\Delta$ 
12:         If( $\mu_k \geq \omega_x$ )
13:              $\Delta_{T_j} = \Delta_{T_j} \cup \{\delta_x\}$ 
14:              $\delta_z = \delta_x$ 
15:      $\Delta = \Delta - \Delta_{T_j}$ 
16:      $j \leftarrow j + 1$ 
    
```

그림 4. 그룹화 알고리즘 (TDP-Selector)

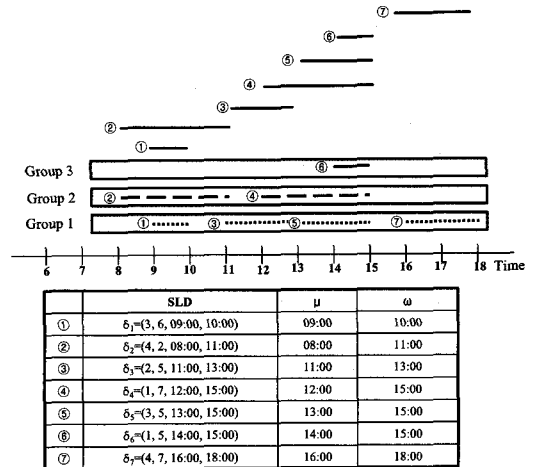


그림 5. TDP-Selector 동작 과정 예제

설정한다. 이어서 δ_5 의 종료 시간과 δ_4 의 시작 시간을 비교하면 δ_4 의 시작시간이 δ_5 의 종료 시간보다 작으므로 비교 대상은 δ_6 가 된다. δ_5 의 시작 시간은 δ_3 의 종료 시간보다 크거나 같으므로 δ_6 를 첫 번째 하위 그룹으로 할당한다. 비교 기준 연결 요구를 δ_6 로 설정하고 계속해서 δ_6 의 시작 시간과 비교를 한다. δ_6 의 시작시간은 작고 δ_7 의 시작 시간은 크거나 같으므로 δ_7 을 첫 번째 하위 그룹에 할당한다. 남아

있는 연결 요구들은 $\delta_2, \delta_4, \delta_6$ 이 되고 같은 방법으로 알고리즘을 수행하면 δ_2, δ_4 는 두 번째 하위 그룹으로 할당이 되고 δ_6 은 세 번째 하위 그룹이 된다.

RWA 단계에서는 경로를 설정하고 파장을 할당한다. 그림 6은 RWAforTDP의 의사코드이다.

```

Input : G, Δ, d
Output : α(G, Δ), P(G, Δ)
01 Algorithm RWAforTDP(G, ΔT, d)
02 α(G, Δ) = ∅, P(G, Δ) = ∅
03 For i = 1 to |ΔT|
04   For j = 1 to |Δin|
05     find shortest path Pij for δij
06     If((path length of Pij) ≤ d)
07       select path Pij for δij
08       α(G, Δ) = α(G, Δ) ∪ δij
09       P(G, Δ) = P(G, Δ) ∪ Pij
10   Delete edges of shortest paths in P(G, Δ)
       from G
    
```

그림 6. 그룹화 알고리즘 (TDP-Selector)

먼저 $\alpha(G, \Delta)$ 와 $P(G, \Delta)$ 집합을 초기화 한다(라인 2). 라인 3은 하위 그룹의 개수만큼 반복하는 for 문의 시작 부분이다. 그리고 라인 4~9에서 각각의 하위 그룹들의 원소의 개수만큼 코드를 반복한다. 계획 연결 요구의 최단 경로를 찾은 후 최단 경로 길이를 적절한 값 d 와 비교해 작거나 같은 경우 그 계획 연결 요구를 $\alpha(G, \Delta)$ 집합에 넣고 해당 최단 거리를 지나는 간선들을 $P(G, \Delta)$ 집합에 포함시킨다(라인 5~9). 마지막으로 $P(G, \Delta)$ 집합 속의 모든 간선들을 네트워크 그래프 $G(V, E)$ 에서 삭제한다(라인 10).

