

# 최단경로 라우팅을 이용한 효율적인 인터넷 트래픽 엔지니어링

정희원 이영석\*

## Efficient Internet Traffic Engineering based on Shortest Path Routing

Youngseok Lee\* *Regular Member*

### 요 약

단일 최단경로 라우팅 알고리즘은 흔히 최대 링크 로드를 최소화하는 인터넷 트래픽 엔지니어링에 적합하지 않다고 알려져있다. 또한 여러 개의 최단경로들을 동시에 사용하여 트래픽을 나누어서 전송하는 방법은 전체 트래픽 디맨드를 고려하지 않기 때문에 최대링크로드를 반드시 최소화시키지는 못한다. 하지만, 여러 개의 최단경로들이 동일 송수신 노드 사이에 존재하면, 기존 라우팅 프로토콜에 변화를 주지 않고도 최대 링크 로드를 최소화시킬 수 있는 최단경로들의 집합을 선택하여 이용할 수 있다. 이러한 최단 경로들을 계산할 때에는 동일 목적지에 대해서는 송신자에 관계없이 동일한 넥스트-홉을 유지해야한다. 본 논문에서는 트래픽 엔지니어링을 위한 최단 경로 라우팅 알고리즘의 최적해를 구할 수 있는 ILP 문제를 제시하고, 근사해를 구하기 위한 휴리스틱 알고리즘을 제안하고 성능을 평가하였다.

### ABSTRACT

Single shortest path routing is known to perform poorly for Internet traffic engineering (TE) where the typical optimization objective is to minimize the maximum link load. Splitting traffic uniformly over equal cost multiple shortest paths in OSPF and IS-IS does not always minimize the maximum link load when multiple paths are not carefully selected for the global traffic demand matrix. However, among all the equal cost multiple shortest paths in the network, a set of TE-aware shortest paths, which reduces the maximum link load significantly, can be found and used by IP routers without any change of existing routing protocols and serious configuration overhead. While calculating TE-aware shortest paths, the destination-based forwarding constraint at a node should be satisfied, because an IP router will forward a packet to the next-hop toward the destination by looking up the destination prefix. In this paper, we present a problem formulation of finding a set of TE-aware shortest paths in ILP, and propose a simple heuristic for the problem. From the simulation results, it is shown that TE-aware shortest path routing performs better than default shortest path routing and ECMP in terms of the maximum link load with the marginal configuration overhead of changing the next-hops.

### I. 서론

웹, 멀티미디어 스트리밍, 및 피어-투-피어(peer-to-peer) 등의 다양한 인터넷 서비스로 인해 인터넷 트래픽은 급속도로 증가하고 있다. 하지만, 네트워크 기술의 발전과 용량의 증가에도 불구하고, 네트

워크에서의 혼잡현상(congestion)은 여전히 발생하고 있다. 트래픽 양의 절대적인 증가 현상 이외에도 링크 단선과 같은 네트워크에서의 결함은 트래픽의 경로를 바꾸게 하여 특정 지역에서의 혼잡현상을 더욱 심화시킨다.

트래픽 엔지니어링은 네트워크 혼잡현상을 제어하

\* 충남대학교 전기정보통신공학부 컴퓨터전공 (yslee@cs.cnu.ac.kr)

논문번호: 030284-0703, 접수일자: 2003년 7월 3일

※ 이 논문은 한국과학재단의 해외 Post-doc. 연수지원에 의하여연구되었음.

고 네트워크 성능을 최적화시키는 효과적인 방법이다. [1][2]에서 언급된 것처럼 트래픽 엔지니어링이라는 것은 인터넷 트래픽의 측정, 모델링, 특성화 및 제어에 관한 기술과 체계적인 방법들을 포함하고 있다. 트래픽 엔지니어링의 목표는 네트워크 자원 사용률과 네트워크 성능을 최적화하여, 효율적이고 안정적인 네트워크 운영을 가능하게 하는 것이다. 대표적인 트래픽 엔지니어링의 최적화 목적함수는 네트워크의 최대 링크 로드를 최소화시키는 것으로 흔히 네트워크의 혼잡현상을 대표하는 값이다<sup>[3]</sup>.

인터넷에서의 트래픽 엔지니어링은 대개 라우팅 프로토콜에 의해서 수행된다. IP 네트워크에서의 IGP(Interior Gateway Protocol)는 링크에 할당된 비용을 이용하여 최단 경로를 계산하는 알고리즘에 기반을 두고 있다. 비록 최단 경로 라우팅 알고리즘이 간단하고 구현이 용이하다는 장점을 가지고 있지만, 단일 경로만을 이용하기 때문에 네트워크의 처리율을 최대화시키지는 못한다. 한편으로, 트래픽을 다중 경로로 분산시켜 최적화된 경로에 트래픽을 할당하는 문제는 최적 라우팅이라고 불리는데, 이는 multi-commodity flow problem으로 해결될 수 있다<sup>[4]</sup>. 최적 라우팅은 또한 중단간 지연시간을 최소화시키는 최소 지연시간 라우팅 문제로도 접근되었다<sup>[5]</sup>. 최적 라우팅은 다중 경로의 형태로 네트워크에서 구현될 수 있다. 인터넷 라우팅 프로토콜에서는 동일한 비용으로 목적지에 도착하는 경로들이 여러 개 존재할 때 다중 최단 경로들을 이용하여 트래픽을 전송할 수 있다. OSPF(Open Shortest Path First)<sup>[6]</sup>와 IS-IS(Intermediate System- Intermediate System)<sup>[7]</sup>과 같은 라우팅 프로토콜에서는 명시적으로 ECMP(Equal Cost Multi-Path)라고 불리는 다중 최단 경로 라우팅 알고리즘을 지원한다. 그리고,

