

구간 추정 기반의 지연시간을 고려한 저비용 유니캐스트 라우팅 방식

김 문 성[†] · 방 영 철^{††} · 추 현 승^{†††}

요 약

멀티미디어 응용 서비스에서는 특정 시간 내에 데이터 전송이 이루어져야 하는 시간 의존성이 있다. 이러한 실시간 특성은 네트워크의 QoS보장을 위한 중요한 요소이다. 네트워크 사용자의 증가와 응용 프로그램의 데이터 전송율의 증가로 네트워크 자원을 효율적으로 사용하기 위한 연구는 계속 진행되고 있다. 종단간(End-to-End) 지연시간 제한 조건을 만족하면서 최소 비용을 갖는(Delay Constrained Least Cost, DCLC) 경로를 찾는 문제는 이미 NP-hard 문제로 알려져 있다. 최소 지연시간 경로의 비용은 최소 비용 경로의 비용보다 상대적으로 높은 경로 비용을 갖으며, 역으로 최소 비용 경로의 지연시간은 최소 지연시간 경로의 지연 시간보다 상대적으로 높은 지연시간을 갖는다. 본 논문에서는 이러한 단점을 극복하여 DCLC문제에 접근하기 위해 링크비용과 지연시간을 확률적으로 조합한 인자를 사용한 새로운 알고리즘을 연구하였다. 최근 Salama에 의해 제안된 DCUR 알고리즘은 최적에 가까운 알고리즘이나, 제한된 알고리즘은 DCUR 알고리즘과 비교하여 종합적인 컴퓨터 시뮬레이션 결과에 의하면 노드 수 200에서 38% 이상의 효과를 보였다. 본 알고리즘의 특징은 선택의 요소로서 새로운 인자를 만들었고, 링크를 순차적으로 선택하지 않고 동적으로 선택하는 방법을 구현하였다는 것이다.

Low Cost and Acceptable Delay Unicast Routing Algorithm Based on Interval Estimation

Moonseong Kim[†] · Young-Cheol Bang^{††} · Hyunseung Choo^{†††}

ABSTRACT

The end-to-end characteristic is an important factor for QoS support. Since network users and required bandwidths for applications increase, the efficient usage of networks has been intensively investigated for the better utilization of network resources. The distributed adaptive routing is the typical routing algorithm that is used in the current Internet. The DCLC(Delay Constrained Least Cost) path problem has been shown to be NP-hard problem. The path cost of LD path is relatively more expensive than that of LC path, and the path delay of LC path is relatively higher than that of LD path in DCLC problem. In this paper, we investigate the performance of heuristic algorithm for the DCLC problem with new factor which is probabilistic combination of cost and delay. Recently Dr. Salama proposed a polynomial time algorithm called DCUR. The algorithm always computes a path, where the cost of the path is always within 10% from the optimal CBF. Our evaluation showed that heuristic we propose is more than 38% better than DCUR with cost when number of nodes is more than 200. The new factor takes in account both cost and delay at the same time.

키워드 : DCLC 문제(Delay Constrained Least Cost(DCLC) Problem), DCUR 알고리즘(DCUR Algorithm), 구간추정(Interval Estimation)

1. 서 론

멀티미디어 기술은 네트워크의 고속화와 더불어 많은 인터넷 응용 기술들을 발생시킨다. 화상회의, 주문형 미디어 서

비스, 인터넷 방송, 원격 쇼핑, 원격 교육 등과 같은 멀티미디어 서비스가 보편화됨에 따라 실시간 멀티미디어 기술은 그 중요성이 매우 커지게 되었다. 멀티미디어 응용 서비스에서는 특정 시간 내에 데이터 전송이 이루어져야 하는 시간 의존성이 있다. 이러한 실시간 특성은 네트워크의 QoS(Quality of Service)보장을 위한 중요한 요소이다.

현재 인터넷에서 사용하고 있는 대표적인 라우팅 알고리즘으로 분산 적응식 경로 배정(Distributed adaptive rout-

* This paper was supported in part by Brain Korea 21 and University ITRC project. Dr. H. Choo is the corresponding author.

† 준 회원 : 성균관대학교 대학원 정보통신공학부

†† 정 회원 : 한국산업기술대학교 교수

††† 종신회원 : 성균관대학교 정보통신공학부 교수

논문접수 : 2003년 9월 5일, 심사완료 : 2004년 2월 3일

ing)이 있다. 이는 거리 벡터(distance vector)를 이용하는 RIP (Routing Information Protocol)[1]와 링크 상태(link state) 정보를 이용하는 OSPF(Open Shortest Path First)[2]로 나누어진다. 거리 벡터 경로 배정은 Bellman-Ford 알고리즘 기반이며, 링크 상태 경로 배정은 Dijkstra의 최단 경로 알고리즘으로 계산 가능한 Shortest Path First(SPF)[3] 알고리즘이다.

