

예측 기반 QoS 라우팅 알고리즘

주미리*, 조강홍**, 박응기*,

*국가보안기술연구소

**동양공업전문대학 모바일 인터넷과

e-mail : mrjoo@etri.re.kr

Prediction-based QoS Routing Algorithm

Mi-Ri Joo*, Kang-Hong Cho**, Eung-Ki Park*,

*National Security Research Institute

**Dongyang Technical College

요 약

본 논문에서는 기존의 QoS 라우팅 알고리즘이 가지고 있는 문제점인 네트워크 상태 정보 오버헤드를 극복하고 네트워크 상태의 정확성을 유지하기 위한 예측 기반 QoS 라우팅 기법인 SP(Shortest-Prediction) 라우팅 알고리즘 모델을 제안하고, 알고리즘의 성능 평가를 위하여 실제 네트워크와 유사한 MCI 네트워크 상에서 시뮬레이션 수행하였으며 라우팅 실패율과 라우팅 부정확율의 비교를 통하여 본 알고리즘의 우수성을 확인하였다.

1. 서론

QoS 라우팅이란 적정 경로(feasible path)를 동적으로 결정하는 라우팅이라고 정의할 수 있으며, 여기서 적정 경로란 사용자의 플로우(flow)가 요구하는 QoS를 지원할 수 있으면서 서비스 제공자나 경로 사용 비용 등과 같은 정책 제약에 위반되지 않는 경로를 의미한다[1][2]. QoS 라우팅은 QoS를 요구하는 플로우에 대하여 이러한 적정 경로를 찾아주되 네트워크의 자원 활용률을 극대화하고, 네트워크 엔지니어링과 네트워크에 발생하는 트래픽 부하 간에 일시적인 불일치가 발생하는 경우의 성능 저하를 최소화 하는 것을 목적으로 한다[2].

즉, 사용자의 플로우의 QoS 요구를 만족시킬 수 있는 적정 경로를 선택하되 그 플로우에 의해 소모되는 네트워크 자원의 양을 절감하고 네트워크 상의 부하 분포를 균등하도록 조절함으로써 네트워크 자원 활용률을 극대화해야한다. 따라서 라우팅 시 고려되어야 할 가장 중요한 내용은 사용자의 QoS 요구 사항에 대한 최적 경로를 선택하는 것과 네트워크 자원을 고르게 활용하여 극대화하는 것이다[2].

본 논문에서는 기존의 QoS 라우팅 알고리즘들이

가지는 문제점인 네트워크 상태(예를 들면, 가용 대역폭) 정보 오버헤드를 극복하고 네트워크 상태의 정확성을 유지하여 네트워크 상의 자원 활용율과 라우팅 성능을 향상시키는 예측 기반 QoS 라우팅 알고리즘을 제시하고자 한다.

2. QoS 라우팅

QoS 라우팅에서 경로를 선택할 때 고려되어야 할 가장 중요한 내용은 다음의 두 가지 항목이다.

- 사용자의 QoS 요구 사항 : 적절한 경로 선택을 위해서 가장 기본이 되는 요구 사항으로 사용자가 요구하는 QoS 요구 사항을 만족시켜야 한다. 사용자가 요구하는 대표적인 QoS 요구 사항은 지연, 지연 지터, 대역폭 등이 있으며, 따라서 이를 메트릭으로 설정하여 적절한 경로를 선택하게 된다.
- 네트워크의 자원 활용 : 네트워크 상의 부하를 균등하게 제공함으로써 네트워크 자원 활용률을 극대화하는 것이 요구된다. 이를 통해 보다 많은 사용자의 요구 플로우를 받아들일 수 있어야 한다.

결과적으로 QoS 라우팅 알고리즘은 사용자의 QoS 요구를 만족시키면서, 네트워크 자원 사용을 최소화하는 것을 목적으로 한다. 이와 관련해서 기존에 다양한 연구가 진행되었고, 이미 제시된 QoS 라우팅 알고리즘은 다음과 같다[1][2][3].

