

QoS 라우팅을 위한 효율적인 동적 네트워크 상태 정보 갱신 방안

(An Efficient Dynamic Network Status Update Mechanism for QoS Routing)

김 지 혜 [†] 이 미 정 ^{**}
(Jeehye Kim) (Meejeong Lee)

요 약 QoS 라우팅은 멀티미디어 응용 프로그램들이 요구하는 QoS 요구사항을 기반으로 이를 만족시킬 수 있는 적정경로를 찾아내는 라우팅 기법이다. QoS 라우팅은 네트워크의 동적인 상태를 반영하여 경로를 결정하기 때문에 응용의 요구를 만족시킬 뿐 아니라 네트워크의 활용율을 증대시키는 효과가 있다. 그러나, 네트워크의 동적인 상태 정보를 파악하기 위해서는 라우터들이 이러한 정보를 서로 교환하는 오버헤드가 발생하게 된다. 이러한 프로토콜의 오버헤드를 줄이기 위해 주기적으로 네트워크 상태 변화를 점검하여 변화의 정도가 어떤 값 이상일 때만 네트워크 상태 정보를 주고 받는 방법이 제안되었다. 그러나 이렇게 주기를 사용하게 되면 주기가 큰 경우에는 라우터의 네트워크 상태 갱신 필요를 판단하는 파라미터의 민감성이 네트워크 상태 갱신에 거의 반영되지 않는다는 문제가 있다. 또한, 부정확한 네트워크 상태 정보에 근거하여 경로를 결정하는 경우 QoS 라우팅에 의해 적정 경로로 선택된 경로에 실제로는 요구된 정도의 자원이 없어서 자원 예약에 실패하게 될 수 있는데, 이렇게 되면 이미 자원 예약이 이루어진 경로의 일부에서 자원 예약을 해제해야 하는 등의 오버헤드가 커지게 된다. 본 논문에서는 이 두 가지 문제에 대한 기존의 네트워크 상태 갱신 정책 개선안을 제시하고 시뮬레이션에 의해 그 성능을 평가하였다.

키워드 : QoS 라우팅, 동적 네트워크 상태 정보, 라우팅 프로토콜, 서비스 품질

Abstract QoS routing is a routing technique for finding feasible path that satisfies QoS requirements required by application programs. Since QoS routing determines such paths in terms of dynamic network state, it satisfies the requirement of applications and increases the utilization of the network. The overhead is, however, generated by routers to exchange the information of the dynamic state of network. In order to reduce this protocol overhead, a timer based update mechanism is proposed in which router checks the change of the network status periodically and network state information is exchanged if the change is greater than a certain value. Using large update period makes, though, routing performance irresponsive to the parameters which determine the update of the network state of the router. In addition to this, large update period may result in inaccurate network state information at routers and cause resource reservation failure. The resource reservation failure generates additional overhead to cancel the resource reservation of the part of the path. In this paper, we propose mechanisms enhancing the existing network state update policy with respect to these two problems. Performance of the proposed schemes are evaluated through a course of simulation.

Key words : QoS routing, dynamic network state update, routing protocols, QoS

· 본 연구는 한국과학재단 목적기초연구(R04-2000-00078) 지원으로 수행되었음.

[†] 비 회 원 : 필승전자연구소 연구원
poohkim@korea.com

^{**} 정 회 원 : 이화여자대학교 컴퓨터학과 교수
lmj@cwaha.ac.kr

논문접수 : 2001년 2월 2일

심사완료 : 2001년 8월 17일

1. 서 론

오늘날의 인터넷은 최선 서비스(best-effort) 라우팅을 기반으로 한다. 이 라우팅 기법은 홉(hop) 수나 실제 거리, 최대 대역폭(bandwidth)과 같은 정적 메트릭에 의한 최단 경로 라우팅을 이용하여 QoS의 보장 없이 단순한 연결성만을 제공한다. 최선 서비스 라우팅 기법

은 전통적인 데이터 트래픽 전송에 적합한 것으로 사용되어 왔으나, 새롭게 등장하고 있는 멀티미디어 응용 프로그램들이 요구하는 성능 보장 요구를 만족시키기에는 한계점이 있다[1, 2]. 따라서, 멀티미디어 응용프로그램이 요구하는 QoS 요구사항을 만족시키기 위하여 네트워크의 동적 변화에 적극적으로 대처하는 라우팅 기법으로 QoS 라우팅이 제안되었다[3]. QoS 라우팅이란, 대역폭이나 지연(delay)같은 플로우의 QoS 요구사항에 기반하여, 플로우를 전송하는데 사용되는 적정 경로(feasible path)를 선택하는 작업을 의미한다[4, 5, 6, 7]. QoS 라우팅을 사용하는 목적은, 사용자의 QoS 요구사항을 만족시키는 동시에 네트워크 활용도를 향상시키는 것이다[8, 9].

현재까지 여러 연구에 의해 이러한 QoS 라우팅의 잠재적인 이점이 제시되어 왔으나, QoS 라우팅 구현의 오버헤드 및 복잡성으로 인해 실제적인 도입에 이르지 못하고 있다. QoS 라우팅에 수반되는 추가 비용에는 두 가지의 주요 구성 요소가 있다.

- 계산 비용(computational cost)
- 프로토콜 오버헤드(protocol overhead)

전자는 좀더 복잡하고 잦은 경로 선택 계산 때문이고, 후자는 동적으로 변하는 네트워크 자원 상태 — 예를 들어, 링크의 가용 대역폭 — 를 알리기 위해 발생하는 것이다. 특히, 네트워크 상태 정보 교환을 위한 프로토콜 오버헤드는 대역폭, 저장 공간(storage), 프로세싱 등과 같은 다중 요소들의 비용 증가를 일으키는 원인이 된다[2, 10]. 네트워크 상태 변화를 자주 점검하고 갱신하게 되면 프로토콜 오버헤드는 증가하지만, 네트워크 상태 정보의 정확성과 라우팅 성능은 높아지게 된다. 반면에 이러한 오버헤드를 줄이기 위해 네트워크 상태 변화를 둔감하게 알리는 경우에는 라우팅 성능이 저하된다. 각 라우터가 지역적인 네트워크 상태 변화에 대한 정보를 언제 네트워크 상의 다른 라우터들에게 알릴 것인지를 결정짓는 네트워크 상태 갱신 유발 정책(이후로 갱신 유발 정책으로 표기)은 이러한 라우팅 오버헤드와 라우팅 성능을 효과적으로 조절하는 것을 목표로 한다[11].

본 논문에서는, 가용 대역폭을 매트릭으로 하는 경우에 적용하는 기존의 갱신 유발 정책에서 프로토콜 오버헤드를 줄이기 위해 갱신 주기를 사용할 경우 갱신 유발 정책 자체의 민감성과는 거의 상관없이 갱신 주기에 의해 라우팅 오버헤드와 성능이 좌우되는 것을 개선하기 위해 새로운 갱신 유발 정책을 제안하였다. 또한, 라우팅 후에 진행되는 자원 예약 과정에서의 실패를 줄이기 위한 보완책을 갱신 유발 정책에 반영하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 연구에서 제시되었던 갱신 유발 정책에 대하여 설명하고, 3장에서는 기존 정책의 문제점을 지적하고 이를 개선한 새로운 갱신 유발 정책들의 세부 내용에 대해 기술한다. 4장에서는 성능 평가를 위한 시뮬레이션 설계 및 그 결과를 분석하고, 5장에서 결론을 맺고자 한다.

