

# 콘텐츠 중심 네트워크에서 정규표현식을 활용한 캐시친화적인 적응형 스트리밍 프레임워크

손동현\*, 최대진\*, 최낙중\*\*, 송정환\*, 권태경\*

## Cache-Friendly Adaptive Video Streaming Framework Exploiting Regular Expression in Content Centric Networks

Donghyun Son\*, Daejin Choi\*, Nakjung Choi\*\*, Junghwan Song\*, Ted "Taekyoung" Kwon\*

### 요약

인터넷을 이용하는 사용자의 관점이 호스트 중심에서 콘텐츠 중심으로 변화하면서 콘텐츠 중심 네트워크(Content Centric Network, 이하 CCN)라는 새로운 패러다임이 소개되었다. 한편, 최근 비디오 스트리밍에 대한 수요가 급증하고 있으며 더 높은 사용자의 만족도를 위한 적응형 스트리밍이 소개되면서 많은 연구가 진행 중에 있다. 따라서 CCN에서도 사용자의 수요에 따라 적응형 스트리밍을 고려할 필요성이 있다. 하지만 CCN에서 기존의 네트워크 구조에서와 동일한 방식으로 적응형 비디오 스트리밍 서비스를 할 경우 CCN 라우터 내 캐시(CS) 충분히 활용하지 못하는 한계점이 있으며 또한 단말의 달라지는 요구 사항을 캐시 활용에 반영할 수 없는 문제점도 있다. 따라서 본 논문에서는 정규표현식을 활용한 콘텐츠 네이밍 방식을 적용하여 기존 적응형 스트리밍 비트레이트 선택 알고리즘의 캐시활용도를 높이면서도 CCN의 기본 프로토콜에 적합한 프레임워크를 제시하고, 단말의 상태에 따라 동적인 표현식 기술 전략 및 선택 알고리즘을 통하여 비디오 스트리밍 품질을 개선하고자 한다.

**Key Words** : CCN, HTTP adaptive streaming, Regular expression, Cache utilization, Context Aware

### ABSTRACT

Content Centric Network (CCN) has been introduced as a new paradigm due to a shift of users's perspective of using Internet from host-centric to content-centric. On the other hand, a demand for video streaming has been increasing. Thus, Adaptive streaming has been introduced and researched for achieving higher user's satisfaction. If an architecture of Internet is replaced with CCN architecture, it is necessary to consider adaptive video streaming in CCN according to the demand of users. However, if the same rate decision algorithm used in Internet is deployed in CCN, there are a limitation of utilizing content store (CS) in CCN router and a problem of reflecting dynamic requirements. Therefore, this paper presents a framework adequate to CCN protocol and cache utilization, adapting content naming method of exploiting regular expression to the rate decision algorithm of the existing adaptive streaming. In addition, it also improves the quality of video streaming and verifies the performance through dynamic expression strategies and selection algorithm of the strategies.

\* 본 논문은 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구 결과물임을 밝힙니다.(2013R1A2A2A01016562)

\*\* 본 논문은 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구 결과물임을 밝힙니다. (B0190-15-2013, 유무선 통합 네트워크에서 접속 방식에 독립적인 차세대 네트워킹 기술 개발)

◆\* First and Corresponding Author: Seoul National University Department of Computer Science and Engineering, dhson@mmlab.snu.ac.kr, 학생정회원

\* Seoul National University Department of Computer Science and Engineering, {djchoi, jhsong}@mmlab.snu.ac.kr, 학생회원, tkkwon@mmlab.snu.ac.kr, 정회원

\*\* Bell Labs, nakjung.choi@alcatel-lucent.com, 정회원

논문번호 : KICS2015-05-158, Received May 28, 2015; Revised July 24, 2015; Accepted August 20, 2015

## I. 서 론

현재의 인터넷의 구조는 호스트 간 통신을 목적으로 설계되었다. 하지만 최근 인터넷은 웹페이지, 동영상 등 다양한 종류의 콘텐츠에 대한 공유의 목적으로 많이 이용되고 있으며, 이러한 추세는 앞으로도 더욱 심화될 것으로 보인다.<sup>[1]</sup> 인터넷의 사용 목적은 변화하였지만, 인터넷의 구조는 여전히 호스트 중심이기 때문에 비효율적인 데이터 전송이 이루어지고 있다. 예를 들면, 사용자들은 콘텐츠를 요청하기 위해 어떤 호스트가 해당 콘텐츠를 보유하고 있는지 알아야 하며, 또한 한 콘텐츠를 요청하는 모든 사용자들이 해당 콘텐츠를 보유하고 있는 서버로 콘텐츠를 요청하여 서버의 과부하를 유발한다. 그리고 동일한 콘텐츠를 전달하기 위해 모든 연결에 같은 콘텐츠를 각각 전송하여야 하므로 불필요한 네트워크 트래픽이 증가한다.

이러한 인터넷의 구조적 문제점을 근본적으로 해결하기 위하여 호스트 주소로 연결을 하는 것이 아니라, 콘텐츠 이름에 기반하여 콘텐츠를 요청하고 라우팅이 이루어지는 콘텐츠 중심 네트워크(Content Centric Network, 이하 CCN)라는 새로운 인터넷 패러다임이 소개되었다.<sup>[2]</sup> 이러한 패러다임을 이용하여 보다 효율적인 콘텐츠 전송을 위한 캐싱(Caching) 방법<sup>[3]</sup>, 라우팅 방법 등 다양한 CCN 관련 연구가 진행 중에 있다.

