

# 지연시간을 고려한 효율적인 분산 유니캐스트 라우팅 알고리즘

## (An Efficient Distributed Delay-Constrained Unicast Routing Algorithm)

신민우<sup>†</sup> 임형석<sup>\*\*</sup>  
(Min-Woo Shin) (Hyeong-Seok Lim)

**요약** 본 논문에서는 연결형 네트워크에서 종단간 지연시간을 만족하고 비용을 최소화하기 위한 휴리스틱 분산 유니캐스트 라우팅 알고리즘을 제시한다. 제시하는 알고리즘은 각 노드마다 네트워크 상태에 대한 지연벡터(delay vector) 및 비용벡터(cost vector)를 유지하고 이 정보를 이용하여 라우팅 경로를 찾는다. 이때 지연시간을 만족하는 경로가 존재하면 언제나 경로를 찾고, 경로를 결정하는데  $O(|E|)$ (단  $|E|$ 는 네트워크를 구성하는 링크의 수)의 메시지 복잡도를 갖는다. 또한 시뮬레이션을 통하여 제안한 알고리즘이 기존의 지연시간 제한 알고리즘들보다 경로의 비용과 지연시간 측면에서 개선된 결과가 있음을 보인다.

**키워드** : 라우팅, 지연시간, 분산 알고리즘, 유니캐스트

**Abstract** We propose a heuristic distributed unicast routing algorithm for minimizing the total cost of the path in a point to point network with delay constraint. The algorithm maintains a delay vector and a cost vector about the network states and finds the path using this information. In this paper, we show that our algorithm always finds a delay-constrained path if such a path exists and has  $O(|E|)$  message complexity ( $|E|$  is the number of links in the network). Also, simulation results show that the proposed algorithm has better cost performance than other delay-constrained routing algorithms.

**Key words** : Routing, Delay Constraints, Distributed Algorithms, unicast

### 1. 서론

최근 들어 멀티미디어 응용과 같은 새로운 실시간 네트워크 응용들이 많이 배출됨에 따라 네트워크 라우팅 알고리즘의 중요성이 부각되고 있다. 이러한 응용들은 보통 엄격한 지연시간 준수를 요구하던가 많은 네트워크 자원을 소비한다. 기존의 전통적인 라우팅 방법들은 보통 홉(hop)수나 최소 지연시간만을 고려하여 종단간 경로를 찾는다. 그러나 이러한 라우팅 방법은 실시간 네트워크 응용에는 적합하지 않다. 따라서 현 네트워크 발전 추세를 고려하면 지연시간을 만족하면서 자원을 효율적으로 관리할 수 있는 알고리즘의 개발은 매우 중요한

의미를 갖는다.

보통 라우팅 문제는 유니캐스트, 멀티캐스트, 브로드캐스트 문제로 분류되는데, 본 논문에서 다루는 유니캐스트 라우팅 프로토콜은 다시 거리벡터(distance-vector) 기반의 프로토콜(routing information protocol(RIP))[1]과 링크상태(link-state)기반의 프로토콜(open shortest path first protocol(OSPF))[2]로 분류된다. 일반적으로 거리벡터 프로토콜들은 Bellman-Ford의 최단경로(shortest path) 알고리즘[3]에 기반을 두며 네트워크 규모의 증가에 따른 확장성이 좋은데, 그 이유는 flooding이나 브로드캐스트와 같은 연산을 포함하지 않고 각 노드는 다른 노드로의 최단경로에 대한 제한된 정보만을 유지하면서 네트워크 위상 변화를 주기적으로 자신과 인접한 노드에게만 보내면 되기 때문이다. 그러나 분산 성향 때문에 네트워크의 상태가 안정되지 않을 경우, 각 노드의 정보가 서로 불일치 할 수 있으며 이로 인

<sup>†</sup> 정 회 원 : 동신대학교 컴퓨터응용학과  
mwshin@blue.dongshinu.ac.kr

<sup>\*\*</sup> 종 신 회 원 : 전남대학교 컴퓨터정보학부 교수  
hslim@chonnam.chonnam.ac.kr

논문접수 : 2001년 10월 4일

심사완료 : 2002년 5월 2일

해 looping 문제가 발생할 수 있다. 반면 링크상태 프로토콜들은 각 노드가 네트워크 위상에 대한 모든 정보를 유지하고 Dijkstra의 알고리즘[4]을 사용하여 주어진 목적노드까지의 최단경로를 구한다. 이러한 중앙 집중적인 성향 때문에 looping 문제는 발생하지 않지만 네트워크 위상 변화에 따른 정보를 flooding을 사용하여 전달하기 때문에 확장성에 문제가 있다. 최근에 Garcia-Luna-Aceves와 Behrens[5]가 링크 벡터(link vector)에 기반하여 looping 문제와 확장성 문제를 해결한 분산형 프로토콜을 제안하였다.

