제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

P2P 를 이용한 릴레이식 방송모델

김귀중*, 옥경달**, 신병호**, 이상범**
*삼성 전자
**단국대학교 전자계산학과
e-mail : okkyoung@dankook.ac.kr, yu_zi@hanmail.net, sblee@dankook.ac.kr

A Relay Broadcasting Model Using P2P

Gui-Jung Kim*, Kyoung-Dal Ok**, Byung-Ho Shin**, Sang-Bum Lee**
*Samsung Electronic co.
** Dept. of Computer Science, Dankook Graduated University

요 약

본 논문에서는 P2P 를 사용한 방송모델을 소개한다. P2P 기술을 이용한 릴레이식 방송모델 시스템을 실질적으로 설계, 구현 한다. 현재의 네트워크 인프라를 통해 서버-클라이언트 방송모델의 한계를 극복할수 있는 방송모델의 구현을 목적으로 한다. 본 논문의 P2P 를 이용한 릴레이식 방송 기술을 이용하여 실질적으로 인터넷 방송에 사용 할 수 있게 최적화된 시스템을 도출하고 그것을 통해 품질 높은 방송 시스템을 설계, 구현하였다. 기존의 서버-클라이언트 방송 모델에서의 여러 문제점 특히 다수의 사용자가 접속시 접속 거부 또는 저품질의 방송을 내보내는 현상을 P2P 기술을 통해 작업을 분담함으로써 접속자 수와 관계없이 방송 품질을 유지하고 인터넷 방송에서 품질에 대한 신뢰성을 유지 할 수 있는 시스템을 구축한다

1. 서론

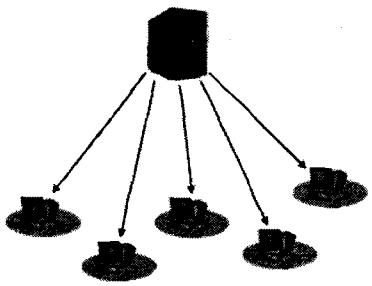
현재 대중화되어 있는 인터넷 방송 모델은 서버-클라이언트 모델이다. 서버-클라이언트 모델은 방송을 요청하는 클라이언트에게 1:1로 방송을 송출해주는 모델로써, 서비스 안정성 면에서의 장점을 기반으로 방송모델의 기본으로 자리 잡았다. 그러나 실제로 방송을 듣기 위해 접속하여 보면 많은 수의 접속자로 인하여 접속 자체가 거부되거나 방송 품질이 떨어지는 경우가 많다. 서버 측 입장에서 보면 1:多의 시스템으로 서비스되기 때문에, 자원의 한계를 가지고 있는 서버는 더욱 많은 수의 클라이언트 요청을 처리하기 위해 이전보다 고성능의 장비를 갖춘 서버 시스템으로 계속하여 보강되어 왔다. 그러나 아무리 고성능, 고가의 서버가 채용되고 보강이 되어도 결국 근본적인 문제는 해결되지 못하고, 또다시 부하에 한계에 도달하는 악순환의 고리를 이어나가고 있다.

본 논문에서는 이러한 악순환의 고리를 끊을 수 있는 방법을 제안하고자 한다. ' 과연 인터넷 방송 송출시 서버에 걸리게 되는 부하를 줄이는 방법에는 어떠한 것이 있을까?' 라는 고민 끝에 선택된 것이 바로 Peer To Peer(이하 P2P)를 이용한 릴레이식 방송모델이다. 최근 사용하고 있는 대부분의 클라이언트 컴퓨터들은 어느 정도 서버의 부하를 나누어 주어도 충분할 것이라는 가정이 이러한 방송 모델을 제안하게 된 바탕이 되었다. 클라이언트에 서버의 일을 나누어 주어 릴레이 식 방송을 한다면, 기존의 서버-클라이언트 모델의 한계를 극복할 수 있으리라 생각한다.

본 논문에서 제안되는 방송모델은 P2P를 이용하기는 하지만 서버가 없는 완벽한 P2P 모델은 아니다. 인터넷을 이용한 방송이라는 특성상 서버는 존재해야만 하고, 구현하고자 하는 목표가 인터넷 방송 송출시에 서버에 걸리는 부하를 줄이자는 것이지 P2P 프로토콜을 구성하는 것은 아니기 때문이다.

