# 대규모 신뢰적 멀티캐스트 세션을 위한 적응형 트리 기반 복구 기법

# (Adaptive Tree-based Recovery Scheme for Large-Scale Reliable Multicast Sessions)

윤 원 용 † 이 동 만 ††

(Wonyong Yoon) (Dongman Lee)

**요 약** 통신의 규모가 사용자 수와 지리적 범위 두 가지 측면에서 커짐에 따라 신뢰적인 멀티캐스트 프로토콜의 implosion 및 exposure 문제는 더욱 심각해진다. 본 논문은 트리 기반 신뢰적 멀티캐스트를 위한 효율적이고 확장성 있는 손실 복구 기법을 제안한다. 먼저 에러 비트맵 정보를 통하여 멀티캐스트 라우팅 트리와 유사한 논리적 트리를 구성함으로써 멀티캐스트 라우팅 트리에서 상위에 위치하는 수신자들이 재전송을 요청한 수신자의 신뢰성을 책임지도록 하고 또한 효율적인 복구를 위해 구성된 트리 상에 독립된 멀티캐스트 주소를 가지는 지역 그룹을 형성한다. 논리적 트리는 세션 멤버쉽이나 멀티캐스트 경로의 변화에 따라 적응적으로 재구성되는데 이는 멀티캐스트 세션의 진행 동안 논리적 트리와 멀티캐스트 라우팅 트리 사이에 불일치를 최소화함으로써 멤버쉽과 경로가 변하는 상황에서도 implosion과 exposure를 감소시키는 강점을 지닌다. 제안한 기법과 정적 트리기반의 신뢰적 멀티캐스트 프로토콜과의 시뮬레이션을 통한 비교는 세션의 크기가 증가할수록 제안한 적응형 트리 기반의 복구방식이 더욱 효율적임을 보여준다.

***Abstract*** As the scale of a network becomes larger in terms of both the number of users and geographic span, reliable multicast protocols suffer more severely from implosion and exposure. In this paper, we propose a scalable, efficient recovery scheme for tree-based reliable multicast transport protocols. The scheme first constructs a logical tree of receivers as close to a multicast routing tree as possible by using error bitmap information. This ensures that the receivers residing at the upper level in a multicast routing tree than ones requesting retransmissions be appointed as parents in a corresponding logical tree. Our scheme also forms local groups with separate multicast addresses in the tree for efficient recovery. In our scheme, the logical tree is adaptively reconfigured as the session membership or the multicast route changes. This enables minimal discrepancy between a logical tree and a multicast routing tree during a given session and helps minimize the implosion and exposure problems even in the presence of membership and route dynamics. We compare our scheme with a static tree-based reliable multicast protocol. Results of the simulation show that our adaptive tree-based recovery scheme outperforms the compared protocol in terms of the implosion and exposure as the session size increases.

## 1. 서 론

신뢰적인 멀티캐스트는 소스로부터의 각각의 패킷이 모든 수신자들에게 에러없이 전달되는 것을 보장해준다. 인터넷과 같이 사용자와 지리적 위치의 측면에서 통신의 규모가 커짐에 따라서 신뢰적인 멀티캐스트 프로토콜은 *implosion* 및 *exposure* [1, 2, 3, 4] 의 두가지 본질적인 문제에 직면하게 되었다. implosion은 패킷의 수신 또는 손실 여부를 알리기 위해 수신측에서 보내지는 ACK 이나 NACK 피드백이 소스 호스트와 네트워크에 집중되는 현상을 일컫는다. 이것은 소스와 네트워

† 학생회원 : 한국정보통신대학원대학교 공학부
wyyoon@icu.ac.kr
†† 종신회원 : 한국정보통신대학원대학교 공학부 교수
dlee@icu.ac.kr
논문접수 : 1999년 9월 17일
심사완료 : 2000년 5월 24일

크를 전체 멀티캐스트 세션의 병목으로 만듦으로써 결과적으로 전체 세션의 성능을 저하시키게 된다. exposure는 손실 복구를 위해서 재전송된 패킷이 중복적으로 보내져서 이전에 받았던 수신자에게도 패킷이 전달되는 것을 말한다. 이것은 불필요한 프로세싱 오버헤드를 수신자에게 주며 네트웍 대역폭의 남용을 초래한다.

신뢰적 멀티캐스트의 확장성(scalability)를 높이기 위한 한가지 방법은 분산 복구(distributed recovery)인데 이 방법은 에러 복구를 소스뿐만 아니라 다른 수신자도 담당할 수 있도록 하는 것이다 [5]. 이것은 implosion을 줄이는데 도움을 준다. 또 다른 방법은 지역 복구(local recovery)인데 손실이 발생한 곳 부근에서 지역적으로 손실을 복구함으로써 재전송을 위한 대역폭과 exposure를 감소시킨다[5, 6, 7, 8]. 트리 기반 프로토콜은 트리 구조 상에 자연스럽게 분산복구와 지역적 복구를 결합시킨다[9, 10]. 트리 기반 프로토콜은 에러복구의 책임이 분산되게 수신자들의 계층적인 트리 - 이하 논리적 트리(logical tree) - 를 만든다. 수신자는 트리의 자식 노드의 피드백을 처리하고 그들에 대한 재전송을 담당한다. 즉 부모노드는 자식 노드들의 신뢰성을 책임진다. 트리 기반 프로토콜은 처리율(throughput) 측면에서 가장 확장성이 뛰어난 것으로 알려져 있다 [11].

트리 기반 프로토콜에서 논리적 트리는 재전송을 요구한 노드들이 멀티캐스트 라우팅 트리상에서 자신보다 위에 있는 다른 수신자를 부모로 가지도록 구성되는 것이 중요하다. 그러나 멤버들이 세션에 참여하거나 탈퇴하면서 또는 세션 진행 동안에 멀티캐스트 경로가 바뀌면서 그림 1에서와 같이 물리적인 부모-자식관계가 논리적 트리에서 보존되지 못하는 전위(inversion) 문제가 발생할 수 있다. 논리적 트리에서 논리적인 자식 C는 피드백을 논리적 부모인 P에게 보내지만 P는 멀티캐스트 라우팅 트리에서는 C의 물리적 자식이므로 문제가 발생한다. C에서 패킷이 손실되었을 때, P로 보내진 패킷들이 물리적으로는 C를 경유하도록 보내지기 때문에 결코 P에 도달되지 못한다. 즉 P는 C의 신뢰성을 책임

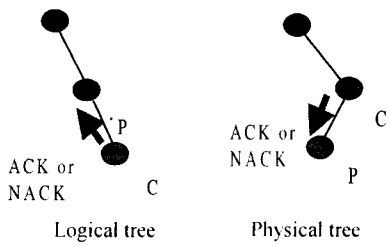


그림 1 전위(inversion)

질 수가 없다. 따라서, 논리적 트리는 세션 멤버쉽이나 멀티캐스트 경로가 변하는 환경에서도 전위 문제를 최소화하도록 구성되어야 한다.

