

확장가능한 2단계 QoS 라우팅 방식

정희원 김 승 훈*

Scalable Two Phases QoS Routing Scheme

Seung-Hoon Kim* *Regular Member*

요 약

본 논문에서는 대규모 고속 통신망에서 분산 멀티미디어 응용을 지원하기 위한 확장가능한 2단계의 QoS 라우팅 방식을 제안한다. QoS 라우팅 문제는 다중적도 최단거리 문제로 변형될 수 있으며 이는 NP-complete로 알려져 있다. 계층적 통신망을 위하여 제안된 라우팅 방식은 두 단계로 이루어져 있다. 첫번째 단계는 특정 응용의 요구와 관계없이 모든 수준의 도메인에서 들어오는 경계노드와 나가는 경계노드 쌍간에 도메인을 경유하는 횡단트래픽을 위한 경로의 QoS distance 값을 정기적으로 선계산한다. 두번째 단계는 응용에서의 요구가 왔을때 실행되며, 계층적 완전경로 계산에 필요한 그래프 구축을 위한 분산알고리즘이 실행된다. 이와같이 자원가용성 정보를 공시하지 않고 조회방식을 사용함으로써 정보의 flood로 생성되는 많은 통신 오버헤드를 줄일 수 있다. 제안된 알고리즘에서는 또한 기존의 방식에서 완전히 무시되거나 혹은 부분적으로 고려되었던 통신망의 위상도 고려하여 계층적 완전경로의 불완전성을 보완하였다. 마지막으로 대략적으로 계산된 계층적 완전경로로부터 QoS 라우팅 문제를 체계적으로 분할하여 수준1 도메인의 경로로 확장하는 분산 알고리즘을 제안하였다.

ABSTRACT

In this paper a scalable QoS routing scheme for distributed multimedia applications in a hierarchical wide area network is proposed. The problem of QoS routing is formulated as a multicriteria shortest path problem, known as NP-complete. The proposed hierarchical routing scheme consists of two phases. In Phase I, every border node periodically pre-computes the QoS distance for the paths between every pair of border nodes in any level of domain hierarchy. This phase is independent of the QoS request from an application. In Phase II, distributed graph construction algorithm is performed to model the network as a graph by retrieving pre-computed QoS distances. The graph is constructed by the on-demand algorithm and contains a part of the network topology which is completely neglected or partially considered by existing routing schemes, thus maintaining more accurate topology information. By using retrieval approach rather than advertising one, no global QoS state information exchange among nodes is needed. In this Phase, distributed partition algorithm for QoS routing problem is also performed, thus eliminating virtual links on the hierarchically complete path.

Key Words : QoS routing; routing problem partition; available resource information retrieval

* 단국대학교 전자컴퓨터학부(edina@dankook.ac.kr)

논문번호 : 030345-0807, 접수일자 : 2003년 8월 7일

※본 연구는 2002학년도 단국대학교 대학연구비의 지원으로 연구되었습니다.

I. 서론

분산 멀티미디어 응용(distributed multimedia application)마다 요구되는 다양한 특성이 있으며, 분산 멀티미디어 시스템은 많은 경우에 이질적(heterogeneous) 환경이므로, 통신망은 다양한 품질의 서비스(Quality of Service)를 제공할 필요가 있다. 응용이 요구하는 서비스를 몇 개의 등급으로 한정하는 경우도 있으나 응용의 다양성을 지원하지 못한다는 단점이 있다. 응용이 요구하는 다양한 QoS를 제공하기 위하여 통신망에 여러가지 기법이 요구된다. 응용의 다양한 QoS 요구(QoS requirements)를 보장하기 위하여, 기존에 설정된 연결의 QoS 에 영향을 미치지 않으면서 새로운 응용의 요구를 만족할 수 있는 충분한 자원을 갖는 경로를 발견하여야 하며 이와 같은 과정을 QoS 라우팅(QoS routing)이라 한다.

지역적으로 수집한 흐름차단통계(blocking statistics)에 근거하여 노드가 통신망 QoS 상태를 추측하고 경로를 선택하는 방식으로 전체 QoS 상태정보를 전혀 필요로하지 않는 지역적인 QoS 라우팅 방식도 제안되었으나[3, 16], 기존에 제안된 많은 QoS 라우팅 방식에서는 현재 통신망에 가용한 자원(available resource) 정보를 바탕으로 응용에서 요구하는 QoS 요구수준을 만족하는 QoS 경로를 계산하고 경로상의 자원을 예약하는 것이 일반적이다[12, 13, 17]. 따라서 QoS 라우팅 정확성은 링크 잔여 대역폭(link residual bandwidth) 등의 라우팅 자원 가용성 정보의 정확성에 밀접한 관련이 있다. 라우팅 자원 가용성 정보를 유지하기 위하여 통신망내의 각 라우터가 자신의 자원 가용성 정보를 계속하여 주기적으로 공시(advertise)하고, 이 정보를 각 라우팅 도메인으로 flood 하였다. 그러나 자원의 사용은 매우 동적으로 변화하고 정보전달의 지연으로 인하여, 자원 가용 상황을 모든 라우터들이 항상 정확하게 유지한다는 것은 거의 불가능하다. 불확실한 자원사용 정보에 의한 QoS 라우팅의 영향을 분석하기 위한 연구도 있었으며[4, 8, 14], 통신망의 규모가 커질수록 이러한 문제점은 더욱 심해질 것이다. 또한 라우팅 지역내에서의 빈번한 가용성 정보교환으로 인하여 라우팅의 확장성 문제를 초래하게 된다. 그러나, QoS 지원으로 도입되는 복잡도가 라우팅 방식의 확장성(scalalble)을 훼손해서는 안된다.

링크상태 전체시계(link-state full view) 방식은 모든 노드들이 전체 통신망에 대한 정보를 가지므로 현재 통신망 정보에 근거하여 전체 통신망 자원을 균등하게 사용하면서 응용의 QoS 요구를 만족하는 QoS 경로를 요구시(on-demand)마다 계산하는 것이 가능하다. 그러나 대규모 통신망에서는 이 방식은 많은 계산과 대규모의 정보 전송을 요구하므로 확장성이 좋지 않다. 따라서 대부분의 대규모 통신망에서는 여러가지 형태의 계층적 도메인간 라우팅 방식을 사용하게 되며, 이는 도메인을 논리적인 노드로, 도메인에서의 경로를 가상 경로(virtual link)로 통합하는데 따른 정보 훼손이라는 단점에도 불구하고 계산량 및 정보 전송량을 줄일 수 있어 확장성에 유리한 장점을 가진다. 이러한 모델의 경우 임의의 노드는 도메인들을 논리적인 노드로 통합한 도메인 수준의 계층적 시계(hierarchical view)를 가진다[1, 2, 10, 21]

계층적 통신망에서는 송신노드(source node)와 수신노드(destination node)가 같은 도메인 내에 있을 경우 적용할 도메인내 라우팅(intra-domain routing)은 물론, 서로 다른 도메인에 있을 경우 적용할 도메인간 라우팅(inter-domain routing)도 지원하여야 한다. 송신노드와 수신노드가 서로 다른 도메인에 있을 경우 송신노드로부터 송신노드 도메인의 나가는 라우터(egress router)까지의 경로, 라우터들 간의 경로, 수신노드 도메인의 들어오는 라우터로부터 수신노드까지의 경로로 나누어 진다. 따라서 임의의 시간에 계층적 통신망의 도메인내에 존재하는 트래픽은 라우터를 경유하지 않는 도메인 내의 트래픽인 내부 트래픽(intra traffic), 송신노드로부터 나가는 라우터까지의 트래픽인 진출 트래픽(outgoing traffic), 들어오는 라우터로부터 수신노드까지의 트래픽인 진입 트래픽(incoming traffic)과 라우터들 간의 트래픽인 횡단 트래픽(transit traffic)이 혼재하게 된다.

계층적 통신망에서 만일 송신노드와 수신노드가 다른 도메인에 존재한다면, QoS 라우팅에서 결정된 QoS 경로는 계층적 완전경로(hierarchically complete path)라는 골격만을 구성하게 되며[21], 경로상의 중간도메인 각각에 대한 상세한 경로는 연결 설립요청이 해당 도메인에 도착하였을때 이루어진다. 계산된 QoS 경로는 전체 통신망에 대한 현재의 정보에 근거하여 임시적으로 결정된 경로이다. 따라서 경로상의 중간 노드에서는 경로에 대한 승락 테스트(admission test)를 하여야 하며, 선택된

QoS 경로상의 노드 및 링크 자원을 예약하는 자원 예약 (resource reservation)이 필요하다. 통신망의 상황이 동적으로 변화하므로 중간노드에서는 승락 요청을 허용하지 못할 수 있으며, 특히 계층적 통신망의 경우 그러할 가능성이 크다. 이러한 경우 PNNI에서의 crankback 라우팅[21]과 같이, 경로의 일부 혹은 전체구간에서 다시 경로설정이 이루어지며 연결설립시간이 지연될 수 있다.

