

최대 EDP를 이용한 경로설정 및 경로 충돌 그래프를 이용한 파장할당 문제 해결 방안

준회원 김 덕 현*, 정회원 정 민 영*, 이 태 진**, 추 현 승***

Routing with Maximum Edge Disjoint Paths and Wavelength Assignment with Path Conflict Graph

Duk Hun Kim* *Associate Member*, Min Young Chung*,
Tae-Jin Lee**, Hyunseung Choo*** *Regular Members*

요 약

경로설정 및 파장할당(RWA: Routing and Wavelength Assignment) 문제는 파장 분할 다중화 기술에 기반한 차세대 광 인터넷 백본망에서 주목되는 기술중의 하나이다. 이러한 광 인터넷 망에서 연결요구에 대해 최적의 경로를 선택하여 선택된 경로에 효율적으로 파장을 할당하는 RWA문제는 매우 중요하다. 본 논문에서는 경로 충돌 그래프 개념과 송신노드 - 수신노드 간의 MAX_EDP(Maximum Edge Disjoint Paths: 최대 링크 비 중복 경로) 알고리즘을 이용한 효율적인 알고리즘을 제안한다. 또한, 이에 대한 모의실험을 통해 제안한 방법과 기존의 BGAforEDP 알고리즘과의 성능분석을 통해 효율성을 검증한다. 모의 실험결과 제안된 방식이 기존의 방식보다 고정된 네트워크 토폴로지(NSFNET)에서는 최대 20%, 랜덤 토폴로지에서는 최대 32% 정도의 성능향상을 보인다.

Key Words : Routing and wavelength assignment(RWA), Edge disjoint paths(EDP), PCG(Path conflict graph)

ABSTRACT

Routing and wavelength assignment problem is one of the most important issues in optical transport networks based on wavelength division multiplexing(WDM) technique. In this paper, we propose a novel approach using path conflict graphs and an algorithm for finding all edge disjoint paths. And then we compare the performance of the proposed algorithm with that of bounded greedy approach for EDP(BGAforEDP). The proposed one outperforms up to about 20% in the fixed traditional topology(NSFNET) and about 32% in random topologies over the BGA for EDP algorithm.

I. 서 론

오늘날의 인터넷 네트워크는 사용자의 급속한 증가에 따른 대용량 대역폭과 빠른 전송속도를 요구

한다. 기하급수적으로 증가하는 인터넷 트래픽을 수용하기 위하여 광 인터넷은 필수적이라 할 수 있다. 이렇게 광 네트워크에 대한 요구가 증가함에 따라 이에 대한 연구 또한 활발히 진행되고 있다. 광 네

* 성균관대학교 정보통신공학부 네트워크 기술 연구실({thekern, mychung}@ece.skku.ac.kr)

** 성균관대학교 정보통신공학부 네트워크 시스템 연구실(tjlee@ece.skku.ac.kr)

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

논문번호 : KICS2004-10-221, 접수일자 : 2004년 10월 8일

※ 이 논문은 BK21과 정보통신부 지원에 의하여 연구되었음.

네트워크에 관련된 여러 가지의 연구들 중 경로설정 및 파장할당 문제로 알려진 RWA(Routing and Wavelength Assignment) 문제는 다양한 관점에서 연구가 진행 중이다¹⁾.

RWA문제는 광 네트워크에서 적절한 경로를 찾고 찾은 경로에 적절한 파장을 할당하는 것으로써 지금까지 NP-complete 문제로 남아있다²⁾. RWA문제를 선형 정수 프로그래밍을 통해 최적으로 해결하기 위한 연구가 있었다^{3),4)}. 그러나 이는 많은 수행시간을 요하기 때문에 이에 대한 대안으로 여러 가지의 휴리스틱(heuristic) 해결방안들이 제안되어 왔다^{6)~11)}. 일반적으로 RWA 알고리즘은 주어진 네트워크의 환경에 큰 영향을 받는다. 따라서, 특정한 네트워크 환경을 가정한 후 연구하는 것이 일반적이다⁵⁾. RWA 알고리즘에 영향을 주는 네트워크 환경으로는 연결요구의 정적/동적 방식, 파장변환기의 유/무, 그리고 다중연결요구의 허용 여부 등이 있다. 우선 연결요구방식에 대해 살펴보면, 네트워크의 연결요구가 미리 정해져 있는지 동적으로 주어지는지에 따라 정/동적인 연결요구로 분류된다. 즉, 연결요구가 한번 주어지면 긴 시간 동안 새로운 연결요구가 들어오지 않는 상황을 정적인 연결요구조건이라 하며, 랜덤으로 연결요구가 생성되는 상황을 동적인 연결요구조건이라 한다. 파장변환기의 사용유무에 따라서도 광 네트워크의 환경은 많은 차이점을 보인다. 파장변환기가 사용되지 않는 상황에서 광 네트워크는 파장연속성이라는 제약을 가진다. 이는 송신 노드에서 수신 노드까지 동일한 파장을 사용해야만 하는 제약이다. 이러한 파장연속성 제약은 파장변환기를 사용함으로써 제거될 수 있다. 그러나 현재까지 파장변환기는 고가이기 때문에 파장변환기의 사용유무에 따라 네트워크비용에 큰 영향을 미친다. 마지막으로, 다중 연결요구에 대한 허용 여부도 고려되어야 한다. 이는 동일한 송/수신 노드간의 연결요구를 복수로 하는 것을 허용하느냐 허용하지 않느냐의 문제이다. 이러한 여러 가지의 네트워크의 환경들에 따라 지금까지 여러 연구가 진행되어 왔다. 주어진 정적 연결요구들에 대해 필요한 파장의 수를 최대한 줄일 수 있는 방법^{6),7)}, 고정된 파장의 수와 동적인 연결요구 상황 하에서 충돌 확률을 낮추는 방법^{8),9)}, 네트워크 비용을 최대한 줄이는 방법^{10),11)} 등이 연구되었다.

