

무선 LAN 환경에서 ALC와 LDPC FEC를 이용한 신뢰적인 멀티캐스트 전송 프로토콜의 구현 및 성능 평가

백일우⁰ 성백동 홍진표
한국외국어대학교

{steigensonne⁰, iceboy98, jphong }@hufs.ac.kr

Implementation and evaluation of reliable multicast protocol with the ALC and LDPC FEC over wireless LAN environment

Il-Woo Paik⁰, Baek-Dong Seong, Jin Pyo Hong

Dept. of Information Communication Engineering, Hankuk University of Foreign Studies

요 약

ALC(Asynchronous Layered Coding)는 LCT(Layered Coding Transport)기반의 폭넓은 확장성과 다양한 전송속도로 신뢰성 있는 멀티캐스트 전송을 주 목적으로 한다. ALC는 다른 RMT 프로토콜과는 달리 재전송을 하지 않고 receiver-driven 형태의 혼잡제어를 하기 때문에 신뢰성을 제공하기 위해 FEC(Forward Error Correction) 스킴을 이용한다. 본 논문은 신뢰적인 멀티캐스트 전송 프로토콜을 위해 ALC와 FEC 스킴 중 하나인 LDPC 코드를 이용하여 구현한다. 인코딩 비율(Encoding ratio)에 따라 일정 수준 이상의 복구 성공 확률에 대해 최대 허용 가능한 패킷 손실률을 측정하여 신뢰치를 측정한다. 마지막으로 일대다의 파일 전송 시, TCP와 비교하여 본 구현물의 유연성과 효율성에 대해 분석 및 평가한다.

1. 서 론

최근 일대 다의 멀티미디어 콘텐츠 전송에 대한 요구들이 점점 늘어남에 따라서 멀티캐스트에 대한 중요성은 증가하고 있다. 이러한 요구 사항은 빠르게 성장하고 있는 무선 네트워크에서도 중요한 키워드가 되고 있다. 하지만 path-loss, shadowing, multipath fading과 같은 특징들이 있는 무선 환경 하에서는 패킷 손실이 빈번히 일어 날 수 있기 때문에 그에 대한 신뢰성이 요구된다. 이에 IETF RMT WG에서는 무선 환경에 적합한 프로토콜로서 FEC를 기반으로 한 ALC(Asynchronous Layered Coding)를 제안하고 있다. 일반적으로 무선 환경에서는 피드백 과정이 쉽지 않기 때문에, 수신측에서 자체적으로 패킷을 복구하도록 하는 FEC(forward Error Control)를 이용하도록 하고 있다. 따라서 본 논문에서는 무선 LAN 환경에서 LDPC FEC 기반의 ALC를 이용하여 신뢰성 있는 멀티캐스트 프로토콜을 구현하며, 무선환경의 실제적인 패킷 손실에 대해 알아보고 TCP와 비교하여 멀티캐스트 파일 전송의 효율성에 대해 평가 및 분석한다.

2. 관련 연구

2.1 RMT WG

IETF RMT WG은 end-to-end간의 멀티캐스트 전송을 하는데 있어서 신뢰성을 보장하는 것을 목적으로 초기 연구방향은 대용량 데이터, 일대 다 멀티캐스트에 초점을 맞추고 한 가지 모델을 통해 모든 요구 조건을 만족시키고자 하였지만 여러 종류의 멀티캐스트 어플리케이션의 특성상 하나의 프로토콜로 다

양한 요구사항을 만족시키는 것이 적합하지 않아서, 현재는 두 개의 protocol Instantiations(PI)에 대해서 표준화하고 있다

2.2 ALC(Asynchronous Layered Coding)

ALC는 RMT에서 표준화하고 있는 두 가지 PI 중에 하나로서 단일 송신자가 여러 수신자들에게 파일 전송하는 동안에 여러 채널들을 통해 신뢰성과 혼잡 제어가 가능하도록 표준화한 프로토콜이다. ALC에서는 에러제어를 위한 FEC Building Block (BB)과 혼잡 제어를 위한 WEBRC BB, 세션 및 채널 관리를 위한 LCT BB를 결합한 것으로, 서비스와 필요에 따라 빌딩 블록을 구성할 수 있도록 되어 있다.[1]

