



A Design and Implementation of High Quality Live Stream Backup Transport System Using Shortest Path in Private Network

Myeong-Ki Jung, Jae-Woong Kim*

Department of Computer Engineering, Kongju National University

ABSTRACT

There is a company that is not available Multicast IPTV live streaming. The company, place the relay server in each branch office, is sent in unicast live stream from the stream server at headquarters. To convert from unicast to multicast format in office. Then, it sends to the client. This is the overlay multicast. In the case the relay server is not in a normal state, all clients of the branch may not be able to view the live stream. This should solve the problem. First, I want to generate a tree structure with vertices relay server where the problem occurred. I seek the shortest path between other branches from the vertex. At this time, the modified dijkstra algorithm is used. I select a backup path associated with shortest path. Clients of the vertex can use the transfer path of the selected backup, connect to the relay server of the other branches, has received the live stream using unicast. Thus, it is possible to view live stream stably.

© 2014 KKITS All rights reserved

KEYWORDS : IPTV, Private Networks, Enterprises, Relay Servers, Live Streamings

ARTICLE INFO: Received 10 June 2014, Revised 14 August 2014, Accepted 14 August 2014.

1. 서론

*Corresponding author is with the Department of Computer Engineering, Kongju National University, 275, Buda-dong, Seobuk-gu, Cheonan-si, Chungcheongnam-do, (1223-24, cheonandae-ro), 331-717, KOREA.
E-mail address: jwkim@kongju.ac.kr

IPTV 서비스는 일반 사용자들을 대상으로 하는 서비스이다. 현재의 IPTV 서비스는 그에 대한 기술, 시설 투자가 많이 이루어져 안정적인 서비스를 제공한다. 그러나 많은 지사/지점을 보유하고 자체 전용망을 구축한 조직의 경우에는 시설 투자가 제대로 이루어지지 않는 것이 사실이다. 그러므로 최

신의 기술을 탑재한 장비를 도입, 운용할 수 없어도 고화질의 영상 전달 뿐만 아니라 안정적인 영상정보를 실시간으로 전달하는데 어려움이 있다.

또한, 구축된 시설의 노후화로 인해 IPTV Multicast가 제대로 전달되지 못하는 네트워크 장비를 운용하는 경우도 많이 존재한다.

이에 본 논문에서는 사설 전용망 (Private Network)를 보유한 중대규모의 조직에서의 IPTV Live Streaming 전송 시 안정적으로 서비스 할 수 있는 방안을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 제 2장 관련연구에서는 기존의 IPTV 구성과 전용망 IPTV의 차이점과 구조, 문제점에 대해서 살펴본다. 제 3장에서는 제 2장에서 살펴본 문제점을 해결할 수 있는 방법을 기술하며, 제 4장에서는 결론을 기술한다.

2. 관련연구

2.1 공용망 IPTV Live Streaming 방식

현대의 IPTV 서비스 방식은 국가적인 단위에서 운용되면 <그림 1>과 같이 구성되며, 주로 사용되는 스트림 전달 방식은 멀티캐스트(Multicast) 방식을 이용하고 있다.

<그림 1>에서 SHO(Super hub office)에서 비디오 콘텐츠들이 수집되어 광역 단위 내의 각각 VHO(Video hub office)로 전송되어 지며 다시 로컬 지역 내의 VSO(Video service office)로 전달되어 진다.

비디오 콘텐츠들은 VSO에서 DSLAM 장비와 가정용 게이트웨이 (RG : Residential Gateway) 장비를 통해 가정 내의 Set-top Box(STB)로 전송되어 시청할 수 있게 되어 진다. 이 때 SHO → VHO → VSO → DSLAM → RG → STB 간의 전송방식은 효율성을 이유로 멀티캐스트 전송 방식을 사용하고 있다[1].

