

論文 2003-40TC-12-5

로컬 QoS 라우팅을 위한 경로선택 알고리즘

(Path Selection Algorithms for Localized QoS Routing)

徐 庚 龍 *

(Kyungryong Seo)

요 약

최근에 글로벌 QoS 라우팅방식의 약점을 개선하려는 노력으로 로컬 QoS 라우팅 방식이 제시되었다. 글로벌 라우팅과는 다르게 로컬 QoS 라우팅에서는 라우터간의 상태교환을 수행하지 않고 virtual capacity를 사용하여 라우터가 독자적으로 라우팅을 수행한다. 로컬 QoS 라우팅에서는 소스와 목적간의 경로가 효과적으로 선택 되어야만 좋은 성능을 보장 받을 수 있다. 본 논문에서는 효과적인 경로 선택을 위하여 몇몇의 휴리스틱 을 제시하고 이를 활용한 경로 선택 알고리즘을 제안하였다. 제안된 알고리즘은 시뮬레이션 과 함께 분석되었으며 경로선택방식에 따라 로컬 QoS 라우팅의 성능이 매우 큰 영향을 받는다는 것을 확인하였다.

Abstract

Recently, localized QoS routing scheme was recently proposed for overcome drawbacks of global state QoS routing schemes. Localized QoS routing dose not exchange QoS states among routers, but use virtual capacity based routing scheme instead. In localized QoS routing, to archive good performance, a set of candidate paths must be selected between the source and the destination effectively. In this paper we propose a few heuristics for effective path selection and develop path selection algorithms based on the heuristics. More detail analysis of the proposed algorithm is presented with simulation results which demonstrate that the path selection method can very affect the performance of localized QoS routing.

Keywords : localized QoS routing, path selection

I. 서 론

오늘날 네트워크는 응용 프로그램에 대역폭, 전송지연, 우회경로, 비용 등 다양한 형태의 QoS(Quality of Service)를 지원할 수 있도록 발전하였다. 응용프로그램은 자신의 QoS 조건을 만족하는 경로를 요청하여 통신의 품질을 보장받고자 하는데 QoS를 만족하는 경로를 선택하여 제공하는 것을 QoS라우팅 이라고 한다^[1-9].

* 正會員, 釜慶大學校 컴퓨터工學科

(Division Electronics, Computer and Telecommunication Engineering Pukyong National University)

接受日字:2003年11月8日, 수정완료일:2003年12月3日

QoS 라우팅을 위하여 각 라우터는 자신의 링크정보를 다른 라우터와 주기적으로 교환하여 전체네트워크의 QoS 상태를 확보한 후 QoS 경로를 선정한다. 이러한 방식을 글로벌 QoS 라우팅 이라고 하는데 네트워크의 전체 QoS 정보를 각 라우터가 공유하고 Dijkstra 의 최소경로 알고리즘에 기초한 다양한 효율적인 라우팅 방법이 연구가 있다^[1, 2, 6].

하지만 네트워크내의 각 상태는 수시로 변화한다. 라우팅 경로가 결정되어 RSVP로 통신에 필요한 지원을 예약하거나 예약된 지원이 해지되는 경우에 해당 지원의 QoS 상태는 변화하게 된다^[4, 9]. 따라서 글로벌 QoS 라

우팅이 원활히 수행되기 위해서는 변화된 상태 값을 빨리 갱신해야 하는데 이를 위하여 라우터간의 상태정보 교환이 빈번하게 이루어야 하고 네트워크에 큰 부담으로 작용한다. 이러한 점을 고려하여 라우터간의 상태교환을 필요로 하지 않는 로컬 QoS 라우팅방법이 제안되었다^[11].

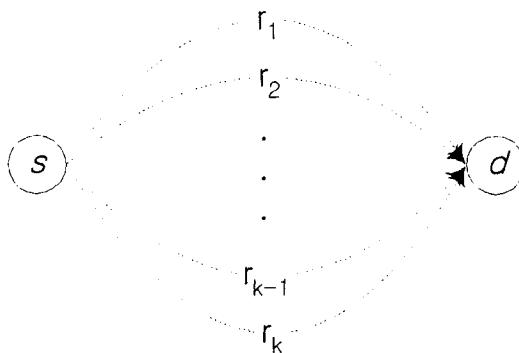


그림 1. 소스 s 와 목적 d 간의 독립적인 경로들
Fig. 1. A set of disjoint paths between source s and destination d.