RIP(Routing Information Protocol) 및 다른 라우팅 프로토콜을 이용하는 몇몇 라우터 또는 스위치들도 ECMP를 지원하고 있다. OSPF와 IS-IS의 다중경로 라우팅을 최적화시키기 위하여 OMP(Optimized Multi-path) [8]가 IETF(Internet Engineering Task Force)에서 제안된 바 있다. 하지만, 인터넷 라우팅 프로토콜에서 사용중인 ECMP는 목적지로 향하는 다중 최단 경로들에 트래픽을 균등하게 배분하고, 전체적인 트래픽 수요를 고려하지 않고 다중 경로를 선택할 수 있기 때문에 최대 링크 로드를 반드시 최소화시키지는 않는다. 최근에는 최적 라우팅 문제가 적절한 링크 비용값을 선택하여 최단 경로 문제로 변환될 수 있다는 것이 [9]에서 밝혀졌다. 또한, [10]에서는 링크 비용값을 적절하게 변경시킴으로써 인터넷 트래픽 엔지니어링의 목표를 쉽게 달성할 수 있다는 결과를 보였다. 하지만, 링크 비용값을 변경하게 되면, 설정된 타이머에 의하여 LSA(Link State Advertisement) 메시지가 플러딩되어 네트워크의 각 라우터가 최단 경로를 새로 계산하여 라우팅 테이블을 업데이트한다. 따라서, 네트워크 상태가 안정되기 까지 시간이 걸리게 된다. LSA 메시지가 플러딩되는 동안에는 순간적으로 라우팅 루프나 트래픽이 급격하게 변하는 현상이 생겨 사용자 트래픽을 방해할 수 있다. 일반적으로, 어떤 링크의 비용을 최적화시킬지 미리 계산하고 이의 영향을 예측하는 것은 쉽지 않다. 최근의 연구에서는 각 라우팅 prefix 별로 ECMP를 실행시켜, 최적 라우팅에 근접하는 알고리즘이 제안되었다<sup>[11]</sup>.

최근에 등장한 MPLS(Multi-Protocol Label Switching)<sup>[12]</sup> 기술은 입력 라우터와 출력 라우터 사이의 경로를 RSVP(Resource Reservation Protocol) [13]와 같은 신호 프로토콜을 이용하여

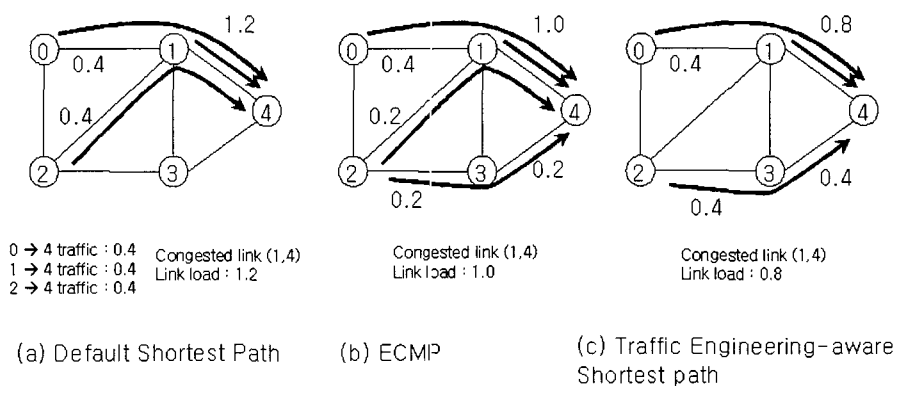


그림 1. 트래픽 엔지니어링에 적합한 최단 경로의 예

명시적으로 경로를 설정하고, IP 패킷들에 고정된 크기의 레이블을 부착하여 고속 스위칭 기능과 트래픽 엔지니어링 기능을 가능하게 하였다. 중간 LSR(Label Switched Router)들은 IP 패킷을 포워딩할 때 일반적으로 사용되는 IP 패킷의 가변 크기의 목적지 주소가 아닌 고정된 크기의 레이블을 이용한다. 이러한 명시적인 경로 설정 방법과 레이블 기반 포워딩 방식 덕분에 MPLS 네트워크에서의 트래픽 엔지니어링이 IP 네트워크에서보다 더욱 효과적으로 수행된다. 하지만, MPLS는 IP 네트워크에서 부가적인 제어 계층이기 때문에 추가적인 관리 비용이 요구된다.