2. 기존의 갱신 유발 정책

QoS 라우팅을 위한 갱신 유발 정책이란 네트워크의 동적인 상태에 대한 새로운 정보를, 네트워크를 구성하고 있는 모든 라우터들에게 언제 전달해야 하는가에 대한 규약이다. 갱신 유발 메커니즘은 상대적인 변화에 의한 갱신 유발, 절대적인 변화에 의한 갱신 유발, 그리고 갱신 주기에 의한 갱신 유발 등의 세 가지로 분류할 수 있다. 상대적인 변화에 의한 갱신 유발이란, 현재 링크 상태와 이전에 공시된 링크 상태 사이의 상대적인 차이가 임의의 임계치를 넘어설 때 갱신이 일어나는 것을 말하며, 절대적인 변화에 의한 갱신 유발이란, 링크 상태값이 몇 개의 클래스로 나뉘어져서, 현재 링크 상태값이 클래스 경계를 넘어서면 갱신이 일어나는 것을 말한다. 이러한 메커니즘은 클래스 경계의 간격이 고정된 크기인가 혹은 지수 분포 크기인가에 따라 추가적인 분류가 가능하다. 이러한 임계치나 클래스 기반 갱신 유발 정책을 사용할 때, 라우팅 성능을 향상시키기 위해서는 갱신 임계치의 값이나 클래스 크기를 낮게 책정하여 네트워크 상태를 민감하게 알려야 하는데, 이렇게 되면 라우터가 발생시키는 갱신 메시지의 수가 증가하여 프로토콜 오버헤드의 증가를 초래하게 된다. 또한, 임계치나 클래스 변화만을 기반으로 네트워크 상태 정보를 갱신하면, 갱신 오버헤드의 양을 예측할 수 없다는 문제점이 있다. 이러한 단점을 보완하기 위해 제시된 것이 갱신 주기(hold-down timer)의 사용이다[12]. 갱신 주기에 의한 갱신 유발 정책은 타이머를 사용하여 고정된 시간 간격마다 네트워크 상태를 갱신하거나 연속적인 두 갱신 사이에 최소 간격이 유지되도록 한다. 갱신 주기에 의한 갱신 유발 정책은 갱신 메시지 오버헤드를 억제하기 위해 임계치나 클래스 기반의 갱신 유발 정책과 함께 사용되기도 한다.

임계치나 클래스 기반의 갱신 유발 정책에서 갱신 주기를 적용하지 않을 때에는 세션 수용이나 세션 반환 시마다 그 세션이 차지하고 있는 경로에 포함된 링크의 가용 대역폭의 변화량을 체크하여 이 변화량이 갱신 유발 조건을 만족시키면 네트워크 정보를 갱신한다. 이와는 달리, 갱신 주기를 적용할 때에는, 각 라우터마다 갱

신 주기를 체크하는 타이머가 있어서 각 라우터에서는 갱신 주기가 만료되면, 자신의 인접 노드로의 가용 링크 대역폭의 변화량을 체크하여 이 변화량이 갱신 유발 조건을 만족시키면 네트워크 정보를 갱신한다.

3. 제안하는 갱신 유발 정책

본 장에서는 2장에서 살펴 본 기존 갱신 유발 정책의 문제점을 지적하고, 보다 효율적인 갱신 유발 정책을 제시하고자 한다. 보다 정확한 네트워크 정보를 이용하기 위해서는 갱신 임계치를 적게 하거나 클래스 크기를 작게 함으로써 네트워크 상태 변화에 갱신 정책이 민감하게 반응하도록 해야 하는데 이것은 프로토콜 오버헤드를 증가시키게 된다. 또한, 갱신 임계치나 클래스에 기반한 갱신 정책만을 적용하면 상태 정보 갱신을 위한 프로토콜 오버헤드가 동적인 네트워크 상황에 전적으로 의존하게 되므로 그 최대치를 가능할 수 없다는 문제점이 있다. 이를 보완하기 위해 연속적인 갱신간에 최소 간격이 지켜지도록 하는 갱신 주기를 적용한 갱신 유발 정책을 병행하면 갱신 메시지의 양이 현저하게 감소될 뿐 아니라 최대 오버헤드를 예측할 수 있다는 장점이 있다.

그러나, 이렇게 갱신 주기를 적용하게 되면 갱신 오버헤드를 줄이기 위해 갱신 주기를 크게 할 경우 클래스의 크기나 임계치와 같은 갱신 유발 정책의 민감성에 영향을 미치는 파라미터가 전혀 라우팅 성능에 반영되지 않는다는 문제점이 있다[10, 11]. 또한, 기존의 갱신 정책에서는 예약 실패를 줄이기 위한 고려가 없다. 일반적으로 QoS 플로우의 경우 데이터 전송이 시작되기 전 경로 선정과 예약이 선행되어야 한다. QoS 라우팅에 의해 결정된 경로에 예약을 시도하였으나 상태 정보의 부정확성으로 인해 요구된 자원이 실제로 경로상에 존재하지 않아 예약이 경로의 중간에서 실패하는 경우에는 경로를 재계산 하는 작업부터 다시 시작해야 할 뿐 아니라 이미 예약이 진행된 경로의 일부에 대해서는 자원을 환원하는 등의 작업이 이루어져야 하므로 부정확한 네트워크 상태정보로 인해 실제로는 적정 경로가 존재하나 이를 발견하지 못하는 경우보다 오버헤드가 크다. 이 장에서는 이러한 두 가지 측면을 보완한 일련의 갱신 정책들을 제시한다.

3.1 네트워크 상태 변화 정도에 따라 이중 임계치를 사용하는 갱신 유발 정책

기존 연구에서 제시된 클래스 기반 갱신 유발 정책의 경우 갱신 주기를 적용할 때의 성능 저하가 완만하여,

갱신 주기의 크기와 관계없이 안정적인 라우팅을 할 수 있다는 장점을 갖고 있다. 그러나, [10]에서 보면 갱신 임계치에 의한 갱신 유발 정책에 비하여 전반적으로 라우팅 성능이 떨어짐을 알 수 있다. 반면에, 갱신 임계치 기반 갱신 유발 정책은 라우팅 성능이 좋은 편이지만, 갱신 주기를 적용하여 그 크기를 점점 크게 하면 라우팅 성능이 급격하게 저하될 뿐만 아니라 갱신 주기가 어느 정도 이상으로 커지면 임계치의 크기에 따른 라우팅 성능 및 오버헤드 조정이 전혀 불가능하다. 이는 갱신 임계치에 의한 갱신 유발이 네트워크의 상대적인 변화율에 의해 결정되기 때문에 네트워크 상태 변화를 반영하는 데는 효과적이지만, 갱신 주기를 동시에 적용하는 갱신 유발 정책을 사용했을 경우 갑작스럽게 발생한 심각한 네트워크 변화에는 신속하게 대처하기가 어렵기 때문이다. 마지막으로 공시된 네트워크 정보의 갱신 이후 트래픽 폭주로 인하여 가용 링크 대역폭 변화율이 매우 높은 상황이라 해도, 갱신 주기가 만료되지 않은 경우에는 네트워크 정보를 갱신할 수 없게 된다. 결국 라우팅은 각 노드(혹은 라우터)가 갖고 있는 기존 네트워크 정보로 이루어지기 때문에, 급격한 네트워크 상태 변화는 곧 라우팅 성능 저하로 이어지게 된다. 따라서, 갱신 주기에 미치지 못하는 단시간 내에 발생한 심각한 네트워크 변화에 대해서는 그에 대처할 수 있는 고유의 정책이 필요하다. 이에 본 논문에서는 기존 스킴에서와 같이 갱신 주기에 이르렀을 때 갱신 발생을 요망하는 갱신 주기를 임계치 $x\%$ 와 심각한 네트워크 변화율을 나타내는 즉시 갱신용 임계치 $y\%$ 의 두 가지 갱신 임계치를 두는 갱신 유발 정책을 제안하고, 이를 H-UTP (Heterogeneous threshold-based Update Triggering Policy)라고 명칭하였다.