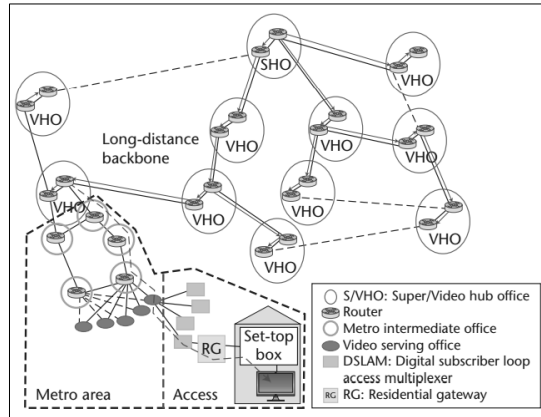


그림 1. 공용망 IPTV 아키텍처
Figure 1. Public IPTV architecture

2.2 전용망 Live Streaming 방식

기업 또는 기관에서는 지역단위에 있는 지사 또는 지점 간에 업무망을 구축할 시에는 직접 또는 ISP에서 제공하는 사설 전용망을 주로 이용한다. 사설 전용망 내에서의 비디오 스트림을 전송하기 위한 일반적인 구성은 여건이 허락한다면 2.1절의 공용망 IPTV Live Streaming 방식으로 멀티캐스트를 지원하는 IP 라우터 또는 스위치 들을 이용하는 것이 최적이라 할 수 있다. 그러나 실제로는 해당 기업 또는 기관의 업무에 주안점을 두어 구축되어진 사설 전용망이기 때문에 <그림 2>와 같이 구성하는 것이 일반적이라 할 수 있다.

<그림 2>에서 Live Encode와 Live Stream Server의 구성은 공용망 IPTV 아키텍처 상의 SHO가 비디오 콘텐츠를 수집해서 전송해 주는 것과 같은 역할을 수행한다.

그러나 하위 지사/지점에서는 멀티캐스트 스위치가 아닌 중계서버(Relay Server)를 두고 있고 해당 중계서버가 Live Stream Server에서 유니캐스트(Unicast) 방식으로 라이브 스트림을 받아와 소속 지사/지점의 클라이언트들에게 멀티캐스트 방식으

로 라이브 스트림을 전송하여 준다. 실제로 지점들은 상위 지사의 라우터를 통해 Live Stream Server에서 직접 유니캐스트 방식으로 라이브 스트림을 전송 받게 된다. 이러한 방식을 채택하는 주된 이유는 사설 전용망 상에서 멀티캐스트 스트림을 전송할 때에는 각 라우터들의 CPU Utilization이 상승하게 되어 실제 기업 활동에 필요한 주요 업무 데이터를 주고받는데 방해적인 요소가 될 수 있기 때문이다.

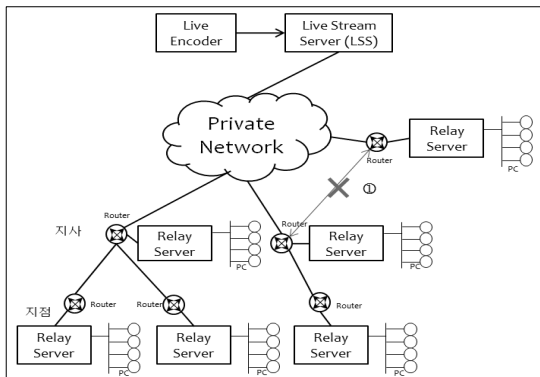


그림 2. 전용망 Live Streaming 구성
Figure 2. Private network live streaming configuration

<그림 2>와 같은 방식과 같이 멀티캐스트 전송과 유니캐스트 전송을 혼합하여 사용하고 있는 하이브리드 전송 방식을 오버레이 멀티캐스트 기술이라 한다. 또한 <그림 2>의 중계서버와 같이 특정 고정 노드를 사용한 멀티캐스트 방식을 Fixed Node Based Overlay 방식이라 한다[2].

오버레이 멀티캐스트 프로토콜 관련 연구들은 네트워크상에 ScatterCast Proxy를 위치 시키고 프록시 간의 유니캐스트 통신을 이용하는 ScatterCast [3], DVMRP (Distance Vector Multicast Routing Protocol)을 이용하는 Narada[4] 등이 있다. 그리고 DNS를 이용하며 각각의 노드들의 중계 용량과 거리를 계산하여 CBT(Core Based Tree)[5]를 구성하

는 VOID[6], 호스트들을 계층적으로 정렬하여 CBT를 구성하는 NICE[7] 등이 CBT 기반의 오버레이 멀티캐스트 프로토콜이다.

