FLUTE를 이용한 멀티캐스트 파일 전송에서 Late-Join과 파일 복구 방안*

김영준° 성백동 홍진표
한국외국어대학교 정보통신공학과
{ddanggae°, iceboy98, jphong}@hufs.ac.kr

Late-Join and repair schemes in multicasting files by the FLUTE protocol

Young-Jun Kim°, Baek-Dong Seong, Jin-Pyo Hong
Dept. of Information Communication Engineering, Hankuk University of Foreign Studies

요 약

최근 wired 및 wireless 환경에서 멀티미디어 데이터 전송에 대한 수요가 증가하고 있다. 또한 네트워크의 대역폭 효율성을 위해 멀티캐스트 전송이 요구되었고, 신뢰적인 멀티캐스트 전송 프로토콜인 RMT 프로토콜이 등장하게 되었다. FLUTE는 ALC 기반의 어플리케이션으로 LAN 환경 및 3GPP 및 DVB-H에서 멀티캐스트 콘텐츠 전송을 위해 사용하고 있다. FLUTE는 동일한 멀티캐스트 세션에서 FDT 인스턴스를 전송한 후에 콘텐츠를 전송한다. 그러나 FDT 인스턴스를 수신하지 못하면 콘텐츠를 수신할 수 없기 때문에, FDT 인스턴스 전송 이전에 멀티캐스트 세션에 Join해야 하는 문제가 발생한다. 본 논문에서는 이러한 문제를 해결하는 FLUTE을 이용한 멀티캐스트 콘텐츠 전송에서 Late-Join과 Late-Join 시 파일 복구 방안을 제안하였다.

1. 서 론

최근 멀티미디어 콘텐츠 전송에 대한 수요가 급격히 증가하고 있다. 또한 사용자의 이동성을 지원하는 wireless 환경에서도 이러한 멀티미디어 데이터의 전송에 대한 요구가 증가하고 있다. 따라서 wired 및 wireless 환경 모두를 만족하는 멀티미디어 데이터 전송을 효율적으로 제공하기 위한 방법들이 활발히 연구되고 있다. 또한 네트워크 대역폭의 효율성을 증가시키기 위해 멀티캐스트 서비스의 필요성이 요구되었고, 신뢰성을 제공하지 않는 IP 멀티캐스트에 "에러제어 및 혼잡제어를 통한 멀티캐스트 데이터 전송에 신뢰성을 제공"하는 신뢰성 있는 수송계층 멀티캐스트 프로토콜(이하 RMT)에 대한 요구도 증가하고 있다.

FLUTE는 RMT 프로토콜 중에 하나인 FEC를 이용한 ALC 기반의 어플리케이션으로 다수의 수신자들에게 단 방향 파일 전송을 목적으로 한다. 이런 FLUTE는 일반적인 LAN 환경뿐만 아니라, 3GPP 및 DVB-H에서 멀티미디어 콘텐츠의 다운로드에 이용하고 있다.

본 논문에서는 FLUTE을 이용한 멀티캐스트 전송에서 Late-Join 및 파일 복구의 한 방안으로 FDT 인스턴스 서버를 통한 HTTP 프로토콜의 FDT 인스턴스 전송을 제안한다. 본 논문의 2장에서는 현재 진행되고 있는 FLUTE의 신뢰적인 전송 방안과 3GPP의 파일 복구 절차를 설명하고, 3장에서는 FLUTE 멀티캐스트 세션에 Late-Join 할 수 있는 새로운 방안을 제안한다. 4장에서는 Late-Join 시의 파일 복구 절차를 다루고, 5장에서 결론을 맺도록 한다.

2. 관련 연구

2.1 FLUTE의 신뢰적인 전송 방안

FLUTE(File Delivery over Unidirectional Transport)은 ALC 기반의 단 방향 파일 전송을 위한 어플리케이션으로 파일을 전송하는데 필요한 여러 가지 정보들과 파라미터들에 대해서 정의하고 관리한다.

FLUTE에서 송신자의 동작 절차는 FDT 인스턴스를 전송한 후, 콘텐츠들을 전송한다. FDT 인스턴스는 전송할 콘텐츠들에 대한 정보를 가지고 있기 때문에 콘텐츠를 전송 받기 위한 수신자들은 FDT 인스턴스를 먼저 수신해야 한다[1].