본 논문에서는 RWA문제해결 방식을 개선하는 새로운 휴리스틱 알고리즘을 제안한다. 네트워크 환경은 정적인 연결요구, 파장변환기 미사용, 동일한

연결요구에 대한 중첩 허용을 가정하고, 주어진 연결요구에 대해 필요한 파장 수를 최대한 줄이는 적용성의 방식을 제안한다. 적용성의 방식이란, 주어진 연결요구에 대한 경로를 찾고 이에 대한 파장을 할당하지 못하였을 경우 다른 가능한 경로를 찾는 방식으로 주어진 경로에 대해 파장할당을 못했을 경우 연결요구를 파기하는 방식과는 차이점을 보인다. 이러한 네트워크 환경은 계산의 편의성, 경로 관리비용, 현실적인 상황을 고려하여 가정하였으며, 이는 제안하는 방식의 비교대상이 되는 제한된 탐욕적인 접근법(BGAforEDP: Bounded Greedy Approach for Edge Disjoint Paths) 알고리즘과 유사한 환경이다. 제안하는 RWA 문제해결방식은 경로설정 부분은 MAX_EDP 알고리즘¹²⁾을 이용하였으며, 파장할당 부분은 경로 충돌 그래프¹⁴⁾를 이용하였다. MAX_EDP 알고리즘을 이용하여 가능한 모든 비 중복 경로를 찾은 후, 경로 충돌 그래프를 이용하여 다른 후보경로들과 가장 적게 중복이 일어나는 경로를 선택하여 파장을 할당하였다. 논문의 최종 목적은 주어진 연결요구집합에 사용되는 파장의 수를 최소화하는 것이다.

논문의 구성은 기존의 BGAforEDP 알고리즘, 최대 EDP를 찾는 MAX_EDP 알고리즘, 경로 중첩 그래프 개념을 2장에서 설명하고, 3장에서 제안하는 알고리즘을 설명한 후, 4장에서 모의 실험 및 결과를 분석한다. 마지막으로 5장에서 결론을 맺는다.

II. 관련 연구

2.1 BGAforEDP 알고리즘

RWA 문제에 대한 해법으로 최단 경로 알고리즘에 기반한 간단한 비 중복 링크 경로 방식이 소개되었다⁶⁾. 이 방식은 주어진 연결요구들 중 임의로 하나의 연결요구를 선택하여 Dijkstra 알고리즘으로 가능한 최단 경로를 찾은 후 네트워크 정보를 통해 경로가 유효한지를 판단하고, 가능하면 파장을 할당하는 적용성 방식이다. 네트워크 환경은 정적인 연결요구, 파장변환기 미사용을 가정하였다. 본 논문에서 제안하는 방법과 같이 입력되는 연결요구들에 필요한 최소의 파장 수를 구하는 것이 그 목적이다.

주어진 네트워크 $G = (V, E)$ 와 (여기서 V 는 정점(node), E 는 에지(edge)를 나타냄) 연결요구 집합 $D = \{(s_1, t_1), \dots, (s_k, t_k)\}$ 를 고려한다. 알고리즘의 수행 순서는 다음과 같다. 먼저, 연결요구집합 D 에서 임의의 한 원소 D_i 를 선택한 후 이에 대한 최단 경

로 P_i 를 구한다(D_i 에 대한 경로를 구할 수 없을 때는 다음 연결요구에 대한 최단경로를 구한다). 이때 구한 P_i 의 길이가 상한 값 $d(\max[\sqrt{|E|}, \text{diam}(G)])$, 이는 효율적인 경로선택을 위해 너무 긴 경로를 미리 차단하기 위한 목적으로 사용됨)보다 작다면, P_i 를 설정 경로 집합 P 에 포함시키고, D_i 를 $\alpha(G, D_{\text{routed}})$ 에 포함시킨 후, 토폴로지 G 상에서 P_i 의 경로에 포함된 링크를 제거한다. 만약, P_i 의 길이가 d 보다 크다면, 연결요구 D_i 는 할당되지 못하고 다음 연결요구에 대해 위의 과정을 반복한다. 이러한 반복과정은 D 집합내의 모든 원소에 대해 수행되며, 이때의 $\alpha(G, D_{\text{routed}})$ 는 동일한 하나의 파장으로 할당 가능한 모든 연결요구집합이 된다. 현재의 파장으로 더 이상 요구를 수용할 수 없으면, D 에서 이미 할당된 $\alpha(G, D_{\text{routed}})$ 의 요구들을 제외한 후 다른 파장에 대해 처음의 과정을 반복한다. 이는 D 집합이 공집합이 될 때까지 반복되며, 최종 결과값은 모든 연결요구를 충족시키는데 필요한 파장의 수가 된다. 이 방법은 알고리즘의 간단함에 비해 비교적 좋은 성능을 보임이 입증되었다. 그러나 연결요구를 임의로 선택하기 때문에 성능결과가 일정하지 않다.

2.2 MAX_EDP 알고리즘

각 연결요구에 대한 최대한의 EDP를 찾는 방법이 제안되었다^[12]. 기본개념은 비 중복 경로집합 선택 프로토콜(DPSP: Disjoint Path-set Selection Protocol)^[13]에 기반한다. 이는 하나의 경로가 선택된 상황에서 또 다른 경로를 선택했을 때 만약 후자의 경로가 전자의 경로에 포함된 링크 중 역방향 링크를 통과하면 이를 제거함으로써 두 개의 비 중복 경로가 생성된다는 개념에 기반한다. 여기서 정 방향 링크란 찾아진 경로에 포함되면서 송신 노드에서 수신 노드로 향하는 방향의 링크를 말하며 역방향 링크란 동일한 조건에 대해 정 방향 링크와 반대되는 방향을 가진 링크를 말한다. 이때 사용되는 네트워크 토폴로지는 방향성으로써 기존의 무 방향성 토폴로지가 있을 때 각 링크를 서로 반대되는 두 개의 방향성 링크로 바꾸어 생성한다.