자원 가용정보의 갱신주기가 통신망의 변화에 대하여 상대적으로 크다면 QoS 라우팅의 성능이 급격히 떨어진다. 그러나 갱신주기가 너무 작다면 flood을 위한 통신 및 계산의 부담이 너무 클 것이다 [6]. 본 논문에서 제안하는 확장가능한 2단계 QoS 라우팅 구조에서는 자원 가용정보를 선계산 (precomputation)하여 필요시 조회(retrieval) 하는 방식을 제안하였다. 기존에도 선계산 방식의 QoS 라우팅이 제안되었으나 [18, 19], 본 논문에서는 횡단 트래픽을 위한 QoS 경로의 distance을 계산하여 제공하는 역할을 하며, 경로 자체보다는 경로의 QoS distance 값을 저장한다. 즉, 2단계 라우팅구조 중 Phase I에서는 특정 응용과는 관계없이 모든 수준의 도메인에서 들어오는 경계노드와 나가는 경계노드 쌍간에 도메인을 경유하는 횡단트래픽을 위한 경로의 QoS distance 값을 정기적으로 선계산하여 데이터베이스에 저장한다.

Phase II는 응용에서의 요청이 왔을때 실행되는 단계로 다시 두 step으로 나누어 진행된다. Step 1에서는 계층적 완전경로 계산에 필요한 그래프 구축 알고리즘을 제안하였다. 제안된 알고리즘은 송신노드로부터 수신노드방향의 선조도메인내의 경계노드들이 실행하는 분산 알고리즘으로 계층적 완전경로의 불완전성을 보완하기 위하여 기존에 제안되었던 방식보다 더욱 자세히 계층적 통신망을 반영하는 그래프를 생성한다. 즉 송신노드와 수신노드 선조 도메인들의 경계노드들은 Phase I 과정에서 구축한 횡단 트래픽을 위한 QoS distance 값들을 데이터베이스에서 조회하여 그래프 구축을 위한 정보를 수집한다. 경계노드들에 의한 정보 공시 방식보다 조회방식을 택하는 이유는, 공시로 인한 통신 오버헤드를 줄이고, 보다 새로운 정보를 사용하기 위함이다. 제안하는 조회 방식의 단점은 송신노드부터 수신노드까지의 단대단 지연(end-to-end delay)이 추가로 필요하다는 것이다. 송신노드와 수신노드의 선조 도메인들은 통신망의 주소체계로부터 어렵지 않게 구할 수 있다.

Phase II, Step 2에서는 계층적 통신망의 현재 상태를 표현한 그래프에서 대략적으로 계산된 계층적 완전경로를 확장하여 수준 1의 도메인까지 상세한 경로를 얻기 위한 QoS 라우팅 문제 분할 알고리즘을 제안하였다. QoS 요구사항을 분할하는 연구는 있었으나 [15, 18, 19, 20] 계층적 완전경로로부터 QoS 라우팅 문제를 체계적으로 분할하여 수준1 도메인의 경로로 확장하는 분산 알고리즘에 대한 연구는 찾아볼 수 없었다. 경로상의 각 간선중에서 가상링크에 해당하는 간선은 해당 도메인에서 경로가 되며, 가상링크의 QoS 값은 해당 도메인의 경계노드가 횡단 트래픽을 위하여 선계산한 값이다. 따라서, 전체경로가 전체 QoS 요구를 만족하기 위해서는, 선계산했던 QoS 값을 만족시키는 부경로(sub-path)를 제공하여야 한다. 이제 들어가는 경계노드를 송신노드로, 나가는 경계노드를 수신노드로, 선계산했던 QoS 값을 QoS 요구값으로 하는 지역적인 QoS 라우팅 문제가 된다. 이와같이 하나의 전체 QoS 라우팅 문제는, 결정된 경로상의 각 가상링크별로 별도의 지역적인 QoS 라우팅 문제로 분할(partition)된다. 이러한 QoS 라우팅 문제의 분할은 가장 저수준의 QoS 라우팅 문제, 즉 수준 1 도메인의 횡단 트래픽 문제가 될때까지 반복한다. QoS 문제 분할은 그래프 구축 알고리즘의 진행 방향과는 반대로 수신노드로부터 송신노드 방향으로 진행된다. 이 과정도 단대단 지연이 필요하다. 그러나 이 단계는 연결 설립시의 자원예약 및 승락테스트와 관련지어 실행될 수 있으므로 추가적인 지연은 아니다. QoS 라우팅 문제 분할을 위하여 QoS 요구를 표현하는 QoS 매개변수도 분할하여야 하며 분할 방식은 그 유형에 따라 다르다. 이를 위하여 역구성 법칙을 간단히 정의한다.

본 논문은 다음과 같이 구성되어 있다. 제 II절은 계층적 통신망의 모델과 앞으로 알고리즘을 표현하는데 유용한 그래프 연산 및 QoS 파라미터 연산을 정의한다. 제 III절에서는 선계산 방식의 도메인 횡단 트래픽을 위한 QoS distance 값을 계산하기 위한 알고리즘을 제시하였으며, 제 IV절에서는 멀티미디어 응용의 요구시에 계층적 완전경로를 계산하기 위하여 계층적 통신망의 현재 상태를 반영하는 그래프를 구축하는 알고리즘과 QoS 라우팅 문제 분할 방식의 라우팅 알고리즘을 제시하였으며 알고리즘 적용 예를 보였다. 마지막으로 결론과 앞으로의 연구를 제 V절에서 유도하였다.

II. 계층적 통신망의 모델 및 그래프 연산 정의

본 논문에서 고려중인 계층적 통신망(hierarchical network)의 모델과 알고리즘을 표현하기 위하여 정의된 그래프 연산 및 QoS 파라미터 연산을 정의한다[5, 9].

1. 계층적 통신망의 모델

고려중인 계층적 통신망 (N, A) 는 유한노드 집합 N 과 유한링크 집합 A 로 구성되어 있다. 이 집합들은 도메인(domain)이라는 영역으로 분할되어 있으며, 이러한 도메인이 계층적으로 조직화되어 있다. 도메인들은 복수개가 허용되는 게이트웨이 링크(gateway link)에 의하여 서로 연결되어 있다. 도메인 계층구조는 부모-자식 관계로 구성된다.

Definition II.1 도메인 계층구조(Domain Hierarchy)

수준 (level) $l, l > 1$, 도메인은 그가 포함하는 수준 $l-1$ 도메인의 부모이다. 최고 수준의 도메인은 부모를 가지지 않으며, 최저 수준의 도메인, 즉 수준 1의 도메인은 자식을 가지지 않는다. 최고수준을 L 이라 하자. D_{l_i} 을 어떤 수준 l 도메인이라 하자. 첨자 i 는 수준 l 의 많은 도메인들 중 특별히 하나를 나타내기 위하여 사용한다. 같은 수준의 임의의 어느 두 도메인 D_{l_i} 과 D_{l_j} 에 대하여, " D_{l_i} 과 D_{l_j} 이 형제 (sibling)"일 필요충분조건은 " D_{l_i} 과 D_{l_j} 이 동일한 부모를 가진다" 이다. " D_{l_i} 가 D_{k_j} 의 선조도메인 (ancestor)"일 필요충분조건은 $D_{l_i} = D_{k_j}$ 혹은 D_{l_i} 이 D_{k_j} 의 부모의 선조일 것"이다. 비슷하게, " D_{l_i} 가 D_{k_j} 의 후손도메인 (descendant)"일 필요충분조건은 " $D_{l_i} = D_{k_j}$ 혹은 D_{l_i} 이 D_{k_j} 의 자식의 후손일 것"이다. □

임의의 노드는 자신의 선조 도메인들과 연관되어 질수 있다. 특별히 노드 x 가 수준 l 선조 도메인 D_{l_i} 과 연관되어지는 것을 $\langle D_{l_i}; x \rangle$ 라고 표기한다. 도메인 D_{l_i} 내의 두 노드 y, z 간의 링크를 $\langle D_{l_i}; y, z \rangle$ 라고 표현하며, 도메인의 수준을 강조

하지 않아도 될 경우 간단히 $\langle D; y, z \rangle$ 라고 표현한다. 도메인 D_{l_i} 의 노드 y 와 도메인 D_{l_j} 의 노드 z 간의 게이트웨이 링크를 $\langle D_{l_i}; y, D_{l_j}; z \rangle$ 라고 표현한다. 이와같이 도메인내의 노드중에서 다른 도메인으로의 게이트웨이 링크를 가지는 노드를 경계 노드(border node)라 한다.

유한링크 집합 A 의 임의의 원소인 링크 $\langle x, y \rangle$ 에 연관된 r 값을 c_{xy}^m 라 하고 x 로부터 y 방향으로 지나가기 위한 QoS 값을 나타낸다, 여기서 $1 \leq m \leq r$. 제안되는 라우팅 방식에서는 QoS 정보가 링크에만 연관되고 노드에는 연관되지 않는다고 가정한다. 일반적으로 노드에 연관된 값을 링크에 연관시키는 방법들이 알려져 있다.

