

통계적 기법을 이용한 경로 선택 알고리즘

(A Route Selection Algorithm using a Statistical Approach)

김 영 민 [†] 안 상 현 ^{**}
(Youngmin Kim) (Sanghyun Ahn)

요 약 현재 사용중인 경로 선택 방법은 최단 경로 알고리즘을 이용하므로 망 자원을 효율적으로 이용하지 못하며 특정 경로로 트래픽이 집중될 경우 혼잡(congestion) 상황을 발생시킬 수 있다. 본 논문에서는 새롭게 요청되는 연결 설정 요구에 대해 요청된 대역폭을 충족시킬 수 있는 경로를 결정하는데 있어서 통계적 기법을 이용함으로써 망을 효율적으로 사용하며, 혼잡 상황을 줄일 수 있는 통계적 경로 선택(Statistical Route Selection; SRS) 알고리즘을 제안한다. MPLS의 등장으로 부하 균등화(load balancing)에 필요한 명시적인(explicit) LSP 설정을 할 수 있게 되었으며, MPLS의 LSP를 설정하기 위해 SRS 알고리즘을 이용할 수 있다. SRS 알고리즘은 경로 선택을 위해 링크들의 이용률을 구하고, 통계적인 기법을 사용하여 가중치를 결정하며, 그 가중치를 이용한 최단 경로를 구한다. 여기서 사용되는 통계적 기법은 링크들의 이용률의 평균과 분산을 이용하는 것으로, 이 정보를 기반으로 링크의 가중치에 대해 분산을 작게 하는 방향으로 경로를 결정함으로써 부하 균등화 효과를 얻게 되어 혼잡 링크 수를 줄이고, 망 자원 이용률을 높인다. 실험을 통해 다른 경로 선택 알고리즘들에 비해 SRS 알고리즘이 망 자원을 효율적으로 이용하여 연결 설정 실패의 수와 혼잡 링크의 수를 줄이는 것을 보인다.

키워드 : 트래픽 공학, 부하 균등화, 통계적 기법, 경로 선택, MPLS

Abstract Since most of the current route selection algorithms use the shortest path algorithm, network resources can not be efficiently used also traffics be concentrated on specific paths resulting in congestion. In this paper we propose the statistical route selection(SRS) algorithm which adopts a statistical mechanism to utilize the network resource efficiently and to avoid congestion. The SRS algorithm handles requests on demand and chooses a path that meets the requested bandwidth. With the advent of the MPLS it becomes possible to establish an explicit LSP which can be used for traffic load balancing. The SRS algorithm finds a set of link utilizations for route selection, computes link weights using statistical mechanism and finds the shortest path from the weights. Our statistical mechanism computes the mean and the variance of link utilizations and selects a route such that it can reduce the variance and the number of congested links and increase the utilization of network resources. Throughout the simulation, we show that the SRS algorithm performs better than other route selection algorithms on several metrics like the number of connection setup failures and the number of congested links.

Key words : Traffic Engineering, Load Balancing, Statistical Mechanism, Route Selection, MPLS

1. 서 론

최선의(best-effort) 서비스만을 제공하는 기존의 IP 기술로는 VoIP(Voice over IP)나 주식거래, 경매 시스템, 멀티미디어 데이터와 같은 QoS를 요구하는 서비스들을

지원하기 어렵다. 현재 이러한 IP의 QoS 문제를 해결하기 위해 IETF를 중심으로 많은 연구가 수행되고 있으며, 이와 관련하여 본 논문에서는 트래픽 공학(Traffic Engineering; TE)적 측면에서 경로 선택 알고리즘을 개선하여 트래픽을 적절하게 분배하는 방법을 제시한다.

현재 IP 라우팅 방식에서는 최단 경로 라우팅(shortest path routing)을 사용하고 있으나, 이 경우 특정 경로로 트래픽이 집중되는 문제가 발생할 수 있다. 이러한 문제를 해결하기 위해 TE가 요구되며, TE를 효율적으로 이용하여 트래픽을 적절히 분배하면 사용자들의 요구사항

[†] 비 회 원 : 이코테크(주) 연구소 초고속사업부 연구원
ymkim92@venus.uos.ac.kr

^{**} 종신회원 : 서울시립대학교 컴퓨터·통계학과 교수
(corresponding author)
ahn@venus.uos.ac.kr

논문접수 : 2001년 1월 4일

심사완료 : 2001년 11월 22일

에 적합한 서비스를 효율적으로 제공할 수 있다. QoS 라우팅은 대역폭이나 지연(delay)과 같은 QoS 제약조건을 충족시킬 수 있는 라우팅을 의미하며, 망에서 자동적으로 TE를 처리하도록 하는 방법으로 사용될 수 있다. 즉, QoS 라우팅과 TE는 상호 보완적인 관계를 갖는다.

가용 대역폭 내에서 효율적인 망 운영을 목적으로 하는 위의 방법들과는 좀 다른 방향으로 대역폭을 크게 늘릴 수 있는 파장 분할 다중화(Wavelength-Division Multiplexing; WDM) 방식이 연구되고 있다. WDM 방식을 이용하면 QoS 문제가 저절로 해결될 것이라는 의견도 있으나, 대역폭이 늘어나면 그에 따라 많은 대역폭을 사용하는 응용들이 등장할 것이고 결국 QoS 문제가 다시 나타나게 될 것이다. 따라서 대역폭을 크게 늘린다 하더라도 망을 효율적으로 운영하기 위한 방법은 필요하다.