2.3 FLUTE(File Delivery over Unidirectional Transport)

FLUTE는 ALC 기반에서 단 방향 파일 전송을 위한 어플리케이션으로, 파일을 전송하는데 있어서 필요한 파일 자체에 대한 정보들이나 전송을 위한 정보들에 대해서 정의해 놓은 프로토콜이다. 파일을 전송하는데 있어서 먼저 FDT 인스턴스의 송신을 통해서 전송에 필요한 정보와 전송할 파일의 여러 속성들에 대한 정보를 보낸 후에 파일을 전송한다.[2]

2.4 3GPP MBMS(Multicast Broadcast multimedia Service)

3GPP MBMS에서 서비스를 하는 과정에서 서비스에 대한 정보를 담은 광고를 할 때나 세션에 참가한 유저들에게 다운로드 콘텐츠를 제공하는 프로토콜로서 ALC를 기반으로 한 FLUTE를 사용하고 있다. 그 이유는 FLUTE는 수신자의 수와 종류에 상관 없이 다양한 크기의 파일들을 전송할 수 있고, 대역폭의 낭비를

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학IT연구센터 육성·지원사업의 연구결과로 수행되었음

방지할 수 있으며, 파일에 대한 지속적인 업데이트가 가능한 특성을 가지고 있기 때문이다.[3]

3. LDPC와 ALC를 이용한 멀티캐스트 프로토콜의 구현

3.1 ALC 프로토콜의 특징 및 구성

ALC는 컨텐츠 전달 프로토콜로서 수많은 수신자들에게 매우 효율적인 전송을 가능하게 한다. 또한 ALC는 단방향성이고 필요에 따라 제한적 전송이 가능하며 피드백을 위한 특정 채널 및 자원이 필요하지 않기 때문에, 무선환경 뿐만 아니라 위성환경이 없기 때문에 신뢰성을 위해 FEC코딩 스킴을 전체적으로 또는 부분적으로 적용하여 신뢰적인 서비스를 제공한다. 전달하고자 하는 Object는 적용되는 FEC 스킴에 따라 FEC 인코딩을 거쳐 소스 블록과 FEC 스킴을 통해 만들어진 추가적인 심볼들을 구성하여 전달된다. ALC의 세션은 하나 이상의 채널로 구성되며, 여러 수신자들은 네트워크 상태에 따라 세션내의 채널을 선택하여 원하는 object를 수신한다. 수신자들은 자신의 컨텐츠 수신에만 전념할 수 있으며, 다른 수신자의 상태나 패킷 손실에도 거의 영향을 받지 않는다. 따라서 ALC는 높은 견고성을 가지며, multi-layered 전송을 이용하여 안정적인 컨텐츠 다운로드에 유효하게 수월하다.

3.1.1 Layered Coding Transport(LCT) Building block

LCT BB는 세션내의 수신자에게 전달할 기본적인 정보의 내용과 특징 그리고 LCT 헤더와 확장 헤더의 형식들을 정의한다. ALC/LCT 세션은 TSI(Transport Session Identifier)로 식별하며 송신자 IP주소와 UDP port로 세션 내의 채널을 식별하는데 사용한다. LCT BB는 TOI(Transport Object ID)를 통해서 각각의 파일 object를 구분한다. 또한 LCT는 CCI(Congestion Control Information) 필드를 두어 사용여부 및 CC BB정보와 관련 정보를 포함한다. LCT는 확장 헤더를 통해 부가정보 및 FEC 관련정보를 추가할 수 있다.[4]

3.1.2 FEC Bluiding Block

FEC BB는 특히 중요한 부분으로 ALC의 확장성과 신뢰성을 제공한다. FEC BB는 FEC 코드 스킴을 가리키는 FEC encoding ID와 FEC instance ID를 정의한다. LCT 헤더내부의 FEC payload ID는 인코딩 심볼들과 복구 심볼들을 나타내며, FEC object transmission information(FEC OTI)는 파일 object의 총길이, 심볼 사이즈 등을 나타낸다. FEC OTI는 수신자와 송신자 사이에 파일 전송이 일어나기 전에 수신자에게 반드시 전달되어야 한다. 이를 통해 수신자는 송신자가 보낼 패킷이 어떠한 형태의 FEC 스킴과 방식으로 구성되어 있는지를 알 수 있다.

