계층 비디오 멀티캐스트를 위한 효율적인 키 분배 방법

(Efficient Key Distribution for Layered Video Multicast)

임 효 준 † 김 종 권 ††
(Hyojun Lim) (Chongkwon Kim)

요 약 멀티캐스트 데이타의 기밀성을 유지하기 위해서는 멀티캐스트 데이타를 그룹의 공통키로 암호화하여 전송하여야 한다. 그러나 멤버가 멀티캐스트 그룹에 동적으로 가입하거나 탈퇴하는 경우에는 그룹의 공통키를 변경하기 위해 필요한 계산량과 메시지의 수가 그룹의 크기에 비례해 커지는 규모확장성 문제에 봉착하게 된다. 이러한 문제를 해결하기 위해 그룹의 멤버나 키에 계층 구조를 두는 여러 가지 방법들이 제안된 바 있으나, 계층 멀티캐스트에 적용할 수 있는 방법은 아직 제안된 바가 없다. 본 논문에서는 계층 비디오 멀티캐스트 환경에 적용할 수 있는 두 가지의 그룹 키 분배 방법을 제안한다. 첫 번째 방법은 각 계층에 대해 별도의 키 계층 구조를 유지하는 계층별 키트리 방법이며, 두 번째 방법은 하나의 공통 키트리를 유지하고 각 계층을 공통 키트리상의 서브트리로 유지하는 통합 키트리 방법이다. 성능 분석 결과, 멤버가 그룹에 가입해 있는 동안 계층 상승을 자주 하는 경우는 계층별 키트리 방법이 유리하며, 계층의 개수가 많고 계층 상승이 잦지 않은 경우는 통합 키트리 방법이 효율적이다.

Abstract To provide confidentiality to multicast data, data should be encrypted using common group key before transmission. However, in an environment where members can join and leave a group dynamically, scalability problem occurs because the computation complexity and the number of multicast packets increase linearly in proportion to the number of group members. Though several schemes have been proposed to solve the scalability problem, there is no method which can be applied to layered multicast. In this paper, we propose two new group key distribution methods that can be applied in layered multicast. One is a per-layer key tree method where separate key hierarchies are maintained for each layer. The other method is a common key tree method where a single common key tree is maintained throughout whole layers. Performance analysis shows that the per-layer key tree method is better when layer increment occurs frequently during hosts are participated in the group, and the common key tree method is better when the number of layers is large and layer increment does not occur frequently.

1. 서 론

인터넷이 대중화되고 고속화되면서 주문형 비디오 서비스와 같은 그룹 통신 서비스들이 등장하고 있다. 이러한 그룹 통신 서비스는 멀티캐스트 통신을 사용함으로써 효율적으로 구현할 수 있다. 멀티캐스트에서는 패킷을 한 번만 전송하면 그 패킷이 필요한 곳에서 복제되어 각 멤버에게 전송되므로 대역폭과 서버의 부하를 크게 줄일 수 있다.

망 상태가 역동적으로 변동하며 수신자가 이질적인 능력을 가지는 환경에서 멀티캐스트 데이타를 효과적으로 전송하는 방법으로 계층화된 멀티캐스트 방법이 대두되었다[1]. 이 방법에서는 하나의 데이타가 여러 계층으로 나누어져 있으며 상위 계층은 하위 계층에 대해 더 추가적인 정보를 담고 있기 때문에 더 많은 계층을 전송받을수록 데이타의 품질이 좋아지게 된다. 계층화된 데이타를 사용하면 통신망의 상태가 좋지 않거나 고품

† 비 회 원 : 서울대학교 컴퓨터공학부
imhyo@popeye.snu.ac.kr
†† 종신회원 : 서울대학교 컴퓨터공학부 교수
ckim@popeye.snu.ac.kr
논문접수 : 2000년 5월 6일
심사완료 : 2000년 10월 2일

질의 데이타를 필요로 하지 않는 수신자는 낮은 계층의 데이타만을 전송받고 고품질의 데이타를 원하는 수신자에게는 높은 계층의 비디오 데이타를 전송해 주어 차별화된 서비스를 제공해 줄 수 있다.

