

멀티캐스팅 프로토콜의 트리 구성에 관한 성능 평가 및 분석

정희원 김현숙*, 정찬윤*, 이용*, 송주석*, 고석주**

Performance Evaluation of Tree Creation in Multicasting Protocol

HyunSook Kim*, ChanYoon Jung*, Yong Lee*, JooSeok Song*, SeokJoo Koh**

Regular Members

요 약

멀티캐스팅이란 일대일의 전형적인 통신 모델과 one-to-all 방식의 브로드캐스팅의 중간적인 일대다 통신 모델이다. 여러 가지 통신 어플리케이션에서 멀티캐스팅 서비스를 제공하기 위해서는 많은 요구사항이 필요하며, 이를 위하여 여러 가지 멀티캐스팅 프로토콜이 연구되어 왔다. 그 중 신뢰도를 보장하기 위하여 논리적인 트리를 구성하고, 이러한 계층적 트리 기반의 복구 메커니즘을 이용하여 오류를 복구, 흐름 및 혼잡을 제어하는 방식이 Tree-based Multicasting Protocol이다.

본 논문에서는 Tree-based Multicasting Protocol에서 기본이 되는 논리적 트리를 구성하는데 있어 Top-down 방식을 따르는 프로토콜과 Bottom-up 방식을 따르는 프로토콜의 트리 구성 방식을 살펴보고, 시뮬레이션을 통하여 멀티캐스팅 그룹에 참여하는 노드수가 증가함에 따라 트리 구성 시 교환되는 메시지 수와 트리 구성 지연 시간, 트리 Depth 등을 비교하였다.

ABSTRACT

Multicasting is a one-to-many communication model and lies in between unicast and broadcast communication. Various real applications require many considerations in order to support the multicasting services and many researches have carried out the study of the various multicasting protocol. Specially, a Tree-based Multicasting Protocol ensures reliability by adopting a hierarchical tree-based repair mechanism and also controls the flow and congestion by using that logical tree.

We consider top-down scheme and bottom-up scheme in the tree creation, which is an important part of Tree-based Multicasting Protocol. Also, we simulate and evaluate the performance, and compare two schemes in terms of the total number of exchanged messages, the tree creation delay and the tree depth.

1. 서론

최근 인터넷의 확장과 더불어 멀티캐스팅의 기술은 인터넷상의 통신 어플리케이션의 질을 향상시킬 수 있는 중요한 기술로 부각되어지고 있다. 즉, 멀

티캐스팅 기술은 다자간의 데이터 전송과 원격강의 및 화상 회의에 사용할 수 있는 통신 어플리케이션의 개발에 사용되어질 수 있다.

다양한 실세계의 응용들이 멀티캐스트 통신 모델에 속할 수 있다. 이러한 응용들은 신뢰성과 지연 요구조건에 따라서 다음과 같이 3가지 큰 부류로

* 연세대학교 컴퓨터과학과 정보통신 연구실(jjoelle, jucys, ylee, jssong}@emerald.yonsei.ac.kr),

** 한국전자통신연구원 표준연구센터(sjkoh@pec.etri.re.kr)
논문번호 : 00441-1128, 접수일자 : 2000년 11월 28일

나눌 수 있다.

첫 번째는 엄격한 지연 요구 조건을 가지는 컨퍼런스와 같은 상호작용을 하는 실시간 응용군이다. 이 범주에 속하는 응용들은 비디오 데이터 고유의 중복성 때문에 손실에 민감하지 않다.

두 번째로 도큐먼트나 소프트웨어의 분배와 같은 신뢰적인 멀티캐스트 응용들은 엄격하게 완전한 신뢰성을 요구한다. 그러나 이러한 응용은 상호작용을 하는 실시간 응용과 같은 엄격한 지연 시간 조건이 요구되지는 않는다.

세 번째 형태의 응용군은 두 가지 극단의 응용군 사이에 존재한다. 즉, 상호작용을 하는 실시간 응용들보다는 덜한 지연 시간을 요구하고 신뢰적인 멀티캐스트보다는 덜한 신뢰성을 요구하는, 음악이나 영화와 같은 스트리밍 서비스가 이에 속한다.