종단간(End-to-End) 지연시간 제한 조건을 만족하면서 최소 비용을 갖는(Delay Constrained Least Cost, DCLC) 경로를 찾는 문제는 이미 NP-hard 문제로 알려져 있다[4]. 이러한 문제를 다루었던 Widyono의 논문에는 CBF(Constrained Bellman-Ford)[5]라는 알고리즘이 제안되었는데, 이것을 이용하면 최적의 DCLC 경로를 찾을 수 있다. 그러나 각 노드마다 네트워크 위상에 대한 완전한 정보를 유지해야 하고, 네트워크 규모가 커짐에 따라 시간복잡도가 지수 승에 비례하여 증가하는 단점이 있다. 최근 Salama에 의해 제안된 DCUR(Delay Constrained Unicast Routing)[6] 알고리즘은 CBF와 비교하여 10%이내의 비용적 성능을 보여 주었기에 최적에 가까운 알고리즘이다. 그러나 알고리즘의 단순성을 위하여 임의의 노드에서 경로 선정시 목적지 노드까지의 최소 비용 경로나 최소 지연시간 경로상의 다음 노드로만 제한을 하여 라우팅 경로의 비용측면에서 다소 비효율적이다.

본 논문에서는 DCUR과 비교하여 비용면에서 우월성을 가능하였다. 노드 수가 200개 이상일 때에는 무려 38% 이상의 효과를 보였다. 본 논문의 구성은 다음과 같다. 2장에서는 네트워크의 모델 및 구간 추정에 대해 간단히 알아보고, 3장에서는 제안된 알고리즘에 대한 소개와 예를 기술한다. 4장에서는 시뮬레이션에 대한 설명과 그 결과에 대해 분석하고 5장에서 본 논문의 결론을 제시한다.

2. 관련 연구

2.1 네트워크 모델

네트워크 모델은 방향성 그래프 $G(V, E)$ (노드들의 집합 V , 링크(link or edge)들의 집합 E)로 표현한다. 링크 $(i, j) \in E$ 의 링크비용(link cost)은 $c_{(i, j)}$ 로 지연시간(delay)은 $d_{(i, j)}$ 로 표현한다. 네트워크 G 에서 노드 u 로부터 노드 v 까지의 경로(path)란 링크들 $(u, i), (i, j), \dots, (k, v)$ 이 있을 때, 노드를 나열한 u, i, j, \dots, k, v 를 말한다. 이것을 $P(u, v) = \{(u, i), (i, j), \dots, (k, v)\}$ 로 표현한다. 한 경로 상에 있는 모든 노드들이 서로 다를 때, 그 경로를 단순경로(simple path)라 한다. 단순경로의 길이는 경로 상에 있는 링크들의 수이다. 이것을 $n(P(u, v))$ 으로 정의 한다. 또한 $\langle u, v \rangle$ 상에서 $(a, b) \langle (c, d) \leftrightarrow n(P(u, b)) \leq n(P(u, d))$ 으로 정의된 이항 연산이라 하자 ($\forall (a, b), (c, d) \in P(u, v)$). 그러면 $(P(u, v), \langle$

은 완전 순서 집합(totally ordered set)이다. 주어진 시작 노드 $s \in V$ 와 목적지 노드 $d \in V$ 에 대하여, $2^{s \rightarrow d}$ 를 s 에서 d 로 갈 수 있는 모든 경로들의 집합으로 나타내겠다.

$$2^{s \rightarrow d} = \{P_k(s, d) \mid \text{all possible paths from } s \text{ to } d, \forall s, d \in V, \forall k \in \Lambda\}$$

여기서 Λ 는 첨자들의 집합(index set)이다.

