공학석사 학위논문

분산 비디오 스트리밍을 위한 계수 기반의 적응적 FEC 제어 기법

이 논문을 [illegible] 제출함

2005년 2월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

이 병 길

이병길의 공학석사 학위논문을 인준함

2004년 12월 일

주 심 공학박사 조 우 현 

위 원 공학박사 송 하 주 

위 원 공학박사 서 경 룡 

목 차

그 림 목 차

분산 비디오 스트리밍을 위한 계수 기반의 적응적 FEC 제어 기법

이 병 길

부 경 대 학 교 대 학 원 컴 퓨 터 공학과

요약

본 논문에서는 계수 기반의 FEC를 제안하고, 이를 이용하여 분산 비디오 스트리밍 제어 기법을 제안한다. 제안한 계수 기반의 FEC는 수신측에 전송된 패킷의 계수 정보를 담은 제어 패킷을 사용하여 FEC의 오버헤드를 줄인다. 제안한 제어 기법에서는 TCP-Friendly 혼잡 제어 기법을 사용하여 각 서버들의 상태를 계산하고 효율적인 서버 선택 과 전송 제어를 목적으로 한다.

또한, 수신측 기반의 제어 패킷을 이용하여 패킷 분배와 고장 탐지를 위한 서버간의 통신으로 인한 오버헤드를 제거할 수 있다.

Counter-based Adaptive FEC for Distributed Video Streaming

Bung Kil Lee

Dept. of Computer Eng., Graduate School,

Pukyong National University

Abstract

In this thesis, we propose a counter-based FEC and control mechanism that to control distributed video streaming. proposed counter-based FEC reduces overhead of the FEC with control packet that include counter information of transmitted packet to receiver side. proposed control mechanism use the TCP-Friendly rate control algorithm for calculate states of multiple servers and efficient server selection with transmission control in objective.

Also, it uses with receiver side base control packet, it will be able to

remove overhead which caused by with communication of servers for packet distribution and failure detection.

1. 서 론

최근 컴퓨터의 성능향상과 인터넷의 광대역, 고속화는 기존의 인터넷 활용에 많은 변화를 가져왔다. 스트리밍(streaming)기술이 제안되면서 인터넷을 이용한 음악 감상 또는 영화나 각종 미디어 시청, 화상 회의와 같은 일들은 이제 일상적인 모습이 되었다. 하지만 스트리밍 서비스를 이용하는 사용자는 급격히 증가하고 있지만 다른 어플리케이션에 비해 큰 대역폭을 사용하며 지연과 손실에 민감한 멀티미디어 데이터의 특성 때문에 best-effor network인 인터넷을 통한 전송은 많은 어려움을 겪고 있다[1,2].

인터넷 환경에서 스트리밍 서비스의 성능을 향상시키기 위하여 여러 가지 해결 방안들이 연구되고 있다. 미디어 스트림을 분할하거나 스트림에 다양한 scaling 방법을 적용하여 요구 대역폭을 감소시키는 코딩기법들에서 부터 패킷의 손실과 지연 문제를 해결하기 위하여 FEC(Forward Error Correction)과 같은 복구기법들이 제안되었다[1,2,3]. 또 네트워크 레벨에서는 edge 구조를 사용한 CDN(Contents Delivery Network)이 상용화 서비스에 활발히 이용 되고 있다. 이러한 제안들은 보통 하나의 서버와 클라이언트가 고정된 단일 경로를 이용하여 스트리밍을 제공한다는 것을 바탕으로 하고 있다. 만약 사용된 단일 경로에 정체가 오랜 시간 지속된다면 지터(jitter)와 높은 손실률로 인하여 비디오 스트리밍의 제공에 어려움을 겪게 된다. 또, 정체가 없는 경우라도 서버와 클라이언트사이의 왕복 시간이 증가하면 비디오 스트림의 성능은 저하될 수밖에 없다.

이처럼 단일 서버와 클라이언트 형태가 가지는 근본적인 제약사항을 극복하고자 단일 서버와 클라이언트 구조가 아닌 다수의 서버가 하나의 클라이언트로 비디오 스트림을 제공하는 분산 비디오 스트리밍 기법이 제안되었다[1,2,10]. 분산 비디오 스트리밍은 인터넷과 같은 환경에서 다수의 송신측에서 수신측으로 연결된 경로 중 정체가 없는 경로를 통하여 스트림을 제공하거나 다수의 서버가 데이터를 전송에 참여하여 대역폭 제한 문제를 극복함으로써 전체 비디오 스트리밍의 성능을 개선할 수 있다.

본 논문에서는 계수 기반의 FEC와 이를 이용하여 분산 비디오 스트리밍을 제어하는 기법을 제안하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 기존에 연구된 방법들에 대하여 살펴보고 3장에서는 계수 기반의 FEC와 분산 비디오 스트리밍 제어기법을 제안하고 4장에서는 실험과 결과를 분석, 마지막으로 5장에서 결론을 기술한다.

1.1 스트리밍(Streaming)

스트리밍은 1995년 리얼 네트워크사가 개발한 리얼 오디오에서 처음으로 선보였다. 인터넷에서 비디오나 오디오등의 멀티미디어 파일을 하드 디스크 드라이브에 다운로드 받아 재생하던 기존 방식과는 달리 실시간으로 재생하도록 하는 기술이다. 즉, 데이터 전체를 전송한 후 재생하는 것이 아니라 일부 데이터를 수신 후 즉시 재생하는 방식을 사용함으로써 실시간 특성을 가진 멀티미디어 데이터 전송 및 재생에 주로 사용된다[3]. 이때 전송되는 데이터가 마치 물이 흐르는 것처럼 처리 된다고 해서 스트리밍이라는 명칭이 붙여졌다. 아무리 큰 대용량의 멀티미디어 자료라도 이를 개별적으로 실행할 수 있는 1~2초 분량의 작은 조각으로 나눠 연속적으로 전송함으로써 이를 수신하는 사람은 전체자료가 모두 수신 될 때까지 기다릴 필요 없이 즉석해서 각 조각의 파일들을 재생할 수 있도록 하는 것이다. 이때 사용되는 일련의 패킷들을 스트림이라고 한다.

스트리밍은 서비스 방식에 따라 주문형(On-Demand) 스트리밍과 라이브(Live) 스트리밍으로 분류되며 기술적으로는 유니 캐스트(Unicast)와 멀티 캐스트(Multicast)로 구분된다[3,4]. 라이브 스트리밍은 디지털 비디오 캠코더와 같은 장비를 이용하여 영상과 음성을 클라이언트에 실시간으로 제공해 주는 방식을, 주문형 스트리밍은 서버에 미리 저장되어 있는 멀티미디어 데이터를 클라이언트가 원할 때 실시간으로 제공해주는 방식을 말한다.

스트리밍 기술이 사용하는 표준 프로토콜은 RTSP(Real Time Streaming

Protocol)이며, 이 규격은 지난 98년 넷스케이프사와 리얼 네트워크, 컬럼비아대학교가 공동 개발하여 IETF(Internet Engineering)에 표준으로 등록하였다.

2. 관련 연구

2.1 CDN(Content Delivery Network)

Edge 구조란 [그림 2-1]과 같이 서버와 클라이언트 사이의 왕복 시간과 정체 구간으로 인한 손실을 해결하기 위하여 다수의 서버를 인터넷의 가장자리에 전략적으로 배치시켜 클라이언트와 최적의 경로를 가지는 서버를 선택하여 이용하는 방식으로 이를 활용한 기술이 CDN(Content Delivery Network)이다. 일반적으로 CDN은 하나의 CP(Content Provider)와 다수의 캐시 서버(혹은 Edge 서버)로 구성되며 CP를 통하여 다수의 캐시 서버로 데이터 전송과 체계적 관리를 수행하도록 한다.

