

무선 네트워크 환경에서 효과적인 영상 스트리밍을 위한 And-Or 트리 기반의 LT 부호화 심볼 패킷화 알고리즘

준회원 이 동 주*, 정회원 김 완*, 양 연 모**, 송 황 준*

A Robust And-Or Tree-based LT Encoded Symbol Packetization Algorithm for Video Streaming over Wireless Network

Dongju Lee* *Associate Member,*

Wan Kim*, YeonMo Yang**, Hwangjun Song* *Regular Members*

요 약

본 논문에서는 무선 네트워크 환경에서의 패킷 손실에 따른 영상 스트리밍 서비스의 영상 열화를 최소화 시키는 And-Or 트리 기반의 LT 패킷화 알고리즘을 제안한다. 심볼들의 효율적인 패킷화를 위해 And-Or 트리를 통한 심볼들 간의 관계를 분석하고 이 관계를 바탕으로 LT 부호화 심볼들이 포함된 패킷들 간의 의존성을 줄여 패킷 손실에 대한 영향을 최소화 시키는 패킷화 알고리즘을 설계한다. 마지막으로 무선 네트워크 환경에서 영상 스트리밍 시스템을 통해 제안한 패킷화 알고리즘이 기존의 패킷화 알고리즘보다 더 향상된 복호화 성능과 영상 화질을 제공할 수 있음을 보인다.

Key Words : Luby Transform codes, Packetization algorithm, And-Or tree, Video streaming, Wireless network

ABSTRACT

In this paper, we propose a robust And-Or tree-based Luby transform encoded symbol packetization algorithm to minimize the quality degradation of video streaming service caused by packet losses over wireless network. To achieve this goal, the relationship among Luby transform encoded symbols are analyzed by using And-Or tree, and the proposed packetization algorithm is designed to minimize packet loss effects by reducing the dependency among packets conveying Luby transform encoded symbols. Finally, experimental results are provided to show the performance of the proposed algorithm.

I. 서 론

최근 네트워크와 영상 전송 기술의 발전으로 무선 네트워크 환경에서의 영상 스트리밍 서비스에 대한 수요가 높아지고 있다. 그러나 무선 네트워크 환경에서의 영상 스트리밍 서비스는 많은 문제점을 지니고

있다. 무선 네트워크는 시변적인 불안정한 특성을 가지며 공기라는 매체를 공유하여 사용하므로 유선 네트워크에 비해 상대적으로 높은 패킷 손실율을 보인다. 특히, 원본 영상 데이터는 그 크기가 매우 크기 때문에 효과적인 영상 스트리밍 서비스를 위해 H.264^[1]와 같은 영상 부호화기를 통해 높은 비율로 데이터를

※ 본 연구는 방송통신위원회의 차세대통신네트워크 원천기술개발사업의 연구결과로 수행되었음 (KCA-2011-09913-05006)

* 포항공과대학교 컴퓨터공학과 멀티미디어 통신/네트워킹 연구실 (ldjblaze, xviii, hwangjun)@postech.ac.kr

** 금오공과대학교 전자공학부 (yangym@kumoh.ac.kr)

논문번호 : KICS2011-01-058, 접수일자 : 2011년 1월 26일, 최종논문접수일자 : 2011년 9월 6일

압축하여 전송한다. 이로 인해 압축된 영상 데이터는 패킷 손실에 매우 취약하며, 무선 네트워크에서의 영상 데이터 전송은 영상 화질을 크게 저하시킬 수 있다. 이러한 무선 네트워크 환경에서 패킷 손실을 복원하기 위해 ARQ (AutoRepeat reQuest) 와 FEC (Forward Error Correction) 기법들을 사용한다. ARQ 기법은 손실된 데이터의 복원을 위해 재전송을 수행한다. 이로 인해 종단간의 지연이 크게 증가하여 지연에 민감한 영상 스트리밍 서비스에 적합하지 않다. 반면, FEC 기법은 기존 데이터에 추가적인 복원 데이터를 함께 전송하여 재전송 과정 없이 자체적으로 수신단에서 손실된 데이터를 복원한다. 그러므로 지연에 민감한 영상 스트리밍 서비스에 매우 적합하다.

최근 다양한 FEC 기법들 중에서 패킷 손실에 강인하고 안정적인 영상 스트리밍 서비스를 위해 파운틴 코드 [2, 3, 4]가 많이 사용되고 있다. 파운틴 코드는 응용 계층에서 동작하는 블록 기반의 FEC 기법으로, 간단한 구조를 가져 부호화 심볼의 양을 유연하게 조절할 수 있으며 낮은 시간 복잡도와 높은 복호화 성능을 제공한다. 파운틴 코드를 활용한 연구도 현재 매우 활발하게 진행되고 있다. [5]에서는 파운틴 코드의 복호화 성공 확률을 분석하여 중요도가 높은 데이터의 복원 성능을 높일 수 있도록 심볼들 간의 링크 수를 조절하는 UEP (Unequal Error Protection) 기법을 제안하였으며 [6]에서는 스케일러블 영상 데이터의 특성을 고려하여 파운틴 코드를 통한 UEP 기법을 제안하였다. 이 기법을 바탕으로 스케일러블 영상 데이터의 중요도에 따라 복원 성공 확률을 조절하여 효과적인 스케일러블 영상 멀티캐스트를 제공하고자 하였다. 또한 [7]에서는 수신단의 피드백 정보를 바탕으로 파운틴 코드의 부호화 심볼 오버헤드 비율을 적절하게 조절할 수 있는 시스템을 제안하여 유연하고 안정적인 멀티미디어 스트리밍 서비스를 제공하고자 하였다. 기존에 제안된 알고리즘은 패킷 네트워크에 따른 패킷화를 고려하지 않고, 파운틴 코드 자체에 집중되어 연구가 진행되어 왔다. 실제로 전송이 이뤄지는 패킷 네트워크의 특성이 반영된 연구는 미미한 실정이다.