임의의 경로 P_k 의 비용과 지연시간은 정의역 $2^{s \rightarrow d}$ 에서 공역 양의실수(\mathbf{R}^+)로의 함수로서 다음과 같이 정의한다. 우선 (P_k, \langle) 가 완전 순서 집합이므로 함수 f_k 를 다음과 같이 정의한다면, 경로 P_k 와 집합 $N_{n(P_k)}$ 는 동형(isomorphism)이며 함수 f_k 는 1-1 대응함수(bijection)가 된다.

$$P_k = \{(u, i), (i, j), \dots, (m, v)\}$$

$$\begin{matrix} f_k \\ \mapsto \end{matrix} N_{n(P_k)} = \{1, 2, \dots, n(P_k)\}$$

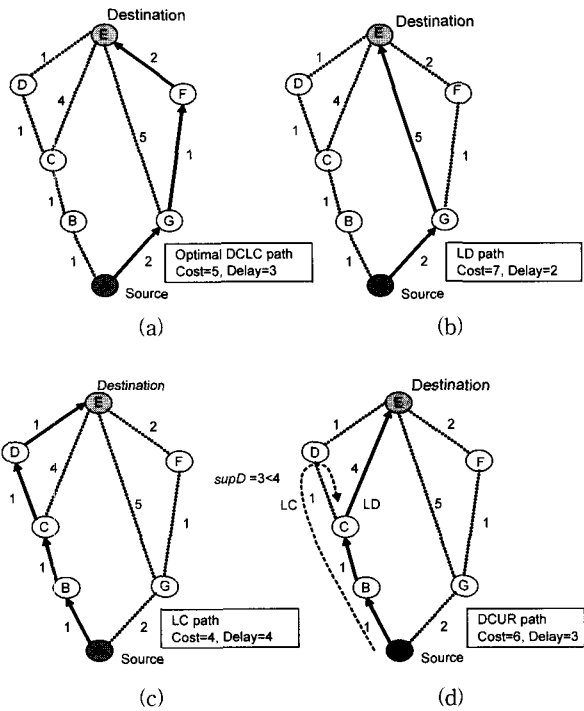
따라서, 경로 비용 함수는 $\phi_C(P_k) = \sum_{r=1}^{n(P_k)} c_{r^{-1}(r)}$ 이며 경로 지연시간 함수는 $\phi_D(P_k) = \sum_{r=1}^{n(P_k)} d_{r^{-1}(r)}$ 이다(여기서, $\forall P_k \in 2^{s \rightarrow d}$).

종단간 지연시간 제한 조건을 $supD$ 로 표현한다면 DCLC 경로를 찾는 문제는 다음과 같이 표현이 가능하다. 경로 P_k 가 $supD$ 의 조건을 만족하는 경로들의 집합을 $2^{s \rightarrow d} (C_2^{s \rightarrow d})$ 로 표현하면, $\min \{\phi_C(P_k) \mid P_k \in 2^{s \rightarrow d}, \forall k \in \Lambda\}$ 를 만족하는 경로 P_k 를 찾는 문제이다.

2.2 선행 알고리즘

(그림 1)의 (d)는 DCUR 알고리즘을 보여 주고 있다. 먼저, 시작 노드 A는 목적지 노드 E를 향해 LD 경로를 찾아 지연시간이 $supD$ 의 조건을 만족하는지 판단 후, 만족하므로 ($2 \leq supD$) LC 경로를 찾아 다음 노드의 후보 B를 선택한다. 또 다시 노드 B에서 목적지 노드 E를 향해 LD 경로를 찾아 지연시간이 잔여 지연시간($supD - 1$)의 조건을 만족하는지 판단 한다. 이것 또한 만족하므로 ($2 \leq supD - 1$) 노드 E를 향한 LC 경로를 찾아 다음 노드의 후보 C를 선택한다. 같은 방법으로 계속하여 다음 노드의 후보로서 D가 선택되었다고 하자. 노드 D에서 LD 경로를 찾아 잔여지연시간의 조건을 만족하는지 판단한 결과 적합하지 못함을 발견하고 노드 C에서 다음 노드의 후보로서 LC 경로를 활용하지 않고 LD 경로를 활용하여 노드 E를 선택하게 되고, 노드 E는 목적지 노드로서 알고리즘이 완료된다. 따라서 $P_{DCUR}(A, E) = \{(A, B), (B, C), (C, E)\}$ 이며 $\phi_D(P_{DCUR}) = 3$ 으로 지연시간 제한 조건 3을 만족하게 된다. 그에 따른 경로 비용은 $\phi_C(P_{DCUR}) = 6$ 을 얻었다. 그러나 최적의 경로는 $P_{Optimal}(A, E) = \{(A, G), (G, F), (F, E)\}$ 이며 경로비용은 ϕ_C

($P_{Optimal}$) = 5으로 $\phi_C(P_{DCUR}) = 6$ 과는 차이가 있다.



