로드 밸런싱 기능을 갖는 루프 프리 지연 제약 라우팅 알고리즘

최 영 수† · 정 진 욱††

요 약

디지털 오디오나 비디오와 같은 멀티미디어 트래픽은 QoS 제약 조건(종단간 지연, 대역폭 가용성, 패킷 분실률, 지터 등)을 요구하고 있다. 이러한 멀티미디어 트래픽의 제약 조건을 보장 하기 위해서는 데이터 전송을 하기 전에 실시간 채널을 설정할 필요가 있으며 그러한 채널의 설정은 QoS 제약 조건을 고려한 효율적인 경로 설정 알고리즘을 필요로 한다. 일반적으로 네트워크에서 지연 제약 조건과 최소 비용을 만족하는 경로를 찾는 것은 NP-complete임이 증명되었다. 따라서 본 논문에서는 간단하고 분산된 루프 프리 라우팅 알고리즘(Loop-free Delay-constrained Routing algorithm with Load balancing : DRL)을 제안한다. 제안한 DRL은 지연 제약 조건을 만족하는 경로를 선택하며 네트워크의 상태에 따라 부하를 분산시키는 기능을 갖고 있다. 또한 지연 제약 라우팅을 위해 각 노드에서는 한정된 네트워크 상태 정보만을 필요로 한다. 시뮬레이션 결과는 DRL이 LDP와 마찬가지로 지연 제약 조건을 만족하는 경로를 제공하면서 DCUR에 비해 루프가 발생하지 않는 경로 설정을 수행한다는 것을 보여주고 있다.

Loop-free Delay-constrained Routing algorithm with Load balancing

Young-Su Choi† · Jin-Wook Chung††

ABSTRACT

Multimedia traffic involving digital audio and video requires QoS constraints (end-to-end delay, bandwidth availability, packet loss rate, jitter, etc.). To guarantee multimedia traffic satisfying these constraints needs to be established real time channel before transmission. The establishment of such channels requires efficient route selection algorithms that are designed to take into account the QoS constraints. The general problem of determining a least-cost delay-constraint route has been proved NP-complete. Therefore, we propose simple and distributed loop-free routing algorithm (Loop-free Delay-constrained Routing algorithm with Load balancing : DRL). The DRL we proposed can select route satisfying delay constraint and provide network load balancing according to network situation. In order to do delay-constrained routing, DRL requires limited network state information to be kept at each node. The simulation results show that DRL selects the loop-free route satisfying delay constraint as well as LDP and executes efficient network load balancing.

키워드 : 지연 제약(delay constraints), 라우팅 알고리즘(routing algorithm), 서비스질(quality of service), 로드밸런싱(load balancing), 멀티미디어 통신(multimedia communication)

1. 서 론

현대의 네트워크는 전통적인 데이터 통신 응용 서비스를 제공 하는 것 외에 비디오나 음성과 같은 멀티미디어 스트림 서비스의 제공이 필수적인 요소가 되었다. 이러한 멀티미디어 스트림 서비스는 반드시 QoS 요구 사항을 필요로 한다. 예를 들면 종단간 지연, 지터, 가용 대역폭, 패킷 분실률 등이 있다. 그러한 여러 QoS 요구 사항에 맞추어 응용 서비스의 성능을 보장하기 위해서는 네트워크에 경로 설정을 위한 기법이 필요하게 된다.

† 준 회 원 : 성균관대학교 대학원 전기전자및컴퓨터공학부
†† 종신회원 : 성균관대학교 전기전자및컴퓨터공학부 교수
논문접수 : 2001년 3월 16일, 심사완료 : 2001년 7월 9일

경로 설정을 위해서는 라우팅 프로토콜이 사용되며 유니캐스트 라우팅 프로토콜로는 디스턴스 벡터 프로토콜(distance-vector protocol)과 링크 상태 프로토콜(link-state protocol)의 두 가지 방식으로 분류할 수 있다. 디스턴스 벡터 프로토콜의 대표적인 프로토콜은 Routing Information Protocol(RIP)이 있으며 Bellman-Ford의 최단 거리 알고리즘(shortest path algorithm)을 사용한다[1]. RIP는 각 노드에 제한된 정보만을 유지하며 경로 설정을 수행한다. 반면 링크 상태 프로토콜의 대표적인 프로토콜은 Open Shortest Path First(OSPF)가 있으며 각 노드는 네트워크 토폴로지에 대한 완전한 정보를 유지하고 있다[2].

제약 기반 라우팅(constraint based routing)은 여러 제한

사항을 따르는 경로를 계산하기 위해 사용되며, QoS 라우팅으로부터 그 개념이 확장되었다. 네트워크에 임의의 플로우가 QoS 요구 사항을 갖고 있을 때 QoS 라우팅은 그러한 QoS 요구 사항을 만족할 수 있는 경로를 찾는다. 반면 제약 기반 라우팅은 정책(policy)과 같은 또 다른 네트워크의 제한 사항을 고려함으로써 QoS 라우팅의 개념을 확장한다. 제약 기반 라우팅의 목적은 QoS 요구 사항을 만족할 수 있는 경로 선택과 네트워크 이용률의 증가로 볼 수 있다. 즉, 네트워크 사용자 관점에서는 만족할 만한 서비스를 제공해주는 한편 네트워크 제공자 관점에서는 제한된 네트워크 자원의 최대 활용이라는 것을 나타낸다.

경로를 결정할 때 제약 기반 라우팅은 네트워크의 토폴로지뿐만 아니라 플로우의 요구 사항, 링크의 가용율, 네트워크 관리자에 의한 정책 등을 고려해야 한다. 그러므로 제약 기반 라우팅은 부하가 많이 걸린 짧은 경로보다 부하가 적은 긴 경로를 찾는다고 볼 수 있다. 따라서 네트워크의 트래픽은 QoS 라우팅보다 훨씬 분산된다.

제약 기반 라우팅을 하기 위해서는 경로 계산을 위해 라우터가 훨씬 더 많은 링크의 상태 정보를 갖고 있어야 한다. 즉, 네트워크 토폴로지 정보와 가용 자원 정보 등이 QoS 경로 계산에 필요하게 된다. 가용 자원 정보로는 특히 링크의 가용 대역폭이 있으며 버퍼량은 충분하다고 가정하여 크게 고려하지 않는다[3]. 이러한 링크 정보를 최신 정보로 유지하기 위해 각 노드는 수시로 링크 상태 정보를 주고 받으며 이 때 정확한 정보 유지와 라우팅 트래픽의 오버헤드 사이에 적당한 타협이 필요하게 된다.

제약 기반 라우팅에서 경로 계산의 복잡성은 경로 설정에 사용한 메트릭에 따라 달라진다. 비용, 홉 수, 대역폭, 신뢰성, 지연, 지터 등 사용할 수 있는 여러 메트릭이 있으며 이러한 메트릭은 다시 세가지 범주로 나누어 볼 수 있다. d (i, j)가 link(i, j)의 메트릭이라고 할 때 임의의 어떤 경로 P = (i, j, k, ⋯, m)에 대한 메트릭 d는 다음과 같이 정의할 수 있다.

$$\text{가산형 } d(P) = d(i,j) + d(j,k) + \cdots + d(l,m)$$
$$\text{승산형 } d(P) = d(i,j) * d(j,k) * \cdots * d(l,m)$$
$$\text{오목형 } d(P) = \min\{d(i,j), d(j,k), \cdots, d(l,m)\}$$

이 정의에 따라 지연, 지터, 비용, 홉 수 등은 가산형(additive) 메트릭, 신뢰성(1-분실률)은 승산형(multiplicative) 메트릭 그리고 대역폭은 오목형(concave) 메트릭으로 볼 수 있다. 제약 기반 라우팅과 관련하여 둘 이상의 가산형이나 승산형 메트릭의 제약 사항을 만족하는 최적화 경로(optimal path)를 찾는 것은 NP-complete 문제임이 증명되었다[4]. 이 증명에서는 모든 메트릭은 독립적이며 링크의 지연과 지터는 알려져 있다는 가정을 하고 있다. 따라서 대역폭 메트릭과 다른 가산형 혹은 승산형 메트릭 중 하나를 만족하는 경로를 찾는 것이 가능할 것이다.