입력 토폴로지를 $G(V, E)$, T (임의의 i 에 대해 $T \ni \tau_i$)를 EDP의 집합이라고 할 때 송신 노드 V_i 에서 수신 노드 V_j 까지의 최대 EDP를 찾는 알고리즘의 수행 순서는 다음과 같다. 우선 무 방향성 토폴로지 $G(V, E)$ 를 방향성 토폴로지 $G'(V, E')$ 로 변환한다. 이를 토대로 V_i 에서 V_j 까지의 최단경로를 Dijkstra 알고리즘을 이용하여 찾는다. 이를 τ_i 이라 하자. τ_i

는 T 에 저장하고 이 경로에 포함된 링크는 토폴로지에서 삭제한다. 송신 노드 V_i 에 연결된 다른 링크를 대상으로 다시 Dijkstra 알고리즘을 수행한다. 만약 이때 찾은 경로(τ_j)가 이미 찾아놓은 경로 τ_k 에 포함되는 링크들의 역방향 링크를 경유하면 해당 링크를 삭제하여 독립적인 두 개의 비 중복 경로(τ_j, τ_j'')를 생성한다. τ_j' 와 τ_j'' 를 τ_k, τ_j 로 갱신한 후 T 에 저장한다. 만약 찾은 경로 τ_j 가 미리 찾아놓은 경로의 역방향 링크를 통과하지 않는다면 별도의 변환 없이 τ_j 를 EDP 집합 T 에 저장한다. 이러한 과정을 송신 노드 V_i 에 연결된 모든 링크를 대상으로 반복한다. 이에 대한 최종결과는 집합 T 이다. 이때 집합 T 의 기수(Cardinal Number)의 최대 값은 $|T| = \min(\text{Degree}_{\text{source}}, \text{Degree}_{\text{destination}})$ 이며, 이때 $\text{Degree}_{\text{source}}, \text{Degree}_{\text{destination}}$ 는 각각 송수신 노드에 연결된 링크의 수이다. 이는 EDP의 특성에 기인한다. 위의 방법으로 찾아진 EDP들은 각 연결요구에 대해 가능한 후보경로가 되며, 본 논문에서는 RWA 문제의 경로 설정을 해결하기 위하여 이 방법을 사용한다.

2.3 경로 충돌 그래프(Path Conflict Graph)

파장 분할 다중화 기술에 기반한 광 네트워크 상의 RWA 문제는 그래프 컬러링 문제로 변환되어 해결될 수 있다. 이는 파장 연속성 제약이 있는 상황에 적용되는 것으로, 이미 많은 연구가 진행된 그래프 컬러링 문제를 RWA 문제에 확장시킨 것이다. 논문^[14]에서 이러한 그래프 컬러링 문제를 해결하기 위한 방법으로 경로 충돌 그래프를 사용했다. 경로 충돌 그래프는 $G_p = (V_p, E_p)$ 와 같이 표현되며 여기서 V_p 는 임의의 연결요구 (s_i, t_i) 에 대한 가능한 후보경로를 나타내며, E_p 는 이들 V_p 간의 네트워크 상에서의 링크 중첩여부를 나타낸다. 즉, 임의의 두 경로 $V_{p,i}$ 와 $V_{p,j}$ 가 네트워크 상에서 하나의 링크이라도 중첩된다면, 경로 충돌 그래프 상에서 두 경로 노드 $V_{p,i}, V_{p,j}$ 사이에는 링크가 생성된다는 것을 말한다. 일반적인 네트워크 상황 하에서 연결요구에 대한 가능한 경로가 하나 이상 존재하므로, 이러한 경로들 중 어느 것을 선택할 것인지에 대한 문제를 위의 경로 충돌 그래프를 이용하여 효율적으로 해결할 수 있다. 본 논문에서는 이러한 경로 충돌 그래프를 사용하여 최소한의 경로 충돌이 예상되는 경로를 선택하여 파장을 할당함으로써 BGAforEDP 알고리즘을 보완할 수 있다.

III. 제안된 RWA 알고리즘

BGAforEDP 알고리즘에서는 입력된 연결요구집합에 경로설정 및 파장할당을 위한 연결요구를 임의로 선택하므로, 수행시간이 빠른 반면 성능 면에서 일정한 결과를 나타내지 못하였다. 이는 하나의 잘못된 경로할당에 의해 전체적인 네트워크의 성능을 악화시킬 수 있는 가능성이 잠재하기 때문이다. 본 논문에서 제안하는 방식은 경로설정 및 파장할당을 위하여 각 연결요구에 대해 가능한 모든 EDP를 찾은 후 이들의 경로 길이와 중복 정도를 고려하였다. 중복 정도는 경로 충돌 그래프를 이용하여 구하였으며, 이는 경로 충돌 그래프 상에서 노드에 연결된 링크의 수로 나타난다. 제안하는 알고리즘에 대한 순서는 아래의 STEP 1~STEP 5와 같다. 알고리즘의 입력은 네트워크 토폴로지 그래프 G 와 연결요구 집합 $D = \{(s_i, t_i), \dots, (s_k, t_k)\}$: s 는 송신 노드, t 는 수신 노드이며, 최종 결과 값은 사용된 파장의 수 λ 이다. 그림 1은 제안하는 알고리즘에 대한 의사코드이다. EDP 들의 집합은 $T = \{\tau_{1,1}, \tau_{1,2}, \dots, \tau_{1,n_1}, \dots, \tau_{k,1}, \tau_{k,2}, \dots, \tau_{k,n_k}\}$ 와 같이 표현된다. 여기서, k 는 입력된 연결요구의 수를 의미하며, n_i ($i:1,2,3,\dots,k$)는 i 번째 연결요구에 대한 최대 EDP의 수이다. m_i 은 집합 T 의 EDP들이 가지는 경로길이들 중 가장 작은 값을 나타내고, m_d 는 경로길이 m_i 을 가지는 EDP들의 중복 정도 중 가장 작은 값을 나타낸다. 또한, $T_{\text{same_partition}}$ 는 T 집합의 EDP $\tau_{i,j}$ 에 대해 같은 분할에 속한 EDP들을 나타내며 (즉, 임의의 i 에 대해 $T_{\text{same_partition}} = \{\tau_{i,j} \mid j = 1,2,\dots,n_i\}$), $T_{\text{connected}}$ 는 선택된 EDP와 경로 충돌 그래프 상에서 링크로 연결된 EDP들의 집합을 나타낸다.

STEP 1: 연결요구집합 D 에 있는 모든 연결요구 각각에 대한 최대 EDP를 찾은 후^[12] EDP들의 집합 T 를 생성한다. 각 EDP는 물리경로, 경로길이, 분할, 충돌 정도의 4가지 정보로 표현되며, 물리경로는 경로 상에 포함된 각 노드의 순서를, 경로길이는 전체 경로의 홉 수를, 분할은 각 EDP가 어느 연결요구에 대한 것인지를 나타내며, 충돌 정도는 경로 충돌 그래프상에서의 중복 정도를 나타낸다(그림 1: 03).