트리의 상위에서 패킷의 손실이 발생한 경우 유니캐스트에 의한 복구는 트리의 리프 노드까지 되풀이된다. 유니캐스트 복구의 반복으로 인한 지연(cascaded delay)은 만족스럽지 못하며, 이러한 지연을 줄이기 위한 시도는 자칫 exposure를 악화시킬 가능성이 높다. 논리적 트리가 갖추어야 할 두번째 요구사항는 지연과 exposure간에 적절한 조율을 할 수 있도록 구성되어야 하는 것이다.

본 논문은 위 두 가지 요구사항을 만족시키는 적응형(adaptive) 논리적 트리를 이용한 확장성 있고 효율적인 복구 기법을 제안한다. 제안한 방법은 먼저 수신자들의 패킷 전달 상태 정보를 이용하여 멀티캐스트 라우팅 트리와 가능한 유사한 논리적 트리를 구성한다. 다음, exposure를 최소화시키는 효율적인 복구를 위해, 논리적 트리 구조 상에 독립된 멀티캐스트 주소를 가지는 지역 그룹(local group)을 형성한다. 세션 멤버쉽이나 멀티캐스트 경로의 변화에 따라 논리적트리는 적응적으로 재구성되면서 최적의 부모-자식 관계를 유지한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 기술한다. 3장에서 적응형 논리적 트리를 구성하기 위한 알고리즘의 기술과 함께 구성된 논리적 트리가 세션 멤버쉽이나 멀티캐스트 경로의 변화에 따라 어떻게 적응해 가는 지를 보여준다. 또한 exposure와 지연사이의 조율을 위한 지역 그룹 구성 방법을 기술한다. 4장에서 전위 문제의 정량화를 제시하고, 제안한 방법의 성능 측정 결과를 정적 트리 기반의 신뢰적 멀티캐스트 프로토콜인 TMTP(Tree-based Multicast Transport Protoco) [9]과 비교한다. 5장은 결론을 맺는다.

## 2. 관련연구

대표적인 트리 기반 신뢰적 멀티캐스트 프로토콜은 TMTP [9]와 RMTP [10]이다. 여기서는 이들 프로토콜들의 논리적 트리 구성 방법을 중심으로 분석해 본다. TMTP [9]는 논리적 트리를 구성하기 위해 expanding ring search를 사용한다. 새로운 멤버(TMTP에서는 domain manager라는 용어를 쓴다)가 멀티캐스트 그룹에 가입할 때 TTL 값을 가진 제어 패킷을 그 그룹으로 멀티캐스트한다. 기존 멤버로부터의 응답이 없으면 TTL 값을 증가하여 다시 멀티캐스트한다. 응답을 받을 때까지 이 과정을 반복하며 이 응답을 보낸 멤버가 새로운 노드의 부모가 된다. TMTP의 TTL을 이용한 트

리 구성 방법은 TTL의 무방향성(lack-of-direction) 결점 때문에 전위 문제(inversion: 라우팅 트리의 부모-자식 관계가 논리적 트리에 제대로 반영되지 못하는 상황)를 배제할 수 없다. 더욱 심각한 것은 논리적 트리가 멤버의 가입/탈퇴 순서에 좌우된다는 점이다. Limited scope multicast를 이용한 재전송 역시 TTL 기법을 이용하므로 트리의 하부 뿐만 아니라 상부로도 재전송될 수 있어서 exposure 문제가 발생 가능하다.

RMTP [10]는 멀티캐스트 세션이 있을 때 DR (Designated Receivers)이 정적으로 정해진다고 가정한다. 송신자와 모든 DR들은 똑같은 TTL 값을 가지는 제어 패킷을 주기적으로 멀티캐스트한다. 새로운 멤버는 이중 가장 큰 값의 TTL을 가지는 DR을 부모로 선택함으로써 논리적 트리를 구성해간다. RMTP의 트리 구성 방법은 주기적으로 전파되는 제어 패킷이 sub-tree multicasting을 이용하여 라우팅 트리의 하부로만 전송되게 하여 전위문제는 발생하지 않는다. 그렇지만 이것은 모든 멀티캐스트 라우터에 sub-tree multicasting이 채택되어져야 가능한 일이다.

현재 RMTP의 DR는 Tracer라는 도구를 이용하여 동적으로 선택되어질 수 있다 [16]. Tracer는 IGMP MTRACE 기능을 이용하여 송신자까지의 경로를 추적할 수 있다. 경로의 통고(path advertisement)와 응답을 교환하여 적절한 부모를 선택할 수 있다. 그러나 expanding ring search를 이용한 주기적인 경로 통보는 그 오버헤드가 크다. 반면 제안한 기법은 positive acknowledgment 패킷에 같이 실리는 에러 비트맵 정보를 이용하므로 정상적인 동작 시 오버헤드는 패킷 크기의 증가 밖에 없다. 물론 멤버쉽이나 경로가 변경될 경우 부가적인 제어 메시지의 교환은 필요하다.

## 3. 제안한 기법

### 3.1 논리적 트리 구성

3.1.1 에러 비트맵(Error bitmap)

제안한 기법은 IP 멀티캐스팅 [12]의 fate sharing property를 이용함으로써 라우팅 트리의 부모-자식 관계를 최대한 반영하는 논리적 트리를 구성한다. 이를 위해 멀티캐스트 세션의 각 수신자들은 에러 비트맵 정보를 유지하고 논리적 트리 상의 부모에게 피드백한다. 에러 비트맵은 번호 S와 비트맵 B으로 되어 있다. 비트맵의 각 비트는 해당 패킷을 수신하였으면 1로, (처음의 전송이) 실패하였으면 0으로 설정한다. 가령 S=5, 비트맵 11010은 이 수신자가 순서번호 5,6,8의 패킷들을 성공적으로 수신하였음을 나타낸다. 각 수신자는 논리적 트리상의 부모에게 자신의 에러 비트맵 정보를 알려 주어야 한다. 노드의 에러비트맵의 한 비트가 1이면, IP 멀티캐스팅의 fate sharing 특성에 의해 그 부모의 해당 비트가 1일 가능성이 높다. 소스의 에러 비트맵은 모두 1로 이루어져 있음에 주의한다. 다음 절에서 세션 멤버들의 에러 비트맵 정보를 이용하여 멀티캐스트 라우팅 트리와 유사한 논리적 트리를 구성하는 방법을 제시한다.