본 논문에서는 파운틴 코드 중 하나인 LT (Luby-Transform) 코드 [3]을 이용하여 심볼들 간의 관계를 분석하고, 이 관계를 바탕으로 파운틴 복호화 성능을 향상시킬 수 있는 패킷화 알고리즘을 제안한다. II에서 파운틴 코드의 기본적인 동작에 대해 설명하고, III에서는 제안한 패킷화 알고리즘을 기술한다. 그리고 IV에서는 패킷화에 따른 LT 복호화 성능 및 영상 화질 비교를 통해 제안한 패킷화 기법의 성능을 검증한

다. 마지막으로 V에서는 결론에 대하여 기술 한다.

II. 파운틴 코드

FEC 기법 중의 하나인 파운틴 코드는 높은 복호화 성공률과 낮은 부호화 및 복호화 처리 시간을 제공하며 응용 계층에서의 구현으로 부호화 데이터의 양을 자유롭게 조절할 수 있는 특징을 가져 최근 여러 분야에서 활발하게 사용되고 있다. 그림 1은 파운틴 코드는 블록 기반의 구조로 크게 소스 블록과 부호화 블록으로 나뉘며 각 블록은 일정한 크기를 가지는 심볼들로 구성된다. 부호화 심볼들은 소스 심볼들과 연결을 이루고 있으며, 하나의 부호화 심볼은 직접 연결된 소스 심볼들의 XOR 연산 값을 가진다. 각 부호화 심볼들은 파운틴 코드의 차수 확률 분포 (Degree Probability Distribution)를 기반으로 차수가 결정되고, 이 결정된 차수만큼 소스 심볼들을 임의로 선택한다^[3]. 이러한 동작을 모든 부호화 심볼들에 대해 수행한 후, 송신단에서는 생성된 부호화 심볼들을 수신단에게 전송한다. 수신단에서는 일반적으로 소스 심볼의 수보다 약간 더 많은 수의 부호화 심볼들을 수신할 수 있다면 모든 소스 심볼들을 성공적으로 복호화 할 수 있다.

대표적인 파운틴 코드에는 LT 코드와 RT (Raptor-Transform) 코드가 있다. LT 코드는 최초의 파운틴 코드로 크기가 큰 소스 심볼을 전송 할 때 확률적으로 오버헤드가 거의 발생하지 않으며, 부/복호화에 필요한 계산량을 최소화 한다. 반면, RT 코드는 LT 코드의 개선된 형태로 LT 코드의 선형적이지 못한 복호화 연산량을 선형화시켰다.

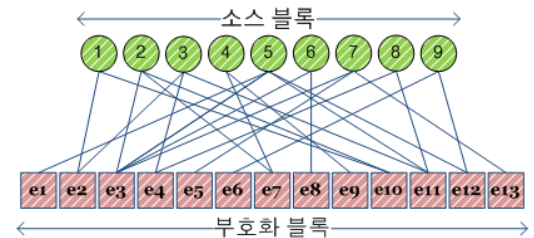


그림 1. 파운틴 코드의 기본 구조

III. 제안하는 LT 부호화 패턴 기반의 심볼 패킷화 알고리즘

그림 2는 제안하는 패킷화 기반의 영상 스트리밍 시스템이다. 제안하는 시스템은 영상 데이터를 영상

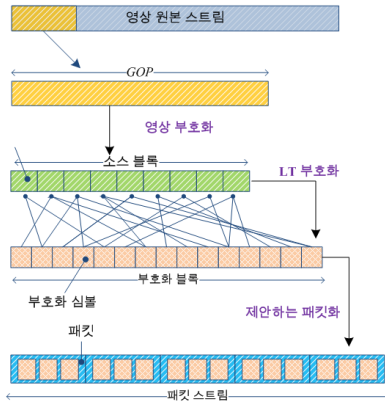


그림 2. 제한하는 패킷화 알고리즘 기반의 영상 스트리밍 서비스 시스템

코덱을 통해 영상 부호화 한 후, 파운틴 코드를 통해 LT 부호화 블록을 생성한다. 그리고 LT 심볼들 간의 관계 분석을 기반으로 하는 패킷화 알고리즘을 적용하여 수신단에 전송한다. 기존의 패킷화 알고리즘은 생성된 LT 부호화 심볼들을 순차적으로 패킷화하여 수신단에 전송한다. 그러나 그림 1에서 보듯이 파운틴 코드의 심볼들은 서로 긴밀한 연결 관계를 맺고 있으며, 이런 연결로 인해 복호화 과정에서 이미 복호화된 소스 심볼들은 다른 소스 심볼들의 복호화에 영향을 준다³⁾. 그러므로 이러한 관계를 고려하지 않고 심볼들을 전송하면 패킷 손실에 의해 파운틴 복호화 성능을 크게 저하시킬 수 있다. 이를 보완하기 위해 제한하는 패킷화 알고리즘은 소스 심볼과 부호화 심볼들간의 관계 분석을 바탕으로 패킷화를 수행하여 패킷 손실에 따른 영향을 최소화 하고, 기존의 패킷화 알고리즘과 비교해 같은 수의 LT 부호화 심볼들을 수신하였을 때, 더 높은 LT 복호화 성능을 제공하고자 한다.