<그림 2>의 오버레이 멀티캐스트의 단점으로는 어느 한 지점/지사의 중계서버가 특정되지 않는 이유로 비활성화 되어 있을 경우에 해당 지점/지사의 모든 클라이언트들은 기본적으로 라이브 스트림을 받아 볼 수 없는 것이 가장 큰 단점이라 할 수 있다. 이러한 점을 개선하기 위해 위의 여러 프로토콜을 이용한 P2P(Peer-to-Peer) 백업 전송 형태의 연구가 많이 진행되고 있다. 그러나 P2P 방식은 해당 클라이언트들의 성능이 어느 정도 뒷받침 되어야 하는데, 기업 또는 기관내의 클라이언트의 성능은 고화질의 라이브스트림을 유니캐스트로 전송받아 멀티캐스트로 송출하기에는 해당 기업 일반 업무에 특화되어 있어 많이 모자란 것이 현실이다.

3. 본론

본 논문에서 제안할 방법은 비활성화 된 한 정점의 중계서버에서 다른 정점의 중계서버로의 최소비용경로를 구하고, 비활성화 된 해당 정점의 클라이언트들을 타 정점의 중계서버에 접속하게 하여 라이브스트림을 안정적으로 받아와 시청할 수 있게 하는데 있다. 그러기 위해서 트리 상에서의 최소비용경로를 산정하고, 각 중계서버의 중계용량과 성능을 고려하여 최종 경로를 확정하여야 한다, 또한 최소비용경로와 확정 최종 경로를 이용하여 클라이언트가 해당 정점의 중계서버에 접속하는 방식을 제시한다.

3.1 전용망 구성에 따른 최소비용경로 알고리즘 개선

일반적인 Dijkstra Algorithm[8]의 경우 그래프에

최적화 되어 있는 최소비용경로 산정을 위한 알고리즘이다. 이는 하나의 노드에서 인접하지 않은 다른 노드와의 연결 경로 산정 시 여러 가지의 경로가 존재할 수 있기 때문에 그래프 운행방식에 맞게 최소비용을 계산한다. 그러나 실제 기업/기관에서 사용되어지는 사설전용망은 그 구성이 트리 기반으로 전용망이 구성되어 있다. <그림 2>의 ①과 같이 서로 다른 지사/지점 간에 직접적인 연결은 없기 때문에 이러한 네트워크 망구성에서 Dijkstra 알고리즘을 이용해서 경로를 구할 경우에는 하나의 노드에서 인접하거나 또는 인접하지 않는 다른 노드로의 경로는 오직 한 가지 경우의 수 밖에 존재하지 않게 된다.

```
function updateDistanceAndPrevious(obp)
begin
for (i=0; i<this->numberOfNodes; i++)
begin
if( (isset(this->map[obp][i])) and
((this->map[obp][i] == this->infiniteDistance) or
(this->map[obp][i] == 0) and
((this->distance[obp] + this->map[obp][i] <
this->distance[Si] )
begin
this->distance[i] = min(this->distance[obp],
this->map[obp][i])
this->previousNode[i] = obp
end
end
end
```

그림 3. 수정된 Dijkstra 알고리즘
Figure 3. Modified Dijkstra Algorithm

그러므로 하나의 노드에서 인접하거나 인접하지 않는 다른 노드로의 여러 경로가 있다는 가정 하에 각각의 경로의 비용 합을 구하여 비용이 적은 것을 최단 경로로 선택하는 방법은 의미가 없다.

최소비용경로로는 Dijkstra 알고리즘을 이용하지

만 각 노드간의 거리(distance)를 비용으로 계산하는 대신 네트워크 대역폭을 비용으로 사용한다. 그러나 각 네트워크 대역폭은 대역폭의 크기가 크면 클수록 거리상으로는 최소가 되는 최소비용이 아닌 최대 비용 요소이므로 각 네트워크 대역폭을 각 노드를 지날 때 마다 합산해 가는 방식이 아닌 지나온 경로에서 최소 네트워크 대역폭을 선택하는 방식으로 Dijkstra 알고리즘을 변경한다.

