논문 98-23-12-15 한국통신학회논문지 '98-12 Vol.23 No.12

다중홉 파장분할다중화 네트워크를 위한 Minimum Congestion을 갖는 논리적 토폴로지의 설계

정회원 이 준 호*, 이 재 용**, 이 상 배**

Design of Minimum Congestion Logical Topology for Multihop Wavelength Division Multiplexing Network

Junho Lee*, Jaiyong Lee**, Sangbae Lee** *Regular Members*

요 약

본 논문에서는 초고속 패킷 전송망으로 유망한 다중홉 파장분할다중화 네트워크의 논리적 토폴로지 설계 알고리즘을 제안한다. 최근의 논리적 토폴로지 설계에 관한 연구 경향은 트래픽 요구량에 따라서 정의된 성능 평가 함수를 최적화 시키는 논리적 토폴로지를 설계하는 것으로 대부분의 연구들에서 트래픽 라우팅 방식으로 multicommodity flow 알고리즘을 사용하고 있다. 그러나, 기존 연구들의 트래픽 라우팅 방식에서는 패킷 전송시 거치게 되는 중간 노드들의 수를 고려하지 않았기 때문에 패킷의 전송 지연을 제한할 수 없다는 문제점이 생긴다. 본 논문에서는 그와 같은 문제점을 해결하기 위해서 중간 경유 노드의 수를 제한하는 트래픽 라우팅 방식을 사용하는 논리적 토폴로지 설계 알고리즘을 제안한다. 본 논문에서는 논리적 토폴로지 설계 문제를 링크의 congestion을 최소화시키는 최적화 문제로 정의하고 정의된 문제를 효율적으로 풀 수 있는 linear programing과 simulated annealing 알고리즘에 기초한 heuristic 알고리즘을 제시한다. 제시된 알고리즘의 성능 평가를 위해서 네트워크의 노드 수와 각 노드의 송신/수신기 수를 변화시키면서 모의 실험을 실행하였다.

ABSTRACT

In this paper, a logical topology design algorithm is proposed for high speed packet switching multihop wavelength division multiplexing(WDM) network. Recently, there have been many researches on a logical topology design. Most of them use the multicommodity flow algorithm for traffic routing in their design algorithms. However, their traffic routing algorithm can not bound the end-to-end packet delay. To remedy such problem, a hop limited traffic routing is used in our logical topology design algorithm. In this paper, we formulate a logical topology design as an optimization problem with the objective of minimization of maximum link congestion. An efficient heuristic algorithm based on simulated annealing and linear programing is proposed to solve such an optimization problem. To analyze and compare the performance of our algorithm, we run our algorithm with various network configurations and traffic patterns.

I. 서 론

광섬유의 대역폭은 수십 Tbps에 달하지만 여러 제약 요인 - 광섬유의 물리적 특성, electro-optic bottleneck 등으로 인해 현재 기술 수준으로 단일 채널에서 가능한 전송 속도는 10-40 Gbps를 넘지 못한다[22]. 이러한

* 서울산업대학교 전자공학과(e-mail :) 정회원, ** 연세대학교 기계전자공학부(e-mail :) 정회원
논문번호 : 97314-0905, 접수일자 : 1997년 9월 5일
※본 연구는 학술진흥재단의 '96 국내 Post-Doc 사업의 지원을 받아 수행되었음.

한계를 극복하기 위한 방법으로 제시되고 있는 것이 파장분할다중화(wavelength division multiplexing) 방식이다. 파장분할다중화 방식은 하나의 광섬유에 서로 다른 파장을 갖는 복수개의 광신호를 사용해서 데이터를 전송하는 것으로 각 광신호의 속도는 전기적 한계를 갖지만 전체 광신호들의 합은 광섬유의 대역폭에 근접하게 된다. 이와 같은 파장분할다중화 방식은 교환기와 교환기간의 트렁크(trunk) 용량을 증가시키는데 사용되기도 하지만 LAN/MAN 등과 같은 컴퓨터 네트워크의 전송 방식으로도 사용될 수 있으며 그와 같은 네트워크를 파장분할다중화 네트워크라고 부른다. 파장분할다중화 네트워크는 단일홉(single-hop) 구조와 다중홉(multi-hop) 구조로 구분되는데 본 논문에서는 고속 패킷 스위칭에 적합한 다중홉 구조의 파장분할다중화 네트워크를 고려한다.

다중홉 파장분할다중화 네트워크는 그림 1(a)와 같은 구조를 갖는다. 그림에서 각 access node는 일종의 소형 패킷 교환기로서 자신에게 연결된 user station들의 패킷 교환뿐 아니라 다른 access node에 연결된 user station에 패킷을 전달해 주는 기능을 수행한다. 다중홉 구조의 특징은 그림 1(b)에서 보이는 것처럼 그림 1(a)와 같은 물리적 구조(물리적 토폴로지)와 무관하게 각 access node 간의 연결 방식(논리적 토폴로지)를 구성할 수 있다는 점이다. 그림 1(b)에서는 두 종류의 논리적 토폴로지의 예를 보이고 있는데 각 화살표는 두 access node 간에 광전변환(opto-electronic conversion) 없이 패킷을 전송할 수 있는 광채널(optical channel)을 나타내며 논리적 링크라고도 부른다. 다중홉 방식은 단일홉 방식과는 달리 논리적 링크를 통해서 직접 연결되지 않은 access node 간의 패킷 전송은 중간의 다른 access node를 거쳐야하기 때문에 어떤 형태의 논리적 토폴로지를 사용하느냐에 따라서 네트워크의 성능이 달라지게 된다. 따라서, 다중홉 파장분할다중화 네트워크를 설계하는데 있어서는 어떤 성능 평가 함수를 최적화 시키는 논리적 토폴로지를 구하는 것이 가장 중요한 문제라고 할 수 있다.