신뢰성이 요구되는 멀티캐스트 응용들을 수용하기 위해서는 일대일 통신에서 신뢰성을 제공하기 위해 사용하는 TCP(Transmission Control Protocol)와 같은 신뢰적인 멀티캐스트 전송 프로토콜이 필요하다. 멀티캐스트 그룹에는 다양한 물리적인 지역에서 많은 사용자들이 자유롭게 가입할 수 있어야 하기 때문에, 신뢰적인 멀티캐스트 프로토콜에서의 가장 큰 주안점은 확장성이다. 확장 가능한 신뢰적인 멀티캐스트 프로토콜에 대한 연구는 활발하게 연구되어 왔다[1]-[3].

종단간의 신뢰성을 보장하는 멀티캐스트를 위해서 RMTP(Reliable Multicast Transport Protocol), RMTP-II, TRAM(Tree-based Reliable Multicast Protocol)과 같은 트리 기반 방식의 프로토콜, SRM(Scalable Reliable Multicast)과 같은 NACK(Negative Acknowledgment) 기반 방식의 프로토콜, RMDP(Reliable Multicast data Distribution Protocol)와 같은 ARQ(Automatic Repeat Request)가 아닌 FEC(Forward Error Correction)방식의 프로토콜들이 제안되었다[4].

그 중 트리 기반 방식이란 멀티캐스트 사용자들을 연결하는 전송 계층의 논리적인 트리를 구성하여, 트리 계층 구조를 이용하여 오류를 복구하고, 흐름 및 혼잡제어를 수행하는 방식을 말한다[5]. 그러므로 이 방식에서는 논리적인 트리를 구성하고 관리, 유지하는 것이 중요한 쟁점이 될 수 있다. IETF(Internet Engineering Task Force)의 워킹그룹 RMT(Reliable Multicast Transport)에서는 Tree Auto-Configuration에 대한 draft[6]를 제시하고 있다. 이 draft에 포함된 ERS(Expand Ring Search)-

based scheme[7][8]에 따르면 트리를 구성하는데 있어 Bottom-up 방식을 따르고 있으며, 이에 반해 트리 구성을 Top-down방식으로 수행하는 프로토콜인 ECTP(Enhanced Communications Transport Protocol)[9]이 있다.

이 논문은 트리를 구성하는 2가지 방식에 대한 성능을 시뮬레이션을 통하여 비교, 분석하고자 한다. 이러한 비교 분석은 멀티캐스트 프로토콜에 대한 표준화가 진행되고 있는 시점에서 중요한 변수로 작용할 수 있을 것으로 본다.

II. Tree Creation 방식

본 장에서는 IETF에서 제시하고 있는 ERS-based scheme을 이용한 트리 구성 방식과 ECTP에서의 트리 구성 방식을 살펴보기로 한다.

1. Top-down 방식

ECTP에서는 기반이 되는 라우팅 프로토콜과 네트워크의 구조와는 독립적인 트리 구성을 목표로 하고 있으며, 3가지 다음과 같은 트리 노드를 가지고 있다.

- TO(Top Owner) : 논리적인 트리의 루트이며, 트리 구성을 시작하게 하고 전체적인 트리 구성을 관리한다.
- LO(Local Owner) : 수신자이며, 한 개 이상의 자식 노드를 가질 수 있고, 그들에 대한 에러 복구를 담당한다.
- LE(Leaf Entity) : LO가 아닌 순수한 수신자이며 트리의 마지막 단자라고 할 수 있다.