3.1 And-Or 트리 기반의 LT 부호화 심볼 관계 분석

반복적인 부호화 과정을 통해 LT 부호화 심볼들간의 관계는 트리의 형태로 표현할 수 있다. 본 논문에서는 확률적 분석을 위한 유용한 도구인 And-Or 트리 [8]를 이용하여 심볼들간의 관계를 유도하였으며, 이를 위해 사용되는 심볼들의 설명은 표 1과 같다. 일반적으로 $2l$ 의 깊이를 가지는 And-Or 트리 T_l 이 존재할 때, 루트 노드의 깊이는 0을 가지며 직접 연결된 자식 노드들의 깊이는 1을 가진다. 또한 트리의 아래로 내려 갈수록 노드들의 깊이는 1씩 증가하게 되며, 모든 노드는 0 또는 1의 값을 가진다. 그리고 깊이

표 1. 심볼 설명

Symbol	Description
$root$	목표 소스 심볼
TES_i	$root$ 의 i 번째 목표 부호화 심볼
N_T	$root$ 의 자식 노드 수
d_{TES_i}	TES_i 의 차수 (degree)
SS_{TES_i}	TES_i 와 연관 있는 소스 심볼들의 집합 (TES_i 의 자식 노드들)
$\lambda_j^{TES_i}$	SS_{TES_i} 에 속하는 j 번째 소스 심볼의 링크 수, $1 \leq j \leq (d_{TES_i} - 1)$
RES_{TES_i}	TES_i 와 연관 있는 부호화 심볼들의 집합
s_{packet}	패킷 당 삽입가능한 부호화 심볼의 수

0, 2, 4, ..., $2l - 2$ 에 위치하는 노드들은 Or-노드라고 칭하며 자식 노드들의 Or-연산을 통해 값이 결정된다. 반면, 1, 3, 5, ..., $2l - 1$ 에 위치하는 노드들을 And-노드라고 칭하며 자식 노드들의 And-연산을 통해 값이 결정된다 [8]. 이 트리에 LT 코드의 심볼들을 맵핑시키면 소스 심볼은 Or-노드로, 부호화 심볼은 And-노드로 간주되며 각각은 서로 부모와 자식 노드들이 된다. 심볼들 간의 관계를 유도하기 위한 간단한 And-Or 트리 구조는 그림 3과 같다. $root$ 는 Or-노드, TES_i 는 And-노드이기에 다른 목표 부호화 심볼들과 상관없이 TES_i 에 의해서 $root$ 가 복호화 되기 위해서는 SS_{TES_i} 에 속한 소스 심볼들이 선행적으로 모두 복호화 되어야 한다. 그러므로 TES_i 를 이용한 $root$ 의 성공적인 복호화는 깊이 3에 위치하는 부호화 심볼들의 영향을 받게 된다. 즉, TES_i 가 $root$ 의 성공적인 복호화에 기여하기 위해서는 SS_{TES_i} 에 속한 소스 심볼들이 선행적으로 복호화가 이루어져야 하며, 그림 3에서 보듯이 수신단에서 TES_i 와 더 많은 RES_{TES_i} 에 속하는 부호화 심볼들을 동시에 수신할 경우, SS_{TES_i} 에 속하는 소스 심볼들을 더 많이 복호화 될 수 있으며, 이를 통해 $root$ 의 복호화 성공률을 높일 수 있다. 이것은 TES_i 와 RES_{TES_i} 에 속한 부호화 심볼들이 서로 강한 의존성을 가지고 있다는 것을 의미한다. 만약 TES_i 가 전송 과정에서 손실되면 트리 상에서 RES_{TES_i} 에 속하는 부호화 심볼들과 $root$ 간의 연결이 끊어지게 되며, 결국 RES_{TES_i} 에 속하는 부호화 심볼들이 $root$ 의 복호화에 아무런 기여를 할 수 없다는 것

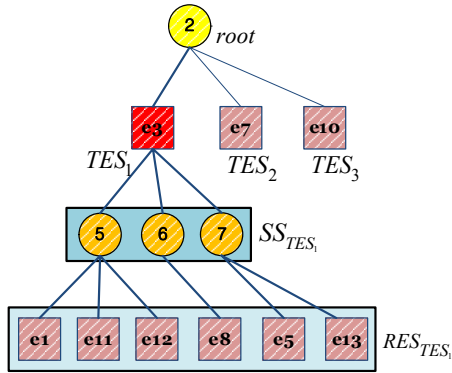


그림 3. 소스 심볼 2를 root로 선택하였을 때, LT 심볼들 간의 관계 유도를 위한 And-Or 트리의 구조 예

을 의미한다. LT 복호화 성공률을 높이기 위해 심볼들 간의 연관 관계가 고려되어야 하며 이를 바탕으로 다음과 같은 2가지 유용한 속성을 얻을 수 있다.