CCN에서는 i) 콘텐츠 요청을 위한 관심(Interest) 패킷, ii) 관심 패킷의 응답인 데이터(Data) 패킷이 사용된다. 관심 패킷은 크게 콘텐츠 이름, Selector, Nonce로 구성되어 있고, 데이터 패킷은 콘텐츠 이름, Signature, Signed Information, 데이터로 구성되어 있다.<sup>[2]</sup> 콘텐츠를 요청하고자 하는 사용자는 콘텐츠의 이름을 식별자(Identifier)로 하여 관심 패킷에 담아 네트워크 내로 전송하고, 네트워크 내 노드들은 해당 콘텐츠를 보유한 가장 가까운 콘텐츠 배포자(Publisher)에게 관심 패킷을 전달(Forwarding)한다. 콘텐츠 배포자는 관심 패킷이 도달하면 그에 해당하는 데이터 패킷을 전송하게 된다. 이 때, 전송 경로상의 중간 노드들은 해당 콘텐츠를 캐싱하고, 따라서 이후 요청에 대해서는 콘텐츠 배포자로 관심 패킷이 도달하기 전에 중간 노드에서 해당 콘텐츠에 대한 데이터 패킷을 미리 전송할 수 있다. 이런 특징으로 인해 네트워크 내에 불필요한 트래픽 등이 제거되고, 사용자는 더 빠르게 콘텐츠를 수신할 수 있다.

이러한 CCN의 장점들에 주목하여, 콘텐츠를 효과적으로 저장할 수 있도록 하는 캐싱 및 효율적으로 네트워크를 구성하기 위한 라우팅 및 콘텐츠 전송의 신

뢰성을 확보할 수 있는 알고리즘 등 다양한 분야에서 연구가 진행되고 있다. 그와 더불어 현재 인터넷에서 사용되고 있는 다양한 서비스를 CCN 상에서 구현하기 위한 연구도 함께 진행되어 왔다. 하지만, 사용자의 네트워크 및 단말 상태에 따른 적응형 스트리밍에 관련된 수요가 꾸준히 증가하고 있음에도 불구하고 CCN 상에서 적응형 스트리밍을 효과적으로 서비스를 하기 위한 방법에 대한 연구는 많이 이루어지고 있지 않다.

따라서 본 연구는 CCN 상에서 효율적인 적응형 비디오 스트리밍 서비스를 위한 방안을 모색하고, 사용자의 안정성과 화질을 고려한 정규표현식 활용 방법 및 라우터 설계를 제안하고 성능을 검증하는 것을 목표로 한다. CCN 상의 적응형 비디오 스트리밍 기술은 CCN의 기본 전송 메커니즘에 부합하면서도, 하나의 콘텐츠에 존재하는 다양한 품질에 대한 요청을 할 수 있어야 하며, 이를 통해 사용자의 네트워크 및 단말 상태에 따른 서비스를 할 수 있어야 한다. 이를 위해 하나의 콘텐츠에 대한 식별자는 다양한 품질을 표현할 수 있으면서, 하나의 콘텐츠를 대표할 수 있어야 한다. 본 논문에서는 정규표현식(Regular Expression)을 콘텐츠의 이름으로 활용하여 하나의 콘텐츠를 식별하고, 식별자 내에서 다양한 품질의 스트리밍 서비스를 나타낼 수 있도록 하는 콘텐츠 네이밍(Naming) 방안을 제안한다. 이러한 네이밍 방식을 사용하면 하나의 콘텐츠를 나타내야 하는 CCN의 프로토콜을 벗어나지 않으면서도 다양한 품질 후보군을 나타낼 수 있으므로, CCN 라우터에 캐싱되어 있는 콘텐츠를 우선적으로 전송이 가능하기 때문에 높은 대역폭을 바탕으로 품질 높은 적응형 스트리밍 서비스가 가능하게 된다.

구체적으로, 본 논문에서 제시한 적응형 스트리밍 서비스에서 사용자의 만족도를 높일 수 있도록 하기 위해 사용자의 단말 상태에 따른 요구사항<sup>[4]</sup>과 이에 적합한 정규표현식 표현 전략을 제시하고, 상황에 맞게 선택할 수 있도록 하는 콘텐츠 네이밍 방안을 제시한 후 기존 적응형 스트리밍 알고리즘에 적용시켜 실험을 통해 성능을 검증한다. 이러한 연구는 CCN 상에서 사용자의 상황을 고려한 적응형 스트리밍 기술에 대한 연구의 초석이 될 것으로 기대한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 CCN의 동작 메커니즘을 변경시키지 않고 적응형 스트리밍 서비스를 제공하기 위해 필요한 요구사항을 정리하고, 이를 만족시키기 위해 정규표현식을 활용하는 방안 및 사용자의 만족도를 향상시키기 위해 단말 상황에 따른 효율적인 정규표현식 표현방안을 제시한

다. 3장에서는 효율적인 적응형 스트리밍 서비스를 위해 2장에서 제시한 정규표현식의 선택 전략을 제시하고 4장에서 실험을 통해 이를 검증한다.