본 논문에서는 링크의 가용 대역폭과 지연을 메트릭으로 사용하여 지연 제약 조건을 만족하는 경로를 설정하면서 네트워크의 로드 밸런싱 기능을 제공하는 유니캐스트 라우팅 알고리즘인 DRL을 제안한다. DRL은 현재 인터넷에서 사용되고 있는 RIP와 같이 제한된 정보만을 유지하는 디스턴스 벡터 방식의 간단하고 분산된 알고리즘이며, 루프가 발생하지 않는 특성을 갖고 있다.

2. 제약 기반 라우팅

제약 조건을 만족시키는 경로를 선택하는 것은 제약 기반 라우팅에서 무엇보다 중요하다. 그러나 이와 함께 고려할 사항이 로드 밸런싱이다. 네트워크에서 로드 밸런싱이란 응용 서비스의 트래픽을 네트워크 전반에 고르게 분산시키는 능력을 의미한다. 즉, 동일한 네트워크 자원으로 더 많은 응용 서비스의 트래픽 플로우를 처리할 수 있으면 로드 밸런싱 효율이 더 좋다고 할 수 있다. 제약 조건을 만족하는 경로 설정을 위해 다음과 같은 방법 중에서 한가지를 선택할 수 있다.

Widest-shortest 경로 : 최소 홉 수 선택. 여러 경로 존재 시 가용 대역폭이 가장 큰 경로 선택
Shortest-widest 경로 : 가용 대역폭이 가장 큰 경로 선택. 여러 경로 존재 시 최소 홉 수 경로 선택
Shortest-distance 경로 : k 홉인 경로 P의 거리(Distance)는 다음과 같이 정의한다.

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

Widest-shortest 경로 방식은 네트워크의 자원을 가장 적게 쓰는 방식으로, 현재의 동적 라우팅(dynamic routing) 방식과 기본적으로 같다. 그러나 네트워크의 부하가 많을 경우 효율적이지 못한 단점이 있다. Shortest-widest 경로 방식은 로드 밸런싱을 강조하는 방식이다. 그러나 네트워크의 자원을 많이 사용하는 단점이 있다. 즉, 어떤 플로우의 요구 대역폭이 1Mbps일 때 2홉으로 전달하는 경우 네트워크는 2Mbps의 자원을 소비하지만, 3홉으로 전달하는 경우 3 Mbps의 자원을 소비하기 때문이다. Shortest-distance 경로 방식은 이 둘의 장점을 취한 방식이다. 네트워크의 부하가 별로 없는 경우 대역폭이 가장 큰 경로를, 부하가 높아지면 최단 거리 경로를 선택하는 방식이다[5].

본 논문에서는 링크의 가용 대역폭과 지연을 메트릭으로 사용하여 지연 제약 조건을 만족하는 경로를 설정하면서 네트워크의 로드 밸런싱 기능을 제공하는 유니캐스트 라우팅 알고리즘인 DRL을 제안하며 이를 다른 제약 기반 라우팅 알고리즘인 DCUR(Delay-Constrained Unicast Routing)[7] 및 LDP(LD Path routing algorithm)와 성능 비교를

통해 그 우수성을 분석한다.

DCUR은 로드 밸런싱을 위해 링크의 이용률을 메트릭으로 사용하는 비용 벡터 테이블과 링크의 지연을 메트릭으로 사용하는 지연 벡터 테이블을 구성하여 제약 조건을 요구하는 응용 서비스의 접속 요청이 있을 때 어느 테이블의 경로를 설정할 지를 결정하기 위해 휴리스틱 함수(heuristic function)를 사용한다. 따라서 휴리스틱 함수를 구성하는 알고리즘에 따라 루프의 발생 가능성이 존재하며 이를 제거하기 위한 알고리즘 역시 포함한다.

LDP는 단순히 지연 제약 조건 만을 만족하는 경로 설정을 위해 링크의 지연을 메트릭으로 사용하는 라우팅 알고리즘이다. 따라서 LDP는 링크의 대역폭과 같은 다른 사항들은 경로 설정 시에 고려하지 않는다.

본 논문의 구성은 3장에서는 제안하는 DRL을 위한 모델 제안의 필요성과 네트워크 모델에 대하여 논의하고 4장에서는 알고리즘 수행을 위해 각 노드가 유지해야 하는 데이터 구조와 DRL 알고리즘에 대하여 설명한다. 5장에서는 기존 알고리즘과의 성능 비교를 위한 시뮬레이션과 결과에 대하여 논의하고 마지막으로 6장에서 결론과 추후 연구 방향에 대하여 논의한다.

3. 모델 제안의 필요성 및 네트워크 모델

3.1 모델 제안의 필요성

최근 네트워크의 환경은 응용 서비스의 QoS를 보장하기 위해 다양한 기법 들을 사용하고 있다. 대표적으로 인터넷 표준화 조직에서 제안하고 있는 것이 차등 서비스(differentiated service) 모델이다. 차등 서비스 모델에서는 네트워크에 존재하는 트래픽 플로우를 몇 개의 클래스로 나누고, 이 클래스 별 우선 순위를 두어 차별적인 QoS를 제공하는 것이 기본 개념이다. 즉, 네트워크의 혼잡이 발생한 경우 우선 순위가 높은 클래스의 플로우를 우선 순위가 낮은 플로우보다 먼저 처리함으로써 QoS를 보장한다는 것이다.

그러나, 우선 순위가 높은 트래픽의 집중 현상이 발생하여 혼잡이 발생한 경우 차등 서비스 모델 만으로는 이를 해결할 방법이 없다. 이는 트래픽 플로우를 네트워크 전반에 고르게 분산시킴으로써 해결하여야 하며 이를 위해 로드 밸런싱 기능을 갖는 라우팅 알고리즘이 필요하게 된다. 이와 함께 응용 서비스의 지연과 같은 요구 사항을 수용하기 위해서는 지연 제약 조건을 만족하는 경로를 설정하는 기능을 갖추어야 한다. 따라서 본 논문에서는 로드 밸런싱 기능을 가지며 지연 제약 조건을 만족하는 경로를 설정하는 라우팅 알고리즘인 DRL을 제안한다.

3.2 네트워크 모델

점대점(point-to-point) 통신 네트워크 N은 유향 링크(directed link)로 접속된 노드 V(스위치, 라우터 등)와 링크 E의 집합 N(V, E)으로 표현할 수 있다. 여기서 각 노드를 v, 노드를 연결하는 링크를 e라고 할 때 $e_i = (v_i, v_{i+1}) \in E$이며, 경로 $P = (v_1, e_1, v_2, e_2, \cdots, e_k, v_k)$ (단, $1 \le i \le k$)이다. 소스 노드 $s \in V$와 목적지 노드 $d \in V$가 주어졌을 때 $P_{all}(s, d) = \{P_1, P_2, \cdots, P_m\}$는 s부터 d까지의 모든 경로의 집합이 된다. 따라서 경로 P_i의 비용과 종단간 지연은 다음과 같이 정의할 수 있다.

$$Cost(P_i) = \sum_{e \in P_i} C(e) \quad (1)$$

$$Delay(P_i) = \sum_{e \in P_i} C(e) \quad (2)$$

여기서 C(e)는 링크 e의 비용을 반영하는 값이며, D(e)는 패킷이 링크 e를 지날 때 겪는 지연을 의미한다. C(e)와 D(e)는 각각 양의 실수 값을 가지며 둘 다 가산형 메트릭의 예이다.