기존의 다중홉 파장분할다중화 네트워크의 논리적 토폴로지에 관한 연구는 크게 다음과 같이 분류할 수 있다.

① 특정 형태의 그래프를 논리적 토폴로지로 제안하고 그 그래프의 제반 특성에 대한 연구[1]-[12]

② 트래픽 요구량에 따라 특정 그래프를 물리적 토폴로지에 구성(embedding) 하는 방식에 대한 연구[13]-[17]

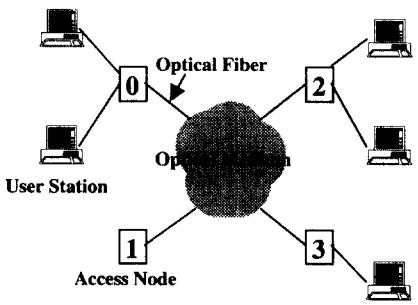

그림 1. (a) 파장분할다중화 네트워크의 물리적 토폴로지
Fig. 1 (a) Physical topology of WDM network

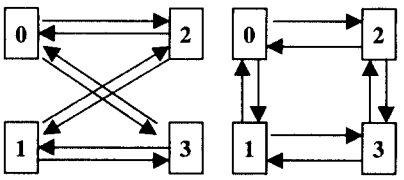

그림 1. (b) 파장분할다중화 네트워크의 논리적 토폴로지의 예
Fig. 2 (b) Logical topology of WDM network

③ 트래픽 요구량에 따라 특정 성능 평가 함수를 최적화 시키는 임의의 논리적 토폴로지 설계 방식에 대한 연구[18]-[21]

①에 속하는 연구들은 일정한 특성을 갖는 regular graph-shufflenet, de bruijin graph, kautz digraph, hypercube, torus 등등의 특성을 유도하였고 이러한 regular graph들을 멀티홉 파장분할다중화 네트워크의 논리적 토폴로지로 사용할 때의 장점을 도출하였다. 그러나 실제 노드와의 대응관계는 고려하지 않았기 때문에 실제 네트워크의 트래픽 분포에 대해 그 장점이 발휘되기 위해서는 다시 regular graph를 물리적 토폴로지에 대응시키는 방법이 고안되어야 만 한다. 이러한 동기에서 수행된 연구들이 ②번의 연구들이다. ②번의 연구들은 예상되는 트래픽 요구량을 입력으로 특정한 성능 평가 함수를 최적화 하는 논리적 토폴로지 대응 방식을 제안하고 있다. 일반적으로 이러한 물리적 토폴로지로의 대응은 그 경우의 수가 네트워크 노드 수를 N이라 하면 N! 이상으로 증가하기 때문에 모든 연구가 heuristic 알고리즘을 제안하고 있다. ②번 항목의 연구들이 특정 형태의 regular graph를 논리적 토폴로지로 사용하는 이유는 i) 그래프의 특성상 라우팅이나 폭주 제어가 손쉽고 ii) 대칭적인 트래픽

분포의 경우 부하 균등화를 쉽게 이룰 수 있다는 데 있다. 그러나 그 확장이 어렵고 노드 고장 등으로 인한 토폴로지 변경시 재 라우팅이 매우 어렵다는 단점이 있다. 이러한 이유로 최근에 활발히 진행되고 있는 ③번 항목의 연구들은 특정 형태의 그래프를 물리적 토폴로지에 대응시키는 대신에 주어진 트래픽 요구량에 따라 그 형태가 달라지는 그래프(대부분의 경우에 regular graph)를 구하는 방법을 제안하고 있다. 이 부류의 연구들은 네트워크 처리율의 최대화, 평균 지연(패킷 지연 혹은 전송 지연) 혹은 최대 링크 congestion의 최소화를 그 성능 평가 함수로 사용하였다.