그림 1에서 보는 바와 같이, H-UTP는 기존의 갱신 주기를 적용하는 임계치 $x\%$ 를 두고 임의의 노드에서 갱신 주기가 만료되면 그 노드의 인접 노드로의 가용 링크 대역폭의 변화율을 조사하여 이 변화율이 $x\%$ 이상이면 네트워크 정보를 갱신한다. 그리고 이와는 별도로, 즉시 갱신용 임계치 $y\%$ ($x < y$)를 두고 세션 수용 및 반환 시에 가용 링크 대역폭의 변화율이 $y\%$ 를 넘는지 조사한 후

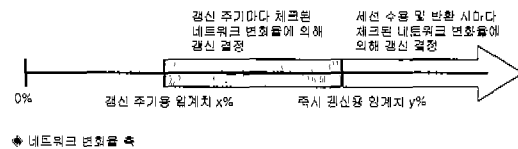


그림 1 H-UTP의 두 가지 갱신 임계치

화율이 이 임계치를 넘으면 갱신 주기와 상관없이 네트워크 정보를 갱신한다. 단, 즉시 갱신용 임계치 $y\%$ 를 넘어 갱신하게 되는 경우에도 갱신 주기 타이머를 재설정하여 주기적인 네트워크 상태 점검 시간을 즉시 갱신이 이루어지는 현재 시각으로부터 갱신 주기만큼 지난 시간이 되도록 해준다. 즉, 기존 갱신 임계치 기반 갱신 유발 정책에 갱신 주기를 적용하면서 즉시 갱신용 임계치를 뚫으로써 네트워크의 동적인 변화가 큰 경우에는 보다 적극적으로 대처할 수 있게 한 것이다.

3.2 자원 예약 실패를 줄이기 위한 갱신 유발 정책

세션 발생 후 경로를 설정해 줄 때에는 현재 실제로 사용한 링크 대역폭으로 설정하는 것이 아니라, 이전에 공시되었던 네트워크 상태 정보를 이용하여 설정하기 때문에 잘못된 경로 설정으로 예약 실패가 일어날 수 있다. QoS 보장을 위해서는 일반적으로 QoS 라우팅에 의해 결정된 경로 상에 자원을 예약하게 되는데, 예약 실패는 QoS 라우팅에 의해 발견한 경로 상에 실제로는 충분한 자원이 없어 예약을 성공적으로 수행할 수 없을 때 일어난다.

예약 실패는 앞서 설명한 것처럼 경로 재계산 및 자원 환원이라는 또다른 프로세싱 오버헤드를 초래하게 된다. 따라서, 이러한 예약 실패를 줄이기 위해서 토폴로지 상의 가용 링크 대역폭의 상태에 따라 민감도가 다른 갱신 임계치를 적용하는 방안인 D-UTP(Double threshold-based Update Triggering Policy)를 제안한다. 즉, 각 노드가 현재 가용 링크 대역폭과 마지막으로 공시한 가용 링크 대역폭의 크기를 비교하여, 현재 가용 링크 대역폭이 마지막으로 공시한 가용 링크 대역폭보다 작다면, 반대의 경우보다 더 낮은 값의 갱신 임계치를 적용하여 더 민감하게 네트워크 상태 정보를 갱신하도록 한다.

3.3 이중 갱신 임계치를 사용하여 자원 예약 실패를 줄이기 위한 갱신 유발 정책

3.2 절에서 제안한 D-UTP는 공시한 가용 링크 대역폭 정보로부터 현재 가용 링크 대역폭이 어느 방향으로 변화하였는지에 따라 서로 다른 갱신 임계치를 적용함으로써 예약 실패를 줄이고자 하였으나, 기존 갱신 임계치 기반 정책의 기본 개념을 그대로 유지하고 있기 때문에, 갱신 주기의 크기가 큰 경우 갱신 임계치의 크기가 라우팅 성능에 거의 영향을 미치지 못하게 된다는 단점을 갖고 있다. DH-UTP(Double and Heterogeneous threshold-based Update Triggering Policy)는 3.2절에서 제안한 D-UTP에 3.1절에서 제안한 H-UTP의 개념을 적용한 것이다.

그림 2에서 보는 바와 같이, DH-UTP는 H-UTP의 갱신 주기용 임계치와 즉시 갱신용 임계치를 두되, 현재 가

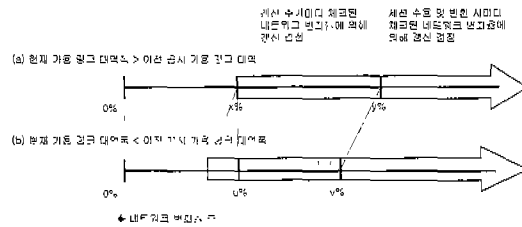


그림 2 DH-UTP의 갱신 임계치

용 링크 대역폭과 마지막으로 공시된 가용 링크 대역폭을 비교하여 현재 가용 링크 대역폭의 값이 더 작은 경우와 그 반대의 경우에 적용하는 파라미터 값의 민감도를 조절한다. DH-UTP의 갱신 주기용 임계치를 각각 $x\%$, $u\%$ 라 하고 즉시 갱신용 임계치를 각각 $y\%$, $v\%$ 라 할 때, 현재 가용 링크 대역폭이 더 작은 경우에 적용하는 $u\%$ 와 $v\%$ 의 값을 $x\%$ 와 $y\%$ 보다 더 민감하게 설정해준다.

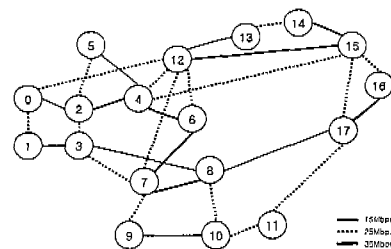
4. 시뮬레이션 설계 및 결과 분석

본 장에서는 시뮬레이션을 통하여, 제안하는 갱신 유발 정책의 성능 및 오버헤드를 비교, 분석하고자 한다. 시뮬레이션은 이벤트 운행정형(event driven)으로[15, 16], Sun Solaris 워크스테이션에서 C 언어로 구현하였으며 각 갱신 유발 정책에서 사용되는 파라미터 값을 변화시켜 보면서 시뮬레이션을 수행하였다. 4.1절에서는 성능 평가를 위한 시뮬레이션 설계 및 수행에 관하여 설명하고, 4.2절에서는 시뮬레이션 결과를 토대로 각 갱신 유발 정책의 성능 및 장단점을 비교·분석한다.

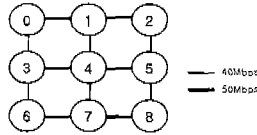
4.1 시뮬레이션 환경

4.1.1 네트워크 토폴로지(Network Topology)

그림 3의 (a)와 (b)는 시뮬레이션에 사용된 네트워크 토폴로지를 나타내고 있다. 사용된 토폴로지는 각각 ISP (Internet Service Provider) 구조와 3×3 메쉬 구조이다. ISP는 각각 25Mbps/sec, 35Mbps/sec, 45Mbps/sec의 대역폭을 가지는 링크들로 구성되고, 메쉬는 각각



(a) ISP 구조



(b) 매쉬 구조

그림 3 네트워크 토폴로지

40Mbps/sec, 50Mbps/sec 두 종류의 링크로 구성되고 가정하였다. 그리고, 두 구조 모두 링크의 대역폭은 양방향에 별개라고 가정하였다.