[그림 1]은 ECTP의 전체적인 트리 구성 과정을 보여주고 있다. 트리 구성의 시작은 TO 노드에서 CR 메시지를 멀티캐스트 하면서 시작된다. CR을 받은 LO 혹은 LE 노드는 TJ 메시지를 보냄으로써 on-tree 노드가 되기 위한 요청을 한다. TO에서는 자신의 자식 노드로 받아들일 수 있는지를 판단한 후, TC의 accept flag를 세팅한 후, 해당 LO에게 전송한다. 만약 자식 노드로 받아들일 수 없는 경우, reject flag를 세팅하여 TC 메시지를 보낸다. accept TC 메시지를 받은 LO는 "on-tree" 노드로써 동작할 수 있게되며, 자식 노드를 받아들이기 위한 HB 메시지를 멀티캐스팅 한다. 만약 어떤 LE가 여러 개의 LO로부터 HB메시지를 받는다면 그 중에

서 가장 최적의 LO를 선택하여 부모 노드로 결정한다. 최적의 부모 노드를 선택하는 메커니즘은 자신과 가장 가까운 거리의 노드를 선택하거나, 가장 최근에 받은 메시지를 보낸 노드를 선택하는 등 구현상 여러 가지가 있을 수 있다. reject TC를 받은 LE는 다른 부모 노드를 찾기 위해 재 시도한다. LE는 CC를 자신의 부모 노드로 보내고, LO 노드에서 통합된 뒤, 트리의 상위 레벨로 올려준다. 마지막으로 TO가 모든 자식 노드로부터 CC를 받게 되면 트리 구성이 종료된다.

표 1. ECTP packets

| | |
|----|-----------------------------|
| CR | Connection Creation Request |
| CC | Connection Creation Confirm |
| TJ | Tree Join Request |
| TC | Tree Join Confirm |
| HB | Heartbeat |

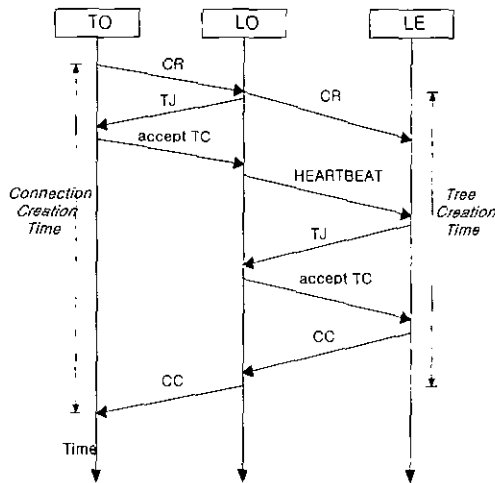


그림 1. ECTP Tree Creation

2. Bottom-up 방식

ACK(Acknowledge)트리가 수신자가 시작하는 방식으로 구성되지 않는 형태로서, 본 연구에서는 ERS를 이용하여 주변 노드를 발견하고 부모 노드를 선택하여 트리를 구성하는 방식을 의미한다.

ACK 트리는 송신자를 루트로 구성되며, 수신자들이 트리의 구성원으로 붙는다. SN(Service Node)이라는 망내의 전용 서버이거나 수신자일 수 있는 노드가 송신자와 순수 수신자사이에서 여러 복구와

제어 정보를 송신자 측으로 통합하여 전달하는 역할을 한다.