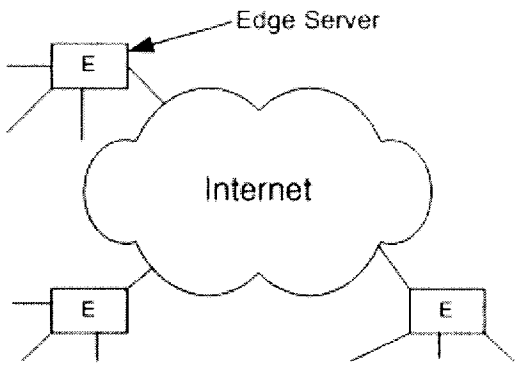


그림 2-1. Edge 구조

CDN은 경로의 다양성을 통해 성능을 향상 시킬 수 있는 효율적인 방법이지만 스트림의 요구 대역폭을 만족하는 캐시 서버가 없을 경우에는 스트리밍 서비스를 제대로 제공할 수 없다는 문제점을 가지고 있다.

2.2 분산 비디오 스트리밍(Distributed Video Streaming)

앞서 살펴본 CDN의 문제를 해결하기 위한 대안으로 제안된 것이 다수의 서버가 하나의 클라이언트로 연속적인 비디오 스트림을 제공하는 분산 비디오 스트리밍 프레임워크이다[1,2,10]. 일반적으로 프레임워크의 구현은 코덱 메커니즘과 스트리밍 프로세스를 제어하는 전송 프로토콜로 이루어지며 분산 비디오 스트리밍 프레임워크는 다음과 같은 장점을 가진다.

① 대용량의 대역폭을 요구하는 고화질 비디오 스트리밍이라도 전송에 참여하는 다수의 전송 서버들이 스트림을 나누어 전송함으로써 대역폭 합을 이용하여 전송할 수 있다. 이는 고화질 비디오 데이터 스트리밍을 보다 용이하게 하여 서비스의 질을 향상 시킬 수 있다.

② 경로의 다양성을 통하여 구간 정체로 인한 패킷 손실과 지연 등의 문제를 보다 쉽게 해결할 수 있도록 하여 높은 안정성과 스트리밍의 성능을 개선할 수 있다.

③ 한 스트리밍 서버의 고장 장애로 인한 서비스가 중지되면 다른 서버를 이용할 수 있도록 하여 장애 허용성을 높일 수 있다.

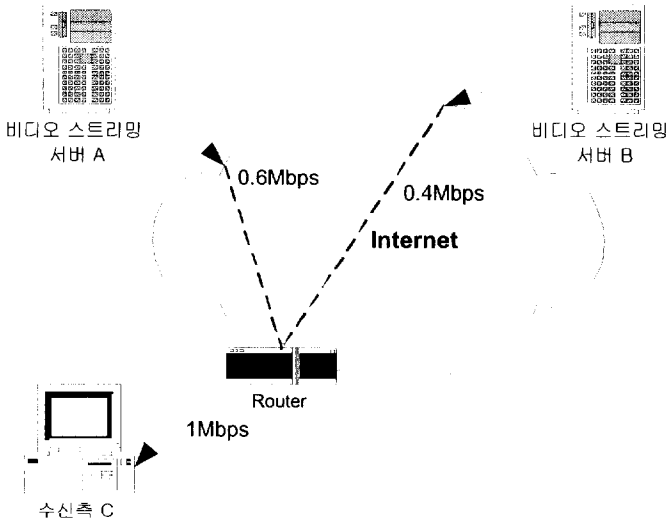


그림 2-2. 분산 비디오 스트리밍의 예

[그림 2-2]는 두개의 비디오 스트리밍 서버를 가지는 분산 비디오 스트리밍의 예를 보여주고 있다. 이 예에서 서버 A와 라우터사이는 0.6Mbps, 서버 B와 라우터사이는 0.4Mbps 그리고 라우터와 수신측 C와의 대역폭은 1Mbps라고 가정한다. 1Mbps의 대역폭을 요구하는 비디오 스트림의 경우라면 라우터와 수신측사이의 대역폭이 1Mbps를 만족할 지라도 서버 A 또는 B와 라우터 사이의 대역폭의 제한으로 전송할 수 없다. 이때 서버 A와 서버 B가 비디오 스트림을 나누어 제공함으로써 대역폭의 제한 문제를 해결할 수 있다. 이때 서버 A와 서버 B사이의 비디오 스트림의 적절한 분배와 제어가 필요하게 되는데 서버들이 이를 직접 협상하게 될 경우 네트워크에 부하를 일으킬 위험이 있다.

분산 비디오 스트리밍 프레임워크를 이용하여 비디오 스트림을 전송할 때

문제를 최대한 간소화하기 위하여 다음과 같은 가정을 전제로 한다.

① 서버와 클라이언트와의 경로 사이에 병목 현상을 일으키는 last hop은 없다고 가정한다. 즉, 클라이언트가 인터넷에 모뎀과 같은 낮은 대역폭의 물리적 구간으로 연결된다면 분산 스트리밍은 제대로 동작하지 못한다. 이러한 경우라면 스트리밍 자체보다도 실제 비디오 데이터를 다운로드 받아서 재생하는 것이 더 좋은 성능을 나타낸다.

② 필요한 video rate, S는 모든 서버의 대역폭의 합으로 제공할 수 있다고 가정한다. 즉, B_i를 서버 i의 대역폭이라고 가정하고 B_r을 수신측의 대역폭이라고 하면 $S \leq \sum B_i$, $i=\{1,...,n\} \leq B_r$ 이어야 한다.

③ 서버들과 클라이언트는 각각 독립적인 경로로 연결되어 있다고 가정 한다.즉, 하나 이상의 정체 구간을 공유하지 않는다고 가정한다. 만약 두 개의 서버 A와B가 하나의 정체 구간을 공유하게 되면 전체 성능을 떨어뜨리게 될 뿐 아니라 하나의 서버의 전송률의 변화는 다른 서버의 전송률에 영향을 미치게 되기 때문이다.

2.3 MPEG

비디오 데이터를 인터넷을 통해 그대로 전송하기에는 용량이 크다. 그래서 비디오 데이터를 효율적으로 전송하기 위한 압축 기술이 필요하게 되었다. MPEG(Motion Picture Expert Group)표준은 1992년에 만들어졌으며 1.5Mbps정도의 낮은 전송속도에서 프로그레시브 비디오의 코딩을 위해 설계되었다.

MPEG는 모든 프레임을 개별 정지화상으로 압축하는 것이 아니라 인접 프레임 사이에 유사점이 많다는 점을 이용하여 동작보상을 하는데 있어서 예측과 보간을 이용한다. 그러나 임의 접근과 같은 VCR식 제어가 가능해야 한다는 등의 여러 이유로 인해 MPEG에서는 자신이 가지고 있는 정보만으로도 복원될 수 있는 프레임이 규칙적으로 삽입되어야 한다. 정지화상으로 압축된 프레임을 I 프레임, 예측만을 한 프레임을 P프레임, 보간을 위한 프레임을 B프레임이라고 한다. MPEG 비디오 데이터는 이들 세 종류의 프레임들이 일정한 패턴으로 섞인 것이다.