본 논문에서는 ③번과 같은 접근 방식을 사용한다. 이와 같은 방식에서는 논리적 토폴로지 설계를 최적화 문제로 정의하고 최적화 문제를 풀 수 있는 heuristic 알고리즘을 제시하게 된다. 기존의 연구에서는 트래픽의 라우팅 알고리즘으로 multicommodity flow 알고리즘을 사용하고 있다. 그러나, 송신측과 수신측 사이의 path를 직접 고려하지 않았기 때문에 패킷 전송시 중간에 거치게 되는 access node의 수를 제한할 수 없다는 단점을 갖고 있다. 다중홉 구조의 가장 큰 문제점은 패킷 전송 도중에 경유하게 되는 중간 노드들에서 전자적 처리가 필요하다는 것이다. 따라서 그와 같은 전자적 처리로 인한 전송 지연을 줄이기 위해서는 송신측과 수신측 사이의 패킷 전송을 위한 path들의 hop 수를 제한하는 것이 필요한 것이다. 본 논문에서는 기존의 연구들과는 달리 논리적 토폴로지 설계를 위한 최적화 문제에 그와 같은 hop 수의 제한을 포함하는 방식을 제안하고 있다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 논리적 토폴로지 설계를 위한 최적화 문제를 정의한다. 제 3 장에서는 정의된 최적화 문제를 풀 수 있는 heuristic 알고리즘에 대해서 설명한다. 제 4 장에서는 제안된 알고리즘의 성능 평가를 위해 수행한 실험 결과를 설명하고 마지막으로 제 5 장에서 결론을 맺는다.

II. 문제 정의

이번 장에서는 논리적 토폴로지 설계를 위한 최적화 문제를 정의한다. 논리적 토폴로지는 directed-graph를 사용해서 표시할 수 있으며 기존의 연구들에서처럼 regular graph 형태를 가정한다. 최적화 문제의 목적 함수를 정의하기 위해서 먼저 다음과 같은 정의를 한다.

정의 2.1 : Congestion C_{ij}

논리적 링크 (i,j)를 지나는 모든 path에 할당된 용량의 합을 논리적 링크 (i,j)의 congestion이라고 부르며 C_{ij}로 표시한다.

정의 2.2 : C_{max}

$C_{max} = \max\{C_{ij} : \text{for all logical link } (i,j)\}$

정의 2.3 : 최적의 논리적 토폴로지

C_{max} 값이 가장 적은 논리적 토폴로지를 최적의 토폴로지라고 정의한다.

이상의 정의를 바탕으로 본 논문에서는 논리적 토폴로지 디자인을 위한 최적화 문제를 몇 가지 제약조건을 만족시키면서 C_{max} 값을 최소화시키는 mixed integer problem으로 정의하였다. 최적화 문제를 정의하기 위해서 사용한 표기를 먼저 보이고 그 뒤에 본 논문에서 정의한 최적화 문제 $P_{min\text{-}congestion}$을 설명한다.

Notation

M : node number

D : node degree(= transmitter/receiver number)

source-destination pair (i,j) : $i,j = 1 \sim M,\ i \neq j$

$(i,j) \neq (j,i)$

R_{ij} : source-destination pair (i,j) 사이의 트래픽 요구량

H : 모든 path의 maximum hop number

G(N,L) : logical topology

N : node set, $\{1 \sim M\}$

L : logical link set, $\{(m,n) : m, n \in N\}$

$l_{mn} = 1$, logical link $(m,n) \in L$인 경우

0, 그외의 경우

Cmax : maximum congestion

NP_{ij} : source-destination pair (i,j) 사이의 총 path number

P_{ij}^k : source-destination pair (i,j) 사이의 k번째 path

$P_{ij} : \{P_{ij}^k : k = 1 \sim NP_{ij}\}$

$P : \{P_{ij}^k$: for all source-destination pair (i,j), $k = 1 \sim NP_{ij}\}$

H_{ij}^k : P_{ij}^k의 hop 수

PC_{ij}^k : path P_{ij}^k에 할당된 용량

$P_{ijk}^{mn} = 1$, P_{ij}^k가 logical link (m,n)을 지나는 경우

0, 그외의 경우

$P_{min\text{-}congestion}$

min C_{max}

subject to

$$l_{mm} = 0,\ m = 1 \sim M \quad (1)$$

$$\sum_{m} l_{mn} = D, \quad n = 1 \sim M \tag{2}$$

$$\sum_{n} l_{mn} = D, \quad m = 1 \sim M \tag{3}$$

$$l_{mn} = 0, 1 \tag{4}$$

$$NP_{ij} \geq 1, \ for\ all\ source-destination\ pair\ (i, j) \tag{5}$$

$$H_{ij}^{k} \leq H, \ for\ all\ P_{ij}^{k} \in P \tag{6}$$

$$\sum_{k=1 \sim NP_{ij}} PC_{ij}^{k} = R_{ij}, \ for\ all\ source-destination\ (i, j) \tag{7}$$

$$\sum_{P_{ij}^{k} \in P} (PC_{ij}^{k} \cdot P_{ijk}^{mn}) \leq C_{\max}, \ for\ all\ (m, n) \in L \tag{8}$$