위에서 살펴본 바와 같이 현재 인터넷 라우팅 프로토콜에서 사용중인 라우팅 알고리즘은 주로 최단경로(Shortest Path) 알고리즘에 기반을 두고 있다. 최단경로 알고리즘은 소스노드에서 목적노드까지 각각의 링크들의 길이의 합을 최소화하는 알고리즘으로 링크 길이의 기준을 무엇으로 하느냐에 따라 속성이 달라진다. 만약 링크길이의 기준을 지연시간으로 하면 SP알고리즘은 최소지연시간 경로를 찾게되고, 링크의 비용으로 하면 최소비용 경로를 찾게된다. 이러한 SP의 많은 변형들이 연구되고 있는데 최근에 Wang과 Crowcroft[6]가 라우팅 기준으로 대역폭과 전송지연시간을 고려한 SP알고리즘을 제안하였다. 그러나 이 알고리즘은 주어진 종단간 지연시간을 만족하는 경로를 찾지는 못한다. 현존하는 지연시간 제한 라우팅 알고리즘들은 대부분 멀티캐스트 트래픽을 위해 고안되었고[7,8,9,10,11] 매우 복잡한 구조를 갖는다. 최근에 Salama[12]와 Quan Sun[13]이 지연벡터와 비용벡터라는 네트워크에 대한 제한된 정보를 각 노드에 유지시키는 방법으로 주어진 지연시간 내에 라우팅이 가능한 분산 유니캐스트 라우팅 알고리즘을 제안하였다. Salama의 알고리즘(DCUR)과 Quan Sun 알고리즘(DCR)은 매우 간단하고 효율적인 알고리즘으로서 경로의 비용 측면에서는 서로 비슷하지만 메시징 복잡도 측면에서 최악의 경우 각각  $O(|V|^3)$ 과  $O(|V|)$ 로 알려져 있다(단  $|V|$ 는 네트워크를 구성하는 노드의 수). 그러나 두 알고리즘 모두 알고리즘의 단순성을 위해 임의의 노드에서의 경로 선택을 목적노드까지의 최소비용 경로나 최소 지연시간 경로상의 다음 노드로만 제한하였기 때문에 라우팅 경로의 비용 측면에서 다소 비효율적이라 할 수 있다. 그러므로 본 논문에서는 거리벡터 기반의 라우팅 프로토콜에 쉽게 적용될 수 있고 메시지 복잡도가  $O(|E|)$ 이며, 위에서 언급한 두 알고리즘보다 최소비용 측면에서 우수한 분산 유니캐스트 라우팅 알고리즘을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 네트워크

모델 및 문제 정의, 3장에서는 관련연구, 4장에서는 제안 알고리즘의 기술, 5장에서는 시뮬레이션의 결과, 6장에서는 결론 및 향후 연구과제에 대해서 언급한다.

## 2. 네트워크 모델 및 문제정의

본 논문의 네트워크 모델과 문제정의는 [12]와 [13]에 정의된 것과 유사하다. 각 링크( $E$ )는 전송지연시간, 비용인자를 가지며 네트워크는 이런 링크들로 각 노드( $V$ )들을 연결한 방향 그래프  $G=(V, E)$ 로 표현된다. 각 링크들의 인자를 위해 다음과 같은 두 함수가 사용된다.

▶  $D(e)$  : 해당 링크의 데이터 패킷 전송지연시간

▶  $C(e)$  : 해당 링크의 자원(resource) 활용도

소스 노드( $s$ )와 목적 노드( $d$ )가 주어진 경우  $s$ 에서  $d$ 까지의 경로  $P(s, d)$ 는 다음과 같이 구성되며  $e_1=(s, v_2)$ ,  $e_2=(v_2, v_3), \dots, e_k=(v_k, d)$ ,  $P(s, d)$ 의 비용은 다음과 같이 정의된다.

$$C(P(s, d)) = \sum_{e \in P(s, d)} C(e) \quad (1)$$

마찬가지로  $P(s, d)$ 의 종단간 지연시간은 다음과 같이 정의된다.

$$D(P(s, d)) = \sum_{e \in P(s, d)} D(e) \quad (2)$$

일반적으로 종단간 지연시간 제한 조건을 만족하면서 최소비용을 갖는 경로를 찾는 문제를 DCLC(Delay-Constrained Least-Cost path problem)라하며 다음과 같이 정의된다.

**문제정의(DCLC)** 임의의 방향 네트워크에 링크 지연함수  $D: E \rightarrow R^+$ , 링크 비용함수  $C: E \rightarrow R^+$ , 소스 노드  $s$ , 목적 노드  $d$ , 전송 지연시간 제한조건  $\Delta$ 가 주어진 경우  $C(P(s, d))$ 를 최소화하면서 다음 조건을 만족하는 경로  $P(s, d)$ 를 찾아라.

$$D(P(s, d)) < \Delta \quad (3)$$

DCLC 문제는 NP-Complete임이 증명되었다[3]. Widyono[9]가 이 문제를 해결하기 위해 제안한 Constrained Bellman-Ford(CBF)알고리즘은 최적의 DCLC 경로를 찾지만 각 노드마다 네트워크 위상에 대한 완전한 정보를 유지해야 하고 네트워크 규모가 커짐에 따라 시간 복잡도가 지수승에 비례하여 증가하는 단점을 가지고 있다. 그러므로 본 논문에서는 DCLC 문제를 해결하기 위해 휴리스틱 방법을 적용한 DDUR(Distributed Delay-Constrained Unicast routing algorithm)을 제안한다.