(N, A) 의 미리 결정된 구별된 두 노드 s 와 t 가 있다 하자. 송신노드 s 로부터 수신노드 t 로의 경로(path)는 일련의 노드 $p = \{s, \dots, t\}$ 로 표현되며, 어떤 노드도 경로상에 두번 이상 나타나지 않는다. 멀티미디어 응용이 s 로부터 t 까지 요청하는 QoS 요구 수준을 $(\sigma^1, \dots, \sigma^r)$ 라 하자.

Definition II.2 Dominated Relation

(N, A) 의 임의의 두 노드 x 와 y 에 대하여, p_1 과 p_2 를 x 로부터 y 까지의 두 경로라 하자. 다음 두 기호 " $<$ " 과 " \leq "를 두 경로간의 상대적 우선 순위를 표기하기 위하여 사용한다. 즉, " $c^m(p_1) < c^m(p_2)$ "일 필요충분조건은 " p_1 이 p_2 보다 파라미터 m 의 측면에서 좋을것"이다. "경로 p_2 가 경로 p_1 에 의하여 dominated"될 필요충분조건은 " $c^1(p_1) \leq c^1(p_2), \dots, c^r(p_1) \leq c^r(p_2)$ "이고 부등호가 어떤 m 에 대하여 성립한다, 여기서 $1 \leq m \leq r$ 이다. □

2. 그래프 연산 정의

고려중인 통신망은 방향성 그래프 $G = (V, E)$ 로 모델되어진다. 여기서 그래프의 정점(vertex)은 원시노드 s , 목적노드 t , s 와 t 의 선조도메인 및 형제도메인으로의 경계 노드를 나타내고, 간선(edge)은 통신망에서의 그들간의 경로를 나타낸다. 모델링되는 그래프에는 다중 간선을 허용하고, 이 경우 하나의 간선은 해당 도메인을 횡단하는 경로와 연관

된다. 통신망의 노드와 그래프의 정점은 동의어가 아니며, 마찬가지로 링크와 간선도 동의어가 아니다. 그러나 논문 표현을 간략하게 하기 위하여 혼동이 없는 경우 혼용하여 사용하기도 한다.

다음과 같은 표현 및 연산을 제안하는 라우팅 방식에서 사용한다. D 를 통신망에서의 도메인, x 를 노드라 하자. $Domain(x)$ 는 노드 x 가 위치하는 도메인을 나타낸다. $Level(D)$ 는 계층구조에서의 도메인 D 의 수준을 나타낸다. 따라서 $Level(D_l) = l$. $Ancestor(x)$ 는 노드 x 의 수준 l 선조 도메인, $Ancestors(x)$ 는 노드 x 의 선조 도메인들의 집합을 구하는 연산이다. $Siblings(D)$ 와 $ChildDomains(D)$ 은 도메인 D 의 형제 도메인과 자식도메인들의 집합을 구하는 연산이다.

$Links(x)$ 를 노드 x 의 들어오는 링크의 집합이라 한다. 이때 링크는 도메인내의 링크는 물론, 게이트웨이 링크 및 하위 도메인의 경로인 가상링크도 가능하다. $IngressBorderNodes(D)$ 를 도메인 D 의 들어오는 경계노드(ingress border node)의 집합이라 하고, 비슷하게 $EgressBorderNodes(D)$ 를 나가는 경계노드(egress border nodes)의 집합이라 한다. 따라서

$$BorderNodes(D) = IngressBorderNodes(D) \cup EgressBorderNodes(D)$$

이다.

3. QoS 파라미터 및 파라미터 연산의 정의

알고리즘을 표현하기 위하여 QoS 파라미터에 대한 연산이 필요하다. 본 논문에서는 이를 위하여 구성 (composition) 및 역구성(inverse-composition) 법칙을 정의하였다. 구성법칙은 분산 멀티미디어 응용에서 사용되어질 QoS 파라미터의 일반적인 세가지 형태인 additive, multiplicative, 그리고 concave 에 따라 다음과 같이 정의된다.

Definition II.3 구성법칙(Composition Rule)

$d(i, j)$ 를 링크 (i, j) 에 대한 한 QoS 척도 $p = (i, j, k, \dots, l, m)$ 을 노드 i 로부터 노드 m 까지 경로라 하자.

■ $d(p) = d(i, j) + d(j, k) + \dots + d(l, m)$ 이면, 척도 d 는 additive 구성법칙을 따른다.

■ $d(p) = d(i, j) \times d(j, k) \times \dots \times d(l, m)$ 이면, 척도 d 는 multiplicative 구성법칙을 따른다.

■ $d(p) = \min d(i, j), d(j, k), \dots, d(l, m)$ 이면, 척도 d 는 concave 구성법칙을 따른다. □

p 를 위한 QoS 값은 경로상 링크들의 해당 척도에 대한 구성규칙(composition rule)으로 얻어진다. 즉, $c^m(p) = \bigoplus_{(x,y) \in p} c_{xy}^m$, 여기서 ' \bigoplus '는 파라미터 m 의 구성규칙을 나타낸다. 각 파라미터에 대하여, ' \ominus '로 표현되는 역구성 법칙은 해당 구성법칙 \bigoplus 으로부터 다음과 같이 정의된다.

Definition II.4 역구성법칙(Inverse Composition Rule)

$$a \ominus b = c \text{ if and only if } a = b \bigoplus c$$

III. Phase I : 선계산 방식의 도메인 횡단 트래픽을 위한 QoS Distance 값 계산

본 논문에서 제안하는 2-phase 라우팅 구조의 Phase I 에서는 임의의 수준 l 도메인 D_{l_i} 에서 들어오는 경계노드 x 와 나가는 경계노드 y 쌍간에 도메인을 경유하는 횡단트래픽을 위한 경로의 QoS distance 값을 정기적으로 선계산 방식으로 구한다. 즉,

$$\langle D_{l_i}; x \rangle : x \in IngressBorderNodes(D_{l_i}) \text{ 와 } \langle D_{l_i}; y \rangle : y \in EgressBorderNodes(D_{l_i})$$

인 경계노드 쌍 $(\langle D_{l_i}; x \rangle, \langle D_{l_i}; y \rangle)$ 에 대하여, 경로 $\{\langle D_{l_i}; x \rangle, \dots, \langle D_{l_i}; y \rangle\}$ 의 QoS distance 값을 구한다. 일반적으로 도메인을 횡단하기 위한 많은 경로가 있을 수 있으므로, 이 경로와 연관된 r 값을 부여하는 많은 방법이 가능하다. 제안하는 라우팅 방식에서는 각 QoS 파라미터 $m \in \{1, \dots, r\}$ 에 대하여 K 다중 경로를 구한다. 각 경로는 D_{l_i} 의 부모 도메인인 D_{l+1_i} 에서 하나의 가상 링크인 $\langle D_{l+1_i}; x, y \rangle_{xy}^{m,h}$ 가 되며 D_{l+1_i} 를 나타내는 그

래프 상에서는 간선 $e_{xy}^{m,h}$ 로 표현된다, 여기서 $m \in \{1, \dots, r\}$ 그리고 $h = 1, \dots, K$. 본 논문을 간략하게 표현하기 위하여, 경로를 간선으로 주로 표현한다. 이때 경로간에는 다음과 같은 dominated 관계를 만족하도록 한다. 첫째, $c^m(e_{xy}^{m,1}) \leq \dots \leq c^m(e_{xy}^{m,K})$ 이다. 둘째, $e_{xy}^{m,h}$ 는 $e_{xy}^{m,1}, \dots, e_{xy}^{m,h-1}$ 에 의하여 dominated 되지 않는다, 여기서 $h = 2, \dots, K$. 이제 경로에는 QoS distance를 나타내는 r 값 $(c^1(e_{xg}^{m,h}), \dots, c^r(e_{xg}^{m,h}))$ 이 할당된다.

본 논문에서 제안하는 라우팅 방식에서는 각 수준의 도메인 내에서 링크-상태 전체-시계 방식을 사용한다. 즉, 한 도메인내의 모든 노드는 서로 링크 정보를 교환하여 도메인을 나타내는 동일한 위상 데이터베이스를 얻을 수 있다. 따라서 $\langle D_i; x \rangle$ 로 부터 $\langle D_i; y \rangle$ 까지의 경로는 도메인 D_i 에서의 어떤 경계노드, 말하자면 $\langle D_i; \tilde{x} \rangle$ 에서 계산한다. 계산된 경로를 위한 QoS distance값을 데이터베이스에 보관하여 도메인간 라우팅에서 요구시 제공한다. $\langle D_i; \tilde{x} \rangle$ 가 유지해야하는 정보는 자신의 도메인내의 모든 경계노드쌍간의 r 값이므로 실제구현시 x데이터베이스에 유지하기 어려울 정도로 많지는 않다.