STEP 2: 파장 수 λ 를 하나 증가 시킨 후 다음 단계로 넘어간다(그림 1: 06).

STEP 3: T 에 대한 경로 충돌 그래프 G_p 를 만들고, m_i 과 m_d 를 결정한다(그림 1: 07, 09).

STEP 4: T 집합 내의 EDP중 경로길이가 m_i 이며, 중복 정도가 m_d 인 EDP $\tau_{i,j}$ 를 선택한다. 만약, 해당 조건을 가지는 EDP가 하나 이상일 때는 랜덤으로 하나를 선택한다. $\tau_{i,j}$ 가 선택되면 이에 대한 $T_{\text{same_partition}}$ 집합과 $T_{\text{connected}}$ 집합이 결정된다. 다음으로, 선택된 EDP $\tau_{i,j}$ 에 대한 연결요구 (s_i, t_i) 이 연결요구 집합 D 에서 삭제되고, EDP 집합 T 에서 앞에서 구한 $T_{\text{same_partition}}$ 집합과 $T_{\text{connected}}$ 집합의 원소들을 제거한다. 이러한 STEP 3, STEP 4의 과정은 T 집합이 공집합이 될 때까지 반복된다(그림 1: 10~16).

STEP 5: 연결요구집합 D 가 공집합인지 아닌지를 확인한다(그림 1: 05). 만약, D 가 공집합이라면, 알고리즘은 종료되고 최종 값 λ 를 알고리즘 출력으로 내보낸다(그림 1: 19). 공집합이 아니면, **STEP 2**로 복귀한다(그림 1: 06).

```

01: INPUT-Network topology  $G$  and a demand set  $D$ 
02: OUTPUT-Total number of used wavelengths for  $D$ 
03: Find all possible EDPs for each demand in  $D$ 
    and make EDPs' set  $T$ 
04:  $\lambda = 0$ 
05: WHILE ( $D$  is not empty) DO
06:  $\lambda = \lambda + 1$ 
07: Draw a  $G_p$  for EDPs in  $T$ 
08: WHILE ( $T$  is not empty) DO
09: Determine the smallest value  $m_i$  and  $m_d$ 
10: IF ( $\tau_{i,j}$ 's length is  $m_i$  &&
    conflict degree is  $m_d$ ) THEN
11: Make sets  $T_{\text{same\_partition}}$  and  $T_{\text{connected}}$  for  $\tau_{i,j}$ 
12:  $D = D - (s_i, t_i)$ 
13:  $T = T - (T_{\text{same\_partition}} \cup T_{\text{connected}})$ 
14: ENDIF
15: Update a  $G_p$  for changed  $T$ 
16: ENDWHILE
17: Refresh a new  $T$  for remained demands in  $D$ 
18: ENDWHILE
19: Return  $\lambda$ 
    
```

그림 1. 제안된 알고리즘의 의사코드.

그림 2와 같은 7개의 노드와 9개의 링크를 가지는 네트워크에서 연결요구집합 $D = \{(A,B), (B,F), (D,E), (D,G), (F,G)\}$ 라고 가정하였을 때, 제안된 알고리즘의 동작절차는 다음과 같다.

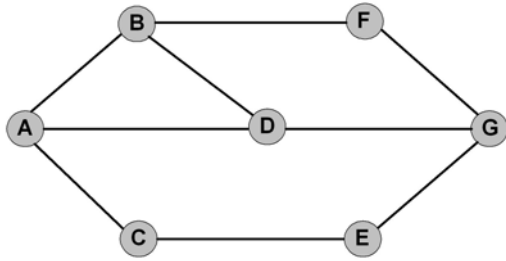


그림 2. 노드7개와 링크 9개로 구성된 간단한 망 형 네트워크

연결요구집합 D 에 대하여 가능한 EDP들을 모두 구한다. 이 경우 총 12개의 EDP $\tau_{1,1}$, $\tau_{1,2}$, $\tau_{1,3}$, $\tau_{2,1}$, $\tau_{2,2}$, $\tau_{3,1}$, $\tau_{3,2}$, $\tau_{4,1}$, $\tau_{4,2}$, $\tau_{4,3}$, $\tau_{5,1}$, $\tau_{5,2}$ 는 각각 연결요구(A,B)에 대한 3개의 EDP [A-B], [A-D-B], [A-C-E-G-F-B], 연결요구 (B, F)에 대한 2개의 EDP [B-F], [B-D-G-F], 연결요구 (D,E)에 대한 2개의 EDP [D-G-E], [D-A-C-E], 연결요구 (D,G)에 대한 3개의 EDP [D-G], [D-B-F-G], [D-A-C-E-G]와 연결요구 (F,G)에 대한 2개의 EDP [F-G], [F-B-D-G]이다. 12개의 EDP들 각각을 노드로 정의하여 경로충돌 그래프를 그리면 그림 3과 같다. 이들 12개의 EDP들 중 가장 짧은 경로 길이($m_f=1$)를 가지는 4개의 EDP $\tau_{1,1}$, $\tau_{2,1}$, $\tau_{4,1}$, $\tau_{5,1}$ 들을 구하고 이 4개의 EDP 중 가장 적은 중복 정도($m_d=0$)를 가지는 $\tau_{1,1}$ 을 선택한다. 선택된 $\tau_{1,1}$ 에 대한 연결요구 (A,B)가 D 에서 삭제되고, 이와 같은 파티션에 속한 EDP ($T_{\text{same_partition}} = \{\tau_{1,1}, \tau_{1,2}, \tau_{1,3}\}$)들이 EDP 집합 T 에서 삭제된다. $T_{\text{connected}}$ 는 공집합이므로 삭제될 대상이 없다. 다음으로 남겨진 T 가 공집합이 아니므로 남겨진 $T = \{\tau_{2,1}, \tau_{2,2}, \tau_{3,1}, \tau_{3,2}, \tau_{4,1}, \tau_{4,2}, \tau_{4,3}, \tau_{5,1}, \tau_{5,2}\}$ 에 대한 경로 충돌 그래프를 그린다(그림 4의 (a)). 집합 T 내의 EDP들 중 가장 짧은 경로 길이($m_f=1$)를 가지는 3개의 EDP $\tau_{2,1}$, $\tau_{4,1}$, $\tau_{5,1}$ 들이 선택되고, 이 3개의 EDP 중 $m_d(=2)$ 를 가지는 2개의 EDP $\tau_{2,1}$, $\tau_{5,1}$ 가 선택된다. 이 경우 랜덤하게 하나를 선택해야 하는데 여기서는 임의로 $\tau_{2,1}$ 를 선택한다. 선택된 EDP $\tau_{2,1}$ 에 대한 연결요구 (B,F)를 D 에서 삭제하고, $T_{\text{same_partition}} = \{\tau_{2,2}\}$ 와 $T_{\text{connected}} = \{\tau_{4,2}, \tau_{5,2}\}$ 를 T 에서 삭제한다. 동일한 방식으로 $\tau_{5,1}$, $\tau_{4,1}$, $\tau_{3,2}$ 가 차례대로 선택되어(그림 4의 (b), (c), (d)) 연결 요구 (F, G), (D, G), (D, E)에 대해 경로설정 및 과장할당을 하게 되고 마지막으로 D 는 공집합이 되어 알고리즘은 끝나게 된다. 이때 사용된 과장 수는 1이다. 같은 조건의 입력에 대해 BGAforEDP의 경우 2개의 과장을 사용한다.

