

다자간 UCC 실시간 공유시스템에 대한 설계와 구현

손성옥⁰, 노재춘
세종대학교 컴퓨터공학과

Design and Implementation of The UCC Realtime Sharing System

Sung-Ok Son⁰, Jaechun No

College of Electronics & Information Engineering
School of Computer Engineering Sejong University, Seoul, Korea
sonsuok@hanmail.net

요 약

본 논문은 IEEE 802.11네트워크 환경에서 다중 노드를 하나의 클러스터링 시스템으로 구축하고, UCC(User Created Contents)와 같은 사용자 멀티미디어 데이터를 모든 노드가 실시간으로 공유할 수 있는 시스템을 제안한다. 이를 위해 본 논문은 응용계층의 새로운 패러다임 모델을 제안 하였고, 이에 따라 Ad Hoc 네트워크기술과 Clustering 기술을 사용하였다. 또한 제안된 기술의 안정된 서비스를 제공하기 위해서 HeartBeat 기술을 활용한 시스템 모델을 제안한다.

Keyword: Ad Hoc, Clustering, HeartBeat, Realtime Sharing

1. 서 론

이동하는 다중 사용자노드가 UCC(User Created Contents)같은 사용자 멀티미디어 데이터를 실시간으로 공유하기 위해서는 특수한 패러다임 모델을 수용해야 된다.

본 논문에서는 특수한 패러다임 모델을 수용하기 위해 애드혹(Ad Hoc)네트워크와 클러스터링(Clustering)기술이 접목된 응용계층의 어플리케이션을 제안한다. 유비쿼터스 환경 구축이 집중적인 관심을 받게 됨에 따라 MANET(Mobility Ad hoc NETwork)에 대한 연구가 활발히 진행되었다.

MANET[1]은 애드혹 네트워크 기술의 한 분야로 이동하는 노드가 접속과 종료를 자유롭게 할 수 있고, 노드 구성원 전체가 주체적으로 망을 이루는 특징이 있다. 따라서 본 논문에서 제안된 시스템은 MANET과 같이 언제든지 접속과 종료가 자유로운 MANET 애드혹기술을 적용 하였다. 또한, 본 연구에서 제안한 UCC 실시간 공유시스템은 고가용성을 목적으로 단일 환경 구축을 위해 클러스터링 기술을 접목시켰으며, 시스템의 신뢰성 및 안정성 구축을 위해 애드혹 기반의 HeartBeat 기술을 통합시켰다.

본 논문의 구성은 다음과 같다. 애드혹 및 클러스터링 관련 연구를 제 2장에서 기술하였고, 제 3장에서는 연구한 기술을 바탕으로 설계와 구현을 기술하였다. 그리고 제 4장에서는 본 논문에서 제안한 UCC 실시간 공유 시스템을 테스트하였고, 제 5장은 결론과 앞으로의 계획을 제시한다.

2. 관련연구

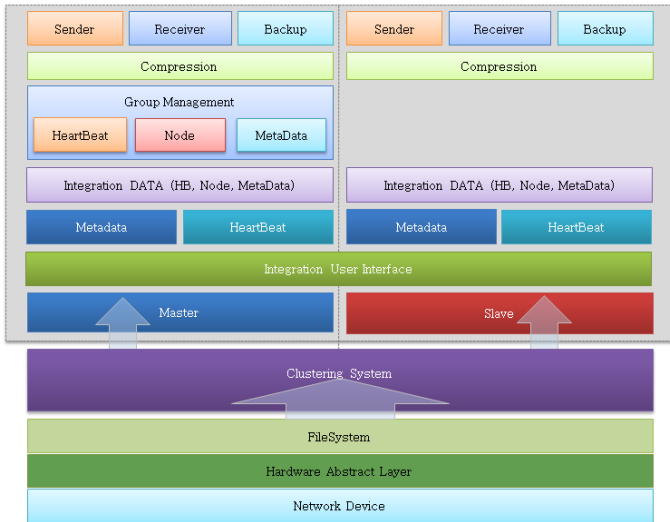
애드혹 네트워크는 액세스 포인트 같은 고정된 기반망이 없는 노드들이 자율적으로 구성되는 임시적인 네트워크이며, 각 노드가 능동적인 망의 주체가 된다[2][3]. 또한 애드혹 네트워크는 노드간 통신을 위해 IEEE 802.11 WLAN(Wirless) 무선 인터페이스를 주로 사용한다. IEEE 802.11 WLAN은 호스트와 액세스 포인트 간에 통신이 이루어지는 Infrastructure모드와, 액세스 포인트 없이 노드간의 통신을 지원하는 애드혹 모드를 지원한다[4]. 애드혹 모드는 멀티홉(Multihop) 라우팅 기능이 MAC이하 계층에서는 지원되지 않기 때문에 네트워크 계층에서 라우팅 기능을 추가하거나, 무선 백본 역할을 수행하는 클러스터헤더를 구성한 후 이를 통해 라우팅 기능을 지원하는 애드혹 기반 클러스터링 라우팅 기술 등을 사용한다[5][6][7].