## II. 정규표현식을 활용한 CCN에서의 적응형 스트리밍 서비스

### 2.1 CCN에서의 효율적인 적응형 스트리밍 서비스를 위한 정규표현식 활용 방안

일반적으로 적응형 스트리밍에서는, 클라이언트가 높은 대역폭을 확보할수록 높은 성능으로 이어진다. CCN에서 클라이언트에게 높은 대역폭을 할당하기 위한 방법은 여러 가지가 있으나, CCN 라우터에 있는 캐시 활용도를 높이는 것이 중요하다. 따라서 CCN에서 캐시 활용도를 높이기 위한 정규표현식 활용방안에 대해 제안하고자 한다.

정규표현식은 문자열에 대한 규칙을 나타내는 식으로, 다양한 문자열을 하나의 문자열 형태로 나타낼 수 있는 특징이 있다. 이러한 정규표현식의 특징을 활용하여 콘텐츠를 요청하면, CCN 프로토콜의 요구사항인 하나의 콘텐츠에 대해 하나의 이름을 가져야 하는 규칙 따르면서도 CCN 라우터에 존재하는 캐시의 히트(Hit) 확률을 높여주어 사용자 만족도를 향상시킬 수 있다는 장점이 있다.

구체적인 정규표현식의 활용 방안으로, 'OR' 연산자 및 와일드카드 등으로 중의적인 표현을 사용하면 다양한 품질의 단일 콘텐츠를 하나의 관심 패킷으로 요청할 수 있다. 정규표현식을 활용하면 중의적인 문장을 하나의 콘텐츠 이름으로 해석하게 할 수 있기 때문에 하나의 콘텐츠가 하나의 고유한 이름을 가져야 하는 CCN의 기본적인 프로토콜에 부합한다. 그러나 이러한 중의적 정규표현식을 사용하지 않을 경우 같은 콘텐츠에 대해 품질 별로 다른 이름을 부여하게 되며, 이는 한 콘텐츠의 이름을 요청해야 하는 CCN 프로토콜에 맞지 않는다. 또한, 궁극적으로 콘텐츠 이름의 개수를 증가시키기 때문에 라우팅 및 캐싱에 있어 확장성(Scability)의 문제를 야기할 수 있다. 정규표현식을 활용할 경우 동일한 콘텐츠에 대해서는 같은 이름을 사용하므로, 콘텐츠 식별자의 개수를 증가시키지 않으며, 이는 네트워크의 라우팅 및 캐싱의 검색 연산의 복잡도를 낮추는 효과가 있다. 그리고 해석한 결과가 하나의 콘텐츠 이름으로 나오기 때문에 CCN 프로토콜에도 적합하다. 또한 CCN상에서 적응형 스트리밍 기술 구현을 위해 다양한 품질의 단일 콘텐츠를 하나의 정규표현식으로 표현하여 요청할 경우, 유

니캐스트(Unicast) 형태의 콘텐츠 요청이 아닌 애니캐스트(Anycast) 형태로 콘텐츠를 요청함으로써, 네트워크 내 존재하는 라우터의 캐 히트의 확률을 향상시킬 수 있는 장점이 있다.

그림 1을 보면, 정규표현식을 활용하지 않은 경우 콘텐츠  $C$ 에 대한 특정 품질  $q1$ 에 대한 단일 요청  $C(q1)$ 만을 보낼 수 있다.<sup>[10]</sup> 반면, 그림 2를 보면 정규표현식 문법 중 'OR' 관계를 이용하고 있다. 이 경우  $C$ 의 품질  $q1, q2, q3$ 에 대해 하나의 정규표현식  $C(q1|q2|q3)$ 으로 표현하여, 동일한 콘텐츠에 대해 다양한 후보(Candidate) 품질을 하나의 관심패킷을 통해 요청할 수 있다. 이와 관련된 연구로 여러 종류의 데이터를 요청할 때 그룹 관심 패킷으로 보내어 트래픽을 감소시키는 연구는 있으나, 특정 시점에서 많은 요청을 한 번에 보내면 변화하는 사용자 단말의 상태를 반영하기 어렵다는 문제점이 있어 스트리밍에는 적합하지 않다.<sup>[9]</sup>

이러한 정규표현식 활용 방법은 콘텐츠 요청자가 정규표현식을 정의하면 CCN 라우터에서 해석하는 것을 기본으로 하고 있으며, 네트워크에 존재하는 CCN 라우터가 캐싱하고 있는 콘텐츠를 우선하여 해석한다면 캐시 히트(Cache Hit) 확률을 더 높일 수 있다. 이는 콘텐츠를 보유하고 있는 배포자(Content Publisher)까지 관심 패킷이 도달하지 않고, 요청 경로(Request Path)상에 존재하는 라우터에서 캐싱된 콘텐츠

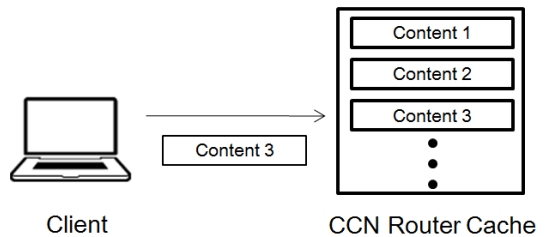


그림 1. 일반 CCN에서의 유니캐스트 기반 콘텐츠 요청방식  
Fig. 1. The content request based on uni-cast in CCN