## 3. 관련연구

이 장에서는 본 논문의 동기가 되고 라우팅 알고리즘

의 수행 환경이 DDUR과 비슷한 DCUR[12] 및 DCR[13]에 대해 간단히 살펴본다. 두 알고리즘의 기본 아이디어는 임의의 노드에서 경로 설정 방향을 목적노드까지의 최소비용 경로 또는 최소 지연시간 경로상의 다음 노드로 제한시킨 점이다. 이를 위해 각 노드에 지연벡터와 비용벡터를 유지하는데 이 벡터들은  $|V|-1$ 개의 entry로 구성되며 한 entry는 네트워크를 구성하는 임의의 노드,  $v_j \in V(v_j \neq v_i)$ 에 대한 정보를 포함한다. 노드  $v_i$ 에서 임의의 노드  $v_j$ 를 위한 하나의 entry가 포함하여야 할 지연벡터 및 비용벡터의 정보는 다음과 같다.

■ 지연벡터의 정보

- ①  $v_i$ 에서 임의의 노드  $v_j$ 까지의 최소 지연시간 경로상의 링크 지연시간 합 :  $D(P_{id}(v_i, v_j))$
- ②  $v_i$ 에서 임의의 노드  $v_j$ 까지의 최소 지연시간 경로상의 다음 노드 :  $id(P_{id}(v_i, v_j))$

■ 비용벡터의 정보

- ①  $v_i$ 에서 임의의 노드  $v_j$ 까지의 최소 비용 경로상의 링크 지연시간 합 :  $D(P_{lc}(v_i, v_j))$
- ②  $v_i$ 에서 임의의 노드  $v_j$ 까지의 최소 비용 경로상의 다음 노드 :  $id(P_{lc}(v_i, v_j))$

그림 1의 (a)와 (b)는 위의 벡터정보를 이용하여 전송 지연시간 제한조건이 3인 경우, A노드에서 E노드로의 경로설정 과정을 각각 DCUR 및 DCR을 사용하여 구한 예이다. 설명의 편의를 위하여 현재 경로설정 요청 메시지를 받은 노드를 *active\_node*, 소스 노드에서 *active\_node*까지의 지연시간의 합을 *delay\_so\_far*,  $id(P_{lc}(active\_node, 목적노드))$ 를 *least\_cost\_next\_node*라 하자. 이때 임의의 *active\_node*에서 경로 설정을 위한 다음 노드의 선택은 DCUR의 경우,  $delay\_so\_far + D(active\_node, least\_cost\_next\_node) + D(P_{id}(least\_cost\_next\_node, 목적노드))$ 가 지연시간 제한조건을 만족하면 다음 노드로 *least\_cost\_next\_node*를 선택하고, 그렇지 않으면  $id(P_{id}(active\_node, 목적노드))$ 를 선택한다. 반면 DCR에서는  $delay\_so\_far + D(P_{lc}(active\_node, 목적노드))$ 가 지연시간 제한조건을 만족하면  $id(P_{lc}(active\_node, 목적노드))$ 를 선택하고, 그렇지 않을 경우에는  $id(P_{id}(active\_node, 목적노드))$ 를 선택한다.

그 후 선택된 노드에 (목적노드, 지연시간 제한조건, *delay\_so\_far*)를 포함하는 경로설정 요청 메시지를 보내고 위의 과정을 목적노드가 경로설정 요청 메시지를 받을 때까지 반복한다. 그림 1에서 A노드가 경로설정 요청 메시지를 받을 경우, DCUR에서는 경로상의 다음 노드

로 B노드가 선택되는데 그 이유는  $delay\_so\_far(=0) + D(A, B)(=1) + D(P_{id}(least\_cost\_next\_node(B), E))(=2)$ 가 전송 지연시간 제한조건(=3)을 만족하기 때문이다. 반면 DCR의 경우,  $delay\_so\_far(=0) + D(P_{lc}(A, E))(=4)$ 가 전송 지연시간 제한조건을 만족하지 않기 때문에 목적노드로의 최소지연시간 경로상의 다음 노드인 H가 경로설정의 다음 노드로 선택된다. 선택된 노드를 다시 *active\_node*로 하고 위의 과정을 반복하면 그림 1의 (a), (b)와 같은 경로가 구해진다. 하지만 DCUR은 경로 설정 시 looping이 발생 될 수 있으며 이를 해결하기 위해 최악의 경우 메시지 복잡도가  $O(|V|^3)$ 로 나빠지게 된다. DCR은 이런 looping문제를 해결하고 메시지 복잡도를  $O(|V|)$ 로 향상시켰지만 그림 1에서 보듯이 두 알고리즘 모두 다음 노드의 선택 기준을 최소비용 경로나 최소지연 시간 경로상의 다음 노드로 한정하고 있기 때문에 만약 A노드에서 G노드를 선택할 경우 주어진 전송 지연시간 제한조건을 만족하면서 비용 측면에서 더 효율적인 경로(A→G→F→E)를 구할 수 있음에도 불구하고 다음 노드로 G노드를 선택할 수 없음을 알 수 있다.