LVS[8], HA[9] 같은 클러스터링으로 구성된 시스템은 heart-beat[10] 기술을 사용하여 각 노드에 문제가

• 본 논문은 Seoul R&BD Program(10557) 지원을 받아 수행되었음

발생하게 되면 즉시 백업서버가 서비스를 이어받아 기존의 서비스를 지속할 수 있도록 한다. Heart-beat 구현을 위해 각 노드는 특정 노드 혹은 모든 노드에게 주기적으로 신호를 보내며, 일정시간 동안 신호가 없으면 해당 노드에 문제가 발생한 것으로 판단하고 정상적인 노드들만이 서비스 처리에 참여하게 된다.

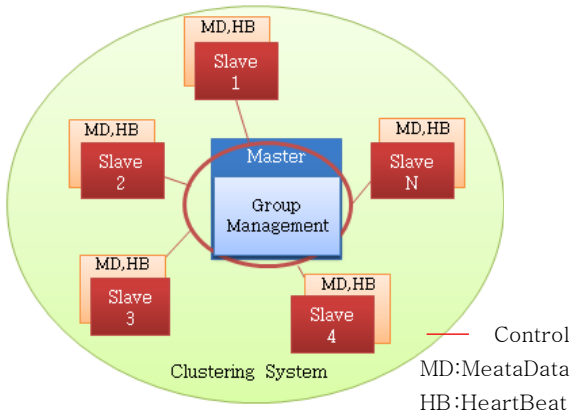
3. 설계와 구현



[그림 1] UCC 실시간 공유시스템

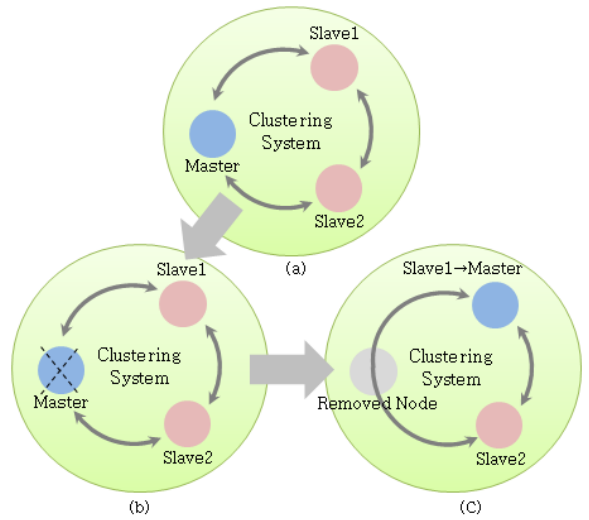
[그림 1]은 본 논문에서 제안하는 실시간 UCC 공유 시스템의 전체구조를 보여준다. 클러스터링 시스템으로 연결된 노드들 중 첫 번째 노드는 클러스터헤더로 설정하였고, 나머지 노드들은 일반노드로 설정하도록 설계하였다. 클러스터 헤더로 설정된 노드는 공유를 위한 전반적인 데이터를 통합, 관리하는 master 역할을 수행하며, 일반노드들은 master에 의해 정보제어가 이루어지는 slave역할을 수행하도록 구성된다.

3.1 노드 구성



[그림 2] 노드 구성

[그림 2]는 그룹별 노드 구성을 보여준다. 클러스터링 시스템에서 그룹별로 라우팅 역할을 수행하는 master가 존재한다. Master노드는 클러스터링 시스템을 생성하고, 그룹을 형성하는 각 멤버들의 노드정보, 메타데이터정보, heart-beat정보들을 통합 관리하는 권한을 갖게 된다. 또한 slave노드가 요청한 요구사항에 대해 응답 서비스를 수행 한다.



[그림 3] Master의 전환

Master노드의 전환 과정은 서비스 중단 시 [그림 3]과 같은 순서로 이루어진다. [그림 3.(a)]에서와 같은 정상적인 클러스터링 시스템에서 [그림 3.(b)]와 같이 master의 서비스가 중단되면, heart-beat에 의해 ‘살아있음’을 증명한 slave들 중 하나가 선택되어 [그림 3.(c)]와 같이 master의 역할을 이어받게 된다. 따라서 클러스터링 시스템의 서비스가 지속적으로 유지되도록 구현하였다.