I-프레임(Intra-coded frame)은 데이터 스트림의 어느 위치에도 올 수 있으며, 데이터의 임의 접근을 위해 사용된다. 또 다른 이미지들의 참조 없이 부호화 할 수 있다. I-프레임은의 압축은 MPEG에서 가장 낮은 압축률을 보인다. I-프레임은 매크로 블록 내에서 지정된 8*8 블럭으로 나눈 후, DCT기법을 사용한 후에 DC계수는 DPCM방법으로 부호화하는데, 연속한 블록 사이의 차이 값을 계산한 후 가변 길이의 코딩을 사용하여 변환한다.

P-프레임(Predictive-coded frame)은 부호화와 복호화를 행할 때 이전의 I-프레임 정보와 이전의 P-프레임의 정보를 사용한다. P-프레임은 연속되는 이미지들의 전체 이미지가 바뀌는 것이 아니라 이미지의 블럭들이 옆으로 이동한다는 점에 착안한 것이다. 즉, 움직임이 있는 경우 앞 화면에 있는 물체 자체의 모양에는 큰 변화 없이 옆으로 이동하는 경우가 대부분이므로, 이전의 화면과 현재의 화면의 차이가 매우 적은 것을 이용하여 차이 값만을 부호화하는 것이다.

B-프레임(Bidirectional-coded frame)은 부호화와 복호화를 행할 때 이전, 이후의 I-프레임과 P 프레임 모두를 사용한다. B-프레임을 사용하면 높은 압축률을 얻을 수 있다. B 프레임은 이전의 I-프레임 또는 P-프레임과 B-프레임 이후의 I-프레임 또는 P-프레임의 차이 값을 가진다.

I-프레임으로 시작하는 연속적인 화상들의 집합을 GOP (Group Of Picture)라고 한다. B-프레임을 사용하기 때문에 MPEG 코드 데이터 스트림의 순서는 실제 복호화 되는 순서와는 다를 수 있다. 즉, 연관된 B-프레임 후에 출력될 P-프레임은 B-프레임의 복원 시에 필요하므로 P-프레임이 먼저 복원되어야 한다.

[그림 2-3]은 설명한 각 프레임들의 의존관계와 GOP 구성의 한 예를 보여준다.

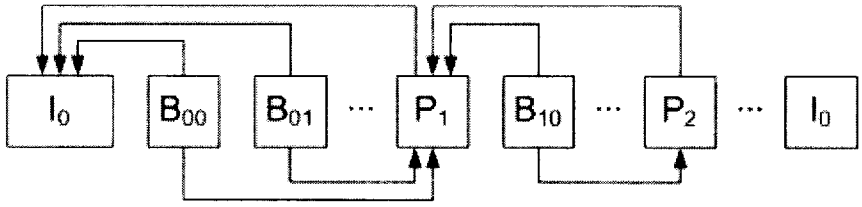


그림 2-3 GOP의 구성과 의존성

2.4 FEC(Forward Error Correction)

인터넷을 통해 멀티미디어 데이터를 전송할 경우 빈번하게 발생하는 패킷 손실은 전송 효율을 떨어뜨리는 요인으로 작용할 수 있다. FEC는 이를 극복하기 위해 제안되었으며 재전송이 필요 없이 원래의 데이터에 추가 정보를 담은 FEC 패킷을 더해 전송하는 방법으로 멀티미디어 데이터 전송에 적합하다고 알려져 있다[5,6].

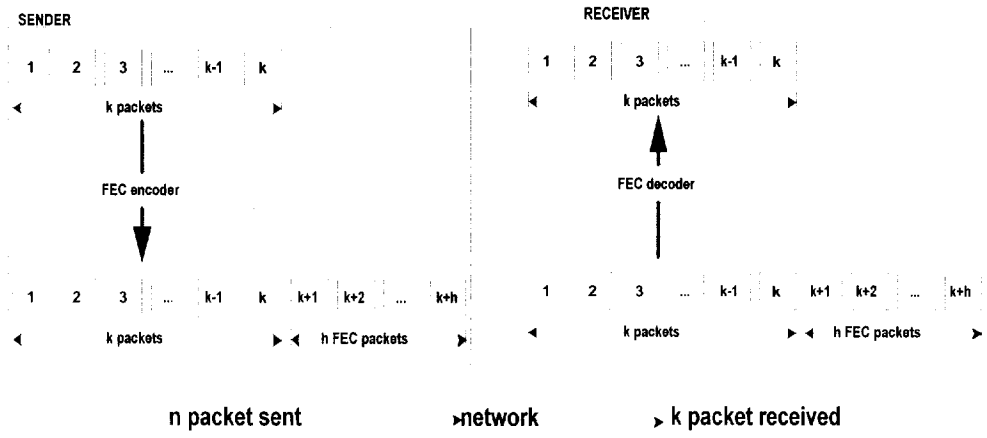


그림 2-4. 일반적인 FEC 스킴

일반적인 FEC 스킴은 [그림 2-4]에서와 같이, 송신측에서 k개의 연속된 미디어 패킷에 h개의 FEC 패킷을 추가로 전송하게 된다. 수신측에서는 이들 패킷들 중 제대로 도착한 패킷들을 이용하여 손실된 패킷을 복구함으로써, 최종적으로 k개의 미디어 패킷을 모두 복구하고자 하는 메커니즘이다[5]. 이러한 FEC 방법은 추가적인 FEC 패킷을 이용하기 때문에 전송 상태가 좋을 경우에는 추가 데이터로 인한 네트워크 대역폭 사용이 증가하는 단점이 있다.

2.5 TCP-Friendly 혼잡 제어 기법

현재의 인터넷 환경은 TCP 프로토콜을 기본적으로 사용하고 있다. TCP 프로토콜은 재전송을 통하여 손실 데이터를 복구하는 방법을 사용하며 혼잡에 대응하도록 설계되어 있다. 재전송 기법은 지연을 가중시킬 수 있기 때문에 실시간 전송을 필요로 하는 비디오 스트리밍에는 적합하지 못하다. 이러한 단점을 보완하기 위해서 기존 비디오 스트림 서비스들은 실시간 데이터라는 전송 미디어의 특성과 적은 프로토콜 오버헤드를 위해 UDP 프로토콜 기반에서 설계 된다.

그러나 네트워크의 상황을 고려할 수 없는 UDP 기반의 스트리밍은 오히려 네트워크의 혼잡 상황 시 경쟁하는 다른 정상적인 TCP 연결들의 전송률을 잠식하는 불공정성 문제를 유발할 수 있다. 이러한 현상의 근본적인 원인은 UDP 프로토콜은 TCP와 같은 혼잡 제어 방법이 없기 때문이다. 전역적인 망에서 스트리밍 서비스를 성공적으로 실현하기 위해서는 현재의 인터넷의 주요한 트래픽인 TCP와 스트리밍 트래픽이 잘 조화 될 수 있어야 한다. 그러므로 같은 네트워크 환경에서 경쟁하는 TCP 기반의 많은 인터넷 어플리케이션들과의 공평한 대역폭 분배는 성공적인 스트림 전송 기반을 평가하는데 매우 중요한 요인으로 작용한다.

TCP-Friendly 혼잡 제어는 TCP가 아닌 프로토콜이 TCP와의 불균형성을 종단 간에서 해결하려는 연구들 중에 가장 대표적인 방법이다. TCP-Friendly란 같은 연결, 상황에서 TCP가 아닌 연결의 평균 전송률이 결국 TCP의

평균 전송률과 유사하게 분포하는 성질을 의미한다.

TCP-Friendly 혼잡 제어 방법은 크게 전송 방법의 지원 유형을 기준으로 단일과 다중 전송 방법으로 구분되며 각 유형들은 다시 전송률과 윈도우 기반의 혼잡 제어 방식으로 나누어진다. 특히 비디오 스트림 전송과 같은 어플리케이션을 위한 연구로는 전송률 기반의 TCP-Friendly 혼잡 제어 방법이 많이 연구되어지고 있다. 그 이유는 이 방법이 다른 방법들에 비해 실시간성과 연속적인 특성을 갖는 비디오 스트리밍의 전송을 제어하는 방법으로 적합하기 때문이다.