(그림 1) 선행 알고리즘 : 시작 노드 A에서 목적지 노드 E까지 지연시간제한 조건은 3, 링크비용은 표시된 값, 지연시간은 모두 1로 가정

DCUR의 단점은, 순차적으로 각 노드마다 $supD$ 의 조건을 만족만 한다면 무조건 LC 경로를 선택한다는 것에 있다. 목적지 노드에 다가서기 전에 잔여 지연시간의 조건이 갑자기 줄어들면 그때부터는 LD 경로의 선택에 비중을 둘 것이기 때문이다. 따라서, 본 논문에서는 단순히 LC 경로와 LD 경로 두 가지만 노드 선택의 인자(factor)로 둘 것이 아니라 새로운 인자를 만들었으며, 순차적인 노드의 선택이 아니라 동적인 노드의 선택을 제안하였다.

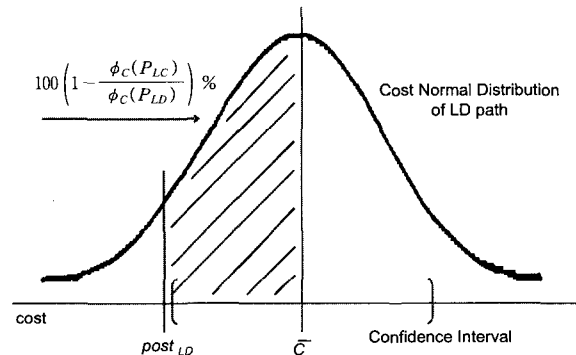
3. 제안하는 알고리즘

제안하는 알고리즘은 시작 노드에서 목적지 노드까지의 경로를 지연시간 제한 조건 $supD$ 를 만족하게끔 하는 송신 측 초기화(source-initiated) 라우팅 방법이다. 시작 노드(s)는 목적지 노드(d)까지의 경로로서 최초에 P_{LC} 와 P_{LD} 두개의 경로를 가상으로 구축한다. 지연시간 제한조건으로서 $supD$ 를 받았다면, 가상으로 구축한 두개의 경로를 사용해서 $\phi_D(P_{LD})$ 와 $\phi_D(P_{LC})$ 을 계산한다. 물론 $\phi_D(P_{LD})$ 의 값이 $supD$ 보다 크다면 요구하는 경로는 찾을 수 없다.

3.1 새로운 인자

먼저, 경로 P_{LD} 가 선택되었다고 가정하자. 일반적으로 지연

시간을 우선적으로 선택한 경로이기에 $\phi_C(P_{LD}) \geq \phi_C(P_{LC})$ 이다. 그렇다면 경로 P_{LD} 는 경로 비용을 $100 \left(1 - \frac{\phi_C(P_{LC})}{\phi_C(P_{LD})} \right) \%$ 줄인다면 $\phi_C(P_{LC})$ 가 된다. 따라서 경로 P_{LD} 의 링크에 표현된 링크비용(link cost)들을 재 측정 하도록 한다. 그러기 위해서, 경로 P_{LD} 의 링크비용들의 산술평균 $\bar{C} = \frac{\phi_C(P_{LD})}{n(P_{LD})}$ 을 계산 한다. 여기서 $100 \left(1 - \frac{\phi_C(P_{LC})}{\phi_C(P_{LD})} \right) \times 2\%$ 의 신뢰구간을 적용시키기 위해 그것의 백분위수(percentiles)를 계산한다. 만약 줄이기를 원하는 값이 50% 이상이 된다면 그 값은 99.9%의 신뢰구간으로 해석하였다. 이유는 정규분포(normal distribution)의 확률밀도함수(probability density function)는 평균에 대칭이어서 50%를 넘어서면, 이 때 2배를 하면 100%를 넘기 때문이다. (그림 2)를 보기 바란다.



(그림 2) $post_{LD}$ 의 표현

(그림 2)에서 $post_{LD}$ 를 표현하였다. $post_{LD}$ 는 지연시간의 인자를 새롭게 바꿀 기준점인 것이다. 그렇기 때문에 백분위수를 찾는 것은 필연적인 것이다. 백분위수를 구하기 위해 누적분포함수(cumulative distribution function)를 사용하였다. 이상적으로는 이산적(discrete)으로 구현해야 하지만, 여기서는 계산의 복잡성을 피하기 위해 모두 연속함수로 표현을 하였다.

$$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$$

따라서, 다음 방정식의 해 $z_{a/2}$ 가 백분위수가 되는 것이다.

$$F(z_{a/2}) - \frac{1}{2} = 1 - \frac{\phi_C(P_{LC})}{\phi_C(P_{LD})}$$

즉, $z_{a/2} = F^{-1} \left(\frac{3}{2} - \frac{\phi_C(P_{LC})}{\phi_C(P_{LD})} \right)$ if $100 \left(1 - \frac{\phi_C(P_{LC})}{\phi_C(P_{LD})} \right) \%$

< 50%이다. 백분위수의 계산을 <표 1>에 기록하였다.