- Linear 알고리즘 : 어떤 경로를 구성하는 모든 링크의 비용을 합하여 가장 낮은 비용을 가지는 경로를 선택한다. 예를 들어, 출발지 x 와 목적지 y 사이의 경로 P_{xy} 를 구성하는 링크를 e_i 라고 할 때, 이 링크 e_i 의 비용 $Cost_i$ 는 사용 대역폭 U_i 와 링크 대역폭 C_i 의 비율 $\frac{U_i}{C_i}$ 로 계산된다. 경로 P_{xy} 의 비용 $Cost_{xy}$ 는 경로를 구성하고 있는 각 링크의 비용의 합인 $\sum_{i=1}^n Cost_i$ 로 계산된다.
- Exponential 알고리즘 : 어떤 경로를 구성하는 모든 링크의 지수 함수를 기준으로 한 비용을 합하여 가장 낮은 비용을 가지는 경로를 선택한다. 이 경우에는 출발지 x 와 목적지 y 사이의 경로 P_{xy} 를 구성하는 링크를 e_i 라고 할 때, 링크 e_i 의 비용 $Cost_i$ 는 사용 대역폭 U_i 와 링크 대역폭 C_i 의 비율의 지수 함수인 $a^{\frac{U_i}{C_i}}$ 로서 계산된다. 여기서 a 는 상수이다. 경로 P_{xy} 의 비용 $Cost_{xy}$ 는 경로를 구성하고 있는 각 링크의 비용의 합인 $\sum_{i=1}^n Cost_i$ 로 계산된다.
- Widest-shortest 알고리즘 : 모든 적정 경로들 중에 홉 카운트가 가장 적은 경로를 선택한다. 만약 동일한 홉 카운트를 가지는 경로가 여러 개 존재한다면, 이들 중 최대 가용 대역폭을 가지는 경로를 선택한다. 만약 동일한 대역폭을 가지는 경로가 여러 개 존재할 경우에는 랜덤하게 하나의 경로를 선택한다.
- Shortest-widest 알고리즘 : 모든 적정 경로들 중에 최대 가용 대역폭을 가지는 경로를 선택한다. 만약 동일한 대역폭을 가지는 경로가 여러 개 존재할 경우에는, 최소 홉 카운트를 가지는 경로를 선택한다. 만일 동일한 홉 카운트를 가지는 경로가 여러 개 존재할 경우에는 랜덤하게 하나의 경로를 선택한다.

- Shortest-distance 알고리즘 : 모든 적정 경로들 중에 가장 짧은 distance를 가지는 경로를 선택한다. distance function은 다음과 같이 정의된다. 여기서, R_{i_j} 는 링크 i_j 의 가용 대역폭이다.

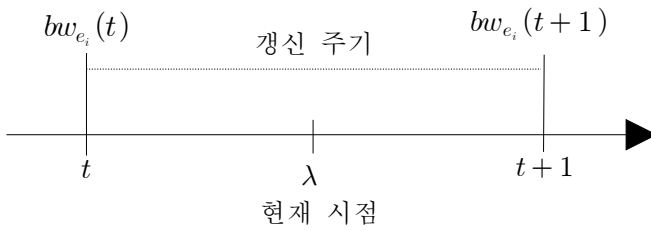
$$dist(p) = \sum_{j=1}^k \frac{1}{R_{i_j}}$$

3. 제안하는 예측 기반 QoS 라우팅 알고리즘

QoS 라우팅은 사용자의 QoS 만족시키는 적정 경로를 찾음과 동시에 네트워크의 자원 활용률을 극대화하는 것을 목적으로 한다. 사용자의 QoS를 만족시키기 위해서는 동적으로 변하는 네트워크에 상태 정보들을 파악하는 것이 필요하다. 이를 위해 네트워크를 구성하는 각 노드들은 주기적으로 또는 어떤 조건에 일치하면 네트워크 상태 정보를 교환하게 된다.