데이타의 기밀성을 보장하기 위해서는 암호화키를 사용해 데이타를 암호화할 필요가 있다. 그룹 통신에서 효율적으로 암호화키를 분배하는 방법에 관한 많은 연구가 있어왔지만 이러한 연구들은 모두 계층 멀티캐스트 환경을 고려하지 않은 방법이므로 계층 비디오 멀티캐스트에 직접 적용할 수 없다[2-9]. 본 논문에서는 계층 멀티캐스트의 구성 요소로 활용할 수 있는 기존의 키 분배 방법들을 살펴보고 그 개선방법을 제안한다. 또 계층 멀티캐스트 환경에서 효율적으로 동작하는 키 분배 방법을 제안하고 그 성능을 분석한다.

2. 계층을 고려하지 않은 그룹 키 분배 방법

2.1 기본 방법

기밀성을 유지하면서 그룹에 데이타를 보내기 위해 가장 손쉽게 생각할 수 있는 방법은 1대1 통신에서 적용되던 기존의 기밀성 유지 방안을 모든 그룹 멤버에 대해 각각 적용하는 방법이다. 이 방법에서 서버는 데이타를 각 멤버의 키로 각각 암호화하여 보내야 하기 때문에 그룹의 크기에 비례해 서버의 부하가 커지게 된다. 또 멀티캐스트 통신을 사용할 수 없으므로 망 자원도 낭비하게 된다.

이러한 문제는 모든 그룹의 멤버가 공유하는 그룹 공통키를 사용함으로써 해결할 수 있다. 그러나 공통키를 사용하더라도 사용자들이 동적으로 그룹에서 탈퇴하는 경우에는 규모확장성 문제에 봉착하게 된다. 그룹 멤버가 탈퇴하는 경우에 탈퇴한 멤버가 이후의 메시지를 해독할 수 없게 하기 위해 공통키를 변경하여야 하는데, 이 변경된 공통키는 탈퇴한 멤버는 해독할 수 없고 남아 있는 멤버만이 해독할 수 있도록 전송하여야 한다. 변경된 공통키를 남아 있는 모든 멤버에게 1대1로 기밀하게 전송해 주면 되지만, 이 방법은 서버의 부하와 전송할 메시지의 양이 그룹의 크기에 비례해 커지므로 그룹의 크기가 커지는 경우는 적용할 수 없다.

현재 그룹키 분배의 규모확장성 문제를 해결하기 위해 여러 연구가 진행되어 왔다. 이러한 연구는 크게 그룹 멤버에 계층구조를 두는 방법[3,4]과 키에 계층구조를 두는 방법[2,5-9]으로 나눌 수 있다.

그룹 멤버에 계층구조를 두는 방법은 전체 그룹 멤버를 여러 부그룹으로 나누어 키의 변경을 부그룹 단위로 할 수 있게 하는 방법이다. 그러나 이 방법들은 부그룹 관리자를 잘 배치하고 각 멤버들이 부그룹 관리자의 위치를 알아내어 적절한 부그룹에 가입하여야 하므로 현재 사용되고 있는 망에 직접 적용하기는 힘든 방법들이다. 따라서 본 논문에서는 키에 계층구조를 두는 방법만을 다루도록 한다.

2.2 LKH(Logical Key Hierarchy)

LKH[2,5,6]에서 키들은 이진 트리 형태로 유지되는데, 이 트리를 키트리(key tree)라고 한다. 그룹의 멤버는 키트리의 말단 노드에 대응된다. 트리상의 모든 노드에는 하나씩의 키가 대응되며, 그룹의 공통키는 루트 노드에 대응되는 키로 정해진다. 그룹의 멤버는 자신에 해당되는 노드로부터 루트에 이르는 경로상에 존재하는 노드들에 대응되는 키만을 알고 있다.

그룹 멤버가 탈퇴하는 경우에 보안을 유지하기 위해서는 탈퇴하는 멤버가 알고 있던 경로 키를 모두 변경하고 영향을 받는 노드에게 변경된 키를 알려야 한다. LKH에서는 트리의 아래쪽에서부터 키를 차례로 변경해서 변경된 키를 두 자식 노드의 키 값으로 암호화해서 전송한다. 탈퇴하는 노드의 트리상에서의 깊이를 d라고 하면 멀티캐스트해야 하는 키의 수는 $2d-1$이 된다.