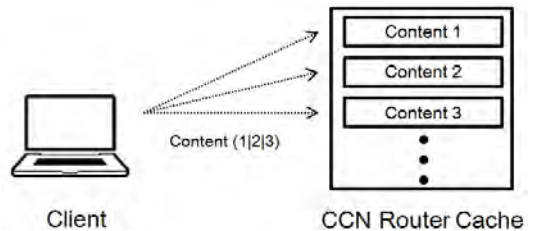


그림 2. CCN에서 정규표현식을 통한 애니캐스트 기반 콘텐츠 요청방식  
Fig. 2. The content request based on any-cast, inflecting regular expression in CCN

트를 전송할 수 있기 때문이다. 다시 말해, 정규표현식을 사용하여 CCN 상에서 적응형 스트리밍 서비스를 할 경우, 네트워크 내 CCN 라우터의 캐시 히트 확률을 높여 콘텐츠 요청자는 요청 경로 내에 존재하는 라우터로부터 짧은 시간 내에 콘텐츠를 받아올 수 있으며, 이는 결과적으로 콘텐츠 요청자에게 높은 대역폭 (Bandwidth) 할당을 하는 효과를 준다. 적응형 비디오 스트리밍 기술이 사용가능한 대역폭을 바탕으로 요청할 세그먼트의 비트레이트를 조절한다는 점을 고려해 볼 때<sup>6-8,14)</sup>, 정규표현식은 콘텐츠 요청자의 평균 비트레이트를 향상시킬 수 있으며 따라서 사용자 만족도를 높일 수 있게 된다. 추가적으로, 관심 패킷이 콘텐츠 배포자까지 전달되지 않으므로, 데이터 전송의 홉 (Hop) 수를 줄여 네트워크 트래픽을 절감하는 효과도 있다.

## 2.2 사용자 상황을 고려한 전략별 정규표현식 정의 방안

효과적인 적응형 스트리밍 서비스를 위해 정규표현식을 활용하면 애니캐스트 기반의 전송 방식을 활용할 수 있게 되어 캐시 친화적(Cache Friendly) 요청을 할 수 있다. 뿐만 아니라, 사용자의 경험 품질(Quality of Experience, QoE)에 맞는 정규표현식을 선택함으로써 사용자 만족도를 향상시킬 수 있다. 본 절에서는 사용자의 경험 품질을 향상시킬 수 있는 대표적인 적응형 스트리밍 전략 선택 전략을 알아보고, 이에 부합하는 정규표현식 표현 전략에 대해서 소개한다. 이를 통해, 사용자의 현재 상황 및 콘텐츠 종류 등 다양한 요소에 적합한 적응형 스트리밍 전략을 제시하고 이에 대응하는 정규표현식을 선택하도록 하여 사용자의 만족도를 향상시킬 수 있도록 한다.

표 1에서 알 수 있듯이, 현재까지 연구된 적응형 스트리밍 알고리즘의 전략은 크게 i) 공격적, ii) 보수적, iii) 혼합적 3가지로 구분된다. 공격적 전략은 고품질의 서비스를 추구하는 사용자가 주로 택하는 방법으로, 기존 적응형 스트리밍 알고리즘에서 정한 비트레이트( $B_r$ )보다 높은 비트레이트(즉, 고품질의 서비스)를 요청한다. 보수적 전략은 반대로 정해진 비트레이트 보다 낮은 비트레이트를 요청하며, 이것은 버퍼링 민감도가 높은 사용자가 안정적인 스트리밍 서비스를 위해 주로 택하는 전략이다. 혼합적 전략은 공격적 및 보수적 전략을 모두 사용하는 전략으로, 하나의 전략에 편중되지 않도록 하는 사용자에게 적합한 전략 전략이다.

표 1. 전략별 정규표현식 정의 방법  
Table 1. Regular expression definition of strategies

Strategy	Regular expression definition
Aggressive	<VideoClipName><SegNum> ( $B_r$   $B_r + \Delta$   ... )
Conservative	<VideoClipName><SegNum> ( $B_r$   $B_r - \Delta$   ... )
Hybrid	<VideoClipName><SegNum> ( $B_r$   $B_r + \Delta$   $B_r - \Delta$   ... )

각 스트리밍 알고리즘 선택 전략은 모두 OR 연산을 통해 정규표현식으로 기술할 수 있다. 표 1에서 VideoClipName은 요청 콘텐츠 이름이고 SegNum은 적응형 스트리밍에서 전송 단위로 인코딩 된 세그먼트 번호를 뜻한다. 델타( $\Delta$ )는 현재 사용 중인 적응형 스트리밍 알고리즘의 비트레이트와 사용자가 요구하는 비트레이트의 차이로, 각 전략별 선택의 강도를 의미한다. 즉, 이 값이 높을수록 비트레이트의 변화 정도가 크며, 낮을수록 현재 서비스되고 있는 상태에서 점진적으로 변화한다는 것을 의미한다. 표 1에서 알 수 있듯이, 공격적인 전략은 델타를 추가하는 형태로, 보수적인 전략은 델타를 감소시키는 형태로, 혼합적인 전략은 두 전략을 모두 포함하는 형태로 OR 연산을 통해 정규표현식으로 표현된다.