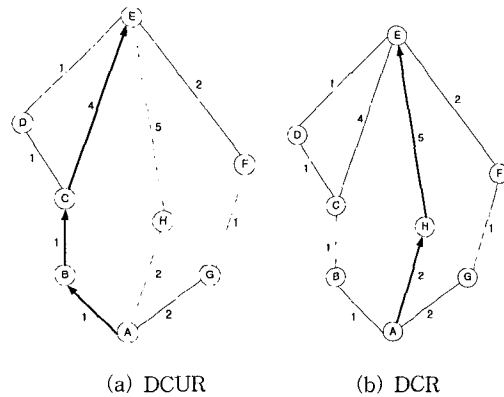


그림 1 전송지연시간 제한조건이 3인 경우 소스 노드 A에서 목적 노드 E까지의 경로설정 예(각 링크에 표시된 값은 cost이고 delay는 모두 1이라고 가정)

4. DDUR 알고리즘

이 장에서는 알고리즘을 제시하기 전에 필요한 용어를 정의하고 경로설정을 위한 라우팅 정보 및 제안 알고리즘에 대해 서술한다.

4.1 용어정의

DDUR에서 사용되는 용어는 다음과 같다.

- $s$  : 소스 노드
- $d$  : 목적 노드
- $\Delta$  : 전송 지연시간 제한 조건
- $P_{lc}(v_i, v_j)$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 비용 경로
- $P_{ld}(v_i, v_j)$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 지연시간 경로
- $D(P_{lc}(v_i, v_j))$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 비용 경로상의 링크 지연시간 합
- $D(P_{ld}(v_i, v_j))$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 지연시간 경로상의 링크 지연시간 합
- $C(P_{lc}(v_i, v_j))$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 비용 경로상의 링크 비용 합
- $C(P_{ld}(v_i, v_j))$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 지연시간 경로상의 링크 지연시간 합
- $id(P_{ld}(v_i, v_j))$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 지연시간 경로상의 다음 노드
- $id(P_{lc}(v_i, v_j))$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 최소 비용 경로상의 다음 노드
- $D(v_i, v_j)$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 링크 지연시간
- $C(v_i, v_j)$  : 노드  $v_i$ 에서 노드  $v_j$ 까지 링크 비용

4.2 라우팅 정보

DDUR의 올바른 경로 설정을 위해 각 노드  $v_i \in V$ 는 진출 링크에 대한 지연시간 및 비용 그리고 비용벡터와 지연벡터에 대한 정보를 유지한다. 지연벡터 및 비용벡터는 3장에서 언급한 DCUR 및 DCR의 벡터 정보에 각각  $C(P_{ld}(v_i, v_j))$ 와  $C(P_{lc}(v_i, v_j))$ 를 추가한 내용으로 구성된다. 이 벡터 정보들은 현재 거리벡터 기반의 라우팅 프로토콜[1]에서 사용중인 거리벡터(distance vectors)를 수집하는 절차를 이용하면 쉽게 획득할 수 있다. 그러나 본 논문은 이런 벡터들을 입력으로 하는 라우팅 알고리즘을 다룬 것임으로 지연벡터나 비용벡터 유지에 대한 자세한 절차는 언급하지 않기로 한다.

4.3 제안 알고리즘

DCUR 및 DCR 그리고 본 논문에서 제안한 DDUR은 임의의 노드에서 목적노드까지의 최소 비용경로가 주어진 전송 지연시간 제한조건을 만족할 경우, 이 후의 경로설정은 목적노드까지의 최소 비용경로가 됨으로써 모두 동일한 경로를 구하게 된다. 하지만 그렇지 않을 경우, 3장에서 언급했듯이 DCUR과 DCR은 경로설정의 선택을 두 가지 경우로 제한함으로써 알고리즘의 간결성을 꾀한 반면 최소 비용경로 측면에서는 다소 비효율적임을 알 수 있었다. 따라서 DDUR에서는 다음 노드의 선택 기준을 보다 확장함으로써 전체적인 경로의 비용을 감소시킨다. DDUR은 소스 노드에서 목적 노드까

지의 경로 설정이 매번 하나의 노드씩 결정되며, 경로설정 요청 메시지를 받은 임의의 *active\_node*에서 목적 노드로의 다음 노드 선택 기준은 다음과 같다.

■  $D(P_{lc}(active\_node, d)) \leq \Delta$ 이면 경로 설정의 다음 노드는 DCR 알고리즘과 같이  $id(P_{lc}(active\_node, d))$ 가 된다.