일반적으로 단일 최단 경로 라우팅 알고리즘은 트래픽 엔지니어링에 효과적이지 않다고 알려져 있다. 대개 동일 노드 사이에 여러 개의 최단 경로들이 존재할 때, IP 라우터는 임의의 한 경로를 선택하여 목적지에 대한 넥스트-홉을 라우팅 테이블에 업데이트한다. 따라서, 최단 경로 라우팅 알고리즘에서는 최대 링크 로드를 고려하지 않는다. 하지만, 네트워크 토폴로지와 트래픽 수요 테이블이 알려져 있을 때에는 임의의 최단 경로를 선택하지 않고 트래픽 엔지니어링을 고려한 최단 경로를 찾을 수 있고, 이를 라우터에서 이용할 수 있다. 즉, 트래픽 엔지니어링에 적합한 최단 경로는 단일 경로를 이용하여 최대 링크 로드를 최소화시킬 수 있는 경로들의 집합이다. 트래픽 엔지니어링에 적합한 최단 경로 라우팅 방법은 간단하고 현재 구현된 라우팅 프로토콜에 쉽게 구현될 수 있다. 또한, 다중 경로 라우팅에 필요한 복잡한 관리 비용을 줄일 수 있다. 예를 들면, 각 라우터에서 허용되는 다중 경로의 개수는 전역 최적화를 고려하여 설정되어야 한다. 그렇지 않으면, 다중 경로를 이용하더라도 최대 링크 로드가 최소화되지 않을 수도 있다. 그리고, ECMP에서는 패킷 순서가 바뀌는 것을 방지하기 위하여 다중 경로에 트래픽을 나눌 때 목적지 라우팅 접두어(Prefix)를 기준으로 사용하는데, 각 라우팅 접두어로 향하는 트래픽의 양이 다를 수가 있기 때문에 로드 밸런싱이 쉽게 되지 않을 수도 있다.

본 논문에서 제안하는 트래픽 엔지니어링에 적합한 최단 경로 라우팅은 단일 경로를 이용하도록 하지만, 전형적인 네트워크 토폴로지와 트래픽 수요에서는 제안한 방법의 성능이 다중경로를 이용하는 최적화와 근접하다는 것을 실험으로 보였다. 또한, 트래픽 엔지니어링에 적합한 최단 경로의 집합은 다중 최단 경로들이 많이 존재할수록 효과적이다.

하지만, 다중 경로들의 개수가 많지 않은 상황에서 제안한 트래픽 엔지니어링에 적합한 최단 경로가 효과적이라는 것을 실험으로 보였다.

따라서, 본 논문에서 제안하는 트래픽 엔지니어링에 적합한 최단 경로를 사용함으로써 기존의 라우팅 프로토콜을 전혀 변경하지 않고, 라우팅 테이블을 재구성함으로써 쉽게 트래픽 엔지니어링의 목표를 달성할 수 있다. 예를 들어, 그림 1-(a)에서 간단한 네트워크 토폴로지와 트래픽 수요가 주어졌을 경우, 노드 2에서 목적지 노드 4로 향하는 넥스트-홉을 노드 1로 설정하는 임의의 최단 경로를 사용하면, 최대 링크 로드는 1.2에 이르게 된다. 그림 1-(b)에서 노드 2과 노드 4 사이에 가능한 다중 최단 경로를 모두 이용하여, 트래픽을 분배하더라도 최대 링크 로드는 1.0이다. 하지만, 그림 1-(c)에서 트래픽 엔지니어링에 적합한 최단 경로를 노드 2가 이용하게 되면, 최대 링크 로드는 0.8로 최소화될 수 있다.

본 논문에서는 그림 1과 같은 트래픽 엔지니어링에 적합한 최단 경로를 찾는 문제를 정수선형계획법(Integer Linear Programming)으로 제시하고, 이의 근사해를 찾는 알고리즘을 제안한다.

2장에서는 트래픽 엔지니어링에 적합한 최단 경로의 문제 정의를 기술하고, 3장에서 이의 근사해를 찾는 알고리즘을 설명한다. 전형적인 네트워크 토폴로지와 트래픽 수요 행렬을 이용한 실험 결과를 4장에서 설명하고, 마지막으로 5장에서는 결론과 향후 계획에 대해서 언급하기로 한다.

## II. 문제 정의

트래픽 엔지니어링에 적합한 최단 경로를 구하기 위하여 2단계로 문제를 정의한다. 첫번째 단계에서는 링크 비용값의 합을 최소화하여 최단 경로의 링크 비용값의 상한값을 구한다. 두번째 단계에서는 첫번째 단계에서 구한 링크 비용값의 상한값을 고정시켜 이를 만족하는 최단 경로들 중에서 최대 링크 로드를 최소화시킨다. 트래픽 엔지니어링에 적합한 최단 경로를 계산하는 것은 네트워크 토폴로지, 트래픽 수요 행렬, 링크 로드 측정값 등을 가지고 있는 트래픽 엔지니어링 서버에서 수행된다고 가정한다. 이러한 2단계 문제 정의는 두 개의 의존적인 최적화 목적함수를 최소화하기 위해서 각 최적화

목적함수를 분리하는 것이 최소값을 찾을 수 있기 때문이다. 두 가지 목적함수들을 부적절한 계수를 이용하여 동시에 최소화한다면 하한값을 찾지 못할 수도 있다.