여기서는 네트워크 흐름의 블로킹 확률을 중요한 QoS 지표로 사용하고 네트워크를 <그림 1>과 같이 소스노드 s 와 목적노드 d 간에 k개의 독립적인 vc (virtual capacity)로 연결된 그래프 모델로 변환하였다. 이 경우 경로를 개설이나 해지하는 경우 블로킹 상태 값이 변화하게 되는데 각각의 라우터는 자신의 블로킹 상태 값을 변화시킨다. 어떤 vc 에 라우팅 경로가 개설되면 QoS 상태가 변화하게 되는데 각 vc 가 독립적으로 동작하므로 vc 에 연결된 각 라우터는 상태교환 없이 개설된 라우팅 경로에 의한 블로킹 상태변화를 스스로 계산할 수 있다.

로컬 QoS 라우팅은 글로벌 QoS 라우팅에 비교하여 매우 단순하고 효율적이며 네트워크의 상태변화에 적응적으로 동작한다. 하지만 여기서 독립적이라고 가정한 vc 는 실제의 환경에서는 어떤 링크를 공유하게 되고 이로 인하여 지원하는 QoS 의 품질이 변하게 된다. 그런데 이 연구에서는 경로선정 방법은 제시하지 않고 이미 선택된 vc 에 대하여 로컬 QoS 라우팅을 수행하고 있다.

임의의 네트워크에는 s,d 간에 수많은 경로가 존재할 수 있다. 또한 결정된 vc 에 따라 명백히 성능이 달라질 수 있다. 하지만 존재 가능한 경로의 수는 네트워크

의 복잡 도에 따라 무수한 경우가 존재하고 최적의 성능을 보장하는 경로를 선정하는 것을 불가능하다. 하지만 효과적으로 선정된 경로는 다른 경로에 비하여 좋은 성능을 보이게 되며 이러한 경로를 결정하는 것이 로컬 QoS 라우팅에서 매우 중요하며 본 논문에서 다루고자 하는 문제이다.

본 논문의 2절에서는 로컬 QoS 라우팅에 관한 정의와 네트워크 모델을 제시 한다. 제 3절에서는 네트워크의 특성을 고려하여 효과적인 vc를 선정하는데 기준이 되는 휴리스틱을 제시 하고 제시된 휴리스틱을 사용하여 실제 네트워크에서 경로를 선정하는 알고리즘을 선정 알고리즘을 제안한다. 제안된 알고리즘에 대한 분석과 성능평가는 제 4절에서 보이고 제 5장은 결론이다.

II. 로컬 QoS 라우팅

본 논문에서 논의되는 QoS 모델은 소스 s 와 목적 d 간에 k 개의 독립적인 경로가 있는 모델에서 시작한다. <그림 1>에서와 같이 소스 s 와 목적 d 간에는 k 개의 독립적인 경로 r_1, r_2, \dots, r_k 가 존재 한다. 각 경로 r_i 는 대역폭 용량 c_i 를 가지고 있다. 이때 전송 요구 flow 가 소스 s 에 평균 λ 로 유입되고 이 평균 정지시간은 μ 이다. 유입 flow 는 포아손 분포를 따르며 μ 는 지수 분포에 따른다. 어떤 flow 가 s 에 유입되면 목적 d 까지 각 경로에 따라 나누어져 흘러가며 각 경로에는 블로킹이 발생한다. 이 경우 블로킹 확률이 QoS 지표이고 QoS 라우팅은 블로킹 확률을 최소화 하는 flow 방법을 구하는 문제로 귀결 되며 블로킹 확률은 다음의 Erlang's Loss Formula 로 구할 수 있다^[11].

어떤 flow 에 대하여 경로 r_i 를 따라 α_i 비율의 만큼의 flow 가 흐른다고 한다면 $\sum_{i=1}^k \alpha_i = 1$ 이 된다. 이때 r_i 의 블로킹 확률 b_i 은 다음과 같다.

$$b_i = E(\nu_i, c_i) = \frac{\frac{\nu_i}{c_i}}{\sum_{n=0}^{c_i} \frac{\nu_i^n}{n!}} \quad (1)$$

이때 $\nu_i = \alpha_i \frac{\lambda}{\mu}$ 는 경로 r_i 의 평균 부하이다. 따라서 최적 QoS 라우팅을 구하는 것은 $\sum \alpha_i^* b_i^*$ 가 최소인 최적 flow 비율 α_i^* 를 구하는 것이다.