수준 l 의 도메인들에 대한 계산이 모두 끝난후, 수준 $l+1$ 의 도메인들에 대한 계산이 수행된다. 수준 $l+1$ 의 도메인, 예를 들어 D_{l+1} 에서의 들어오는 경계노드 x 와 나가는 경계노드 y 쌍간에 경로를 계산하기 위하여 필요한 정보는 다음과 같이 정의된다.

$$\begin{aligned} View(D_{l+1},) = & \\ \{ \langle D, \{ \langle D, x \rangle : x \in BorderNodes(D) \} \rangle, & \\ \{ y : \langle D, x, y \rangle \in Links(\langle D, x \rangle) \} \} & \\ : D \in ChildDomains(D_{l+1},) \} & \end{aligned}$$

그림 1은 Phase I에서의 QoS 라우팅 알고리즘으로, 계층적 통신망에서 각 수준의 도메인별로 수행

되는 분산 도메인내 라우팅 알고리즘이다. 즉, 각 도메인 D_i 별로 동시에 계산이 가능하다, 여기서 $l = 1, \dots, L$. 이 단계의 라우팅은 실제 응용의 요구와는 별도로 background로 실행된다. 알고리즘의 시간 복잡도는 적당한 K 좋은 알고리즘으로 대략적으로 측정이 가능하다. 예를 들어, K -번째 최단 경로에서 $(K+1)$ -번째 최단 경로로의 전이를 위하여 시간 복잡도 $O(M)$ 의 알고리즘이 알려져 있으며, 여기서 M 은 해당 도메인의 링크수가 된다 [7, 22]. 이 경우 제안된 알고리즘의 시간복잡도는 한 도메인내에서 $O(rKPM)$ 이다. 여기서 P 는 한 도메인내에서 계산할 경로의 수가 된다. 즉,

$$P = |IngressBorderNodes(D_{l_i})| \times |EgressBorderNodes(D_{l_i})|.$$

IV. Phase II : 멀티미디어 응용의 요구시 계산하는 QoS 라우팅

멀티미디어 응용으로부터 송신노드에 QoS 요구시, Phase I에서 구해진 QoS distance 값을 조회 방식으로 이용하는 QoS 라우팅이 Phase II에서 수행된다. Phase II도 2 steps으로 이루어져있다. 먼저 Step 1에서 계층적 통신망의 현재 자원사용을 반영하는 그래프를 구축한다. 그래프 구축을 위한 정보는 각 수준의 경계 노드에서 데이터베이스에 저장되어 있으므로, 그래프 구축 알고리즘은 송신노드부터 목적노드로 이동하며 수행하는 분산알고리즘이다. Step 2에서 임의의 경로선택 알고리즘을 사용하여 구축된 그래프상에서 QoS 경로를 계산한다. 이 경로는 가상링크들이 포함된 계층적 완전경로이다. 계층적 완전경로로부터 모두 수준1 도메인내의 링크와 게이트웨이 링크로만 구성된 경로로 바꾸기 위하여, QoS 문제분할 방식의 라우팅 알고리즘을 제안한다.

1. Phase 2, Step 1 : 통신망의 현재 자원사용을 반영하는 그래프 구축

본 논문에서 제안하는 도메인간 라우팅을 위한 그래프를 구축하기 위하여 관련되는 도메인들의 집합을 $VisibleDomains(s, t)$ 라 하자. 먼저 송신노드와 수신노드의 가장 낮은 수준의 공통 선조 도메

```

algorithm QoS_Distance_Procomputation(  $N, A$  )
begin
  for  $l=1$  to  $L$  do
    begin
      for each domain  $D_{l_i}$  do
        begin
          for  $\langle D_{l_i}; \tilde{x} \rangle: x \in \text{IngressBorderNodes}(D_{l_i})$ 
          and  $\langle D_{l_i}; y \rangle: y \in \text{EgressBorderNodes}(D_{l_i})$  do
            for  $m=1$  to  $r$  do
              for  $h=1$  to  $K$  do
                begin
                   $\langle D_{l_i}; \tilde{x} \rangle$  computes the  $h$ -th best
                  path  $e_{xy}^{m,h}$  with respect to parameter
                   $m$  for  $\{\langle D_{l_i}; x \rangle, \dots, \langle D_{l_i}; y \rangle\}$  and
                  its  $r$  values  $(c^1(e_{xy}^{m,h}), \dots, c^r(e_{xy}^{m,h}))$ ,
                  by using an appropriate  $K$  best path
                  algorithm such as  $K$  shortest path
                  algorithm,  $K$  largest capacity algorithm,
                  etc.;
                end
                 $\langle D_{l_i}; \tilde{x} \rangle$  sends results to  $\langle D_{l_{i+1}}; \tilde{x} \rangle$ ;
              end
             $\langle D_{l_{i+1}}; \tilde{x} \rangle$  gathers information of  $\text{View}(D_{l_{i+1}})$ ;
          end
        end
      end
    end
  end

```

그림 1: Phase I - QoS Distance 선계산을 위한 분산 알고리즘

인을 \hat{D}_k 라 하자. 당연히 $\text{Level}(\hat{D}_k) = k$.

$$\hat{D}_k = \{D_{l_i} | D_{l_i} \in \text{Ancestors}(t) \text{ and } D_{l_i} \in \text{Siblings}(D_{l_i}): D_{l_i} \in \text{Ancestors}(s)\}.$$

$$\text{VisibleDomains}(s) = \{D: D \in \text{Siblings}(\hat{D}_l): \hat{D}_l = \text{Ancestor}_l(s) \text{ and } l = 1, \dots, k\}.$$

$$\text{VisibleDomains}(t) = \{D: D \in \text{Siblings}(\hat{D}_l): \hat{D}_l = \text{Ancestor}_l(t) \text{ and } l = 1, \dots, k\}.$$

$$\text{VisibleDomains}(s, t) = \text{VisibleDomains}(s) \cup \text{VisibleDomains}(t).$$

이제, 그래프 구축을 위하여 관련되는 도메인, 정

```

function Construct_Graph( ( $N, A$ ),  $s, t$  )
begin
   $\hat{D}_k = \{D_{l_i} | D_{l_i} \in \text{Ancestors}(t) \text{ and } D_{l_i} \in \text{Siblings}(D_{l_i}): D_{l_i} \in \text{Ancestors}(s, t)\};$ 
   $k = \text{Level}(\hat{D}_k)$ ;
   $\text{View}(s, t) = 0$ ;
  for  $l=1$  to  $k$  do
    begin
       $\hat{D}_l = \text{Ancestor}_l(s)$ ;
       $\text{View}(s) = \{\langle \hat{D}_l, \{\langle \hat{D}_l : x \rangle: x \in \text{BorderNodes}(\hat{D}_l), \{y: \langle \hat{D}_l : x, y \rangle \in \text{Links}(\langle \hat{D}_l : x \rangle)\}\rangle\}$ ;
       $\text{View}(s, t) = \text{View}(s, t) \cup \text{View}_l(s)$ ;
    end
    for  $l=k-1$  downto  $1$  do
      begin
         $\hat{D}_l = \text{Ancestor}_l(t)$ ;
         $\text{View}(t) = \{\langle \hat{D}_l, \{\langle \hat{D}_l : x \rangle: x \in \text{BorderNodes}(\hat{D}_l), \{y: \langle \hat{D}_l : x, y \rangle \in \text{Links}(\langle \hat{D}_l : x \rangle)\}\rangle\}$ ;
         $\text{View}(s, t) = \text{View}(s, t) \cup \text{View}(t)$ ;
      end
    construct graph  $G(V, E)$  from  $\text{View}(s, t)$ ;
    return(  $G(V, E)$  );
  end

```

그림 2: Phase II, Step 1 - 요구시 계산방식의 그래프 구축 알고리즘

점 및 간선으로 이루어진 $\text{View}(s, t)$ 는 다음과 같다.

$$\text{View}(s, t) = \{\langle D, \{\langle D : x \rangle: x \in \text{BorderNodes}(D), \{y: \langle D : x, y \rangle \in \text{Links}(\langle D : x \rangle)\}\rangle\} : D \in \text{VisibleDomains}(s, t)\}$$

그림 2는 계층적 통신망 (N, A) 의 현재 상황을 표현하는 그래프를 구축하기 위한 함수이다. 함수는 분산 알고리즘으로 각 수준별로 수행한 후 그 결과를 누적하며 다음 수준으로 전달한다. 즉, 도메인 $\hat{D}_l \in \text{VisibleDomains}(s)$ 의 선계산을 담당했던 경계노드 $\langle \hat{D}_l; \tilde{x} \rangle$ 는 $\text{View}_l(s)$ 를 자신의 데이터베이스에서 조회하여, 그 결과를 다음 수준의 경계노드인 $\langle \hat{D}_{l+1}; \tilde{x} \rangle$ 에게 누적 전달한다, 여기서

$l=1, \dots, k$. 비슷하게, 도메인 $\hat{D}_l \in VisibleDomains(t)$ 의 경계노드 $\langle \hat{D}_l; \tilde{x} \rangle$ 는 $View_l(t)$ 를 자신의 데이터베이스에서 조회한 후 그 결과를 $\langle \hat{D}_{l-1}; \tilde{x} \rangle$ 에게 누적전달한다, 여기서 $l=k, \dots, 1$. 최종적으로 수신노드 t 는 $View(s, t)$ 로부터 그래프 $G(V, E)$ 를 구축한다. 함수 **Construct_Graph**는 분산알고리즘이나 특성상 순차적으로 진행된다. 각 수준 도메인의 경계노드에서는 데이터베이스를 조회하기 위한 상수시간 $O(1)$ 이 소요되므로, 전체 시간복잡도는 $O(L)$ 로 단대단 지연의 시간이 소요된다. 그림 3에 계층적 통신망의 예와 함수에 의하여 구축된 그래프를 보인다.