서비스 수준(level)을 정의하고 자원을 예약하는 일련의 기법들을 사용해도 가용 대역폭이 없으면 QoS 제공이 불가능하며, 따라서 연결을 설정할 때마다 트래픽을 적절히 분배하여 새로 요청되는 QoS 요구사항에 대비해야 할 필요가 있다.

TE를 구현하기 위해서는 더 많은 정보들이 라우터들 사이에 교환되어야 하고 복잡한 알고리즘을 수행해야 하므로 비용이 증가한다. 그러나, 링크 수가 많은 잘 연결된 망에서는 TE를 통해 이득을 얻을 수 있으며 5장에서 실험을 통해 이를 입증한다.

본 논문에서는 트래픽을 미리 예측할 수 없는 일반적인 상황에서, 도메인 내의 진입노드와 진출노드 사이에 경로를 설정할 때, 대역폭을 보장하면서 망을 효율적으로 사용하는 라우팅 경로 설정 방법을 제시한다. 즉, 라우팅 경로를 구할 때 필요한 각 링크들의 가중치를 적절하게 선정함으로써, 링크 이용률들의 분산을 최소화하는 새로운 경로 선택 알고리즘인 SRS(Statistical Route Selection) 알고리즘을 제안한다. 이렇게 구해진 경로들은 최근 각광받고 있는 MPLS[1]의 명시적인(explicit) LSP를 설정하는데 사용될 수 있다.

2장에서는 TE와 관련된 고려사항들에 대해서, 3장에서는 관련 연구들에 대해 알아본다. 4장에서는 본 논문에서 새롭게 제안한 경로 선택 방법인 SRS에 대해 살펴보고, 5장에서는 실험 결과를 보이고, 6장에서 결론을 맺는다.

2. 고려 사항

이번 장에서는 TE를 수행하기 위해 필요한 고려 사항으로, 단일 경로를 사용할 것인지 다중 경로를 사용

할 것인지, 트래픽 요구를 미리 알고 있다는 가정을 이용할 것인지, 메트릭(metric)으로는 어느 것을 사용할 것인지 등을 중심으로 살펴본다.

2.1 단일 경로 대 다중 경로

일반적인 라우팅에서는 단일 경로를 이용하지만, 근원지와 목적지간에 비용이 같은 최단 경로가 둘 이상 존재한다면 이들을 모두 이용하여 부하를 분배할 수 있다. ECMP[2]에서는 이렇게 구해진 다중 경로에 트래픽을 동일하게 분배하며, OMP[3]에서는 전체적인 망의 여건을 고려하여 각 경로마다 다른 양의 트래픽을 나누어 분배함으로써 ECMP보다 부하를 더 잘 분배한다. OMP에서는 다중 경로를 이용하기 때문에 한 경로에 문제가 발생했을 때 나머지 경로들을 이용해 문제가 발생한 경로의 트래픽을 우회시킬 수 있으나, 알고리즘이 복잡하고 망의 부하가 일정하지 않기 때문에 부하를 다중 경로상에 효과적으로 분배하는데 어려움이 있으며, 트래픽이 한 곳으로 집중될 수 있다. ECMP나 OMP를 이용해 부하를 잘 분배하더라도 전송계층에서의 문제는 여전히 존재한다. 그 예로 다중 경로를 이용하기 때문에 각 경로마다 다르게 측정되는 RTT(Round Trip Time)을 결정하는 문제가 있으며, 각 경로마다 전송속도가 다르기 때문에 패킷의 전달순서가 바뀔 수 있고, 이로 인해 더 많은 버퍼가 요구된다. 또한 경로마다 상이한 최대 패킷 크기(MTU)를 결정하는 문제도 발생한다 [4]. 한 세션에 대해서 모든 트래픽을 같은 경로로 전달하면 이러한 문제를 해결할 수도 있지만, 이렇게 되면 멀티미디어 데이터와 같은 대량의 트래픽에 대해서는 다중 경로를 사용하는 이득을 얻을 수 없다. 따라서, 본 논문에서는 문제의 단순화를 위해 단일 경로를 사용하는 TE 방식을 제시한다.