식 (1)은 k 개의 경로가 서로 독립적인 경우 성립되며 이때는 각 경로에 연결된 각 라우터는 경로상의 flow에만 영향을 받게 된다. flow 량이 변하여 상태변화가 발생하더라도 이 변화 값은 flow 변화에 따라 직접계산이 가능하므로 각 라우터가 독립적으로 QoS 상태를 변화시킬 수 있다. 하지만 일반적인 경우 각 소스 s 와 목적 d 간에 독립적인 경로만 존재 하는 것은 아니다.

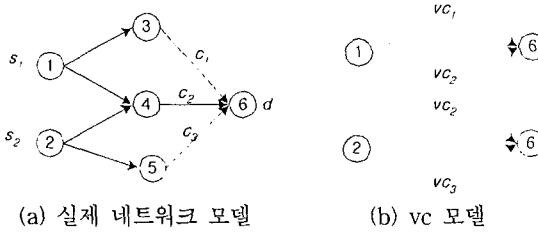


그림 2. kite 네트워크 모델
Fig. 2. Kite network model.

<그림 2>는 kite 네트워크 모델의 예를 보여준다. 실제 네트워크는 <그림 2(a)>로 일반적인 그래프 모델이다. 하지만 Erlang's Loss Formular을 적용하는 QoS 라우팅을 위하여 <그림 2(b)> 모델로 변환한다. 이때 경로 $1 \rightarrow 3 \rightarrow 6$, $2 \rightarrow 5 \rightarrow 6$ 은 서로 독립적이므로 식 (2)을 사용하여 쉽게 변환할 수 있다.

$$\begin{aligned} vc_r &= E_{vc}^{-1}(\nu_r, b_r) : \\ &= \min \{c \geq 0 : E(\nu_r, c) \leq b_r\} \end{aligned} \quad (2)$$

여기서 ν_r 은 어떤 flow 가 경로 r 에 대하여 부하 ν_r 을 흘려보내고 있을 때 관측되는 블로킹 확률이 b_r 인 경로의 가상 용량이다.

그런데 경로 $1 \rightarrow 4 \rightarrow 6$, $2 \rightarrow 4 \rightarrow 6$ 은 에지 $4 \rightarrow 6$ 을 공유한다. 이 경우 공유되는 에지를 통하여 흐르는 flow들은 에지의 대역폭을 공유하게 되며 서로 간에 영향을 미치게 된다. 따라서 에지를 공유하는 vc 가 있는 경우 블로킹 확률은 새롭게 계산되어야 하며 이 영향으로 로컬 QoS 라우팅 특성을 손상하지 않아야 한다. 만약 m 개의 $vc_i, i=1, 2, \dots, m$, 가 용량 c 를 가진 에지를 공유한다고 하면 공유에지의 블로킹 확률은

$$b = \sum_{i=1}^m E(\nu_i, c) \quad (3)$$

로 계산된다.

따라서 각 경로의 블로킹 확률은 식 (1) 과 식 (3)에 의하여 결정된다.

III. 경로선택

앞에서 설명한 바와 같이 네트워크 모델에서 가상 경로가 결정되면 로컬 QoS 라우팅을 수행할 수 있는 가상경로 모델로 쉽게 전환이 이루어진다. 하지만 특정한 토플로지의 그래프에서 에지를 공유하는 경우 소스 s 와 목적 d 까지 경로는 매우 다양한 형태로 존재 한다. 가장 복잡한 형태는 완전그래프로 네트워크가 구성되었을 때 볼 수 있다.

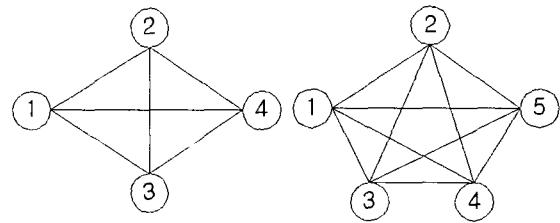


그림 3. 완전그래프 K_4, K_5
Fig. 3. Complete graph K4, K5.

<그림 3>은 노드 개수가 4인 경우와 5인 경우의 완전 그래프를 보여준다. 1번 노드를 소스로 하고 목적노드로 각각 4번, 5번을 선택하였을 때 경로를 살펴보면 다음과 같다.

표 1. K_3, K_4 의 경로

Table 1. Paths of K_3, K_4 .