3.2 ALC의 LDPC(Low Density Parity Check) 코드 적용

무선환경에서 ALC는 재전송을 하지 않기 때문에 FEC 스킴은 반드시 요구된다. FEC는 Block의 크기에 따라 Small Block FEC Codes와 Large Block FEC Codes로 나뉜다. Block 크기가 작은 FEC 코드는 Reed-Solomon erasure (RSE), Raptor FEC, Tornado FEC 등을 들 수 있고, block 크기가 큰 경우는 LDPC가 대표적인 예이다. 일반적으로 이용되는 무선환경에서 ALC의 서비스 다운로드 모델의 파일 크기는 보통 500kb 내외 또는 1Mb 정도의 파일로 가정하고 있다. 하지만 컨텐츠의 크기는 점차 증가하며, 고품질의 다운로드 서비스로 인한 대용량 파일 전송의 필요성이 증가함에 따라 Large Block 크기의 FEC 스킴이 요구된다. 따라서 무선환경에서의 다운로드 시 1Mb 이상의 파일 크기에 대한 서비스가 예상되고 대용량 컨텐츠 전송을 위해서 Block 사이즈가 큰 LDPC FEC 스킴이 요구된다. LDPC

코드 스킴은 LDPC Staircase와 LDPC Triangle 두 가지가 있고, 소스와 인코딩의 관계는 Parity Check Matrix로 나타낸다. Parity Check Matrix는 인코딩 심볼들 즉, 소스 심볼과 복구 심볼들을 의미하여 이 심볼들의 연관관계에 대해 나타낸다. 이를 바탕으로 수신자는 같은 행렬을 이미 가지고 있어, 수신한 심볼들을 가지고 있는 행렬을 기반으로 다른 패킷의 정보를 알아낼 수 있다. [5]

3.3 ALC 디자인

본 구현물은 그림 1에서와 같이 총 4개의 쓰레드로 구성되며, 송신자/수신자의 기능적 측면에서 구분한다면, 송신자는 Output/Timer 쓰레드, 수신자는 Receive/Input 쓰레드로 나뉘어진다. Application-Timer-Output 쓰레드 간의 연결선들은 어플리케이션 또는 타이머 쓰레드가 Output 쓰레드에게 메시지 전달하기 위한 내부 포트로서 사용자가 API를 호출하였을 때 Output 쓰레드에게 사용자 요청을 알려거나 Timer 쓰레드가 타임아웃 발생시 이를 알리기 위한 용도로 사용한다. Timer 쓰레드는 FEC 인코딩 비율에 따라 패킷 전송주기마다 타임아웃을 Output 쓰레드에게 전달하며 추후 다중 세션을 지원할 경우를 고려하여 각 타임아웃을 Delta list로 관리하도록 구현하였다. 그리고 Output 쓰레드는 데이터를 송신 버퍼(send buffer)에 1024 바이트 단위로 복사한 후 FEC 인코딩을 하여 ALC 패킷을 전송한다.

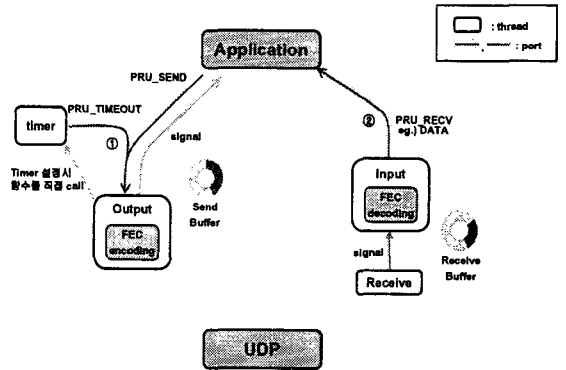


그림 1 ALC 프로토콜 구현 세부 구조

수신자의 Receive 쓰레드는 단순히 패킷 수신 역할만 하는 쓰레드로서 실제 ALC 패킷의 처리는 input 쓰레드에서 담당한다. 수신한 패킷은 해당하는 소스 블록의 번호와 매칭 과정을 거쳐 FEC 디코딩 작업을 거친다. 만약 수신자가 late join을 하였을 경우, 첫 Source Block Number에 대한 패킷을 재구성 할 수 없을 때에는 해당 패킷을 제거하고, 다음 Source Block Number에 해당하는 ALC 패킷부터 처리한다. FEC 디코딩을 통해 성공적으로 Source Block을 재구성하면 이를 어플리케이션에게 알리기 위해 PRU_RECV 포트 메시지를 내부 포트로 전송한다. 이번 구현에서 CC는 제외되었고, LDPC FEC 코드는 MCLv3 오프 소스[6]를 이용하였다.