Slave 노드들은 자신의 노드정보나 메타데이터의 변화가 생기면 변화된 데이터를 master에게 전송한다. 전송된 데이터는 master노드에 의해 통합되며, 생성된 통합데이터는 클러스터시스템에 참가한 모든 slave노드들에게 재전송된다. Slave에 의해 수신된 통합데이터는 클러스터링 시스템의 실시간 공유 정보와 실시간 노드 정보로 활용된다.

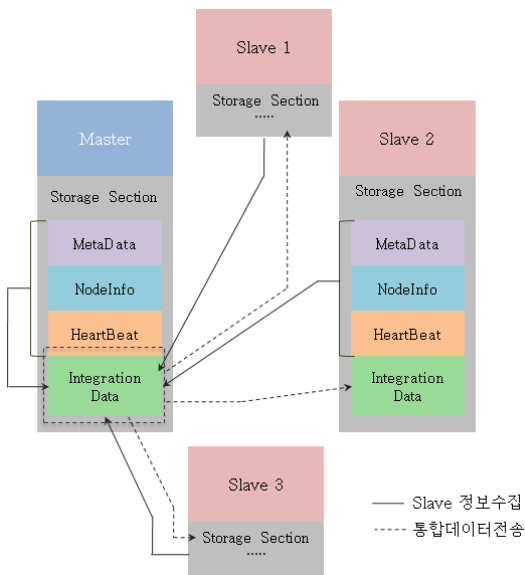
Slave노드가 보유한 통합데이터는 master 중단 시 새로운 master 선출을 위해 참조된다. 즉 통합데이터에 포함된 노드 우선순위[표 1]를 참조하여 우선순위가 가장 높은 slave노드를 선출하여 master노드로 전환되도록 구현되었다.

[표 1] 노드 우선순위

우선순위	노드 정보	내용
1	Heart-beat	상태정상
2	System free memory size	여유공간
3	Node number	접속순서

3.2 그룹 관리

[그림 4]는 master 노드에서 통합데이터 관리과정을 보여준다. 통합데이터에는 클러스터링에 참가한 노드들의 메타데이터, 노드정보, heart-beat 데이터가 포함되도록 구현하였다.



[그림 4] 통합데이터관리

모든 노드는 자신이 공유하고 있는 UCC 정보를 사전에 정의해둔 [표 2]와 같은 메타데이터 프로토 타입으로 메타데이터를 생성하고 master노드로 전송한다.

[표 2] 메타데이터 프로토타입

형식	내용
Metadata number	일련에 증가되는 정수
Metadata count	메타데이터의 총 수
UCC name	UCC의 이름
UCC ext	UCC의 형식(이미지, 동영상..등)
UCC size	UCC의 Size
UCC createtime	UCC의 생성시간
Location	UCC의 현재 위치(노드번호)

노드 정보관리는 [표 3]와 같은 알고리즘으로 설계되었으며 master노드의 그룹관리정책에 의해 관리되도록 구현하였다.

[표 3] 노드 정보관리

```

Algorithm for group node  $N_i (i = 0, 1, 2, \dots, n)$ 

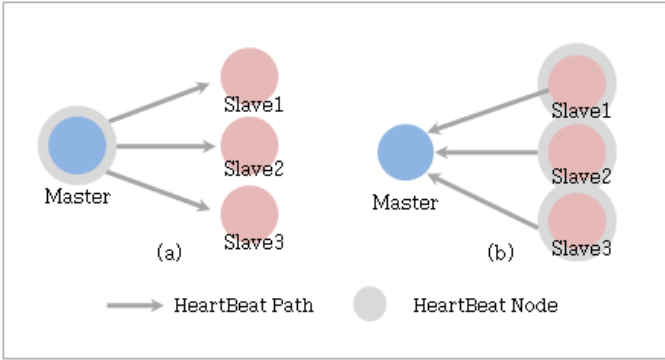
On initialization:
    Insert a new master  $N_0$ ;
On node insertion:
    for  $i \leftarrow 1$  to  $n$ 
        if ( $N_i$  is NULL){
            Insert a new slave  $N_i$ ;
            return TRUE;
        }
    return FALSE;
On node elimination:
    if(success on deletion  $N_i$ ) return TRUE;
    else return FALSE;
    
```

노드정보는 [표 4]와 같은 데이터 형식으로 관리되는데 그 내용은 다음과 같다. 이러한 노드정보는 각 노드가 관리하고 변경된 정보가 발생하면, master 노드로 변경된 정보를 전송하도록 구현 하였다.