네트워크 상태 정보를 교환하는 방법은 상대적인 가용 자원 변화율 또는 임계치 기반 갱신 유발 정책, 절대적인 가용 자원 변화 혹은 클래스 기반 갱신 유발 정책, 그리고 타이머 기반 갱신 유발 정책 등이 있다. 이런 방법들은 모두 기본적으로 네트워크 상태 정보 전달을 위한 오버헤드를 최소화하면서 네트워크 상태 정보의 정확성을 높이고자 한다. 경로 선택을 위해서 가장 최근에 받은 네트워크의 상태 정보를 이용하게 되는데 시간이 지남에 따라 네트워크 상태는 계속적으로 변함으로 그 정확성이 떨어지게 된다. 이와 같이 정확성이 떨어지는 정보를 기준으로 경로 선택을 할 경우에 그 성능이 떨어져 사용자의 QoS를 만족시키지 못하게 된다. 즉, 네트워크의 정확한 상태 정보는 QoS 라우팅의 성능에 가장 큰 영향을 미치기 때문이다.

[그림 1]에서 보는 바와 같이 어떤 링크 e_i 에 대한 가용 대역폭을 bw_{e_i} 라고 하고 어떤 라우터 v_i 에서 이 정보를 마지막으로 수신한 시점을 t 라고 할 경우에 이 링크 e_i 의 가용 대역폭을 $bw_{e_i}(t)$ 라고 하자. 현재 시점을 λ 라고 하면 다음 $t+1$ 시점의 가용 대역폭 $bw_{e_i}(t+1)$ 이 전송될 때까지 가장 최근의 상태 정보를 이용하여 경로 선택을 수행하게 된다. 그러나 실제로 링크 e_i 의 가용 대역폭은 빈번하게 변경되기 때문에 시간이 지날수록 정확한 적정 경로 선택이 어려워지므로 경로 선택 성능이 떨어지게 된다.



[그림 1] 상태 정보 갱신

본 논문에서 제안하는 SP 라우팅 알고리즘은 기존의 Shortest-widest 알고리즘에 다음 시점의 대역폭 예측값을 적용하여 그 성능을 확장시킨 방안이다.

여기서 가용 대역폭 예측값 $\widehat{bw}_{e_i}(t+1)$ 를 계산하기 위하여 실제 많이 사용되고 계산의 복잡성이 높지 않은 AR(Autoregressive Model)을 사용하였다.

소스 노드 s 와 목적지 노드 d 사이의 모든 적정 경로들 $P_{all}(s,d)$ 중에 마지막으로 수신한 t 시점에서, 이 경로를 구성하는 각 링크의 가용 대역폭 $bw_{e_i}(t)$ 와 가용 대역폭 예측값 $\widehat{bw}_{e_i}(t+1)$ 을 계산한다. 이때, 각 링크의 가용 대역폭 $bw_{e_i}(t)$ 와 가용 대역폭 예측값 $\widehat{bw}_{e_i}(t+1)$ 중 어느 값을 기준으로 하느냐에 따라 알고리즘이 SP-AVG, SP-MIN, SP-MAX로 구분되며 이 값을 기준으로 각 경로의 mrb (maximum reservable bandwidth)를 구한 후 가장 큰 mrb 를 가지는 경로를 선택하게 된다. 만약 동일한 mrb 를 가지는 경로가 존재할 경우에는 가장 작은 홉 카운트를 가지는 경로를 선택한다. 이 때에도 동일한 비용을 가지는 경로가 여러 개 존재할 경우에는 네트워크 부하의 균등을 위해 무작위로 하나의 경로를 선택한다. 이 방식은 시간에 따라 달라지는 네트워크 상태 정보에 대해 현재와 그 다음 시점의 중간에 기준을 두고 경로 선택을 하기 때문에 양 시점에 비교적 적합한 경로 선택 기능을 수행한다.

- SP-AVG 알고리즘 : 가용 대역폭과 가용 대역폭 예측값의 평균값을 계산하고 각 경로의 mrb 를 구한다. 이 들 중 가장 큰 mrb 를 가지는 경로를 선택한다.

$$Cost_{bw}(P_i) = mrb \left\{ \frac{bw_{e_i}(t) + \widehat{bw}_{e_i}(t+1)}{2} \right\}$$

$$\max \{ Cost_{mrb}(P_i) \}$$

- SP-MIN 알고리즘 : 가용 대역폭과 가용 대역폭 예측값 중에 최소값을 계산하고 이를 기준으로

각 경로의 mrb 를 구한다. 이 들 중 가장 큰 mrb 를 가지는 경로를 선택한다.