지연 제약 조건 Δ를 요구하는 접속 요청이 주어 졌을 때 선택된 경로 P_i는 $Delay(P_i) \le \Delta$를 만족해야 한다. P_i는 P_{all}의 원소이며, s부터 d까지는 부하를 분산시킬 경로가 존재할 수 있다.

$$\min_{P_i \in P'(s,d)} Cost(P_i) \quad (3)$$

따라서 비용 함수의 메트릭을 Shortest-distance 경로로 정의하고 이 때 구해진 경로에 지연 함수를 함께 사용하면 지연 제약 조건을 위배하지 않으면서 네트워크의 상태에 따라 부하를 분산 시키는 경로를 찾는 알고리즘을 만들 수 있게 된다. 식 (3)에서 P'(s, d)는 종단간 지연 제약 조건 Δ를 만족하는 s부터 d까지 경로의 집합을 의미한다.

4. 데이터 구조 및 알고리즘

4.1 데이터 구조

본 절에서는 DLR을 위해 각 노드에서 유지해야 하는 정보에 대하여 논의한다.

목적지 노드를 d, 현재 노드로부터 목적지 노드까지의 최소 비용(least cost value)을 lcv, 최소 비용을 따르는 경로의 지연 값의 합(sum of delay value)을 sdv, 최소 비용의 경로에 따른 다음 노드(least cost next node)를 lcn, 현재 노드로부터 목적지 노드까지의 최소 지연 값(least delay value)을 ldv, 최소 지연의 경로에 따른 다음 노드(least delay next node)를 ldn이라고 할 때 최소 비용의 경로를 위한 비용 벡터 테이블과 최소 지연의 경로를 위한 지연 벡터 테이블의 각 항목은 다음과 같다.

- 비용 벡터 테이블 항목
  목적지 : d

최소 비용 값 : lcv
지연 값의 합 : sdv
최소 비용 경로에 따른 다음 노드 : lcn

- 지연 벡터 테이블 항목
  목적지 : d
  최소 지연 값 : ldv
  최소 지연 경로에 따른 다음 노드 : ldn

위의 두 벡터 테이블은 로드 밸런싱을 위한 최소 비용의 경로와 지연 제약 조건을 위한 경로를 위해 현재 사용되고 있는 디스턴스 벡터 기반 라우팅 프로토콜(예를 들면 RIP)을 통한 라우팅 정보 교환 주기 동안 준비된다. 이러한 정보는 단순하고 또한 디스턴스 벡터 기반 라우팅 프로토콜이 필요로 하는 정보와 유사하여 똑 같은 프로시저를 통해 구성할 수 있다.

또한 sdv를 유지하는데 드는 비용은 비용 벡터 테이블의 최소 비용 값을 유지하기 위해 드는 비용과 같다. 즉, sdv를 계산하기 위해 각 노드는 자신의 비용과 지연을 이웃 노드에게 알리고, 이를 토대로 최소 비용의 경로(최소 비용의 다음 노드)를 따라 계속 그 값을 합하여 구해 나가게 된다. 결과적으로 최소 비용의 경로를 구할 때 sdv도 같이 계산된다.

따라서 최신의 정보가 각 노드에 유지되고 있으며 DRL에 의한 접속 설정 중에는 비용 벡터 테이블의 항목이나 지연 벡터 테이블의 항목의 변경이 없다는 것을 가정한다. 디스턴스 벡터 라우팅 프로토콜을 통해 네트워크의 상태 변화에 따른 안정화 절차나 루프 제거 절차에 대해서는 다양한 기법[6]들이 알려져 있다. 본 논문에서 논의 하지 않을 부분은 라우팅 정보의 교환과 구성 절차 그리고 안정화 절차와 루프 제거 절차이다.

RIP의 권고 사항[1]은 각 노드에서 벡터 값의 수렴을 어렵게 한다는 이유로 링크의 이용률과 같은 동적(dynamic) 메트릭을 사용하지 않도록 하고 있다. 그러나 QoS 라우팅을 필요로 하는 멀티미디어 통신 환경에서는 동적 메트릭을 사용하는 것이 네트워크 관리의 관점에서 훨씬 더 효율적이다. 따라서 본 논문에서는 비용 벡터로 가용 대역폭의 역수를 사용한다. 즉, 링크의 가용 대역폭이 넓을수록 비용은 낮아지게 되며, 반대로 가용 대역폭이 적을수록 비용은 높아지게 된다. 이는 네트워크의 로드 밸런싱 효과로 나타나게 되며 이를 5장에서 시뮬레이션을 통해 증명한다.

4.2 알고리즘

각 노드에서 경로를 선택하기 위한 DRL 알고리즘을 송신지 노드와 중계 노드로 나누어 설명한다.

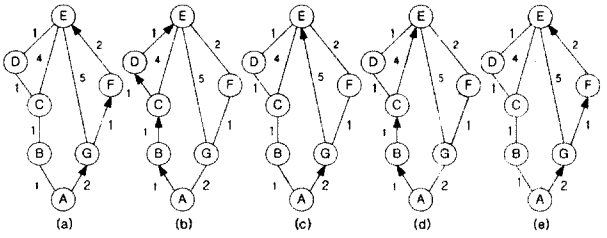
먼저 접속 요청 메시지는 응용 서비스로부터 송신지 노드에게 전달된다. 따라서 접속 요청은 네트워크 관점에서 송신지 노드로부터 시작한다. 송신지 노드 s가 지연 제약 조건 Δ를 갖는 목적지 노드 d로의 접속 요청을 받게 되면 먼저 지연 벡터 테이블의 ldv를 검사하게 된다. 즉, 네트워크의 현재 상태가 최소 지연 값을 허용할 수 있는지를 비교한다. 만일 ldv가 Δ 보다 크다면 현재 네트워크는 접속 실패 메시지를 해당 응용 프로세스에게 보내어 접속할 수 없음을 알리고 알고리즘은 종료한다. 그러나 ldv가 Δ 보다 작아서 이를 허용할 수 있다면 네트워크는 경로 설정을 통해 접속 요청을 처리할 수 있음을 나타내게 된다. 그 후 송신지 노드에서는 최소 비용의 경로(LC 경로 : Least Cost 경로)를 선택할 지 아니면 최소 지연의 경로(LD 경로 : Least Delay 경로)를 선택할 지를 결정하기 위해 다시 비용 벡터 테이블의 sdv 항목과 Δ를 비교한다. LD 경로와 LC 경로 둘 다를 만족하는 경우, LC 경로를 따라 경로 설정하는 것이 더욱 효율적일 것이다. 응용 서비스는 최대한 Δ 만큼의 지연을 갖는 접속 요청을 하고 있으므로 네트워크에서 이보다 더 작은 지연을 제공하는 경로를 찾을 필요는 없기 때문이다. 따라서 sdv가 Δ 보다 작다면 최소 비용의 지연 제약 조건을 만족하는 경로를 통해 접속을 할 수 있게 된다. 그러나 sdv가 Δ 보다 크게 되면 알고리즘은 지연 벡터 테이블을 따라 ldn으로 접속 요청 메시지를 보내게 된다.