- 속성 1:** RES_{TES_i} 에 속하는 부호화 심볼들을 더 많이 이용 가능하면 SS_{TES_i} 에 속한 소스 심볼들의 복호화 성공률을 향상 시킬 수 있다 [5].
- 속성 2:** root의 복호화 성공률은 TES_i 와 RES_{TES_i} 에 속하는 부호화 심볼들을 같은 패킷에 삽입함으로써 향상 시킬 수 있다.

단, $RES_{TES_i}(root_p)$ 와 $RES_{TES_j}(root_q)$ 를 각각 소스 심볼 p 와 q 가 root인 경우의 TES_i 와 TES_j 의 연관 있는 부호화 심볼들의 집합이라고 할 때 $RES_{TES_i}(root_p) \cap RES_{TES_j}(root_q)$ 는 공집합이 아닐 수도 있다. 즉, 일부 LT 부호화 심볼들은 서로 다른 집합에 동시에 포함될 수도 있다는 것을 의미한다. 본 논문에서는 TES_i 가 속한 패킷에 같이 삽입되는 RES_{TES_i} 에 속한 부호화 심볼의 수를 조절함으로써 이러한 And-Or 트리들 간의 의존성을 간접적으로 고려한다. 그리고 패킷화 과정에서 트리들 간의 의존도를 완전히 제어하기 위해서는 높은 수준의 계산 복잡도를 요구하는 분석이 필요하다. 본 논문에서는 계산 복잡도를 줄이기 위해 트리에 대한 분석은 깊이 4로 제한한다.

3.2 제안하는 LT 부호화 심볼 패킷화 알고리즘

TES_{sel} 을 $\{TES_i, 1 \leq i \leq N_T\}$ 중에서 root의 복호화에 기여하기 위해 선택된 목표 부호화 심볼일 때, 3.1절에서 언급한 속성들을 바탕으로 제안하는 패킷

화 알고리즘은 $RES_{TES_{sel}}$ 에 속하는 많은 부호화 심볼들을 TES_{sel} 와 같은 패킷에 삽입 되도록 설계해야 한다. $P_{RES_{TES_{sel}}}$ 를 TES_{sel} 와 같은 패킷에 삽입되는 $RES_{TES_{sel}}$ 에 속하는 부호화 심볼들이라고 정의하고, 이 패킷을 목표 패킷이라 할 때, $RES_{TES_{sel}}$ 에 속하는 부호화 심볼들 중의 몇 개의 부호화 심볼들이 목표 패킷에 TES_{sel} 과 같이 삽입될 것인지는 제안하는 패킷화 알고리즘의 제어 변수 R_{thr} 에 의해 결정된다. 만약 R_{thr} 이 0일 경우, 제안하는 패킷화 알고리즘은 기존의 패킷화 알고리즘과 동일한 동작을 수행한다. 본 논문에서는 앞의 속성들을 기반으로 제안하는 패킷화는 $s_{packet} \geq 2$ 을 가정한다. 그 이유는 $s_{packet} = 1$ 인 경우 즉, 패킷당 부호화 심볼이 하나씩만 삽입되면 TES_{sel} 와 $RES_{TES_{sel}}$ 를 같은 패킷에 삽입할 수 없으며, 이것은 3.1절 속성 2을 만족시키지 못해 패킷화에 따른 성능을 기대할 수 없다. 3GPP MBMS [9]에서 파운틴 코드의 심볼 크기에 따른 패킷 당 심볼 수가 2개 이상인 경우를 권고하고 있으며, [10]에서는 파운틴 코드의 전체 소스 심볼의 수를 고려하면서 심볼 크기가 작은 경우 좋은 성능을 얻을 수 있다고 언급하고 있다. 그러므로 $s_{packet} \geq 2$ 인 경우에서의 패킷화 고려는 충분히 의미를 가진다. 제안하는 패킷화 알고리즘의 동작은 다음과 같다.

- 단계 0:** 첫 번째 소스 심볼을 root 노드로 결정한다.
- 단계 1:** root 노드의 $\{TES_i, 1 \leq i \leq N_T\}$ 에서 $d_{TES} \geq 2$ 이며 패킷화 되지 않은 TES_{sel} 를 선택한다. (만약 모든 $TES_i, 1 \leq i \leq N_T$ 가 이미 패킷화 되었다면 단계 4를 수행한다.)
- 단계 2:** TES_{sel} 를 다음과 같은 규칙에 의해 선택된 목표 패킷에 삽입한다. (규칙 i는 규칙 (i+1)보다 우선순위가 높다) 만약 규칙들에 의해 선택할 수 있는 목표 패킷이 더 이상 존재하지 않을 경우 단계 4를 수행한다.
 - 규칙 1:** 만약 빈 패킷이 존재하면 그 패킷을 목표 패킷으로 결정한다.
 - 규칙 2:** 만약 패킷의 사용 가능한 공간이 $R_{thr} + 1$ 보다 크면 그 패킷을 목표 패킷으로 결정한다.
- 단계 3:** $RES_{TES_{sel}}$ 에 속한 부호화 심볼들 중에서 R_{thr} 개를 임의로 선택하여 $P_{RES_{TES_{sel}}}$ 를 결정한다. (단 $|A|$ 를 집합 A의 크기 (Card-

inality) 라고 할 때, $\min\{(s_{packet} - 1), |RES_{TES_{sel}}|\}$
 $\geq R_{thr}$ 를 만족한다.) 그리고 $P_{RES_{TES_{sel}}}$
 를 그림 4와 같이 목표 패킷에 삽입한다.