2.2 On-line 대 Off-line

대부분의 TE 방식들은 모든 진입노드와 진출노드 사이에 트래픽 요구가 이미 주어져 있다는 가정을 사용한다. 이러한 가정은 망 설계과정에서 유효하게 사용될 수 있으나, 일반적인 인터넷 환경에는 적합하지 않다. 모든 트래픽 요구들이 미리 알려져 있다고 가정하는 off-line 모델은 현재 주어진 조건하에서 망을 가능한 한 효율적으로 사용하는 것을 목적으로 한다. 하지만, 이런 목적으로 인해서 라우팅 경로가 모두 결정된 후에 임의의 진입노드와 진출노드 사이에 이용할 수 있는 대역폭이 남아 있지 않을 수도 있다. 즉, 이후의 연결 요청에 대한 고려 없이 현재 off-line에서 이용할 수 있는 연결 요청만을 위해 최적화되므로 새로운 연결 설정 요구가 들어왔을 때 현재 상태로는 대역폭이 부족하여, 기존에

설정되어 있던 경로들을 다시 라우팅해야 하는 상황이 일어날 수 있다. 인터넷 환경에서는 새로운 연결 설정 요구가 들어올 수 있다는 가정을 배제할 수 없고, 이미 설정된 경로를 다시 라우팅하는 부담을 덜기 위해서는 망 상황에 탄력적으로 대응할 수 있는 on-line 모델이 더 적합한 모델일 것이다. 따라서, 본 연구에서는 on-line 모델을 가정한다. On-line 모델에 관한 최근 연구로는 [5], off-line 모델에 기반한 연구로는 [6]이 있다.

On-line 모델은 망의 트래픽 상황에 따라 링크 비용을 주기적으로 또는 특정 사건(event)이 발생한 시간에 변화시켜서 트래픽을 분배하며, 따라서 기본적인 라우팅을 위한 정보에 추가적으로 각 링크들의 가용 대역폭에 대한 정보를 필요로 하는 단점을 지닌다.

2.3 단일 메트릭 대 다중 메트릭

대역폭, 지연, 지터(jitter) 등 여러 개의 메트릭을 이용하면 좀 더 최적화된 경로를 구할 수 있지만 이를 위해 라우터들이 더 많은 정보를 수집해야 하고, 이렇게 수집된 정보들을 다른 라우터로 전달하기 위해 추가적인 대역폭을 낭비하게 된다. 또한 더 많은 정보를 이용하므로 계산에 드는 비용도 커지게 된다.

다중 메트릭으로 사용할 수 있는 대표적인 메트릭으로는 대역폭과 지연이 있으나, 지연은 대역폭과 어느 정도의 상관관계를 지닌다. 위성통신과 같은 경우를 제외한 일반적인 경우, 대역폭이 커지면 지연이 작아지며, 대역폭이 작아지면 지연은 커진다. 따라서 경로 설정과 TE에 대역폭만을 이용하더라도 다중 메트릭을 사용한 효과를 얻을 수 있다

3. 관련 연구

[7]에서는 여러 가지 경로 설정 알고리즘들에 대한 비교 연구결과를 제시했다. 연결 설정을 위해 가장 일반적으로 사용되는 방법은 SP(Shortest Path) 알고리즘에 의한 것으로 이 방법은 단순하며 최소 홉을 사용하기 때문에 자원을 절감할 수 있다. 다음으로 최단 경로 알고리즘에 약간의 기능을 추가한 방법으로 WSP(Widest-Shortest Path)가 있다. 이 방법은 최단 경로를 구할 때, 한 지점에서 같은 비용을 갖는 둘 이상의 링크를 만났을 경우 대역폭이 큰 링크를 선택하는 방법으로 진입노드와 진출노드 사이에 같은 최소 홉을 갖는 경로가 여러 개 존재할 때, 해당 경로를 구성하는 링크의 대역폭의 최소 값이 가장 큰 경로를 선택하는 방법이다. 이 방법은 SP의 모든 특성을 반영하면서 대역폭이 큰 경로를 사용하므로 어느 정도의 트래픽 분배도 할 수 있다는 장점이 있다. SWP(Shortest-Widest Path)는 대역폭이 가장 큰

경로 중 최단 경로를 선택하며, 트래픽 분배를 최우선 목표로 함으로써 자원 이용량이 너무 커지는 단점이 있다. 라우팅의 단순함을 유지하면서 좋은 성능을 낼 수 있는 방법으로 SDP (Shortest-Distance Path) 알고리즘이 있다. 이 알고리즘에서 거리(distance)는 가용 대역폭의 역수라 의미하며, 거리를 최소로 하는 경로를 선택한다.

$$dist(P) = \sum_{i=1}^k \frac{1}{r_i}, r_i \text{ 는 가용 대역폭} \quad (1)$$

위의 4가지 방법의 성능을 비교한 결과, SDP가 전반적으로 가장 좋은 성능을 가지며, SWP는 가장 낮은 성능을 보이므로 5장의 실험결과와의 비교 대상에서 제외시켰다[7].

TE의 목적은 망 자원 이용률과 트래픽 성능(traffic performance)을 동시에 최적화시킬 수 있는 효율적이고 신뢰할 수 있는 망을 구성하는 것이다[8]. 그러나 TE 요구사항들은 서로 배타적이어서 지연을 최소화시키기 위해 최단 경로만을 이용한다면, 최단 경로에 포함되는 링크들이 높은 빈도로 사용되어 망 자원 이용률이 불균형을 이루게 되며, 따라서 추후에 QoS를 요구하는 요청을 만족시키지 못하는 상황이 발생할 수 있다. 또한, 망 자원 이용률을 높이기 위해 우회경로를 이용한다면 경로를 구성하는 링크들의 수가 늘어나 그만큼 트래픽 성능을 감소시킬 수 있다. 따라서 본 연구에서는 TE 요구사항들에 대한 절충점, 즉, 부하 균등화를 적절히 수행해 망 자원 이용률도 높으면서 트래픽 성능을 높일 수 있는 경로 설정 방법을 찾고자 한다.