다음의 상수값들이 문제정의에 사용된다.

- $G=(N, E)$  : 노드 집합,  $N$ , 과 링크 집합,  $E$ , 로 이루어진 그래프,  $G$ .
- $A=A_{sd}$  : 트래픽 요구 행렬.
- $C_{ij}$  : 링크  $(i,j)$ 의 용량.
- $w(i,j)$  : 링크  $(i,j)$ 의 비용함수 (본 논문에서는 모든 링크 비용함수의 값은 1로 지정되어 있지만, 다양한 값으로 일반화시킬 수 있다).

위와 같은 상수값들이 주어질 때, 다음의 정수값을 취하는 변수를 결정해야한다.

- $X_{ij}^{sd}$  : 송신자 노드  $s$  에서 수신자 노드  $d$  로 향하는 트래픽이 링크  $(i,j)$ 에 할당된 것을 나타냄.
- $Y_{ij}^d$  : 노드  $i$  에서 목적지 노드  $d$  로 향하는 트래픽들이 링크  $(i,j)$ 에 할당된 것을 나타냄.

최종적으로는 각 노드에서의 목적지에 대한 벡스트-흐름이 동일하게 유지되면서 모든 송신자-수신자 사이의 최단 경로로 트래픽이 할당되어 최대 링크 로드( $L_{max}$ )를 최소화하는 것이 문제의 개요이다.

첫번째 단계의 문제 정의는 다음과 같다.

$$\text{Minimize } H_{total} = \sum_{i,j,s,d} X_{ij}^{sd} \cdot w(i,j) \quad (1)$$

$$\sum_j X_{ij}^{sd} - \sum_j X_{ji}^{sd} \begin{cases} 1, & i = s, \forall i, s, d \\ 0, & i \neq s, d, \forall i, s, d \\ -1, & i = d, \forall i, s, d \end{cases} \quad (2)$$

$$\sum_{s,d} \Lambda_{sd} \cdot X_{ij}^{sd} \leq L_{max} \cdot C_{ij}, \quad \forall i, j \quad (3)$$

$$X_{ij}^{sd} \leq Y_{ij}^d, \quad \forall i, j, s, d \quad (4)$$

$$\sum_j Y_{ij}^d \leq 1, \quad \forall i, d \quad (5)$$

$$\text{Int } X_{ij}^{sd}, Y_{ij}^d \in (0,1) \quad (6)$$

첫번째 최적화 함수 (1)은 경로의 비용값 합을 최소화시키는 것이다. 제한조건 (2)는 출발지 노드, 중간 노드, 목적지 노드에서의 플로우 입출력에 관한 규칙이다. 제한조건 (3)은 링크  $(i,j)$ 에 할당되는 트래픽의 양이 링크 용량보다는 작아야한다는 것을 나타낸다. 또한, 제한조건 (4)와 (5)는 목적지 주소 기반 패킷 포워딩 조건을 설명하는 것으로 노드  $i$ 에서 목적지  $d$ 로 향하는 벡스트-흐름은  $j$  노드 하나 이하이어야 한다는 것을 표현한 것이다. 따라서, 첫번째 단계의 문제에서는 각 노드 쌍의 최단 경로 비용의 합을 구한다.

두번째 단계에서의 문제는 첫번째 단계에서 구한 최단 경로 비용의 합을 만족하는 후보들 중에서 가장 작은 최대 링크 로드 값을 가지는 것을 선택하게 한다. 두번째 단계의 문제 정의는 다음과 같다.

$$\sum_j X_{ij}^{sd} - \sum_j X_{ji}^{sd} \begin{cases} 1, & i = s, \forall i, s, d \\ 0, & i \neq s, d, \forall i, s, d \\ -1, & i = d, \forall i, s, d \end{cases} \quad (8)$$

$$\sum_{s,d} \Lambda_{sd} \cdot X_{ij}^{sd} \leq L_{max} \cdot C_{ij}, \quad \forall i, j \quad (9)$$

$$X_{ij}^{sd} \leq Y_{ij}^d, \quad \forall i, j, s, d \quad (10)$$

$$\sum_j Y_{ij}^d \leq 1, \quad \forall i, d \quad (11)$$

$$\sum_{i,j,s,d} X_{ij}^{sd} \cdot w(i,j) \leq H_{total} \quad (12)$$

$$\text{Int } X_{ij}^{sd}, Y_{ij}^d \in (0,1) \quad (13)$$

제한조건 (12)이 추가되어 각 경로가 목적지 기반 패킷 포워딩 조건을 만족하는 최단경로를 찾게 된다. 이러한 모든 경로들 중에서 최적화 함수 (7)에 의해서 최대 링크 로드가 최소화된다.

위의 문제는 정수 변수를 포함하기 때문에, 네트워크 규모가 커지게 되면 문제 해결이 쉽지 않게 된다. 목적지 기반 포워딩 조건을 삭제한 문제가 *NP-hard* 라는 것이 [14]에서 증명되었고, 각 노드 사이의 최단 경로의 개수( $P_{sd}$ )가 알려져 있을 경우에도, 전체 문제의 공간은  $\prod_{i=1}^N \prod_{j=1, j \neq i}^N |P_{sd}|$ 로 지

수적으로 분포하게 된다. 따라서, 다음 절에서 복잡한 ILP 문제 해결을 위하여 간단한 휴리스틱을 제안한다.