$$C_{\max}, \ PC_{ij}^{k} \geq 0 \tag{9}$$

위의 정의에서 constraint (1)-(4)까지는 논리적 토폴로지가 D-regular graph가 되어야 함을 나타낸다. 여기에 더하여서 논리적 토폴로지는 strongly-connected graph가 되어야 하며 이것은 문제 정의에서 생략하였다. Constraint (5)는 모든 source-destination 사이에 적어도 하나 이상의 path가 존재한다는 것을 의미하며 이것은 source-destination 사이의 트래픽이 여러 path들에 의해서 전달될 수 있음을 나타낸다. 본 논문에서는 트래픽 요구량을 source-destination에 할당되어야 할 대역폭으로 가정한다. 이것은 파장분할다중화 네트워크에 ATM 전송 방식을 적용하는 경우에 유용하다. ATM 셀 전송을 위해서는 각 access node 간의 virtual path(VP)를 설정해 놓는 것이 유리한데 그런 경우에 위와 같은 문제 정의가 유용하게 된다. 그러나, 기본적으로 constraint (7),(8)은 source-destination 사이의 트래픽 라우팅을 위한 multicommodity flow 알고리즘을 path-arc formulation으로 표시한 것으로 트래픽 요구량의 특정한 의미와는 무관하다. Constraint (6)은 path의 hop 수를 제한하기 위한 것으로 path의 hop 수는 path가 지나게 되는 논리적 링크의 수가된다. 이와 같은 path의 hop 수가 커질 수록 source-destination 사이의 중간 노드에서 전자적인 처리가 많이 필요하게 된다. 따라서, 그와 같은 전자적 처리로 인해 발생하는 지연을 고려하기 위해서 이와 같이 path의 hop 수를 제한하게 된 것이다.

III. 제안된 알고리즘

앞 절에서 정의한 논리적 토폴로지 설계 문제 $P_{min\text{-}congestion}$은 그 정확한 해를 구할 수 없다[20]. 따라서 heuristic 알고리즘을 사용하여 최적에 가까운(suboptimal) solution을 구하였다. 본 논문에서는 simulated annealing(SA)[23]을 이용하였다. Simulated annealing은 확률적 특성을 갖는 근역 탐색(local search) heuristic으로 일반적인 heuristic algorithm이 근역 최적값(local optimum)을 구하는데 반해 genetic algorithm[23]과 더불어 전역 최적화(global optimization)에 이용될 수 있는 알고리즘이다. 이것은 simulated annealing의 탐색방식에 기인하는 것으로 현재 solution 값과 그 neighbor solution 값을 비교하여 neighbor solution 값이 현재 solution 값보다 최적일 경우 neighbor solution을 현재 solution으로 하여 다시 탐색을 한다. 그러나 반대의 경우 기존의 방식처럼 현재 solution을 그대로 유지하지 않고 일정한 확률을 가지고 neighbor solution을 현재 solution으로 채택한다. 이러한 특성으로 인해 근역 최적점에서 벗어나 다른 더 나은 최적점을 찾을 수 있게 된다.

SA algorithm을 적용하기 위해 다음과 같은 정의를 한다.

정의 3-1

X : feasible solution set of $P_{min\text{-}congestion}$ = {G(N,L)}
z(x) : objective function defined over X
neighbor(x) : randomly selected feasible solution in the neighborhood of x
neighbor(x) ≠ x, neighbor(x) ∈ X
T : current temperature
$N_{EQUILIBRIUM}$: max equilibrium iteration number
$N_{ANNEALING}$: max annealing iteration number

이를 이용 $P_{min\text{-}congestion}$을 위한 SA algorithm을 나타내면 다음과 같다.

SA algorithm for $P_{min\text{-}congestion}$ (SA_$P_{min\text{-}congestion}$)

Step 1. Select a initial feasible solution x_0 at random
  Set $x_{OPT} = x_{CURRENT} = x_0$
  Set $Z_{OPT} = Z_{CURRENT} = z(x_0)$
  i = 1
Step 2. j = 1
Step 3. Set x_{NEW} = neighbor($x_{CURRENT}$)
  Set $Z_{NEW} = z(x_{NEW})$
  If $Z_{NEW} < Z_{OPT}$ then
    set $Z_{OPT} = Z_{NEW}$
    set $x_{OPT} = x_{NEW}$
  end if
  If $Z_{NEW} > Z_{CURRENT}$ then

set $x_{CURRENT}$ such that

$$x_{CURRENT} = \begin{cases} x_{NEW} \text{ with probability } \exp(-\frac{Z_{NEW} - Z_{CURRENT}}{T}) \\ x_{CURRENT} \text{ with probability } 1 - \exp(-\frac{Z_{NEW} - Z_{CURRENT}}{T}) \end{cases}$$

Step 4. If $j < N_{EQUILIBRIUM}$ then
set $j = j+1$
goto step 3
end if

Step 5. If $i < N_{ANNEALING}$ then
set $i = i+1$
change current temperature T
goto step 2
end if

다음에는 $SA\_P_{min\text{-}congestion}$ 알고리즘을 초기해, neighbor (x), z(x) 풀이 방법의 순으로 설명한다.

3.1 초기해

모든 heuristic algorithm에 있어서 초기해의 선정은 아주 중요하다. 그 이유는 어떤 초기해에서 출발하였나에 따라 최종 결과의 성능이 달라질 수 있기 때문이다. 그러나 좋은 초기해를 미리 알 수는 없다. 따라서 각기 다른 초기해를 이용 알고리즘을 여러 번 수행하여 그 중에서 가장 좋은 결과를 택하는 방식이 많이 사용된다. 본 논문에서도 여러 초기해를 이용 $SA\_P_{min\text{-}congestion}$ 알고리즘을 수행시켜 가장 좋은 결과를 택하였다.