4.1.2 트래픽 모델(Traffic Model)

각 노드에서 발생하는 세션은 QoS 요구사항으로서 대역폭만을 표시하도록 하였으며, 각 세션은 송신원(source)과 수신원(destination) 그리고 요청 대역폭으로 특징 지워진다. 세션 발생 간격은 지수 분포 랜덤 변수(exponentially distributed random variable)로서, 평균 발생 간격은 3초이며, 세션 길이는 플로우 유지 시간으로서 이 역시 지수 분포 랜덤 변수이고, 평균 세션 길이는 180초(3분)이다. 세션이 발생하면 이 세션의 송신원과 수신원이 결정되는데, 송신원 노드는 균일 분포 확률(uniformly distributed probability)에 의해 결정되며, 수신원 노드 역시 송신원 노드를 제외한 다른 모든 노드 중 균일 분포 확률에 의해 결정된다. 세션의 요청 대역폭은 $[L, U]$ 구간 사이에서 균일하게 분포되도록 하였는데, 시뮬레이션에서 사용된 대역폭의 최소값 L 은 64Kbits/sec, 최대값 U 는 6Mbps/sec 혹은 10Mbps/sec이다. 최대값 U 를 6Mbps/sec로 하였을 때와 10Mbps/sec로 하였을 때의 네트워크 전체의 평균적인 링크 활용률은 각각 30%, 50% 정도이다. 각 시뮬레이션에서는 총 100,000개의 세션 요청이 발생되도록 하였다.

4.1.3 라우팅 알고리즘 및 각 갱신 유발 정책에 필요한 파라미터

시뮬레이션에서 사용하는 라우팅 알고리즘은 “on-demand widest-shortest” 경로 알고리즘으로, 이를 구현하는 데에는 기존의 최단 경로 알고리즘인 Bellman-Ford 알고리즘이나 Dijkstra 알고리즘이 사용될 수 있는데, 여기서는 Dijkstra 알고리즘의 변형을 통해 라우팅 알고리즘을 구현하였다[13, 14]. Dijkstra 알고리즘으로 “widest-shortest” 경로 알고리즘을 구현하는 방법은 다음과 같다 : 경로를 계산하기 전에, 세션이 요구하는 대역폭에 대하여 이를 수용할 수 없는 링크들은 네트워크 토폴로지로부터 제거한다. 그리고 나서 송신원과 수신원 사이의 최소의 홉 수를 갖는 경로를 찾으면,

이 중 가장 넓은 대역폭을 갖는 경로가 요청에 부응하는 경로가 된다. 만약 하나 이상의 “widest-shortest” 경로가 있으면, 그 중 하나가 임의로 선택된다.

갱신 유발 정책에서 사용되는 파라미터들로는 임계치와 갱신 주기가 있다. 각 갱신 유발 정책에 따라 필요로 하는 임계치들의 종류와 갱신 주기를 요약하여 정리하면 다음과 같다.

표 1 갱신 유발 정책에 사용되는 파라미터들

갱신 유발 정책	파라미터	공통사항
갱신 임계치	임계치 $a\%$	갱신 주기 t (sec)
H-UTP	갱신 주기용 임계치 $x\%$ ~ 즉시 갱신용 임계치 $y\%$	
D-UTP	$b_{curr} < b_{last}$ 시에 적용하는 임계치 $u\%$, $b_{curr} > b_{last}$ 시에 적용하는 임계치 $v\%$	
DH-UTP	$b_{curr} < b_{last}$ 시에 적용하는 갱신 주기용 임계치 $k\%$ ~ 즉시 갱신용 임계치 $l\%$, $b_{curr} > b_{last}$ 시에 적용하는 갱신 주기용 임계치 $m\%$ ~ 즉시 갱신용 임계치 $n\%$	

(단, b_{curr} : 현재 가용 링크 대역폭, b_{last} : 마지막으로 붕괴된 가용 링크 대역폭)

4.2 시뮬레이션 결과 분석

QoS의 라우팅 성능 및 프로토콜 오버헤드는 다음과 같은 세 가지 요인에 의해 영향을 받으며, 서로 다른 갱신 유발 정책은 서로 다른 종류의 파라미터를 사용하기 때문에 갱신 유발 정책 간에 직접적으로 성능 측정치를 비교하기보다는 갱신 유발 정책 별로 성능 특성을 파악하고자 한다.

- 사용되는 갱신 유발 정책
 - 갱신 유발 정책에서 사용하는 파라미터의 민감도 수준
 - 갱신 주기의 적용 유무 및 갱신 주기의 크기
- 그리고, 시뮬레이션 결과를 표시하는데 있어서 각 갱신 유발 정책은 다음과 같은 기호로 표현한다.

표 2 갱신 유발 정책의 기호 표현법

갱신 유발 정책	기호 표현법
갱신 임계치	T
H-UTP	H
D-UTP	D
DH-UTP	DH

4.2.1 프로토콜 오버헤드

프로토콜 오버헤드는 시뮬레이션 동안 각 노드에서 발생시키는 갱신 메시지의 총 수를 모두 합산한 것으로 측정한다. 프로토콜 오버헤드를 분석하기 위한 시뮬레이션 환경은 ISP 토폴로지 상에서 링크 활용률이 30%일 때를 기준으로 한다. 그림 4는 기존에 제시된 갱신 임계치 기반의 갱신 유발 정책에서 갱신 주기의 크기에 따른 프로토콜 오버헤드를 나타내는 그래프이다. 시뮬레이션에서 사용된 갱신 임계치는 각각 10%, 60%, 90%이다. 그림에서 보는 바와 같이, 갱신 임계치 정책에서 갱신 주기를 적용하지 않을 때에는 갱신 임계치의 민감도에 따른 갱신 메시지 발생량에 큰 차이가 있다. 그러나, 갱신 주기가 커지면서 갱신 메시지 발생량은 급격하게 감소하고 민감도에 의한 오버헤드 차이는 점차 줄어들어, 갱신 주기가 클 때에는 프로토콜 오버헤드에 거의 영향을 미치지 않는 것을 알 수 있다.

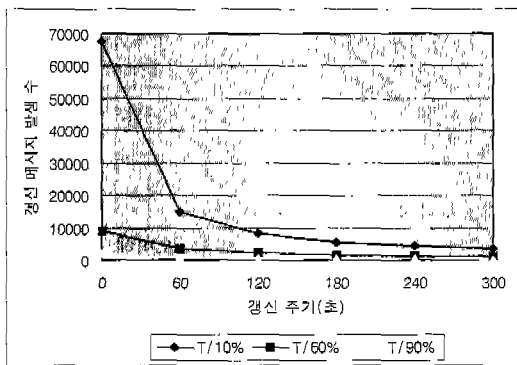


그림 4 갱신 임계치 기반 갱신 유발 정책의 프로토콜 오버헤드

그림 5는 H-UTP에서의 프로토콜 오버헤드를 나타내는 그래프이다. H-UTP에서는 갱신 주기용 임계치를 고정값으로 두고, 즉시 갱신 임계치를 변화시킨다. 갱신 주기용 임계치는 60%, 즉시 갱신용 임계치는 각각 80%, 90%, 100%로 설정한다. 기존의 갱신 임계치 정책에서는 갱신 주기의 크기가 커짐에 따라 프로토콜 오버헤드가 급격하게 감소하고 갱신 주기가 클 때에는 갱신 임계치의 민감도와 프로토콜 오버헤드의 양이 독립적인데 비해, H-UTP는 갱신 주기가 증가함에 따른 프로토콜 오버헤드의 감소가 완만하게 이루어질 뿐만 아니라, 갱신 주기가 클 때에도 갱신 유발 정책 파라미터의 민감성이 오버헤드에 영향을 미침을 알 수 있다. 그림에서

보는 바와 같이, 즉시 갱신용 임계치가 각각 80%, 90%, 100%일 때를 비교해 보면, 즉시 갱신 임계치가 작을수록 프로토콜 오버헤드 발생량이 더 많은 것을 볼 수 있다. 이것은 H-UTP가 갱신 주기에 영향을 받지 않는 즉시 갱신용 임계치를 이용하여 심각한 네트워크 변화에 대한 네트워크 상태 갱신 정보를 공시할 수 있기 때문이다. H-UTP에서 갱신 주기가 작을 때에는 갱신 주기용 임계치에 의한 갱신 메시지가 주를 이루지만, 갱신 주기의 크기가 크면 즉시 갱신용 임계치에 의한 갱신 메시지 비중이 커지게 된다. 따라서, 기존 갱신 임계치는 갱신 주기가 커질수록 갱신 유발 정책 파라미터의 민감도에 따른 프로토콜 오버헤드 발생량의 차이가 줄어드는 반면에, H-UTP는 오히려 프로토콜 오버헤드 격차가 좀 더 벌어지는 현상을 볼 수 있다.