새로운 멤버가 그룹에 가입하는 경우에는 단방향 함수(one-way function)를 이용하여 멤버의 가입을 효율적으로 처리할 수 있다. 새로운 멤버를 트리의 말단 노드에 적절히 배치한 다음 가입한 노드로부터 루트에 이르는 경로상의 키들에 단방향 함수 f를 적용시켜 키를 변경한다. 그리고 이 변경된 키를 새로 가입한 멤버에게 알려주는 것이다. 함수 f는 단방향 함수이기 때문에 새로 가입한 멤버는 가입하기 이전의 키 값들에 대한 정보를 전혀 알 수 없다. 이 방법을 사용하면 키를 멀티캐스트하지 않고 멤버의 가입을 처리할 수 있다.

2.3 Canetti의 방법

Canetti의 방법[8]에서도 노드 v_i마다 하나씩의 키 K_i가 대응된다. 한 멤버가 탈퇴하는 경우에 키의 변경은 의사 난수 발생기(pseudo-random generator) $G(x)$를 이용해 이루어진다. $G(x)$는 입력 x의 두 배의 길이를 가지는 출력을 발생시키는 함수이다. $G(x)$의 왼쪽 반을 $L(x)$, 오른쪽 반을 $R(x)$라고 하자.

그림 1에서 멤버 H_1이 탈퇴하였다고 가정하자. 서버는 난수 r을 발생한 후에 K_4는 $L(r)$, K_2는 $L(R(r))$, K_1은 $L(R(R(r)))$로 변경한다. 그리고 r 값은 K_9로 암호화하여 전송하며, $R(r)$은 K_5, $R(R(r))$은 K_3으로 암호화하여 전송한다. 이렇게 하면 각 노드는 $G(x)$를 이용해 자신에게 필요한 노드의 키 값만을 계산할 수

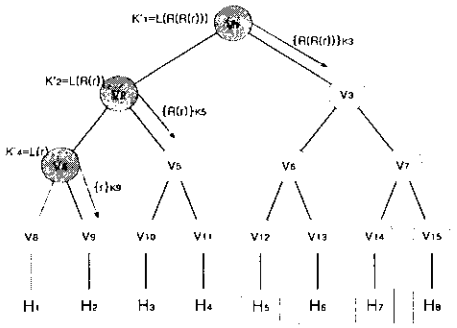

그림 1 Canetti의 방법에서의 키 변경과정

있다.

Canetti의 방법에서는 그룹 멤버가 탈퇴할 때 d개의 키만 멀티캐스트하면 된다.

2.4 Canetti 방법의 개선

Canetti의 방법에서는 노드가 탈퇴할 때마다 한 개의 키 r을 생성하고 d개의 키를 그룹에 전송해야 한다. 예를 들어 그림 1에서는 난수 r을 생성해 K_9로 암호화하여 전송하게 된다. 그러나 r을 별도로 생성하지 않고 r을 K_9로 정하면 난수를 생성시킬 필요도 없고 멀티캐스트 메시지의 수도 하나 줄일 수 있다. 이 개선 방법은 LKH에도 적용할 수 있다.

이 개선 방법을 Canetti의 방법에 적용하면 멀티캐스트해야 하는 키의 수를 $d-1$로 줄일 수 있다. 또 멤버의 탈퇴시에 서버가 난수를 발생시킬 필요가 없어진다.

3. 계층 멀티캐스트에서의 키분배

본 절에서는 계층 멀티캐스트에 적용할 수 있는 두 가지 키분배 방법을 제안한다. 이 방법은 2절에서 설명한 키분배 방법을 구성요소로 하여 동작한다.

3.1 계층 멀티캐스트 시스템 구조

계층 멀티캐스트 구조에서 송신 노드는 N개 계층의 데이타를 전송한다. $k+1$ 계층의 데이타는 1부터 k 계층의 데이타에서 포함하지 않은 추가적인 정보만을 담고 있으므로 $k+1$ 계층의 데이타는 1부터 k 계층의 데이타가 모두 있을 때에만 의미가 있다. 따라서 k 계층까지 가입한 멤버는 1부터 k 계층까지의 모든 데이타를 수신받게 된다.

각 멤버는 멀티캐스트 그룹에 속해 있는 동안 동적으로 가입한 계층의 수를 변경할 수 있다. 계층의 상승 또는 하강은 한 계층 단위로 이루어진다고 가정하자. 즉 k 계층까지 가입한 멤버는 $k+1$로 계층을 늘리거나 $k-1$로 계층을 줄일 수 있다.