그림 7은 TDP-RWA 알고리즘의 수행 과정을 보여준다. (a)에서는 첫 번째 하위 그룹의 연결 요구인 $\delta_1, \delta_3, \delta_5, \delta_7$ 의 경로 설정 및 그 경로의 간선들을 첫 번째 파장 ($\lambda=1$)으로 할당 과정을 보여준다. 이 하위 그룹들은 모두 시간적으로 겹치지 않기 때문에 경로에 상관 없이 같은 파장으로 할당될 수 있다. (b)에서 먼저 (a)과정에서 파장 할당된 간선들을 그래프에서 삭제한다. 두 번째 하위 그룹에서 첫 번째 파장으로는 연결 요구 δ_8 만이 할당되게 된다. (c)의 과정에서는 (b)에서 파장이 할당된 간선들을 삭제한 후 세 번째 그룹의 연결 요구들에 대해서 경로 설정 및 그 경로의 간선들을 첫 번째 파장으로 할당한다. 모든 하위 그룹의 연결 요구들에 대해서 알고리즘이 수행된 후 다시 파장을 하나 증가 시켜 할당 되지 않은 연결 요구들이 있

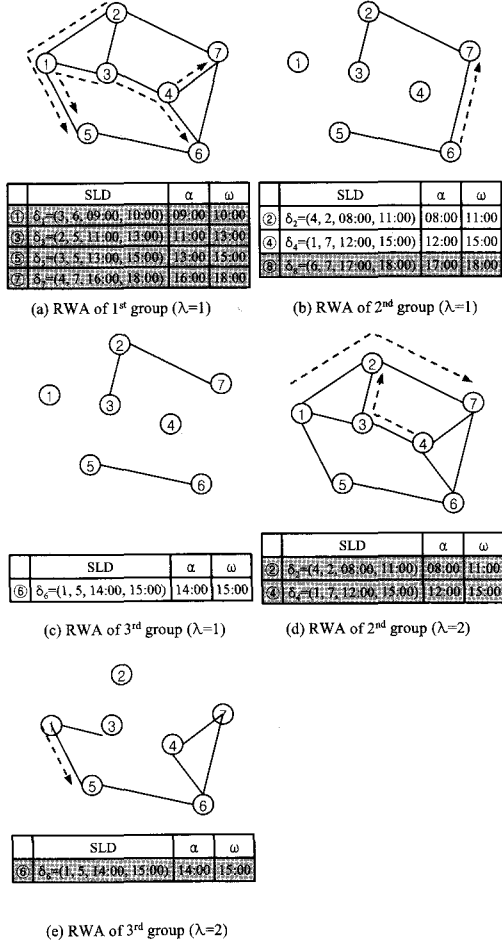


그림 7. TDP-Selector 동작 과정 예제

는 그룹에서부터 알고리즘을 수행한다. (d)에서는 두 번째 그룹의 연결 요구인 δ_2, δ_4 에 대해서 경로 설정 후 그 경로의 간선들을 두 번째 파장 ($\lambda=2$)으로 할당한다. (e)에서는 (d)의 과정에서 파장 할당된 간선들을 삭제한 후 연결 요구 δ_6 를 두 번째 파장으로 할당한다. 그러므로 알고리즘 수행 후 사용된 파장은 2개가 된다.

IV. 성능 평가

본 장에서는 제안하는 TDP-RWA 알고리즘과 sRWA[3], 조합 최적화 알고리즘(Combinatorial Optimal Solution : COS)^[3]의 성능을 비교 평가한다. 이때 비교되는 척도는 사용되는 파장의 수와 수행시간으로 한다. 성능 평가를 위해 랜덤 네트워크를 사용하였다. 먼저 주어진 노드의 수($|V|$)와 각 노드 쌍들 간에 간선(edge)의 존재 확률(p_e)에 의해서 연결

된 랜덤 그래프를 생성한다. 그리고 임의의 두 노드 사이에 연결 요구(demand)가 존재할 확률 p_l 로부터 연결 요구를 생성하고 동일한 소스(source) 노드와 목적(destination) 노드 사이의 연결 요구는 복수개(N_c)가 될 수 있다. $T_{service}$ 는 평균 서비스 시간으로 각 계획 연결 요구들의 서비스 시간을 평균한 값이다. 이 값이 작을수록 계획 연결 요구들 간에 시간 중복 가능성이 줄어든다. 본 논문에서는 p_e , p_l , $T_{service}$ 의 값을 변화시키면서 $N_c=3$, $N_c=5$ 인 두 경우에 대해서 각각 1000번의 시뮬레이션을 수행하여 평균적으로 사용되는 파장의 수와 수행 시간을 구하였다. 성능 평가에 사용되었던 시뮬레이션은 C언어로 작성되었으며 AMD 애슬론 1800+ CPU와 512MB 메인메모리 그리고 Windows XP 운영체제에서 시행하였다.