정규표현식을 통한 요청은 다양한 비디오 품질 요청을 하나의 식으로 표현하는 것이기 때문에 관심 패킷을 받은 노드가 다양한 비디오 품질 요청에 대해 모두 검색을 수행할 수 있어야 하며, 해당 콘텐츠에 대해 여러 품질을 모두 가지고 있을 경우 하나의 품질을 선택하여 요청자에게 전송해야 한다.

## III. 단말에 상태에 따른 정규표현식 선택 전략

### 3.1 적응형 스트리밍에서 사용자 단말 상황에 따른 사용자의 요구사항 변화

적응형 비디오 스트리밍에서는 사용자의 경험 품질 뿐만 아니라 단말 상태에 따라서 비트레이트를 선택하는 것 또한 사용자 만족도에 영향을 준다. 사용자 단말의 상태를 나타내는 대표적인 지표로는 현재 확보하고 있는 버퍼 잔량이 있다.<sup>14)</sup> 따라서 본 논문에서는 단말의 상태를 버퍼를 중심으로 파악하고 표현하도록 하였다. 단말의 비트레이트 결정 알고리즘(Rate Decision Algorithm)이 사용되는 시기는 한 세그먼트를 받은 직후이며, 이때 과거의 할당된 대역폭을 바탕으로 비트레이트를 결정한다. 이 시기는 재생시간과 비교하여 크게 세 가지로 나눌 수 있으며, 이 시기는

보유하고 있는 버퍼의 증감에 영향을 준다.

$$\frac{Bandwidth_{Real} - Bandwidth_{Predicted}}{Adaptation\ Period} \approx Buffer_{Gain} \quad (1)$$

$$Buffer_{Retention} + Buffer_{Gain} = Buffer_{Retention} \quad (2)$$

(1)은 HAS 알고리즘에서 예측하였던 대역폭 ( $Bandwidth_{Predicted}$ )과 실제 대역폭( $Bandwidth_{Real}$ )의 차이에 따라 버퍼에 주는 영향( $Buffer_{Gain}$ )에 대한 것이며, (2)는 실제 보유하고 있는 버퍼 잔량 ( $Buffer_{Retention}$ )에 주는 영향에 관한 것이다. 만약 예측하였던 대역폭이 실제 대역폭보다 더 크다면 비트레이트 결정의 시기는 한 세그먼트의 재생 시간보다 느리고, 이 경우 보유하고 있는 버퍼의 잔량은 감소한다. 반면 예측하였던 대역폭이 실제 대역폭보다 작다면 비트레이트 결정의 시기는 한 세그먼트 재생 시간보다 빠르며, 버퍼 잔량의 증가를 기대할 수 있다. 만약 예측한 대역폭과 실제 대역폭이 같을 경우 버퍼는 늘 하나만 필요하게 되며, 메모리 소비 측면 및 비디오 서비스 품질에서 가장 좋은 결과를 보여준다.

버퍼 잔량의 변화는 사용자 만족도를 위해 지향해야 할 서비스의 품질과 밀접한 관련이 있다. 위에서처럼 대역폭 할당과 대역폭 사용의 관계에서 보면, 버퍼 잔량이 줄어들었다는 것은 할당된 대역폭보다 높은 비트레이트를 선택하였다는 것을 뜻하며, 따라서 높은 화질의 영상 서비스가 가능하나 남아있는 버퍼의 수의 부족으로 인해 버퍼링 현상이 일어나기 쉽다.<sup>[8]</sup> 반면 버퍼의 잔량이 늘어나게 되면, 단말에서는 버퍼링 현상이 적어지고 안정적인 비디오 스트리밍 서비스가 가능하나 실제 대역폭보다 낮은 대역폭을 활용하였기 때문에 높은 화질의 영상을 서비스하기 힘들다. 따라서 버퍼링 현상이 없으면서도 높은 화질의 서비스를 할 수 있도록 비트레이트 선택을 통해 버퍼의 잔량을 적절히 조절하는 것이 사용자의 경험품질을 증가하는데 반드시 필요하다고 할 수 있다.

### 3.2 사용자 단말 상태에 따른 정규표현식 선택 알고리즘

적응형 스트리밍 서비스를 받는 사용자 단말의 상태는 대역폭의 변화에 따라 달라지기 때문에 하나의 전략으로 어느 상황에서나 높은 사용자의 경험품을 유지하기는 어렵다. 따라서 사용자 상태에 따라 다른 정규표현식 전략을 동적으로 선택하는 방안이 대해 소개하고자 한다.