3.1.2 논리적 트리

논리적 트리 구성 알고리즘의 설명을 위해 먼저 다음과 같이 관계 연산자와 동작(operation)을 정의한다.

*Definition 1(Child)*: 두 노드 $N_i$, $N_j$에 대해, 모든 k = 1, 2, ... $l$ 에 대해 $B_k(N_i) \leq B_k(N_j)$ 이면 $N_i \subset N_j$ 라고 한다.

*Definition 2(Parent)*: 노드 $N_i$, $N_j$에 대해, 모든 k = 1, 2, ... $l$ 에 대해 $B_k(N_i) \geq B_k(N_j)$ 이면 $N_i \supset N_j$ 라고 한다.

*Definition 3(Adoption)*: $N_j$가 논리적 트리에서 $N_i$의 자식으로 되면 $N_i \leftarrow N_j$ 라고 표기하고 $N_i$가 $N_j$를 수용(adopt)한다고 말한다.

두 관계 연산자에 대해 다음의 정리가 성립한다.

**Theorem 3.1**: $N_i \subset N_j$ 이면 $N_i \supset N_j$ 이다.

*Proof*: $N_i \subset N_j$ 이면 모든 k = 1, 2, ... $l$ 에 대해 $B_k(N_i) \leq B_k(N_j)$ 이므로, *Definition 2*에 의해 $N_i \supset N_j$ 이다. □

그림 2에 새로운 수신자가 참여하거나 기존 수신자가 탈퇴할 때 논리적 트리를 구성하는 알고리즘이 기술되어 있다. EB는 에러 비트맵을, P(X)는 노드 X의 부모를 나타낸다. 멤버 가입 알고리즘은 다음과 같다. 신뢰적 멀티캐스트 세션에 들어온 새로운 멤버는 일단 소스(source) 즉 송신자에게 자신의 IP 주소를 알림으로써 소스의 임시 자식(tentative child)이 된다. 새로운 멤버는 멀티캐스트 그룹의 주소와 소스의 IP 주소를 안다고 가정한다. 소스는 새 멤버의 에러 비트맵 정보를 받으면 그것을 다른 자식들과 비교하여 새 멤버 N이 다른 자식들과 어떠한 관계를 가지는 지 검사한다. 4 가지 경우가

```
Receive a message M from children;
if (M is EB from a tentative child TC) {
   if (∃ a child C s.t. C ⊂ TC) {
      P(TC) := S;
      P(C):= TC;
   } else if (∃ a child C s.t. C ⊃ TC)
      Redirect EB to C;
   else
      P(TC) := S;
} else if (M is a leave from a child C) {
   for ∀ D who is a child of C, P(D) := S;
   Delete C;
} else if (M is a leave message from TC)
   Delete TC;
```

(a) Algorithm at the sender S

```
Receive a message M from its parent
and children;
if (M is EB about a TC from parent) {
   if (∃ a child C s.t. C ⊂ TC) {
      P(TC) :=N;
      P(C):= TC;
   } else if (∃ a child C s.t. C ⊃ TC)
      Redirect EB to C;
   else
      P(TC) := N;
} else if (M is a leave from a child C) {
   for ∀ D who is a child of C, P(D) := N;
   Delete C;
}
```

(b) Algorithm at intermediate node N

그림 2 가입/탈퇴 알고리즘

가능하다.

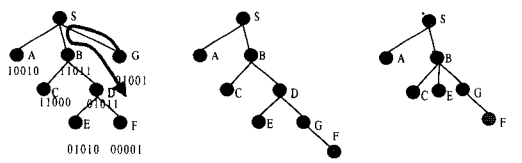
Case i) N이 어떤 자식과도 어떠한 관계를 가지지 않는다면 소스는 N을 정식 자식(regular child)으로 받아들인다. 즉 S ← N

Case ii) N ⊃ C(소스의 자식 중 하나) 이면 S ← N 이고 N ← C

Case iii) N ⊂ C 이면 S는 N의 에러 비트맵과 IP 주소를 C에게 넘겨준다. 다음 노드 C에서 동일한 판단 과정이 반복되며 이러한 반복은 적절한 N의 부모를 찾을 때까지 트리의 아래를 따라 계속된다.

Case iv) B(N) = B(C) 이면 C ← N

멤버 탈퇴 알고리즘은 간단하다. 세션을 탈퇴하려는 참여자는 그 사실을 부모에 알린다. 부모는 탈퇴하려는 참여자의 직속 자식들을 자신의 자식으로 받아들이고 그 자식들에게 부모정보를 자기로 변경하라고 통보한다. 실패(failure)로 인해 미처 알리지 못하고 트리에서 나가는 경우가 가능하다. 이때 그 자식들은 트리에서 분리되게 된다. 자식들은 이 사실을 알게 되면 소스의 임시 자식으로 다시 트리에 붙는다. 그 다음 가입 알고리즘과 동일하게 적절한 부모를 찾아갈 수 있다.




(a) A logical tree with G as a tentative child of the sender

(b) The logical tree with G as a child of the node D

(c) The logical tree after the leave of D

그림 3 가입/탈퇴 알고리즘의 동작 예

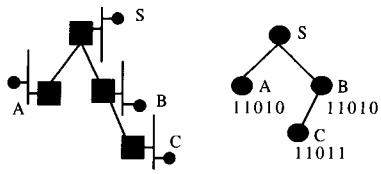
그림 3의 (a)에 새로운 멤버 G가 소스 S의 임시 자식으로 트리에 가입된 모습을 볼 수 있으며 에러 비트맵 정보를 이용하여 G가 적절한 위치 즉 D의 자식과 F의 부모로서 트리에 정식으로 가입됨을 (b)에서 보이고 있다. 그림 (c)는 멤버 D가 그룹에서 탈퇴할 때 탈퇴 알고리즘을 적용한 뒤의 모습을 나타낸다.

3.1.3 관계 이상(relation anomaly)

다음 정리는 에러 비트맵 정보와 물리적인 라우팅 트리 사이에 존재하는 이상(anomaly)를 밝힌다.

**Theorem 3.2:** Ni ⊂ Nj 는 멀티캐스트 라우팅 트리에서 Ni가 Nj의 자식임을 보장하지 않는다.

*Proof:* S를 송신자로 하는 멀티캐스트 세션의 네트워크 토폴로지가 그림 4 (a)와 같다고 하자. 사각형은 네트워크 라우터를 원은 호스트를 나타내는데 이 토폴로지는 그림 4 (b)의 트리와 같이 간단히 표현할 수 있다.




