

통계적 기법을 이용한 경로 선택 알고리즘

김영민^o 안상현
서울시립대학교 전산통계학과
(ymkim92, ahn)@venus.uos.ac.kr

A Route Selection Algorithm using a Statistical Approach

Youngmin Kim^o Sanghyun Ahn
Dept. of Computer Science & Statistics, University Of Seoul

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 사용할 수 있도록 하는 통계적 경로 선택(Statistical Route Selection; SRS) 알고리즘을 제안한다. MPLS[4]의 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 링크 이용률의 평균과 분산을 이용하는 것으로, 이 정보를 기반으로 링크의 가중치에 대해 분산을 작게 하는 방향으로 경로를 결정함으로써 부하 균등화 효과를 얻게 되어 망 자원 이용률을 높인다. 실험을 통해 SP, WSP, SDP[3] 알고리즘에 비해 SRS 알고리즘이 망 자원을 효율적으로 이용하여 연결 설정 실패의 수와 혼잡 링크의 수를 줄이는 것을 보인다.

1. 서론

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 TE(Traffic Engineering)가 요구되며, TE를 효율적으로 이용하여 트래픽을 적절히 분배하면 사용자들의 요구사항에 적합한 서비스를 제공할 수 있다.

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.

2장에서는 관련 연구들을 기술하고, 3장에서는 본 논문에서 새롭게 제안한 경로 선택 방법인 SRS에 대해 살펴보고, 4장에서는 실험 결과를 보이고, 5장에서 결론을 맺는다.

2. 관련연구

[3]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 경로를 구할

때, 한 지점에서 같은 비용을 갖는 둘 이상의 링크를 만났을 경우 대역폭이 큰 링크를 선택하는 방법으로 WSP(Widest-Shortest Path)가 있다. 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 이용률이 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 역수를 의미하며, 거리를 최소로 하는 경로를 선택한다.

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

위의 4가지 방법의 성능을 비교한 결과, SDP가 전반적으로 가장 좋은 성능을 보인다 [3].

TE의 목적은 망 자원 이용률과 트래픽 성능(traffic performance)을 동시에 최적화시킬 수 있는 효율적이고 신뢰할 수 있는 망을 구성하는 것이다 [1]. 본 논문에서는 TE 요구사항들에 대한 절충점, 즉, 부하 균등화(load balancing)를 적절히 수행해 망 자원 이용률도 높으면서 트래픽 성능을 높일 수 있는 경로 설정 방법을 찾고자 한다.

3. SRS(Statistical Route Selection) 알고리즘

한 링크의 가용 대역폭은 그 링크의 총 대역폭에서 현재

사용중인 대역폭을 빼 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 최소화하는 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:

$$\text{minimize}(V(X)), X \text{는 링크들의 이용률의 집합} \quad (2)$$

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.

3.1 링크 이용률 구하기

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:

$$\text{링크 } n \text{의 현재 이용률 : } Util(L_n) \quad (3)$$

$$\text{m번째 측정된 링크 } n \text{의 이용률 :} \quad (4)$$

$$\begin{aligned} \text{NowUtil}(L_n)_m &= (\text{RequestedBandwidth} \\ &+ \text{UsedLinkBandwidth}) / \text{TotalLinkBandwidth} \end{aligned}$$

$$\text{m번째 측정시의 } Util(L_n) : \quad (5)$$

$$\begin{aligned} Util(L_n)_m &= \alpha * Util(L_n)_{m-1} + \\ &(1 - \alpha) \text{NowUtil}(L_n)_m, \quad (0 \leq \alpha \leq 1) \end{aligned}$$

3.2 통계적인 접근법

SDP에서는 가용 대역폭의 역수를 이용하여 링크 가중치를 구하며, 이 경우 정규화 개념이 반영되지 않으며, 가용 대역폭의 절대적인 값 자체로 가중치를 구하는 것은 문제가 있다. 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 1 (a)는 12개의 링크로 구성된 망에서 각 링크의 이용률이 중간 정도의 한 지점에 비슷한 값을 갖고 분포하는 것을 보이며, 이때 오른쪽으로 갈수록 큰 이용률 값을 뜻한다: 정렬에 의한 방법을 보여주는 그림 1 (b)는 중간에 집중적으로 분포한 특성을 표현하지 못하며 계산 시간이 $O(n^2)$ 이다.

이러한 문제를 해결하기 위해 통계적인 접근법을 사용한다. 그림 1 (c)는 통계적 접근법에 의한 가중치 결정의 한

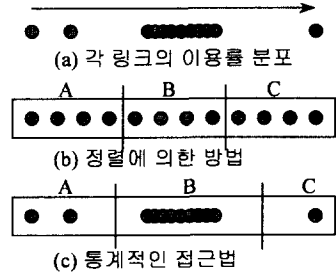


그림 1. 링크 이용률을 이용한 가중치 결정

예를 보인다. 각 링크의 이용률이 구해지면 이 이용률들의 평균과 분산을 구하여 가중치를 결정하는데 사용한다:

$$X = \{ Util(L_0), Util(L_1), \dots, Util(L_n) \} \quad (6)$$

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

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

위에서 구한 링크들의 이용률(Util)과 평균, 분산에 대해 중심 극한 정리[2] (Central Limit Theory)를 사용해 링크 이용률들의 분포를 정규화시켜 분포대별로 분류한다. 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.