2. 구축된 그래프 상에서 QoS 경로 선택 및 연결

본 논문의 전개를 위하여 구축된 그래프 $G(V, E)$ 상의 정점 s 로부터 정점 t 까지 응용에서 요구하는 QoS 수준을 만족하는 QoS 경로를 계산하는 알고리즘이 필요하다. 지면의 제약으로 인하여 본 논문에서는 2단계의 QoS 라우팅 구조를 제안하며, [11]에서 이러한 구조에 적합한 경로선택 알고리즘을 별도로 제안한다. 그러나 QoS 경로 선택을 위하여 기존에 제안되었던 [10]와 같은 QoS 경로선택 알고리즘을 본 논문에서 사용할 수 있으며 이를 함수 **Select_QoS_Path**라 하자.

또한 두개의 QoS 경로 p_1 과 p_2 를 연결(concatenate)하여 연결된 QoS 경로 p_3 를 생성하는 함수 **Concatenate_QoS_Paths**도 필요하다. 이 함수는 또한 p_1 과 p_2 의 r 값을 파라미터 각각의 구성법칙을 적용하여 생성된 새로운 r 값을 계산한다..

3. Phase II, Step 2 : QoS 라우팅 문제 분할 방식의 QoS 라우팅

본 논문에서 제안하는 QoS 라우팅 방식중 Phase II를 그 5에 제시하였다. 먼저 전절에서 언급한 Step 1의 함수 **Construct_Graph**를 수신노드 t 가 호출하여 통신망 (N, A) 에서 s 로부터 t 까지의 그래프 $G(V, E)$ 를 구축한다. 고리즘에서 $G(V, E)$ 는 전역변수로 사용된다. 구해진 그래프

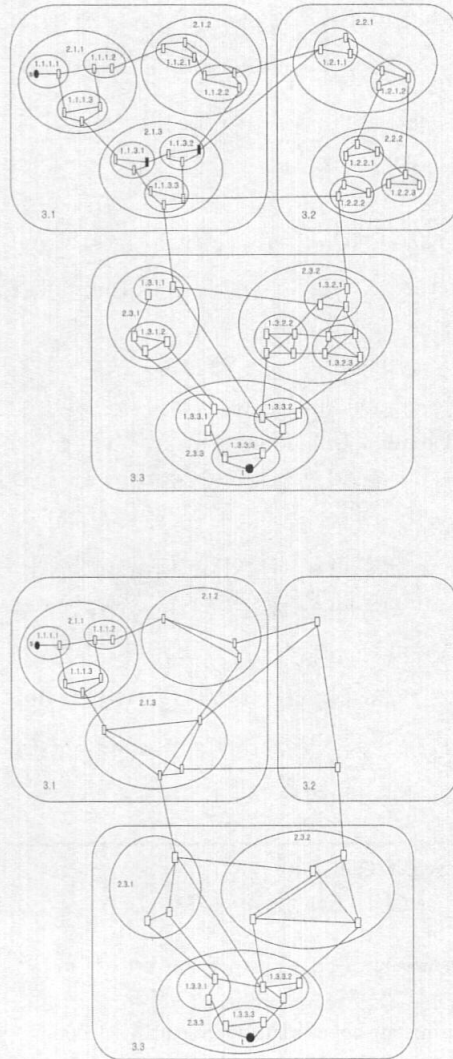


그림 3: 계층적 통신망의 예와 함수로부터 구축된 그래프

상에서 목적노드 t 는 함수 **Select_QoS_Path**를 호출하여 계층적 완전경로 $\hat{p} = \{s, \dots, t\}$ 를 계산한다. 이제 경로 \hat{p} 를 역으로 따라가면서, 경로상의 가상링크를 모두 수준 1 도메인내의 라우팅 문제로 될 때까지 분할한 후 해당 가상링크에 대한 QoS 부경로를 계산하는 방식의 분산 알고리즘이다.

경로 p^* 를 알고리즘이 진행하면서 수준 1 도메인까지 분할하여 발견한 QoS 경로라 하고, 경로의 r 값을 $(\gamma^1, \dots, \gamma^l)$ 하자, 즉, p^* 는 현재 고려중인 노드로부터 t 까지 수준 1 도메인의 링크로 이루어

어진 QoS경로이다. t 의 부모 도메인으로 들어오는 경계노드를 $\langle \hat{D}_1 : x \rangle$ 라 할때, \hat{p} 의 부경로인 $\{ \langle \hat{D}_1 : x \rangle, \dots, t \}$ 는 가상경로가 없으므로 더 이상 분할할 필요가 없다. 따라서 이 부경로를 p^* 의 초기값으로 하고, 부경로의 r 값을 $(\gamma^1, \dots, \gamma^r)$ 의 초기값으로 한다.

이제 \hat{p} 상의 각 가상링크 $\langle D_{l-1} : x, y \rangle$ 를 경로방향의 역방향, 즉 t 에 가까운 가상링크로부터 차례로 고려한다.

이 가상링크와 그의 r 값을 만족하는 부경로를 하위도메인에서 순환적(recursive)으로 찾기 위하여 순환함수 **Partition_QoS_Routing**을 호출한다. 함수의 호출결과는 하위도메인에서 제공하는 부경로 및 그의 r 값이 된다. 이 부경로를 p^* 와 연결하여 새로운 p^* 를 생성함으로써 발견된 경로를 연장한다.

s 의 부모도메인으로부터 나가는 경계노드를 $\langle \hat{D}_1 : y \rangle$ 라 하면, \hat{p} 의 부경로인 $\{ s, \dots, \langle \hat{D}_1 : y \rangle \}$ 도 더 이상 분할할 필요가 없다. 따라서 이 부경로를 p^* 와 연결하면 최종적인 경로인 p^* 를 생성한다. 이제 경로 p^* 는 경로상의 모든

링크가 수준1 도메인내의 링크이고 가상링크가 없는 경로인 것에 주목하라.

Partition_QoS_Routing은 수준 l 의 가상링크인 $\langle \hat{D}_l : x, y \rangle$ 와 링크의 QoS 요구수준인 $(\sigma_l^1, \dots, \sigma_l^r)$ 를 입력 매개변수로 하여, 수준 $l-1$ 도메인 수준에서 QoS 요구수준을 만족하는 QoS 경로 $p_{comp}^* = \{ \langle D_{l-1} : x \rangle, \dots, \langle D_{l-1} : y \rangle \}$ 를 계산한 후, p_{comp}^* 와 그의 r 값 $(\gamma_{comp}^1, \dots, \gamma_{comp}^r)$ 를 반환하는 순환함수이다. 이때 경로 p_{comp}^* 상에 가상링크는 존재하지 않게된다.

먼저 경계 노드 $\langle D_{l-1} : y \rangle$ 는 자신의 데이터베이스에 저장하고 있는 $View_{l-1}(\langle D_{l-1} : y \rangle)$ 로부터 그래프 $G'(V', E')$ 을 구축하며, $G'(V', E')$ 를 추가하여 $G(V, E)$ 를 확장한다. 수준 1의 가상경로, 즉 $l=1$ 인 경우, 이제 간단히 수준 1도메인내의 라우팅 문제가 되므로 더 이상의 QoS 문제 분할이 필요없다. 따라서 **Select_QoS_Path**를 호출하고 그 결과를 반환한다. 편의상 $View_0$ 는 수준 1도메인내의 $View$ 라 하자. $l>1$ 인 경우, 경계 노드 $\langle D_{l-1} : y \rangle$ 는 함수 **Select_QoS_Path**을 호출하여 QoS 요구수준 $(\sigma_l^1, \dots, \sigma_l^r)$ 을 만족하는 수준 $l-1$ 의