2. 인터넷 방송모델

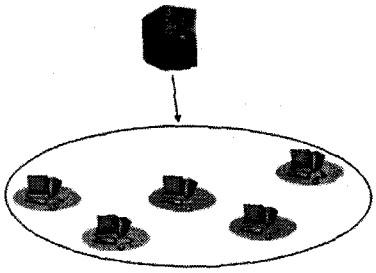
2.1 서버-클라이언트 모델



<그림 1> 서버-클라이언트 모델

<그림 1>의 서버-클라이언트 모델의 장점은 서버와 클라이언트가 직접적으로 연결되어 있어 데이터 전송의 안정성을 보장할 수 있다는 점이다. 단점으로는 하나의 서버에서 모든 클라이언트에 데이터를 보내야 하기 때문에 고속의 시스템을 가져야 하고, 서비스 능력에 한계가 있다는 것이다.

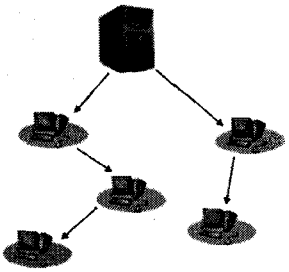
2.2. 멀티캐스팅 모델



<그림 2> 멀티캐스팅 모델

<그림 2>의 에 보이는 모델은 궁극적으로 방송 모델들이 발전해 나갈 방향이라고 볼 수 있다. 각각의 클라이언트들은 서로 다른 네트웍 안에 있지만 멀티캐스팅에서는 그룹화를 지원함으로써 서버는 단 한번의 데이터 전송만으로도 전체 클라이언트들에게 데이터를 전송할 수 있다. 이 모델은 IPv6가 확립되어야만 실용화 가능하며 인프라 또한 새롭게 구축되어야 한다.

2.3. 릴레이식 모델



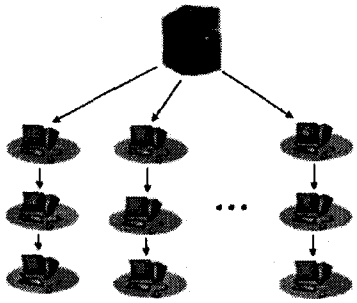
<그림 3> 릴레이식 모델

위의 두 모델의 장점과 단점을 섞어놓은 것이 P2P기반의 릴레이식 방송 모델이다. 릴레이식 방송모델은 멀티 캐스팅과 같이 서버의 부하를 최소화 할 수는 없지만 현재의 네트웍 인프라에서 완벽히 지원될 수 있으며 서버-클라이언트 모델보다도 효과적인 데이터전송을 수행할 수 있다.

서버-클라이언트 모델과 비교할 때 동일한 5개의 클라이언트에게 데이터를 전송할 때 서버-클라이언트 모델은 5번의 동일한 데이터를 전송해야 했지만 P2P기반의 릴레이식 모델에서의 서버는 적은 수의 데이터 전송으로도 전체에 데이터를 전송할 수 있다.

3. 구성 및 구현

3.1. 시스템 구성



<그림 4> 전체적 구성

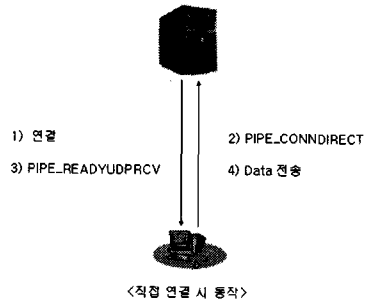
<그림 4>와 같이 20*3의 클라이언트가 연결되도록 하였다. 하나에 서버에 직접 연결되는 클라이언트는 총 20대이고, 각 클라이언트는 3개의 Level로 구성되어 총 60대의 클라이언트가 동작되도록 구현하였다.

완벽한 P2P 모델과는 달리 이 방송 모델에서는 모든 동작의 관리는 서버를 통해 이루어진다. 클라이언트가 서버에 연결을 요청하면 전체 연결 상태를 감지하고 있는 서버가 알맞은 자리에 클라이언트를 할당한다. 처음 클라이언트가 서버에 접속하면 1 node 1 level이 할당된다. 이후 접속 순서에 따라 차례로 1 node 2 level, 1 node 3 level, 2 node 1 level, 2 node 2 level의 순서로 할당된다. 만약 클라이언트의 접속 해지로 비어 있는 자리가 생기면 그 자리에 새로운 클라이언트를 할당한다.

서버와 직접 연결된 1 level에 위치한 클라이언트에게는 서버가 직접 데이터를 전송하고, 다른 클라이언트에 연결된 2, 3 level에 위치한 클라이언트에게는 각각의 상위 level의 클라이언트가 데이터를 전송하게 된다.