거쳐온 경로의 지연 값을 구하기 위해 접속 요청 메시지를 다음 노드로 전달하기 전에 현재 노드로부터 다음 노드까지의 지연 값을 구해야 한다. 현재 노드까지의 지연 값은 dsf(delay_so_far)로 표현되며, 송신지 노드로부터 현재 노드까지의 지연 값의 합을 의미한다. 따라서 송신지 노드의 dsf는 0이 되며, 임의의 노드에서 LD 경로 혹은 LC 경로를 선택하기 위해 사용되는 값이 된다. 현재 노드와 다음 노드 사이의 지연 값은 3.2절에서 논의한 D(cn, nn)에 의해서 구할 수 있다. cn(current node)은 현재 노드를 의미하며, nn(next node)는 LD 경로 혹은 LC 경로에 의한 다음 노드를 의미한다. 즉, 다음 노드로 보내는 접속 요청 메시지의 dsf에는 dsf + D(cn,nn)의 값을 설정하게 된다. 마지막으로 송신지 노드에서는 다른 접속 요청과의 구분을 위해 식별자를 발행하고, 현재 접속 요청을 위한 라우팅 엔트리를 생성하며 선택된 다음 노드로 접속 요청 메시지를 보내게 된다.

접속 요청 메시지를 받은 중계 노드는 이전 노드에서 보낸 dsf에 근거하여 Δ와 ldv + dsf, Δ와 sdv + dsf를 비교하여 LD 경로 혹은 LC 경로를 선택하게 된다. 마찬가지로 다음 노드로 메시지를 보내기 전에 dsf를 계산하고 송신지 노드와 마찬가지로 라우팅 엔트리를 생성하며, 이러한 절차가 목적지 노드인 d에 도달할 때까지 반복적으로 수행된다.

결과적으로 DRL을 통해서 LD 경로를 따라가더라도 임의의 노드에서 LC 경로를 찾게 되면 그 노드로부터 목적지까지는 LC 경로를 따라 전송하게 된다. 즉, LC 경로를 따라가다가 LD 경로를 찾는 경로는 발생하지 않는다는 것이

다. 이는 경로 설정 중에 루프가 발생하지 않는다는 것을 의미하게 되며 휴리스틱 함수을 사용함으로써 접속 경로 설정 단계 중에 루프를 검출하고 제거하는 알고리즘을 포함하고 있는 DCUR에 비해 간단하고 효율적이라는 것을 의미한다.



(그림 4-1) 경로 설정 예

(그림 4-1)은 각 알고리즘별 경로 설정을 나타내고 있다. 예제 네트워크에서는 단순한 경로 설정을 위해 무향 링크를 가정했으며, 각 링크의 지연은 모두 1로 가정하여 그림에 표시하지는 않았다. 링크와 같이 표기된 숫자는 비용을 나타낸다. 송신지 노드를 A, 목적지 노드를 E, 지연 제약 조건 3을 가정했을 때 경로 설정을 나타낸다.

먼저 (a)의 경우 최적화된 경로 설정을 나타낸다. 즉, 지연 제약 조건 3을 만족하며, 비용 5인 A-G-F-E의 경로를 설정하여 최적화된 경로 설정을 나타낸다.

(b)의 경우는 지연 제약 조건과는 관계없이 최소 비용의 경로를 나타낸다. 즉, A-B-C-D-E의 경로 비용은 4이며, 지연은 4가 된다.

(c)의 경우 LDP 알고리즘에 따른 최소 지연 경로를 나타낸다. 즉, A-G-E의 경로를 선택하게 되며, 경로 비용은 7, 지연은 2가 된다.

(d)의 경우 DCUR 알고리즘에 따른 경로를 나타낸다. 즉, A-B-C-E의 경로를 선택하며, 경로 비용은 6, 지연은 3이 되어, 지연 제약 조건을 만족하는 경로 설정을 하고 있다.

(e)의 경우 DRL 알고리즘에 따른 경로를 나타낸다. (a)의 예와 마찬가지로 지연 제약 조건을 만족하는 최소 비용의 경로 설정을 하고 있다. <표 4-1>에 목적지 E에 대해 각 노드가 유지하고 있는 비용 벡터 테이블과 지연 벡터 테이블을 정리하였다. 괄호 문자로 표기된 노드가 각 노드에서 경로 설정에 사용한 다음 노드이다.

〈표 4-1〉 DRL의 라우팅 테이블

노드 / 테이블	A		G		F	
	항 목	값	항 목	값	항 목	값
비용 벡터	지연값의합	4	지연값의합	2	지연값의합	1
	다음노드	B	다음노드	(F)	다음노드	(E)
지연 벡터	최소지연값	2	최소지연값	1	최소지연값	1
	다음노드	(G)	다음노드	E	다음노드	E

5. 시뮬레이션

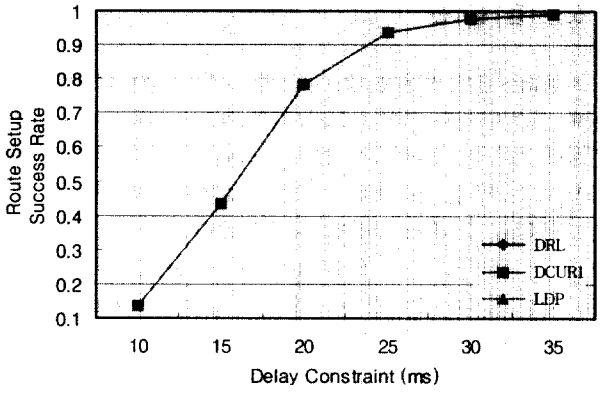
시뮬레이션에 사용한 네트워크의 크기는 $4000 * 2400$ Km^2 이며 링크의 전파 지연 속도를 빛의 속도의 2/3로 계산할 때 실험에 사용한 네트워크는 $20 * 12$ $msec^2$이 된다. 네트워크의 노드 수는 각 20, 50, 100 노드로 구성했으며 링크는 전이중의 50Mbps, 75Mbps, 100Mbps, 125Mbps, 155Mbps로 구성하고 각 링크의 큐잉 요소(Queuing component)는 무시했다. 또한 링크의 지연은 대칭적이나 링크의 비용은 비대칭적으로 설정했다. 즉, $D(u, v) = D(v, u)$이지만, $C(u, v) \neq C(v, u)$이다. 이는 링크의 길이가 같으면 지연도 같게 되지만, 링크의 비용은 전이중 방식이므로 다르게 표현되기 때문이다. 네트워크를 생성하기 위해 NS-2 (Network Simulator-2)와 함께 제공되는 GT-ITM 토폴로지 생성기[8]를 사용하여 순수 랜덤 그래프(Pure random graph)를 생성하였다. 생성된 그래프의 평균 노드 정도(Degree)는 4이상으로 설정했으며 시뮬레이션 시간은 60초를 기준으로 했다[9].

각 시뮬레이션에서는 가용 대역폭의 역수를 비용으로 사용하는 DRL과 이용률을 비용으로 사용하는 DCUR, 그리고 지연을 비용으로 사용하는 LDP를 비교하였다. 또한 결과의 신뢰성을 높이기 위해 네트워크 토폴로지를 변경하며 각 실험을 반복했으며, 그 평균값을 결과 그래프로 표현했다. 각 실험은 먼저 토폴로지를 생성하고, 이 때 각 알고리즘별 비용을 토대로 라우팅 테이블을 구하여 시뮬레이션 프로그램의 입력으로 사용하였다. 또한 시뮬레이션 프로그램에 대역폭 요구 사항과 지연 제약 조건을 변경하여 가며 각 알고리즘의 경로 설정율, 응용 서비스의 접속 성공율 등을 계산하였다.