(a) Network topology (b) Abstracted physical tree

그림 4 멀티캐스트 라우팅 트리에서의 ⊂ 관계 이상

그림 4 (b) 에서 C는 멀티캐스트 라우팅 트리 상에서 C의 자식이 아니지만 B⊂C 관계가 성립한다. 이것은 물리적 부모인 B의 지역 네트워크 인터페이스에서 손실을 겪지만 C에서는 그러한 손실이 발생하지 않은 경우에 가능하다. 이러한 이상(anomalous) 현상을 related anomaly라 부른다.

수신자 A와 C는 A ⊂ C 관계를 보이지만 A는 멀티캐스트 라우팅 트리 상에서 C의 자식이 아니다. 이것은 무관한 손실(unrelated loss)때문인데 이러한 이상현상을 unrelated anomaly라고 부른다. 그림 4의 예에서 A와 C는 우연히 첫 네 개의 비트에 대해 같은 값을 가지고 있고 A는 다섯 번째 패킷의 손실을 겪은 반면 C는 그 패킷을 수신한 상태이다. 이 예와 같이 unrelated anomaly는 두 노드가 멀티캐스트 패킷 전송 경로를 완전히 공유하지 하지 못하는 경우 발생 가능하다. 하지만 B와 C처럼 한 노드의 경로가 다른 노드의 경로에 완전히 포함되면 unrelated anomaly는 발생할 수 없다. 대신 related anomaly는 발생할 수 있다. □

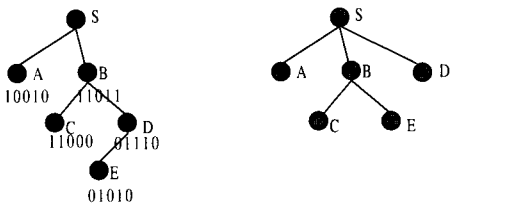
이러한 이상 현상의 존재는 주목할 만 하지만 그 영향은 크지 않다고 할 수 있다. 왜냐하면 related anomaly는 로컬 라우터에서 종단 호스트까지의 손실은 거의 발생하지 않기 때문에[13] 무시할 수 있는 것이고, unrelated anomaly는 다음과 같이 회피하거나 감소시킬 수 있기 때문이다.

**Theorem 3.3:** 에러 비트맵 정보의 크기를 증가함으로써 unrelated anomaly를 회피하거나 감소시킬 수 있다.

*Proof:* 멀티캐스트 라우팅 트리의 두 노드가 unrelated anomaly를 겪을 확율을 $P_{mis}$라 하자. $P_{mis}$는 4.2 절의 수식 (3)과 같이 구해진다. 수식 (3) $K_B$는 에러 비트맵 정보의 크기를 나타낸다. $K_B$를 무한대로 증가시키면 $P_{mis}$는 0으로 수렴하고 $K_B$가 증가함에 따라 $P_{mis}$는 단조 감소하므로, 에러 비트맵의 정보를 증가함으로써 unrelated anomaly를 회피하거나 적어도 그 가능성을 감소시킬 수 있다. □

### 3.2 논리적 트리의 적응적 재구성

라우팅 트리를 잘못 추정하여 논리적 트리와 라우팅 트리의 괴리가 발생하는 것은 드물지만 완전히 배제할 수는 없다. 또한 네트워크 경로가 변하면 그에 따라 두 트리간의 괴리가 생기고 전위현상으로 인한 피해를 보게 된다. 이 때문에 논리적 트리를 재구성할 필요가 있는데 주기적으로 피드백 되는 에러 비트맵 정보를 이용한다. 그 주기는 시간적이거나 공간적(spatial)일 수 있다. 즉 시간적인 주기로 에러 비트맵 정보가 피드백되거나, 비트맵의 크기가 최대 크기 $K_B$(4.2절에서 자세히 설명됨)에 이를 때마다 피드백될 수 도 있다.




(a) A logical tree with anomaly (b) A logical tree with D as a tentative child of the sender

그림 5 논리적 트리 유지

참여자 T가 자신의 자식 C와 그관계가 더 이상 성립하지 않음을 알게 되면 T는 자신의 부모 P에게 C의 새로운 위치를 찾아달라고 요청한다. C와 그관계가 성립하는 최초의 조상 N을 발견할 때까지 논리적 트리를 따라 상위로 같은 과정이 반복된다. 그 순간 N에서부터 트리 구성시 가입 알고리즘과 똑같은 과정이 반복되어 C의 정확한 위치를 찾을 수 있다. 그림 5 (a)에서 잘못된 추정 또는 네트워크 경로의 변화로 D의 위치가 더 이상 유효하지 않을 때 (b)와 같이 소스 S의 자식으로 채택된다.

제안한 논리적 트리 구성/재구성 알고리즘은 다음과 같은 두 가지 중요한 특성을 지닌다.

*Property 1*: 가입/탈퇴 알고리즘은 동적 멤버쉽의 순서에 완전히 독립적이다.

*Property 2*: 논리적 트리가 경로 변화에 적응하여 재구성된다.

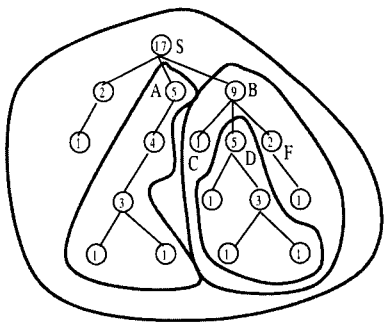
두 가지 특성은 논리적 트리와 멀티캐스트 라우팅 트리 간의 괴리를 최소화하는데 기여하는데 특히 멤버쉽 변화나 경로 변화 시 전위 문제를 최소한으로 피할 수 있게 한다.

### 3.3 지역 그룹(Local group) 조직

제안된 신뢰성 지원 기법은 논리적 트리 구성/재구성 알고리즘에 의해 트리 구조에 지역그룹(local group)을 조직함으로써 트리 구조의 단점인 반복 지연(cascaded delay)를 줄일 수 있게 하는데 이를 위해 논리적 트리의 각 노드는 weight 값을 유지한다. 노드 N의 weight, W(N)은 다음과 같이 정의된다.