3. 분산 비디오 스트리밍 제어 기법

본 장에서는 분산 비디오 스트리밍에서 각 스트리밍 서버의 전송을 효율적으로 제어하기 위하여 다음과 같은 수신측 기반의 제어 기법을 제안한다. 제안한 기법은 전송 손실과 지연으로 인한 스트리밍 전송 성능 저하를 막기 위하여 FEC를 사용할 때 전송된 패킷의 양을 계수로 표현한 정보를 이용하여 부가정보의 양을 동적으로 조절하는 계수기반의 FEC와 이를 이용한 분산 서버의 제어 기법으로 구성된다.

3.1 서버 선택 방법

먼저 제안한 기법에서 사용된 전송 서버 선택 방법은 다음과 같다. 클라이언트는 수신한 각 서버들의 RTT를 이용하여 각 서버의 손실률(loss rate)과 대역폭을 계산한다. 그리고 TCP-Friendly 혼잡 제어 기법을 이용한 각 서버들의 대역폭을 수식 (1)을 이용하여 측정할 수 있다.

$$B=\frac{s}{RTT\sqrt{\frac{2p}{3}}+Trto(3p\sqrt{\frac{3p}{8}})(1+32p^2)} \quad (1)$$

여기서 B는 TCP-Friendly 혼잡 제어를 사용한 서버와 클라이언트 사이의 사용 가능한 대역폭이고 RTT는 서버와 클라이언트 사이의 왕복시간을 의미하며 Trto는 TCP의 재전송 시간을 그리고 s는 세그먼트의 크기, p는 패킷 손실률을 나타낸다. 이 과정을 반복하여 각 서버의 사용 가능 대역폭과 지연 시간 그리고 손실률

을 측정한 후 이 세 개의 데이터를 이용하여 서버 S를 다음과 같은 배열로 표현한다.

$$S_i=\{ B_i , D_i , L_i , S_e \} , i=\{1...n\}$$

- B_i : 서버 i의 가용 대역폭
- D_i : 서버 i의 딜레이
- L_i : 서버 i의 손실률
- $S_e = B_i * (1 - L_i)$: 서버 i의 기댓값

1부터 n까지의 서버들 사이에서 실제 전송 서버 선택을 위하여 서버들을 S_e 를 기준으로 내림차순으로 정렬한 후 B_r 이라고 할 때, 정렬된 n개의 서버 S_i = {1...n}를 이용하여 $\sum_{i=1}^{n} Si*Bi \geq Br$ 를 만족 할 때까지 서버를 선택한다. 이렇게 기댓값을 기준으로 선택한 이유는 지연시간은 버퍼링이나 효율적인 패킷 분배를 통하여 해결할 수 있고 손실률과 대역폭을 최대한 고려하기 위해서이다.

3.2 계수 기반의 적응적 FEC 제어 기법

기댓값을 기준으로 서버들을 차례대로 선택한 후에 패킷 손실과 지연 등의 발생으로 인한 전송 성능 저하 문제를 개선하기 위하여 FEC기법을 사용한다. 하지만 앞서 살펴본 바와 같이 기존의 고정적인 FEC는 전송 상태가 좋을 때에는 추가 FEC 데이터가 오버헤드로 작용, 대역폭 낭비를 초래하여 결과적으로 전체 성능을

떨어뜨리는 단점이 있다.

이러한 단점을 개선하기 위하여 부가 데이터의 양을 채널 상태에 따라 동적으로 조절할 수 있도록 계수 정보를 포함한 제어 패킷을 사용하여 부가 정보의 양을 줄이는 방법을 제안한다.

FEC 코딩에 의해 서버에서 생성하는 총 패킷의 양 n은 n=k+h에 의해 결정되는데, 이때 k는 원래의 데이터 패킷의 양이고 h는 추가 FEC 패킷의 양이다. k와 h중 어느 것이라도 최소한 k개 이상 수신되어야 수신측은 원래의 미디어 패킷을 복구해 낼 수 있다. 여기에서 이 k를 최적 수신량이라고 가정하면 수신측은 k를 이용하여 P_{num}이라는 계수 정보를 다음과 같이 계산한다.

- P_{num}=k - 수신한 패킷의 수

패킷이 도착할 때마다 수신측은 P_{num}을 계산하여 제어 패킷에 포함시켜 송신측에 전달한다. P_{num}=0이 될 때까지 계산되며 0이 된 후에도 패킷을 수신하게 되면 0의 값을 유지한다.

송신측은 P_{num}필드가 0 제어 패킷을 수신하게 되면 전송이 성공적으로 완료되었다고 판단하고 남은 패킷의 송신을 중지함으로써 부가 정보의 양을 줄인다. 설명한 알고리즘을 도식화 하면 [그림 3-1]과 같은 형태가 된다.