	K_3	$1 \rightarrow 4$	$1 \rightarrow 2 \rightarrow 4$	$1 \rightarrow 3 \rightarrow 4$		
K_4		$1 \rightarrow 5$	$1 \rightarrow 2 \rightarrow 5$	$1 \rightarrow 2 \rightarrow 3 \rightarrow 5$	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$	
			$1 \rightarrow 3 \rightarrow 5$	$1 \rightarrow 2 \rightarrow 4 \rightarrow 5$	$1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5$	
			$1 \rightarrow 4 \rightarrow 5$	$1 \rightarrow 3 \rightarrow 4 \rightarrow 5$	$1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5$	
				$1 \rightarrow 3 \rightarrow 2 \rightarrow 5$	$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5$	
				$1 \rightarrow 4 \rightarrow 2 \rightarrow 5$	$1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5$	
				$1 \rightarrow 4 \rightarrow 3 \rightarrow 5$	$1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5$	

<표 1>에서와 같이 K_3 의 경우는 3개의 경로가 존재하고 모두 독립적이다. 따라서 별 어려움이 없이 QoS 라우팅을 위한 가상경로를 구할 수 있다. 하지만 K_4 의 경우 존재하는 경로의 수는 크게 증가 하여 16

개의 경로가 존재하고 경로 간에는 공유하는 에지가 있음을 볼 수 있다. 일반적인 K_n 의 경우 경로길이 n 인 경로의 개수가 $(n-2)!$ 로 표시됨을 쉽게 알 수 있다.

이렇게 다양한 경로가 존재 할 때 어떤 것을 가상 경로로 설정해야 하는가 하는 문제는 해답이 제시되어 있지 않다. 예를 들면 K_4 의 경우 경로 길이가 1인 경우와 2인 4개의 경로를 가상경로로 설정할 수 있다. 이때 4개의 경로는 서로 독립적임을 쉽게 확인할 수 있다. 여기에 경로길이 3인 $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$ 경로를 추가하여 5개의 경로를 사용하거나 16개의 경로를 모두 사용하는 것도 가능하다.

문제는 모든 경로를 사용할 때 존재 가능한 경로의 수 때문에 다항식의 복잡도로 QoS 라우팅을 할 수 없을 뿐 아니라 경로의 수가 많아질수록 계산량이 증가한다는 것이다. 따라서 적절한 범위에서 경로를 선정하여야 할 필요성이 있다.

본 논문에서는 효율적인 경로선택을 위한 휴리스틱으로 다음의 사항들을 고려하였다.

- 경로길이: 통상적으로 경로길이가 짧은 경우 좋은 성능을 보장한다. 또한 불필요하게 자원이 낭비되는 경유를 배제할 수 있다. 본 논문에서 고려되는 경로 길이는 특정한 값을 가지지 않으므로 흡을 사용한다.
- 공유링크: 앞에서 설명한 바와 같이 공유링크는 문제를 복잡하게 하며 가상경로의 블로킹 확률은 특별한 경우를 제외하고는 공유링크의 블로킹 확률에 제한된다. 따라서 선택되는 경로에 공유링크가 적을수록 좋은 성능을 기대할 수 있다.
- 경로의 수: 선택되는 경로의 수가 많으면 flow 가 경로에 분산되므로 성능이 좋아지는 것은 명백하다. 하지만 일정 이상의 경로는 실제 flow 전송에 기여를 하지 않고 QoS 라우팅을 위한 계산을 더욱 복잡하게 한다.
- 자원분배: 네트워크에는 하나이상의 flow가 전송되게 된다. 따라서 가용한 자원이 균등하게 사용되도록 하여야 한다. 하지만 이 정도는 QoS 라우팅의 성능과는 별개이다

효율적인 경로의 선정을 위하여 위의 휴리스틱이 적절히 고려되어야 한다. 최적의 경로를 결정하거나 모든 휴리스틱을 동시에 고려하는 것은 불가능 하므로 특정한 휴리스틱에 우선순위를 두는 경로선정 알고리즘을

제안하였다.

네트워크는 $N(V, E)$ 의 그래프로 모델링 되며 $V(N), E(N)$ 은 각각 그래프 N 의 노드 집합과 에지 집합을 나타낸다.

```
Procedure: disjointPATH  $N(V, E), s, d$ 
Path(i) = shortestPATH( $N, s, d$ );
if Path(i) =  $\emptyset$  then END Procedure;
else
 $N^{(i)} = N \cap \overline{E}(\text{Path}(i)), i=i+1;$ 
disjointPATH( $N^{(i)}, s, d$ );
end if
end
```

알고리즘 *disjointPATH*는 네트워크에서 최소비용의 독립적인 경로를 선정하도록 하는 휴리스틱을 적용하였다.