$$W(N) = \sum_{i=1}^{Nc} W(Ci) + 1 \qquad (1)$$

$Nc$는 노드 N의 자식 수를, $Ci$는 각 자식을 의미한다. Weight 값이 할당된 논리적 트리의 예가 그림의 6 (a)에 있다. 참여자가 가입하고 탈퇴함에 따라, 영향을 받은 노드들은 weight 값을 바꾸고 변경 사실을 부모에게 알린다. 그 부모는 다시 자신의 weight 값을 바꾸고 바뀐 값을 재계산을 일으킨 자식 외의 모든 다른 자식에게 알린다. 이 weight값을 이용하여 논리적 트리 상에 지역그룹을 형성하는 알고리즘이 그림 6 (b)에 제시되어 있다. W(N)은 노드 N의 weight 값을 $T_h$는 주어진 threshold 값이다. P(N)은 노드 N의 직계 부모가 아니라 노드 N을 둘러싸는 지역 그룹 중 가장 작은 그룹의 루트 노드를 나타낸다. split() 프로시저에서 split된 노드 N은 새로운 멀티캐스트 주소를 할당 받고 그 주소




(a) An illustration of the algorithm($T_h$=4)

```
when (W(N) or W(P(N)) changes)
{
  if (T_h ≤ W(N) ≤ W(P(N))-T_h)
    if (N is not split) split(N);
  else
    if (N is split) merge(N);
}
```

(b) The local group organization algorithm

그림 6 지역 그룹 조직

를 직속 자식에게 알린다. 통보 과정은 트리의 하부로 퍼져 나간다. 이러한 통보 과정을 통해 split된 노드의 자손들은 새로운 그룹의 멀티캐스트 주소와 split된 노

드의 주소 및 weight 값을 알 수 있게 된다. merge() 프로시저는 반대의 동작을 수행한다.

각 지역 그룹에 독립적인 멀티캐스트 주소를 할당하는 것은 멀티캐스팅을 이용함으로써 복구 시간과 exposure를 동시에 줄이기 위함이다. 네트워크 레벨의 서브트리(sub-tree) 멀티캐스팅도 중복된 패킷을 완전히 배제하지 못한다는 점을 주목해야 한다. 지역 그룹 조직 알고리즘의 원리는 지역 그룹은 독립된 멀티캐스트 주소를 할당받기에 충분한 멤버를 포함하여야 하고 자신(지역 그룹 자신)을 포함하는 다른 지역 그룹에 포함되기에 충분할 정도로 상대적으로 작은 크기여야 한다는 것이다.

멀티캐스트 세션의 각 참여자들은 패킷 손실을 감지하면 논리적 트리 상의 부모로 negative acknowledgment를 보낸다. 이때 타이머를 구동하는데 만일 그 패킷을 재전송 받지 못한 채 타이머가 종료되면 같은 과정이 반복된다. Negative acknowledgment를 받은 부모는 자식의 상태 정보를 찾아보아 유니캐스트 재전송을 할 것인지 멀티캐스트 재전송을 할 것인지 결정한다. 그 자식에 대해 멀티캐스트 주소 정보가 존재하면 그 주소로 멀티캐스트하고 그렇지 않으면 유니캐스트 주소로 유니캐스트 재전송을 한다.

각 노드는 세 가지 종류의 제어 정보를 부모에게 피드백한다. positive acknowledgments, negative acknowledgments, 에러 비트맵이 그것이다. 자식으로부터 positive acknowledgment를 받으면 자신의 모든 자식으로부터 positive acknowledgment를 받았는지 검사하는데 만일 그렇다면 그 패킷에 대한 버퍼 공간을 안전하게(safely) 해제할 수 있다. Positive acknowledgment는 버퍼 관리에 관여하는 것이므로 매 패킷마다 보낼 필요가 없다. 사실 positive acknowledgment는 주기적인 에러 비트맵 정보와 동일한 패킷에 함께 보내는 것이 효율적이다. Negative acknowledgments는 손실 정보의 주기적인 전송으로 인한 잠재적인 지연을 회피함으로써 손실 패킷을 적시에 복구할 수 있게 한다. 즉, 각 노드는 수신된 패킷간에 공백이 생기면 즉시 부모에게 negative acknowledgment을 보내고 그 패킷에 대한 타이머를 구동한다. 재전송 받지 못한 채 타이머가 종료되면 위 과정을 반복하는데 타임아웃은 negative acknowledgment 패킷이나 재전송 패킷이 손실되었음을 의미하기 때문이다. negative acknowledgment을 받은 부모는 유니캐스트 또는 멀티캐스트 재전송을 결정하기 위하여 자식 노드에 대한 상태정보를 열람한다. 만일 그 자식에 대한 멀티캐스트 주소가 있으면 멀티캐스트로 재전송하고 그렇지 않으면 유니캐스트로 재전송한다.

예를 들어 그림 6 (b)에서 노드 B는 자식 C, D, F로부터 positive acknowledgments를 받는다. C, D, F 모두로부터 positive acknowledgment를 받으면 B는 버퍼 해제를 할 수 있다. B가 자식들로부터 negative acknowledgment를 받으면 그림 7과 같은 상태정보 테이블에서 해당 자식의 정보를 열람한다. C와 F에 대해서는 요청된 패킷을 유니캐스트 주소 $UA_C$, $UA_F$ 로 각각 재전송하고 D에 대해서는 멀티캐스트 주소 $MA_D$ 로 재전송한다. 유니캐스트인지 멀티캐스트인지에 상관 없이 모든 재전송은 요청 받은 즉시 이루어진다는 점이 주목할 만하다.

| Relation | ID | W | Address |
|---|---|---|---|
| parent | S | 17 | Unicast address $UA_S$ |
| self | B | 9 | Unicast address $UA_B$ |
| 〃 | 〃 | 〃 | Multicast address $MA_B$ |
| child | C | 1 | Unicast address $UA_C$ |
| child | D | 5 | Unicast address $UA_D$ |
| 〃 | 〃 | 〃 | Multicast address $MA_D$ |
| child | F | 2 | Unicast address $UA_F$ |

그림 7 그림(6)의 노드 B의 상태정보

## 4. 성능 평가

### 4.1 전위 문제의 정량화(Quantification of inversion problem)

왜 전위 문제를 가급적 회피해야 하는지 전위 문제의 역효과(side effects)의 정량화를 통해 알아본다. 그림에 나타난 멀티캐스트 라우팅 트리를 예로 든다. 라우팅 트리에서 S와 $P$간의 링크 수를 s, $P$와 $N_1$간의 링크 수를 t, $P$와 $N_2$간의 링크 수를 u라고 한다. 편의상 하나의 링크를 지나는데 걸리는 시간은 1 시간 단위(time unit)이고 링크의 대역폭은 1 단위 시간당 하나의 패킷으로 가정한다.