<그림 3>은 Dijkstra 전체 알고리즘 중에서 하나의 노드와 그 이전 노드의 거리 합을 구하는 부분을 경로 중 최소의 네트워크 대역폭(Bandwidth) 값으로 선택하는 것을 보여준다.

3.2 최소비용경로 산정

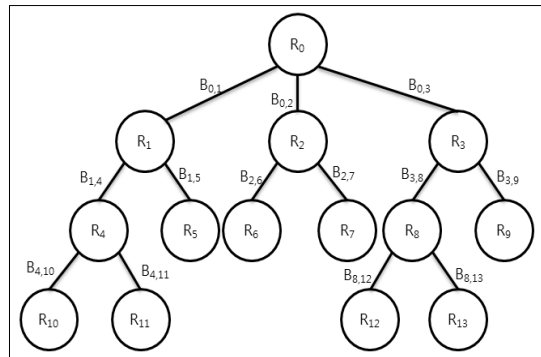


그림 4. 중계서버 경로도
Figure 4. Relay Server Route

<그림 4>는 중계서버 경로를 계산하기 위한 기본 경로도이다. 메인서버를 R0로 놓고 타 중계서버를 Rn으로 설정한다. 각 중계서버간의 노드를 네트워크 대역폭로 설정하여 B(parent,current)로 한다. 또한 <그림 4>에서 표시하지 않았지만 각 중계서버의 CPU 사용률을 Cn으로, bps로 표현되는 네트워크 트래픽을 Tn, Memory 유희율을 Mn으로 설정한다.

만일 Rm가 비활성화 되었을 시 해당 Rm 중계 서버가 속해 있는 네트워크망에 접속한 클라이언트들은 Rm을 기준으로 R1 ~ Rn로의 최소비용경로를 산정한다.

최소비용경로는 3.1에서 제시된 수정된 Dijkstra 알고리즘을 사용하여 Rm 기준으로 R1 ~ Rn까지의 경로의 길이(length)와 최소 네트워크 대역폭을 구한다. 그런 후 Dijkstra 알고리즘에서 나온 경로를 역정렬 하는 방식을 사용한다.

```

for(i=0; i<relay server count; i++ ) {
    if server is shutdown then set Ci is max num;
    set node is server#,B(parent,i),Ci,Ti,Mi,IPi
    node append to nodes;
}
result ← modified_dijkstra( nodes );
sort result order by
    node length asc,
    node bandwidth desc, bps asc,
    memory usage desc,
    cpu usage asc;
print result ip list;
    
```

그림 5. 최소비용 목록 출력 알고리즘
Figure 5. Minimum cost list output algorithm

<그림 5>는 변형된 Dijkstra 알고리즘을 적용한 Rm 네트워크망을 기준으로 한 R1 ~ Rn까지의 순서를 정하는 최소비용 목록 출력 알고리즘이다.

<그림 5>에서 Dijkstra 알고리즘으로는 경로의 길이와 최소 네트워크 대역폭을 결정하고 그 값을 바탕으로 경로의 길이는 오름차순, 네트워크 대역폭은 내림차순, 네트워크 트래픽(bps)은 오름차순, CPU 사용률은 내림차순, 메모리 유휴율은 내림차순으로 정렬한다. 이렇게 정렬된 목록은 해당하는 중계서버 IP 주소 목록으로 출력하여 클라이언트에게 전달하여 준다. <그림 5>의 알고리즘은 클라이언트 측에서 요청할 때 마다 재계산된다.

3.3 클라이언트 작동 알고리즘

클라이언트가 처음 구동되면 콘텐츠 관리시스템(CMS : Contents Management System)에 자신이 속해있는 기본 중계서버의 IP 주소를 요청하게 된다. 소속 중계서버의 IP를 받은 후 해당 중계서버에 멀티캐스트 접속 요청을 하게 된다. 요청받은 해당 중계서버는 해당 서버에 소속되어 있는 IP 대역의 클라이언트인 경우 멀티캐스트 접속을 자동 승인하게 된다. 그런 후 CMS 서버에서 Live 방송 명령이 전달되어 지면 해당 명령을 멀티캐스트 접속 요청이 허가된 클라이언트에 전달하여 라이브 스트림을 시청하게 된다.