$SA_P_{min\text{-}congestion}$의 초기해는 임의로 선택된다. G(N,L)은 strongly connected D-regular graph이어야 하므로 초기해 생성시 strongly connected 여부를 확인하여야 한다. strongly connected D-regular graph를 임의로 생성시키는 알고리즘은 알려진 것이 없다. 본 논문에서는 먼저 D-regular graph를 0-1 integer feasibility problem[23]으로 생성시키고 strong connectivity check 알고리즘[24]을 통하여 strong connectivity가 만족될 때까지 random change - link 혹은 node exchange를 하여서 초기해를 구하였다.

3.2 Neighbor(x)

SA 알고리즘 적용을 위해서는 구하고자 하는 G(N,L)의 neighbor를 정의하고 그 neighbor를 임의로 구하는 방식이 규정되어야 한다. 본 논문에서 $SA_P_{min\text{-}congestion}$을 위해 규정한 neighbor의 정의는 다음과 같다.

정의 3-2 : Neighbor(x)
현재 solution x에서 임의의 두 노드를 선택하여 그 out-going 혹은 in-going n-edge를 교환하여 생성되는 strongly connected D-regular graph (n = 2 or 2D)

기존의 연구에서 이용되던 방법은 branch-exchange [19] 혹은 out-going n-edge exchange[20] 등이 있는데 본 논문에서는 그와 더불어 in-going n-edge exchange를 더 고려하였다. 그림 2에 (2,2)-Shufflenet 형태의 G(N,L)에서 각 방식의 예를 보인다. 그림 2(a), (b)는 임의로 선택된 두 노드가 각각 1,2 그리고 5,6 일 때의 2-edge exchange의 예가 된다. 이러한 예에서 알 수 있듯이 임의의 두 노드간 edge-exchange는 그 결과로 생성되는 neighbor의 regularity를 보장해 준다. 그러나 항상 strong connectivity를 만족시켜주지는 않는다. 또한 자기 루프(self loop)나 다중 edge가 생길 수 도 있다. 그 예를 그림 3에 보인다. 그림 3(b), (c)의 경우는 선택된 두 노드 사이의 edge exchange 전에 그 타당성을 바로 판단할 수 있다. 그러나 (a)의

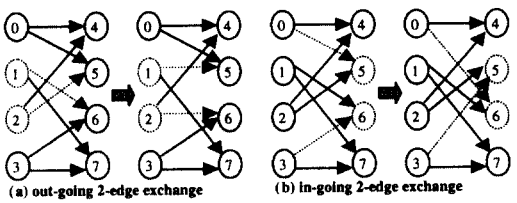

그림 2. (2,2)-Shufflenet에서 2-edge exchange의 예
Fig. 2 2-edge exchange for (2,2)-Shufflenet

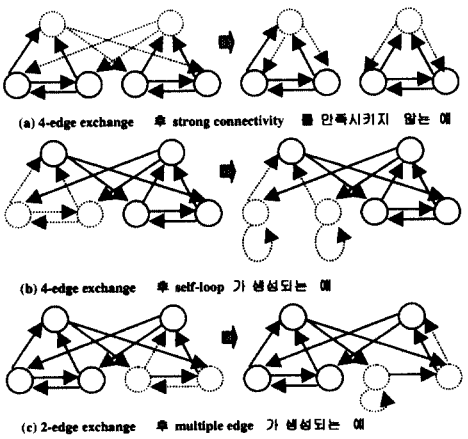

그림 3. Invalid n-edge exchange의 예
Fig. 3 Invalid n-edge exchange

경우는 전체 그래프에서 검사하여야 하므로 edge exchange 후 생성된 neighbor(x)에서 strong connectivity check 알고리즘을 이용하여 그 타당성을 검사하였다.

본 논문에서 사용된 random n-edge exchange 방식은 다음의 4가지이다.

1. Out-going 2D-edge exchange
2. In-going 2D-edge exchange
3. Out-going 2-edge exchange
4. In-going 2-edge exchange

각 방식은 G(N,L)의 adjacency matrix A를 이용하여 구현된다. 본 논문에서 제안된 in-going edge exchange 알고리즘을 보이면 다음과 같다.

In-going 2D-edge exchange

G(N,L)의 adjacency matrix를 다음과 같이 정의한다.

$$A = [a_1, a_2, \cdots, a_N],\ a_i = ith\ column\ vector = \begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{iN} \end{pmatrix}$$

Step 1. Set k = 0

Step 2. Select two column pair (i,j) with probability $\frac{1}{{}_{N-k}C_2}$ without replacement

Step 3. If $a_{ij} = 1$ *or* $a_{ji} = 1$ then
  set k = k + 1
  if k < ${}_N C_2$ then
    go to step 2
  else
    go to step 5
  End if
 Else
  Swap column a_i, a_j
 End if

Step 4. check strong connectivity of A
  If failure then
   swap column a_i, a_j
   k = k + 1
   if k < ${}_N C_2$ then
    go to step 2
   End if
  End if