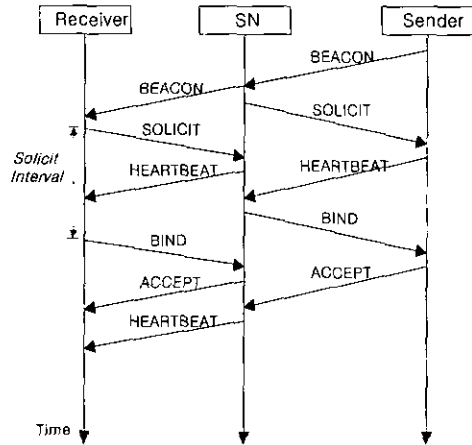


그림 2. ERS-based Tree Creation

[그림 2]는 ERS-based 트리 구성도를 보여주고 있다. 트리 구성은 송신자로부터의 BEACON 메시지에 의해 시작된다. BEACON 메시지를 수신하면 각 노드(SN, 순수 수신자)들은 송신자로부터의 자신까지의 상대적인 거리정보를 메시지내의 TTL 정보를 통해 알게된다. 자신의 이웃 노드들을 파악하기 위해 SOLICIT 메시지를 주변으로 TTL scope 멀티캐스트하고, 이에 응답하는 SN중에서 최상의 부모를 선택한다. 최상의 부모노드 선택은 송신자로부터 얼마나 그 SN이 가까운지가 가장 중요하고, 그 외에 SN의 현재 자식 노드 수나 용량, 트리 수준, IP 주소가 영향을 준다. 이렇게 최상의 부모노드를 선택하게 되면 그 SN에 BIND 메시지를 보내어 트리에 가입을 요청한다. 요청이 수락되면 "on-tree" 노드가 되며, 요청이 거절될 때에는 그 다음의 최상의 부모 노드를 선택하여 재가입을 시도한다.

표 2. ERS packets

| | |
|-----------|----------------------------------|
| BEACON | trigger tree creation |
| SOLICIT | query neighbors for metric info. |
| HEARTBEAT | Periodic keep-alive |
| BIND | request to SN to join tree |
| ACCEPT | acceptance of BIND from SN |
| REJECT | rejection of BIND from SN |

III. 성능 평가 및 분석

이 장에서는 앞에서 살펴본 2가지 프로토콜에서의 트리 구성을 분석해보고자 한다. 멀티캐스팅에 참여하는 노드의 수가 증가함에 따라 각각의 프로토콜의 트리 구성시 교환되는 메시지 수를 비교함으로써 안정성 및 확장성을 평가한다. 시뮬레이션은 SMPL[10]을 이용한 event-driven 방식을 사용한다.

1. 시뮬레이션 모델

1.1 시뮬레이션 Topology.

시뮬레이션에 사용한 네트워크의 Topology는 [그림 3]과 같다. x, y축의 값은 상대적인 거리를 나타내며, random number를 발생시켜 선택하였다. (0,0)의 위치에 TO가 있으며, 각각의 노드들은 2차원 평면에 위치하고 있다고 가정한다. 또한 전체 노드 중 LO 노드는 임의로 결정한다. ERS-based에서는 SN 노드이나 이후 LO로 통일하여 표기한다.

이 밖에 시뮬레이션을 위한 몇 가지 가정을 다음과 같이 한다.

- 노드의 90%가 “on-tree” 노드가 되었을 때를 시뮬레이션 종료 조건으로 한다.
- 트리 구성의 연속성을 위하여 TO 노드는 LO만을 자식 노드로 가진다.
- 교환되는 모든 메시지는 종류와 상관없이 크기가 같으며, 처리 시간 또한 같다고 가정한다.
- 각 메시지는 시뮬레이션에 필요한 element들만을 갖는다.
- Link Propagation Delay = 0.1로 가정한다.

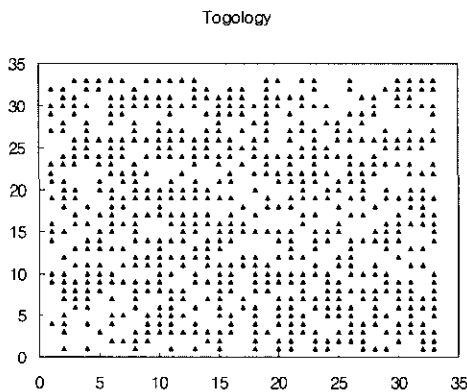


그림 3. 시뮬레이션에 사용된 Network Topology (노드 수: 1000개 일 때)

1.2 성능 평가 요소

시뮬레이션을 통한 성능 평가 시 트리 구성 프로토콜의 성능을 결정짓는 중요한 요소들은 다음과 같다.