TE에 관한 최근의 연구로 2장에서도 언급한 MIRA [5]가 있다. MIRA는 MPLS 환경을 위주로 하여 방해(Interference)를 최소화 하는 경로를 찾는 것을 목적으로 하며, 진입노드와 진출노드의 정보를 최대한 활용하여 각 링크들의 가중치를 구한 후 링크들 중에 요청된 대역폭을 만족하지 못하는 링크들을 제외한 망에서 가중치를 최소화 하는 최단 경로를 선택한다.

4. SRS(Statistical Route Selection) 알고리즘

한 링크의 가용 대역폭은 그 링크의 총 대역폭에서 현재 사용중인 대역폭을 뺀 나머지 용량을 뜻하며, 진입노드와 진출노드 사이의 경로 설정시 요구되는 대역폭을 수용할 수 있는 가용 대역폭을 가진 링크를 가용 링크라고 하고, 도메인 내의 모든 라우터들과 가용 링크로 이루어진 망을 이용하여 실질적인 라우팅이 이루어진다. 이렇게 하면 대역폭을 보장하는 라우팅이 가능해진다.

본 연구에서는 링크 이용률의 분산을 최소화하여 혼잡 상황을 최대한 적게 일으키는 경로 선택 알고리즘을

제시함으로써 부하 균등화 라우팅을 한다. 분산은 각 데이터들이 그들의 평균에서 얼마만큼 멀리 떨어져 있는가를 나타내며, 분산 값이 작다는 것은 모든 링크들이 고르게 사용된다는 것을 가리키는 척도가 될 수 있다. 따라서 링크들의 이용률의 분산을 최소화하도록 트래픽을 분배하는 것을 목적으로 하며 식 (2)를 목적함수로 한다:

$minimize(V(X))$, X 는 링크들의 이용률의 집합 (2)

분산을 최소화하기 위해 링크 이용률들의 평균에서 멀리 떨어진 정도를 이용하여 링크의 가중치에 차등을 둔다. 즉, 이용률의 값이 평균보다 작을수록 링크의 가중치를 작게 하고 클수록 가중치를 크게 하여 이용률이 작은 쪽의 링크가 경로에 포함될 확률을 높이는 방법을 제시한다.

4.1 링크 이용률 구하기

가중치를 변경하지 않고 고정된 값을 사용하는 알고리즘(SP, WSP, SWP)의 경우 가중치를 변경하기 위한 정보를 유지할 필요가 없으며, 이 정보를 다른 라우터들에게 전달하는 비용도 없다. 하지만 TF의 목적중의 하나인 망 자원 이용률을 높이기 위해서는 현재 망의 상황을 반영할 수 있는 방법이 필요하며, 이를 위해 트래픽 상황을 고려하여 가중치를 변경한 후 변경된 가중치에 의해 최단 경로를 구하게 된다. 이때 더 많은 여분의 대역폭을 가진 링크를 새로운 경로에 포함시킬 수 있도록 가중치를 결정한다.

본 논문에서 제시하는 가중치를 결정하는 알고리즘은, 각 링크들의 이용률을 측정하여 그들의 평균과 분산을 구하고 새로운 연결 설정 요청에 대해 링크 이용률들의 분산을 줄이는 방향으로 가중치를 결정한다. 가중치를 구하는데 필요한 링크 이용률을 구하는 방법은 다음과 같다:

link n 의 현재 이용률: $Util(L_n)$ (3)

m 번째 측정된 link n 의 이용률: (4)

$NowUtil(L_n)_m = (RequestedBandwidth + UsedLinkBandwidth) / TotalLinkBandwidth$

m 번째 측정시의 $Util(L_n)$: (5)

$Util(L_n)_m = \alpha * Util(L_n)_{m-1} + (1 - \alpha) * NowUtil(L_n)_m$,

$(0 \leq \alpha \leq 1)$

식 (3)은 가장 최근의 상태, 즉 실질적으로 라우팅 정보로 사용되는 링크의 이용률을 나타내며, 식 (4)는 시간에 따라 또는 트래픽의 요청 상태에 따라 링크의 이용률이 변할 때, m 번째에 측정된 링크의 이용률을 나타내는 것으로 새로운 진입노드와 진출노드 사이의 연결 설정시 요구되는 대역폭($RequestedBandwidth$)을 포함

시켜서 대역폭이 작은 링크에게 대역폭이 큰 링크보다 상대적으로 더 높은 이용률을 부여하여 대역폭이 큰 링크가 더 많이 이용되도록 한다. 그림 1의 경우 (a), (b) 두 개의 링크가 있으며 (a)는 대역폭이 10이고 현재까지 3의 대역폭이 사용되어 0.3의 이용률을 가지고 있고 (b)는 대역폭이 5이며 아직 사용된 대역폭이 없어 0의 이용률을 가진다. 이 상태에서 대역폭 4를 보장해야 하는 새로운 연결 요청이 들어오게 되면 0의 이용률을 가진 (b)가 더 낮은 이용률 값을 갖기 때문에 요구 대역폭을 고려하지 않을 경우 (a)보다 더 작은 가중치를 갖게 되어 경로에 포함될 확률이 커진다. 그러나, 요구 대역폭을 포함시키면 (a)는 0.7의 이용률을 갖고 (b)는 0.8의 이용률을 갖게 되어 (a)가 경로에 포함될 확률이 커진다. 식 (5)에서는 현재의 측정값과 이전에 구해진 값을 α 라는 가중치를 주어 더 함으로써, 현재 측정값에 민감하게 반응하는 단점을 보완하며, 실험에서 α 값으로 0.5를 사용한다.