■  $D(P_{lc}(active\_node, d)) > \Delta$ 인 경우는 *active\_node*와 이웃한 노드들을  $v_i, (i=1, \dots, k)$ 라 할 때, 각  $v_i$ 에 비용 계산 요청 메시지를 보내고 비용계산 요청 메시지를 받은  $v_i$ 는 아래와 같은 방법으로 비용을 계산한 후 계산된 비용을 포함하는 응답 메시지를 *active\_node*에 되돌려 보낸다. *active\_node*는 응답 메시지 중 목적노드까지 최소 비용을 유도할 수 있는  $v_i$ 를 선택하여 경로 설정의 다음 노드로 선택한다.

( $v_i$ 의 비용 계산) 만약  $delay\_so\_far + D(active\_node, v_i) + D(P_{ld}(v_i, d)) > \Delta$ 이면 목적 노드까지의 경로를 구할 수 없으므로 "path not found"라는 응답 메시지를 *active\_node*에 보낸다. 그렇지 않을 경우,  $delay\_so\_far + D(active\_node, v_i) + D(P_{lc}(v_i, d)) \leq \Delta$ 를 만족하면  $v_i$ 의 비용을  $C(active\_node, v_i) + C(P_{lc}(v_i, d))$ 로 설정하고, 만족하지 않으면  $C(active\_node, v_i) + C(P_{ld}(v_i, d))$ 로 설정한다.

그 후 선택된 노드에 ( $d, \Delta, delay\_so\_far, path\_direction$ )을 포함하는 경로설정 요청 메시지를 보내고 위의 과정을 목적노드가 경로설정 요청 메시지를 받을 때까지 반복한다. 단 *path\_direction*은 경로설정의 다음 노드가  $id(P_{lc}(active\_node, d))$ 이면 LC값을, 그렇지 않은 경우는 LD값을 갖는다.

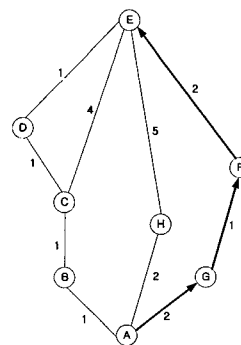


그림 2 DDUR에서  $\Delta$ 가 3인 경우 소스노드 A에서 목적 노드 E까지의 경로설정 예(각 링크에 표시된 값은 cost이고 delay는 모두 1이라고 가정)

그림 2는  $\Delta$ 가 3이고 소스 노드가 A 목적 노드가 E인 경우, DDUR에 의해 수행되는 경로 설정 과정을 보여준다. 그림에서  $P_k(A, E) = ((A, B), (B, C), (C, D), (D, E))$ 이고  $D(P_k(A, E)) = 5$ 이며  $P_{id}(A, E) = ((A, G), (G, E))$ 이고  $D(P_{id}(A, E)) = 2$ 임을 알 수 있다. 먼저  $D(P_{id}(A, E)) < 3$ 은 만족하지만  $D(P_k(A, E)) < 3$ 이 성립하지 않기 때문에 A는 이웃 노드 B와 H,G에게 비용 계산 요청 메시지를 보낸다. 요청 메시지를 받은 B와 H,G는 각각 위에서 언급한 비용 계산 방법에 따라 비용을 계산 한 후 비용을 포함하는 응답 메시지를 생성하여 A에게 되돌려 준다. A는 G로부터 받은 비용의 값이 가장 최소이므로 경로 설정의 다음 노드로 G를 선택하고 그 후 경로 설정 요청 메시지를 생성하여 G로 보낸다. G를 다시 *active\_node*로 하여 위의 과정을 목적노드인 E가 경로 설정 요청 메시지를 받을 때까지 반복하면 경로 설정이 성공적으로 끝나게 되고 그림 2와 같은 경로가 구해진다.

**정리 1 :** DDUR에 의해 생성되는 경로는 loop를 포함하지 않는다.

**증명 :** 주어진  $s$ 와  $d$ , 그리고  $\Delta$ 에 대해, 만약  $D(P_{id}(s, d)) < \Delta$ 를 만족하면  $s$ 에서  $d$ 까지의 경로는 최소비용 경로가 되고 최소비용 경로는 loop-free하다는 것은 자명한 사실이다.  $D(P_{id}(s, d)) > \Delta$ 인 경우는 주어진 전송 지연시간 제한조건을 만족하는 경로가 없음을 의미하며 그렇지 않다면  $s$ 는 인접한 이웃노드들에게 비용 계산 요청 메시지를 보냄으로써 경로 설정 과정을 시작한다. 비용 계산 요청 메시지를 받은 노드는 비용을 계산하여 요청 메시지를 보내온 노드로 되돌려 준다. 다음 노드로의 경로 설정은 반환된 비용 중 최소 값을 보내온 노드가 되므로 loop이 형성되기 위해서는 최소한 하나 이상의 요청 메시지가 생성되어야 한다.