만일 해당 클라이언트의 기본 중계서버가 여러 가지 사유로 비활성화 되어 응답하지 않는 경우에는 CMS 서버에 요청하여 위 3.2절에서와 같이 계산되어져 전달되어진 Backup Relay Server List를 이용하여 클라이언트는 해당 서버에 유니캐스트 접속을 요청한다.

요청을 받은 중계서버는 First Come, First Served 방식으로 이미 정의된 유니캐스트 최대 접속 허용 수와 비교하여 만일 현재 유니캐스트 접속된 인원수가 유니캐스트 최대 접속 허용 수 보다 경우 클라이언트에 “access denied” 신호를 송출하고 접속을 해제한다. 그러나 현재 유니캐스트 접속 수가 유니캐스트 최대 접속 허용수의 한도 이내라면 접속을 허용한다. 클라이언트는 “access denied” 신호가 수신된 경우 목록의 다음 중계서버 주소로 다시 유니캐스트 접속을 요청하는 것을 반복한다.

최대 n개의 중계서버가 설치되어 있는 경우 모든 중계서버에서 “access denied” 신호를 수신하게 되면 직접 스트리밍서버에 접속을 하여 라이브 스트림을 유니캐스트 형식으로 받아와 시청하게 된다. 스트리밍서버에 접속 시에는 별도의 인증 또

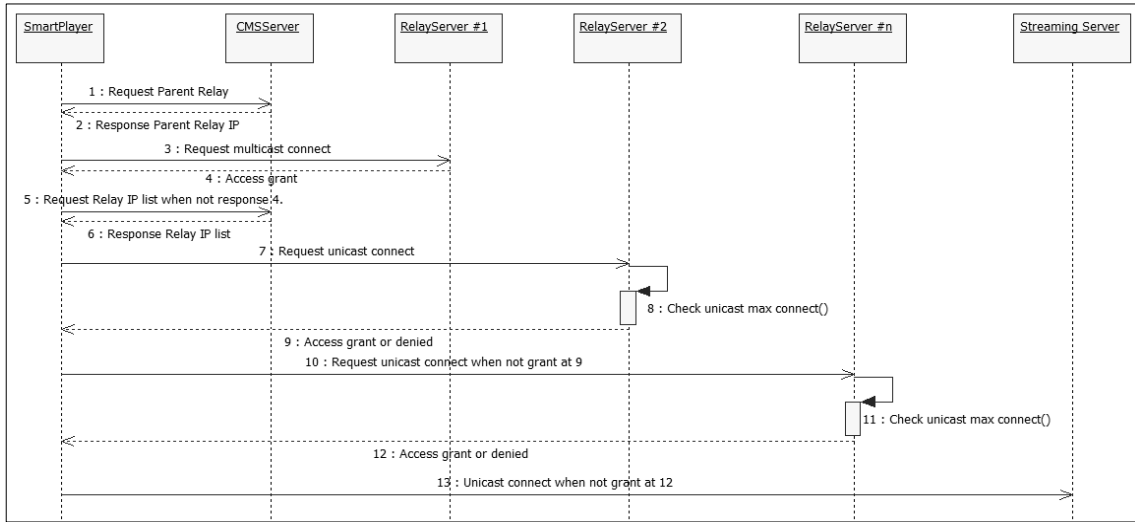


그림 6. 클라이언트의 중계서버 선택 순차다이어그램
Figure 6. Relay Server Selection Sequence Diagram for Clients

는 허가 과정을 제한하지는 않았다.

유니캐스트 형태로 접속된 클라이언트의 라이브 스트림의 시청방식은 MPEG-DASH를 이용함으로써 네트워크 상황에 따른 Dynamic Media Stream Transport 방식[9]을 이용할 수 있다. 본 논문에서는 HTTP Live Streaming[10] 방식을 채택하였다.

<그림 6>은 최소비용경로로 산정된 중계서버의 선택 방식을 보여준다. <그림 6>의 8과 11은 해당 중계서버에서 유니캐스트로 접속 가능한 클라이언트의 접속한도 개수를 미리 정의하고, 그 접속한도 개수 내에 들 때에는 접속을 허용하며 그 수를 초과한 경우 접속을 불허하는 부분이다. 이는 해당 지역의 클라이언트들이 특정 중계서버 한 곳으로 집중되는 현상을 막아주게 된다.