### III. 트래픽 엔지니어링을 위한 최단 경로 라우팅 알고리즘

제안하는 휴리스틱 알고리즘은 우선 각 노드 사이에서 존재하는 최단 경로들을 이용하여, 최대 링크 로드(Lmax)가 최소화되는 경로의 집합을 찾는다.

- [1 단계] 모든 노드 (s,d) 사이의 다중 최단 경로들을 구한다.
- [2 단계] 모든 노드 (s,d) 사이의 목적지 기반 포워딩 조건을 만족하는 최단 경로를 하나를 선택하여 초기화한다.
- [3 단계] 최대 링크 로드(Lmax) 값을 가지는 가장 혼잡한 링크(Lc)를 찾고, 이 링크를 거치는 트래픽 플로우들의 집합 {Fi} 을 찾는다.
- [4 단계] 각 플로우, Fi, 에 대하여 현재 최단 경로를 사용하지 않는 다른 최단 경로들로 변경한다. 이에 따라, 자동적으로 최단 경로가 변경되어야 하는 유도 플로우들을 구한다. 이는 목적지 기반 포워딩 규칙을 만족시켜준다.
- [5 단계] 선택된 Fi 플로우와 유도 플로우들의 최단 경로들을 변경해서 최대 링크 로드 값이 줄어들게 되면 최단 경로를 변경한다.
- [6 단계] Fi 플로우를 검사되었다고 표시한다.
- [7 단계] 경로를 변경한 후 새롭게 계산한 최대 혼잡 링크가 이전 링크(Lc)와 다르다면, [3 단계]로 간다.
- [8 단계] {Fi}에 검사할 플로우들이 남아있다면, [4 단계]로 가고, 그렇지 않다면, 종료한다.

각 노드에서 목적지 기반 포워딩 규칙을 만족시키기 위하여, [4 단계]에서 특정 플로우의 현재 최단 경로를 다른 경로로 바꿀 때 자동적으로 최단 경로들이 바뀌게 되는 유도 플로우들을 찾는다. 이러한 유도 플로우들은 그림 2에서처럼 분기되는 부분 최단 경로(branched-sub-shortest path)를 지나가는 모든 플로우들을 찾으면 된다. 그리고, [4 단계]에서 가장 혼잡한 링크를 지나가는 어떤 플로우를 선택할 지, 그리고, 각 플로우의 어떤 가용한 최단 경로를 선택하는지에 따라서 다양한 알고리즘이 가

능하게 된다. 하지만, 이러한 선택 방법은 항상 유도 플로우를 생성하기 때문에 선택의 방법에 따른 차이가 성능에는 크게 영향을 끼치지 않는다. 본 논문에서는 임의의 플로우를 선택하고, 임의의 가용 최단 경로를 선택하는 방법을 이용하였다.

제안한 알고리즘의 계산 복잡도는  $O(|M|^3 + |E||M|(|M - 1)^2)$  이다. 각 노드 사이의 다중 최단 경로를 찾는 단계에서 선형Dijkstra 알고리즘의 복잡도는  $O(|M|^3)$ 이다. 최대 혼잡 링크( $O(|E|)$ )에 대하여 지나가는 플로우들 ( $O(|M|(|M-1))$ )과 이들의 유도 플로우들 ( $O(|M-1|)$ )에 대해서 비교해 보는 복잡도가 추가된다.

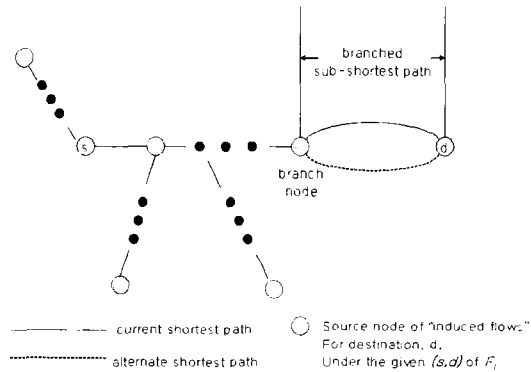


그림 2. 유도 플로우

### IV. 성능평가

#### 4.1 실험 1: 대표적인 샘플 네트워크와 트래픽 요구 행렬

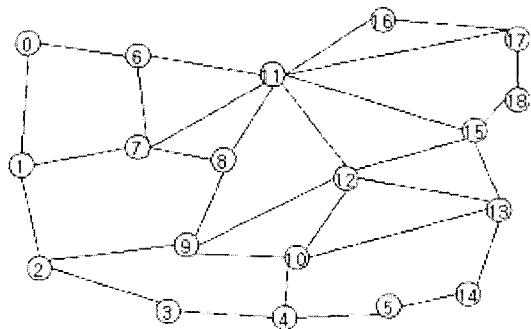


그림 3. 샘플 네트워크 토폴로지

그림 3과 같이 19개의 노드와 62개의 방향성 링크로 연결된 샘플 네트워크 토폴로지에 [15]에서 사용한 1시간마다 측정된 24개의 샘플 트래픽 수요 행렬을 이용하여 실험을 하였다. 트래픽의 잦은 변동

은 트래픽의 측정 구간을 길게 하여 평균화된다고 가정한다. 각 링크의 용량은 310 Mbps라고 가정하였다.