지금 S가 보낸 패킷이 손실되어 P에게 전달되지 못했고 그 다음 패킷은 1 단위 시간후에 보내어져 손실없이 $P$와 $N_1$에 도착되었다고 하자. 두 번째 패킷이 도착하는 순간, $P$와 $N_1$은 이전 패킷의 손실을 감지하고 논리적 트리상의 부모에게 재전송 요청을 보낸다. 이때 전위가 있는 논리적 트리와 전위가 없는 논리적 트리에 대해 손실 패킷을 복구하는데 필요한 지연시간과 대역폭이 표에 계산되어 있다. 표에 의하면 전위가 있는 논리적 트리의 경우 $2t+mins(s, t)$ 만큼이 더 들게 된다.

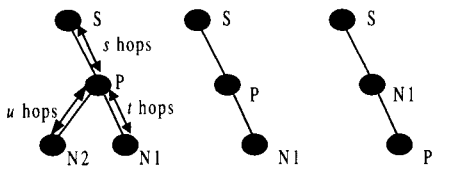
표 1 전위 문제의 정량화

| | 손실 감지 | 손실 복구 |
|---|---|---|
| P | s+1 | 2s |
| N1 | s+t+1 | 2max(s,t) |

(a) 전위가 없는 경우

| | 손실 감지 | 손실 복구 |
|---|---|---|
| P | s+1 | 2(s+2t) |
| N1 | s+t+1 | 2(s+t) |

(b) 전위가 있는 경우




(a) Multicast routing tree (b) logical tree w/o inversion (c) logical tree with inversion

그림 8 성능평가를 위한 예

### 4.2 최적의 에러 비트맵 크기(Optimal error bitmap size)

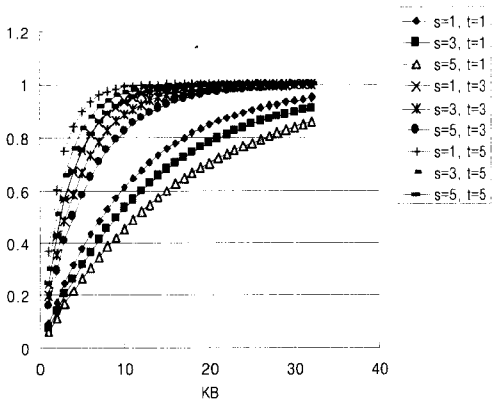
본 절에서는 '최적의' 논리적 트리 구성을 위해 에러 비트맵의 비트맵 크기(비트 수) $K_B$가 얼마로 설정되어야 하는지 알아본다. $K_B$는 그림의 $N_1$과 $N_2$가 서로 부모-자식 관계를 맺지 않도록 충분히 커야 한다. 또한 $K_B$ 크기의 에러 비트맵으로 $P$와 $N_1$이 부모-자식 관계를 맺어야 한다는 사실을 판단할 수 있어야 한다.

$P_{P,N1}$을 $P$와 $N_1$이 각각 부모와 자식임으로 판명될 확률을 나타내기로 하자. 이 확률은 1에서 부모-자식으로 판명되지 않을 확률을 뺀 값이다. 부모-자식으로 판명되지 않을 확률은 패킷이 $P$에 까지 전달되지 않을 확률과 패킷이 $P$와 $N_1$ 모두에게 손실 없이 도착될 확률의 합이다. 즉 패킷이 둘 모두에 대해 공히 손실되거나 아니면 공히 전달될 확률이다. p를 링크의 패킷 손실 확률이라 할 때, $P_{P,N1}$은 아래와 같이 주어진다.

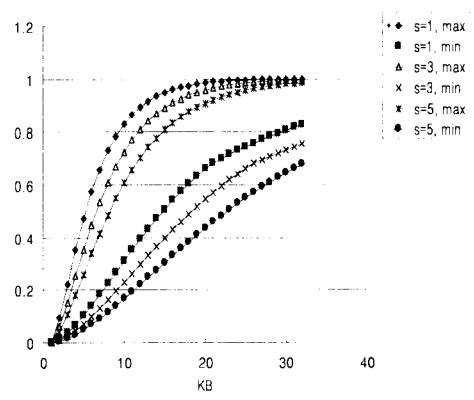
$$P_{P,N_1} = 1-\{1-S+ST\}^{K_B} \quad (2)$$
$$where \quad S=(1-p)^s,\ T=(1-p)^t$$

이제 $N_1$과 $N_2$이 부모-자식 관계에 있지 않은 것으로 판명될 확률을 $P_{N1,N2}$이라 하면, 이 값은 1 - $P_{mis}$ 로 나타낼 수 있다. 여기서 $P_{mis}$ 은 부모-자식 관계에 있는 것으로 잘못 판명될 확률을 말한다. 이러한 오판(misjudgment)은 상관관계가 없는 손실(unrelated losses)로 인해 발생할 수 있다(3.1.3 절을 참조). $P_{mis}$은 $N_1$이 $N_2$와 ⊃관계를 가지는 것으로 잘못 판명될 확률과 $N_2$가 $N_1$와 ⊃관계를 가지는 것으로 잘못 판명될 확률을 합한 값에서 $N_1$, $N_2$ 가 동일한 수신 상태를 가질 확률을 빼면 된다. 이를 계산하면 아래와 같다.

$$P_{N_1,N_2} = 1-P_{mis}$$
$$= 1-[\ (1-S(1-T)U)^{K_B} + (1-S(1-U)T)^{K_B} \quad (3)$$
$$-\ (1-S+S(1-T)(1-U)+STU)^{K_B}]$$
$$where \quad S=(1-p)^s,\ T=(1-p)^t,\ U=(1-p)^u$$




(a) $P_{P,N1}$ the probability that parent-child relationship between $P$ and $N_1$ is revealed ($p$=0.1)




(b) $P_{N1,N2}$ the probability that $N_1$ and $N_2$ are distinguished($p$=0.1)

그림 9 s, t 값에 따른 $P_{P,N1}$ 과 $P_{N1,N2}$

산술적 결과(numerical results)가 그림 9에 제시되어 있다. 그림 (a)에서 $P_{P,N1}$은 $K_B$, $s$, $t$의 함수로, (b)에서 $P_{N1,N2}$은 $K_B$ 및 $s$ 의 함수로 나타내져 있다. 링크간 홉 수 $t$ 와 $u$, 는 1에서 5 사이의 범위에 있다. 식별하기 쉽게 하기 위해 확률의 최대값과 최소값만을 명시하였다. 그림 9의 (a)와 (b)에서 확률이 $K_B$ 가 증가함에 따라 급격한 비율로 1에 가까워진다는 사실을 알 수 있다. 이 결과에 따라 $K_B$ 는 32로 설정되면 논리적 트리 구성에는 별 무리가 없는 것으로 결론 맺을 수 있다. 주기적인 에러 비트맵 교환이 진행될수록 정보량은 32, 64,…로 점점 증가하고 이에 따라 확률 값들도 1에 가까워지게 된다. 확률이 1에 가깝다는 것은 논리적 트리와 멀티캐스트 라우팅 트리가 거의 유사함을 의미한다.