〈표 1〉 백분위수

$\eta = [100(1 - \frac{\phi_C(P_{LD})}{\phi_C(P_{LC})})] \%$ The function [x] gives the integer closest to x.						50 이상 $\eta : 3.29$			
η	$Z_{\alpha/2}$	η	$Z_{\alpha/2}$	η	$Z_{\alpha/2}$	η	$Z_{\alpha/2}$	η	$Z_{\alpha/2}$
49	2.33	48	2.05	47	1.88	46	1.75	45	1.65
44	1.56	43	1.48	42	1.41	41	1.34	40	1.28
39	1.23	38	1.18	37	1.13	36	1.08	35	1.04
34	0.99	33	0.95	32	0.92	31	0.88	30	0.84
29	0.81	28	0.77	27	0.74	26	0.71	25	0.67
24	0.64	23	0.61	22	0.58	21	0.55	20	0.52
19	0.50	18	0.47	17	0.44	16	0.41	15	0.39
14	0.36	13	0.33	12	0.31	11	0.28	10	0.25
09	0.23	08	0.20	07	0.18	06	0.15	05	0.13
04	0.10	03	0.08	02	0.05	01	0.03	00	0.00

백분위수를 구했다면, $post_{LD}$ 의 값을 계산한다. $post_{LD} = \bar{C} - z_{\alpha/2} \frac{S_{LD}}{\sqrt{n(P_{LD})}}$. 여기서 S_{LD} 는 표본표준편차로서, $S_{LD} = \sqrt{\frac{1}{n(P_{LD})-1} \sum_{r=1}^{n(P_{LD})} (c_{f_k(r)} - \bar{C})^2}$ 의 구조를 갖는다. 함수 f_{LD} 는 2.1절에서 소개하였다. 또한 $n(P_{LD})=1$ 일 경우는 분모에 1을 감(減)하지 않겠다. 여기까지 기준점인 $post_{LD}$ 의 값을 구하였고, 링크비용(link cost)을 이용한 새로운 인자를 만들겠다.

$$Cfct_{(i,j)}(c_{(i,j)}) = \max\{1, 1 + (c_{(i,j)} - post_{LD})\}$$

둘째로, 경로 P_{LC} 가 처음에 설정되었다고 가정하자. 만일 $\phi_D(P_{LC}) > supD$ 이면, $\phi_C(P_{LC})$ 은 최적 값이므로 우리의 관심은 단지 $\phi_D(P) \leq supD$ 을 만드는 경로 P 를 찾는 데 있다. 일반적으로 비용을 우선적으로 선택한 경로이기에 $\phi_D(P_{LC}) \geq \phi_D(P_{LD})$ 이다. 그렇다면 경로 P_{LC} 는 경로 지연시간을 $100(1 - \frac{\phi_D(P_{LD})}{\phi_D(P_{LC})})\%$ 줄인다면 $\phi_D(P_{LD})$ 가 된다. 첫 번째와 같은 방법으로 $post_{LC}$ 를 구하고 새로운 인자를 다음과 같이 만들었다.

$$Dfct_{(i,j)}(d_{(i,j)}) = \max\{1, 1 + (d_{(i,j)} - post_{LC})\}$$

경로 각각의 링크 마다 링크비용(link cost)과 지연시간(delay)을 대변하는 $Cfct_{(i,j)}(c_{(i,j)})$, $Dfct_{(i,j)}(d_{(i,j)})$ 의 값을 구하고 두개의 값을 곱함으로써 링크의 질을 평가하겠다. 사실, 좋은 링크의 선택은 지연시간(delay)당 링크비용(cost)이 가장 낮은 링크의 선택일 것이다. 다시 말해, 지연시간당 1/링크비용이 가장 높은 링크의 선택이다. 그렇다면, $\frac{1}{Cfct_{(i,j)}(c_{(i,j)}) \times Dfct_{(i,j)}(d_{(i,j)})}$

이므로, 이 값이 낮으면 성능이 저하되므로, 분모의 값이 높은 것을 제외시키도록 할 것이다.