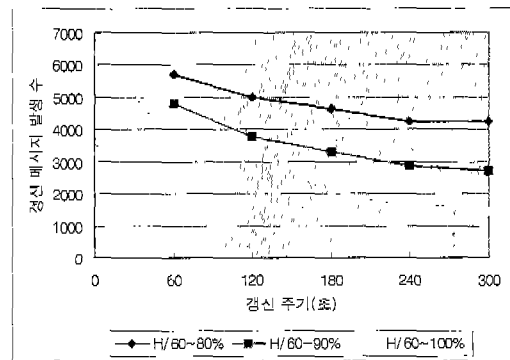


그림 5 H-UTP의 프로토콜 오버헤드

그림 6은 D-UTP의 프로토콜 오버헤드를 나타내고 있다. D-UTP에서는 현재 가용 링크 대역폭이 마지막으로 공시된 가용 링크 대역폭 보다 클 때 적용하는 갱신 임계치를 90%로 고정시키고, 반대의 경우에 적용하는 갱신 임계치의 민감도를 30%, 50% 70%로 조정해 보았다. 그림에서 보는 바와 같이, D-UTP는 기존의 갱신 임계치 경우와 유사하게 갱신 주기가 없거나 작을 때에는 갱신 임계치의 민감도에 따른 오버헤드 발생량에 차이가 뚜렷하게 나타난다. 현재 가용 링크 대역폭이 마지막으로 공시된 가용 링크 대역폭보다 작을 때 사용하는 갱신 임계치가 작을수록 프로토콜 오버헤드 발생량이 많으나, 갱신 주기가 커질수록 프로토콜 오버헤드는 급격히 감소하여, 갱신 주기가 클 때에는 D-UTP에 적용하는 갱신 임계치의 민감도에 따른 오버헤드 발생량의 격차가 좁혀진다.

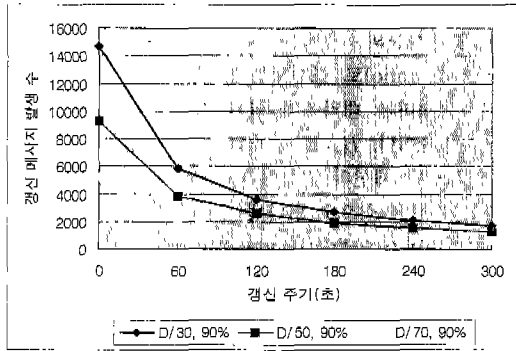


그림 6 D-UTP의 프로토콜 오버헤드

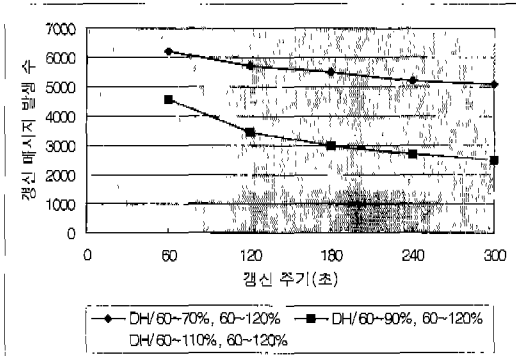


그림 7 DH-UTP의 프로토콜 오버헤드

그림 7은 DH-UTP의 프로토콜 오버헤드를 나타내는 그래프이다. DH-UTP에서는, D-UTP와 마찬가지로 현재 가용 링크 대역폭이 마지막으로 공시된 가용 링크 대역폭보다 클 때 적용하는 갱신 임계치를 고정시키고, 반대의 경우에 적용하는 갱신 임계치에 변화를 주었다. 두 경우 모두 갱신 주기용 임계치는 60%로 놓고, 즉시 갱신용 임계치의 민감도를 조절하여 갱신 주기의 변화에 따른 프로토콜 오버헤드 양을 비교해 본다. 그림에서 보는 바와 같이, DH-UTP는 H-UTP와 비슷한 그래프 기형을 보이고 있다. DH-UTP 역시 갱신 주기용 임계치와 즉시 갱신용 임계치를 이용하여 갱신 유발을 하기 때문에, 갱신 주기가 크더라도 즉시 갱신 임계치의 민감도에 따른 프로토콜 오버헤드 발생량 차이가 분명하게 나타남을 볼 수 있다.

제한한 세 가지의 갱신 유발 정책들은 모두 기존 갱신 임계치 정책에 비해 프로토콜 오버헤드 비용이 증가하였다. 그러나, 증가된 오버헤드는 모두 상대적으로 심

각한 네트워크 변화를 알리거나 예약 실패 가능성을 줄이기 위한 것이므로 라우팅 성능에 중요한 정보를 교환하기 위한 오버헤드이다. H-UTP와 DH-UTP는 즉시 갱신 임계치의 사용으로 인하여 갱신 주기의 크기에 상관없이 갱신 유발 정책에 사용된 파라미터의 민감도에 따라 프로토콜 오버헤드가 변화하게 된다. 반면에, D-UTP는 갱신 주기의 크기가 작은 경우에는 파라미터 값의 민감도가 프로토콜 오버헤드에 큰 영향을 미치지 만, 큰 경우에는 민감도가 미치는 영향은 작다. 이 논문 에 수치적 결과를 제시하지는 않았으나 동일한 실험을 링크 활용률이 다른 경우에 대해서도 적용해 보았는데, 링크 활용률에 따른 프로토콜 오버헤드 발생량은 절대적인 수치에만 차이가 있을 뿐, 프로토콜 오버헤드 특성 자체에는 직접적인 영향을 주지 않음을 볼 수 있었다.

4.2.2 예약 실패

예약 실패는 시뮬레이션 동안 발생하는 예약 실패 수를 모두 합산하여 나타낸다. 예약 실패 역시 ISP 토폴로지 상에서 링크 활용률이 30%일 때를 기준으로 시뮬레이션 결과를 분석하도록 한다.

그림 8은 D-UTP의 예약 실패에 관한 그래프이다. D-UTP의 예약 실패 시 감소 효과를 분석하기 위하여 갱신 임계치 90%를 적용한 기존 정책과 비교하도록 한다. 그림에서 보는 바와 같이, D-UTP는 기존 갱신 임계치 정책보다 예약 실패가 더 적다. 이는 예약 실패를 일으키는 경우인 현재 가용 링크 대역폭이 마지막으로 공시된 링크 가용 대역폭보다 작을 때에 네트워크 변화에 보다 민감하게 반응하도록 하여 이 때의 갱신 임계치 값을 작게 설정해 주었기 때문이다. 또한, 갱신 주기의 크기의 변화에 따른 예약 실패의 특성을 살펴보면, 갱신 주기가 없거나 작을 때에는 현재 가용 링크 대역폭이 공시된 가용 링크 대역폭 보다 작을 때 적용하는

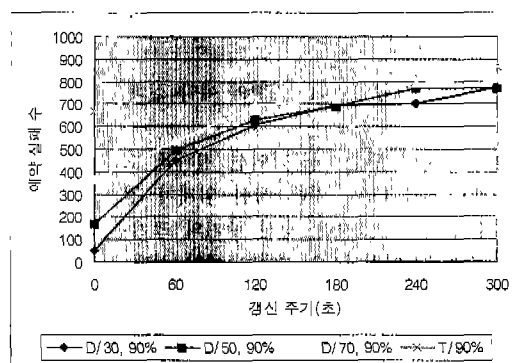


그림 8 D-UTP의 예약 실패

갱신 임계치가 낮을수록 예약 실패가 적게 나타나며, 갯신 임계치의 민감도에 따라서 예약 실패의 차이가 뚜렷하다. 그러나, 갯신 주기의 크기가 점점 커질수록 예약 실패의 수가 크게 증가하고, 갯신 주기의 크기가 클 때에는 갯신 임계치의 민감도에 따른 예약 실패의 차이도 점점 줄어드는 것을 볼 수 있다. 이것은 예약 실패를 줄이기 위해 현재 가용 링크 대역폭이 작은 경우에 적용하는 갯신 임계치를 더 낮게 설정하더라도, 두 개의 갯신 사이의 최소 간격인 갯신 주기의 크기가 크기 때문에, 갯신이 순수한 네트워크 변화에 의해서 이루어진다고 보다는 갯신 주기에 의해 결정되기 때문이다.