경로 $\hat{p}_{l-1} = \{ \langle D_{l-1} : x \rangle, \dots, \langle D_{l-1} : y \rangle \}$ 을 계산한다. 먼저 p_{comp}^* 와 $(\gamma_{comp}^1, \dots, \gamma_{comp}^r)$ 를 각각 빈경로와 zero-벡터로 초기화한다. 이제 \hat{p}_{l-1} 상의 각 가상링크 $\langle D_{l-1} : z, w \rangle$ 를 경로방향의 역방향, 즉 $\langle D_{l-1} : y \rangle$ 에 가까운 가상링크로부터 차례로 고려한다. 이 가상링크와 그의 r 값을 만족하는 부경로를 하위도메인에서 순환적으로 찾기위하여, $\langle D_{l-1} : w \rangle$ 는 **Partition_QoS_Routing**를 다시 순환적으로 호출한다. 함수의 호출결과는 하위도메인에서 제공하는 부경로 및 그의 r 값이 된다. 이 부경로를 p_{comp}^* 와 연결하여 새로운 p_{comp}^* 를 생성함으로써 발견된 경로를 연장한다. 모든 가상경로상에 대한 처리가 끝나면 $p_{comp}^* = \{ \langle D_{l-1} : x \rangle, \dots, \langle D_{l-1} : y \rangle \}$ 가 구해진다.

function Select_QoS_Path

$(G(V, E), s, t, (\sigma^1, \dots, \sigma^r))$

begin

select a feasible QoS path p from s to t to meet the QoS requirements $(\sigma^1, \dots, \sigma^r)$ by using an appropriate QoS path selection algorithm;

let r values of p be $(\gamma^1, \dots, \gamma^r)$;

return $(p, (\gamma^1, \dots, \gamma^r))$;

end

function Concatenate_QoS_Paths $(p_1, (\gamma_1^1, \dots, \gamma_1^r),$

$p_2, (\gamma_2^1, \dots, \gamma_2^r))$

begin

let p_3 be the concatenated path of p_1 and p_2 ;

$(\gamma_3^1, \dots, \gamma_3^r) = (\gamma_1^1, \dots, \gamma_1^r) \oplus (\gamma_2^1, \dots, \gamma_2^r)$;

return $(p_3, (\gamma_3^1, \dots, \gamma_3^r))$;

end

그림 4: QoS 경로선택 및 연결 알고리즘

```

algorithm QoS_Routing
    ((N, A), s, t, ( $\sigma^1, \dots, \sigma^r$ ))
begin
    G(V, E) = Construct_Graph((N, A), s, t);
    ( $\hat{p}, (\gamma^1_{tmp}, \dots, \gamma^r_{tmp})$ ) =
    Select_QoS_Path(G(V, E), s, t, ( $\sigma^1, \dots, \sigma^r$ ));
     $\hat{D}_1 = Ancestor_1(t)$ ;
    let  $\langle \hat{D}_1 : x \rangle$  be the ingress border node on  $\hat{p}$ ;
     $p^* = \{ \langle \hat{D}_1 : x \rangle, \dots, t \}$  and
        let its r-value be ( $\gamma^1, \dots, \gamma^r$ );
    for each virtual-link  $\langle D_l : x, y \rangle$  on  $\hat{p}$  in reverse
        order do
    begin
        let virtual-link  $\langle D_l : x, y \rangle$  be the edge  $e_{xy} \in E$  of
            G(V, E) and its r-value ( $c^1(e_{xy}), \dots, c^r(e_{xy})$ );
        ( $p_{tmp}, (\gamma^1_{tmp}, \dots, \gamma^r_{tmp})$ ) = Partition_QoS_Routing
            ( $\langle D_l : x, y \rangle, (c^1(e_{xy}), \dots, c^r(e_{xy}))$ );
        ( $p^*, (\gamma^1, \dots, \gamma^r)$ ) = Concatenate_QoS_Paths
            ( $p_{tmp}, (\gamma^1_{tmp}, \dots, \gamma^r_{tmp}), p^*, (\gamma^1, \dots, \gamma^r)$ );
    end
     $\hat{D}_1 = Ancestor_1(s)$ ;
    let  $\langle \hat{D}_1 : y \rangle$  be the egress border node on  $\hat{p}$ ;
     $p_{tmp} = \{s, \dots, \langle \hat{D}_1 : y \rangle\}$  and
        let its r-value be ( $\gamma^1_{tmp}, \dots, \gamma^r_{tmp}$ );
    ( $p^*, (\gamma^1, \dots, \gamma^r)$ ) = Concatenate_QoS_Paths
        ( $p_{tmp}, (\gamma^1_{tmp}, \dots, \gamma^r_{tmp}), p^*, (\gamma^1, \dots, \gamma^r)$ );
     $p^*$  is a found QoS path and
        ( $\gamma^1, \dots, \gamma^r$ ) its r values;
end

function Partition_QoS_Routing
    ( $\langle \hat{D}_l : x, y \rangle, (\sigma^1_l, \dots, \sigma^r_l)$ )
begin
     $l = Level(\hat{D}_l)$ ;
    construct graph  $G'(V', E')$  from
        View_{l-1}( $\langle D_{l-1} : y \rangle$ );
    expand graph G(V, E) such that
         $V = V \cup V'$  and  $E = E \cup E'$ ;
    if ( $l = 1$ )
        return Select_QoS_Path
            ( $G'(V', E'), \langle \hat{D}_0 : x \rangle, \langle \hat{D}_0 : y \rangle, (\sigma^1_l, \dots, \sigma^r_l)$ );
    ( $\hat{p}_{l-1}, (\gamma^1_{tmp}, \dots, \gamma^r_{tmp})$ ) = Select_QoS_Path
        ( $G'(V', E'), \langle D_{l-1} : x \rangle, \langle D_{l-1} : y \rangle, (\sigma^1_l, \dots, \sigma^r_l)$ );

```

```

 $p^*_{comp} = \{ \}$ ;
    ( $\gamma^1_{comp}, \dots, \gamma^r_{comp}$ ) = (0,  $\dots$ , 0);
    for each virtual-link  $\langle D_{l-1} : z, w \rangle$  on  $\hat{p}_{l-1}$ 
        in reverse order do
    begin
        let virtual-link  $\langle D_{l-1} : z, w \rangle$  be
            the edge  $e_{zw} \in E$  of G(V, E) and
            its r-value ( $c^1(e_{zw}), \dots, c^r(e_{zw})$ );
        ( $p_{tmp}, (\gamma^1_{tmp}, \dots, \gamma^r_{tmp})$ ) =
        Partition_QoS_Routing
            ( $\langle D_{l-1} : z, w \rangle, (c^1(e_{zw}), \dots, c^r(e_{zw}))$ );
        ( $p^*_{comp}, (\gamma^1_{comp}, \dots, \gamma^r_{comp})$ ) =
        Concatenate_QoS_Paths
            ( $p_{tmp}, (\gamma^1_{tmp}, \dots, \gamma^r_{tmp}), p^*_{comp}, (\gamma^1_{comp}, \dots, \gamma^r_{comp})$ );
    end
    return ( $p^*_{comp}, (\gamma^1_{comp}, \dots, \gamma^r_{comp})$ );
end

```

그림 5: phase II, Step 2-QoS 라우팅 문제 분할방식의 QoS 라우팅 알고리즘

이제 경로 p^*_{comp} 는 경로상의 모든 링크수가 수준1 도메인내의 링크이고 가상링크가 없는 경로인 것에 주목하라.

함수 **Partition_QoS_Routing**는 알고리즘 **QoS_Distance_Precomputation**의 정기적인 선계산방식을 보완하는 부수적인 기능을 수행하도록 할수 있다. 즉, **Partition_QoS_Routing** 수행 결과 얻어진 경로값은 도메인을 횡단하는 트레픽을 위한 QoS distance값으로 사용할 수 있으므로 필요시 해당 데이터베이스를 갱신하여 보다 새로운 값으로 QoS distance 값을 유지하도록 할 수 있다.

알고리즘 **Partition_QoS_Routing**도 분산알고리즘으로 QoS문제 분할을 위하여 필요한 QoS라우팅 정보를 데이터베이스에 저장하고 있는 수준의 경계 노드에서 수행하게 된다. 따라서 t 로부터 s 까지의 단대단 지연이 추가로 필요로 한다. 그러나 최근의 초고속통신망에서 이러한 지연은 그다지 크지 않으며, 본 논문에서 제안한 라우팅 과정과 계산된 경로에 대한 자원예약 과정과 계산된 경로에 대한 자원예약 과정과 연계하여 수행한다면 라우팅을 위하여 추가로 필요한 지연시간은 그다지 크지 않다고 하겠다.