3.2 계층별 키트리 방법

앞에서 설명했듯이 계층 멀티캐스트에서는 각 계층마다 별도의 키를 사용하여야 한다. 손쉽게 생각할 수 있는 방법은 2절에서 설명한 키분배 방법을 각 계층별로 따로따로 적용하는 방법이다. 이것을 계층별 키트리 방법이라고 정의하자. 계층 k에 대해 키트리 T_k를 유지한다. 총 N개의 계층이 있다면 N개의 키트리가 유지되며 각 계층의 공통키는 해당 계층의 키트리의 루트 키로 정한다. 계층이 누적관계라고 하면 k 계층까지 가입한 멤버는 $T_1, T_2, \cdots, T_k$의 총 k개의 키트리에 포함되게 된다. 결국 T_k에는 $k, k+1, \cdots, N$ 계층까지 가입한 멤버가 모두 속하게 된다. 따라서 T_1에는 그룹의 모든 멤버가 속하게 될 것이다.

k 계층까지 수신받던 멤버가 $k+1$ 계층까지로 계층을 늘리고 싶은 경우는 T_{k+1}에 새로 가입하면 된다. T_{k+1}에의 가입은 2절에서 설명한 방법을 사용하면 된다. k 계층까지 수신받던 멤버가 $k-1$ 계층까지로 계층을 줄이고 싶은 경우는 T_k에서 탈퇴하면 된다. T_k에서의 탈퇴 역시 2절에서 설명한 방법을 사용하면 된다. 멤버가 그룹에 가입하거나 탈퇴하는 경우는 $T_1, T_2, \cdots, T_k$에 모두 가입하거나 탈퇴하여야 한다.

3.3 통합 키트리 방법

통합 키트리 방법에서는 각 계층별로 별도의 키트리를 유지하지 않고 전체에 단 1개의 키 트리만을 유지한다. 이것을 통합 키트리라고 정의한다. 통합 키트리는 그림 2와 같은 구조를 가지고 있다.

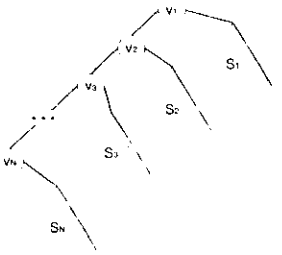

그림 2 통합 키트리의 구조

루트 노드 v_1은 계층 1에 해당되는 공통키 L_1을 유지한다. 이 노드의 왼쪽 자식 노드는 노드 v_2이며 오른쪽 자식 노드는 서브트리 S_1의 루트 노드이다. 서브트리 S_1의 말단노드들은 1번 계층까지만 가입한 멤버들에

대응된다. 마찬가지로 노드 v_2는 계층 2에 해당되는 공통키 L_2를 유지하며 왼쪽 자식노드는 노드 v_3, 오른쪽 자식 노드는 서브트리 S_2의 루트 노드가 된다. 서브트리 S_2의 말단노드들은 계층 2까지만 가입한 멤버들에 대응되게 된다.

노드 $v_1, v_2, \cdots, v_N$은 멤버들이 가입하거나 탈퇴하더라도 추가되거나 없어지지 않고 키트리상에 항상 존재하는 노드들이다. 2절에서의 키트리와 유사하게 모든 노드는 기껏해야 2개의 자식 노드를 가지며 모든 멤버는 트리의 말단노드 중의 하나에 대응된다. 모든 노드는 자신이 대응되는 노드로부터 루트노드 v_1에 이르는 경로상의 모든 키를 유지하고 있다. k 계층까지 가입한 노드는 서브트리 S_k에 속해 있으므로 $L_1, L_2, \cdots, L_k$를 알게 된다.

통합 키트리 방법에서의 서브트리 S_k와 계층별 키트리 방법에서의 T_k는 큰 차이가 있다. S_k는 계층 k까지만 가입하고 $k+1$ 이상의 계층에는 가입하지 않은 멤버들만을 말단 노드로 가지는데 반해, T_k는 계층 k를 수신받는 멤버는 모두 포함하므로 $k+1$ 이상의 계층까지 가입한 멤버들도 모두 포함한다. 따라서 T_k의 말단노드의 수는 항상 S_k의 경우보다 크거나 같게 된다.