그림 9는 DH-UTP의 예약 실패에 관한 그래프이다. DH-UTP의 예약 실패는 60%~120%를 적용한 H-UTP 정책과 비교한다. 그림에서 보는 바와 같이, DH-UTP는 H-UTP보다 예약 실패를 감소시키며, 현재 가용 링크 대역폭이 마지막으로 공시된 가용 링크 대역폭보다 작을 때 적용하는 즉시 갯신용 임계치의 민감도에 따라 예약 실패 수의 차이가 크게 나타난다. 특히, DH-UTP의 예약 실패 특성이 기존 갯신 임계치 정책이나 D-UTP 정책과 다른 점은 갯신 주기의 크기가 커지더라도 예약 실패 증가율이 낮으며, 즉시 갯신 임계치의 민감도에 따른 예약 실패 차이가 크게 나타난다는 것이다.

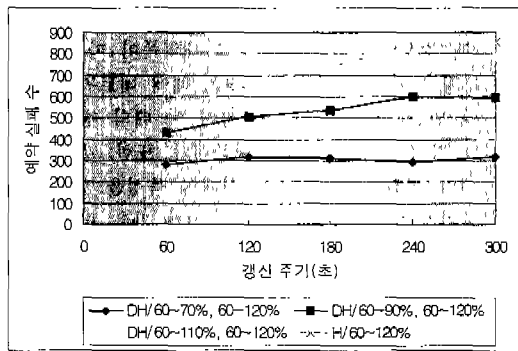


그림 9 DH-UTP의 예약 실패

예약 실패에 관한 시뮬레이션 결과를 요약하면 다음과 같다. D-UTP는 기존 갯신 임계치 정책보다 예약 실패를 줄일 수 있지만, 갯신 주기의 크기에 큰 영향을 받기 때문에, 이것이 클 때에는 예약 실패 감소 효과가 떨어지게 된다. 이를 보완하기 위해 제안한 DH-UTP에서는 갯신 주기의 크기가 커지더라도 즉시 갯신 임계치에 의한 갯신 유발이 이루어지기 때문에, 즉시 갯신 임계치의 민감도에 따라 예약 실패를 조절할 수 있다.

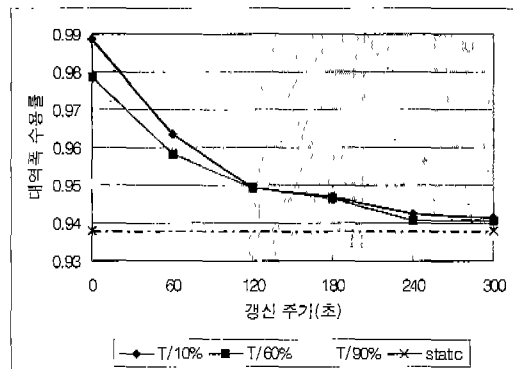
4.2.3 라우팅 성능

라우팅 성능의 측정은 대역폭 수용률(bandwidth acceptance ratio)로 표현되는데, 대역폭 수용률이란 각 세션이 요청하는 대역폭의 합과 실제 수용된 대역폭의 합간의 비율이다. B_{acc} 를 수용된 세션의 집합이라 하고, B_{tot} 를 요구된 세션의 집합이라고 할 때, 대역폭 수용률은 다음과 같은 식으로 나타낼 수 있다.

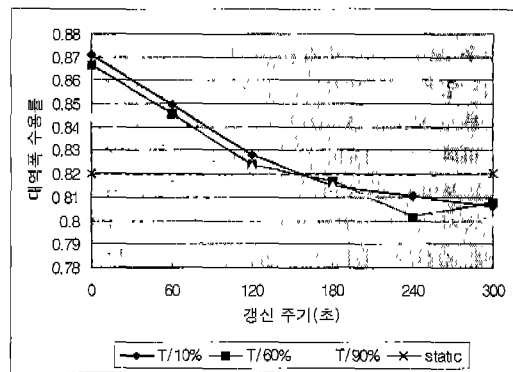
$$\text{대역폭 수용률(bandwidth acceptance ratio)} = \frac{\sum_{i \in B_{acc}} \text{bandwidth}(i)}{\sum_{i \in B_{tot}} \text{bandwidth}(i)}$$

또한, 본 시뮬레이션에서는 각 갯신 유발 정책들의 라우팅 성능을 비교 평가하기 위한 기준으로 네트워크 정보 갯신이 일어나지 않는 정적(static) 라우팅의 성능을 함께 보이도록 한다.

그림 10의 (a)와 (b)는 ISP 토폴로지 상에서 링크 활용률이 각각 30%, 50%일 때의 기존 갯신 임계치 기반의 갯신 임계치 정책에서의 라우팅 성능을 나타내는 그



(a) 링크 활용률이 30%일 때



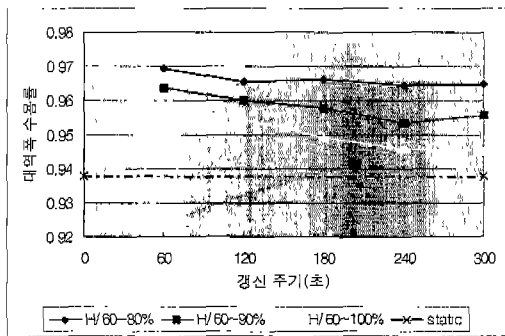
(b) 링크 활용률이 50%일 때

그림 10 갯신 임계치 기반 갯신 유발 정책의 라우팅 성능

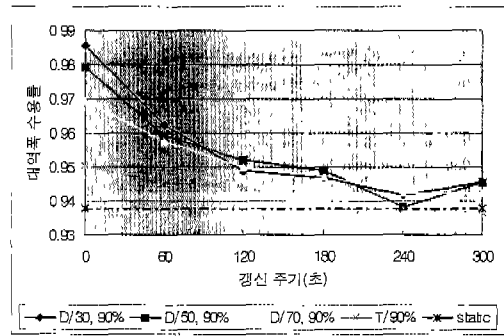
래프이다. 기존 갱신 임계치 정책은 갱신 주기에 의해서 라우팅 성능의 특성이 달라진다. 그림에서 보는 바와 같이, 갱신 주기의 크기가 작을 때에는 갱신 임계치의 민감도에 따라 라우팅 성능에 차이가 나타나지만, 갱신 주기의 크기가 크면 갱신 임계치의 민감도에 따른 라우팅 성능 차가 거의 없을 뿐 아니라 라우팅 성능이 급격히 저하하여 정적 라우팅 수준에 이르게 된다. 특히, 링크 활용률의 높을 때에는 이러한 현상이 더욱 두드러지게 나타나며, 갱신 주기가 어느 정도 커지게 되면 정적 라우팅보다도 성능이 떨어진다.