$$Cost_{bw}(P_i) = mrb \{ \min(bw_{e_i}(t), \widehat{bw}_{e_i}(t+1)) \}$$

$$\max \{ Cost_{bw}(P_i) \}$$

- SP-MAX 알고리즘 : 가용 대역폭과 가용 대역폭 예측값 중에 최대값을 계산하고 이를 기준으로 각 경로의 mrb 를 구한다. 이 들 중 가장 큰 mrb 를 가지는 경로를 선택한다.

$$Cost_{bw}(P_i) = mrb \{ \max(bw_{e_i}(t), \widehat{bw}_{e_i}(t+1)) \}$$

$$\max \{ Cost_{bw}(P_i) \}$$

4. 시뮬레이션

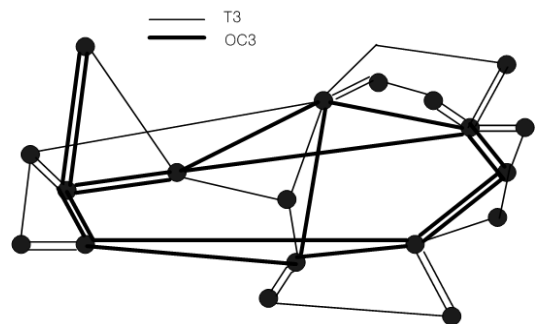
4.1 시뮬레이션 환경

본 논문에서 제시하는 예측 기반 라우팅 알고리즘과 기존에 제시된 라우팅 알고리즘의 성능을 평가하기 위해 라우팅 시뮬레이터를 작성하였다. 라우팅 시뮬레이터는 Java 버전 1.4으로 구현하였으며, 실험에 사용된 기종은 Sun Ultra60(운영 체제는 Solaris 2.8)과 펜티엄3 PC 733MHz(운영 체제는 윈도우 XP)이다.

4.2 네트워크 토폴로지

그림 2는 MCI 네트워크 토폴로지로 MCI 네트워크 토폴로지는 미국의 실제 백본 네트워크와 가장 유사하게 만들어진 환경으로 기존 제안 알고리즘들이 성능 평가를 위해 수행한 대표적인 시뮬레이션 네트워크 환경이다. 본 논문에서도 기존 제안 알고리즘과 본 논문에서 제시한 알고리즘의 성능 평가를 하기 위해 MCI 네트워크 토폴로지를 사용하였다.

MCI 네트워크 토폴로지는 19개의 노드와 77개의 OC-3(155Mbit/s)와 T3(45Mbit/s)의 선로들로 구성되어 있다.



[그림 2] MCI 네트워크 토폴로지

4.3 트래픽 부하

본 시뮬레이션 환경에서는 다양한 특성을 가지는 트래픽을 발생하여 실험의 주관성을 배제하였다. 먼저 1 ~ 5 Mbit/s 사이의 CBR(Constant Bit Rate) 트래픽을 균일 분포(uniform distribution)에 따라 랜덤하게 발생하였고, 이 때 플로우 유지 시간(Holding Time)은 평균 1200초 또는 2400초 값을 가지고 지수 분포(exponentially distribution)에 따라 발생시켰다. 시뮬레이션 네트워크 상태의 트래픽 부하를 제공하기 위해 위의 트래픽의 도착율을 0.3, 0.4, 0.5, 0.6, 0.7 초로 설정하여 지수 분포에 따라 발생시켰다. 또한, 네트워크 상태 정보 갱신 주기는 10초, 30초, 60초를 기준으로 각 알고리즘의 성능을 평가하였다.