그림 8은 20개의 노드를 가진 랜덤 네트워크에서 $p_e = 0.4$, $T_{service} = 4$ 로 고정시킨 후 p_l 의 증가에 따른 파장수의 변화를 나타낸 그래프이다. p_l 이 증가함에 따라 사용되는 파장의 수가 증가하였다. 그 이유는 연결 요구들이 증가함에 있다. 계획 연결 요구들의 시간 중복성을 이용하지 못하는 BGAforEDP 알고리즘은 p_l 이 커질 수록 파장수가 더욱 증가하며 특히 중복 연결 요구가 많을 경우($N_c=5$) 그 현상이 더 크다. TDP-RWA 알고리즘은 BGAforEDP에 비해 25%의 성능 향상을 보였다. sRWA와는 비슷한 성능을 보이며 조합 최적화 방법과는 약간의 성능 하락을 보였다.

그림 9는 18개의 노드를 가진 랜덤 네트워크에서 $p_l = 0.3$, $T_{service} = 4$ 로 고정시킨 후 p_e 를 증가시키면서 파장수의 변화를 나타낸 그래프이다. p_e 가 증가함에 따라 모든 그래프는 감소하는 경향을 보인다. 이것은 연결 요구의 수는 고정이지만 네트워크의 크기가 커지므로 연결 간선수가 증가하여 파장의 수가 줄어들게 되기 때문이다. 역시 마찬가지로 시간의 중복성을 이용하지 못하는 BGAforEDP 알고리즘은 다른 계획 연결 요구에 적합한 알고리즘과 커다란 격차를 보였으며 본 논문에서 제안한 TDP-RWA 알고리즘의 성능은 sRWA보다 2.1% 낮다. 그리고 조합 최적화 알고리즘은 TDP-RWA, sRWA와 비교해 9.1%~9.3% 성능 향상을 보였다.

그림 10은 20개의 노드를 가진 랜덤 네트워크에서 $p_l = 0.3$, $p_e = 0.3$ 으로 고정시킨 후 $T_{service}$ 를 증가시키면서 파장수의 변화를 나타낸 그래프이다. 평균 서비스 시간($T_{service}$)은 계획 연결 요구들의 시간의 중복성에 영향을 미친다. 평균 서비스 시간이 작다

면 그만큼 연결 요구들 간에 서비스 시간이 중복되지 않을 확률이 높기 때문에 사용되는 파장의 수가 줄어들 수 있다. 반면에 평균 서비스 시간이 크다면 연결 요구들 간에 서비스 시간이 중복될 확률이 높기 때문에 사용 파장 수가 늘어난다. 본 논문에서

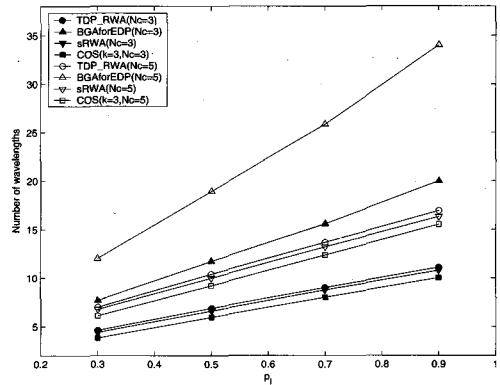


그림 8. p_l 의 증가에 따른 파장 수 변화($p_e=0.4$, $N_c=3$ or 5, $T_{service}=4$)

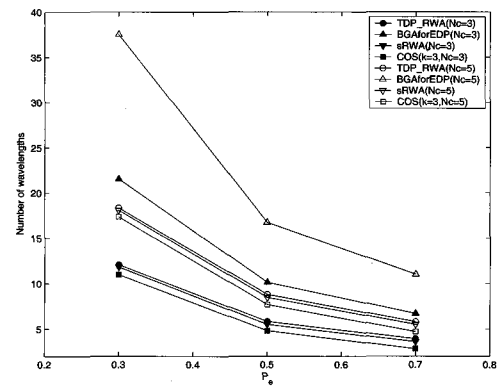