단계 4 : 만약 더 이상 root를 선택할 수 없는 경우 나머지 패킷화 되지 않은 부호화 심볼들을 패킷들의 남은 공간에 삽입한다. 그렇지 않을 경우 다음 소스 심볼을 root로 선택하고 단계 1~3을 반복한다.

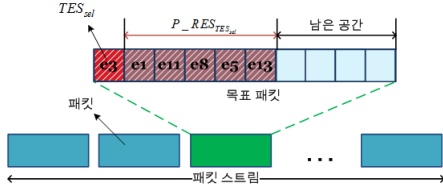


그림 4. $s_{packet} = 10$, $R_{thr} = 5$ 일 때 제안한 패킷화 알고리즘의 수행 예

3.3 제안하는 LT 부호화 심볼 패킷화 알고리즘의 수식적 분석

제안한 패킷화 알고리즘의 성능 척도는 성공적으로 수신한 $RES_{TES_{sel}}$ 에 속하는 LT 부호화 심볼들의 수이다. $R_{RES_{TES_{sel}}^{prop}}$ 와 $R_{RES_{TES_{sel}}^{conv}}$ 를 각각 제안한 패킷화 알고리즘과 기존의 패킷화 알고리즘을 적용하였을 때 손실 없이 성공적으로 수신단에서 수신한 $RES_{TES_{sel}}$ 에 속하는 부호화 심볼들이라고 정의할 때, $|R_{RES_{TES_{sel}}^{prop}}| > |R_{RES_{TES_{sel}}^{conv}}|$ 를 만족할 경우 제안한 패킷화 알고리즘이 기존의 패킷화 알고리즘보다 더 좋은 성능을 가진다고 할 수 있다. 패킷화 알고리즘 적용에 따른 성능 분석을 위해 TES_{sel} 가 전송과정에서 손실된 경우와 정상적으로 수신단에서 이용 가능한 경우의 두 가지 상황을 고려한다. 먼저 TES_{sel} 가 전송과정에서 손실된 경우는 앞서 언급했듯 $RES_{TES_{sel}}$ 가 root의 복호화에 아무런 영향을 주지 못하기 때문에 무시할 수 있다. 두 번째는 TES_{sel} 가 정상적으로 수신단에서 수신한 경우이다. 이 경우 패킷 손실이 랜덤하게 발생한다고 가정하면 다음과 같이 $|R_{RES_{TES_{sel}}^{prop}}|$ 와 $|R_{RES_{TES_{sel}}^{conv}}|$ 를 다음과 같이 계산할 수 있다.

$$|R_{RES_{TES_{sel}}^{conv}}| = \frac{|RES_{TES_{sel}}|}{n_{pkt}} + \left\{ \frac{(n_{pkt} + 1) \cdot |RES_{TES_{sel}}|}{n_{pkt}} \right\} \cdot (1 - PLR) \quad (1)$$

$$|R_{RES_{TES_{sel}}^{prop}}| = \{R_{thr} + \alpha\} + \{|RES_{TES_{sel}}| - (R_{thr} + \alpha)\} \cdot (1 - PLR) \quad (2)$$

$$\alpha = \frac{|RES_{TES_{sel}}| - R_{thr}}{n_{sym} - (R_{thr} + 1)} \cdot \{s_{packet} - (R_{thr} + 1)\} \quad (3)$$

여기서 n_{sym} 은 생성된 LT 부호화 심볼의 수, n_{pkt} 는 전송되는 패킷수 (즉, n_{sym}/s_{packet} 보다 작지 않은 최소의 정수), α 는 패킷화 되지 않고 남은 부호화 심볼들을 패킷들의 남은 빈 공간에 삽입하는 과정에서 목표 패킷에 포함된 $RES_{TES_{sel}}$ 에 속하는 평균 부호화 심볼의 수를 의미한다. 그리고 PLR 은 패킷 손실률을 의미한다. 이를 바탕으로 $|R_{RES_{TES_{sel}}^{prop}}| > |R_{RES_{TES_{sel}}^{conv}}|$ 을 항상 보장하기 위해 R_{thr} 은 다음 식을 만족해야 한다.

$$R_{thr} > \frac{|RES_{TES_{sel}}| \cdot \{n_{sym} - 1 - n_{pkt} \cdot (s_{packet} - 1)\}}{n_{pkt} \cdot \{n_{sym} - |RES_{TES_{sel}}| - s_{packet}\} + |RES_{TES_{sel}}|} \quad (4)$$

일반적으로 LT 부호화 심볼의 차수 확률 분포는 $\{\Omega_1 \dots \Omega_k\}$ [3]로 나타내며, k 는 소스 심볼의 수, Ω_i 는 부호화 심볼이 차수 i 를 가질 확률을 의미한다. LT 코드의 차수 확률 분포를 생성 다항식 $\Omega(x) = \sum_{i=1}^k \Omega_i x^i$ 로 표현하면 부호화 심볼의 평균 차수를 $\Omega'(1)$ 로 나타낼 수 있다. 그러므로 $\lambda_j^{TES_{sel}}$ 와 $|RES_{TES_{sel}}|$ 를 다음과 같이 표현할 수 있다.