알고리즘의 동작은 주어진 네트워크 $N(V, E)$ 에서 s 와 d 간의 최소비용경로 하나를 잘 알려진 Dijkstra의 알고리즘을 사용한 *shortestPATH*로 구한다. 이때 구해진 *Path(i)*의 에지를 $N(V, E)$ 에서 제거한 후 얻어진 네트워크에서 다시 과정을 반복한다. 알고리즘은 최단경로를 찾을 수 없을 때 종료된다.

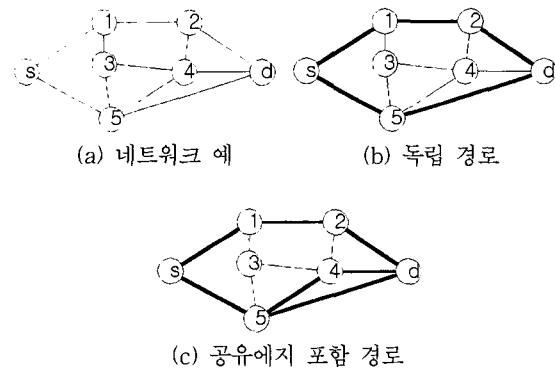


그림 4. 예제 네트워크

Fig. 4. An example network.

*disjointPATH*를 <그림 4(a)>의 네트워크에 적용시켜 보면 최초의 경로 $s \rightarrow 5 \rightarrow d$ 를 구한다. 이때 경로에 포함된 링크는 제거되고 $s \rightarrow 1 \rightarrow 2 \rightarrow d$ 경로를 찾는다. 이때 경로의 링크를 제거하면 s 와 d 사이에는 어떤 경로도 존

재하지 않고 <그림 4(b)>의 독립된 2개의 경로를 구하게 된다.

알고리즘의 복잡 도는 shortestPATH 가 $O(n^2)$ 값을 가지고 이를 최대 소스에 연결된 에지의 수만큼 반복하므로 $O(n^3)$ 가 된다.

이 방식으로 구한 경로는 일반적인 경우에는 좋은 성능을 보이지만 결합도(connectivity) 가 약한 네트워크의 경우 독립적으로 존재하는 경로의 수가 매우 적게 된다. 이러한 경우에는 좋은 성능을 보장할 수 없고 공유에지의 사용이 필연적이다.

공유에지를 사용하는 경우에는 공유에지의 공유정도를 제한하여야 할 필요가 있다.

```
Procedure: findPATH N(V, E), s, d, k
Initialize i=0;
Path(i) = shortestPATH(N, s, d);
p = |E(Path(i))|+k;
do
    Path(i) = shortestPATH(N, s, d);
    l =  $\sum_{\forall E_j \in \text{Path}(i)} |E_j|, i=i+1$ ;
    if l > p then END Procedure;
     $|E_j| = |E_j| + 1$  for  $\forall E_j \in \text{Path}(i)$ ;
end do
end
```

경한 뒤 다시 알고리즘을 적용하면 $s \rightarrow 1 \rightarrow 2 \rightarrow d$ 가 최단 경로이다. 다음단계에서는 거리비용이 4인 $s \rightarrow 5 \rightarrow 4 \rightarrow d$ 를 구한다.

알고리즘의 복잡 도는 k 값과 관련 있는데 일반적인 경우 지수함수의 복잡 도를 가진다.

*findPATH*의 단점은 거리가 먼 독립된 경로가 배제될 수 있다는 점이다. 즉 최단경로와 k 의 합 보다 큰 흡의 경로는 찾지 못하게 된다. 독립된 경로는 어떠한 경우에도 성능에 좋은 영향을 보이게 되므로 이를 찾는 것은 매우 중요하다. 이 경우 *disjointPATH*를 병행하여 수행하면 해결이 가능하다.

```
Procedure: localPATH N(V, E), s, d, k
Path(i) = shortestPATH(N, s, d);
if Path(i) =  $\emptyset$  then END Procedure;
else
    for  $\forall E_j \in \text{Path}(i)$ 
         $|E_j| = |E_j| + 1$ ;
        if  $|E_j| \geq k$  then  $N = N \cap \overline{E_j}$ ;
    end for
    i=i+1;
    localPATH( N , s, d );
    end if
end
```

findPATH 는 *disjointPATH*와 동일한 휴리스틱을 적용하였지만 공유 에지를 포함하는 경로를 선정할 수 있도록 구성되었다. 알고리즘은 최초의 최단경로를 *shortestPATH*로 구한다. *disjointPATH* 에서는 이때 구해진 경로의 링크를 제거하여 공유에지의 발생을 봉쇄했지만 여기서는 에지의 값을 흡의 비용과 같이 1 증가시켰는데 이 값을 적절히 변화 시키면 또 다른 경로 선정방식에 활용 될 수 있다.