FLUTE에서 송신자는 신뢰성을 제공하기 위해서 ALC에서 제공하는 FEC를 사용하기 때문에 데이터를 전송할 때, 실제 데이터와 복구 데이터를 같이 전송한다. 이 때, FEC 코드를 사용하면 동일한 패킷을 사용해서 여러 수신자의 다양한 손실 패턴을 복구할 수 있다. 그리고 피드백과 재전송 없이 데이터를 복구할 수 있다[2].

송신자는 각각의 파일을 전송할 때 TOI(Transmission Object Identifier)로 구분하고, ALC 헤더 안에 SBN(Source Block Number)과 ESI(Encoding Symbol Identifier)를 포함해서 수신자들에게 전송한다. 이 때, 하나의 파일에서 SBN과 ESI의 조합을 통한 FEC 페이로드는 유일하게 된다[3].

그리고 FLUTE는 가장 간단한 compact FEC 코드와 Reed-solomon과 같은 small FEC 코드, 그리고 LDPC와 같은 large FEC 코드 등을 전송 환경에 맞게 다양하게 적용할 수 있다[4].

2.2 3GPP/MBMS에서 FLUTE 전송과 파일 복구 절차

3GPP에서 멀티캐스트 콘텐츠의 다운로드를 위하여 FLUTE를 사용한다. 3GPP에서 FLUTE는 혼잡제어 빌딩블록을 사용하지 않고, FEC 빌딩블록은 No-Code FEC를 사용하여 FEC 인코딩을 하지 않는다. 3GPP에서 콘텐츠 다운로드 전송 중에 콘텐츠의 일부가 에서는 손실 되었을 경우, FEC 코드를 사용하지 않기 때문에, 파일 복구 절차를 통해서 파일을 복구한다[5].

수신자는 모든 콘텐츠의 수신이 끝나면 멀티캐스트 세션을 나온다. 그 후 콘텐츠의 손실된 부분을 확인해서 파일 복구 서버에

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구 결과로 수행되었음

게 HTTP 프로토콜을 사용해서 복구 요청 메시지를 전송한다. 파일 복구 서버는 요청한 콘텐츠의 손실된 부분을 복구 응답 메시지를 통해서 전송하거나, 다른 파일 복구 서버의 주소를 알려준다. 많은 수신자가 복구 요청 메시지를 보낼 경우나 파일의 대부분을 다시 전송해야 할 경우에는 MBMS 다운로드 세션의 주소를 알려준다[5].

- GET www.example.com/news/latest.3gp?mbms-*rel6-FLUTE-repair*&SBN=5;ESI=12&SBN=20;ESI=27 HTTP/1/1

위와 같이 수신자는 파일 복구 요청 메시지를 보낸 후, 파일 복구 서버에게 파일 복구 응답 메시지, MBMS 다운로드 세션의 주소, 다른 복구 서버의 주소, HTTP 에러 코드 등의 응답 메시지를 받는다[5].

HTTP Header		
Length Indicator	FEC Payload ID	Encoding Symbols
Length Indicator	FEC Payload ID	Encoding Symbols
Length Indicator	FEC Payload ID	Encoding Symbols

그림 1. 파일 복구 응답 메시지의 데이터 형태

그림 1은 파일 복구 서버의 응답 메시지에 대한 데이터 전송 형태이다. 파일 복구 요청 메시지를 전송한 수신자에게 콘텐츠의 손실된 모든 부분을 한번에 HTTP 프로토콜을 사용해서 전송한다.

파일 복구 서버를 사용하면, FDT 인스턴스를 수신한 후의 콘텐츠에 대해서 신뢰성을 제공할 수 있다. 즉, 콘텐츠를 전송 받는 중에 발생하는 손실에 대해서는 파일 복구 서버를 통해서 재전송 받게 된다.