적응형 스트리밍 사용자 단말의 버퍼 잔량은 매우 다양하기 때문에 두 가지의 Threshold를 기준으로 세 가지의 상태로 분류하고자 한다. Threshold는 비디오 화질 상상을 위한 지표인 ' $Threshold_{upper}$ '와 스트리밍 안정성을 위한 지표인 ' $Threshold_{lower}$ '로 정하기로 한다. 단말이 확보하고 있는 버퍼잔량이 Threshold들과 비교하였을 때 해당하는 위치에 따라 단말 상태가 달라지며, 이 Threshold들의 범위는 (3)과 같다.

$$0 < Threshold_{lower} < Threshold_{upper} < TargetBuffer_{max} \quad (3)$$

먼저  $Threshold_{upper}$ 의 경우는 적응형 스트리밍 알고리즘이 목표로 하는 최대 버퍼 ( $TargetBuffer_{max}$ )보다는 적으나 영상 재생 시 끊김의 확률이 매우 적은 버퍼의 기준이다.  $Threshold_{upper}$ 와  $TargetBuffer_{max}$  사이에 사용자 단말의 버퍼 잔량이 존재하면 안정 구간에 있다고 할 수 있다. 단말의 버퍼 잔량이 이 안정 구간에 해당하면 버퍼는 매우 안정적인 상황이나 현재 단말에 할당된 대역폭을 충분히 사용하지는 못하는 상태로 볼 수 있다. 반면  $Threshold_{lower}$ 보다 사용자 단말의 버퍼 잔량이 적을 경우 할당된 대역폭을 충분히 사용하고 있는 상태지만 버퍼가 불안정한 상태로 버퍼링 현상이 일어나기 쉽다.

적응형 스트리밍 서비스를 받는 단말의 안정성과 비디오 화질을 위해 두 Threshold에 따른 버퍼를 고려한 정규표현식 전략 선택은 코드 1과 같다. 만약 현재 단말의 버퍼 잔량이  $Threshold_{upper}$ 보다 큰 상황에서

**PROCEDURE** adaptationRegularExpression(Buffer, RDA\_Bitrate):

```

IF buffer >  $Threshold_{upper}$  THEN
    Regular_Expression = Aggressive(RDA_Bitrate)
ELSE IF buffer <=  $Threshold_{lower}$  THEN
    Regular_Expression = Conservative(RDA_Bitrate)
ELSE
    Regular_Expression = Hybrid(RDA_Bitrate)
ENDIF
RETURN Regular Expression
    
```

코드 1. 단말의 버퍼 상태에 따른 정규표현식 선택 전략 코드

Code 1. Pseudo code of the decision strategy based on device's buffer status

는 버퍼가 안정적이라고 간주한다. 따라서 공격적인 비트레이트 선택 전략을 택하여 비디오 화질을 높인다. 만약  $Threshold_{lower}$  보다 단말이 보유한 버퍼 잔량이 적으면 보수적인 비트레이트 선택 전략을 택하여 안정성을 높인다. 만약 단말의 버퍼의 상태가 두  $Threshold$  사이에 있으면 혼합형 비트레이트 선택 전략을 선택하여 화질과 안정성의 균형을 맞춘다. 이와 같이 버퍼의 상태에 따라 정규표현식 전략을 다르게 선택하여 적응형 스트리밍 서비스를 받는 사용자의 경험 품질을 증가시킬 수 있다.

실제로는 사용자마다 요구사항 및 만족도가 다른데, 이는  $Threshold$  값들을 조절함으로써 보다 세밀하게 충족시킬 수 있다. 예를 들면,  $Threshold_{upper}$  를 보다 높게 설정하면 공격적인 비트레이트 선택 전략을 택하는 구간이 좁아지기 때문에 안정성이 높아지고, 낮게 설정하면 같은 구간이 넓어지기 때문에 화질 상승이 가능하다. 한편,  $Threshold_{lower}$  를 낮게 설정하면 보수적인 비트레이트 선택 전략을 택하는 구간이 좁아지기 때문에 화질이 높아지고, 높게 설정하면 같은 구간이 넓어지기 때문에 안정성이 높아진다. 위와 같이  $Threshold$ 를 조절함으로써 본 프레임워크의 동작을 보다 세밀하게 조절할 수 있고, 이를 통해 다양한 사용자 요구사항에<sup>[4]</sup> 유연하게 대처할 수 있다.

#### IV. 정규표현식을 위한 CCN 라우터 설계 및 실험

##### 4.1 정규표현식을 위한 CCN 라우터 설계

본 논문에서 제안한 CCN에서의 적응형 스트리밍 프레임워크의 동작을 위해서는 콘텐츠 네트워크 내 라우터에 정규표현식을 해석하고 이와 관련된 연산을 수행하기 위해 그림 3의 붉은 부분의 모듈이 추가되어야 한다. 즉, CCN 라우터가 요청 콘텐츠 이름에 따라 CS, PIT, FIB 색인을 검색하기 전에, 관심 패킷 내의 정규표현식 부분을 먼저 추출하여 콘텐츠 단위로 해석해야 한다. 그 후 각 콘텐츠에 대해 CS 검색을 수행하고, 콘텐츠가 존재하는 경우 해당 콘텐츠를 데이터 패킷으로 만들어 전송한다. 만약 CS에 콘텐츠가 존재하지 않는 경우에는 관심 패킷 내의 이름 부분에 기반하여 기존의 CCN 라우터와 동일한 작업을 수행한다.