따라서 그래프의 형태는 변하지 않고 에지 값만 변함에 다시 반복 수행된다. 이 경우 증가된 에지 값은 공유 경로수와 같다. 또한 p는 구하는 경로의 길이를 제한하는데 이 값은 최소경로에 k 만큼의 여유를 허용한다. 따라서 p 값 이내로 구할 수 있는 경로가 존재하지 않으면 알고리즘은 종료된다.

<그림 4(a)>에 알고리즘을 적용하면 최초의 경로는 $s \rightarrow 5 \rightarrow d$ 로 구해진다. 이때 경로에 포함된 에지 값을 변

*localPATH*는 공유에지의 수를 우선적 휴리스틱으로 사용하였다. 공유에지가 발생하면 이 에지가 다른 경로와 얼마나 공유되었는지를 알아낸 다음 제한 값 이상의 공유가 발생하면 네트워크에서 제거하여 다시 공유될 수 있도록 한다.

<그림 4>의 예에서는 k가 2일 경우 $s \rightarrow 5 \rightarrow d$, $s \rightarrow 1 \rightarrow 2 \rightarrow d$, $s \rightarrow 5 \rightarrow 4 \rightarrow d$ 를 순서대로 구한 후 $s \rightarrow 5$ 링크를 제거한다. 하지만 그래프에는 s와 d 사이에 다른 경로가 존재하므로 알고리즘은 계속된다.

알고리즘의 복잡 도는 k 값에 따라 결정된다. k 값이 1인 경우 공유에지는 존재할 수 없고 *disjointPATH* 와 같이 $O(n^3)$ 의 복잡 도를 가진다.

일반적인 경우 네트워크의 결합도(connectivity)에 따라 k의 함수가 되는데 완전그래프의 경우가 가장 복잡하고 존재 가능한 경로의 수는 $(k-1)!$ 이다. 따라서 일반적인 경우 복잡 도는 $O((k-1)!n^3)$ 으로 계산되는데 실제의 경우 k 값은 2, 3 정도의 정수로 한정되어 이 경

우에는 disjointPATH 와 복잡도가 같다.

제안된 알고리즘은 단일 소스인 경우에 적용된다. 본 연구에서 제시된 QoS 모델은 복수개의 소스가 허용되는데 이 경우에는 복수개의 소스를 모두 연결하여 단일 소스의 모델로 쉽게 변환이 가능하다.

IV. 성능평가

제안된 알고리즘의 성능을 평가하기 위하여 [11]에서 사용된 다음의 네트워크 모델을 사용하였다.

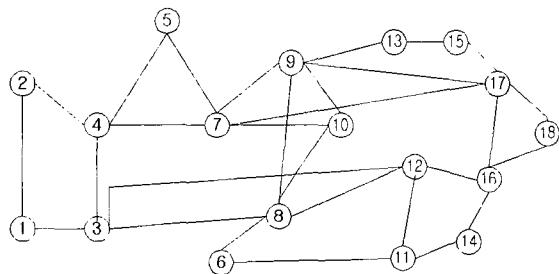


그림 5. 성능평가를 위한 ISP 네트워크

Fig. 5. The isp network used in performance evaluation.

문제를 단순화하기 위하여 모든 링크는 양 방향으로 전송이 가능하고 각각의 링크 동일하게 C의 용량을 가진다고 가정한다. 소스에는 평균 λ 의 flow 가 포아손 분포에 따라 도착하고 목적노드 까지 해당되는 경로를 통하여 나누어 전송된다. 이때 각 흐름의 자연시간은 평균 $\frac{1}{\mu}$ 값을 가지며 지수분포에 따른다. [11] 에서와 같

이 네트워크에 부과되는 부하가 $\rho = \lambda N \bar{h} / \mu L C$ 로 표시 된다. 여기서 N은 소스 노드의 수이며 \bar{h} 는 모든 flow의 평균 경로 길이를 나타낸다. 실험에 사용된 각종 변수는 $C = 20$, $N = 18$, $L = 60$, $\bar{h} = 2.36$, $\mu = 60sec$ 로 하였다. 소스에 공급되는 flow의 λ 값은 부하의 변화에 따라 조정되었다.