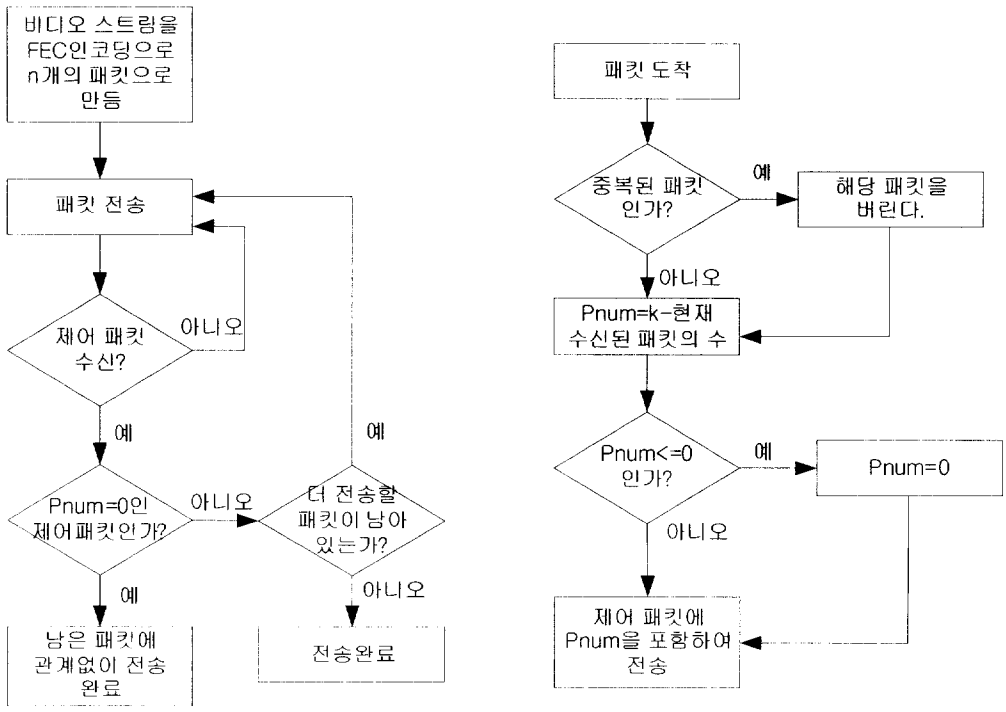
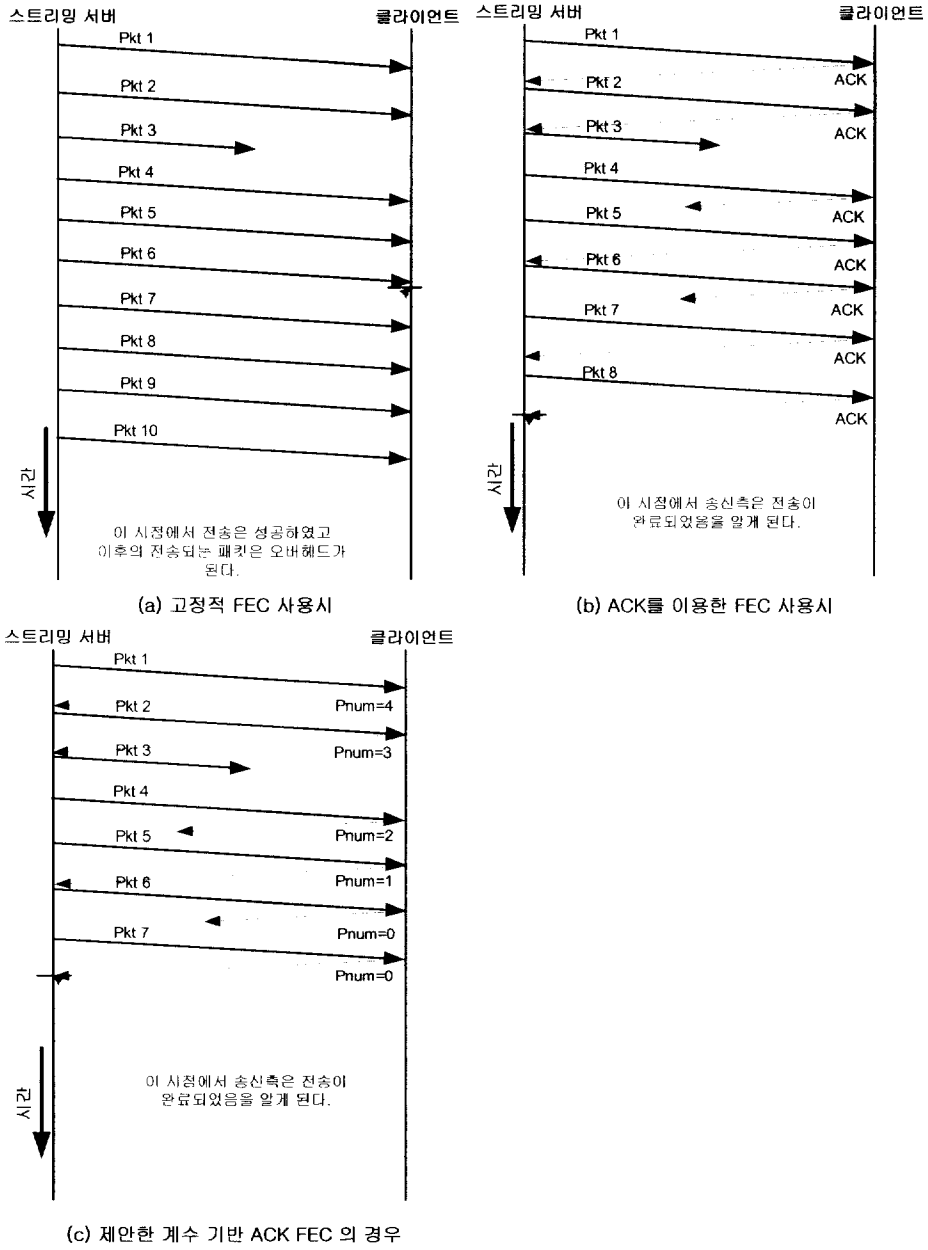


그림3-1. 송/수신측 동작 알고리즘

설명한 알고리즘을 적용하면 채널의 전송 상태가 극히 좋은 경우, 송신측은 $P_{num}=0$ 값을 가진 제어 패킷을 수신으로 남은 FEC 중복 패킷의 전송을 최소한으로 줄일 수 있다. 즉, 이러한 상황이라면 기존의 고정적 FEC가 가지는 오버헤드를 줄일 수 있다.

반면에 채널의 상태가 나쁜 경우를 고려하면 제어 패킷의 전달도 어려워지게 되는데, 이때는 송신측은 최적 수신량 k보다 많은 패킷을 전송하게 된다. 하지만 이렇게 높은 오류율로 인하여 빈번한 패킷 손실이 일어나는 경우라면 비디오 스트림의 성공적인 전달이 중요하게 되므로 이때에 전송된 오버헤드의 양은 무시할 수 있다. 즉, 최악의 경우에는 기존의 FEC가 가지는 장

점을 그대로 가지는 것이다.



(a) 고정적 FEC 사용시

(b) ACK를 이용한 FEC 사용시

(c) 제안한 계수 기반 ACK FEC 의 경우

그림3-2. 각 기법의 전송 시나리오 비교

[그림 3-2]는 RS(10,5) 코드를 사용했을 때, 고정적 FEC와 ACK를 이용한 FEC 그리고 제안한 계수기반 FEC의 전송 시나리오를 비교한 그림이다. 이 때 전송되는 3번째 패킷이 손실되고, 수신측에서 피드백을 제공한다면 동일한 양의 피드백 패킷이 손실된다고 가정하였다.

[그림 3-3]에서는 다수의 피드백 패킷이 손실될 경우를 나타낸다.

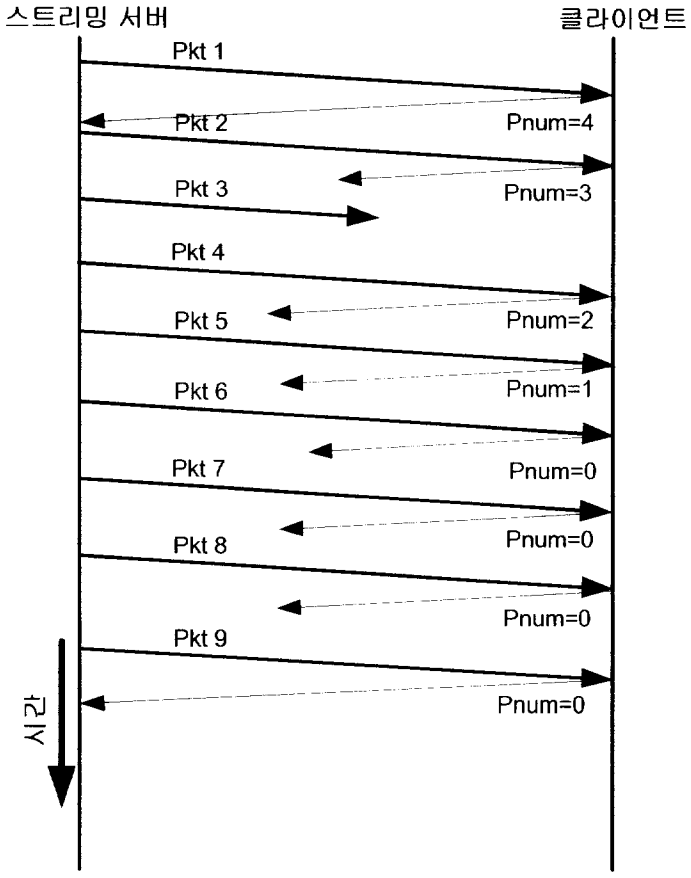


그림3-3. 제어 패킷이 계속해서 손실되는 경우의 예

[그림 3-2]의 (b)처럼 채널 상태를 피드백 해주는 FEC의 형태는 피드백

정보가 다수 손실될 경우 제대로 된 상태 정보를 피드백 해주기가 어렵게 된다. 하지만 제안한 기법(c)에서는 P_{num} 계수가 0인 패킷이 손실되더라도 다음번 계수정보 P_{num}이 0인 제어패킷이 도착하게 되면 전송이 완료됨을 볼 수 있다.