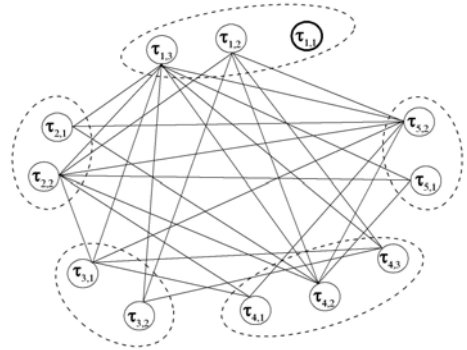


그림 3. $D=\{(A,B),(B,F),(D,E),(D,G),(F,G)\}$ 에 대한 경로 충돌 그래프

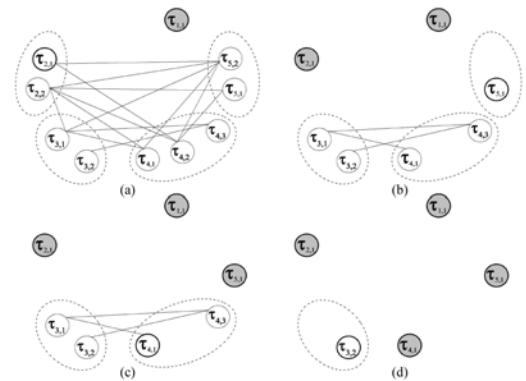


그림 4. (a) 연결요구 (A,B); (b) 연결요구 (A,F), (B,F); (c) 연결요구 (A,F), (B,F), (F,G); (d) 연결요구 (A,F), (B,F), (F,G), (D,G) 들에 대한 요구가 이루어진 후의 경로 충돌 그래프

IV. 결과 검증 및 분석

제안된 알고리즘의 성능을 평가하기 위하여 모의 실험을 수행하였다. 자체 개발된 윈도우 XP 환경 C++언어 기반 시뮬레이터를 통해 실험하였으며, 비교를 위하여 2장에서 설명한 BGAforEDP 알고리즘을 함께 실험하였다. 실험된 네트워크 토폴로지는 고정 형과 랜덤 형 토폴로지이다. 고정된 토폴로지는 잘 알려진 NSFNET을 사용하였으며, 연결요구 생성은 P_r 값을 사용하였다. P_r (Probability of Request)는 연결요구 생성에 필요한 확률 값을 나타내며, 임의로 선택된 두 개의 노드에 대해 P_r 의 확률로 연결요구가 발생함을 의미한다. 랜덤 토폴로지는 노드의 개수와 P_e 값에 따라 여러 가지를 만들어 사용했으며, 여기서 P_e 는 링크의 확률(Probability of Existence)를 나타낸다. 즉, 토폴로지 생성에 있어 각 노드들 사이에 P_e 의 확률로 링크가 생성됨을 의

미하며, P_e 의 값이 낮을수록 성긴 토폴로지가, 높을수록 조밀한 토폴로지가 생성된다. 고정된 토폴로지에서와 마찬가지로 랜덤 토폴로지에 대해서도 P_l 값을 변화시켜가며 모의실험을 수행하였다. 이와 더불어 [6]의 실험에서는 고려하지 않았던, 동일한 연결요구에 대한 중첩 허용 정도(N_c)를 고려하여 각 N_c 값에 대해 실험하였다.

먼저, 고정된 토폴로지는 그림 5와 같으며, 이는 14개의 노드와 21개의 링크로 구성되어있다. 이 토폴로지에 대해 P_l 값을 각각 0.4, 0.6, 0.8으로 하여 실험하였다. 이때 각 P_l 값에 대해 생성되는 연결요구의 수는 $P_l \times 14C_2$ 이다. 여기서, 14는 NSFNET의 총 노드 수이며, 이중 임의로 2개를 선택하여 P_l 의 확률로 연결요구를 생성했다는 의미이다. 실험은 BGAforEDP 알고리즘과 이 논문에서 제안한 2가지를 대상으로 하였으며, 그림 6은 이에 대한 결과를 보여준다. 이는 다중 연결요구의 정도를 나타내는 $N_c = 1, 2, 3$ 에 대해 각각 1000번씩 수행한 평균값이다. 그림 6에서 해당 입력에 대한 필요파장의 수 (λ)가 N_c 값에 따라 변화하는 것을 볼 수 있다. λ 와 N_c 의 관계는 비례관계이며, 이는 N_c 값이 증가함에 따라 입력되는 연결요구수가 증가하기 때문이다. $P_l = 0.4, 0.6, 0.8$ 에 대해 제안된 알고리즘은 BGAforEDP 알고리즘보다 각각 약 7%, 9%, 10% 정도까지의 성능 향상을 보인다.