실험은 크게 두 단계로 나누어 진행했다. 첫 번째 실험은 지연 제약 조건을 만족하는 경로를 설정할 수 있는지, 설정할 수 있는 경우 사용한 홉 수는 얼마인지를 각 알고리즘별로 분석하기 위한 것이었다. 두 번째 실험은 각 알고리즘의 로드 밸런싱 능력을 상대적으로 비교 분석하기 위해, 응용 서비스의 접속 요청이 발생한 경우 각 알고리즘 별로 이를 얼마나 수용하는 지를 비교 분석했다.

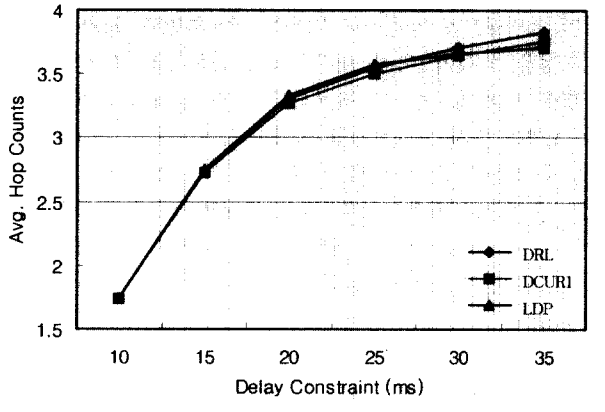
첫 번째 실험에서는 각 20 노드, 50 노드, 100 노드일 때 지연 제약 조건을 10ms, 15ms, 20ms, 25ms, 30ms, 35ms로 변경하며 무작위 송신지에서 무작위 목적지로의 접속 요청을 하였을 때 DRL, DCUR, LDP의 평균 홉 수, 평균 경로 설정 성공률을 측정했다. 즉, 각 알고리즘 별 경로 설정 기능의 일반적인 항목을 비교 분석하기 위한 것이다.

(그림 5-1)에서와 같이 각 라우팅 알고리즘은 모두 동일한 경로 설정 성공률을 나타냈다(그래프가 하나로 보이는 것은 모두 동일한 곡선을 그리고 있기 때문이다.). 즉, 지연 제약 조건이 10ms일 때 경로를 설정할 확률이 매우 낮았으나 지연 제약 조건이 20ms가 되면 약 80%의 성공률을 보이며, 25ms일 때는 95%의 성공률을 보이고 있다. 이는 DRL, DCUR, LDP 알고리즘 모두가 네트워크 상에 지연 제



(그림 5-1) 경로 설정 성공율

약 조건을 만족하는 경로가 존재한다면 모두 경로 설정할 수 있다는 것을 나타낸다. 그러나 이러한 결과가 곧 응용 서비스의 접속 요구에 대한 접속 성공율을 의미하지는 않는다. 멀티미디어 통신 환경에서 응용 서비스는 일반적으로 지연 제약 조건과 함께 일정한 대역폭을 필요로 하기 때문이다.



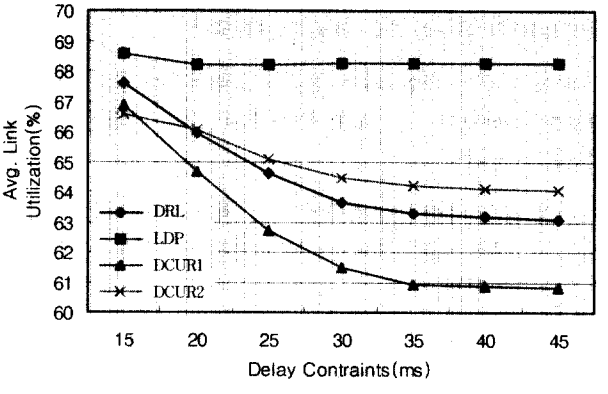
(그림 5-2) 평균 홉 수

(그림 5-2)는 각 라우팅 프로토콜이 경로 설정에 사용한 평균 홉 수를 나타낸다. 지연 제약 조건이 엄격할 때는 모두 LD 경로를 따라 경로를 설정하게 되므로 평균 홉 수가 비슷하게 나타나게 되나, 지연 제약 조건이 완화되면서 알고리즘에 따라 평균 홉 수가 다르게 나타난다. 즉, LDP는 항상 LD 경로를 따르게 되므로 가장 짧은 경로를 따라 경로를 설정하게 되지만, DRL과 DCUR은 QoS 라우팅을 위해 각각 LC 경로를 찾게 되고, 이에 따라 더 많은 홉을 경유하여 경로 설정을 하게 된다. (그림 5-2)에서 평균 홉 수의 차이가 커 보이지 않는 이유는 무작위 송신지 노드에서 무작위 목적지 노드로의 경로 설정을 실험했기 때문에 송신지와 목적지가 같은 경우도 포함되어 있으며, 20노드, 50노드, 100노드의 실험 결과를 평균하여 표현했기 때문이다. 지연 제약 조건이 엄격할 때는 평균 홉 수가 2 미만이므로 이웃 노드를 넘어서 경로 설정을 하지 못하며, 지연 제약 조건이 완화 되면서 여러 홉을 거쳐 경로 설정할 수 있음을 보이고 있다. 이 때 DRL과 DCUR은 로드 밸런싱을 위해 LDP보다 긴 평균 홉 수를 사용하여 경로 설정을 하고 있다.

두 번째 실험은 DRL, DCUR, LDP의 각 알고리즘이 응용 서비스의 접속 요청에 대해 어느 정도의 로드 밸런싱을 하는지를 파악하기 위한 것이다.

50 노드를 갖는 네트워크에서 토폴로지와 각 링크의 비용(DRL의 경우 가용 대역폭의 역 수, DCUR의 경우 이용률, LDP의 경우 지연)을 변경하여 가며 각 라우팅 알고리즘이 응용 서비스의 접속 요청을 어떻게 처리하는 지를 반복적으로 실험했다.

실험에 사용한 네트워크의 노드 정도(degree)는 평균 4 이상이며, 0.1초 단위로 1Mbps에서 5Mbps 사이의 대역폭(평균적으로 3Mbps)을 필요로 하는 접속 요청을 60초 동안 생성했다. 이 때 모든 송신지 노드에서 모든 목적지 노드로 접속 요청을 발생시켜 가며 네트워크의 동작을 관찰했다. 실험에 사용한 네트워크는 50Mbps, 75Mbps, 100Mbps, 125Mbps, 155Mbps의 대역폭을 갖는 링크를 무작위로 생성하였으며, 또한 각 링크의 비용도 무작위로 설정했다. 이 때 반복 실험에 사용된 각 링크의 평균 대역폭은 102.71Mbps 이며, 평균 이용률은 63%, 평균 가용 대역폭은 37.96Mbps 이다.



(그림 5-3) 평균 이용률

(그림 5-3)은 각 라우팅 알고리즘이 응용 서비스의 접속 요청 시 각 트래픽 플로우에 사용한 경로의 평균 이용률을 나타내고 있다. DCUR과의 비교를 보다 구체적으로 하기 위해 DCUR의 메트릭을 이용률과 가용 대역폭의 역수 두 가지를 사용하였다. 즉, 그림에서 DCUR1로 표시된 부분은 비용 벡터의 메트릭이 이용률일 때이며, DCUR2로 표시된 부분은 DRL과 마찬가지로 가용 대역폭의 역수일 때를 표현한 것이다.