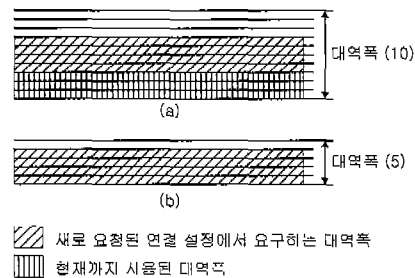


그림 1 요청 대역폭 포함에 의한 영향도

4.2 통계적인 접근법

SDP에서는 가용 대역폭의 역수를 이용하여 링크 가중치를 구하며, 이 경우 정규화 개념이 반영되지 않는다. 즉, 가용 대역폭의 절대적인 값 자체로 가중치를 구하는 것은 문제가 있다. 예를 들면, 가용 대역폭이 1000인 링크를 1000개로 구성된 경로의 비용도 $1000 * 1/1000$ 로 1의 비용을 가지고, 1의 가용 대역폭을 가진 링크 하나로 구성된 경로도 1의 비용을 갖게 된다. 따라서, 링크 가중치를 대역폭의 절대값에 무관하게 일정 범위내의 값을 갖도록 정규화해야 할 필요가 있다.

이렇게 가중치를 결정하기 위한 방법으로 링크의 이용률 값을 순차적으로 정렬한 후 일정 비율로 나누어 미리 정해진 값을 부여하는 방법이 있을 수 있다. 그림 2 (a)는 12개의 링크로 구성된 망에서 각 링크의 이용률이 중간인 한 지점에 비슷한 값을 갖고 분포하는 것

을 보이며, 이때 오른쪽으로 갈수록 큰 이용률 값을 뜻한다. 그림 2 (b)는 정렬에 의한 방법을 보여주며 중간에 집중적으로 분포한 특성을 표현하지 못하고 계산 시간이 $O(n \log n)$ 이다.

이러한 문제를 해결하기 위해 통계적인 접근법을 사용한다. 그림 2 (c)는 통계적 접근법에 의한 가중치 결정의 한 예를 보인다. 각 링크의 이용률이 구해지면 이 이용률들의 평균과 분산을 구하여 가중치를 결정하는데 사용한다:

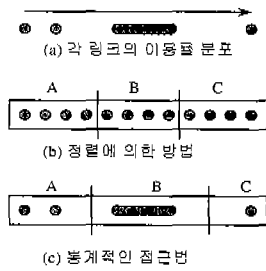


그림 2 링크 이용률을 이용한 가중치 결정

$$X = \{ Util(L_n), Util(L_1), \dots, Util(L_n) \} \quad (6)$$

$$\text{link } n \text{의 이용률들의 평균 : } E(X) \text{ or } \mu \quad (7)$$

$$\text{link } n \text{의 이용률들의 분산 : } V(X) \text{ or } \sigma^2 \quad (8)$$

위에서 구한 링크들의 이용률(Util)과 평균, 분산에 대해 중심 극한 정리[9] (Central Limit Theory)를 사용해 링크 이용률들의 분포를 정규화시켜 분포대별로 분류를 한다. 예를 들어, 10개의 구간으로 나누어 가중치를 부여하는 경우, 하위 10% ([0.0-0.1]), 하위 20%([0.1-0.2], ..., 상위 10% ([0.9-1.0]) 등으로 분류할 수 있다. 표 1에 이러한 분류의 한 예를 제시한다. $Z(L_n)$ 은 링크 n 의 이용률을 통계적 기법을 이용하여 정규화한 값이며, 이 값이 어느 구간에 속하는가에 따라 가중치가 $W[x]$ 로 결정된다:

$$Z(Util(L_n)) = (Util(L_n) - \mu) / \sigma \quad (9)$$

$W[x]$ 는 x 가 작을수록 작은 값을 갖고, x 가 클수록 큰 값을 갖는다. $W[x]$ 를 가중치로 하여 경로를 결정하면 링크 이용률의 평균보다 작은 값을 갖는 링크들이 라우팅에서 선택될 확률이 높아지게 되어 링크 이용률

의 분산 값이 작아진다. 따라서, 이 방법은 트래픽 성능을 좋게 하기 보다는 자원 이용률을 높여 트래픽 분배를 더 잘 하는 데에 치중하게 된다.

4.3 구간별 가중치

최단 경로를 구할 때 대부분의 경우 각 링크의 가중치란 동일한 값으로 하여 계산하지만, 본 논문에서는 TE를 위해 가중치를 적절하게 조절하여 효율적인 경로를 설정하고자 한다.