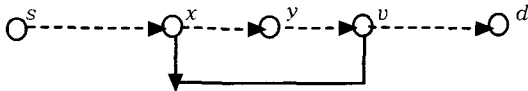


그림 3 v에서 비용 계산 요청 메시지가 생성되어 x가 선택된 경우

그림 3은 이런 경우를 보여준다. 그림 3에서  $P_e$ 를 노드  $s, x, y, v$ 를 포함하는 경로라 하자. 현재  $v$ 까지 경로가 설정되었고 다음 노드 선택을 위해  $v$ 에서 요청 메시지가 생성되었다고 하자. 이때 loop이 형성되기 위해서는 다음 노드의 선택이  $P_e$ 상의 임의의 노드라야 한다. 그림 3에서 보듯이  $x$ 를 그런 노드라 하면  $x$ 에서  $y$ 로

의 *path\_direction*은 LC가 될 수 없고 다음의 식을 만족해야만 한다.

$$D(s, x) + D(P_k(x, d)) > \Delta \quad (4)$$

$x$ 는  $v$ 의 요청 메시지에 의해 선택된 노드이므로  $v$ 에서  $x$ 로의 *path\_direction*은 LC나 LD가 될 수 있다. 만약 *path\_direction*이 LC라고 하면 다음의 식이 성립한다.

$$D(P_e) + D(v, x) + D(P_k(x, d)) < \Delta \quad (5)$$

그러나  $D(s, x) < D(P_e)$ 이기 때문에 (4)과 (5)는 서로 모순이다. 그러므로 다음 노드로  $P_e$ 상의 임의의 노드는 선택되어질 수 없다. 만약 *path\_direction*이 LD라면 다음의 식을 만족해야 한다.

$$C(P_{id}(x, d)) > C(x, y) + C(P_{id}(y, d)) \quad (6)$$

$$C(P_{id}(y, d)) > C(y, v) + C(P_{id}(v, d)) \quad (7)$$

$$C(P_{id}(v, d)) > C(v, x) + C(P_{id}(x, d)) \quad (8)$$

위의 식으로부터  $C(P_{id}(v, d)) > C(v, x) + C(x, y) + C(y, d) + C(P_{id}(v, d))$ 를 유도할 수 있는데 이는 명백한 모순이므로  $P_e$ 상의 임의의 노드가 다음노드로 선택되어질 수 없다.

**정리 2 :** DDUR은  $s$ 에서  $d$ 까지 주어진 전송 지연시간 제한조건을 만족하는 경로가 존재할 경우 언제나 지연시간 제한조건을 만족하는 경로를 찾는다.

**증명 :** DDUR은  $s$ 에서  $d$ 까지 주어진 전송 지연시간 제한 조건을 만족하는 경로가 최소한 하나 이상 존재해야만 경로 설정과정을 시작한다. 경로설정 과정 중 임의의 노드  $v_i$ 에서의 다음 노드가  $id(P_k(v_i, d))$ 이면 이는  $v_i$ 에서  $d$ 까지 최소비용 경로가 주어진 전송 지연시간 제한 조건을 만족한 경우이므로 최소비용 경로가 찾고자 하는 경로가 된다. 그렇지 않은 경우,  $v_i$ 의 다음 노드는  $v_i$ 의 이웃 노드 중에서 선택된다. 이때 선택된 노드를  $w$ 라 하면  $w$ 에서 다음 조건은 언제나 성립된다.

$$delay\_so\_far + D(P_{id}(w, d)) < \Delta$$

이는 선택된 노드  $w$ 에서 전송 지연시간 제한 조건을 만족하는 경로가 최소한 하나 이상 존재하는 것을 의미하며 위의 과정은 다음 노드로  $d$ 가 선택될 때까지 반복되므로 언제나 전송 지연시간 제한조건을 만족하는 경로를 찾게 된다.

**정리 3 :** DDUR의 최악의 경우 메시지 복잡도는  $O(|E|)$ 이다.

**증명 :** DDUR은 loop-free하고 임의의  $s$ 에서  $d$ 까지 매번 하나의 노드씩 경로가 결정된다. 경로설정 중 비용 계산 요청 메시지가 발생하지 않을 경우는 다음 노드의

선택이  $O(1)$ 만에 가능하지만 최악의 경우 경로설정 과정의 모든 노드에서 비용 계산 요청 메시지가 발생될 수 있다. 이 때 기껏해야 자신과 이웃한 노드들로만 요청메시지가 전달되므로 DDUR의 최악의 경우 메시지 복잡도는  $O(E)$ 이 된다.

5. 시뮬레이션

본 논문에서는 네트워크 생성을 위해 [14]에서 제시된 랜덤 그래프 생성기를 사용하였다. 이 생성기는 처음에 좌표상의 임의의 위치에 노드들을 생성시키고 아래와 같은 확률로 노드들간의 링크를 연결한다.

$$P(u, v) = \beta \exp\left(-\frac{l(u, v)}{La}\right) \quad \alpha, \beta \in (0, 1)$$