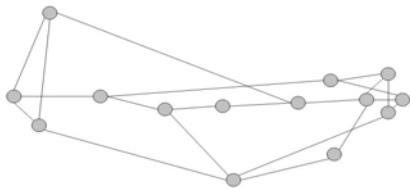


그림 5. NSFNET 토폴로지.

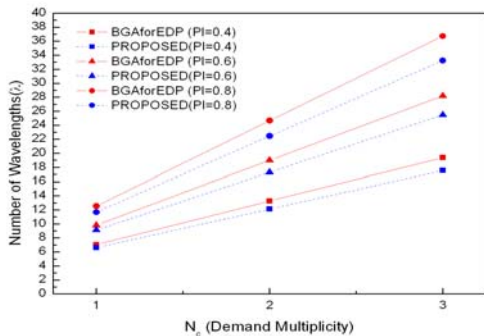


그림 6. NSFNET 에서의 모의실험에 대한 결과 그래프 (두 가지 방식비교 - 파장 vs. 연결 중첩 정도).

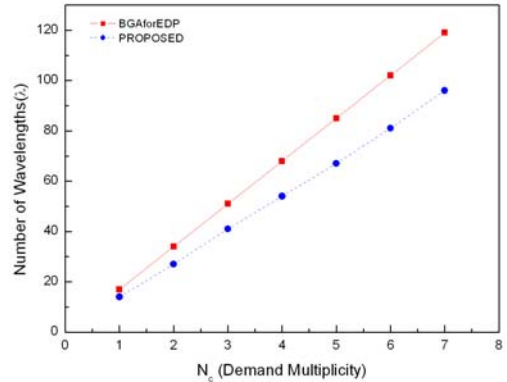


그림 7. NSFNET 에서의 모의실험에 대한 결과 그래프 (N_c 값에 따른 비교 - 파장 vs. 연결 중첩 정도).

상기한 NSFNET 토폴로지에서 모의실험 결과에서 성능향상에 영향을 주는 변수는 N_c 값과 P_l 값이다. 앞에서 서술한 대로 P_l 값에 따른 성능향상의 비는 확연한 차이를 보인다. 따라서 많은 연결요구가 있는 상황, 즉 P_l 값이 클수록 제안하는 방식의 성능은 기존의 방식보다 더욱 좋은 효과를 보인다. 다음으로 N_c 값에 대한 비교결과는 살펴보자. 결과그래프를 보면 N_c 값에 따른 두 방식의 결과가 선형적인 것을 볼 수 있다. 이를 통해 N_c 값이 제안하는 알고리즘의 성능향상에 많은 영향을 끼치지 않는다는 것을 알 수 있다. 좀더 객관적인 결과를 위하여 다양한 N_c 값에 따른 모의실험을 수행하였다. 전체적인 모의실험 환경은 앞서 수행한 모의실험들과 같다. 다만, N_c 값에 따른 좀더 정확한 경향을 알아보기 위해 P_l 값을 1로 고정시키고 토폴로지 또한 고정된 NSFNET일 때에 한하여 수행하였다. 모의실험횟수는 모든 실험에 대해 동일한 결과가 나오므로 반복수행은 하지 않았다. 이에 대한 결과를 그림 7에 나타내었다. 여기에서 보는 바와 같이 두 가지 방식의 결과 그래프는 완전한 선형은 아니지만 일정한 기울기 값을 갖는 직선형태를 보인다. 이는 N_c 값의 증가에 따라 두 가지 방식에서 사용되는 파장의 수가 일정한 비율로 증가한다는 것을 의미한다. 즉, N_c 값이 성능향상에는 큰 영향을 미치지 않음을 알 수 있다. 좀더 자세하게 살펴보면 N_c 값이 1일 때를 제외하면, 성능향상 비율은 약 20% 정도로 유지됨을 볼 수 있다.

다음으로, 제안하는 알고리즘의 좀더 일반적인 경향을 보이기 위해 고정된 토폴로지가 아닌 랜덤 토폴로지에 대해 실험을 하였다. 앞의 실험과 비슷한 실험환경이며, 랜덤 토폴로지를 만들기 위해 P_e 값을

사용하였다. P_e 값은 각각 0.2, 0.4, 0.6, 0.8 의 4가지 경우를 고려하였고 노드 수는 10개일 경우와 15개일 경우를 고려하였다. 그 외에 P_l 값은 각각 0.4, 0.6, 0.8 의 3가지 경우에 대해, N_c 값은 각각 1, 2, 3일 때에 대해 각각 1000번 수행한 결과의 평균값을 구하였다. 이에 대한 결과 값을 아래의 그림 8~11에 나타내었다. $P_e = 0.2$ 일 때의 실험 결과(그림 8)를 보면 제안하는 방식이 오히려 BGAforEDP보다 못한 결과를 보인다. 이는 제안하는 방식이 여러 개의 EDP를 찾는 과정에서 최적의 후보 경로가 되는 작은 길이의 경로를 후보대상에서 제외하기 때문에 네트워크의 링크가 적은 경우에는 오히려 BGAforEDP 알고리즘보다 못한 성능을 나타낸다. 그러나 P_l 의 값이 큰 상황 즉, 연결요구의 수가 증가할수록 제안하는 방식의 효과는 확연하게 드러난다. $P_e = 0.4$ 인 상황(그림 9)에서는 최대12%의, $P_e = 0.6, 0.8$ 인 상황(그림 10~11)에서는 각각 최대22%, 32%의 BGAforEDP 대비 성능향상을 보인다. 특히, $P_l = 0.8, P_e = 0.8$ 인 경우에는 제안된 알고리즘이 기존

의 BGAforEDP 알고리즘보다 32%의 성능향상을 보인다. 이는 NSFNET에서의 모의실험결과와 마찬가지로 P_l 값이 성능에 많은 영향을 준다는 것을 알 수 있다. 즉, 높은 P_l 값에서 제안한 알고리즘이 더 좋은 성능향상을 보인다. 이는 제안하는 알고리즘이 기존의 방식과 달리 연결요구를 수용할 때 경로의 길이가 짧고 중복 정도가 낮은 연결요구를 먼저 수용함으로써, 하나의 파장으로 수용 가능한 연결요구가 많음에 기인한다. 또 다른 변수인 N_c 에 대한 결과는 비록 정확히 선형적인 결과는 아니지만 대체로 NSFNET에서의 실험과 비슷한 양상을 보인다. 즉, N_c 값은 성능향상에 큰 영향을 미치지 않는 것으로 볼 수 있다.