4. QoS 라우팅 문제 분할 예

그림 3에서 계산된 QoS 경로와 이에 대하여 수행된 QoS 라우팅 문제 분할의 예를 그림 6과 그림 7에 제시하였다. 편의상 수준 1 도메인내의 경로는 생략하였다. 그림 6(a)에서는 최초 구축된 그래프 $G(V, E)$ 에서 s 부터 t 까지 계층적 완전경로 \hat{p} 를 굵은 실선으로 표시하였다. 그림 6(b)에서는 가상링크 $\langle D_1; x, y \rangle$ 에 대한 함수 **Partition_QoS_Routing** 실행과정을 보여준다. 가상링크가 수준1이므로 더 이상의 QoS 문제분할은 이루어지지 않는다. 함수를 실행하는 나가는 경계노드 $\langle D_1; y \rangle$ 는 검은색으로 표시하였다. 그림 6(C)에서는 가상링크 $\langle D_2; x, y \rangle$ 에 대한 **Partition_QoS_Routing** 실행을 보여주며, 그 결과 $View_1(\langle D_1; y \rangle)$ 로부터 확장된 그래프를 그림 6(d)에서 보여준다. 이제 그래프 $G(V', E')$ 에서 경로 \hat{p}_1 을 계산하며, 이 경로에는 수준1 가상링크를 두개 가지므로, 결국 두개의 수준 1도메인을 횡단하는 QoS 라우팅 문제로 분할된다. 이제 각 문제에 대하여 **Partition_QoS_Routing**을 수행한다. 그림 6(e)로부터 그림 6(g)는 가상링크 $\langle D_3; x, y \rangle$ 에 대한 연속적인 문제 분할의 결과를 보여준다. 같은 요령으로 계속 QoS 문제를 분할하며 경로를 계산하여 그림 6(k)와 같은 최종적인 경로 p^* 를 계산한다. 이제 경로 p^* 는 가상링크 없이 수준1 도메인내의 링크만으로 구성된 것에 주목하라.

V. 결 론

연결형의 고속 통신망에서 QoS 요구사항을 만족하는 QoS 경로를 계산하는 문제는 다중척도 최단거리 문제로 NP-complete라고 알려져 있다. 기존에 제안되었던 대부분의 QoS 라우팅 방식은 현재 통신망의 자원 가용성 정보를 주기적으로 공시하기 위하여 flood 하였다. 라우팅 기능을 대규모 통신망에 적합하도록 하기 위한 가장 일반적인 방식은 노드 및 링크를 계층구조로 구성하는 것이다. 그러나 이러한 통합과정에서 노드 및 링크에 대한 자세한 QoS 정보가 손실된다. 또한 QoS 라우팅에서 결정된 경로는 계층적 완전경로이고 경로상의 중간 도메인 각각에 대한 상세한 경로는 후에 결정되며, 필요시 부분적인 경로수정 및 재 라우팅이 이루어지

기도 한다.

본 논문에서 제안하는 확장가능한 2단계 QoS 라우팅 구조에서는 자원 가용정보를 선계산하여 필요시 조회하는 방식을 제안하였다. 이를 위하여 Phase 1에서 모든 수준의 도메인에서들어오는 경계노드와 나가는 경계노드 쌍간에 도메인을 경유하는 횡단트래픽을 위한 경로의 QoS distance 값을 정기적으로 선계산하여 데이터베이스에 저장한다. 이 계산은 특정 응용과는 관계없으며 경로 자체보다는 경로의 QoS distance 값만을 저장한다.

응용에서의 요구가 왔을때 Phase II가 다시 두 step으로 나누어 실행된다. Step 1에서는 계층적 완전경로 계산에 필요한 그래프 구축 분산 알고리즘으로 송신노드로부터 수신노드방향의 선조도메인내의 경계노드들에서 실행된다. 이와같이 자원가용성 정보를 공시하지 않고 조회방식을 사용함으로써 정보의 flood로 생성되는 많은 통신 오버헤드를 줄일 수 있다. 또한 본 논문에서는 계층적 완전경로의 불완전성을 보완하기 위하여 기존에 제안되었던 방식보다 더욱 자세히 계층적 통신망을 반영하는 그래프를 생성한다. 제안하는 조회 방식의 단점은 송신노드로부터 수신노드까지의 단대단 지연(end-to-end delay)이 추가로 필요하나, 최근의 초고속통신망에서 이러한 지연은 그다지 크지 않다.

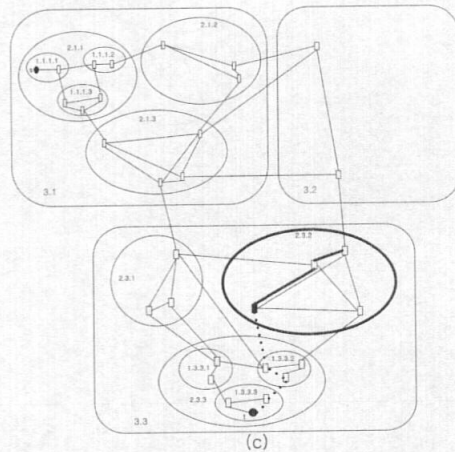
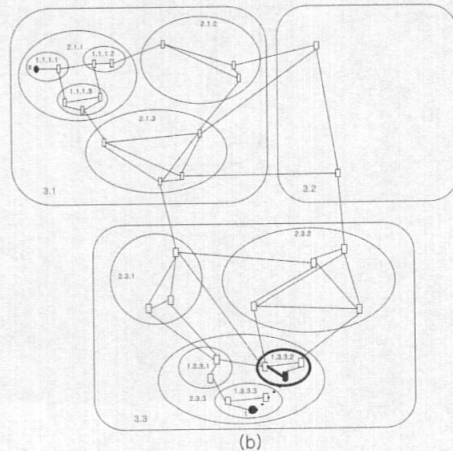
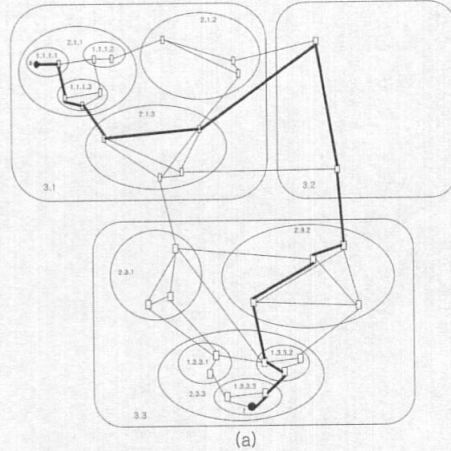
Step 2에서는 대략적으로 계산된 계층적 완전경로를 확장하여 수준 1의 도메인까지 상세한 경로를 얻기 위한 QoS 문제 분할 방식을 제안하였다. 즉, 계층적 완전경로로부터 QoS 문제를 체계적으로 분할하여 수준1 도메인의 경로로 확장하는 분산 알고리즘이 실행된다. QoS 문제 분할은 그래프 구축 알고리즘의 진행 방향과는 반대로 수신노드로부터 송신노드 방향으로 진행하며 단대단 지연이 필요하다. 그러나 이 단계는 연결 설립시의 자원예약 및 승락 테스트와 관련지어 실행될 수 있으므로 추가적인 지연은 아니다.

본 논문에서 제안하는 라우팅 방식에서 사용된 QoS 경로 계산 알고리즘은 기존에 제안되었던 임의의 알고리즘도 사용 가능하나, 제안된 QoS 라우팅 문제 분할에 적합한 알고리즘 개발이 필요하다. 이 알고리즘은 현재 개발되어 모의실험을 진행하고 있으며, 지면의 제약으로 인하여 본 논문에서는 2단계의 QoS 라우팅 구조를 제안하며, 2단계 라우팅 구조에 적합한 경로선택 알고리즘을 별도로 제안한다. 또한 Phase II, Step 2에서 실행되는 요구시 라우팅과 Phase I의 선계산 라우팅을 연관시키는 연

구도 필요하다. 즉, 요구시 실행되는 도메인내 라우팅과 선계산으로 실행되는 도메인내 라우팅이 서로 연관되어 횡단 트래픽을 다른 트래픽에 우선하도록 하는 정책이 필요하며 현재 연구가 진행중이다. 또한 도메인내 자원 사용을 균등하게 사용하여 횡단 트래픽을 위한 QoS distance 값을 만족시키는 경로를 발견시킬 확률을 높이기 위한 방안에 대한 연구도 필요하다.

참 고 문 헌

- [1] C. Alaettinoglu and A.U. Shankar, "The Viewserver Hierarchy for Interdomain Routing: Protocols and Evaluation" *IEEE JSAC*, 13(8), pp.1396-1410, October 1995.
- [2] Baruch Awerbuch and Yuval Shavitt, "Topology Aggregation for Directed Graphs," *IEEE/ACM Transactions on Networking*, 9(1), pp.82-90, Feb 2001.
- [3] Shigang Chen, Klara Nahrstedt, "An Overview of Quality-of-Service Routing for the Next Generation HighSpeed Networks: Problems and Solutions," *IEEE Network*, pp. 64-79, November/December 1998.
- [4] Israel Cidon, Raphael Rom, and Yuval Shavitt, "Bandwidth Reservation for Bursty Traffic in the Presence of Resource Availability Uncertainty," *Computer Communications*, 22(10), pp.919-929, June 1999.
- [5] J.C.N. Climaco and E.Q.V. Martins, "On the Determination of the Nondominated Paths in a Multiobjective Network Problem," *Methods in Operations Research*, 40, pp.255-258, 1981.