위와 같이 설계된 시스템의 CMS 부분은 <그림 7>과 같이 웹을 기반으로 하여 구현하였고, 중계서버 프로그램은 <그림 8>과 같이 리눅스 기반에서 C++로 구현하였으며, 클라이언트의 플레이어는 <그림 9>와 같이 DELPHI를 이용하여 윈도우 응용프로그램을 구현하였다.



그림 7. 웹기반 콘텐츠 관리 시스템
Figure 7. Web based Contents Management System

```

[root@REL2 ~]# /home/servers/SmartRelayLive/SmartRelay -e /home/servers/SmartRelayLive/SmartRelay.ini
SmartRelay v1.0 (CentOS 6.5 build 20140908)
[2014-06-09 20:41:22] eth0 IP address is 192.168.200.6
[2014-06-09 20:41:22] MAX UNICAST : 0
[2014-06-09 20:41:22] SmartDNS connect success. Local IP : 192.168.200.6
[2014-06-09 20:41:22] RPC 192.168.200.6 2014-06-09 20:41:22 FIRST
[2014-06-09 20:41:22] Server listen at 15888 port
[2014-06-09 20:41:22] Command receive from cms
192.168.101.6:192.168.200.6:192.168.40.6
[2014-06-09 20:41:22] SEND : CMS LIVE START 25 2014년 6월 9일 1번째 테스트 2014-06-09 20:35:00 00:2
0:00 1 224.15.88.1:21000 3 0 0 3 192.168.101.6 10 192.168.200.6 0 192.168.40.6 0
[2014-06-09 20:41:22] Send target clients START : 0
[2014-06-09 20:41:23] Command receive from CMS
[2014-06-09 20:41:23] SEND : CMS TEXT UPDATE 7 장시후 오후 5시 25분부터 방송을 시작합니다. (테스트) 20
14-06-09 17:23:00 00:02:00 3 192.168.101.6 192.168.200.6 192.168.40.6
[2014-06-09 20:41:23] Send target clients START : 0
[2014-06-09 20:41:24] Live stream check : Pass - with Audio - (/usr/bin/rtpdump -r rtmp://192.168.99.
163/live/channel181.sdp -e /dev/null -i timeout 10 -s stop 0.01)
[2014-06-09 20:41:24] Multicast process 13950 launched.
[2014-06-09 20:41:32] RPC 192.168.200.6 2014-06-09 20:41:32 LIVE 0 1 0 0 2 0 0 0 3 0 0 4 0 0 0 5 0
0 0 0 0 0 7 0 0 8 0 0 0 9 0 0 0 10 0 0
[2014-06-09 20:41:32] RPC 192.168.200.6 2014-06-09 20:41:32 LIVE 0 1 0 0 2 0 0 0 3 0 0 4 0 0 0 5 0
0 0 0 0 7 0 0 0 8 0 0 0 9 0 0 0 10 0 0
    
```

그림 8. 중계서버 데몬 프로그램
Figure 8. Relay Server Daemon Program



그림 9. 클라이언트 플레이어 프로그램
Figure 9. Player Program for Clients

4. 결 론

본론에서 Dijkstra 알고리즘은 변형하여 해당 네트워크 대역폭을 기준으로 모든 중계서버에 대해 최소비용경로를 산정하는 알고리즘을 설계하였다. 그런 후 클라이언트는 해당 클라이언트의 기본 중계서버가 작동을 하지 않는 경우 다른 중계서버로 유니캐스트 접속을 요청하고 해당 요청을 받은 중계서버는 해당 유니캐스트 최대 접속 허용수를 기준으로 접속을 허가하거나 불허하는 방식을 취하였다. 위와 같이 함으로써 중계서버가 작동을 하지 않을 경우 기존에는 해당 클라이언트들이 메인 스트리밍서버로 모두 접속하여 스트리밍서버에서 정상적으로 서비스를 할 수 없었던 문제를 해결하고 네트워크 부하를 최대한으로 분산 처리 할 수 있다. 또한 중계서버가 작동하지 않게 된 지역의 모든 클라이언트들이 특정 다른 지역의 중계서버에 집중하는 것을 피할 수 있게 되어 기관 내의 모든 클라이언트가 라이브스트림을 안정적으로 시청할 수 있다.