- 노드 수
멀티캐스팅을 위한 논리적 트리 구성에 참여하려고 하는 총 노드 수에 따라 성능이 달라질 수 있으므로 총 노드 수가 결정되어야 한다.
- LO 노드의 비율
중간 노드인 LO 노드는 자식 노드를 가질 수 있고 자신의 자식 노드에 대한 책임을 가지고 있으므로 성능에 결정적인 역할을 할 수 있다.
- 최대 자식 노드 수
LO 노드가 가질 수 있는 최대 자식 노드의 수는 트리 구성 시 트리의 depth를 결정짓는 중요한 요소이다.
- Reserved LO 노드의 수
LO 노드는 또 다른 LO 혹은 LE 노드를 자식 노드로 가질 수 있다. 이때 최대 자식 노드 수 중 일부는 반드시 LO 노드가 되도록 미리 설정을 해 둘 수 있다. 이것은 트리가 계속 형성될 수 있도록 하기 위한 연결 고리를 만드는 것으로 이 값에 따라 “on-tree” 노드가 되기 위한 시도에서 거절(reject) 당할 수 있으므로 성능에 영향을 끼칠 수 있다.

그리고 다음과 같은 성능 평가 요소들이 고려되어야 한다.

- 총 메시지 수 비교
트리 구성시 교환되는 총 메시지의 수는 전체 네트워크의 로드(load)에 영향을 끼치게 되며, 또한 각각의 노드에서 처리해야 할 로드의 양과 관계가 있다.
- 트리 구성 지연(delay) 시간 비교
트리를 구성의 시작을 초기화시키는 CR 혹은 Beacon 메시지의 멀티캐스팅 시점부터 트리 구성을 마치는 데까지 걸리는 지연 시간을 측정한다. Bottom-up approach는 트리 구성의 종료를 명시적으로 나타내기 어려우므로, 두 프로토콜을 비교하는데 동등한 조건을 적용하기 위하여 전체 노드 중 임의의 수만큼의 노드들이 “on-tree” 노드가 되는 시점까지를 측정한다.
- Tree Depth 비교

트리의 Depth는 멀티캐스팅의 논리적 트리 구성 과정뿐만 아니라, 이후 트리를 관리하고 유지하

는데 큰 영향력을 끼치는 요소이므로 고려되어야 할 중요한 요소이다.

2. 시뮬레이션 결과

[그림 4(a)]는 노드 수가 증가함에 따라 트리 구성을 위해 교환되는 총 메시지의 수를 비교한 것이고, (b)는 트리 구성에 소요되는 시간을 비교한 것이다. 총 메시지수의 경우 ECTP가 ERS-based 방식에 비하여 훨씬 적은 메시지를 발생하는 시키나, 트리 구성 지연 시간은 더 큰 것을 알 수 있다. 특히, 노드 수 400에서 두 프로토콜 모두 노드 수 200인 경우 보다 더 적은 시간이 걸리는 것으로 나타났다.

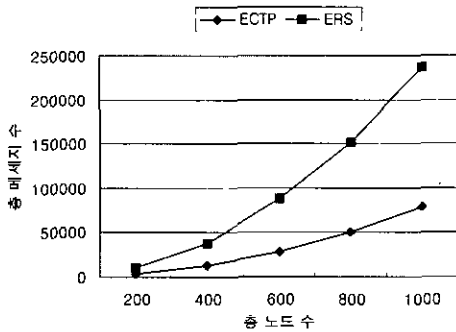


그림 4. (a) 노드 수에 따른 메시지 수

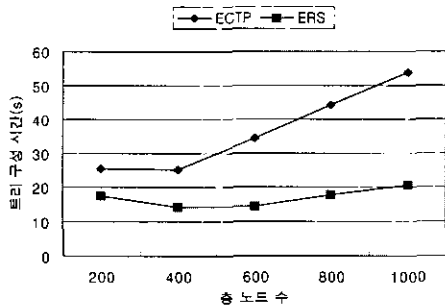


그림 4. (b) 노드 수에 따른 트리 구성 지연 시간

[그림 5]에서는 전체 노드 수에서 LO 노드수가 차지하는 비율에 따른 변화를 비교하였다. 그 결과 LO 노드가 전체 노드수의 20%가량을 차지할 때, 트리 구성 지연이 가장 적은 것으로 나타났다. 그러나 시뮬레이션 결과, 교환되는 총 메시지 수는 LO 노드 수가 증가함에 따라 더 많은 메시지가 교환되는 증가 그래프를 보였다.