코드 2는 이러한 정규표현식 관련 연산 모듈의 상세 알고리즘을 수도 코드로 표현한 것이다. 관심 패킷에는 목적에 따라 자유롭게 이용될 수 있는 Nonce 필드가 존재한다. 정규표현식을 이 필드에 넣으면 관심

패킷에 존재하는 콘텐츠 이름은 변하지 않으므로 기존의 CCN 라우터 프로세스를 바꾸지 않고 단순한 모듈 추가만으로 적응형 스트리밍 프로토콜 동작이 가능하게 된다. 따라서 코드 2에서 알 수 있듯이, 본 논문에서는 캐시 검색 연산을 최소화하기 위하여 왼쪽에 정의된 요청일수록 우선순위가 높도록 판정하여 최초로 검색되는 콘텐츠를 요청자로 전송하게 했다. 그리고 CCN 라우터는 우선 관심패킷의 Nonce 부분에 저장되어 있는 콘텐츠에 대한 정규표현식을 해석하여 콘텐츠의 리스트에 저장한다. 그 후 저장된 각 콘텐츠에 대해 CS 검색을 통해 콘텐츠가 존재할 경우에는 해당 콘텐츠를 Data 패킷에 넣어서 전송하고, 그렇지 않을 경우에는 관심 패킷의 콘텐츠 이름 부분을 이용하여 일반적인 CCN 라우터의 연산을 수행한다. 클라이언트가 현재 상태 및 경험 품질에 따라서 필요한 품질의 콘텐츠를 정규표현식으로 요청하기 때문에, 최종적으로 클라이언트의 적응형 스트리밍 알고리즘이 동작하게 된다.

그림 3 및 코드 2와 같은 모듈의 추가는 하나의 관심 패킷에 대해 기존의 CCN 라우터에 비해 추가적인

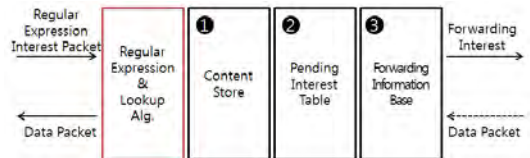


그림 3. 정규표현식 요청을 처리하기 위한 CCN 라우터 설계 및 흐름도  
Fig. 3. Design of CCN Router for processing regular expression request

```

PROCEDURE RegularExpressionModule
(interest_packet):
    regular_expression<- (interest_packet
->nonce);
    contents_in_expression ← TranslatingAndGetC
ontents(regular_expression);
    FOREACH content IN contents_in_expression
        IF CS_lookup(content) is TRUE THEN
            RETURN data packet with content
        ENDIF
    ENDFOREACH
ENDPROCEDURE
... Ordinary CCN Process ...
    
```

코드 2. CCN 라우터에서 정규표현식 처리를 위한 콘텐츠 검색 알고리즘 코드  
Code 2. Pseudo code for processing regular expression Interest packet in CCN router



처리 시간을 요구한다. 하지만, 적응형 스트리밍을 위해 일반적인 콘텐츠 라우팅 방식을 사용할 경우, 더 많은 관심 패킷이 콘텐츠 배포자에게 전달되므로 전체적인 네트워크의 트래픽이 증가하게 되고, 라우터 내에 패킷마다 입출력 시간이 존재하기 때문에 결과적으로 더 많은 시간이 소요된다. 따라서 정규표현식 관련 모듈을 콘텐츠 라우터의 기존 모듈 앞에 추가하는 것이 적응형 스트리밍의 구현에 있어서 더 효과적이다.

#### 4.2 실험정의 및 환경

CCN에서 적응형 스트리밍 알고리즘의 동작 및 효과를 확인하기 위해 'Network Simulator 3'<sup>[16]</sup>에 기반하여 네트워크 토폴로지를 구성하고, Threshold 값에 따라 변하는 경험 품질과 안정성과 평균 비트레이트의 개선을 확인한다.

트리 (Tree) 형태의 토폴로지를 기반으로, 하나의 비디오 콘텐츠 배포자와 두개의 CCN 라우터, 그리고 10개의 클라이언트를 그림 4와 같 형태로 구성하였다. 콘텐츠 배포자에 병목 현상을 가정하기 위해 콘텐츠 배포자와 바로 연결된 CCN 라우터의 대역폭은 5Mbps, 지연 시간은 50ms로, CCN 라우터끼리 연결된 링크의 대역폭은 10Mbps, 지연시간 10ms로, 그리고 CCN 라우터와 클라이언트 간의 링크는 대역폭으로 3Mbps, 지연시간 1ms로 설정하였다.

약 10분 정도의 영상을 재생하는 시뮬레이션으로, 캐시 교체 알고리즘으로는 가장 일반적인 Least Recently Used (LRU) 알고리즘을 사용하였으며, 캐시의 크기는 현재 업로드 된 전체 콘텐츠의 1%에 해당하는 5,467 바이트로 설정하였다. 사용할 적응형 스트리밍 알고리즘은 널리 상용화된 DASH 기법<sup>[3]</sup> Microsoft Silverlight Ver.2의 사용자 단말 비트레이트 결정 알고리즘 부분을 사용하여 실험을 수행하였다.<sup>[15]</sup> 그리고 각 세그먼트들은 2436Kbps, 1636Kbps,

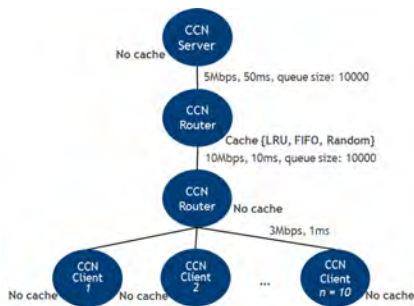


Fig. 4. Topology for simulation  
그림 4. 시뮬레이션을 위한 토폴로지

1233Kbps, 866Kbps, 608Kbps, 427Kbps, 300Kbps로 총 7개의 비트레이트로 인코딩 되었으며 각 세그먼트의 재생 시간은 2초로 가정하였다.