링크 손실 확률 p의 영향을 알아보기 위해, $P_{P,N1}$ 와 $P_{N1,N2}$를 p의 함수로 나타낸 것이 그림 10의 (c) 와 (d) 이다. 여기서 $K_B$ 는 32로 정해진다. p가 0이나 1일 때 두 확률은 0이다. 이는 메시지 손실이 전혀 없거나 모든 메시지가 손실되면 제안된 기법이 동작하지 않음을 나타낸다. 그러나 손실 확률이 0라는 것은 신뢰성 제공 메커니즘이 필요 없음을 나타내고, 손실 확률이 1이라는 것은 어떠한 신뢰성 제공 방법도 동작할 수 없을 함의하기 때문에 아무 문제가 되지 않는다. p가 0을 넘어서기 시작하면 두 확률은 급격히 증가를 보이면서 1에 근접하고 어느 정도 정점에 머무르다 일정 구간을 지나면 점차적으로 감소하면서 결국 0에 이름을 볼 수 있다. 이 결과는 제안한 기법이 현실적인 손실 확률의 범위에서 잘 동작하는 것을 함의하고 있다. 특히 그림 10의 산술적 결과는 $t$ 와 $u$ 값이 클수록 두 확률이 큰데 이는 통신의 범위가 광대역일수록 - 즉 수신자들이 지리적으로 넓게 분포되어(sparsely distributed) 있을수록 - 제안한 기법이 잘 동작함을 의미한다.

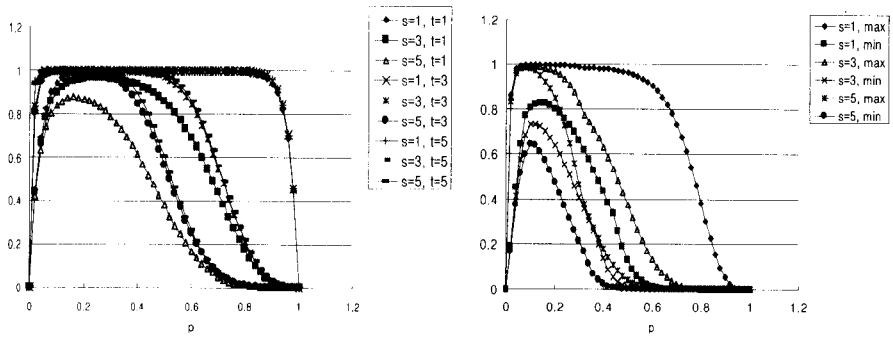
### 4.3 시뮬레이션

본 절은 제안한 기법의 모의실험 결과와 TMTP [9] 와의 비교분석을 기술한다.

4.3.1 시뮬레이션 모델

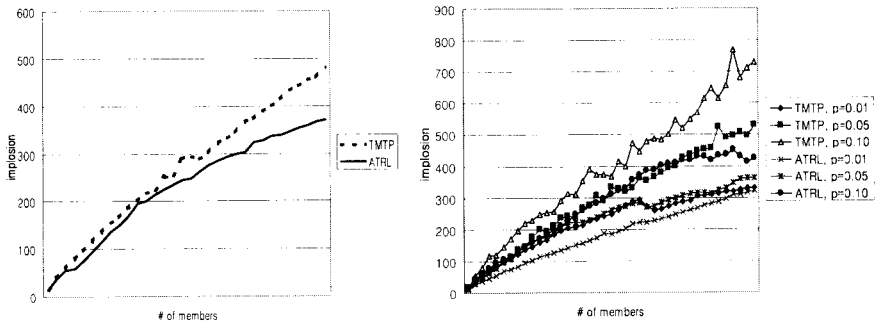
GT-ITM(Georgia Tech Internetwork Topology Models) [14]을 이용하여 400개의 노드를 가진 샘플 네트워크들을 만들고, 이들 네트워크 상에 멀티캐스트 세션을 생성하여 event-driven 시뮬레이션 프로그램 [15]에 의해 모의실험을 수행하였다. 다음은 시뮬레이션 모델의 요약이다.

- 주어진 멀티캐스트 세션에서 임의의 한 노드가 송신자로 선택되고 그 송신자는 일정한 비율로 패킷을 그룹으로 멀티캐스트한다.
- 제안한 기법과 TMTP에 대해 수신자가 동적으로 세션에 참가하도록 하여 확장성을 평가한다.
- 수신자는 최대 300으로 제한한다.
- 링크의 지연 값은 20ms에서 100ms사이의 임의의 값을 가진다(uniformly distributed).
- 링크의 손실확률은 각각의 링크에 동일하게 적용하였는데, 그 값은 0.01에서 0.1까지의 값이다.
- 손실은 임의로(randomly) 일어난다고 가정한다.
- 지역그룹 형성을 위한 문턱 값 $T_h$ 는 10으로 정한다.
- 호스트에서 패킷을 읽어서 네트웍에 쓸 때까지의 지연(delay) 값은 10ms로 정하는데 이 값은 실제로 실험한 측정값이다.
- NACK를 보내기 위한 타이머의 값은 요청자(requester)와 재전송자(retransmitter) 간의 왕복시간(RTT)으로 정한다.




(a) $P_{P,N1}$ as a function of p (b) $P_{N1,N2}$ as a function of p

그림 10 $p$가 $P_{P,N1}$과 $P_{N1,N2}$에 미치는 영향




(a) average implosion (b) implosion at loss probability$p$

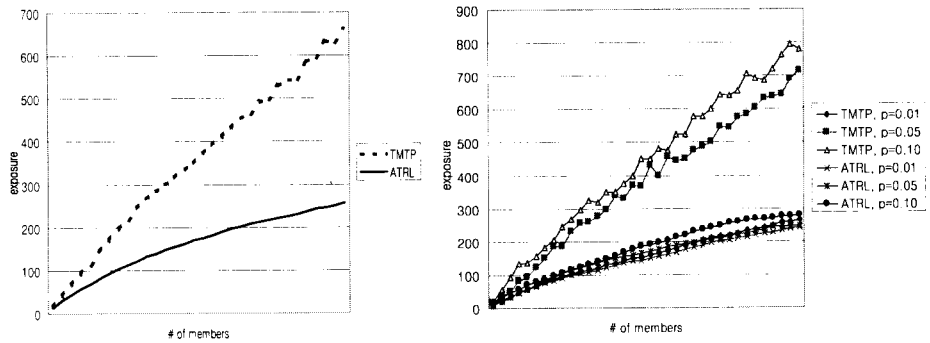
그림 11 평균 피드백 implosion

· 억제(suppression) 타이머의 값은 0에서 왕복시간의 반값의 사이에서 선택된다.