#### 4.1.1 트래픽 엔지니어링을 위한 동적인 최단 경로 라우팅

1시간마다 측정된 링크 로드 에 대해서, 트래픽 엔지니어링에 적합한 최단 경로를 찾아서 라우팅 테이블을 변경하는 것을 실험하였다. 그림 4에서 임의의 최단 경로(Default SP)를 사용하는 것과 2개의 다중 최단 경로를 이용하는 ECMP보다 제안한 휴리스틱 알고리즘이 최대 링크 로드를 평균 각각 25%와 8.5% 정도 감소시켰다. CPLEX [16]에 의한 단일 경로를 이용한 최적해와 비교할 때에는 4.9% 정도 차이가 발생하였다. 모든 정수 변수를 실수화 시켜서 선형계획법(LP: Linear Programming)으로 문제를 해결한 최적 다중 경로 라우팅과 비교할 때에도 6.6% 정도 최대 링크 로드가 증가하였다. 이는 실험된 네트워크 토폴로지와 트래픽 요구 행렬에 대해서는 최적의 단일 경로 라우팅 방법(Optimal Single SP)과 다중 경로 라우팅(Optimal Multi-path)과 큰 차이가 나지 않기 때문이었다. 이 네트워크에서는 평균 1.56 개 정도의 최단 경로가 각 노드 쌍에 존재하는데, 이러한 최단 경로들을 잘 활용하면, 복잡한 최적의 다중 경로 라우팅에 비해서 간단한 최단 경로를 구할 수 있다는 것을 암시한다.

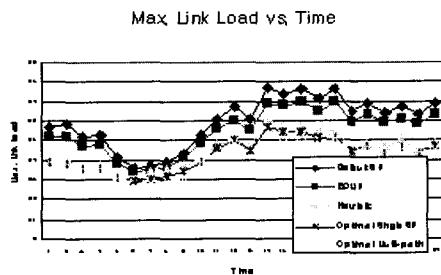


그림 4. 트래픽 엔지니어링에 적합한 동적인 최단 경로 라우팅

이러한 트래픽 엔지니어링에 적합한 최단 경로를 각 IP 라우터에서 활용하기 위해서는, 각 목적지 라우팅 프레픽스에 대한 넥스트-홉을 변경해야 할 수도 있다. 즉, 임의의 최단 경로를 이용하는 라우팅 테이블과 다른 넥스트-홉을 변경해야 한다. 이러한 라우팅 테이블의 변경 횟수는 제안한 휴리스틱을 이용할 때의 설정 오버헤드가 된다. 라우팅 프레픽스를 변경하는 것은 다중 최단 경로의 개수를 임시로

2개로 설정하여 새로운 넥스트-홉을 설정하고, 이전의 넥스트-홉을 삭제한다. 그림 5와 그림 6에서 각 트래픽 엔지니어링 구간에서 설정되는 라우팅 테이블의 변경 횟수와 라우팅 테이블을 변경하는 노드의 수를 나타내었다.

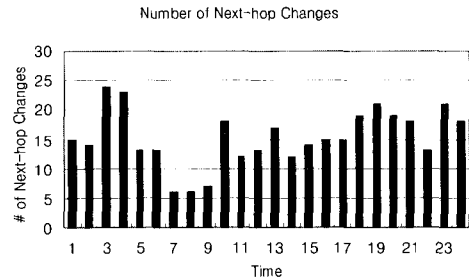


그림 5. 라우팅 테이블의 변경 횟수

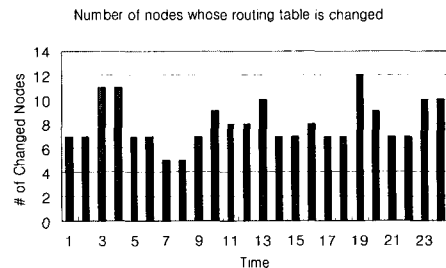


그림 6. 라우팅 테이블을 변경하는 노드의 수

이 실험에서는 모든 가능한 라우팅 테이블의 수에 대한 변경횟수의 % (= 넥스트-홉 변경횟수 / (N(N-1) \* 100) )는 평균 4.5% 정도이고, 9개의 노드에서 라우팅 테이블 변경이 필요하였다. 평균적으로 하나의 노드에서 2개의 라우팅 테이블의 변경이 필요하였다.

#### 4.1.2 트래픽 엔지니어링을 위한 정적인 최단 경로 라우팅

트래픽 엔지니어링이 수행되는 시점에서 매번 라우팅 테이블을 변경하지 않고, 실험에서 사용된 1일 트래픽 패턴과 같이 알려져 있는 트래픽 요구 행렬에 대해서는 대표적인 트래픽 요구 행렬을 이용하여 라우팅 테이블을 고정하여 트래픽 엔지니어링을 수행할 수도 있다.