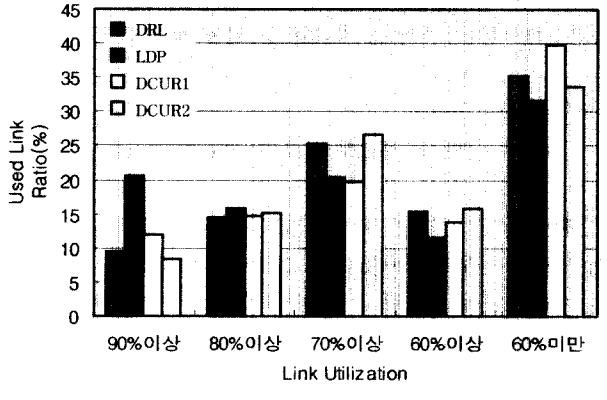
지연 제약 조건이 엄격할 경우 모두 LD 경로를 따라 경로 설정을 하기 때문에 링크의 이용률에 큰 차이가 없으나 지연 제약 조건이 완화되면서 각 알고리즘별로 링크의 평균 이용률이 다르게 나타난다. 즉, LDP의 경우 링크의 가용 대역폭이나 이용률과는 무관하게 경로를 설정하므로 항

상 높은 링크 이용률을 보인다. 반면, DCUR1의 경우 다른 라우팅 알고리즘에 비해 낮은 링크의 이용률을 보여 링크의 이용률 면에서는 우수한 성능을 나타내는 것처럼 보이며 DCUR2의 경우는 오히려 DRL보다도 낮은 성능을 나타내는 것처럼 보인다.

그러나 여기에는 고려해야 할 중요한 사항이 있다. 즉, 낮은 링크의 이용률이 응용 서비스의 접속 성공과 직접적으로 연결되지는 않는다는 것이다. 예를 들면 A 노드와 B 노드 사이에 각각 대역폭이 10Mbps인 링크로 구성된 경로1과 100Mbps인 링크로 구성된 경로2의 두 경로가 있다고 하자. 경로1의 링크 수와 경로 2의 링크 수는 같고 경로1의 각 링크 이용률은 70%, 경로 2의 각 링크 이용률은 90%라고 하면 DCUR1은 경로 1을 경로 설정에 이용할 것이다. 이 때 1Mbps의 대역폭을 요구하는 트래픽 플로우가 계속적으로 발생한다면, DCUR1은 3개의 요청 밖에 처리할 수 없을 것이다. 그러나 가용 대역폭의 역수를 비용으로 사용하는 DRL이나 DCUR2는 경로 1의 비용을 1/3*링크수, 경로 2의 비용을 1/10*링크수로 계산하고 있기 때문에 경로2로 경로를 설정할 것이고 10개의 요청을 처리할 수 있을 것이다.

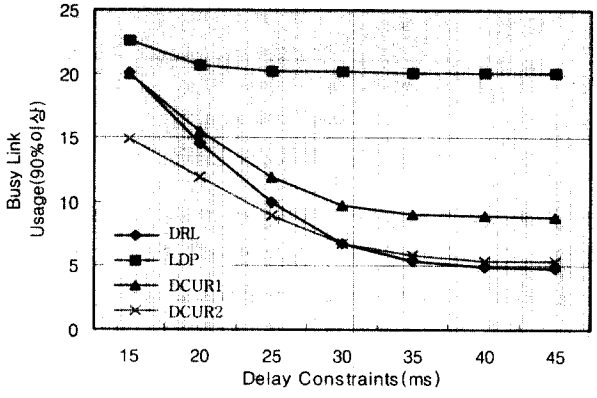
물론 이러한 가정은 라우팅 정보 교환 주기 내에 발생할 수 있는 단적인 예일 뿐이다. 그러나 각 트래픽 플로우가 발생했을 때마다 라우팅 테이블을 갱신하지 않는다면 이러한 가정은 타당성을 갖게 된다. 실제로 라우팅 정보 교환은 일정한 주기를 가지고 일어나게 되며, 라우팅 정보 교환 주기 내에 발생된 모든 트래픽 플로우는 모두 동일한 라우팅 테이블에 의해 경로가 설정된다고 볼 수 있다. 즉, 이전에 두 경로의 예가 설득력을 갖는다고 볼 수 있다. 물론 라우팅 정보 교환 주기를 짧게 할수록 더욱 효율적인 경로 설정이 가능하겠지만 그에 따른 오버 헤드를 고려해야 한다.

(그림 5-4)는 각 알고리즘 별로 트래픽 플로우를 처리할 때 사용된 링크의 비율을 표현하고 있다. 실험에 사용된 네트워크의 평균 이용률이 63%이고, DRL과 DCUR1 및 DCUR2의 사용된 링크의 비율이 70-80%를 중심으로 한 포물선을 그리고 있는 것으로 보아 모두 로드 밸런싱을 위한 경로 설정을 하고 있다고 볼 수 있지만 중요한 것은 90%이상의 이용률을 보이는 링크를 얼마나 사용했는가 하는 것이다.



(그림 5-4) 이용률별 사용된 링크 비율

즉, 모든 트래픽 플로우의 경로 설정 시 90%가 넘는 이용률을 갖는 링크에 대해 DRL은 약 9.5%를 사용했지만 DCUR1은 12%, DCUR2는 8.5%를 사용한 것으로 확인되었다. 이는 응용 서비스의 접속 성공율에 중요한 요소로 작용한다.



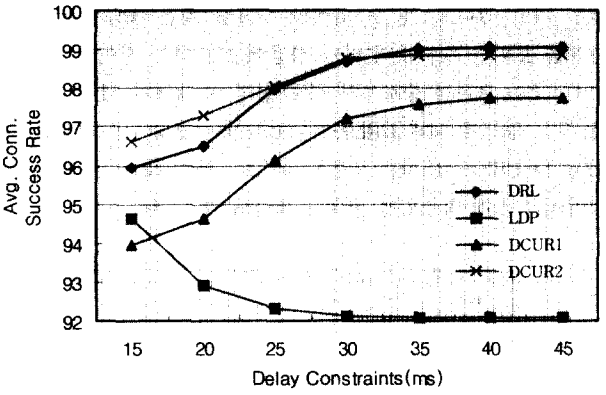
(그림 5-5) Busy Link 사용율

(그림 5-5)는 (그림 5-4)에서 90%이상의 이용률을 갖는 링크만을 지연 제약 조건과 함께 표현한 것이다. 각 알고리즘이 지연 제약 조건이 주어졌을 때 얼마나 많은 과부하 링크(Busy Link)를 사용하여 접속 요청을 처리하고 있는지를 표현하고 있다. 지연 제약 조건이 엄격한 경우 LD 경로를 따라 경로 선택이 이루어 지므로 선택의 여지가 없게 된다. 그러나 지연 제약 조건이 완화되면서 DRL와 DCUR1 및 DCUR2는 점차 LC 경로를 찾아 경로를 선택하게 된다. 이 때 DCUR1의 경우 링크의 이용률을 비용으로 사용하고 있기 때문에 DRL보다 가용 대역폭이 적은 경로를 선택할 확률이 높아지게 된다. 즉, 라우팅 정보 교환 주기 내에 일정한 대역폭을 요구하는 접속 요청이 계속하여 발생하는 경우 DCUR1에서는 접속 요청에 대해 자원의 부족(링크의 대역폭 부족)으로 인한 접속 실패가 일어날 수 있게 된다. 결과적으로 과부하 링크를 사용할 확률이 높아지게 되면 이는 응용 서비스의 접속 실패로 나타나게 된다.