이상의 분류에 의해 각 링크들은 각각 $W[x](a \leq x \leq b)$ 중 하나에 속하게 되고 각각의 가중치 $W[x]$ 는 다음과 같이 구한다:

$$W[x] = x^l, l > 0 \text{ 인 실수} \quad (9)$$

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

SRS (Statistical Route Selection) 알고리즘 입력

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

출력

TE를 고려하면서 d 를 수용할 수 있는 i 와 e 사이의 경로

알고리즘

1. 모든 링크들에 대해 이용률 $Util(L)$ 를 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) 참조)
4. 링크들 중 d 를 만족시킬 수 없는 링크들을 라우팅에서 제외시킨다.
5. W 를 가중치로 하여 Dijkstra의 최단 경로 알고리즘을 적용시켜 i 와 e 사이의 경로를 결정한다.

분산을 이용하여 가중치를 구하면 이용률을 최대한 고려

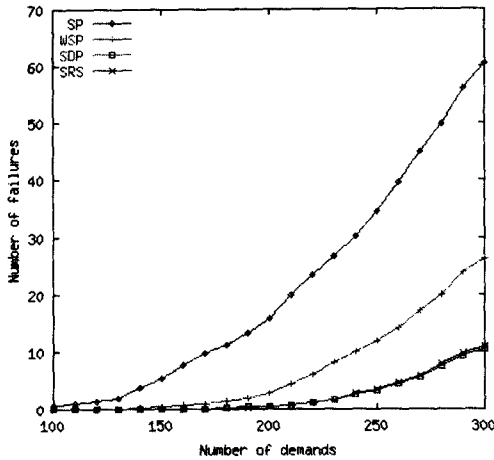


그림 2. 연결요청수의 증가에 따른 연결설정 실패수

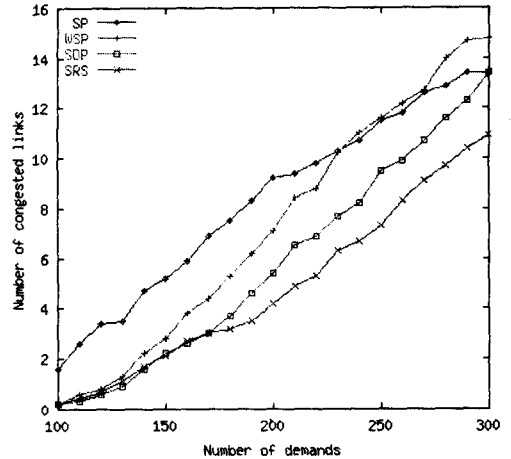


그림 3. 연결요청수의 증가에 따른 혼잡 링크수

하면서도 정렬을 하지 않아도 되기 때문에 계산 시간에서 이득을 얻을 수 있다. 즉, 정렬을 이용하여 이용률들을 분류하면 $O(n^2)$ 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.

4. 실험결과

실험에서 사용된 그래프는 GT-ITM[5]이라는 자동 그래프 생성기를 이용하여 생성되었으며, 30개의 노드에 링크수 98개의 양방향 그래프로 노드의 평균 차수는 3.25이다. 각 링크의 대역폭은 50, 100, 200의 3가지 값을 노드의 차수에 비례해 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 수를 측정하였으며, 링크의 이용률이 0.7 이상인 링크를 혼잡 링크로 가정했다. 또한, 동일한 조건에서 총 10번의 상이한 트래픽을 발생시켜 그에 대한 평균을 최종적으로 보여준다.

식 (9)에서의 l 값으로는 0.5, 1, 2, 3을 대입한 결과, 2를 사용하는 경우 좋은 성능을 보였기 때문에 실험에서 l 값으로 2를 사용하였다.

그림 2, 3은 SP, WSP, SDP, SRS들에 대한 성능비교를 보여준다. 그림 2에서 SP, WSP은 SDP, SRS에 비해 많은 연결 실패수를 가지며, 그림 3에서도 더 많은 연결을 설정한 SRS와 SDP가 혼잡 링크수가 더 적어 좋은 성능을 보인다. 그러나, SRS와 SDP에 대해서 연결 설정수는 비슷하나, 혼잡 링크수에서는 SRS가 더 좋은 성능을 보인다.

5. 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 필요가 있으며, SDP의 장점과 SRS의 장점을 수용한다면 더 좋은 성능을 낼 수 있을 것으로 생각되어 향후 연구과제로 남긴다.

6. 참고 문헌

- [1] D. Awduche et al., "Requirements for Traffic Engineering Over MPLS," *RFC 2702*, Sep 1999.
- [2] R. V. Hogg, A. T. Craig, "Introduction to Mathematical Statistics," Prentice Hall, pp. 246-253, 1995.
- [3] Q. Ma, "QoS Routing in the Integrated Services networks," *Ph.D. thesis, CMU-CS-98-138*, Jan 1998.
- [4] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," *Internet draft, draft-ietf-mpls-arch-06.txt*, Aug 1999.
- [5] E. W. Zegura, "GT-ITM: Georgia tech internet network topology models (software)," <http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz>.