k 계층까지 수신받던 멤버가 $k+1$ 계층까지로 계층을 늘려 전송받고 싶은 경우를 고려하여 보자. 서버는 서브트리 S_k에서 해당 노드를 제거한다. 그리고 그 노드로부터 서브트리 S_k의 루트에 이르는 경로상의 키를 변경시켜 주어야 한다. $v_1, v_2, \cdots, v_k$에 해당하는 키는 멤버가 계층 상승 이전이나 이후에 모두 알고 있어야 하는 값이므로 변경시킬 필요가 없다. 해당 노드가 제거되고 나면 다시 해당 노드를 서브트리 S_{k+1}에 추가시킨 후에 추가된 노드로부터 v_{k+1}에 이르는 경로상의 키들을 모두 변경해 준다.

k 계층까지 수신받던 멤버가 $k-1$ 계층까지로 계층을 줄여 전송받고 싶은 경우를 고려하여 보자. 서버는 서브트리 S_k에서 해당 노드를 제거한 후에 그 노드로부터 v_k에 이르는 경로상의 키를 모두 변경해 준다. 그 후 서브트리 S_{k-1}에 노드를 추가하고 그 노드로부터 서브트리 S_{k-1}의 루트에 이르는 경로의 키를 모두 변경해 준다.

새로운 멤버가 계층 k까지 가입하는 경우는 해당 멤버를 서브트리 S_k에 추가시키고 해당 노드로부터 루트에 이르는 경로상의 키들을 변경하면 된다. k 계층까지 가입해 있던 멤버가 탈퇴하는 경우는 해당 노드를 제거하고 루트에 이르는 경로상의 키들을 모두 변경하면 된다.

4. 성능 비교

여기서는 3절에서 제안한 두 가지 키분배 방법의 성능을 비교해 보도록 하자. 키분배 방법의 성능은 여러 가지 측면에서 비교해 볼 수 있으나, 본 논문에서는 가장 중요한 성능 척도인 멀티캐스트되는 키의 수를 기준으로 성능을 비교한다.

총 N개의 계층이 존재한다고 하자. 또 계층 k까지만 가입한 멤버의 수를 n_k라고 하자. 총 멤버의 수 n은 $\sum_{i=1}^{N} n_i$가 된다. 키 변경을 위해서는 2.4에서 설명한 방법을 사용한다고 가정한다. 2.4의 방법에서는 트리에 새로운 멤버를 가입시킬 때는 키를 멀티캐스트할 필요가 없으며, 깊이가 d인 멤버를 탈퇴시킬 때 $d-1$개의 키를 멀티캐스트해야 한다.

계층별 키트리 방법에서는 각 계층별로 한 개씩의 키트리를 유지한다. 계층 k에 해당하는 키트리를 T_k라고 하자. 분석을 단순화시키기 위해 트리 T_k는 완전이진트리(complete binary tree)라고 가정하자. 키트리 T_k에는 $k, k+1, \cdots, N$ 계층까지 가입한 멤버가 모두 포함되므로 T_k의 말단노드의 수는 $\sum_{i=k}^{N} n_i$가 된다. T_k가 완전이진트리이고 $\sum_{i=k}^{N} n_i$개의 말단노드가 있으므로 T_k의 말단노드의 깊이는 $\lceil \log \sum_{i=k}^{N} n_i \rceil$가 된다.

통합 키트리 방법에서 서버는 통합 키트리 S를 유지한다. S는 3.2에서 설명한 것처럼 $S_1, S_2, \cdots, S_N$을 서브트리로 가진다. 트리 S_k 역시 완전이진트리라고 가정하자. S_k가 완전이진트리이고 S_k에는 n_k 개의 말단노드가 있으므로 S_k상에서의 이 말단노드의 깊이는 $\lceil \log n_k \rceil$가 된다. S_k의 부모 노드 v_k의 깊이는 $k-1$이므로 S_k의 말단노드의 S상에서의 깊이는 $\lceil \log n_k \rceil + k$가 된다.

먼저 k 계층까지 수신받던 호스트가 계층을 하나 늘려 $k+1$ 계층까지 수신받는 경우를 고려하여 보자. 계층별 키트리 방법에서는 해당 멤버를 T_{k+1}에 추가시키기만 하면 되므로 변경된 키를 멀티캐스트할 필요가 없다. 한편 통합 키트리 방법을 사용하면 서브트리 S_k에서 제거한 후에 S_{k+1}에 추가시켜야 하므로 서브트리 S_k에서 제거할 때 $\lceil \log n_k \rceil - 1$개의 키가 멀티캐스트