$$\lambda_j^{TES_{sel}} = \frac{\Omega'(1) \cdot n_{sym}}{k} = \Omega'(1) \cdot \gamma \quad (5)$$

$$|RES_{TES_{sel}}| = \sum_{j=1}^{d_{TES_{sel}}-1} (\lambda_j^{TES_{sel}} - 1) = (\Omega'(1) - 1) \cdot (\Omega'(1) \cdot \gamma - 1) \quad (6)$$

여기서 γ 은 LT 부호화 오버헤드 비율 (즉, $\gamma = n_{sym}/k$)을 의미한다. 최종적으로 우리는 식 (7)과

$$R_{thr} > \beta$$

$$\beta = \frac{(\Omega'(1) - 1) \cdot (\Omega'(1) \cdot \gamma - 1) \cdot \{n_{sym} - 1 - n_{pkt} \cdot (s_{packet} - 1)\}}{n_{pkt} \cdot \{n_{sym} - (\Omega'(1) - 1) \cdot (\Omega'(1) \cdot \gamma - 1) - s_{packet}\} + (\Omega'(1) - 1) \cdot (\Omega'(1) \cdot \gamma - 1)} \quad (7)$$

같이 유도된 R_{thr} 의 조건을 만족시킬 때, 제안한 패킷화 알고리즘이 기존의 패킷화 알고리즘보다 더 좋은 성능을 가지는 것을 증명 할 수 있다.

IV. 실험 결과

4.1 실험 환경

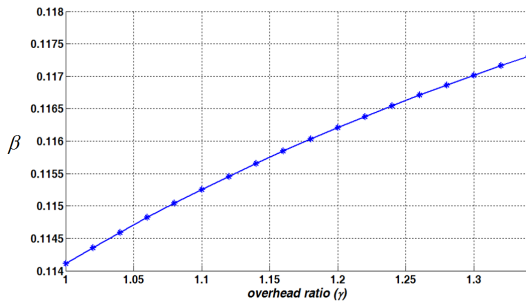
실험을 위해서 제안한 시스템은 Java 와 C/C++을 사용해 구현하였으며, 실험 환경은 다음과 같다. 영상 코덱은 H.264/AVC JM 소프트웨어를 사용하였으며 테스트 영상 시퀀스는 15fps 및 CIF 크기의 Foreman, Harbour를 사용하였다. GOP (Group of Picture)는 하나의 I 프레임 과 29 개의 P 프레임으로 구성하였다. 그리고 영상 복호화 후 프레임의 손상된 부분을 보완 (Concealment) 하기 위해 프레임 복사 기법을 사용하였으며 LT 코드의 차수 확률 분포 $\Omega(x)$ 는 Robust Soliton 분포를 적용하였다³⁾. 시뮬레이션은 크게 두 가지로 나뉜다. 먼저 제안한 패킷화 알고리즘이 기존의 패킷화 알고리즘보다 더 좋은 LT 복호화 성능을

보이는지 검증하고, 이를 바탕으로 실제 무선 네트워크 환경을 고려하여 제안한 패킷화 알고리즘을 적용한 시스템이 더 좋은 화질의 영상 스트리밍 서비스를 제공할함을 보인다. 마지막으로 각 실험은 1000번 반복 수행하였다.

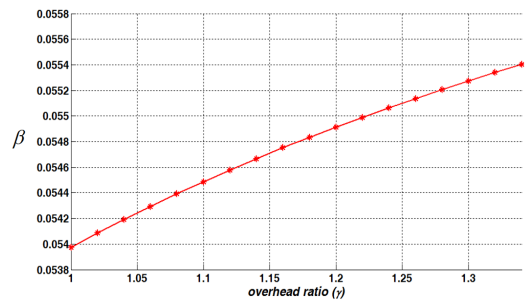
4.2 제안하는 패킷화 알고리즘의 성능 검증

제안하는 패킷화 알고리즘이 기존의 패킷화 알고리즘보다 더 좋은 LT 복호화 성능을 제공하기 위해서 3.3 절에서 유도한 R_{thr} 의 조건을 만족시켜야 한다. 그림 5는 k 와 s_{packet} 가 주어졌을 때 γ 의 증가에 따른 β 값이다. 결과에서 보는 것과 같이 요구되는 R_{thr} 의 최소 치수 (Lower Limit) 는 1보다 작은 것을 알 수 있다. 즉, 제안한 R_{thr} 이 1보다 크거나 같으면 식 (7)를 만족시키기 때문에 제안한 패킷화 알고리즘이 기존의 패킷화 알고리즘보다 더 좋은 LT 복호화 성능을 제공해 줄 수 있음을 알 수 있다.