3. FLUTE 멀티캐스트 세션에 Late-Join 방안 제안

3.1 FLUTE의 절차 및 문제점

수신자 측 FLUTE의 동작 절차는 멀티캐스트 세션에 대한 정보를 가지고 있는 SDP를 수신한 후에 이를 통해서 원하는 콘텐츠의 멀티캐스트 세션에 Join한다. 그 후에 FDT 인스턴스를 수신해야 콘텐츠를 수신할 수 있다. 콘텐츠 전송되는 중에 Join하는 수신자는 FDT 인스턴스를 수신하지 않았기 때문에 콘텐츠의 정보를 확인할 수 없다. 그렇기 때문에 콘텐츠를 수신할 수 없고, 멀티캐스트 세션에서 대기해야 한다. 따라서 기존의 FLUTE 수신자는 송신자 측에서 FDT 인스턴스가 다시 전송된 이후부터 콘텐츠의 수신이 가능하다. 이처럼 수신자는 수동적인 형태로 단지 FDT 인스턴스를 기다려야 하는 문제가 발생한다[1].

3.2 FLUTE 멀티캐스트 세션에 Late-Join 방안

FLUTE의 콘텐츠 서버는 멀티캐스트 세션에서 콘텐츠를 계속 전송하기 때문에, HTTP 프로토콜을 연결하고 FDT 인스턴스를 전송할 경우 많은 오버헤드를 가진다. 따라서 FDT 인스턴스 서버를 분리해서 Late-Join을 할 경우 FDT 인스턴스 서버에 HTTP 프로토콜을 사용해서 FDT를 수신하고, 멀티캐스트 세션에 Join해서 콘텐츠를 수신한다.

그림 2는 FDT 인스턴스 서버와 콘텐츠 서버를 분리한 FLUTE의 망 구성도이다. 그림과 같이 콘텐츠 서버는 FDT 인스턴스를 전송한 후 콘텐츠들을 순차적으로 전송한다. Late-Join을 한 수신자는 FDT 인스턴스 서버에게 FDT 인스턴스를 수신한 후 멀티캐스트 세션에서 콘텐츠를 수신한다. 이 때, 수신자는 FDT 인스턴스 서버의 위치를 알고 있어야 하기 때문에 콘텐츠 서버는 SDP를 전송할 때 속성을 나타내는 a 필드를 추가해서 FDT 인스턴스 서버의 URI를 표시한다[6].

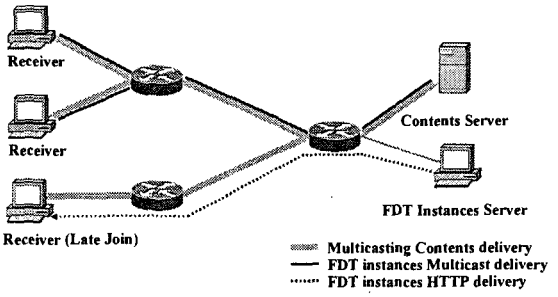

그림 2. FDT 인스턴스 서버를 포함한 FLUTE 망 구성도

아래는 SDP의 디스크립션의 일부로 FDT 인스턴스를 정의하는 부분이다.

- *a=FDT-instance: http://fdt.example.com/FDT.xml*

Late-Join하는 수신자는 SDP를 확인한 후 FDT 인스턴스 서버에게 FDT 인스턴스를 요청하고, FDT 인스턴스를 수신한 이후 멀티캐스트 세션에서 콘텐츠를 수신한다.

아래는 수신자가 FDT 인스턴스 서버에게 보내는 FDT 인스턴스 요청 메시지이다.

- GET fdt.example.com/FDT.xml HTTP/1.1

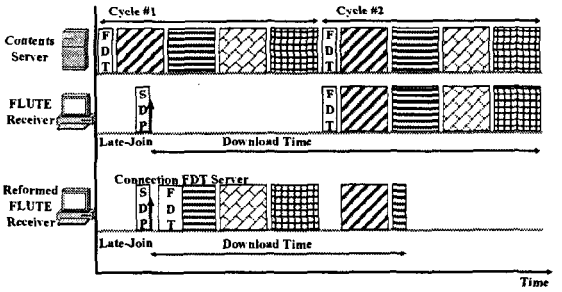

그림 3. Carousel 서비스 모델에서의 FLUTE의 Late-Join

그림 3은 FLUTE Carousel 서비스 모델에서 기존의 FLUTE와 FDT 인스턴트 서버를 이용한 FLUTE의 Late-Join 후의 전송 시간을 나타낸다. 기존 FLUTE의 수신자는 FDT 인스턴스를 수신하기 전까지 대기한다. FDT 인스턴스를 수신한 이후에 콘텐츠를 수신한다.