3.2 제안하는 알고리즘의 설명

시작 노드(s)와 목적지 노드(d)까지의 경로로서 최초에 P_{LD} 와 P_{LC} 두개의 경로를 가상으로 구축하고 지연시간 제한 조건으로 $supD$ 를 받았다면, 가상으로 구축한 두개의 경로를 사용해서 두개의 경로 중에 우선 선택되는 경로는 다음과 같다. 우선 $path(x)$ 함수를 다음과 같이 정의하여, $path(x) = \begin{cases} LD & \text{if } x \geq supD \\ LC & \text{otherwise} \end{cases}$, 처음에 선택되는 경로는 $P_{path(\frac{\phi_D(P_{LD}) + \phi_D(P_{LC})}{2})}$ 으로 하겠다.

첫 번째, 경로 P_{LD} 가 선택되었다고 가정하자. 우선, 경로 선택을 위한 함수들을 새로이 정의하겠다.

$$\xi(f_k^{-1}(r)) = Cfct_{f_k^{-1}(r)}(c_{f_k^{-1}(r)}) \times Dfct_{f_k^{-1}(r)}(d_{f_k^{-1}(r)}), \quad \forall r \in N_{n(P_D)}$$

$$\phi(P_k(u, v)) = \sum_{r=1}^{n(P_k)} \xi(f_k^{-1}(r)), \quad \forall P_k \in 2^{u \cup v}$$

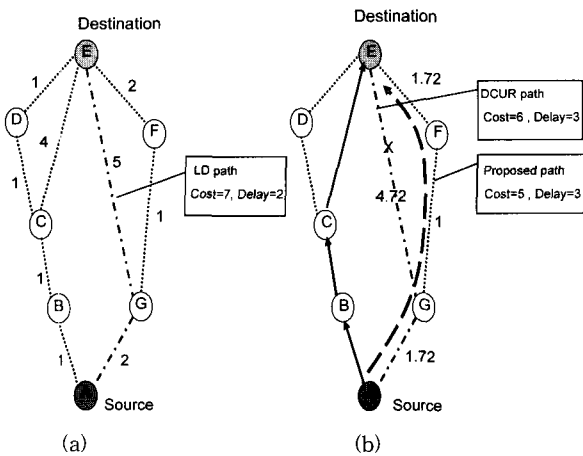
경로 집합 P_{LD} 의 원소를 가지고 $\xi(f_k^{-1}(r))$ 의 값을 모두 계산해 본다. 링크 (u, v) 가 $\max\{x \mid x = \xi((i, j)), \forall (i, j) \in P_{LD}\}$ 를 만족하는 링크이면 우리는 링크 (u, v) 를 대체할 새로운 경로를 찾아야 하는 것이다. 새롭게 찾을 경로의 지연시간 제한 조건으로는 $supD' = supD - \phi_D(P_{LD}) + d_{(u,v)}$ 이 될 것이다. $\min\{\phi(P_k(u, v)), \forall k \in A\}$ 를 만족하는 경로 선택의 후보들을 $P_{candidate}(u, v)$ 라 하자. 경로 $P_{candidate}$ 가 $\phi_D(P_{candidate}) < supD'$ 과 $\phi_C(P_{candidate}) < c_{(u,v)}$ 을 만족한다면 링크 (u, v) 대신 경로 $P_{candidate}(u, v)$ 으로 대체한다. 따라서 새로운 경로를 얻는다. $P'_{LD}(s, d) = P_{LD}(s, d) \setminus \{(u, v)\} \cup P_{candidate}(u, v)$.

이러한 계산을 링크의 수만큼 반복해서 한다. 경로 $P'_{LD}(s, d)$ 의 모든 원소를 실험하고 더 이상 변화가 없다면 최종적으로 이 경로를 사용한다.

두 번째, 경로 P_{LC} 가 선택되었다고 가정하자. 단지 $\phi_D(P_{candidate}) < d_{(u,v)}$ 을 만족한다면 경로 $P_{candidate}(u, v)$ 으로 대체할 것이다. $P'_{LC}(s, d)$ 을 변화시켜 가고, $\phi_D(P'_{LC}(s, d)) \leq supD$ 이 되는 순간부터 첫 번째에 했던 방법과 같은 방식으로 새로운 경로 $P'_{LC}(s, d)$ 를 만들어 갈 것이다. 만약에 $supD$ 보다 크다면, 경로 P_{LD} 를 선택하여 첫 번째 방법을 다시 사용할 것이다.