[표 4] 노드 정보

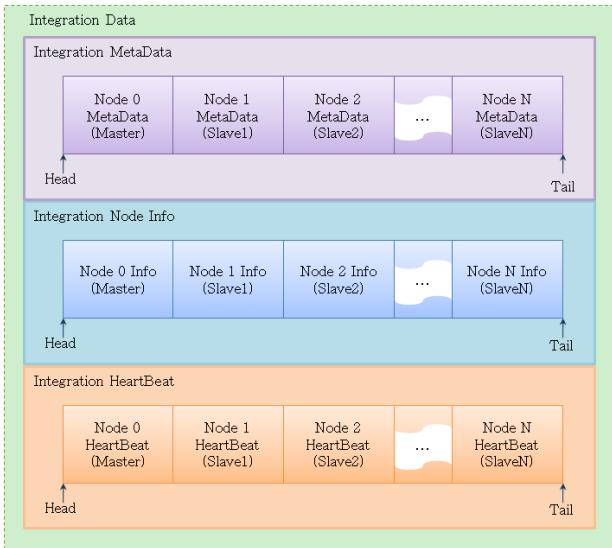
형식	내용
Node number	노드 번호
Node count	노드의 수
Node ip	노드의 아이피
Node memory size	노드의 총 공유 크기
Node free size	노드의 여유 공간
Master ip address	master 아이피 주소
Metadata count	UCC 데이터의 수

본 연구에서 제안하는 heart-beat시스템은 Linux-HA, LVS[11][12]에서 사용하는 heart-beat방식을 기반으로 시스템을 설계하였다. [그림 5]는 본 연구에서 제안하는 heart-beat 시스템이다. [그림 5(a)]는 master 노드가 slave노드들에게 heart-beat을 보내 master노드가 "살아있음"을 slave노드들에게 알리고, [그림 5(b)]는 slave노드에서 자신이 "살아있음"을 master노드에게 heart-beat을 보내 master노드가 인지 할 수 있도록 구현하였다.



[그림 5] Heart-beat

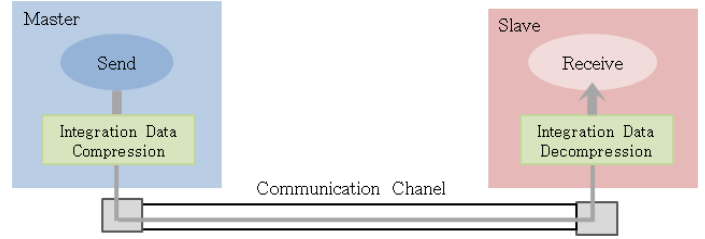
통합데이터는 클러스터링 시스템에 참가하는 모든 노드 정보, 메타데이터 정보, heart-beat 정보 이렇게 3가지 정보를 분리하여 master노드가 관리 하도록 [그림 6]과 같이 설계하였다. 이러한 통합데이터는 master에 의해 모든 노드가 동일하게 유지하도록 구현하였다.



[그림 6]통합데이터

본 연구에서 제안된 클러스터링 시스템은 모든 정보유지를 통합데이터로 관리하기 때문에 노드간에 공유하는 UCC 데이터의 양이 증가하거나 클러스터링 시스템에 참여하는 노드의 수가 많아지면, 통합데이터의 크기도 함께 증가되는 구조로 설계 되어있다. 따라서 통합데이터의 크기가 커지면 네트워크로 전송되는 데이터가 증가되고, 결국 네트워크 대역폭이 감소되어 안정된 클러스터링 서비스를 방해 할 수 있기 때문에 본 논문에서는 통합 데이터를 압축하여 네트워크로 전송 하도록 구현하였다. [그림 7]은 본 연구에서 제안한 통합데이터 압축 방법이다.