4. ALC 성능 평가 및 분석

4.1 ALC/LDPC의 시스템 환경 및 환경 요소

ALC는 무선환경 또는 위성환경에서 다수의 수신자에게 컨텐츠를 전달하기 위해 디자인되었다. 본 논문의 실험에서는 대표적 무선환경인 802.11 Ad-hoc 환경에서 1개의 송신자에서 파일을 전송하여 하나 또는 다수의 수신자를 두어 본 구현물의 신뢰성과 효율성에 대해 실험한다. 실험의 기본적인 시스템 환경과

조건은 다음과 같다.

- Wireless 환경 : IEEE 802.11b wireless LAN
- 수신자 및 송신자의 시스템 환경 : Pentium 4 1.86Ghz, 1G RAM
- LDPC FEC 코드 스킴 : LDPC Triangle
- 기본 심볼 크기 : 1024(1K) Bytes
- 기본 컨텐츠(파일) 크기 : 3 Mbytes

4.2 패킷 손실과 인코딩 비율에 따른 ALC의 성능평가

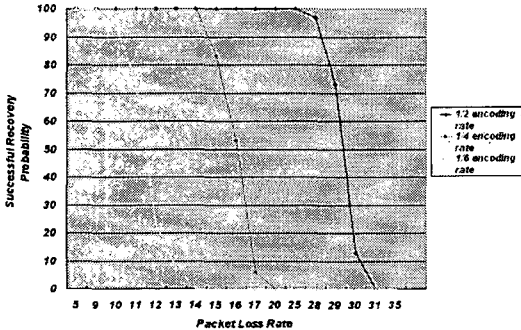


그림 2 인코딩과 패킷 손실 비율에 따른 복구 확률

그림 2의 실험은 4.1에서 언급한 시스템 환경과 최적의 거리에서 인코딩 비율을 1/2, 1/4, 1/6으로 하였을 때, 각각의 상황에서 패킷 손실률을 점차 증가시키면서 복구 확률을 구한 것이다. 실질적으로 무선환경에서의 파일 전송 시, 손실 패턴은 매우 불안정적이다. 본 실험에서는 매 패킷 마다 패킷 손실률을 적용한다. 인코딩 비율이 1/6일 때는 11%이상의 패킷 손실이 일어나면 복구 성공률이 급격히 떨어지고, 1/4일 때는 약 16%의 패킷 손실이 발생하면, 80%이상의 확률로 복구가 가능하다. 부가정보가 가장 많은 1/2일 때는 약 28%정도 손실까지 80%이상의 복구 확률을 나타낸다. 이는 본래의 파일을 80%정도를 복구할 수 있다는 의미가 아닌, 80% 확률로 완벽히 복구할 수 있다는 의미이다. 앞서 언급하였듯이 FEC의 복구능력이 가지는 의미는 복구 정도가 아닌 복구 여부이다. 이 실험은 인코딩 비율에 따라 11%에서 최대 28%정도의 패킷 손실률까지 80%이상의 확률로 복구가 가능하다는 것을 보여주며, 1/2 인코딩 비율은 약 26%의 손실까지 거의 완벽한 성공률을 나타낸다. No-FEC코딩 일 때, 2%의 패킷 손실만 일어나도 대역폭의 효율은 70%이상 떨어지며, 12%의 패킷 손실 시, 최대 대역폭 효율은 거의 0으로 떨어져 최악의 효율을 가진다[7]. 이것은 1/2 인코딩 비율 뿐만 아니라, 1/4의 비율로도 충분히 극복할 수 있는 수치이다.