이에 반해 DRL 및 DCUR2는 가용 대역폭이 넓은 경로를 찾아 경로를 설정함으로써 일시적으로 폭주하는 트래픽을 보다 많은 경로로 분산시킴을 알 수 있다.

LDP의 경우는 지연을 비용으로 사용하기 때문에 링크의 이용률과는 무관하게 경로 설정을 함으로써 다른 라우팅 알고리즘에 비해 응용 서비스의 접속 실패 확률이 더욱 높게 나타난다.

(그림 5-6)은 평균 3Mbps의 대역폭(1Mbps에서 5Mbps의 무작위 대역폭)을 필요로 하는 응용 서비스의 접속 요청에 대한 접속 성공율을 나타낸다. 로드 밸런싱 성능을 비교 분석하기 위해, 응용 서비스의 접속 요청 건수에 대한 각 알



(그림 5-6) 평균 접속 성공율

고리즘별 성공적인 접속 처리 건수의 비율을 그래프로 표현했다. 즉, 단위 시간 내 응용 서비스의 접속 요청 총 수를 x라하고 이 때 네트워크가 성공적으로 처리한 응용 서비스의 접속 수를 y라 할 때 응용 서비스의 접속 성공율은 y/x*100으로 표현되며, 이는 각 알고리즘별로 상대적인 로드 밸런싱 능력을 나타낸다.

50 노드를 갖는 네트워크에서 모든 송신지 노드로부터 모든 목적지 노드로의 접속 요청을 가정하면 총 2,500개의 접속 요구가 발생한다. (그림 5-6)은 대역폭과 지연 제약 조건을 만족하는 접속 요청 중에서 어느 정도의 비율로 경로 설정에 성공하는 지를 나타내고 있다. 지연 제약 조건이 점차로 완화되면서 DRL은 99%의 응용 서비스 접속 성공율을 보이고 있다. 반면에 DCUR1은 DRL보다 평균 2%정도 낮은 성공율을 보이고 있으며 DCUR2는 지연 제약 조건이 엄격할 때 DRL보다 높은 성공율을 보인다. 그러나 지연 제약 조건이 25ms에서 30ms 사이인 경우 DCUR2는 DRL과 비슷한 성공율을 보이며 30ms가 넘어서면 DRL이 DCUR에 비해 근소하지만 높은 성공율을 보인다.

결과적으로 일정 수준의 대역폭 요구를 수반하는 멀티미디어 통신 환경을 고려할 때 이용률을 메트릭으로 사용하기 보다는 가용 대역폭을 사용하는 것이 보다 효율적임을 알 수 있으며 루프가 발생하지 않는 DRL이 DCUR이나 LDP에 비해 효율적인 경로 설정을 하고 있다.

6. 결론 및 추후 연구 방향

본 논문에서는 멀티미디어 통신을 위해 네트워크에서 고려해야 할 사항을 알아보았다. 일반적으로 여러 제약 조건을 만족하는 경로를 찾는 것은 NP-complete 문제임을 감안할 때 선택할 수 있는 제약 조건은 대표적으로 대역폭과 지연일 것이다. 이는 멀티미디어 통신, 즉 차별화된 서비스를 필요로 하는 응용 서비스를 위한 최소한의 제한 사항이며, 또한 네트워크 관리에서 요구하는 최소한의 제한 사항이기도 하다.

이를 동기로 사용자에게는 접속 지향의 지연 제약 조건을 만족하는 경로 설정을 하면서 네트워크 관리자에게는 로드 밸런싱 기능을 제공하는 간단하고 분산된 루프프리 라우팅 알고리즘 DRL을 제안하고 다른 라우팅 알고리즘과 비교하여 그 우수성을 시뮬레이션을 통하여 증명하였다.

DRL은 응용 서비스의 지연 제약 조건 요구가 있을 때 네트워크에 이를 만족하는 경로가 존재한다면 이러한 경로를 찾아 경로를 설정할 수 있다는 것을 시뮬레이션 결과를 통해 입증했다. 이때 LDP 및 DCUR과 같은 제약 기반 알고리즘과 동일한 홉 수를 사용하여 제약 조건을 만족하는 경로 설정을 함으로써 성능에 큰 차이가 없음을 시뮬레이션 결과는 보여주고 있다.

또한 DRL은 트래픽 플로우를 네트워크 전반에 고르게 분산시키는 로드 밸런싱 기능 면에서 다른 알고리즘에 비해 우수한 성능을 갖고 있다는 것을 시뮬레이션 결과는 보여 주고 있다. 즉, 가용 대역폭이 보다 넓은 경로를 찾아 경로를 설정함으로써 동일한 네트워크 자원으로 보다 많은 응용 서비스 트래픽 플로우를 지원함으로써 응용 서비스의 접속 성공율을 높이고, 네트워크 자원의 이용률을 최대한 높이고 있다.

DCUR과 같은 라우팅 알고리즘은 휴리스틱 함수를 사용함으로써 경로 설정 중에 루프가 발생할 가능성이 있으나, DRL은 루프가 발생하지 않으며, 따라서 더 간단한 경로 설정을 수행하며 더 적은 메시지로 경로 설정이 가능한 장점이 있다.

현재의 통신 요구 사항이 점차 지연에 민감해지고, 더 많은 대역폭을 요구하는 시점에서 DRL은 그러한 기대에 부응할 것이다. 또한 네트워크 운영의 관점에서 차별화된 서비스 제공을 위해 트래픽 플로우를 분산시켜야 하고 네트워크 자원의 이용률을 극대화해야 한다는 요구 사항을 감안할 때 DRL을 이러한 요구 사항을 수용하는 알고리즘이라 하겠다.

추후 연구 사항으로는 보다 개선된 로드 밸런싱 기능 지원을 위해 DRL을 멀티캐스트 라우팅으로 확장하여 한차원 높은 멀티미디어 통신을 지원할 수 있을 것이다.

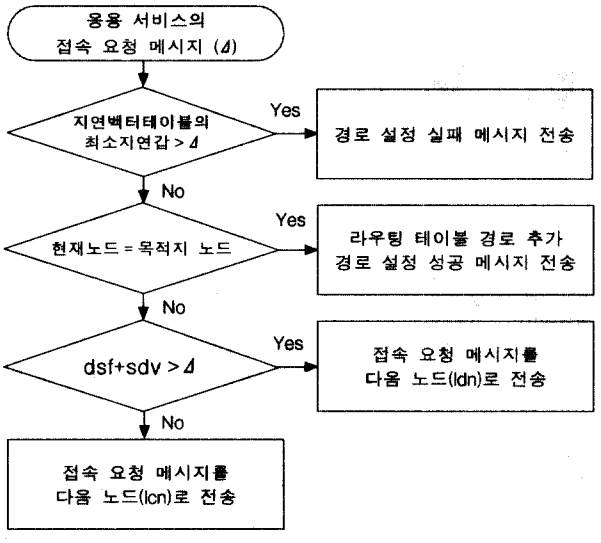
부 록

DRL의 플로우 차트 및 Pseudo Code

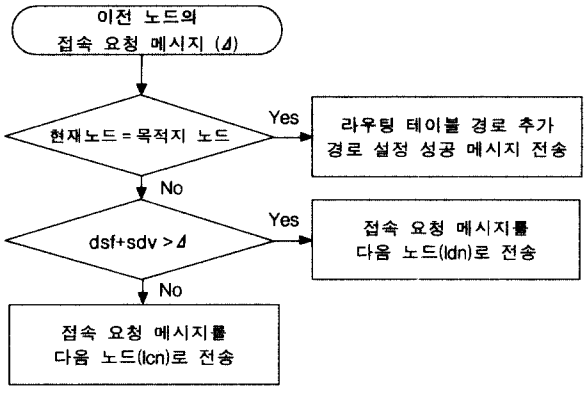
DRL의 경로 설정을 위한 처리 흐름을 송신지 노드와 중계 노드로 나누어 표현했다. (그림 A-1)은 송신지 노드에서의 처리 흐름을, (그림 A-2)는 중계 노드에서의 처리 흐름을 간단하게 표현하고 있다. 송신지 노드와 중계 노드의 차이점은 송신지 노드에서는 응용 서비스가 요청한 지연 제약 조건을 만족하는 경로를 설정할 수 있는 지 여부를 검

사하는 부분이 있다는 것이다.