여기서  $-l(u, v)$ 는 노드  $u$ 에서 노드  $v$ 사이의 거리이고,  $L$ 은 임의의 두 노드사이의 최대 거리이다.  $\alpha$ 와  $\beta$ 는  $(0, 1)$ 사이의 값으로  $\alpha$ 가 클수록 먼 거리에 있는 노드간의 링크 존재 비율이 증가하고  $\beta$ 가 클수록 그래프의 평균 차수가 증가한다. 라우팅 시뮬레이션 환경은 Salama 알고리즘(DCUR)과 Quan Sun(DCR) 알고리즘, 그리고 최적 알고리즘으로 알려진 CBF와의 성능 비교를 위해 [15]에서 사용한 환경을 수정하여 사용하였다. 네트워크 환경은 링크들의 수용능력이 동일하게 155 Mbps 인 전 이중성 ATM망을 가정하였고 각 노드들은 non-blocking ATM 스위치를 나타낸다. 또한 각 링크는 5 Mbps에서 125 Mbps 사이의 background traffic을 갖도록 하였고 각 링크의 비용  $C(e)$ 는 reserved bandwidth를, 지연시간  $D(e)$ 는 propagation delay를 사용하였다. 본 논문에서 제안한 DDUR과 위에서 언급한 DCUR, DCR, CBF와 소스 노드에서 목적 노드까지의 평균 링크 비용, 평균 링크 지연시간 등을 비교하기 위해서 95% 신뢰수준을 사용하여 실험을 반복하였다.

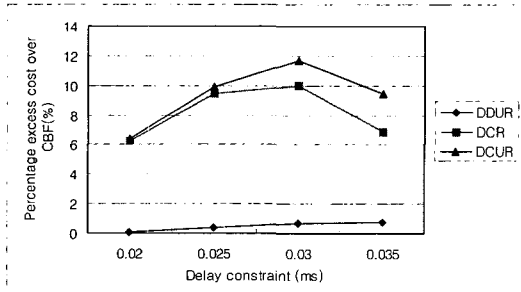


그림 4 노드 수가 20이고 평균 노드 차수가 5일 때 CBF에 대한 평균 초과 비용 비율

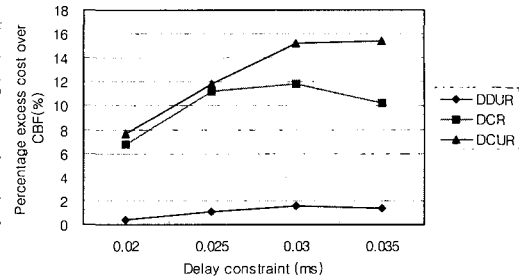


그림 5 노드 수가 50이고 평균 노드 차수가 5일 때 CBF에 대한 평균 초과 비용 비율

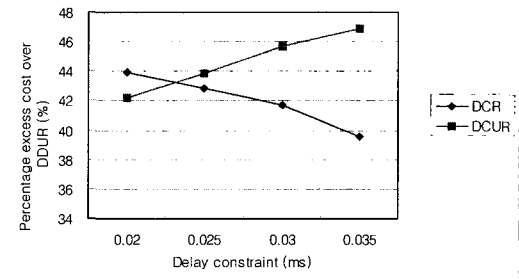


그림 6 서로 다른 경로를 찾을 경우 DDUR에 대한 평균 초과 비용 비율(노드 수가 20이고 평균 노드 차수가 5인 경우)

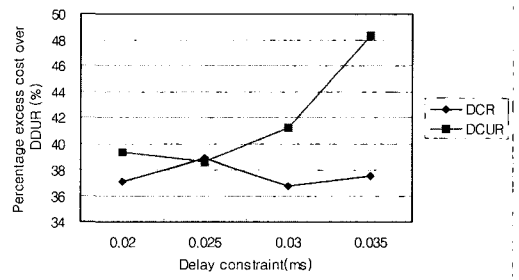


그림 7 서로 다른 경로를 찾을 경우 DDUR에 대한 평균 초과 비용 비율(노드 수가 50이고 평균 노드 차수가 5인 경우)

그림 4와 5는 네트워크를 구성하는 노드의 수가 각각 20, 50일 때 CBF에 대한 상대적인 초과 비용을 나타낸 것이다. 그림에서 보듯이 DCR과 DCUR은 각각 최악의 경우 약 12%, 15%정도의 비용 초과가 발생하지만

DDUR은 2%미만으로 주어진  $\Delta$ 에 대해 거의 최적 비용의 경로를 찾음을 알 수 있다. 그러나 세 알고리즘 모두 주어진 소스 노드에서 목적 노드까지 최소 비용 경로가 주어진 전송 지연시간 제한조건을 만족하면 동일한 경로(최소 비용 경로)를 구하게 된다. 그런데 전송 지연시간 제한 조건이 커짐에 따라 목적 노드까지의 최소비용 경로가 주어진 전송 지연시간 제한조건을 만족하게 될 확률이 높아지고 따라서 세 알고리즘이 동일한 경로를 구하게 되는 경우 수가 증가함을 알 수 있었다. 그러므로 더욱 더 정확한 비교를 위해 DDUR과 DCUR 그리고 DDUR과 DCR이 서로 다른 경로를 찾을 경우, 그 경우만을 추출하여 이들 각각에 대한 평균 비용 분석이 필요하였다. 그림 6과 7은 네트워크를 구성하는 노드의 수가 각각 20, 50일 때 위에서 언급한 각각의 경우에 대해서 DDUR에 대한 평균 초과 비용을 구한 것이다. 그림에서 보면 서로 다른 경로를 찾을 경우, DDUR이 DCUR 및 DCR에 비해 약 40%정도의 비용 개선 효과가 있음을 알 수 있다. 또한 그림에는 나타나 있지 않지만 시뮬레이션의 결과 평균 노드의 차수가 증가할수록 DDUR의 전체 라우팅 경로 비용이 현저히 감소함을 알 수 있었다. 그러나 현 네트워크 라우터들의 평균 차수가 4~5정도[12]임을 감안하여 평균 차수 5이상에 대해서는 언급하지 않았다.