4.2장의 과정을 통해 각 링크들은 각각 $W[x] (a \leq x \leq b)$ 중 하나에 속하게 되고 각각의 가중치 $W[x]$ 는 다음과 같이 구한다:

$$W[x] = x^l \quad (10)$$

위 식에서 l 값으로 3가지 종류를 사용할 수 있다. l 값이 1 인 경우에는 선형성을 갖는 가중치를 의미하며, l 값이 1 보다 큰 값을 가지면 x 값이 큰 쪽을 강조하여 이용률이 매우 높은 링크를 적게 사용하고자 할 때 사용할 수 있고, l 값이 1보다 작고 0보다 큰 경우에는 x 값이 작은 쪽을 강조하여 이용률이 매우 낮은 링크를 많이 사용하고자 할 때 사용한다.

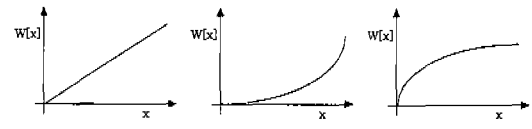


그림 3 l 값과 가중치와의 관계

분산을 이용하여 가중치를 구하면 이용률을 최대한 고려하면서도 정렬을 하지 않아도 되기 때문에 계산 시간에서 이득을 얻을 수 있다. 즉, 정렬을 이용하여 이용률들을 분류하면 $O(n \log n)$ 만큼의 계산시간이 필요하지만 분산을 이용할 경우 $O(n)$ 시간만에 분류가 가능하다. 이렇게 결정된 가중치를 이용해서 최단 경로를 구하게 되면 각 링크들의 이용률인 $Util(L_n)$ 들의 값을 비슷하게 유지하여 분산을 낮출 수 있고 부하 균등화 효과를 얻을 수 있다.

이상의 내용들을 기반으로 본 연구에서 제시하는 경로 선택 알고리즘인 SRS 알고리즘을 기술하면 다음과 같다:

표 1 구간별 가중치 할당 분류의 예(정규분포표 참조)

분류	0.0~0.1	0.1~0.2	0.2~0.3	0.3~0.4	0.4~0.5	0.5~0.6	0.6~0.7	0.7~0.8	0.8~0.9	0.9~1.0
$Z(Util(L_n))$	$-\infty \sim -1.28$	~ -0.84	~ -0.52	~ -0.25	~ 0	~ 0.25	~ 0.52	~ 0.84	~ 1.28	$\sim \infty$
$W[x]$	$W[1]$	$W[2]$	$W[3]$	$W[4]$	$W[5]$	$W[6]$	$W[7]$	$W[8]$	$W[9]$	$W[10]$

SRS (Statistical Route Selection) 알고리즘

입력

그래프 $G(N, L)$ 와 링크들의 대역폭의 집합 B
 진입노드 i 와 진출노드 e , 그들 사이에 요구되는 대역폭 b

출력

TE를 고려하면서 b 를 수용할 수 있는 i 와 e 사이의 경로

알고리즘

1. 모든 링크들에 대해 이용률 $Util(L)$ 을 구한다. (식 (3), (4), (5) 참조)
2. 이용률들의 집합인 X 에 대해 평균과 분산을 구한다. (식 (6), (7), (8) 참조)
3. 중심 극한 정리를 이용하여 링크들에 가중치 W 를 부여한다. (식 (9), (10) 참조)
4. 링크들중 b 를 만족시킬 수 없는 링크들을 라우팅에서 제외시킨다.
5. W 를 가중치로 하여 Dijkstra의 최단 경로 알고리즘을 적용시켜 i 와 e 사이의 경로를 결정한다.

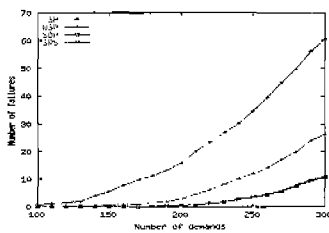
평균차수는 각각 2.2, 3.25, 4.6이다. 66개의 링크를 사용하면 평균차수가 작아서 비교하는 알고리즘들에서 성능의 차이가 작고, 138개의 링크를 사용했을 때는 98개를 사용했을 때에 비해 큰 성능 향상을 보여주지 않아 98개의 링크를 사용한 그래프만을 결과에서 보여준다. 각 링크의 대역폭은 50, 100, 200의 3가지 값을 노드의 차수에 비해 차수가 높을수록 높은 대역폭을 할당하였다. 경계 노드 결정은 총 노드수의 30%인 9개로 하였으며, 트래픽은 경계노드들 중 임의의 두 노드 쌍에 대해 2, 4, 6, 8, 10을 동일 확률로 하는 요청된 대역폭 값을 갖도록 했다.

진입노드와 진출노드 사이에 임의의 트래픽을 발생시켜 연결 설정 실패의 수, 혼잡(congested) 링크의 수, 이용률들의 평균, 이용률들의 분산, 연결설정을 위해 각 링크들마다 사용된 대역폭의 총합을 측정하였으며, 링크의 이용률이 0.7 이상인 링크를 혼잡 링크로 가정했다. 또한, 동일한 조건에서 총 10번의 상이한 트래픽을 발생시켜 그에 대한 평균을 최종적으로 보여준다.