4.3 ALC와 TCP의 다중연결 성능 비교 평가

ALC의 주목적인 멀티캐스트 전송을 위해 하나 이상의 수신자들을 대상으로, 유니캐스트의 대표적인 예인 TCP와 ALC의 일대다 파일 전송을 실험한다. 앞선 실험과 동일한 시스템 환경에서, 수신자의 수를 늘려가며 TCP와 ALC의 소요시간을 측정한다. 다수의 수신자의 수를 증가시키면서 같은 크기의 파일을 전송하여, 각 프로토콜의 전송 및 복구 메커니즘을 통해 수신자가 파일을 수신하여 복구할 때까지의 최종 시간을 측정하였다. 이때의 손실률은 유선환경 대비 무선 환경의 이론적인 손실률인 0.01~0.12를 적용하였다. 송신자는 각 스스 블록당 해당 심볼들을 랜덤하게 전송하기 때문에, 패킷 손실이 전혀 발생하지 않아도 본래의 파일 크기 이외에 부가정보를 수신하고 디코딩 과정을 거쳐야 한다. LDPC 코드 특징은 복구 심볼들은 반드시 손실이 일어났을 때만, 이용되는 것이 아니라 수신된 패킷과 복구

심볼들을 이용하여, 이후 정상으로 받게 될 패킷에 대해서도 복구 과정이 일어난다. 이는 아무런 패킷 손실이 발생하지 않는다면, 디코딩의 과정에 소요되는 시간은 인코딩 비율이 높아짐에 따라 더욱 효율적이다.

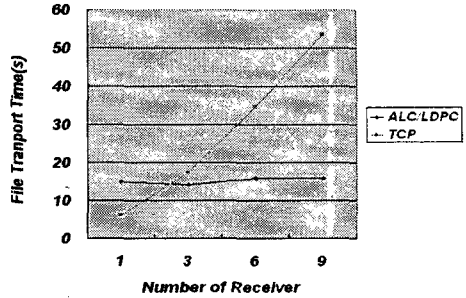


그림 3 일대다 파일 전송시 ALC와 TCP의 비교

그림 3에서와 같이 일대 일의 경우에 ALC는 위와 같은 특징으로 TCP보다 더 많은 시간이 소요된다. 따라서 TCP의 경우 수신자의 수가 적을 때, 모든 패킷을 수신하여 본래의 파일을 복구하는데 소요되는 시간이 ALC보다 짧다. 하지만 수신자의 수가 증가할수록 TCP의 소요시간은 비례하여 증가하는 반면, ALC의 소요시간은 수신자의 수에 상관없이 일정하게 소요된다. 따라서 ALC를 이용한 본 구현들은 많은 수신자에게 전송 할수록 더욱 높은 효율성을 보여준다.

5. 결론 및 향후 과제

본 논문은 신뢰적인 멀티캐스트 전송 프로토콜을 위해 ALC와 LDPC 코드를 이용하여 구현하고, 여러 조건 및 환경에서 평가 및 분석하였다. 첫 번째 실험은 3가지 인코딩 비율에 따라 패킷 손실률을 증가시켜, 80%이상의 복구 성공 확률을 가질 수 있는 최대 패킷 손실률을 측정하여 피드백이 없는 ALC의 신뢰치를 가능하였다. 두 번째 실험으로 TCP와의 비교실험을 통하여 멀티캐스트 환경에서 ALC/LDPC는 수신자의 수가 증가함에 따라 보다 효율적이고 유연한 성능을 보여준다. 하지만 피드백의 부재와 어떠한 FEC 코드로도 완벽한 신뢰성을 제공할 수 없기 때문에, 응용 계층에서는 ALC의 특징에 적합한 복구 메커니즘의 필요성에 대해 다양한 각도에서 연구해야 한다.

6. 참고문헌

- [1] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., and J. Crowcroft, Asynchronous Layered Coding (ALC) Protocol Instantiation, RFC 3450, December 2002.
- [2] T. Paila, M. Luby, R. Lehtonen, V. Roca, R. Walsh, FLUTE - File Delivery over Unidirectional Transport, RFC 3926, October 2004.
- [3] 3GPP TS 26.346 V6.3.0, Multimedia Broadcast/Multicast Service (MBMS): Protocols and codecs, December 2005.
- [4] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M., and J. Crowcroft, Layered Coding Transport (LCT) Building Block, RFC 3451, December 2002.
- [5] V. Roca, C. Neumann, D. Furodet, Low Density Parity Check (LDPC) Forward Error Correction, IETF RMT WG, draft-ietf-rmt-bb-fec-ldpc-00.txt, October 2005.
- [6] MCLv3, <http://www.inrialpes.fr/planete/people/roca>
- [7] Jin-Hwan Chung, Sung-Eun Kim, Copeland, J., "Reliable wireless multicast using fast low-density codes", The 57th IEEE Semiannual Vehicular Technology Conference VTC 2003-Spring Volume 2, pp. 1218-1222, April 2003.