만약 계속적인 제어 패킷의 손실이 발생한다 하더라도 송신측이 n개의 패킷을 전부 전송하기 전에 제어 패킷이 도착하게 되면 기존의 FEC보다 적은 양의 패킷을 전송 할 수 있다.

3.3 분산 비디오 스트리밍 제어 기법

선택된 각 서버들의 대역폭을 고려하여 각 서버들이 전송해야할 패킷을 분배하고 각 서버의 패킷 전송을 제어하기 위하여 3.2절에서 제안한 계수 기반 FEC를 확장하여 다음과 같은 분산 비디오 스트리밍 제어 기법을 제안한다.

비디오 스트림을 전송할 서버들은 정해진 FEC 코딩을 사용하여 각각 전체 패킷을 모두 생성한다. 이때 각 서버가 전송할 패킷을 분배하기 위하여 제어 패킷에 다음과 같은 추가 정보를 포함하여 [그림]과 같은 제어 패킷 형식을 정의 한다.

c	OF 1	OF 2	OF 3	OF 4	OF 5	Pnum	Sync

그림3-4. 확장된 제어 패킷 형식

- c : 체크비트. Offset 필드가 수정되거나 Sync 필드가 수정되었을 때 전송서버에 수정되었음을 알리기 위하여 사용한다.

- OF1~5 : Offset 필드. 해당 서버가 전송해야할 패킷의 Offset을 표현하는 필드로써 전송서버는 Sync 필드를 기준으로 자신의 Offset필드에 기록된 값을 확인하여 패킷을 전송한다.

- P_{num} : k-현재 수신측에 수신된 패킷의 수.

- Sync : Offset의 기준이 되는 패킷의 Sequence Number를 표시한다.

OF필드를 통하여 수신측이 대역폭을 고려하여 각 서버가 전송해야할 패킷을 분배함으로써 각 송신 서버들이 데이터 분배를 위한 정보를 교환할 필요가 없어지게 된다. 이는 정보 교환으로 이한 부하를 줄이고 패킷 분배를 수신측이 담당함으로써 절차를 간소화 할 수 있는 장점을 가진다.

송신측은 제어 패킷을 수신하여 자신의 OF 필드와 Sync 필드를 확인함으로써 자신이 전송해야 할 패킷들을 확인하고, P_{num}필드를 체크하여 0인 제어 패킷을 수신할 때 까지 송신을 계속하게 된다.

[그림 3-5]은 두 개의 전송서버를 이용한 분산 비디오 스트리밍일 경우의 전송 시나리오이다. 스트리밍 서버 A와 서버 B에서 클라이언트로 스트림을 전송할 때 대역폭은 동일하다고 가정하였다. 또 분배 계산을 간소화하기 위하여 패킷의 분배는 A는 홀수 번째 패킷을 B는 짝수 번째 패킷을 전송하는 RS(10,5)를 사용한다고 가정하였다.

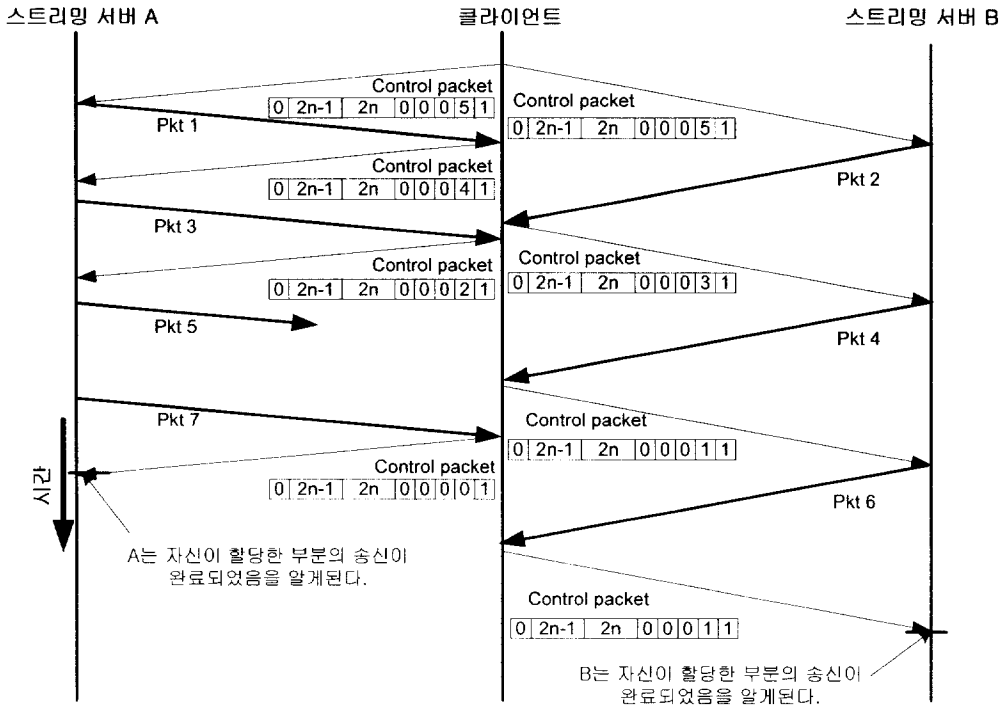


그림3-5. 2개의 서버를 가지는 분산 비디오 스트리밍 전송 시나리오

[그림 3-5]에서 클라이언트는 Sync 필드에 1을 표시하여 첫 번째 패킷을 기준으로 전송하라는 것과 각 서버의 OF필드에 2n-1과 2n을 표시하여 전송함으로써 수신한 서버는 이것을 체크하여 해당 패킷을 전송하기 시작한다.
이과정은 P_{num}이 0인 제어패킷이 도착할 때 까지 반복된다.

[그림 3-6]은 동일한 조건 하에서 스트리밍 서버B에 장애가 발생하여 한 개의 패킷을 전송 후 장애가 발생할 때의 전송 시나리오를 보여준다.
이때 수신측은 A의 OF 필드와 Sync 필드를 수정하여 전송함으로써 스트리밍 서버 A가 나머지 부분을 보내도록 요청한다. 이것은 전송 가능한 스트리밍 서버가 2대뿐 일 경우를 가정하였다.

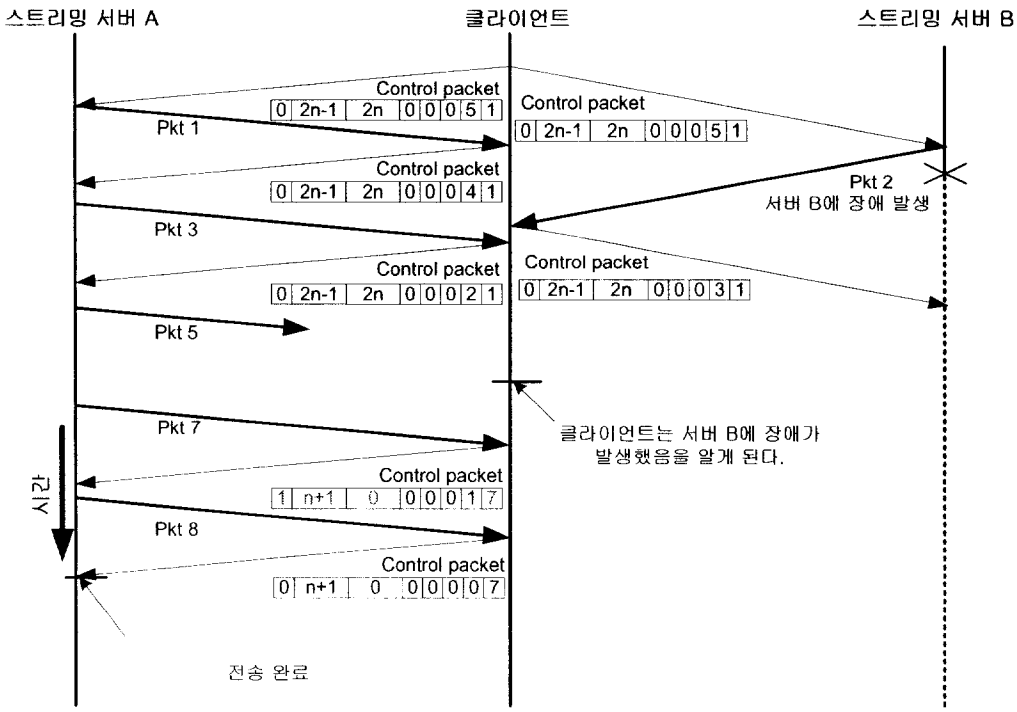


그림3-6. 하나의 서버에 장애 발생 시 시나리오.