[그림 6(a),(b)]는 최대 자식 노드 수의 증가에 따른 시뮬레이션 결과이다. 여기에서 최대 자식 노드

수 5와6의 경우는 트리를 더 이상 구성하지 못하고 중간에 종료된 중간결과 값이다. 이것은 최대 자식 노드수의 제한으로 "on-tree" 노드가 될 수 없는 경우가 발생하기 때문이다.

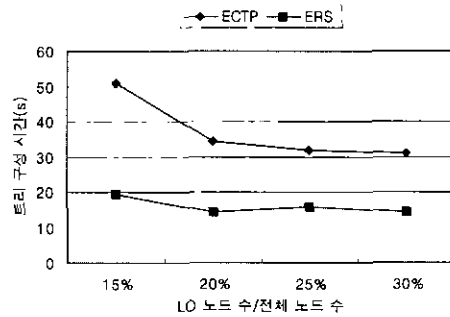


그림 5. LO 비율에 따른 트리 구성 지연 시간

[그림 6(b)]에서는 최대 자식 노드수가 10 이하인 경우 트리 구성 지연 시간이 큰 차이를 보이고 있는 반면, 10 이상에서는 상대적으로 감소하는 폭이 적은 것을 보여주고 있다.

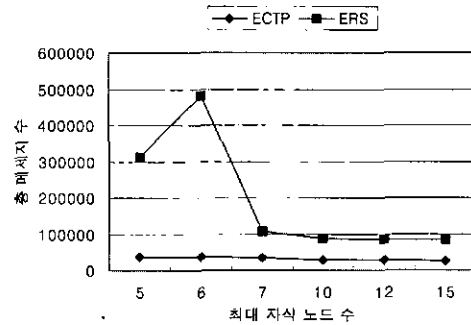


그림 6. (a) 최대 자식 노드 수에 따른 총 메시지 수

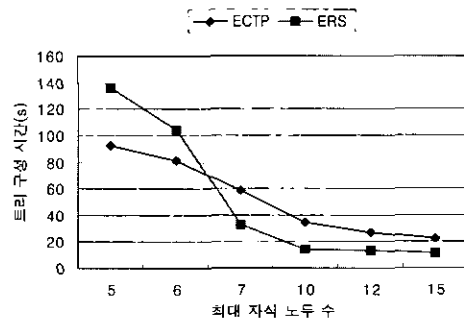


그림 6. (b) 최대 자식 노드 수에 따른 트리 구성 지연 시간

시뮬레이션 결과, Tree Depth는 ECTP가 ERS-based에 비하여 1-2정도 적은 것을 알 수 있었다. 특히, [표3,4]에서는 Reserved LO 수와 최대 자식 노드 수에 따른 트리 Depth를 보여준다. 두 가지 경우 모두 조금씩 감소하는 양상을 보이기는 하나, 큰 차이는 없는 것을 알 수 있다.

표 3. Reserved LO 수에 따른 트리 Depth (최대 자식 노드 수 = 10)

| Reserved LO 수 | 1 | 2 | 3 | 4 | 5 |
|---------------|---|---|---|---|---|
| ECTP | 7 | 6 | 6 | 6 | 6 |
| ERS | 8 | 8 | 8 | 7 | 7 |

표 4. 최대 자식 노드 수에 따른 트리 Depth (Reserver LO = 2)

| 최대 자식 노드 수 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|------------|-----|-----|---|---|---|----|----|
| ECTP | > 9 | > 7 | 8 | 7 | 6 | 6 | 6 |
| ERS | > 7 | > 9 | 8 | 8 | 8 | 8 | 8 |

IV. 결론 및 향후 연구 계획

지금까지 트리 기반 멀티캐스팅 프로토콜의 트리를 구성하는 두 가지 방식을 살펴보았다. ECTP의 경우 Top-down 방식을 따르므로 비교적 간단하고 명료한 과정을 거쳐 트리를 구성한다. 반면 Bottom-up 방식을 따르는 ERS-based의 경우는 조금 복잡한 과정을 거쳐 트리 구성시 교환되는 메시지의 수는 상대적으로 많지만, 하위 수준에서 병렬적으로 동시에 여러 노드가 트리 구성에 참여하므로 트리 구성 지연 시간에서 효과적이다. 이것은 트리 구성에 참여하는 총 노드수가 많아질수록 뚜렷한 차이를 보인다.