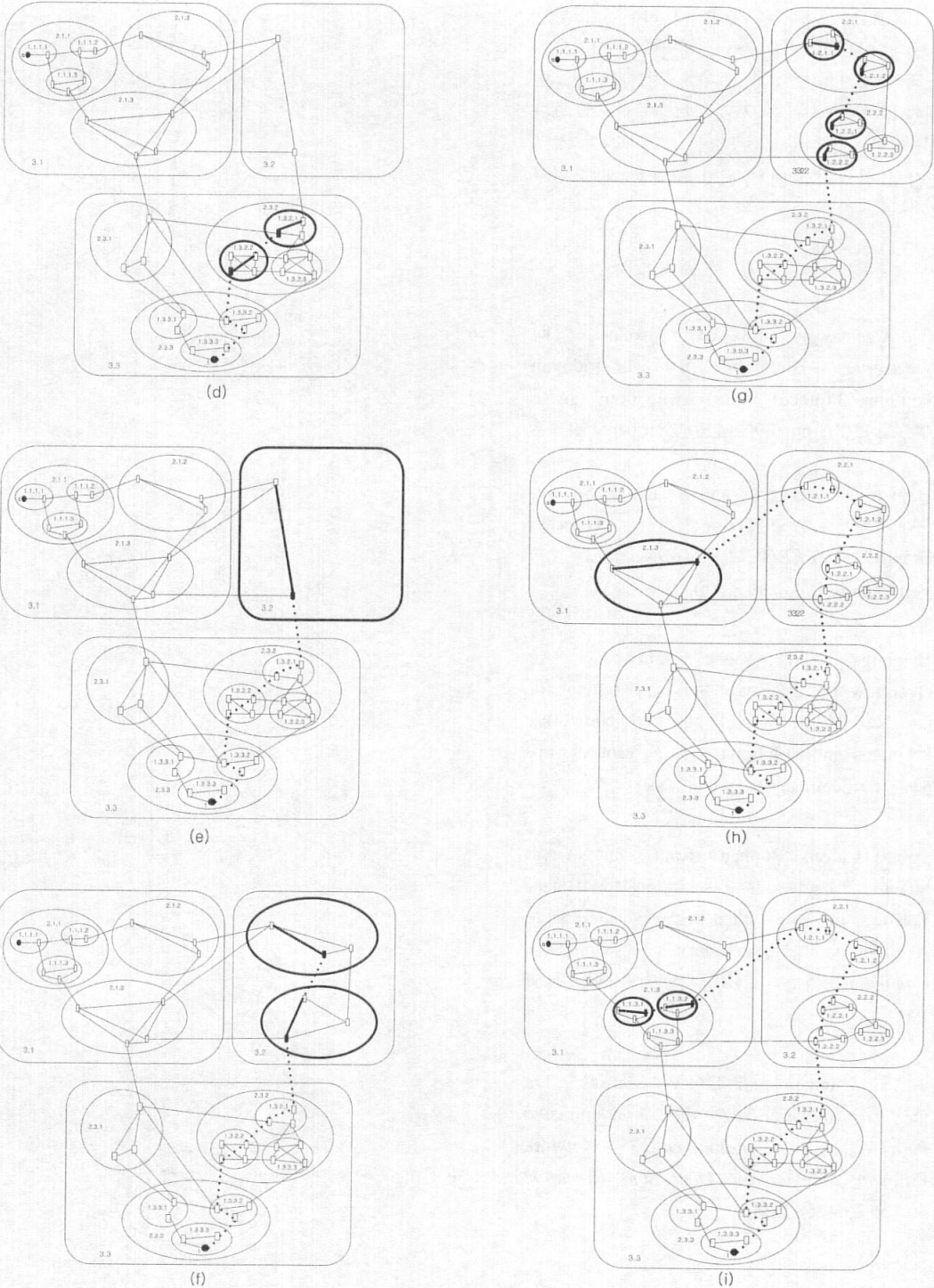
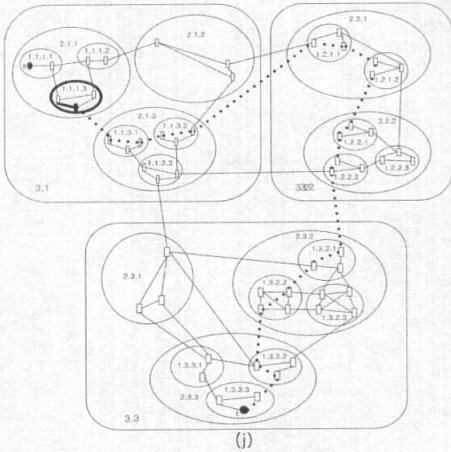
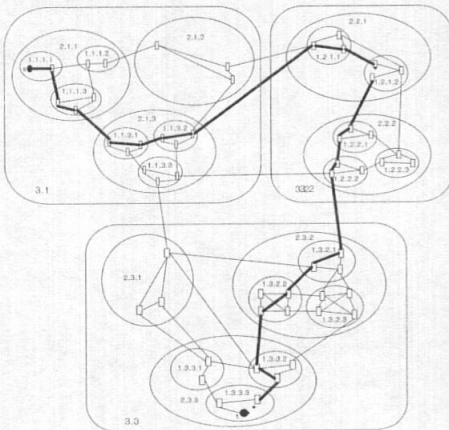


그림 6: 계산된 전체 QoS 경로 및 QoS 라우팅 문제 분할의 예



(j)



(k)

그림 7: 계산된 전체 QoS 경로 및 QoS 라우팅 문제 분할의 예(계속)

[6] Fang Hao, Ellen W. Zegura, "On Scalable QoS Routing: Performance Evaluation of Topology Aggregation," *IEEE Infocom* 2000, pp.147-156, 2000.

[7] M. Gondran and M. Minoux, *Graphs and Algorithms, Wiley-International Series*, 1984.

[8] R. Guerin and A. Orda, "QoS Routing in Networks with Inaccurate Informations: Theory and Algorithms," *IEEE/ACM Transactions on Networking*, 7(3),

[9] M.I. Henig, "The Shortest Path Problem with Two Objective Functions," *European Journal of Operational Research*, 25, pp.281-291, 1985.

[10] S. Kim, K. Lim, and C. Kim, "A Scalable QoS-based Inter-Domain Routing Scheme in a High Speed Wide Area Network," *Computer Communications*, 21(4), pp.390-399, 1998.

[11] 김승훈, "QoS 라우팅 문제분할에 적합한 경로선택 알고리즘," *기술문서*.

[12] 김승훈, 김치하, "분산 멀티미디어 응용을 위한 대규모 고속 통신망에서의 QoS-근거 계층적 도메인간 라우팅 방식," *한국통신학회논문지*, 24(7B), pp.1239--1251, July 1999.

[13] 홍종준, 김승훈, 이균하 "QoS 경로의 효율적인 선계산을 위한 부하균등 비용산정 방식," *한국통신학회논문지*, 26(11C), pp.140-147, November 2001.

[14] Dean H. Lorenz, Ariel Orda, "QoS Routing in Networks with Uncertain Parameters", *IEEE/ACM Transactions on Networking*, 6(6), pp.768-778, December 1998.

[15] Dean H. Lorenz, Ariel Orda, "Optimal Partition of QoS Requirements on Unicast Paths and Multicast Trees," *IEEE/ACM Transactions on Networking*, 10(1), pp.102-114, 2002.

[16] Srihari Nelakuditi, Zhi-Li Zhang, and Rose P. Tsang, "Adaptive Proportional Routing: A Localized QoS Routing Approach," *IEEE Infocom 2000*, pp.1566-1575, 2000.

[17] Ariel Orda, "Routing with End-to-End QoS Guarantees in Broadband Networks", *IEEE/ACM Transactions on Networking*,

7(3), pp.365-374, June 1999.

- [18] Ariel Orda and Alexander Sprintson, "QoS Routing: The Precomputation Perspective," *IEEE Infocom 2000*, pp.128-136, 2002.
- [19] Ariel Orda and Alexander Sprintson, "A Scalable Approach to the Partition of QoS Requirements in Unicast and Multicast," *IEEE Infocom 2002*, pp.685-694, 2002.
- [20] Danny Raz and Yuval Shavitt, "Optimal Partition of QoS Requirements with Discrete Cost Functions," *IEEE JSAC 2000*, 18(12), pp.2593--2602, December 2000.
- [21] Private Network-Network Interface Specification Version 1.0 (PNNI 1.0), *ATM Forum PNNI Subworking Group*, af-pnni-0055.000, March 1996.
- [22] C.C. Skiscim and B.L. Golden, "Solving k -Shortest and Constrained Shortest Path Problems Efficiently," *Annals of Operations Research*, 20, pp.249-282, 1989.

김 승 훈(Seung-Hoon Kim)

정회원



1985년 2월 : 인하대학교 전자
계산학과 졸업
1989년 8월 : 인하대학교 전자
계산학과 석사
1998년 2월: 포항공과대학교
전자계산학과 박사
2001년 9월 ~현재 : 단국대학교
전자컴퓨터학
부 조교수

<주관심분야> 멀티미디어통신망, 초고속통신망