그림 11의 (a)와 (b)는 ISP 토폴로지 상에서 링크 활용률이 각각 30%, 50%일 때의 H-UTP의 라우팅 성능을 나타내고 있다. 기존 갱신 임계치 정책이 갱신 주기의 크기가 커짐에 따라 급격한 성능 저하를 보이는 반면에, H-UTP는 그 감소폭이 완만하게 나타난다. 또한, 기존 갱신 임계치 정책은 갱신 주기가 클 때 갱신 임계치 민감도에 따른 라우팅 성능 차이가 크게 나타나지 않는데 비해, H-UTP에서는 즉시 갱신용 임계치의 민감도에 따라 라우팅 성능에 차이를 보이고 있다. 이는 즉시 갱신용 임계치에 의해 갱신 주기에 관계없이 네트워크의 심각한 변화가 있을 때마다 네트워크 정보 교환을 위한 갱신이 일어나서 보다 정확한 네트워크 정보를 이용하여 라우팅 계산을 할 수 있기 때문이다. 그림에서 보는 바와 같이, H-UTP는 갱신 주기 증가에 따른 라우팅 성능 감소율이 두 가지의 링크 활용률 경우에서 거의 비슷하게 나타난다.

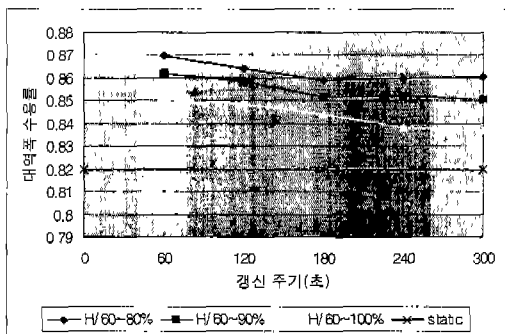
그림 12의 (a)와 (b)는 ISP 토폴로지 상에서 링크 활용률이 각각 30%, 50%일 때의 D-UTP의 라우팅 성능을 나타내고 있다. D-UTP는 기존 갱신 임계치 정책의 라우팅 성능 변화와 비슷한 패턴을 보이고 있다. 그러나, 파라미터 값 90%를 적용한 기존 갱신 임계치 정책과 비교하면, 예약 실패 감소와 필요한 경우 적절하게 갱신 메시지를 더 발생시킴으로 인해 라우팅 성능이 향상되었음을 볼 수 있다. 또한, D-UTP는 갱신 주기가 없거나 작을 때에는 현재 가용 링크 대역폭이 더 작을 때 적용하는 갱신 임계치의 민감도에 따라 라우팅 성능



(a) 링크 활용률이 30%일 때

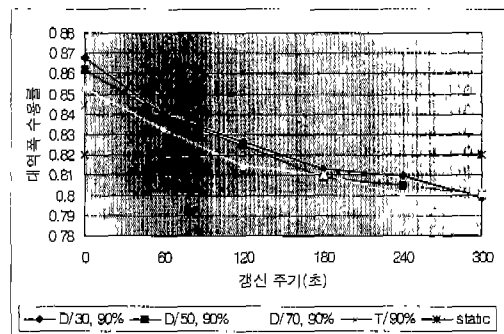


(a) 링크 활용률이 30%일 때



(b) 링크 활용률이 50%일 때

그림 11 H-UTP의 라우팅 성능



(b) 링크 활용률이 50%일 때

그림 12 D-UTP의 라우팅 성능

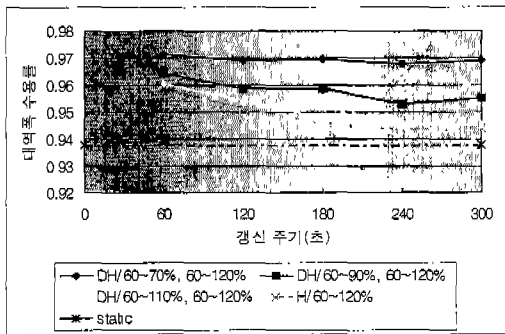
에 차이를 보이고 있으나, 갱신 주기의 크기가 커짐에 따라 급격한 성능 저하를 일으키고 갱신 임계치의 민감도에 따른 성능 차이가 불분명해진다. 갱신 주기의 크기가 클 때에는 기존 갱신 임계치 정책과 마찬가지로 정적 라우팅 수준 이하로 떨어지며, 링크 활용률이 높을 때 라우팅 성능이 더 많이 떨어진다.

그림 13의 (a)와 (b)는 ISP 토폴로지 상에서 링크 활용률이 각각 30%, 50%일 때의 DH-UTP의 라우팅 성능을 나타내고 있다. DH-UTP는 H-UTP와 마찬가지로, 갱신 주기가 커지더라도 라우팅 성능 감소율이 매우 적다. 이는 DH-UTP가 즉시 갱신용 임계치에 의해 네트워크 상태 변화에 빠르게 대응할 수 있기 때문이다. 또한, 예약 실패를 방지하기 위해 현재 가용 링크 대역폭이 마지막으로 공시된 가용 링크 대역폭보다 작을 때 사용하는 즉시 갱신 임계치를 반대의 경우보다 더 낮게 설정함으로써, 파라미터 값 60~120%를 사용하는 H-UTP에 비해 네트워크 정보 공시가 많기 때문에 라우팅 성능이 향상된다. 또한, DH-UTP는 갱신 주기가

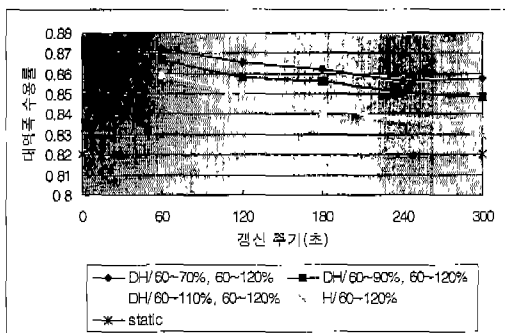
클 때에도 즉시 갱신용 임계치의 민감도에 따른 성능 차이가 나타나고 있으며, 링크 활용률이 높을 때 정적 라우팅에 비한 라우팅 성능 향상 효과가 더욱 두드러지게 나타남을 알 수 있다.

4.2.4 토폴로지 구조에 따른 라우팅 성능

그림 14는 네트워크 토폴로지가 라우팅 성능에 미치는 영향을 살펴보기 위하여, 메쉬 구조에서 링크 활용률 50%일 때의 라우팅 성능을 나타낸 것이다. 이 그림에서 보는 바와 같이, 메쉬 구조에서 갱신 유발 정책 및 갱신 주기에 따른 라우팅 성능의 특성은 ISP와 비슷하다. 기존 갱신 임계치 기반 정책이나 D-UTP의 경우에는 갱신 주기의 크기가 커짐에 따라 라우팅 성능이 급격하게 떨어지는 반면에, H-UTP와 DH-UTP는 즉시 갱신용 임계치에 의해서 라우팅 성능에 차이를 유지하면서 성능의 감소폭도 완만하게 나타난다. 다만, ISP 구조와 비교했을 때 다른 점은 트래픽 로드가 크더라도 모든 갱신 유발 정책들의 성능이 정적 라우팅보다 떨어지지 않는다는 것이다. 이는 메쉬 구조의 특성상 한 쌍의 송신원과 수신원 사이에 여러 개의 경로가 존재하기 때문에 이러한 경로들을 충분히 활용할 수 있기 때문이다.