3.2. 기본 동작

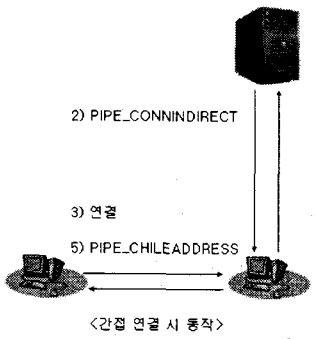
3.2.1. 직접 연결인 경우 (1 level)



<그림 4> 1 level

① 클라이언트가 서버에 연결 요청을 하면, 서버는 새로운 클라이언트의 연결을 받아들이고 클라이언트에게 클라이언트의 정보를 요청한다.
② 클라이언트의 정보를 받은 서버는 전체 연결 정보와 비교하여 클라이언트를 알맞은 자리에 배치한다. 새로운 클라이언트가 1 level에 위치할 경우 PIPE _CONNDIRECT라는 메시지를, 2 또는 3 level에 위치할 경우 PIPE_CONNINDIRECT라는 메시지와 데이터를 클라이언트에 보낸다.
③ PIPE_CONNDIRECT라는 메시지를 받은 클라이언트는 자신의 IP와 UDP포트를 저장하고 서버에게 PIPE_READYUDPRCV라는 메시지를 보낸다.
④ 메시지를 받은 서버는 클라이언트에게 데이터를 전송한다.

3.2.2. 간접 연결인 경우 (2, 3 level)



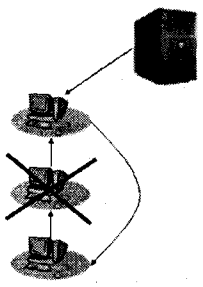
<그림 5> 2,3 level

①, ② 의 동작은 직접 연결과 같다.
③ 상위 클라이언트에 연결을 한다.
④ 요청을 받으면 상위 클라이언트는 연결을 하고, 상위 클라이언트는 연결복구를 위해서 PIPE_PARENTADDRESS 메시지를 하위 클라이언트에게 보낸다.
⑤ 하위 클라이언트는 자신의 정보를 상위 클라이언트에게 제공한다.
⑥ 데이터 전송한다.

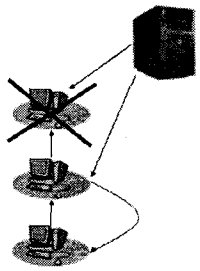
3.3. 상위 클라이언트의 종료

본 모델에서 가장 문제가 될 부분은 2,3 level에 연결된 클라이언트가 상위 클라이언트에게서 데이터를 받는 도중 상위 클라이언트가 종료되는 경우이다. 이러한 경우 데이터를 받고 있던 클라이언트는 다른 클라이언트나 서버로부터 데이터를 전송 받아야 한다. 하위 클라이언트가 상위 클라이언트의 종료상황을 인식하고 그것을 서버에 알려 서버의 연결 정보를 갱신하고 그에 따라 각 클라이언트의 위치를 조정 함으로서 이 문제를 해결할 수 있다.

하위 클라이언트가 상위 클라이언트의 상태를 인지하는 방법은 정상적 종료인지 비정상적 종료인지에 따라 다르다. 정상적인 종료의 경우 소켓이 close 되어 발생하는 FD_CLOSE 메시지를 하위 클라이언트에서 인지하여 서버에 알려주게 된다. 비정상적 종료는 이러한 방식으로 해결할 수 없다. 그래서 타이머를 도입하여 일정기간(1000 mS) 상위 클라이언트에서 데이터가 들어오지 않을 경우 서버에 접속이 종료되었음을 알리는 방식으로 문제를 해결한다.



<그림 6> 2 level의 종료



<그림 7> 1 level의 종료

위의 그림과 같이 2 level이 종료하였을 때 3 level이 2 level이 되어 1 level로부터 데이터를 전송받고, 1 level이 종료하였을 경우 2, 3 level이 각각 1, 2 level이 되도록 구현한다.