(그림 A-1), (그림 A-2)에서 dsf는 현재 노드까지의 지연 값의 합을 의미하며, sdv는 비용 벡터 테이블의 경로를 따라 현재 노드로부터 목적지 노드까지 경로 설정을 할 때 지연 값의 합을 의미한다. 따라서 이 두 값의 합이 응용 서비스가 요청한 지연 제약 조건 Δ보다 작다면 최소 비용의 경로를 따라 경로를 설정하게 된다. 만약 두 값의 합이 Δ보다 크다면 최소 지연의 경로를 따라 경로를 다음 노드로 메시지를 보내게 된다.



(그림 A-1) 송신지 노드에서의 처리 흐름



(그림 A-2) 중계 노드에서의 처리 흐름

다음은 응용 서비스의 접속 요청에 대한 송신지 노드와 중계 노드에서의 처리 과정을 Pseudo Code로 나타낸 것이다.

route_setup_message(route_ID,source,destination,delay constraint,delay_so_far, current node,next node)를 통해 각 노드사이에 경로를 설정한다.

다음은 응용 서비스부터 접속 요청을 받았을 때 송신지 노드에서 수행하는 초기화 함수이다.

```
Init_of_Route_Setup(source sa, destination d, delay_constraint Δ)
{
   if (ldv > Δ)
      send route_setup_failure_message to source application ;
   else {
      pn = null ;
      cn = s ;
      dsf = 0 ;
      r = route_ID ;
      Call Route_Setup(r, s, d, Δ, dsf, pn, cn)
   }
}
```

다음은 각 노드에서 경로 설정 시 호출되는 함수이다. 각 노드에서는 현재 노드가 목적지 노드(d)인지를 판단하고, 목적지 노드라면 접속 완료를 통보한다. 목적지 노드가 아니라면 dsf를 토대로 LC 경로 혹은 LD 경로를 결정하게 되며, 다음 노드까지의 dsf를 계산하여 다음 노드로 경로 설정 메시지를 보내게 된다.

```
Route_Setup(route_ID r,source s,destination d, delay_constraint
Δ,delay_so_far dsf, previous_node pn,current_node cn) {
  if (cn = d)
  Call Route_Setup_Completion(r,s,d,Δ,dsf,pn) ;
  else
  Call Route_Setup_Processing(r,s,d,Δ,dsf,pn,cn) ;
}
```

현재 노드가 목적지 노드인 경우 경로 설정 완료 메시지를 보낸다.

```
Route_Setup_Completion (route_ID r,source s, destination
d, delay_constraint Δ,delay_so_far dsf,previous_node pn) {
  nn = null ;
  Call Route_Add (r,s,d,Δ,dsf,pn,nn)
  send route_setup_success_message to s ;
}
```

현재 노드가 목적지 노드가 아닌 경우 dsf에 의해 다음 노드를 결정하고, 다음 노드까지의 dsf를 계산하여 다음 노드로 경로 설정 메시지를 보낸다.

```
Route_Setup_Processing(route_ID r,source s, destination
d, delay_constraint Δ,dealy_so_far dsf, previous_node
pn,current_node cn) {
  if (dsf + sdv > Δ) {
    dsf = dsf + D (cn, ldn)
  nn = ldn ;
  Call Route_Add (r, s, d, Δ, dsf, pn, nn) ;
  send route_setup_message (r,s,d,Δ,dsf,cn,ldn) to ldn ;
  }
  else {
  dsf = dsf + D (cn, lcn)
  nn = lcn ;
  Call Route_Add(r, s, d, Δ, dsf, pn, nn) ;
  send route_setup_message (r,s,d,Δ,dsf,cn,lcn) to lcn ;
  }
}
```

현재 노드의 라우팅 테이블 엔트리에 설정된 경로를 추가한다.

```
Route_Add(route_ID r,source s,destination d, delay_constraint
  Δ, delay_so_far dsf,previous_node pn,next_node nn) {
   route_ID = r ;
   source = s ;
   destination = d ;
   delay_constraint = Δ ;
   delay_so_far = dsf ;
   previous_node = pn ;
   next_node = nn ;
   create routing table entry with route_ID,

source,destination,delay_constraint,delay_so_far,previous_node,
   next_node ;
}
```

참 고 문 헌

[1] C. Hedrick. Routing information protocol, June, 1988. RFC 1058.

[2] J. Moy, OSPF Version 2. Mar, 1994. RFC 1583.

[3] R. Guerin, S. Kamat, A. Orda, T. Przygienda, and D. Williams, QoS Routing Mechanisms and OSPF extensions, Internet draft<draft-guerin-QoS-routing-ospf-03.txt>, Jan. 1998.

[4] Z. Wang and J. Crowcroft, Quality of Service Routing for Supporting Multimedia Applications, IEEE JSAC, Sept. 1996.

[5] Q. Ma, QoS Routing in the Integrated Services networks, Ph. D. thesis, CMU-CS-98-128, Jan. 1998.

[6] C. Cheng, R. Riley, S. Kumar, and J. Garcia-Luna-Aceves, A loop-free extended Bellman-Ford routing protocol without bouncing effect, in Proceeding ACM SIGCOMM89, pp.224-236. Sep. 1989.

[7] S. Reeves, and F. Salama, A Distributed Algorithm for Delay-Constrained Unicast Routing, IEEE/ACM Transactions on Networking, Vol.8. No.2. pp.239-250. Apr. 2000.

[8] http : //www.isi.edu/nsnam/ns/ns-topogen.html.

[9] S. Bradner, Benchmarking Methodology for Network Interconnect Devices, RFC 2544, March. 1999.

최 영 수

e-mail : yschoi@songgang.skku.ac.kr

1988년 성균관대학교 정보공학과(공학사)

1997년 성균관대학교 대학원 정보공학과 (공학석사)

1997년~현재 성균관대학교 대학원 전기전자및컴퓨터공학부 박사과정

1991년~1993년 삼성SDS 정보통신추진실

1997년~1998년 ETRI 네트워크컴퓨팅부 위촉연구원

관심분야 : 네트워크관리, 인터넷 QoS, 초고속 통신망

정 진 욱

e-mail : jwchung@songgang.skku.ac.kr

1988년 성균관대학교 전기공학과(공학사)

1997년 성균관대학교 대학원 전자공학과 (공학석사)

1997년~현재 서울대학교 대학원 계산통계학과 이학박사

1982년~1985년 한국과학기술연구소 실장

1981년~1982년 Racal Milgo Co. 객원연구원

1985년~현재 성균관대학교 전기전자및컴퓨터공학부 교수

관심분야 : 컴퓨터 네트워크, ATM 트래픽 관리, 네트워크보안, 고속 및 무선 네트워크 프로토콜