## 6. 결론

본 논문에서는 지연시간을 고려한 효율적인 분산 유니캐스트 라우팅 알고리즘(DDUR)을 제안하였다. DDUR은 라우팅을 위해 각 노드에 비용벡터와 지연벡터라는 네트워크의 제한된 정보를 유지하며, 이 정보는 현 라우팅 프로토콜에서 사용중인 거리벡터와 유사하다. DDUR은 간단하면서도 네트워크 규모에 따른 확장성이 보장되는 알고리즘으로서 loop-free하고 주어진 전송 지연시간 제한조건을 만족하는 경로가 존재할 경우 언제나 조건에 맞는 경로를 찾음을 알 수 있었다. 또한 라우팅 경로를 결정하는데 최악의 경우  $O(|E|)$ 의 메시지 교환이 발생함을 보였고, 최적 알고리즘인 CBF, 라우팅 환경이 비슷한 DCR, DCUR과의 시뮬레이션을 통한 비교 분석 결과 DDUR의 성능이 경로의 비용 측면에서 매우 만족한 결과를 보임을 알 수 있었다.

앞으로의 연구 방향은 DDUR을 기본으로 하여 지연시간을 고려한 브로드캐스트 라우팅 알고리즘이나 멀티캐스트 라우팅 알고리즘을 개발하는 것이라 할 수 있다.

## 참고 문헌

- [1] Hedrick, C, Routing Information Protocol, Internet RFC 1058, June 1988.
- [2] J. Moy, "OSPF Version 2," Internet RFC 1583, <http://ds.internic.net/rfc/rfc1583.txt>, March 1994.
- [3] R. Bellman, Dynamic Programming, Princeton University Press, 1957.
- [4] E. Dijkstra, "Two Problem in Connection with Graphs," *Numerische Mathematik*, vol. 1, no. 5, pp. 269-271, October 1959.
- [5] J. Garcia-Luna-Aceves and J. Behrens, "Distributed, Scalable Routing Based on Vectors of Link States," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1383-1395, October 1995.
- [6] Wang, Z. and Crowcroft, J., "Quality-of-Service Routing for Supporting Multimedia Applications," *IEEE Journal on Selected Areas in Communications*, 14(7), 1996.
- [7] Kompella, V.P., et al., "Multicast Routing for Multimedia Communication," *IEEE/ACM Transactions on Networking*, 1(3), pp.286-292, 1993.
- [8] Kompella, V.P., et al., "Two Distributed Algorithms for the Constrained Steiner Tree Problem," in *Proceedings of the Second International Conference on Computer Communications and Networking*, pp.343-349, 1993.
- [9] Widyono, R., "The Design and Evaluation of Routing Algorithms for Real-Time Channels," Technical Report TR-94-024, Tenet Group, Department of EECS, University of California at Berkeley, 1994.
- [10] Zhu, Q., et al, "A Source-Based Algorithm for Near-Optimum Delay-Constrained Multicasting," *IEEE INFOCOM'95*, pp.377-385, 1995.
- [11] Wi, S. and Choi, Y., "A Delay Constrained Distributed Multicast Routing Algorithm," *12th International Conference on Computer Communication, ICC'95*, pp.833-838, 1995.
- [12] Salama, H.F., Reeves, D.S. and Viniotis, Y., "A Distributed Algorithm for Delay-Constrained Unicast Routing," *IEEE INFOCOM'97*, pp.84-91, 1997.
- [13] Q. Sun and H. Langendoerfer, "A New Distributed Routing Algorithm for Supporting Delay-Sensitive Applications," *Computer Communications*, vol. 21, no. 6, pp. 572-578, May 1998.
- [14] Waxman, B.M., "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, 6(9), pp.1617-1622, 1988.
- [15] Salama, H.F., et al, "Evaluation of Multicast

Routing Algorithms for Real-Time Communication on High-Speed Networks," in High Performance Networking VI, IFIP 6th International Conference on High Performance Networking, pp.27-42, 1995.



신 민 우

1990년 전남대학교 전산학과 학사. 1992년 전남대학교 전산학과 석사. 1996년 12월 ~ 1998년 8월 (주)멀티정보 연구 개발 팀장. 2000년 2월 ~ 현재 동신대학교 컴퓨터응용학과 전임강사. 관심분야는 그래프이론, 라우팅 프로토콜 및 알고리즘

리즘

임 형 석

정보과학회논문지 : 정보통신  
제 29 권 제 3 호 참조