3.3 예 제

시작 노드 A에서 목적지 노드 E까지의 경로 P_{LD} 와 P_{LC} 를 계산한다. 그래서 $\phi_D(P_{LD})=2$, $\phi_D(P_{LC})=4$ 이며 선택되는 경로는 $P_{path(\frac{\phi_D(P_{LD})+\phi_D(P_{LC})}{2})} = P_{LD}$ 이다. 함수 f_{LD} 가 결정되며 $Image(\xi(f_{LD}^{-1}(\tau))) = \{1.72, 4.72\}$ 이고 $Image(\xi((G, E))) = 4.72$ 의 값이 가장 커서 링크 (G, E) 를 대체할 다른 경로를 $\min\{\phi(P_k(G, E)) \mid P_k \in 2^{G \cdot E}, \forall k \in A\}$ 의 값을 만족하는 경로 $P_{candidate}(G, E) = \{(G, F), (F, E)\}$ 으로 결정한다. 결국 $P(A, E) = \{(A, G), (G, F), (F, E)\}$ 이다. $\phi_C(P_{DCUR}) = 6$ 임에 반(反)해서 $\phi_C(P) = 5$ 이다.



(그림 3) 제안된 알고리즘의 예 : 시작 노드 A에서 목적지 노드 E까지 지연시간 제한 조건은 3, 지연시간은 모두 1로 가정, (a)에 표현된 값은 링크 비용, (b)에 표현된 값은 ξ

4. 성능 평가

우리는 제안된 알고리즘의 성능을 측정하기 위해서, DCUR 알고리즘과 링크비용(link cost)면에서 비교를 하였다. 랜덤

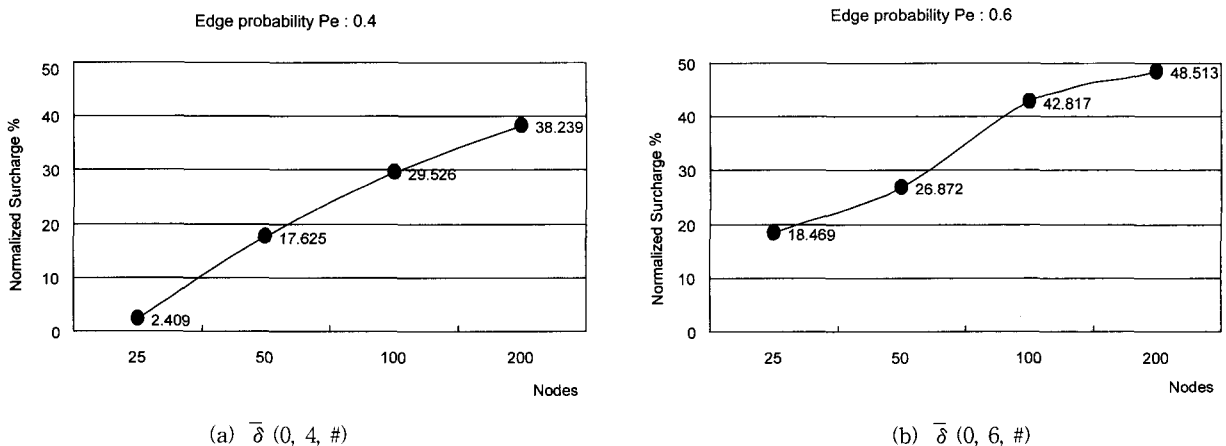
네트워크는 n (네트워크에서의 노드 수)과 pe (노드들 간의 링크가 있을 확률)를 파라미터로 받아 성립한다[7, 8]. 네트워크 모델은 25, 50, 100, 200개의 노드 수로 하였고, 각각의 링크 간 확률은 0.4, 0.6으로 하였다. 각 노드마다 네트워크를 랜덤하게 12개의 모델을 택하였고, 각 모델마다 랜덤으로 시작 노드와 목적지 노드를 선택하여 50번의 시행을 하였기에 총 600번의 시행을 하였다. 지연시간 제한 조건($subD$)은 구간 $[\phi_D(P_{LD}), \phi_D(P_{LC})]$ 에서 랜덤으로 선택하였다. 더욱이 DCUR보다 제안한 알고리즘이 비용적으로 성능이 우월하다는 것을 보이기 위해 함수 $\bar{\delta}$ 를 다음과 같이 정의하였다(Normalized Surcharge[9]). 링크 간 확률(pe)과 각 노드 수($\#$)를 구분하기 위해 함수마다 위치자를 사용하였다.

$$\bar{\delta}(pe, \#) = \frac{\phi_C^{(pe, \#)}(P_{DCUR}) - \phi_C^{(pe, \#)}(P_{proposed})}{\phi_C^{(pe, \#)}(P_{proposed})} \times 100$$

(그림 4)를 보면, 제안된 알고리즘이 DCUR보다 항상 우수하다는 것을 알 수 있다. 노드 수 200에서는 (그림 4)(a), (그림 4)(b) 차례대로 38, 49%의 값을 나타내었다.