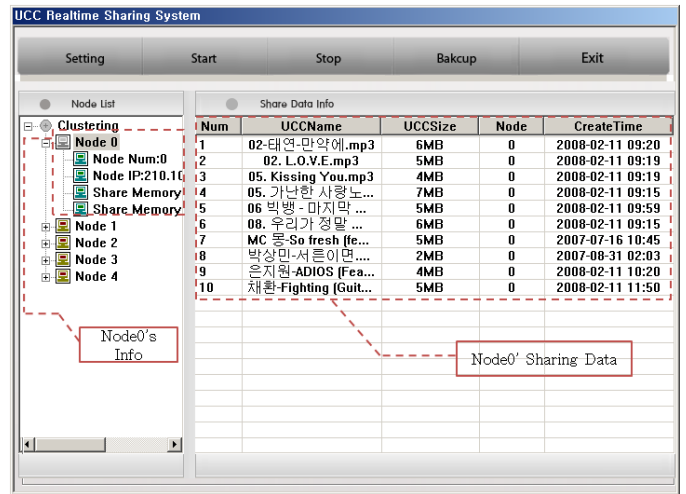
본 연구에서는 통합데이터 압축을 위해 Huffman Code[13]를 이용하여 통합데이터를 무손실(Lossless)압축 방법으로[14][15] 구현하였다.



[그림 7] 통합데이터 압축전송

4. 실험 및 결과

실험에는 총 4대의 노드를 사용하였고 UCC실시간 공유 시스템은 [그림 8]과 같다. 각 노드의 운영체제는 윈도우즈 시스템을 기반으로 무선인터넷 IEEE 802.11b 환경에서 Win32API를 사용하여 구현 하였다. 시스템의 좌측 화면은 클러스터링 시스템의 노드 정보를 표시하였고 노드를 선택하면 선택된 노드의 공유정보가 우측의 공유정보 창으로 출력되도록 구현하였다. 그리고 좌측 노드정보 최상단 클러스터링이 선택되면 현재 클러스터링 시스템 모든 노드의 UCC 공유정보가 실시간으로 우측에 출력되도록 구현하였다.



[그림 8] UCC 실시간 공유 시스템

5. 결론 및 향후 연구과제

본 논문에서 제시한 클러스터링 시스템은 애드혹 기술과 클러스터링기술을 사용하여 복잡한 절차 없이 쉽게 클러스터링 시스템을 구성하여 UCC를 실시간으로 공유할 수 있었다. 그리고 heart-beat 기술을 이용하여 클러스터링 시스템에 참여한 노드들의 접속과 종료가 자유로울 수 있었다. 그러나 통합데이터 관리가 master노드로 집중되는 방식을 사용하였기 때문에 master노드에서만 병목현상이 발생할 가능성이 있어 자원효율성 좋지 못한 단점을 지니고 있다.

현재 성능 향상을 위해 통합데이터를 master 노드뿐

만 아니라 클러스터링 시스템에 참여하고 있는 모든 노드가 통합데이터 관리를 분담할 수 있도록 현재 시스템보다 효율적인 시스템을 설계 중에 있고, PDA나 PMP 같은 임베디드 플랫폼에서 지원이 가능하도록 구현 중에 있다.

6. 참고문헌

- [1]C.K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems", Prentice Hall PTR, 2002.
- [2]Charles E. Perkins, "Ad Hoc Networking", Addison-Wesley, 2001
- [3]M. S. Corson and J. P. Macker. "Mobile Ad hoc Netwrking(MANET):Rrouting Protocol Performance Issues and Evaluation Considerations", IETF RFC 2501. Jan. 1999.
- [4]William Stallings, "Wireless Communications And Networks", Prentice Hall, 2005.
- [5]김상하, "Ad Hoc 네트워크 기술", 한국전자통신연구원
- [6]남상우, "Mobile Ad Hoc Network 연구 동향 및 무선 접속 기술", 한국전자통신연구원 이동통신연구소
- [7]C.-C.Chiang, H.-K.Wu, W.Liu, and "Routing in Clustered Multihop, Wireless Networks with Fading ChaProc of IEEE SICON '96, Apr
- [8]Linux Virtual Server, <http://www.linuxvirtualserver.org>
- [9]Hight-availability Linux Projejt, <http://linux-ha.org>
- [10]Alan Roberson, "LINUX-HA-HeartBeat System design", The 2000 ASL conference, Oct 2000/
- [11]Load Balancing and High Availability Solution, "Ultra Monkey", <http://www.ultramonkey.org>
- [12]HighAvailability, "Heartbeat", <http://www.linux-ha.org/HeartbeatProgram>
- [13]Huffman, D. A. (1952), "A method for the construction of minimum redundancy codes", Proc. IRE, Vol. 40. 9. Sept., pp.1098-1101.
- [14]Jain, A. K. (1981), "Image data compression: A review", Proc. IEEE, Vol.69, March, pp.349-389.
- [15] 박상호, "정규 허프만 트리를 이용한 허프만 코드의 효율적인 디코딩" 한국컴퓨터정보학회 , 2007.