Step 5. End

In-going 2-edge exchange

$in-edge(i) = \{j : a_{ij} = 1\}$

Step 1. Set k = 0

Step 2. Select two pair (i,j) with probability $\frac{1}{{}_{N-k}C_2}$ without replacement

Step 3. Set n = 0

Step 4. Select two in-edge pair (in-edge(i), in-edge(j)) with probability $\frac{1}{(D^2-1)}$ without replacement

Step 5. If in-edge(i) == in-edge(j) or i == in-edge(j) or j == in-edge(i)
  or $a_{in-edge(i),j} = 1$ *or* $a_{in-edge(j),i} = 1$ then
  n = n + 1
  if n < D^2 then
   go to step 4
  else
   go to step 7
  End if
 else
  set $a_{in-edge(i),i} = 0$, $a_{in-edge(i),j} = 1$
   $a_{in-edge(j),j} = 0$ $a_{in-edge(j),i} = 1$
 End if

Step 6. check strong connectivity
  If failure then
   set $a_{in-edge(i),i} = 1$, $a_{in-edge(i),j} = 0$
    $a_{in-edge(j),j} = 1$ $a_{in-edge(j),i} = 0$
   set n = n + 1
   if n < D^2 then
    go to step 4
   End if
  End if

Step 7. Set k = k + 1
 if k < ${}_N C_2$ then
  go to step 2
 End if

Step 8. End

3.3 z(x) 풀이 방법

z(x)는 현재 선택된 논리적 토폴로지의 목적 함수 값을 나타낸다. 기존의 연구들에서는 바로 multicommod

ity flow 문제를 풀어서 이 값을 구하였지만 본 논문에서는 source-destination 사이의 path의 hop 수를 제한하였기 때문에 다음과 같은 알고리즘을 사용해서 z(x) 값을 구하였다.

Step 1. 모든 source-destination 사이에서 hop 수가 H 이하인 모든 elementary path들을 구한다.
Step 2. 다음과 같은 linear program을 푼다.

$$\min C_{max}$$
subject to
$$\sum_{k=1..NP_{ij}} PC_{ij}^{k} = R_{ij},\ for\ all\ source-destination\ (i,j)$$
$$\sum_{P_{ij}^{k} \in P} (PC_{ij}^{k} \cdot P_{ijk}^{mn}) \le C_{max},\ for\ all\ (m,n) \in L$$
$$C_{max},\ PC_{ij}^{k} \ge 0 \qquad (9)$$

Step 1은 모든 source-destination (i,j)의 P_{ij}를 구하는 것으로 본 논문에서는 elementary path enumeration algorithm[25]을 사용하였다. Step 1에 의해서 NP_{ij}, P_{ijk}^{mn} 등이 결정되며 step 2에서는 그와 같은 값들과 트래픽 요구량 R_{ij}를 입력으로 linear program을 풀어서 C_{max} - 즉, z(x) 값을 구하게 된다.

IV. 성능 평가

제안된 알고리즘의 동작을 검증하고 성능을 평가하기 위해서 표 1과 같은 실험 환경을 설정하고 제안된 알고리즘을 적용해서 논리적 토폴로지의 설계를 수행하였다. 각 실험 환경마다 4가지 종류의 트래픽 패턴에 대해서 알고리즘을 적용하였는데 트래픽 패턴은 각 access node의 트래픽이 다른 access node로 분산되는 형태를 나타낸다. 본 논문에서 사용된 트래픽 패턴은 다음과 같다.

$$R_{ij} = total\ traffic\ demand\ of\ access\ node\ i \times scaling_{ij}$$
$$\sum_{j} scaling_{ij} = 1.0,\ scaling_{ii} = 0.0$$

- Uniform type

$$scaling_{ij} = \frac{1}{access\ node\ number - 1},\ i \neq j$$

- Ring type

$$scaling_{i(i+1)} = 0.5,\ scaling_{ij} = \frac{0.5}{access\ node\ number - 2},\ j \neq i+1$$

- Centralized type

$$scaling_{i}0 = 0.6,\ scaling_{ij} = \frac{0.4}{access\ node\ number - 2},\ j \neq 0$$

- Random type

access node 간의 트래픽 분포가 임의의 형태를 가지며, 경우에 따라서는 두 access node 간에 전송이 이루어지지 않을 수도 있다.

표 1. 실험 환경

실험환경	Access Node 개수	송신/수신기 개수
1	8	2
2	10	2
3	10	3
4	20	4
5	20	5

이상과 같은 트래픽 패턴들을 표 1의 각 실험 환경에 적용한 결과를 표 2에 요약해 놓았다. 표 1에서 보이는 lower bound는 Minimum Flow Tree(MFT)[26] 방식으로 계산된 것으로 제안된 알고리즘의 성능을 평가하기 위해서 제시한 것이다. 표 2의 결과를 통해서 제안된 알고리즘이 lower bound에 비해서 평균적으로 17.4% 정도 높은 minimum congestion 값을 갖는 것을 알 수 있다. 그러나, 제안된 알고리즘의 성능은 이 값보다는 작을 것으로 예측되는데 그 이유는 첫째로 MFT 방식에 의한 lower bound 값이 실제 lower bound 보다 작은 값을 가지기 때문이며 두 번째로는 제안된 알고리즘에서는 hop 수의 제한을 두었기 때문이다.