그림 6은 제안한 패킷화 알고리즘의 변화에 따른 평균 LT 복호화 실패율을 기존의 패킷화 알고리즘을

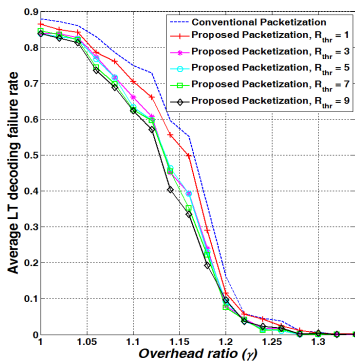


(a) $k = 1000, s_{packet} = 10$

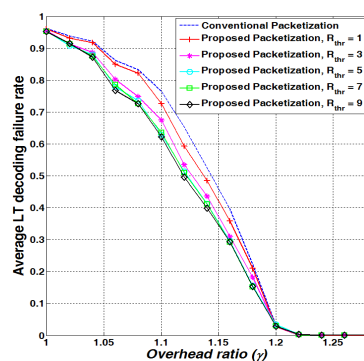


(b) $k = 2000, s_{packet} = 10$

그림 5. γ 에 따른 식 (1)을 만족시키는 R_{thr} 의 최소 치수



(a) $k = 1000, PLR = 0.1$



(b) $k = 2000, PLR = 0.1$

그림 6. R_{thr} 변화에 따른 제안한 패킷화 알고리즘과 기존 패킷화 알고리즘의 평균 LT 복호화 실패율 비교 ($s_{packet} = 10$)

적용한 경우와 비교한 결과이다. 복호화 실패율은 전체 소스 심볼 중에서 복호화가 제대로 이뤄지지 못한 심볼의 비율을 나타낸다. 결과에서 보이는 것과 같이 제안한 패킷화 알고리즘은 다양한 소스 심볼의 수와 γ 에 대해, 그림 5의 결과와 같이 R_{thr} 이 1보다 크면 더 낮은 LT 복호화 실패율을 가진다. 즉, 더 좋은 LT 복호화 성능을 제공해 주는 것을 확인 할 수 있다.

표 2는 제안한 패킷화 알고리즘과 기존의 패킷화 알고리즘의 패킷화 시간을 비교한 것이다. 제안한 패킷화 알고리즘으로 인한 평균 추가 지연 시간은 k 가 1000 및 2000일 때, 각각 28.88 ms 및 157.46 ms 정도로 작으며, 또한 패킷화 알고리즘은 초기에 한 번만 수행한 뒤, 그 이후로는 앞서 계산된 패킷화 패턴 값을 계속 사용하기 때문에 전체 영상 스트리밍 서비스 시간에는 거의 영향을 주지 않는다.

표 2. 제안한 패킷화 알고리즘과 기존 패킷화 알고리즘의 패킷화 시간 비교

k	γ	패킷화 시간(ms)	
		기존 패킷화	제안한 패킷화
1000	1.0	11.81	37.36
	1.1	12.96	38.21
	1.2	13.79	42.85
	1.3	15.77	51.44
2000	1.0	55.11	179.46
	1.1	63.25	216.44
	1.2	71.14	238.65
	1.3	78.69	263.49

4.3 영상 시퀀스를 이용한 패킷 손실 패턴에 따른 패킷화 알고리즘의 성능 비교

본 절에서는 영상 시퀀스를 이용하여 실제 무선망에서의 영상 스트리밍 서비스 환경에서 제안한 패킷화 알고리즘의 성능을 LT 복호화 실패율과 PSNR (Peak Signal-to Noise Ratio)를 측정하여 비교, 분석하였다. 그림 7은 R_{thr} 가 5인 경우, 패킷 손실이 랜덤하게 발생할 때 패킷화 알고리즘에 따른 LT 복호화 실패율과 각 영상 시퀀스에 대한 PSNR을 비교한 결과이다. R_{thr} 값은 실험적으로 얻은 최적의 값이다. 모든 영상 시퀀스에 대해 제안한 패킷화 알고리즘이 더 좋은 LT 복호화 성능과 PSNR을 제공하는 것을 알 수 있다. 그리고 실제 사용자가 직관적으로 화질을 느낄 수 있도록 직접 영상 시퀀스의 화질을 비교하였다. 그림 8과 같이 기존의 패킷화 알고리즘의 경우 상대적으로 손상된 부분이 많이 나타나 화질 저하 현상이 크게 발생하는 것을 알 수 있다.

V. 결 론

본 논문은 패킷 기반의 무선 네트워크 환경에서 패킷 손실에 강인한 영상 스트리밍 서비스를 위한 LT 코드 기반의 패킷화 알고리즘을 제안하였다. 제안한 패킷화 알고리즘은 LT 심볼들간의 관계를 And-Or 트리를 이용하여 분석하고 이 관계를 바탕으로 LT 복호화를 위해 서로 영향을 미치는 LT 부호화 심볼들을 같은 패킷에 삽입하도록 설계하였다. 실험 결과에서