4.4 성능 평가 메트릭

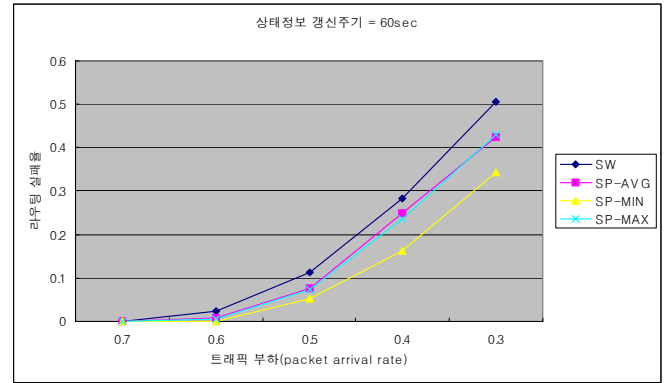
적정 경로를 찾는 라우팅 알고리즘 측면에서의 성능 평가는 기존에 다른 연구에서 많이 다루었듯이 라우팅 실패율(call blocking rate)과 라우팅 대역폭 실패율(bandwidth blocking rate)로 나누어 볼 수 있다.

- 라우팅 실패율(Call Blocking Rate) : 라우팅 실패율은 전체 사용자 플로우 횟수에 대한 사용자 플로우가 적정 경로를 찾지 못하고 실패하는 횟수의 비율을 나타낸다.
- 라우팅 부정확율(Routing Inaccuracy) : 라우팅 부정확율은 라우팅 알고리즘이 사용자가 요청한 플로우의 대역폭을 만족시키는 경로를 선택했지만, 실제 링크상의 대역폭의 부족으로 인해 라우팅이 실패되는 비율을 의미한다.

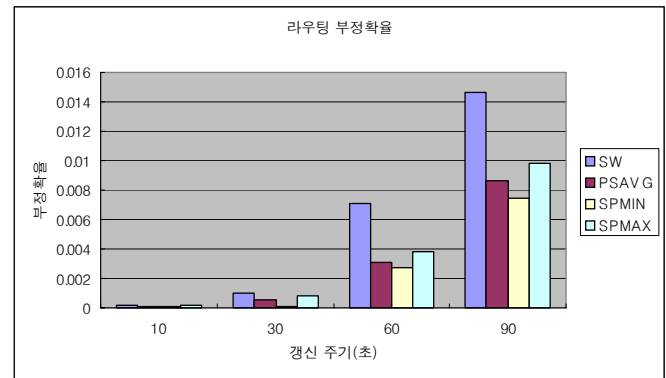
위에서 제시한 성능 평가 메트릭을 기준으로 기존에 제시된 라우팅 알고리즘과 본 논문에서 제안하는 예측 기반 QoS 라우팅 기법을 비교함으로써 그 성능 향상 여부를 비교 분석할 수 있을 것이다.

4.5 시뮬레이션 결과

[그림 3]과 [그림 4]에서 보는 바와 같이 기존의 SW 라우팅 알고리즘에 비해 본 논문에서 제안하는 SP알고리즘의 성능이 모두 우수한 것을 볼 수 있으며 그 중에서 SP-MIN이 가장 낮은 라우팅 실패율과 라우팅 부정확율을 보임으로서 가장 우수한 성능을 보임을 확인하였다.



[그림 3] 라우팅 실패율



[그림 4] 라우팅 부정확율

5. 결론

본 논문에서는 기존의 알고리즘들이 가지는 문제점인 시간에 따른 네트워크 상태 정보의 부정확성과 상태 정보 오버헤드를 극복하고자 예측 모델을 이용한 예측 기반 라우팅 알고리즘 기법인 SP 라우팅 알고리즘을 제안하였고, 기존의 다른 알고리즘과의 성능 평가를 위하여 시뮬레이션을 수행하였다. 성능 비교를 위하여 라우팅 실패율과 라우팅 부정확율을 비교함으로써 제안하는 알고리즘의 우수성을 확인하였다. 추가적으로 다양한 기존 알고리즘과의 성능 평가를 위한 시뮬레이션 연구를 계속할 계획이다.

참고문헌

- [1] Q. Ma, "Quality-of-service routing in integrated services networks", PhD Thesis, Carnegie Mellon University, 1998
- [2] Q. Ma and P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees", Proceedings of IEEE International Conference on Network Protocols, Atlanta, Georgia, October 1997
- [3] R. Guérin, and A. Orda, "QoS Based Routing in Networks With Inaccurate Information: Theory and Algorithms", in proceedings of INFOCOM, 1997