5. 결 론

우리는 시작 노드에서 목적지 노드까지 데이터를 전송할 때, 지연시간이 제한되어 있는 문제에 대해 연구했다. 종단간(End-to-End) 지연시간 제한 조건을 만족하면서 최소 비용을 갖는(Delay Constrained Least Cost, DCLC) 경로를 찾는 문제는 이미 NP-hard 문제로 알려져 있다. 본 논문에서는 DCLC문제에 접근하기 위해 링크비용과 지연시간을 확률적으로 조합한 인자를 사용한 새로운 알고리즘을 연구하였다. DCLC 문제의 해법으로 제안한 알고리즘은 최적에 가깝다는 DCUR 알고리즘과 비교하여 노드 수 200에서 38% 이상의 효과를 보였다. 제안한 알고리즘의 특징은 선



(그림 4) 각 링크간 확률(pe)마다 $\bar{\delta}$ 의 값

택의 요소로서 새로운 인자를 만들고, 링크를 순차적으로 선택하지 않고 동적으로 선택하는 방법을 구현하였다. 또한 이 알고리즘을 멀티 캐스팅에 적용시키는 것은 향후 연구 과제이다.

참 고 문 헌

- [1] C. Hedrick, "Routing information protocol," <http://www.ietf.org/rfc/rfc1058.txt>, June, 1988.
- [2] J. Moy, "OSPF Version 2," <http://www.ietf.org/rfc/rfc1583.txt>, March, 1994.
- [3] D. Bertsekas and R. Gallager, 'Data Networks', 2nd Ed., Englewood Cliffs, NJ : Prentice-Hall, 1992.
- [4] M. Garey and D. Johnson, 'Computers and intractability : A Guide to the Theory of NP-Completeness,' New York : Freeman, 1979.
- [5] R. Widyono, "The Design and Evaluation of Routing Algorithms for Real-Time Channels," International Computer Science Institute, Univ. of California at Berkeley, Tech, Rep, ICSI TR-94-024, June, 1994.
- [6] D. S. Reeves and H. F. Salama, "A distributed algorithm for delay-constrained unicast routing," IEEE/ACM Transactions on Networking, Vol.8, pp.239-250, April, 2000.
- [7] A. S. Rodionov and H. Choo, "On Generating Random Network Structures : Trees," Springer-Verlag Lecture Notes in Computer Science, Vol.2658, pp.879-887, June, 2003.
- [8] A. S. Rodionov and H. Choo, "On Generating Random Network Structures : Connected Graphs," Information Networking, Proc. ICOIN-18, pp.1145-1152, February, 2004.
- [9] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast

routing for multimedia communication," IEEE/ACM Trans. Networking, Vol.1, No.3, pp.286-292, June, 1993.



김 문 성

e-mail : moonseong@ece.skku.ac.kr
 2000년 건양대학교 수학과(학사)
 2002년 성균관대학교 수학과(이학석사)
 2002년~현재 성균관대학교 정보통신공학부
 컴퓨터공학전공 재학(박사과정)
 관심분야 : Routing protocols, optical networks, mobile computing, and security.



방 영 철

e-mail : ybang@kpu.ac.kr
 1994년 University of Oklahoma, computer science(학사)
 1997년 University of Oklahoma, computer science(석사)
 2000년 University of Oklahoma, computer science(박사)
 2000년~2002년 한국전자통신연구원(ETRI) 선임연구원
 2002년~현재 한국산업기술대학교 전임강사
 관심분야 : Routing protocols, optical networks, mobile computing, and parallel and distributed computing



추 현 승

e-mail : choo@ece.skku.ac.kr
 1988년 성균관대학교 수학과(학사)
 1990년 University of Texas at Dallas, computer science(석사)
 1996년 University of Texas at Arlington, computer science(박사)
 1997년~1998년 특허청 심사관사무관
 1998년~현재 성균관대학교 정보통신공학부 부교수
 관심분야 : Switching networks, mobile computing, parallel and distributed computing, and storage area networks