그림 9. p_e 의 증가에 따른 파장 수 변화($p_l=0.3$, $N_c=3$ or 5, $T_{service}=4$)

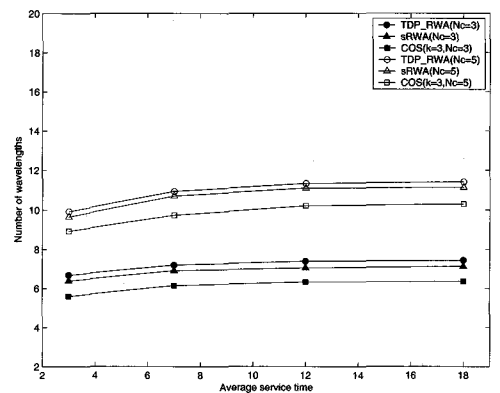


그림 10. $T_{service}$ 의 증가에 따른 파장 수 변화($p_l=0.3$, $p_e=0.3$, $N_c=3$ or 5)

제한한 TDP-RWA 알고리즘은 sRWA 알고리즘은 비교하여 2.98%의 성능 하락 보였으며 조합 최적화 알고리즘(COS)은 다른 휴리스틱 알고리즘에 비해 9.8% 이상의 성능 향상을 보였다.

그림 11은 20개의 노드를 가진 랜덤 네트워크에서 $p_e = 0.4$, $T_{service} = 4$ 로 고정시킨 후 p_l 을 증가시키면서 평균 수행 시간의 변화를 나타낸 그래프이다. 조합 최적화 알고리즘은 2000초 이상의 수행 시간이 걸린다. 반면에 BGAforEDP, TDP-RWA, sRWA 등의 휴리스틱 알고리즘은 5초~37초의 수행 시간을 나타내었다. 그림 11에서 가장 좋은 성능을 보이는 것은 BGAforEDP 알고리즘이다. 왜냐하면 연결 요구의 시간적 중복현상을 이용하지 않기 때문이다. 제한한 TDP-RWA는 sRWA와 비교하여 54%까지 성능 향상을 보여준다. 그 이유는 sRWA 알고리즘은 각각의 연결 요구의 경로에 파장을 할당 할 때마다 간선(edge) 비교를 하기 때문이다.

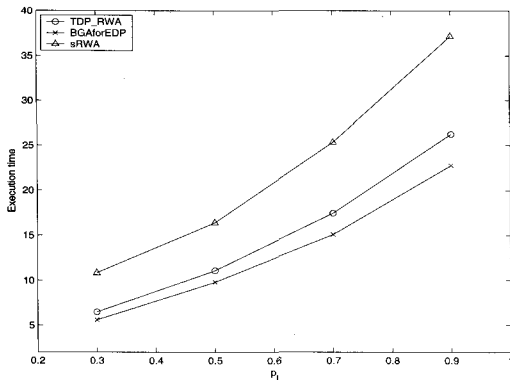


그림 11. p_l 의 증가에 따른 평균 수행 시간의 변화($p_e=0.4$, $N_c=3$, $T_{service}=4$)

V. 결론

본 논문에서는 광전달망에서 계획 연결 요구를 가지는 경우에 경로 설정과 파장 할당을 위한 효과적인 휴리스틱 알고리즘을 제안하였다. 기존의 조합 최적화 알고리즘은 시간 비용이 크고 FF(First Fit) 알고리즘에 기반한 sRWA 알고리즘은 간선(edge)을 비교함에 있어서 시간의 추가 비용과 메모리의 오버헤드(overhead)가 크다. 본 논문에서 제안한 휴리스틱 알고리즘 TDP-RWA는 그룹화를 통해 시간적 비공유 경로를 이용하면서 빠르게 경로 설정과 파장 할당을 하는 알고리즘이다. 랜덤 네트워크를 이용하여 성능 평가를 하였으며, 제안한 알고리즘은 추가적인 메모리 오버헤드(overhead)가 없으며 사용

되는 파장수의 성능 면에서는 sRWA와 비슷하다. 또한 최적화 알고리즘의 사용 파장수에 근접하면서 빠른 수행 시간을 보였다.