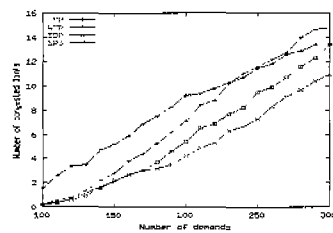
그림 4는 SP, WSP, SDP, SRS 각각에 대한 성능비교를 보여준다. 그림 4 (a)와 (b)에서는 임의의 트래픽을 발생시켜 연결 설정 실패수와 혼잡 링크수를 측정하며, 4.3장에서 설명한 SRS의 l 값으로는 2를 사용하였다. 그림 4 (a)에서 SP, WSP은 SDP, SRS에 비해 많은 연결 실패수를 가지며, SDP와 SRS는 비슷한 연결

5. 실험결과

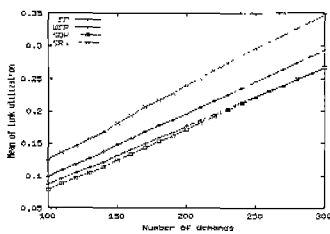
실험에서 사용된 그래프는 GT-ITM[10]이라는 자동 그래프 생성기를 이용하여 노드와 링크를 임의(random)로 생성하였으며, 30개의 노드에 링크 수는 각각 66, 98, 138개의 양방향 그래프로 각 그래프에 대한 노드의



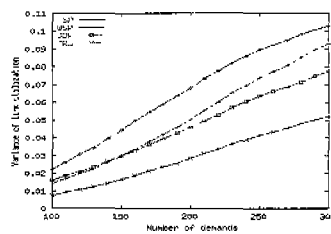
(a) 연결요청수의 증가에 따른 연결설정 실패수



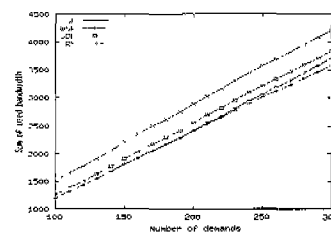
(b) 연결요청수의 증가에 따른 혼잡 링크수



(c) 연결요청수의 증가에 따른 링크 이용률의 평균



(d) 연결요청수의 증가에 따른 링크 이용률의 분산



(e) 연결요청수의 증가에 따른 총 대역폭 사용량

그림 4 여러 가지 경로 선택 알고리즘들의 성능비교

실패수를 가진다. 혼잡 링크의 수를 나타내는 그림 4 (b)에서는 더 많은 연결을 설정한 SRS와 SDP가 오히려 혼잡 링크수가 더 적은 것을 볼 수 있으며, SRS가 SDP보다 더 좋은 성능을 보인다.

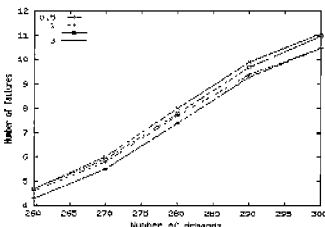
그림 4 (c), (d), (e)는 동일한 환경 즉, 같은 수의 연결 설정을 가지도록 하여 링크 이용률의 평균과 분산, 총 대역폭 사용량을 측정할 그래프로서, (c)에서는 SDP와 WSP가 비슷하게 좋은 성능을 보이고, (e)에서는 WSP가 가장 좋다. 특이한 점으로 SP, WSP, SRS 알고리즘들이 그림 4 (c)와 (e)에서 같은 순위의 성능을 보이는 반면, SDP는 (c)에서는 작은 값을 갖지만 (e)에서는 SRS와 비슷한 값을 갖는다는 것이다. 이것은 SDP가 대역폭이 큰 링크를 다른 방법들보다 상대적으로 많이 사용한다는 것을 뜻한다. 그림 4 (d)는 망 자원 이용률의 측면을 보이며, SRS가 가장 작은 분산을 갖는 것을 알 수 있고, 따라서 SRS가 이용률의 분산을 작게 하는데 우수한 성능을 보임을 알 수 있다.

그림 5는 그림 4에서의 비슷한 실험 방식을 사용하지만, SRS에 대해서 식 (10)에서의 l 값을 변화시켜 어떤 값을 사용하는 것이 좋은 성능을 보이는지 살펴본다. 그림 5 (a)에서는 전반적으로 비슷한 값을 갖지만 l 값이 작을 때 약간 더 좋은 성능을 나타낸다. 그림 5 (b)에서는 (a)에서와는 반대로 l 값이 클 때 더 좋은 성능을 보이며, 2와 3인 경우 비슷한 값을 가진다. 따라서,

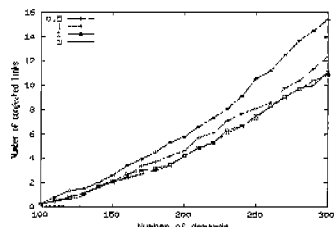
이것은 l 값을 더 크게 하더라도 많은 이득을 얻을 수 없다는 것을 의미한다. 그림 5 (c)와 (e)에서는 0.5, 1, 2, 3 순으로 망 자원을 적게 사용한다는 것을 보이며, (d)에서는 3, 2, 1, 0.5 순으로 망 자원 이용률을 분산시킨다. 또한 트래픽 성능과 망 자원 이용률, 혼잡 링크 수 등을 전반적으로 고려할 때 1을 l 값으로 이용하는 것이 TE의 측면에서 충분히 바람직함을 알 수 있다.