표 2의 결과에서 실험환경 5의 결과가 가장 좋지 않음을 알 수 있다. 그 이유는, access node 수와 node degree(즉, 송신/수신기 수)가 클 수록 access node 간의 path가 많아지는데 본 논문에서처럼 path의 hop 수를 제한하는 경우 그렇지 않은 경우에 비해서 minimum congestion 값이 증가하는 비율이 커지기 때문이다. 이것은 바꾸어 얘기하면 hop 수를 고려하지 않는 기존의 연구들의 결과는 access node 수와 node degree가 큰 환경에서는 본 논문에서 제안된 알고리즘보다 적은 congestion 값을 갖는 논리적 토폴로지를 구성할 수는 있지만 그 반면에 트래픽 라우팅이 hop 수가 큰 path들을 통해서 이루어져서 전송 지연이 커지는 결과를 가져오게 된다고 할 수 있다.

제안된 알고리즘의 성능을 기존의 연구와 비교하

기 위해서 [20]과 동일한 환경에 대해서 제안된 알고리즘을 실행하였다. 표 3에 제안된 알고리즘과 [20]의 성능을 비교해 놓았다. 표 3의 결과를 보면 제안된 알고리즘의 결과가 [20]의 결과에 비해서 평균적으로 3.0% 정도 높은 값을 갖는다는 것을 알 수 있다. 이 결과는 물론 제안된 알고리즘이 access node 간의 path의 hop 수를 제한하고 있기 때문이다. 표 3의 결과를 통해서 제안된 알고리즘이 [20]과 거의 유사한 minimum congestion 값을 가지며 전송지연이 작은 논리적 토폴로지를 설계한다는 것을 확인할 수 있다.

표 2. 표 1 실험환경에 대한 알고리즘 수행 결과

실험환경	트래픽 패턴	Minimum Congestion	하한값	하한값과의 차이(%)
1	Uniform	5707.8	5574.3	2.4
	Ring	5920.5	5223.0	13.4
	Centralized	6427.7	5138.8	25.1
	Random	5301.4	4276.7	24.0
2	Uniform	7948.4	7751.9	2.5
	Ring	7961.4	7205.4	10.5
	Centralized	8203.9	7071.0	16.0
	Random	7400.0	6439.9	14.9
3	Uniform	4326.9	4078.0	6.1
	Ring	4165.9	3819.1	9.1
	Centralized	4298.0	3750.9	14.6
	Random	4009.5	3505.8	14.4
4	Uniform	6902.2	6081.4	13.5
	Ring	6306.7	5645.4	11.7
	Centralized	6540.8	5526.8	18.3
	Random	6412.3	5607.1	14.4
5	Uniform	6317.3	4726.0	33.7
	Ring	5905.7	4390.4	34.5
	Centralized	5632.2	4298.6	31.0
	Random	5996.8	4336.1	38.3

표 3. 기존의 연구와의 비교 결과

트래픽 패턴	제안된 알고리즘	기존의 연구	기존의 연구와의 차이(%)
Uniform	66.7	66.6	0.2
Quasi-Uniform2	69.6	66.5	4.7
Ring	133	127	4.7
Quasi-Uniform1	63.2	60.8	3.9
Disconnected	290.3	278	4.4
Central	335	335	0

V. 결 론

본 논문에서는 초고속 패킷 전송망으로 유망한 다중홉 파장분할다중화 네트워크를 위한 논리적 토폴로지 설계 알고리즘을 제안하였다. 기존의 연구들과는 달리 본 논문에서는 패킷이 경유하게 되는 중간 노드들의 수를 제한할 수 있는 트래픽 라우팅 방식을 사용하는 새로운 설계 알고리즘을 제시하였다. 제시된 알고리즘의 성능 평가를 위해서 네트워크의 노드수와 각 노드의 송신/수신기 수를 변화시키면서 모의 실험을 실행하였다. 모의 실험 결과를 통해서 제안된 알고리즘이 이론적인 하한값보다 평균적으로 17.4% 정도 높은 링크 congestion을 갖는 논리적 토폴로지를 설계한다는 것을 알 수 있었다. 또한, 기존의 연구와의 비교를 위해서 수행한 모의 실험을 통해서 제안된 알고리즘이 패킷 전달 경로의 hop 수를 제한하면서도 기존의 연구 결과와 크게 차이가 나지 않는 링크 congestion 값을 갖는 논리적 토폴로지를 설계할 수 있음을 확인할 수 있었다.