참고 문헌

- [1] *Architecture of Optical Transport Network*, ITU-T Recommendation G. 872, Nov. 2001.
- [2] *Advanced Networking for Research and Education*, Online, Available: <http://abilene.internet2.edu/>
- [3] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro and R. Douville, "Routing and Wavelength Assignment of Scheduled Lightpath Demands," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 8, pp. 1231-1240, Oct. 2003.
- [4] 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.
- [5] H. Zang, J. P. Jue, and B. Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-routed Optical WDM networks," *SPIE/Baltzer Optical Networks Magazine*, vol. 1, no. 1, pp. 47-60, Jan. 2000.
- [6] 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.
- [7] D. Kirovski and M. Potkonjak, "Efficient Coloring of a Large Spectrum of Graphs," in *Proc. 35th Conf. Design Automation*, pp. 427-432, June 1998.
- [8] J.M. Kleinberg. "Approximation Algorithms for Disjoint Paths Problems," Ph. D. dissertation, MIT, May 1996.
- [9] J. Clausen, "Branch and Bound Algorithms-Principles and Examples," Online, Available: <http://www.imm.dtu.dk/~jha/>.
- [10] J. Clausen and M. Perregaard, "On the Best Search Strategy in Parallel Branch-and-

bound-Best-first-search vs. Lazy Depth-first-search," *Annals of Operations Research*, no. 90, pp. 1-17, 1999.

[11] F. Glover and M. Laguna, *Tabu Search*, MA: Kluwer-Academic, 1997.

[12] J. Kuri, N. Puech, M. Gagnaire, and E. Dotaro, "Routing Foreseeable Lightpath Demands Using a Tabu Search Meta-heuristic," in *Proc. GLOBECOM 2002*, pp. 2803-2807, Nov. 2002.

안 현 기 (Hyunki Ahn)

준회원



2004년 2월 성균관대학교 정보통신공학부 졸업

2004년 3월~현재 성균관대학교 정보통신공학부 석사과정

<관심분야> 무선 LAN/PAN, 광 네트워크 라우팅

이 태 진 (Tae-Jin Lee)

정회원



1989년 2월 연세대학교 전자공학과 졸업

1991년 2월 연세대학교 전자공학과(석사)

1995년 12월 University of Michigan, Ann Arbor(석사)

1999년 5월 University of Texas

at Austin(박사)

1999년 8월~2001년 2월 삼성전자 중앙연구소 책임 연구원

2001년 3월~현재 성균관대학교 정보통신공학부 조교수
<관심분야> 통신 네트워크 성능 분석 및 설계, 무선 LAN/PAN, Ad-hoc/센서 네트워크, 광 네트워크

정 민 영 (Min Young Chung)

정회원



1990년 2월 KAIST 전자공학과 졸업

1994년 2월 KAIST 전자공학과(석사)

1999년 12월 KAIST 전자공학과(박사)

2000년 1월~2002년 2월 ETRI

선임 연구원

2002년 3월~현재 성균관대학교 정보통신공학부 조교수
<관심분야> 유무선 홈 네트워크, 이동통신 네트워크, IP 라우터 시스템, 광 네트워크 관련 성능분석

추 현 승 (Hyunseung Choo)

정회원



1988년 2월 성균관대학교 수학과 졸업

1990년 12월 University of Texas at Dallas, 컴퓨터공학과(석사)

1996년 12월 University of Texas at Arlington, 컴퓨터공학과

(박사)

1997년 8월~1998년 3월 특허청 심사관(사무관)

1998년 3월~현재 성균관대학교 정보통신공학부 부교수

<관심분야> 광 네트워크, 이동컴퓨팅, 라우팅 프로토콜, 그리드 컴퓨팅