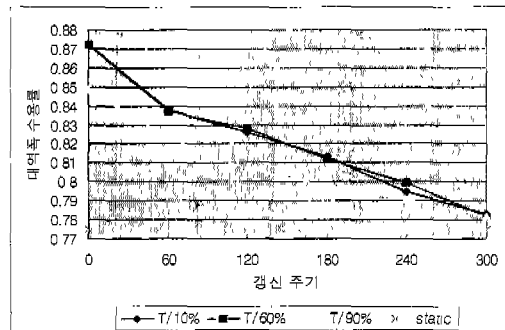


(a) 링크 활용률이 30%일 때

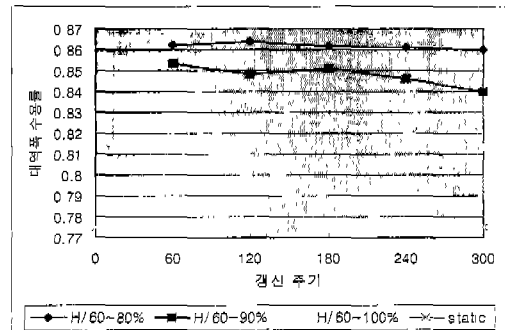


(b) 링크 활용률이 50%일 때

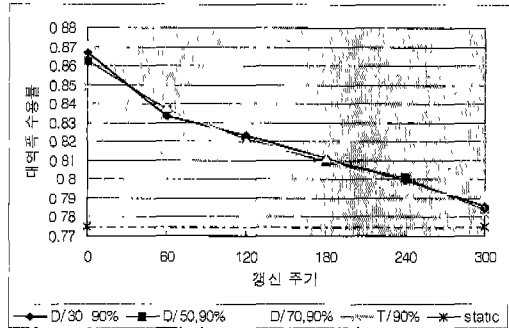
그림 13 DH-UTP의 라우팅 성능



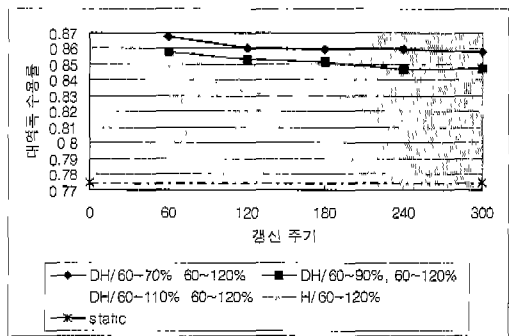
(a) 갱신 임계치 기반 정책



(b) H-UTP



(c) D-UTP



(d) DH-UTP

그림 14 메쉬 구조에서의 각 갱신 유발 정책의 라우팅 성능

5. 결론

QoS 라우팅은 네트워크의 동적인 상태 정보를 이용하여 적정 경로를 찾아줌으로써 라우팅 성능과 네트워크 활용률을 향상시킬 수 있다. 그러나, 이를 위해 동적인 네트워크 상태 정보를 빈번하게 교환해야 하므로 프로토콜 오버헤드를 증가시키게 된다. 이러한 프로토콜 오버헤드를 줄이기 위해 갱신 주기를 적용하는 방안이 제시되었으나 갱신 주기가 커지면 갱신 유발 자체의 민감성이 전혀 반영되지 않고 갱신 유발 정책의 부정확한 네트워크 정보로 인한 라우팅 성능의 저하 및 자원 예약 실패의 발생을 초래하게 된다. 이에 본 논문에서는 이러한 일련의 문제점을 보완하기 위해 H-UTP, D-UTP, DH-UTP 등 세가지의 갱신 유발 정책을 제안하였다. H-UTP는 심각한 네트워크 변화에 대하여 갱신을 유발하는 즉시 갱신용 임계치를 높으로써 갱신 주기가 큰 경우에 라우팅 성능이 급격히 저하하는 것을 막

고 임계치의 민감도에 따라 라우팅 성능을 조절할 수 있다. D-UTP는 현재 가용 링크 대역폭이 표시된 가용 링크 대역폭보다 적은 경우 더 민감한 갱신 유발 정책을 사용함으로써 예약 실패를 감소시키고 기존 갱신 임계치 기반의 갱신 유발 정책에 비하여 라우팅 성능도 향상시켰으나, 갱신 주기가 클 때에는 그 효과를 기대하기 어렵다. DH-UTP는 즉시 갱신용 임계치의 사용으로 인하여 갱신 주기에 상관없이 임계치의 민감도에 따라 라우팅 성능을 조절할 수 있으며, 예약 실패 감소로 인한 라우팅 성능 향상 효과가 나타났다. 각 스킴의 라우팅 성능은 링크 활용률, 즉 트래픽 로드(load)에도 영향을 받으며, 특히 트래픽 로드가 불수록 제안하는 스킴의 상대적인 효율성이 커짐을 볼 수 있었다. 라우팅 성능은 네트워크 토폴로지에도 영향을 받으며 네트워크 토폴로지에 다중 경로가 많은 경우 정적 라우팅의 성능에 비해 제안하는 스킴들의 라우팅 성능 향상이 더 클 수 있었다.

참고 문헌

- [1] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, "A Framework for QoS-Based Routing in the Internet," Internet Draft, 1998.
- [2] F. Ergun, R. Shinha, L. Zhang, "QoS Routing with Performance-Dependent Costs," IEEE INFOCOM, 2000.
- [3] Z. Wang, J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications," IEEE JSAC, 1996.
- [4] Q. Ma, P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees," ICNP, 1997.
- [5] G. Apostolopoulos, R. Guerin, S. Kamat, "Implementation and Performance Measurements of QoS Routing Extensions to OSPF," IEEE INFOCOM, 1999.
- [6] R. Guerin, A. Orda, "QoS-based Routing in Networks with Inaccurate Information : Theory and algorithms," IEEE INFOCOM, 1997.
- [7] R. Guerin, A. Orda, D. Williams, "QoS Routing Mechanisms and OSPF Extensions," Internet Draft, 1996.
- [8] G. Apostolopoulos, R. Guerin, S. Kamat, "Intra-Domain QoS Routing in IP Networks : A Feasibility and Cost/Benefit Analysis," ACM Journal Network on Transaction, 1999.
- [9] Q. Ma, P. Steenkiste, "Routing Traffic with Quality-of-Service Guarantees in Integrated Services Networks," IEEE INFOCOM, 1999.
- [10] G. Apostolopoulos, R. Guerin, S. Kamat, S. K.

- Tripathi, "Quality of Service Based Routing : A Performance Perspective," ACM SIGCOMM, 1998.
- [11] 김지혜, 이미정, "QoS 라우팅에서 새로운 갱신 유발 정책에 따른 성능 개선에 관한 연구", 한국정보과학회 추계학술대회, 2000.
- [12] G. Apostolopoulos, R. Guerin, S. Kamat, S. K. Tripathi, "Improving QoS routing Performance Under Inaccurate Link State Information," ITC16, 1999.
- [13] G. apostolopoulos, S. K. Tripathi, "On Reducing the Processing Cost of On-Demand QoS Path Computation," ICNP, 1997.
- [14] G. apostolopoulos, S. K. Tripathi, "On the Effectiveness of Path Pre-Computation in Reducing the Processing Cost of On-Demand QoS Path Computation," in Journal of High Speed Networks, Vol.7, No.2, pp.77-98, 1998.
- [15] Q. Ma, P. Steenkiste, "Quality-of-Service Routing for Traffic with Performance Guarantees," IFIP, 1997.
- [16] F. Hao, E. W. Zegura, "On Scalable QoS Routing : Performance Evaluation of Topology Aggregation," IEEE INFOCOM, 2000.



김 지 혜

1995년 ~ 1999년 이화여자대학교 컴퓨터학과(학사). 1999년 ~ 2001년 이화여자대학교 과학기술대학원 컴퓨터학과(공학석사). 2001년 ~ 현재 텔슨전자 연구소 연구원. 관심분야는 QoS 라우팅, 라우팅 프로토콜, CDMA 무선 이동통신



이 미 정

1983년 ~ 1987년 이화여자대학교 전자계산학 학사. 1987년 ~ 1989년 University of North Carolina at Chapel Hill 컴퓨터학 석사. 1990년 ~ 1994년 North Carolina State University 컴퓨터공학 박사. 1994년 ~ 현재 이화여자대학교 공과대학 컴퓨터학과 부교수. 관심분야는 고속 통신 프로토콜 설계 및 성능 분석, 비디오 전송을 위한 트래픽 제어, QoS 라우팅, 다중경로 라우팅, Ad-hoc 네트워크