FDT 인스턴스 서버를 사용한 FLUTE의 수신자는 HTTP 프로토콜을 통해서 FDT 인스턴스를 수신한 후, 전송중인 콘텐츠를 수신한다. 그리고 이후의 콘텐츠 전송에서 전송 받지 못한 부분만을 수신한다. 이때 HTTP 프로토콜의 TCP 연결을 하는 오버헤드가 있지만, 기존의 FLUTE 보다 더 좋은 전송 시간을 가진다.

4. Late-Join 후의 파일 복구 절차

FLUTE는 지나 치게 많은 패킷이 손실하여 FEC 코드로 복구할 수 없을 때는 콘텐츠의 수신을 포기하고 멀티캐스트 세션을 나오게 된다. 3GPP의 FLUTE는 많은 패킷을 손실할 경우에 파일 복구 서버를 통한 재전송을 한다. Late-Join을 한 수신자가 HTTP 프로토콜을 사용해서 FDT 인스턴스를 수신하면 전송 중인 콘텐츠를 수신할 수 있다. 그리고 복구 서버를 이용해서 전송 받은 콘텐츠의 이전 부분과 손실된 부분을 재전송 받는다.

FLUTE는 SDP를 사용해서 파일 복구 서버의 URI의 정보를 전

송한다. 아래는 SDP의 디스크립션의 일부로 파일 복구를 정의하는 부분이다. 세션의 속성을 나타내는 a 필드를 사용하고, 속성의 값은 파일 복구 리스트를 가지고 있는 메타데이터 URI를 지정한다[6].

- a=File-Repair:
  http://flute.example.com/procedure.xml

```
<?xml version="1.0  encoding="UTF-8 ?>
<associateProcedureDescription
        xmlns="urn:flute:params:xml:ns:associatedProcedure"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <postFileRepair
                offsetTime="5
                randomTimePeriod="10 >
                <serverURI>"http://filerepair1.example.com/"</serverURI>
                <serverURI>"http://filerepair2.example.com /"</serverURI>
                <serverURI>"http://filerepair3.example.com /"</serverURI>
        </postFileRepair>
        <bmFileRepair sessionDescriptionURI=
                "http://flute.example.com/session.sdp"/>
</associatedProcedureDescription>
```

그림 4. 복구 서버의 리스트를 가지고 있는 procedure.xml

그림 4는 파일 복구 서버의 메타데이터로, 파일 복구 서버들의 URI와 멀티캐스트 세션의 정보를 가지고 있는 SDP의 URI를 가지고 있다. 수신자는 멀티캐스트 세션에서 나온 후에 콘텐츠를 복구한다. 이때 콘텐츠의 손실이 적으면 복구 서버 중 하나를 선택해서 복구한다. 그리고 손실이 클 경우에는 멀티캐스트 세션에 다시 Join해서 콘텐츠를 수신한다.

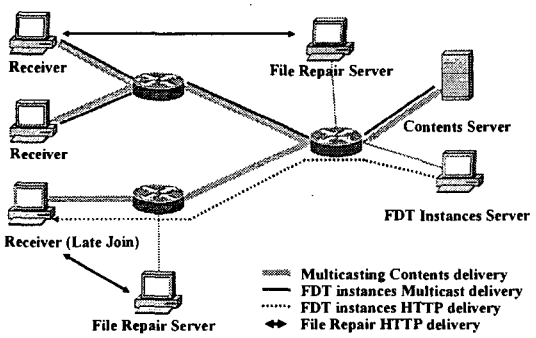

그림 5. FDT 인스턴스 및 파일 복구 서버를 포함한 FLUTE 망

그림 5는 FDT 인스턴스 서버와 파일 복구 서버를 포함한 전체 FLUTE의 망을 나타낸다. 파일 복구 서버는 콘텐츠의 전송이 끝난 후에 손실된 콘텐츠의 부분을 HTTP 프로토콜을 사용해서 재전송 한다.