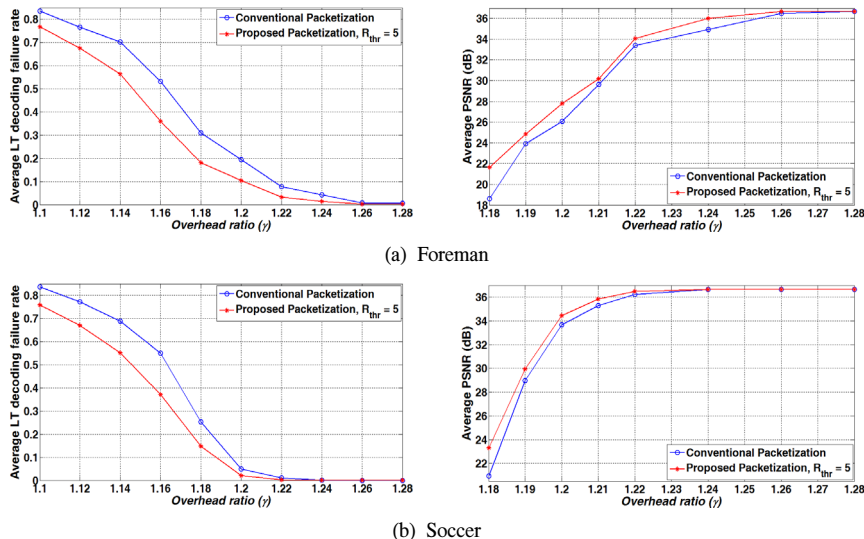


그림 7. 랜덤 패킷 손실, $s_{packet} = 10$, $R_{thr} = 5$ 일 때, 패킷화 알고리즘에 따른 평균 LT 복호화 실패율 및 PSNR 비교



(a) Foreman ($\gamma=1.24$)



(b) Soccer ($\gamma=1.22$)

그림 8. 제안한 패킷화 알고리즘과 기존 패킷화 알고리즘의 영상 화질 비교

제안한 패킷화 알고리즘은 기존 패킷화 알고리즘보다 더 좋은 LT 복호화 성능을 제공하였으며 랜덤 패킷 손실이 발생하는 네트워크 환경에서 고화질의 영상 스트리밍 서비스를 사용자에게 제공할 수 있음을 확인 하였다.

참 고 문 헌

[1] ITU-T Recommendation H.264 version 3, "Advanced video coding for generic audiovisual services," Nov. 2007.

[2] J. C. MacKay, "Fountain codes," IEE Proceedings-Communications, Vol.152, No.6, pp.1062-1068, 2005.

[3] M. Luby, "LT codes," Proceedings of Annual Symposium on Foundations of Computer Science, pp.271-280, 2002.

[4] A. Shokrollahi, "Raptor codes," IEEE Transactions on Information Theory, Vol.52, No.6, pp.2551-2567, Jun. 2006.

[5] N. Rahnavard, B. N. Vellambi and F. Fekri, "Rateless codes with unequal error protection property," IEEE Transaction on Information Theory, Vol.53, No.4. pp.1521~532, Apr. 2007

[6] D. Vukobratovic, V. Stankovic, D. Sejdinovic, L.Stankovic and Z. Xiong, "Scalable video multicast using expanding window fountain codes," IEEE Transactions on Multimedia, Special Issue on Quality-driven Cross-layer Design for Multimedia Communications,

Vol.11, issue 6, pp.1094-1104, Aug. 2009.

[7] S. Ahmad, R. Hamzaoui, and M.Al-Akaidi, "Adaptive unicast video streaming with rateless codes and feedback," IEEE Transactions on Circuits and Systems for Video Technology, Vol.20, No.2. pp.275-285, Feb. 2010

[8] M. Luby, M. Mitzenmacher and A. Shokrollahi, "Analysis of random processes via And-Or tree evaluation," Proc. of the 9th Annual ACM-SLAM Symposium on Discrete Algorithms, pp. 364-373, Jan. 1998.

[9] 3GPP, "3GPP TS 26.346 V8.1.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs," Dec. 2008.

[10] J. Afzal, T. Stockhammer, T. Gasiba and W. Xu, "Video streaming over MBMS: a system design approach," Journal of Multimedia, Vol.1, No.5, Aug 2006.

이 동 주 (DongJu Lee)

준회원



2009년 2월 부산대학교 정보컴퓨터공학과(학사)
2011년 2월 포항공과대학교 컴퓨터공학과(석사)
<관심분야> 채널 코딩, 영상 스트리밍, 무선 네트워크

김 완 (Wan Kim)

정회원



2007년 8월 홍익대학교 컴퓨터공학과(학사)
2008년 3월~현재 포항공과대학교 컴퓨터공학과(통합)
<관심분야> WiMAX, 미래 인터넷, 네트워크 시뮬레이션, 채널 코딩, 영상 스트리밍

양 연 모 (Yeon Mo Yang)

정회원



1990년 2월 KAIST 전기전자
공학파(학사)

1999년 2월 GIST 메카트로닉
스(석사)

2006년 2월 GIST 메카트로닉
스(박사)

2006년~2008년 2월 대구경북
과학기술원 선임연구원

2008년 3월~현재 금오공과대학교 전자공학부(교수)

<관심분야> EPONs, 센서 네트워크, 메카트로닉스,
융합기술, 네트워크 시뮬레이션

송 황 준 (HwangJun Song)

정회원



1990년 2월 서울대학교 제어계
측공학과(학사)

1992년 2월 서울대학교 제어계
측공학과(석사)

1999년 5월 Univ. of Southern
California, EE-Systems(박사)

2000년~2005년 2월 홍익대학
교 전자전기공학부(조교수)

2005년 2월~현재 포항공과대학교 컴퓨터공학과

(부교수)

<관심분야> 멀티미디어 네트워킹, 영상압축, 통방용
합기술