종합적인 성능평가를 해보면 본 논문에서 제안한 방식이 사용되는 파장 수의 측면에서 본 성능에서 기존의 방식보다 대부분의 경우에서 더 나은 결과를 보인다. 이에 따라 성능향상에 미치는 변수를 알아보기 위하여 여러 가지의 환경에서 모의실험을 하였으며, 결론은 보다 조밀한 네트워크상황에서 보

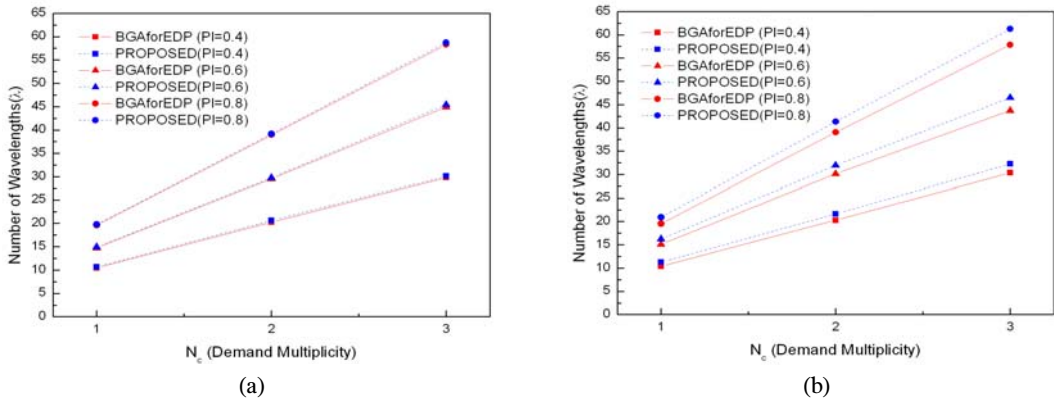


그림 8. (a) 노드 10개; (b) 노드 15개이고, P_e 값이 0.2인 랜덤 토폴로지에서의 실험결과 (λ vs. N_c).

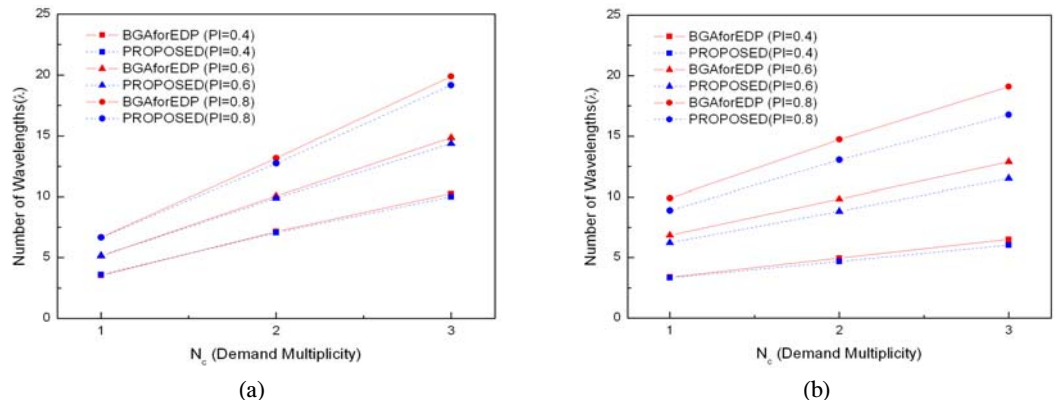
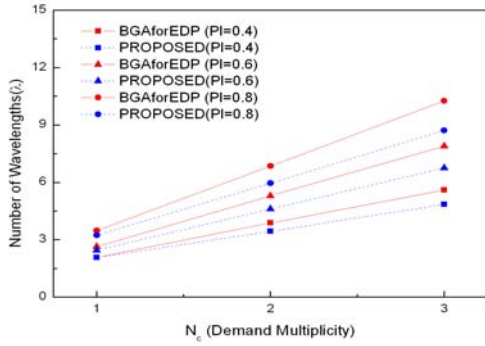
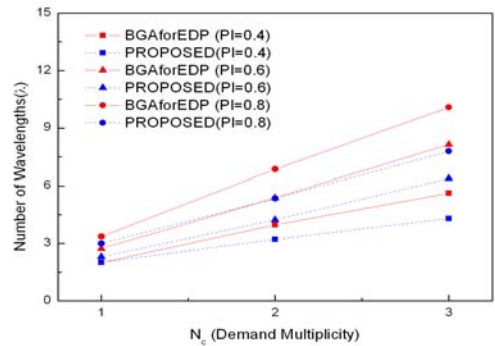


그림 9. (a) 노드 10개; (b) 노드 15개이고, P_e 값이 0.4인 랜덤 토폴로지에서의 실험결과 (λ vs. N_c).

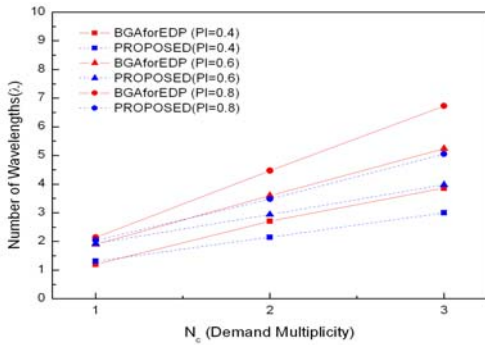


(a)

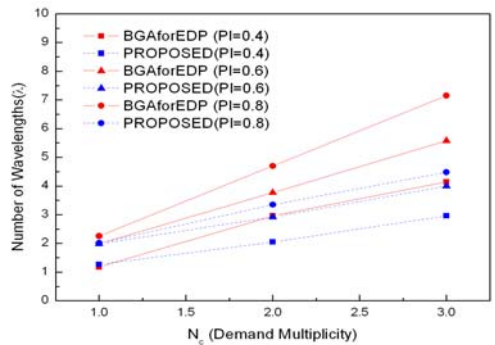


(b)

그림 10. (a) 노드 10개; (b) 노드 15개이고, P_e 값이 0.6인 랜덤 토폴로지에서의 실험결과 (λ vs. N_c).