그림 6은 FLUTE Carousel 서비스 모델에서 Late-Join 한 수신자에 대한 복구 성능을 나타낸다. 그림은 일반적인 FLUTE 멀티캐스트 채널과 파일 복구 채널로 구분해서 콘텐츠를 수신하고 콘텐츠를 복구한다.

기존 FLUTE의 수신자는 패킷의 손실이 있어도 FEC 디코딩을 사용해서 빠르게 패킷을 복구한다. 그러나 2번째 콘텐츠와 같이 패킷 손실이 클 경우에는 FEC 디코딩으로 복구하지 못하기 때문에 콘텐츠를 포기한다. 이처럼 손실이 큰 환경에서는 FEC 코드를 사용해도 신뢰적으로 수신할 수 없다.

3GPP의 FLUTE 수신자는 파일 복구 서버에게 파일 복구 요청 메시지를 전송해서 손실된 패킷을 재전송 받는다. 패킷의 손실에 관계없이 3GPP의 FLUTE 수신자는 모든 콘텐츠를 신뢰적으로 수신한다. 그러나 파일 복구 서버와 연결하는 과정에서 오버헤드가 발생한다.

FDT 인스턴스 서버를 사용한 FLUTE의 경우에는 Late-Join을 한 시점에서 콘텐츠를 수신한다. 그렇기 때문에 받지 못한 콘텐츠의 일부분과 패킷의 손실된 부분을 파일 복구 서버를 통해서 재전송 받는다. 그렇기 때문에 모든 콘텐츠를 신뢰적으로 수신하고, 콘텐츠의 전송 시간도 짧아지는 장점을 가진다.

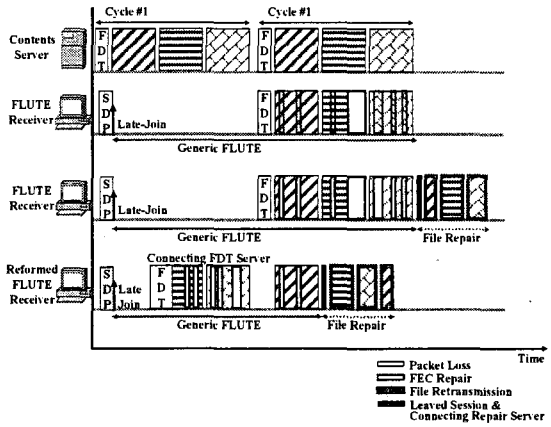

그림 6. Carousel 서비스 모델에서 Late-Join 후의 복구 성능

5. 결론 및 향후 연구과제

본 논문은 HTTP 프로토콜을 사용한 FDT 인스턴스의 전송을 통한 Late-Join과 Late-Join 시의 파일 복구 방법을 제안하였다. 수많은 수신자들에게 콘텐츠를 멀티캐스트로 전송할 때 Late-Join을 통한 전송 시간의 단축은 콘텐츠 전송 서비스 측면에서 큰 장점을 가진다.

향후 연구과제로 본 논문에서 제안한 HTTP 프로토콜을 사용한 FDT 인스턴스 전송에 대한 구현 및 시뮬레이션을 통하여 기존의 FLUTE와의 성능을 비교할 계획이다.

참고 문헌

[1] Paila, T., Luby, M., Lehtonen, R., Roca, V., and R. Walsh " FLUTE - File Delivery over Unidirectional Transport" , IETF RMT WG, RFC 3926, October 2004.

[2] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, " The Use of Forward Error Correction (FEC) in Reliable Multicast" , IETF RMT WG, RFC 3453, December 2002.

[3] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., and J. Crowcroft, " Asynchronous Layered Coding (ALC) Protocol Instantiation" , IETF RMT WG, RFC 3450, December 2002.

[4] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, " Forward Error Correction (FEC) Building Block" , IETF RMT WG, RFC 3452, December 2002.

[5] "Multimedia Broadcast/Multicast Service (MBMS) Protocols and codecs(Release 6)", 3GPP TR 26.946 V1.0.0, September 2005.

[6] Handley, M., Perkins, C. and E. Whelan, " Session Announcement Protocol" , RFC 2974, October 2000.