QoS 라우팅 방법은 [11]에서 제안한 psr(proportional sticky routing)을 사용하였다.

<표 2>는 <그림 4>의 네트워크를 대상으로 각 알고리즘에 대하여 임의로 소스와 목적을 결정하여 구한 경로에 대한 결과이다. 단 소스와 목적이 직접적으로 연결된 경우는 제외 하였다. disjointPATH 의 경우는 구해

진 경로는 모두 독립적이므로 예상대로 적은 수의 경로를 구하였다. findPATH, 와 localPATH 의 경우 k를 3로 제한하였는데 구하여진 경로의 수는 findPATH 가 많았지만 localPATH에서 구한 결과가 길이도 짧고 공유에지의 수도 적음을 알 수 있다.

표 2. 알고리즘의 경로 성능 비교

i = 경로개수의 평균

\bar{L}_p = 경로 길이의 평균

\bar{E}_p = 공유에지 개수

Table 2. Path comparison of algorithms.

	1 source			2 source			3 source		
	i	\bar{L}_p	\bar{E}_p	i	\bar{L}_p	\bar{E}_p	i	\bar{L}_p	\bar{E}_p
d_PATH	2.1	3.1	0	2.3	2.7	0	2.5	2.4	0
f_PATH	3.9	3.6	1.9	3.9	3.5	2.1	4.1	3.3	2.0
l_PATH	3.8	3.3	1.6	3.8	3.2	1.7	3.8	3.1	1.7

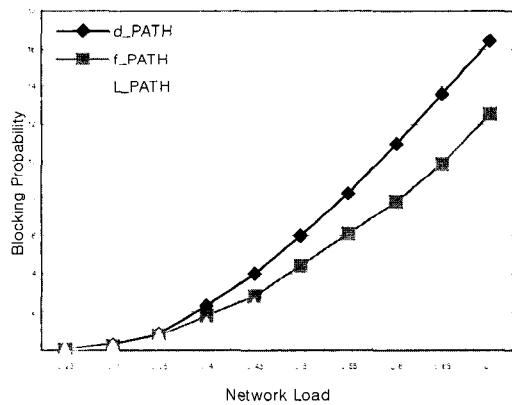


그림 6. 알고리즘에 따른 성능평가

Fig. 6. Performance comparison of algorithms.

<그림 6>는 부하의 증가에 따른 블로킹 확률의 변화를 보여준다. 이 결과는 경로선택 알고리즘이 성능에 어떤 영향을 보이는지 직접적으로 보여 주는데, 우선 disjointPATH 는 예측대로 제한된 경로의 수로 인하여 좋지 않은 결과를 보인다. 하지만 findPATH는 <표 2>에서와 같이 localPATH 에 비하여 더 많은 경로를 선정하였음에도 localPATH 에 비하여 결과가 좋지 않다. 이결과로 경로의 길이 보다 공유에지의 수가 성능에 더 큰 영향을 미치는 것으로 볼 수 있다.

<그림 7>은 <그림 5>의 네트워크에 대하여 k를 변화시켜 localPATH에서 더 많은 경로를 선정 할 수

있도록 하여 블로킹 확률의 변화를 보여 준다. 결과에서 와 같이 k 가 증가하면 성능이 항상 되는 것을 확인할 수 있는데 일정 이상의 k 값에는 성능향상이 잘 이루어 지지 않음을 볼 수 있다. 실험에서는 k 가 5인 경우 k 가 3인 경우와 비교하여 2% 정도의 성능향상을 보였는데 계산시간은 4배 이상 소요 되었다.

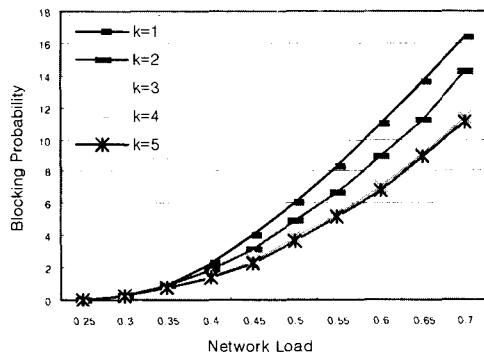


그림 7. k 의 변화에 따른 localPATH 의 성능평가
Fig. 7. Performance of localPATH with variable k .