6. 결론

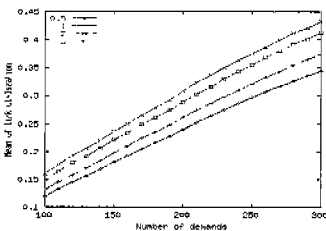
본 논문에서는 계속해서 새로운 연결 요청이 들어오는 동적인 망에서 망을 효율적으로 이용하면서 대역폭을 만족시키는 경로를 선택하는 SRS(Statistical Route Selection) 알고리즘을 제안했다. SRS 알고리즘은 링크들의 이용률에 대한 평균과 분산을 구하고 중심 극한 정리를 이용해 각 링크들의 이용률을 정규화한 후 정규화된 값들을 몇 개의 구간으로 나누어 그들 구간에 적절한 가중치를 할당함으로써 분산을 작게 하는 방향으로 수행된다. 망 자원을 효율적으로 이용하면서 트래픽 성능도 높여야 한다는 TE의 두 가지 목적중에서 SRS 알고리즘은 망 자원을 효율적으로 이용하는 쪽에 비중을 두고 경로를 선택하는 방식을 택했다. 결과적으로 혼잡 링크수에서 SRS는 SP, WSP, SDP보다 더 좋은 성능을 보여준다. 그러나, 트래픽 성능의 측면을 고려한다면 단순하면서도 좋은 성능을 갖는 SDP를 주목할 필요



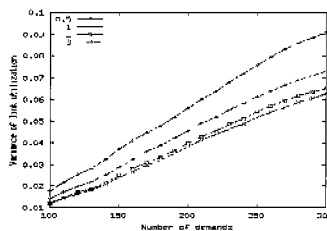
(a) 연결요청수의 증가에 따른 연결설정 실패수



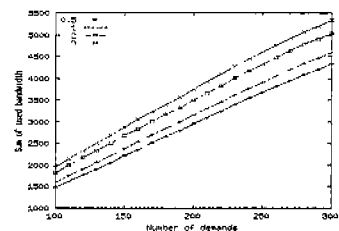
(b) 연결요청수의 증가에 따른 혼잡 링크수



(c) 연결요청수의 증가에 따른 링크 이용률의 평균



(d) 연결요청수의 증가에 따른 링크 이용률의 분산



(e) 연결요청수의 증가에 따른 총 대역폭 사용량

그림 5 여러 종류의 l 값을 사용했을 경우의 성능비교

가 있으며, SRS의 통계적 기법에 SDP의 가용 대역폭의 역수, 즉 절대적인 대역폭 값을 이용하는 방법을 수용한다면 더 좋은 성능을 낼 수 있을 것으로 생각되어 향후 연구과제로 남긴다.

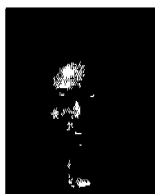
참 고 문 헌

- [1] E. Rosen, A. Viswanathan, and R. Callon, "Multi-protocol Label Switching Architecture," *Internet draft, draft-ietf-mpls-arch-06.txt*, Aug 1999.
- [2] J. Moy, "OSPF version 2," *RFC 2328*, Apr 1998.
- [3] C. Villamizar, "OSPF Optimized Multipath (OSPF-OMP)," *Internet draft, draft-ietf-ospf-omp-02.txt*, Feb 1999.
- [4] R. Perlman, "Interconnections, Second Edition, Bridges, Routers, Switches, and Internetworking Protocols," Addison-Wesley, 1999.
- [5] M. Kodialam, T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," *Proceedings of Infocom 2000*.
- [6] B. Fortz, M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," *Proceedings of Infocom 2000*.
- [7] Q. Ma, "QoS Routing in the Integrated Services networks," *Ph.D. thesis, CMU-CS-98-138*, Jan 1998.
- [8] D. Awduche et al., "Requirements for Traffic Engineering Over MPLS," *RFC 2702*, Sep 1999.
- [9] R. V. Hogg, A. T. Craig, "Introduction to Mathematical Statistics," Prentice Hall, pp. 246-253, 1995.
- [10] E. W. Zegura, "GT-ITM: Georgia tech internet-work topology models (software)," <http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz>.



김 영 빈

1999년 2월 서울시립대학교 컴퓨터·통계학과 졸업(이학사). 2001년 2월 서울시립대학교 컴퓨터·통계학과 졸업(이학석사). 2000년 8월 ~ 현재 아코테크(주) 연구소 초고속 사업부 주임연구원. 관심분야는 QoS 라우팅, 멀티캐스트, MPLS



안 상 현

1986년 서울대학교 컴퓨터공학과 졸업. 1988년 서울대학교 컴퓨터공학과 석사. 1993년 University of Minnesota 전산학 박사. 1988년 ~ 1989년 데이콤 사원. 1994년 3월 ~ 1998년 2월 세종대학교 전산학과 교수. 1998년 3월 ~ 현재 서울시립대학교 컴퓨터·통계학과 교수