4.3.2 시뮬레이션 결과

시뮬레이션 결과는 그림 10과 11에 나타나 있다. 제안된 기법은 ATRL(Adaptive Tree-based Recovery with Local groups)라 표기되어 있다. 그림 10 (a)는 acknowledgment당 링크들이 implosion된 평균 회수를 나타낸다. 피드백은 positive acknowledgment와 negative acknowledgment를 포함한다. 멀티캐스트세션의 크기가 커질수록 제안된 기법과 TMTP 모두 피드백 implosion의 선형적 증가를 보여준다. 하지만, TMTP의 implosion 오버헤드는 제안된 기법보다 점점 더 커진다. 특히, negative acknowledgement 의 양은 현저하게 차이가 나게 된다. 그 차이는 전위 현상의 증가와 TMTP에서 negative acknowledgement를 제한된 범위의 멀티캐스트(limited scope multicast)의 범위에서 비롯된다. 그림 (b)에서는 다양한 손실 확률(0.01-0.1)에 따른 implosion을 보여준다. 예상대로 손실 확률의 증가에 따라서 implosion도 증가한다.

그림11 (a)는 손실된 패킷을 복구하기위한 재전송 패킷이 평균 몇 개의 링크를 지났는지 그 회수를 보여준다. 결과는 그림 10과 비슷하다. 그러나, 제안된 기법이 TMTP보다 더 낫다는 것을 알 수 있다. 데이타 exposure의 총합에서 제안된 기법과 TMTP가 차이를 보이는 것은 implosion과 같은 이유에서 비롯된다. 그림 11 (b)는 손실 확률의 변화에 따른 exposure의 변화를 보여준다. 둘 다 손실 확률에 비례하고, 흥미롭게도 exposure에 미친 손실 확률의 영향이 제안된 기법보다 TMTP에서 더 컸다.




(a) average exposure (b) exposure at loss probability$p$

그림 12 평균 데이타 exposure

## 5. 결 론

정적인 트리-기반 기법을 멤버쉽 변화나 경로 변화에 적응할 수 있도록 하고 트리 구조에 지역 그룹을 형성함으로써 대규모 신뢰적 멀티캐스트 세션에서 implosion과 exposure를 최소화할 수 있는 기법을 제시하였다. 종단간 멀티캐스트 복구 기법이므로 현재의 네트워크 하부구조에 아무런 변화가 필요하지 않다. TTL 기법의 무방향성에 의해 초래되는, 기존의 트리 기반 프로토콜의 단점을 보완한다. 본 연구의 기여는 전위 문제에 대한 통찰, 멀티캐스트 라우팅 트리에 가까운 논리적 트리를 구성하기 위해 에러 비트맵 정보를 이용한 점, 트리 기반 프로토콜을 지역 그룹 기법과 통합한 점 등을 들 수 있다.

제안한 기법은 일대다(one-to-many) 분배형 멀티캐스트 응용에서 효과적으로 사용될 수 있다. 현재 적응형 트리기반 복구기법을 다대다(many-to-many) 멀티캐스트 세션에서 사용될 수 있도록 확장하는 연구가 진행중이다. 또한 트리기반 프로토콜 뿐만 아니라 다른 접근 방식의 신뢰적 멀티캐스트 프로토콜과의 시뮬레이션을 통한 비교가 이루져야 할 것이다.

## 참 고 문 헌


[1] B. Rajagopalan, "Reliability and Scaling Issues in Multicast Communication," *ACM SIGCOMM 92*, August 1992

[2] C. Papadopoulos, G. Parulkar, and G. Varghese, "An Error Control Scheme for Large-Scale Multicast Applications," *IEEE INFOCOM 98*, March 1998

[3] S. Pingali, J. F. Kurose, D. Towsley, "A Comparison of Sender-initiated and Receiver-initiated Reliable Multicast Protocols," *ACM SIGMETRICS 94*, May 1994

[4] M. Yamamoto, J. Kurose, D. Towsley, H. Ikeda, "A Delay Analysis of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols," *IEEE INFOCOM 97*, April 1997

[5] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," *ACM SIGCOMM 95*, pp. 342-356, August 1995

[6] M. Hoffman, "A Generic Concept for Large-scale Multicast," *Int. Zurich Seminar on digital communications (IZS'96)*, February 1996

[7] M. Hoffman, "Enabling Group Communication in Global Networks," *Global Networking 97*, November 1996

[8] S. K. Kasera, J. F. Kurose, and D. F. Towsley, "Scalable Reliable Multicast Using Multiple Multicast Groups," *ACM SIGMETRICS 97*, June 1997

[9] R. Yavatkar, J. Griffioen, and M Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications," *ACM Multimedia 95*, 1995

[10] J. C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," *IEEE INFOCOM 96*, March 1996

[11] B. N. Levine and J. J. Garcia-Luna Aceves, "A Comparison of Reliable Multicast Protocols," *ACM Multimedia Systems*, 1998

[12] S. Deering, "Host Extensions for IP Multicasting," *Request For Comments 1112*, August 1989

[13] M. Yajnik, J. Kurose, and D. Towsley, "Packet Loss Correlation in the MBone Multicast Network," Proc. Global Internet Conference, November 1996

[14] K. Calvert and E. Zegura, GT-ITM: Georgia Tech Internetwork Topology Models, http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html

[15] W. Yoon, Multicast Simulator, http://vega.icu.ac.kr/~wyyoon/reliable/sim/

[16] B. N. Levine, S. Paul, and J. J. Garcia-Luna Aceves, "Organizing Multicast Receivers Deterministically by Parcket-Loss Correlation," *ACM Multimedia 98*, 1998



윤 원 용

1996년 서울대학교 컴퓨터공학 학사. 1998년 서울대학교 컴퓨터공학 석사. 1998년 ~ 현재 ICU 박사과정 재학중. 관심분야는 reliable multicast, multicast congestion control, mobile computing, multimedia streaming



이 동 만

1982년 2월 서울대학교 컴퓨터공학 학사. 1984년 2월 KAIST 전산학 석사. 1987년 2월 KAIST 전산학 박사. 1987년 3월 ~ 1988년 3월 KAIST post doc. 1988년 4월 ~ 1997. 9월 Hewlett-Packard 책임연구원. 1997년 10월 ~ 현재 ICU 부교수. 관심분야는 네트워크 및 분산 시스템, scalable network architecture for distributed virtual enviroment, reliable multicast protocol, fault-tolerant group communication, layered multimedia multicast, web caching, collaborative computing framework