4. 시뮬레이션

본 장에서는 시뮬레이션에 사용된 시스템과 파라미터 및 실험 모델에 대하여 설명한다. 시뮬레이션 시스템은 셀러론 733 Mhz의 CPU와 512 M 메모리로 구성된 리눅스(커널 2.4.x)의 시스템을 사용하였으며, 시뮬레이션에 사용한 툴은 버클리 대학에서 개발한 분산 객체 네트워크 시뮬레이터인 NS-2(Network Simulator)를 사용하였다[11].

4.1 시뮬레이션 모델

제안한 계수 기반의 적응적 FEC 기법의 부가 패킷 조절 성능을 평가하기 위하여 FEC를 사용하지 않았을 때와 RS(179,124)코드를 사용하는 고정적 FEC 방식, ACK를 피드백 정보로 활용하는 RS(179,124)코드의 FEC 그리고 동일한 RS코드를 적용한 제안한 계수 기반 FEC의 성능을 비교 측정하였다.

망 모델은 두 개의 전송 서버와 한 개의 수신측을 가지며 전송 서버에서는 1024Kbyte 크기의 고정된 패킷 크기를 512Kbps 속도로 전송하는 모델을 사용하였다.

5. 시뮬레이션 결과 및 분석

5.1 비트에러율의 증가에 따른 성능 변화

[그림 5-1]은 비트 에러율의 증가에 따른 고정적 FEC와 ACK기반 FEC 그리고 제안한 계수 기반 FEC의 성능 변화를 실험한 결과이다.

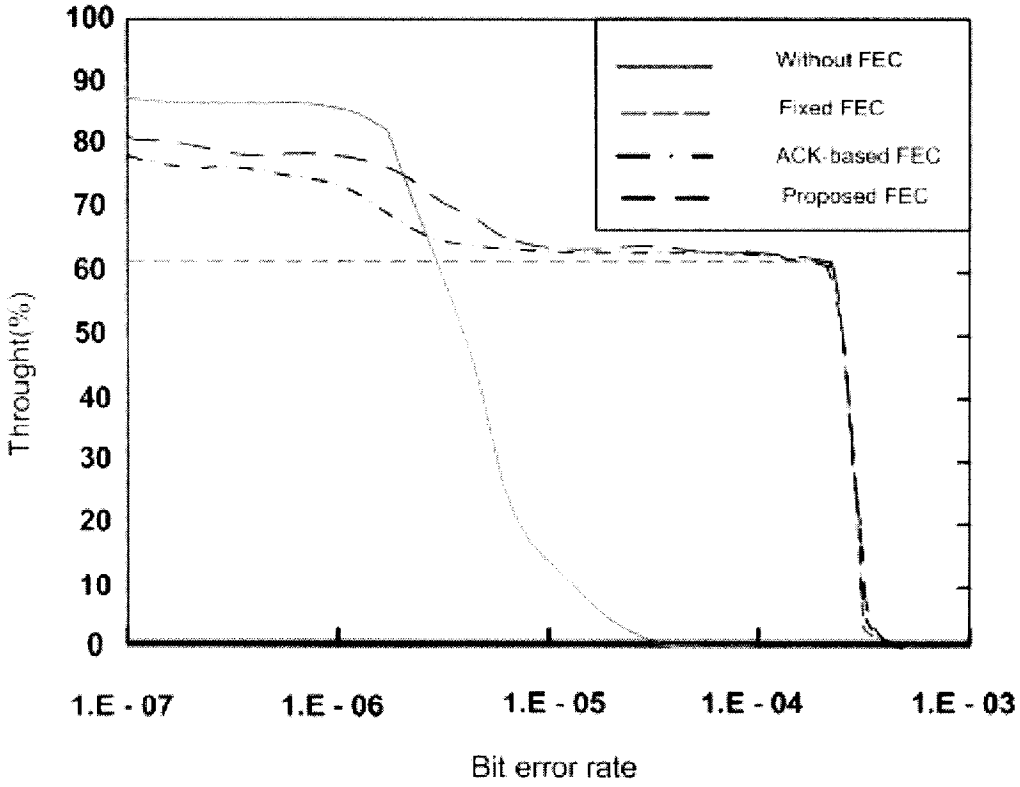


그림5-1. 비트 에러율의 변화에 따른 성능 비교

비트 에러율이 낮을 때에는 FEC를 적용하지 않았을 때 가장 높은 성능이 나타내지만, 10-6 이후로 에러율이 증가에 따라 성능이 급격하게 나빠지는

것을 볼 수 있다.

반면에 고정적 FEC는 에러율이 낮을때에는 오버헤드로 인하여 FEC를 적용하지 않았을 때에 비해 낮은 성능을 보이지만, 에러율의 증가에도 성능의 변화가 거의 없이 일정한 성능을 보여주고 있다. 이것은 솔실률이 높은 환경에서는 FEC를 적용함으로써 성능이 향상됨을 확인 할 수 있다.

피드백 정보를 이용하여 에러율이 낮은 구간에서 오버헤드를 개선하려고 하는 접근방법은 고정적 FEC에 비하여 에러율이 낮은 환경에서 좋은 성능을 나타내는 것을 볼 수 있다. 실험을 통하여 채널 상태가 좋은 상황일 경우에는 피드백 정보를 주고받는 오버헤드보다 고정적 FEC가 가지는 오버헤드가 크므로 이러한 방법이 채널상태가 좋을 경우 FEC의 오버헤드를 줄일 수 있다는 것을 확인하였다.

특히, 제안한 계수 기반의 FEC는 다른 2개의 FEC에 비해 에러율이 낮은 구간에서 조금 더 좋은 성능을 나타내는 것을 알 수 있다. 하지만 에러율이 계속해서 증가하게 되면 고정적 FEC와 유사한 성능을 보이게 되는데. 이것은 제어 패킷의 수신이 어렵기 때문이며 최악의 경우 고정적 FEC의 성능은 유지하는 것으로 나타났다.

5.2 패킷 손실률의 증가에 따른 부가 데이터 전송 비율

[그림 5-2]는 경로의 패킷 손실률의 증가할 때 각 방법의 부가 데이터 전송 비율을 측정한 결과이다. 전체 전송에서 부가 데이터의 전송 비율의 최대 값은 고정적 FEC이고 최저 값은 FEC를 사용하지 않았을 때를 기준으로 봤을 때, ACK 기반의 FEC보다 부가데이터의 양이 줄어든다는 것을 확인할 수 있다. 이것은 피드백 정보가 손실될 때에는 부가 데이터 전송을 효율적으로 줄일 수 없었지만 동일하게 피드백 정보가 손실될 때에는 기존의 방법에 비해 성능이 향상됨을 알 수 있다.