그림 7에서 24개의 샘플 트래픽 요구 행렬 중에서 동적인 라우팅 테이블 변경방법의 최대 링크 로드와 가까운 결과를 보이는 첫번째 트래픽 요구 행렬을 사용한 결과와 최악의 경우를 보여주고 있다. 첫번째 트래픽 요구 행렬에 대한 라우팅 테이블을

고정하여 사용하였을 경우 평균 1.7 % 정도 증가한 최대 링크 로드 값을 보여주는 반면, 7번째 최악의 경우는 평균 26.5 % 증가하였다. 이 때 ECMP와 비교하면, 최대 링크 값이 증가한 것을 보여주지만, 임의의 최단 경로 라우팅의 경우보다는 여전히 향상된 결과를 보여주었다.

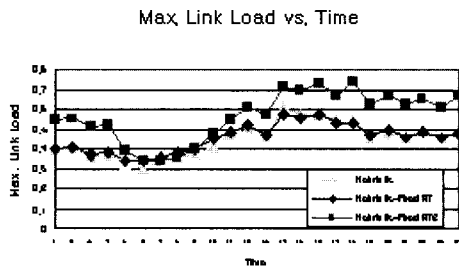


그림 7. 고정된 라우팅 테이블을 사용한 경우

즉, 대표적인 트래픽 요구 행렬에 대한 고정된 라우팅 테이블을 사용하였을 경우에도 임의의 최단 경로를 이용하는 것보다 최대 링크 로드 값을 줄일 수 있다는 것을 알 수 있다.

4.1.3 단일 링크 결함에 의한 영향

단일 링크의 결함에 의한 영향을 조사하기 위하여, 각 링크의 결함에 따른 네트워크 토폴로지에서 24개의 트래픽 요구 행렬들을 수행한 최대 링크 로드 값을 그림 8에서 보여주고 있다.

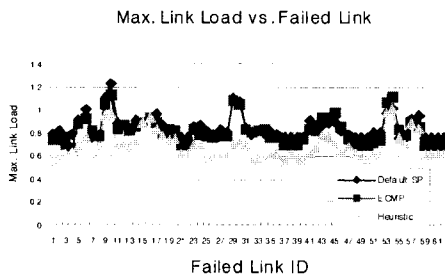


그림 8. 링크 결함에 의한 영향

각 링크의 결함의 최악의 경우를 대비하기 위해서는 이 네트워크의 용량은 적어도 대략 1.3 배 정도 증설되어야한다. 즉, 링크 9의 결함으로 인한 임의의 최단 경로에서 최대 링크 로드 값이 급격하게 증가하였음을 볼 수 있다. ECMP를 이용하는 경우는 몇몇 링크의 결함에 대해서는 임의의 최단 경로보다 최대 링크 로드값을 증가시키는 것을 볼 수 있다. 이는 전체 트래픽 요구 행렬을 고려하지 않고, 트래픽을 분할하였기 때문이다. 모든 링크의 결

함에 대해서 본 논문에서 제안한 트래픽 엔지니어링에 적합한 최단 경로의 라우팅 테이블이 최대 링크 로드를 줄여주었다.

4.2 실험 II: 랜덤 네트워크와 트래픽 요구 행렬

임의의 네트워크 토폴로지와 트래픽 요구 행렬을 생성시켜 제안한 알고리즘의 성능과 임의의 최단 경로 라우팅, ECMP 및 최적 다중 경로 라우팅의 성능과 비교하였다(최적 단일 경로 라우팅의 해는 ILP으로 해결해야 하는데, 네트워크 규모가 커지게 되면 적절한 시간내에 종료하지 않기 때문에 생략하였다.). 네트워크의 노드 개수는 {10, 20, 30, 40, 50}으로 2차원 평면에 임의로 분포되고, 각 노드에 대해서 {30, 60, 90, 120, 150} 개의 링크를 임의로 연결하였다. 이 실험 네트워크에서 노드의 차수는 라우터의 인터페이스가 평균 3개 정도라는 것을 가정하였다. 각 링크의 용량은 [150, 310] Mbps 사이에 분포하도록 하였고, 각 노드 사이의 트래픽 요구는 [0, 10] Mbps 로 분포하였다.

그림 9는 각 네트워크에 대해서 30번의 실험을 한 평균값을 그린 것이다. 임의의 최단 경로 방법을 이용할 때에는 대부분의 경우 최대 링크 로드 값이 1을 초과하였다. 이는 특정 링크에서의 혼잡현상으로 인해 트래픽이 손실된다는 것을 나타낸다. ECMP는 최대 링크 로드 값을 평균 29 % 정도 향상시키지만, 제안한 알고리즘은 평균 55 % 정도 향상시킨다. 최적의 다중 경로 라우팅과 비교하면 평균 19 % 정도 차이가 발생하였다.