References

- [1] R. Doverspike, G. Li, K. N. Oikonomou, K. K. Ramakrishnan, R. K. Sinha, D. Wang and C. Chase, *Designing a reliable IPTV network*, Internet Computing, IEEE, 13.3, pp. 15-22. 2009.
- [2] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy and A. E. Mohr, *Chainsaw: Eliminating trees from overlay multicast*. In *Peer-to-peer systems IV*. pp. 127-140. Springer Berlin Heidelberg. 2005.
- [3] Y. D. Chawathe, *Scattercast: an architecture for internet broadcast distribution as an infrastructure service*. Doctoral dissertation, University of California. 2000.
- [4] Y. H. Chu, S. G. Rao, S. Seshan and H. Zhang, *A case for end system multicast*. *Selectd areas in communications*. IEEE Journal on, 20(8), pp. 1456-1471. 2002.
- [5] A. Ballardie, *Core based trees (CBT) multicast routing architecture*. 1997.
- [6] P. Francis, *Yoid: Extending the internet multicast architecture*. 2000.
- [7] S. Banerjee, B. Bhattacharjee and C. Kommareddy, *Scalable application layer multicast*. Vol. 32, No. 4, pp. 205-217. ACM. 2002.
- [8] S. Skiena, *Dijkstra's algorithm. Implementing discrete mathematics: Combinatorics and graph theory with mathematica, reading, MA: Addison-Wesley*, pp. 225-227. 1990.
- [9] T. Stockhammer, *Dynamic adaptive streaming over HTTP--: standards and design principles*. In *Proceedings of the second annual ACM conference on Multimedia systems*. pp. 133-144. ACM. February, 2011.
- [10] K. Honkanen, *HTTP Live Streaming*. 2011.

전용망 최소비용경로를 이용한 고품질 백업 전송 시스템 설계 및 구현

정명기, 김재웅

공주대학교 컴퓨터공학과

요 약

Multicast IPTV live streaming을 사용할 수 없는 기업(기관)에서는 각 지사(지점)에 중계서버를 두고 본사(본점)의 스트림서버에서 라이브스트림을 유니캐스트 형태로 전송받는다. 이를 각 지사(지점)내에서 다시 멀티캐스트 형태로 변환하여 클라이언트에 전송하는 오버레이멀티캐스트 방식을 사용한다. 이때 각 중계서버가 정상상태가 아닌 경우에는 해당 지사(지점)의 모든 클라이언트는 라이브스트림을 시청 할 수 없는 상황이 발생한다. 이런 상황을 해결하기 위해 각 지사(지점)을 기점으로 하는 트리구조를 생성한다. 각 기점에서 다른 지사(지점)들 간의 최단 경로를 구하고 그에 따른 백업 전송 경로를 선택한다. 비정상인 중계서버가 소속된 클라이언트들은 이렇게 선택된 백업 전송 경로를 이용하여 유니캐스트 형태로 타 지사(지점)에 접속하여 라이브스트림을 전송받는다. 이렇게 함으로써 안정적으로 라이브스트림을 시청할 수 있다.



Jae-Woong Kim received the bachelor's degree and the MS degree in the Department of Computer Engineering from the Jungang University in 1983 and 1988, respectively.

He received the Ph.D. degree in the Department of Computer Engineering from Daejun University in 2000. He has been a professor in the Department of Computer Engineering at Kongju National University since 1992. His current research interests include software engineering. He is a life member of the KKITS.

E-mail address: yjkim@kongju.ac.kr



Myeong-Ki Jung received the bachelor's degree in the Department of Electric Engineering Education from the Chungnam National University in 1994. He

received the MS degree in the Department of Computer Multimedia Engineering from Kongju National University in 2004. His current research Interests include software engineering, web engineering, IPTV. He is a regular member of the KKITS.

E-mail address: bright@kongju.ac.kr