(a)



(b)

그림 11. (a) 노드 10개; (b) 노드 15개이고, P_e 값이 0.8인 랜덤 토폴로지에서의 실험결과 (λ vs. N_c).

다 많은 연결요구가 있는 상황에서 본 논문에서 제안하는 방식이 상당히 큰 성능향상을 보인다. 이는 제안된 알고리즘의 연결요구 선택방식에서 그 이유를 찾을 수 있다. 제안된 알고리즘의 경우 입력으로 들어온 연결요구에 대해 경로 길이와 중복 정도로 우선순위를 주어 연결요구를 할당하므로, 먼저 사용되는 파장으로 할당되는 연결요구의 수가 더 많다. 그러므로 파장이 늘어날수록 해당 파장으로 할당해야 하는 연결요구의 수는 줄어든다. 이는 마지막 파장의 경우 모든 연결요구를 할당하고 남은 토폴로지상의 사용 가능한 링크의 수가 많음을 의미한다. 이런 이유로 인하여 연결요구의 수가 늘어날수록 더욱 높은 성능을 보인다. 마지막으로, 연결요구의 중첩 정도에 따른 성능비교를 알아보기 위한 모의 실험결과 연결요구 중첩 정도는 알고리즘의 성능향상에 큰 영향을 끼치지 않는 것으로 나타났다.

V. 결론

본 논문에서 최대 EDP와 경로 충돌 그래프를 이

용한 RWA문제 해결의 새로운 방안을 제안하였다. 제안된 방법의 연구 목적은 연결요구를 충족시키는 파장의 수를 최소화하는 것에 있다. 이를 위하여 정적인 연결요구상황과 파장변환기가 없는 상황을 가정하였다. 또한, 좀 더 현실적인 환경을 위해 다중 연결요구를 허용했다. 모의실험을 통한 성능평가로부터 제안된 알고리즘은 기존의 BGAforEDP방식보다 사용된 파장의 수 측면에서 최대 32%만큼의 성능향상을 보인다. 향후 좀 더 나은 성능을 위하여서는 본 논문에서 제안한 알고리즘을 간략화 하면서 수행시간을 다소 줄이는 연구가 필요하다.

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," *SPIE/Baltzer Optical Networks Magazine*, vol.1, no.1, Jan. 2000.

- [2] C. Siva Ram Murthy, "WDM optical networks,": concepts, design, and algorithms, *Prentice Hall PTR*, 2002.
- [3] Ramaswami R., K. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks," *IEEE/ACM Transactions on Networking*, vol.3, no.5, pp. 489-500, October 1995.
- [4] Kolliopoulos S., C. Stein, "Approximating Disjoint-Path Problems Using Greedy Algorithms and Packing Integer Programs," *Lecture Notes in Computer Science*, vol. 1412, 1998.
- [5] Choi J., N. Golmie, F. Lapeyriere, F. Mouveaux and D. Su, "A Functional Classification of Routing and Wavelength Assignment Schemes in DWDM Networks: Static Case," *Proceedings of VII International Conference on Optical Communication and Networks*, Jan. 2000.
- [6] P. Manohar, D. Manjunath, and R. K. Shevgaonkar, "Routing and Wavelength Assignment in Optical Networks from Edge Disjoint Path Algorithm," *IEEE Communication Letters*, vol.5, pp.211-213, May 2002.
- [7] D. Banerjee and B. Mukherjee, "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.
- [8] Y. Zhang, K. Taira, H. Takagi and S. K. Das, "An Efficient Heuristic for Routin and Wavelength Assignment in Optical WDM Networks," *IEEE International Conference on Communications*, vol.5, pp. 2734-2739, 2002.
- [9] 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.
- [10] M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," *IEEE/ACM Transactions On Networking*, vol.7, no.5, pp. 767-778, Oct. 1999.
- [11] I. Chlamtac, A. Farago, and T. Zhang, "Lightpath(wavelength) routing in large WDM networks," *IEEE Journal on Selected Areas in Communications*, vol.14, no.5, pp. 909-913, June 1996.
- [12] 김명희, 방영철, 정민영, 이태진, 추현승, "EDP들의 참조 테이블을 이용한 실용적인 경로 설정 및 파장 할당 알고리즘," *정보과학회 논문지*, 정보통신 4월호, 제31권 제2호 pp. 123~130, 2004.
- [13] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," *MobiHoc* 2002.
- [11] G. Li and R. Simha, "The Partition Coloring Problem and its Application to Wavelength Routing and Assignment," *First Workshop on Optical Networks*, Dallas, TX, 2000.

김 덕 헌 (Duk Hun Kim)

준회원



2004년 2월 성균관대학교 정보통신공학부 졸업
2004년 3월~현재 성균관대학교 전자전기공학과 석사과정 <관심분야> 무선 랜, 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

이 태 진 (Tae-Jin Lee)

정회원



1989년 2월 연세대학교 전자
공학과 졸업

1991년 2월 연세대학교 전자
공학과(석사)

1995년 12월 미국 University
of Michigan, AnnArbor,
EECS(석사)

1999년 5월 미국 University of Texas ECE(박사)

1999년 8월~2001년 2월 삼성전자 중앙 연구소 책
임 연구원

2001년 3월~현재 성균관대학교 정보통신공학부 조
교수

<관심분야> 통신 네트워크 성능 분석 및 설계, 무선
PAN/LAN, 광 네트워크 Ad-hoc/센서 네트워크

추 현 승 (Hyunseung Choo)

정회원



1988년 2월 성균관대학교 수
학과 졸업

1990년 12월 미국 University
of Texas at Dallas, 컴퓨터
공학과(석사)

1996년 12월 미국 University
of Texas at Arlington 컴퓨
터공학과(박사)

1997년 8월~1998년 3월 특허청 심사관(사무관)

1998년 3월~현재 성균관대학교 정보통신공학부 부
교수

<관심분야> 광 네트워크, 이동컴퓨팅, 라우팅 프로
토콜, 그리드 컴퓨팅