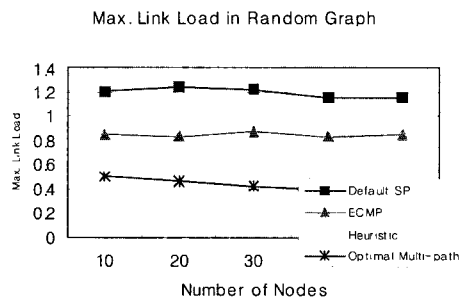


그림 9. 임의의 네트워크에서의 실험

네트워크에서 사용가능한 각 노드 쌍의 최단 경로 개수가 증가할수록 제안한 알고리즘의 성능이 개선된다는 것을 그림 10에서의 평균 최단 경로 개수를 통해서 확인할 수 있다. 그림 11에서는 임의의 최단 경로 라우팅을 이용할 때와 라우팅 테이블의 차이는 평균 20 % 정도라는 것을 알 수 있다.

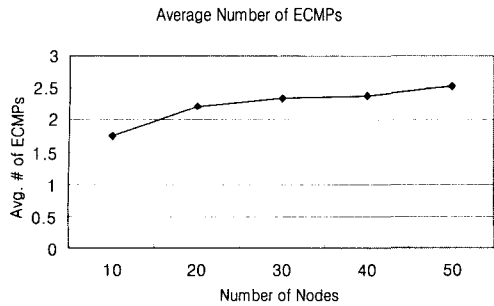


그림 10. 다중 최단 경로 개수

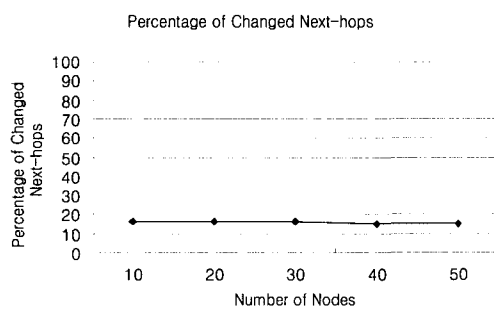


그림 11. 라우팅 테이블의 변경횟수

### V. 결 론

본 논문에서는 트래픽 엔지니어링에 적합한 최단 경로를 찾는 문제를 ILP로 제시하여 최적해를 구하도록 하였고, 이 문제를 빠르게 해결하기 위하여 휴리스틱 알고리즘을 제안하여 성능을 평가하였다. IP 라우터에 각 목적지 라우팅 프레임에 대한 최단 경로를 선택할 때, 임의의 방식을 선택하는 대신 제안한 알고리즘이 찾은 최단 경로에 대한 넥스트-홉을 선택함으로써 최대 링크 로드를 감소시킬 수가 있었다. 제안한 트래픽 엔지니어링에 적합한 최단 경로를 이용하기 위한 설정 오버헤드는 라우팅 테이블을 변경하는데 드는 비용은 일반적인 네트워크에서는 그리 크지 않다는 것을 실험으로 보였다. 특히, 고정된 라우팅 테이블을 이용하여 라우팅 테이블 변경 비용을 없애더라도 임의의 최단 경로를 사용하는 경우보다 성능이 개선된다는 것을 보였다. 또한, 링크 결합에 의한 혼잡현상이 발생하는 경우에도 제안한 라우팅 방법이 최대 링크 로드를 감소시킨다는 것을 확인하였다.

### 참 고 문 헌

- [1] D. Awduche, A. Chiu, A. Elwalid and X. Xiao, "Overview and Principles of Internet Traffic Engineering," IETF RFC3272, 2002
- [2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS," IETF RFC2702, 1999
- [3] D. Bertsekas, and R. Gallager, Data Networks, Prentice Hall, 1992
- [4] R. Ahuja, T. Magnanti, and J. Orlin, Network Flows, Prentice Hall, 1993.
- [5] S. Vutukury and J. J. Garcia-Luna-Aceves, "A Traffic Engineering Approach based on Minimum Delay Routing," ICCCN'2000, 2000
- [6] J. Moy, "OSPF Version 2," IETF RFC2328, 1998
- [7] R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments," IETF RFC1195, 1990
- [8] Optimized Multipath, <http://www.fictitious.org/omp>
- [9] Z. Wang, Y. Wang, and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," IEEE INFOCOM, 2001
- [10] B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," IEEE INFOCOM, 2000.
- [11] A. Sridharan, R. Guerin and C. Diot, "Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks," IEEE INFOCOM, 2003
- [12] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF RFC3031, 2001
- [13] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF RFC 3209, 2001
- [14] Y. Wang, and Z. Wang, Explicit Routing Algorithms for Internet Traffic Engineering, IEEE ICCCN, 1999
- [15] A. Gencata and B. Mukherjee, "Virtual-



Topology Adaptation for WDM Mesh  
Networks Under Dynamic Traffic," IEEE  
INFOCOM, 2002

[16] CPLEX, <http://www.ilog.com>

이 영 석(Youngseok Lee)

정회원



1990.3 ~ 1995.2: 서울대학교

컴퓨터공학과(학사)

1995.3 ~ 1997.2: 서울대학교

컴퓨터공학과(석사)

1997.3 ~ 2002.8: 서울대학교

컴퓨터공학부(박사)

2002.10 ~ 2003.7: University of California, Davis  
방문연구원

2003.7 ~ 현재: 충남대학교 전기정보통신공학부 전  
임강사

<주관심 분야> 차세대 인터넷, MPLS, 트래픽 엔지  
니어링, WDM