V. 결 론

본 논문에서 우리는 로컬 QoS 라우팅을 위한 경로선택 문제를 다루었다. 로컬 QoS 라우팅은 가상의 vc를 통하여 독립적으로 QoS 라우팅을 수행하기 때문에 각 라우터간의 상태교환이 필요 없고 상태교환에 필요한 자원의 소모가 없어서 성능에 나쁜 영향을 끼치지 않는다. 하지만 주어진 네트워크에서 적절한 vc를 선정하는 데는 매우 어려운 점이 있다. 실제로 특정의 네트워크에서 존재 가능한 경로의 수는 매우 많으며 경로가 많다 하더라도 좋은 성능을 보장하지 않음을 실험적으로 보였다.

특정한 네트워크에 대하여 존재 가능한 모든 경우의 경로를 조사하여 성능을 평가하기는 불가능 하기 때문에 좋은 성능을 보장하는 결로 선정 알고리즘이 필요하다.

본 논문에서 최소경로 우선 휴리스틱과 공유에지를 최소화 하는 휴리스틱을 적용한 경로 선택 알고리즘을 제안 하였는데 공유에지의 수를 특정한 정수 값으로 제한하면 $O(n^3)$ 의 복잡도로 가상경로를 구할 수 있다.

주어진 네트워크에 대하여 여러 가지 성능평가를 수행하여 공유에지를 최소화 하고 경로의 길이를 최소화

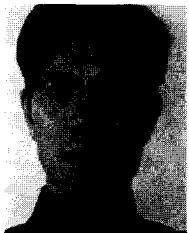
하는 휴리스틱이 매우 잘 동작함을 보였다. 또 경로의 길이보다도 공유에지가 성능에 더욱 영향을 미친다는 것을 확인할 수 있었다. 또한 경로의 수가 많으면 성능이 증가할 것으로 예측되는데 일정 이상의 경로가 있는 경우에는 경로수의 증가에 따른 성능개선의 효과는 미미한 것을 알 수 있었다.

참 고 문 헌

- [1] G. Apostopoulos, Roch Guerin, Sanjay Karnat, Satish K. Tripathi, "Quality of Service Based Routing: A Performance Perspective", in Proceedings of ACM SIGCOMM, September 1998, pp. 17-28.
- [2] S. Chen and K. Nahrstedt, "Distributed QoS Routing with Imprecise State Unformation." ICCN'98, October 1998.
- [3] J. Chen, P. Druschel, D. Subramanian, "A New Approach to Routing with Dynamic Metrics", IEEE INFOCOM 1999.
- [4] C. Aurrecoechea, Andrew T Campbell, Linda Hauw, "A survey of QoS architectures", in ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, Vol. 6, No. 3, May 1998, pp. 138-151.
- [5] Y. Zhong and X. Yuan, "Impact of Resource Reservation on the Distributed Multi-path Quality of Service Routing Scheme." The Eighth International Workshop on Quality of Service (IwQoS2000), Pittsburgh, PA, June 2000, pp. 95-104.
- [6] X. Yuan and X. Liu, "Heuristic Algorithms for Multi Constrained Quality of Service Routing", IEEE INFOCOM, April 2001.
- [7] Zheng Wang, Jon Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", in IEEE JSAC, Vol. 14, No. 7, September 1996, pp. 1288-1234.
- [8] S. Keshav, R. Sharma, "Issues and trends in router design", In IEEE Communications Magazine, May 1998, Vol. 36, No. 5, pp. 144-151.

- [9] E. Crawley, R. Nair, B. Rajagopalan, H. andick, "A framework for QoS-Based Routing in the Internet", Request For Comment 2386, Category: Informational, August 1998.
- [10] Qingming Ma, Peter Steenkiste, "Quality of Service Routing for Traffic with Performance Guarantees", in IFIP Fifth International Work shop on Quality of Service, NY, May 1997, pp. 115-126.
- [11] S. Nelakuditi, Zhi-Li Zhang, and Rose P. Tsang, "Adaptive Proportional Routing: A Localized QoS Routing Approach", In IEEE Infocom, Trans on Networking, Vol. 10NO. 06, Dec. 2002, pp. 790-804.

저자소개



徐 庚 龍(正會員)
 1983년 : 부산대학교 전기기계공학
 과(공학사). 1990년 : 한국과학기술
 원(KAIST) 전자공학과(공학석사).
 1995년 : 한국과학기술원(KAIST)
 전자공학과(공학박사). 1991년~현
 재 : 부경대학교 전자컴퓨터 정보통
 신공학부 부교수. <주관심분야 : 분산시스템, 컴퓨터 네
 트워크>