또한 여러 가지 다른 요소들이 영향을 끼치겠으나, LO 노드의 비율은 전체 노드수의 20%정도를 유지할 때, 트리 구성 지연 시간에 관한 성능이 높을 것으로 기대되며, 최대 자식 노드의 수로 인한 변화는 최대 자식 노드의 수가 10이하인 경우에 큰 폭의 변화를 나타낸다.

그러므로 위의 여러 가지 사항들을 고려하여 멀티캐스트 그룹에 참여하고자 하는 노드들의 토폴로지에 적합한 트리 구성 프로토콜을 선택하여야 하며, 또한 이것은 트리 구성에 한정된 결과이므로 트

리를 유지, 관리하는 문제 및 실제 멀티캐스팅에 대한 결과와 더불어 고려되어야 할 것이다.

참고 문헌

- [1] Sanjoy Paul, "Multicasting On The Internet And Its Applications", Kluwer Academic Publishers, 1998.
- [2] Katia Obraczka, "Multicast Transport Protocols: A Survey and Taxonomy," IEEE Communication Magazine, Jan. 1998
- [3] 고석주, 강신각, "멀티캐스트 신뢰전송 기술 및 표준화 동향", ETRI 주간기술동향, 제 00-16호, pp. 16-33, 2000, 4.
- [4] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, M. Luby, "The Reliable Multicast Design Space for Bulk Data Transfer," RFC 2887, August 2000.
- [5] B. Whetten, D. Chiu, S. Paul, "TRACK ARCHITECTURE A SCALEABLE REAL-TIME RELIABLE MULTICAST PROTOCOL," Internet Draft draft-ietf-track-arch-00.txt, July 2000.
- [6] IETF, "Reliable Multicast Transport Building Block: Tree Auto-Configuration", draft-ietf-rmt-building-blocks-01.txt, 2000. 7.
- [7] Dah-Ming Chiu, Stephen Hurst, "TRAM: A Tree-based Reliable Multicast Protocol", 1998, 7.
- [8] R. Yavakar, J. Griffioen, M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Application," University of Kentucky, 1995
- [9] ITU-T Q.13/7 X.ectp, "Enhanced Communication Transport Protocol", Working in Progress
- [10] M. H. MacDougall, "Simulating Computer Systems, Techniques and Tools," MIT Press, 1987

김 현 숙(HyunSook Kim)

정회원



1997년 2월 : 덕성여자대학교
진산학과 졸업
1999년 8월 : 연세대학교
컴퓨터과학과 석사
1999년 9월~현재 : 연세대학교
컴퓨터과학과 박사과정

<주관심 분야> Next Generation Internet, Multi-casting, Wireless Communication

정 찬 윤(ChanYoon Jung)

정회원



1999년 2월 : 연세대학교
컴퓨터과학과 졸업
2001년 2월 : 연세대학교
컴퓨터과학과 석사
<주관심 분야> IP Multicast,
QoS, Mobile IP, MPLS

이 용(Yong Lee)

정회원



1992년 2월 : 덕성여자대학교
진산학과 졸업
1996년 8월 : 연세대학교
컴퓨터과학과 석사
2001년 2월 : 연세대학교
컴퓨터과학과 박사
~현재 : 한국정보보호센터

<주관심 분야> Next Generation Internet, Multi-casting, IMT-2000

고 석 주(SeokJoo Koh)

정회원

현재 한국전자통신연구원 표준연구센터
한국통신학회 학회지 제 17권 9호 참조

송 주 석(JooSeok Song)

정회원

현재 연세대학교 컴퓨터과학과 교수
한국통신학회 논문지 제 22권 10호 참조