참 고 문 헌

1. Michael G. Hluchyj, ShuffleNet : An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks, Journal of Lightwave Technology, Vol. 9, No. 10, pp.1386-1397 Oct., 1991.
2. Kumar N. Sivarajan, Lightwave Networks Based on de Bruijin Graphs, IEEE/ACM Trans. on Networking, Vol. 2, No. 1, pp.70-79, Feb., 1994.
3. Swie Tsing Tan, Embedded Unidirectional Incomplete Hypercubes for Optical Networks, IEEE/ACM Trans. on Networking, Vol. 41, No. 9, pp.1284-1289, Sept., 1993.
4. M. Ajmone Marsan, Topologies for Wavelength-Routing All-Optical Networks, IEEE/ACM Trans. on Networking, Vol. 1, No. 5, pp.534-546, Oct., 1993.
5. I. Chlamtac, Lightnets: Topologies for High-Speed Optical Networks, Journal of Lightwave Technology, Vol. 11, No.5/6, pp.951-961, May/June, 1993.
6. Dhritiman Banerjee, The Multidimensional Torus: Analysis of Average Hop Distance and Application as Multihop Lightwave Network, Proc. IEEE ICC94, pp.1675-1680, May, 1994.
7. K. Wendy Tang, CayleyNet : A Multihop WDM-

Based Lightwave Network, Proc. IEEE INFOCOM94, pp.1260-1267, June, 1994.
8. Subrata Banerjee, Hypercube Connected Rings : A Fault-Tolerant and Scalable Architecture for Virtual Lightwave Network Topology, Proc. IEEE INFOCOM94, pp.1236-1243, June, 1994.
9. Geetha Panchapakesan, ON MULTIHOP OPTICAL NETWORK TOPOLOGY USING KAUTZ DIGRAPHS, Proc. IEEE INFOCOM95, pp.675-682, April, 1995.
10. Sung-Woo Park, A Virtual Topology for WDM Multihop Lightwave Networks, Proc. IEEE INFOCOM95, pp.701-708, April, 1995.
11. S. Jiang, Regular Multicast Multihop Lightwave Networks, Proc. IEEE INFOCOM95, pp.692-700, April, 1995.
12. F. Ayadi, Bilayered ShuffleNet: A New Logical Configuration for Multihop Lightwave Networks, Proc. IEEE GLOBECOM93, pp.1159-1163, Nov., 1993.
13. Aura Ganz, Wavelength Assignment in Multihop Lightwave Networks, Proc. IEEE INFOCOM93, pp.1367-1374, March, 1993.
14. Subrata Banerjee, Algorithms for Optimized Node Arrangements in ShuffleNet Based Multihop Lightwave Networks, Proc. IEEE INFOCOM93, pp. 557-564, March, 1993.
15. S. Jiang, DESIGN OF MULTICAST MULTILAYERED LIGHTWAVE NETWORKS, Proc. IEEE GlOBECOM93.
16. Zhensheng Zhang, A Heuristic Wavelength Assignment Algorithm for Multihop WDM Networks with Wavelength Routing and Wavelength Reuse, Proc. IEEE INFOCOM94, pp.534-543, June, 1994.
17. Biswanath Mukherjee, SOME PRINCIPLES FOR DESIGNING A WIDE-AREA OPTICAL NETWORK, Proc. IEEE INFOCOM94 pp.110-119, June, 1994.
18. Aura Ganz, Efficient Algorithm for Virtual Topology Design in Multihop Lightwave Networks, IEEE/ACM Trans. on Networking, Vol. 2, No. 3, June, 1994.
19. Jean-Francois, Logically Rearrangeable Multihop Lightwave Networks, IEEE Trans. on Communications, Vol. 39, No. 8, Aug., 1991.
20. Bülent Yener, Logical Embeddings for Minimum Congestion Routing in Lightwave Networks, CTR TR CUCS-008-93, Columbia Univ., 1993.
21. Rajiv Ramaswami, Design of Logical Topologies for Wavelength-Routed All-Optical Networks, Proc. IEEE INFOCOM95, pp.1316-1325, April, 1995.
22. Rolf Heidemann, 10-GB/s Transmission and Beyond, Proceedings of the IEEE, Vol. 81, No. 11, pp.1558-1567, Nov., 1993.
23. Katta G. Murty, Operations Research Deterministic Optimization Models, Prentice-Hall, 1995.
24. M. N. S. Swamy, Graphs Networks and Algorithms, A Wiley Interscience Publication, pp.464-466, 1981.
25. M. Gondran, Graphs and Algorithms, ch. 3, John Wiley & Sons Ltd., 1984.
26. Bülent Yener, "A Study of Upper and Lower Bounds for Minimum Congestion Routing in Lightwave Networks," Proc. IEEE INFOCOM'94 pp. 138-147, June, 1994.

이 준 호(Junho Lee) 정회원
1988년 : 연세대학교 전자공학과 학사
1990년 : 연세대학교 전자공학과 석사
1996년 : 연세대학교 전자공학과 박사
1998년 3월～현재 : 서울산업대학교 전자공학과 전임강사
<연구분야> WDM 네트워크 설계 및 성능 평가, ATM 스위칭 시스템, IP switching

이 재 용(Jaiyong Lee) 정회원
한국통신학회 논문지 제24권 제 12호
현재 : 연세대학교 기계 전자 공학부 교수

이 상 배(Sangbae Lee) 정회원
한국통신학회 논문지 제24권 제 12호
현재 : 연세대학교 기계 전자 공학부 교수