4. 성능

◎ 서버-클라이언트 Vs 릴레이 모델

릴레이 모델은 서버-클라이언트 모델과 비교했을때이론적으로는 level을 n까지 두었을 경우 서버의 부담은 1/n로 줄어든다. 즉, 같은 성능의 서버일 경우 릴레이 모델이 서버-클라이언트 모델보다 3배 더 많은 클라이언트를 감당할 수 있다. 그러나 데이터의 전송 이외의 모든 제어가 서버에서 이루어지기 때문에 예상만큼의 성능 향상이 이루어지지는 않는다. 또한 데이터 전송이 전체 성능에서 차지하는 부분이 100%가 아니므로 전체 성능이 비약적으로 향상되지는 않는다.

◎ Local & 고성능 서버

성능 테스트의 운영체제는 Windows 2000이고, Ram은 128메가로 고정된다.

실행 환경

서버> CPU : Intel Pentium III 800MHz
클라이언트> CPU : Intel Pentium III 800MHz

실행 결과

<표 1> 서버-클라이언트 모델

클라이언트의	0	5	10	15	20
메모리(KB)	7396	12176	12180	12200	12316
CPU(%)	0	27	31	34	34
UDP(datagrams/	0	47	95	140	180

<표 2> 릴레이 모델

클라이언트의	0	5	10	15	20
메모리(KB)	10524	13792	13768	13772	13908
CPU(%)	0	22	24	24	24
UDP(datagrams/	0	20	40	47	66

◎ 원격지 & 저성능 서버

실행 환경

서버> CPU : Intel Pentium II 350MHz
클라이언트> CPU : Intel Pentium III 800MHz

실행 결과

<표 3> 서버-클라이언트 모델

클라이언트의	0	5	10	15	20
메모리(KB)	8112	1143	1147	1153	1163
CPU(%)	0	32	34	37	40
UDP(datagrams/	0	49	100	149	201

<표 4> 릴레이 모델

클라이언트의	0	5	10	15	20
메모리(KB)	1215	1424	1425	1428	1434
CPU(%)	0	21	25	30	32
UDP(datagrams/	0	20	39	51	69

◎ 클라이언트의 자원 소모 변화

실제 프로그램이 사용되려면 서버부분의 효율적 이용도 중요하지만, 클라이언트에 걸리는 부하가 크지 않아야 한다. 클라이언트에게 피부로 느낄만한 성능의 저하를 가져다 주어서는 안되기 때문이다. 다음에서는 저사양 클라이언트의 자원소모 변화를 살펴보았다.

실행환경

서버> CPU : Intel Pentium III 800MHz
클라이언트> CPU : Intel Pentium II 350MHz

실행 결과

<표 5> 클라이언트 리소스

하위 클라이언트	없는 경우	있는 경우
메모리(KB)	7260	7282
CPU(%)	1-2	2-3
UDP(datagrams/sec)	20	31

5. 결론

첫 번째 테스트에서 릴레이 모델이 클라이언트 모델보다 성능이 향상되었음을 알 수 있다. UDP를 통한 데이터 전송은 level 수와 같은 약 3배의 성능 향상을 가져왔고. CPU 점유율은 약 8-10%의 감소를 가져왔다. 원격지와 로컬 모두에서 서버의 성능 차이를 제외하고는 비슷한 결과를 보여준다.

메모리에서 나타난 약간 다른 결과는 릴레이 모델과 서버 클라이언트 모델이 가지는 약간의 차이에서 나온것으로 보인다.

중요한 부분중의 하나였던 클라이언트의 부담 또한 거의 없는 것으로 나타났다. 실제로 컴퓨터를 사용하는 입장에서 CPU 점유율 1%의 증가나 메모리 20 KB의 증가는 느끼기 힘들다. 이러한 손해는 더 나은 방송 서비스를 제공받기 위해 감당할 수 있는 양으로 생각된다.

참고문헌

[1]Network Programming for Microsoft Windows 2E, MICROSOFT PRESS, Anthony Jones 외, 1999-07-03
[2]NETWORK PROGRAMMING FOR MICROSOFT WINDOWS, MICROSOFT PRESS,Don Jones, 1999-09-18
[3]WIN32 NETWORK PROGRAMMING, ADDISON WESLEY, DAVIS, 1996-10-15
[4]PROGRAMMING WINDOWS 5/E, MICROSOFT PRESS, Charles Petzold, 1998-12-20
[5]TCP/IP ILLUSTRATED, VOLUME 1, ADDISON WESLEY, W. Richard Stevens, 1994-07-27
[6]microsoft.public.win32.programmer.mmedia
[7]microsoft.public.win32.programmer.directx.audio
[8]microsoft.public.win32.programmer.networks
[9]Lino, Seok (lino@freechal.com) COWON System