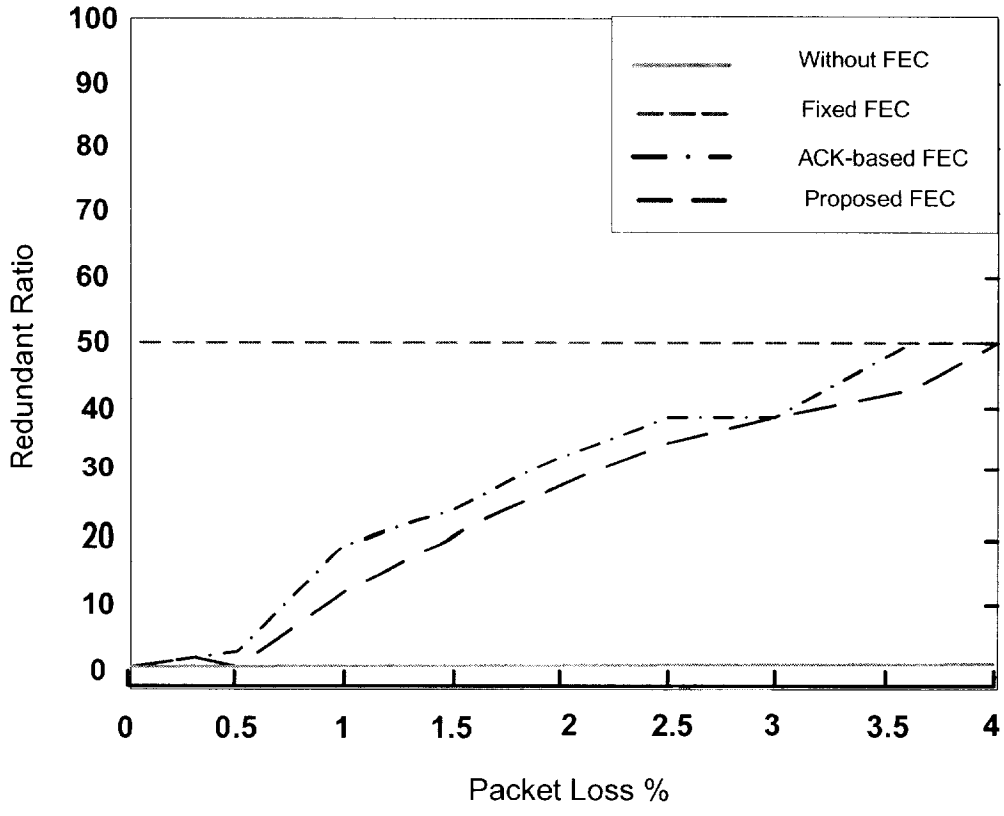


그림5-2. 패킷 손실률의 변화에 따른 부가 데이터 전송 비율 비교

6. 결론

인터넷의 대중화 추세로 이제 일상생활에서도 비디오 스트리밍 서비스의 사용이 급격히 증가하고 있지만 대부분의 스트리밍 서비스는 인터넷의 구간 정체 또는 패킷 손실로 인하여 만족할 만한 서비스를 제공하지 못하고 있다.

본 논문에서는 인터넷 환경에서 비디오 스트리밍의 성능을 개선하기 위하여 제안된 분산 비디오 스트리밍을 효율적으로 제어하기 위하여 계수기반의 FEC를 제안하고 이를 이용한 제어 기법을 제안하였다.

수신측은 Sflag와 Pnum을 포함한 제어 패킷을 사용하여 수신측으로 전달된 데이터의 수신 상태와 해당 송신측이 송신해야 할 패킷을 알림으로써 추가 FEC패킷의 전송을 줄이고 송신서버들 사이의 정보 교환 없이 패킷 분배가 가능하도록 하였다. 또한 수신측이 패킷 분배를 담당하게 함으로써 분산 스트리밍에서 한 개의 송신서버가 장애를 일으키더라도 적응적으로 대응할 수 있음을 보였다.

참고문헌

[1] Thinh Nguyen and Avideh Zakhor, "Distributed Video Streaming Over Internet", in Proceedings of SPIE Conference on Multimedia Computing and Networking, San Jose, California, January 2002.

[2] Thinh Nguyen and Avideh Zakhor, "Distributed Video Streaming with Forward Error Correction", Packet Video Workshop 2002, Pittsburgh PA, USA.

[3] John G. Apostolopoulos, "Reliable Video Communication over Lossy Packet Networks using Multiple State Encoding and Path Diversity", VCIP, Jan. 2001.

[4] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani and R. D. Gitlin, "AIRMAIL: A link-layer protocol for wireless networks", wireless networks, vol.1, pp.47-60, Feb. 1995.

[5] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin and D. Ganesan, "Building Efficient Wireless Sensor Network with Low-Level Naming", SOSP01, Oct. 2001.

[6] M. Zorzi, "Some Results on Error Control for Burst-Error Channels Under Delay Constraints", IEEE Trans. VT. vol.50, no. 1, Jan 2001.

[7] Kadur S., Golshani F. and Millard G., "Delay-jitter control in multimedia application", Multimedia System, No. 4, pp.30-39, 1996.

[8] T. Nguyen, P. Mehra, A. Zakhor, "Path Diversity Media Streaming over Best Effort Packet Switched Networks", Dissertation, U.C. Berkeley,

2003.

[9] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast application", in Architectures and Protocols for Computer Communication, Octorber 2000, pp. 43-56.

[10] T. Nguyen and A. Zakhor, "Matching Pursuits Based Multiple Description Video Coding for Lossy Environment", International Conferen ce on Image Processing 2003.

[11] NS-2 Simulator

http://www.isi.edu/nsnam/ns

[12] 김형준, 안종석, "IEEE 802.1 MAC 프로토콜을 이용하는 무선 랜의 전송 성능향상을 위한 적응적 FEC 기법", 정보과학회 추계학술대회, vol 29, no.02, October 2002.

감사의 글

오늘이 있기까지 학부 생활과 대학원 생활 6년 동안 부족한 제자를 지속적인 보살핌과 배려 그리고 성심을 다해 지도 편달해 주신 서경룡 교수님께 진심으로 감사의 글을 올립니다. 그리고 부족한 저의 논문을 심사해 주시고 세심한 지도와 많은 관심을 보여주신 조우현 교수님, 송하주 교수님께 감사를 드립니다. 또한 학위 과정 동안 부족한 저에게 열정적으로 많은 가르침을 주셨던 학과의 여러 교수님들께도 감사드립니다.

또한 논문을 준비하면서 어려운 시기임에도 불구하고 음으로 양으로 아낌없는 지원과 조언을 해준 석주와 연구실 식구들 정식이형과 귀여운 동생들 영록, 태일 그리고 많은 조언을 아끼지 않았던 데이터베이스 연구실에 희숙이 누나를 비롯한 동기, 선배, 후배들 모두에게도 감사의 마음을 전합니다.

멀리 떨어져 있지만 항상 격려와 지원을 해준 친구이상의 친구 경민이와 든든한 환조형, 지치고 힘들 때 마다 일어설 수 있도록 도와준 재희, 늘 부족하고 덜렁거리는 저를 도와주신 모든 분들에게 감사의 뜻을 전합니다.

끝으로 항상 곁에서 보살펴주시고 사랑해주시는 우리 부모님, 철없는 동생을 멀리서도 걱정하는 누나에게 사랑과 감사의 마음을 담아 이 논문을 바칩니다.

2005년 2월 이 병 길 올림