#### 4.3 실험결과

##### 4.3.1 Threshold 정도에 따른 적응형 스트리밍 성능 차이

그림 5와 그림 6은  $Threshold_{upper}$ 와  $Threshold_{lower}$ 의 값에 따른 10개 클라이언트의 평균 비트레이트와 평균 버퍼링 시간을 나타낸다. 그림 5에서 알 수 있듯이,  $Threshold_{upper}$ 가 낮을수록 캐시 히트율 (Hit Ratio)과 평균 비트레이트가 높다. 이것은  $Threshold_{upper}$ 가 낮아질수록 단말에서 안정상태의 구간이 넓어지기 때문에 클라이언트는 비디오 화질 상승에 초점을 맞춘 공격적 전략을 택할 확률이 높아지기 때문이다. 한편, 그림 6에서는  $Threshold_{lower}$ 가 높아질수록 버퍼링 시간이 줄어드는 결과를 확인할 수 있는데, 그 이유는  $Threshold_{lower}$ 가 높을수록 단말에서 불안정 상태 구간이 커지기 때문에 클라이언트에서 안정성을 위해 보수적 전략을 택하는 경우가 많아진다. 따라서 Threshold에 따라 다른 경험품질

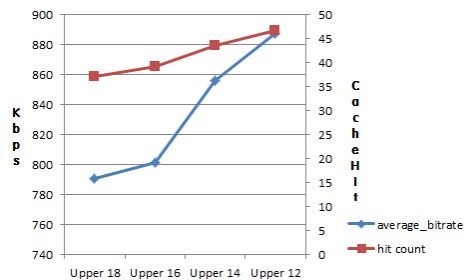


그림 5. upper threshold에 따른 평균 비트레이트와 캐시 히트율  
Fig. 5. Average bit-rate and cache hit-ratio along upper threshold

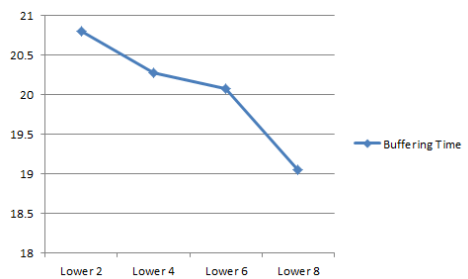


그림 6. lower threshold에 따른 평균 버퍼링 시간(초)  
Fig. 6. Average buffering time along Lower Bound(sec.)

언을 수 있기 때문에 적절히 정하는 것이 중요하며, 본 실험에서는 화질과 안정성을 위해  $Threshold_{upper}$  를 14,  $Threshold_{lower}$  를 6으로 고정하여 실험을 진행하였다.

#### 4.3.2 CCN에서 적응형 스트리밍에 정규표현식을 활용 시 성능 비교

그림 7은 일반적인 콘텐츠 중심 네트워크에서의 적응형 스트리밍 알고리즘과 본 논문에서 제안하는 스트리밍 알고리즘에 대해 콘텐츠 라우터에서의 캐시 히트율 차이를 나타낸다. 그림 7에서 알 수 있듯이, 제안된 알고리즘의 캐시 히트율이 약 96%정도 더 높은 것을 확인할 수 있다. 이는 클라이언트가 상황 및 경험 품질에 맞게 요청한 스트리밍 콘텐츠가 더 빠르게 클라이언트로 전송되었음을 의미하며, 캐시 히트율의 증가는 곧 네트워크 트래픽 양의 감소로 이어진다. 이는 제안된 알고리즘이 애니캐스트 기반으로 비디오 세그먼트 요청을 요청하기 때문에 상대적으로 일반 콘텐츠 라우팅에 비해 캐시 내 요청 콘텐츠가 존재할 확률이 높기 때문이다.

그림 8은 클라이언트 측에서 제안된 알고리즘을 사용하였을 때와 일반적인 경우에 대한 평균 비트레이트

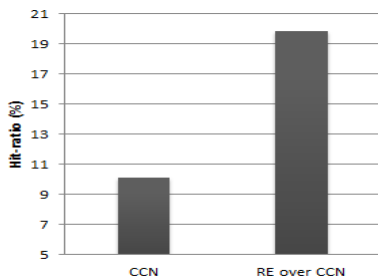


그림 7. 제안한 스킴과 현재 CCN과의 평균 캐시 Hit-ratio 비교  
Fig. 7. Comparison of cache hit-ratio between proposed scheme & current CCN

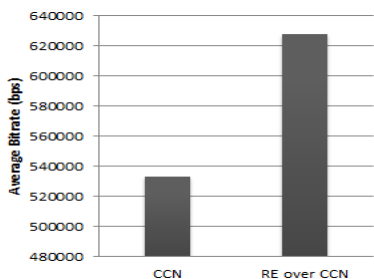


그림 8. 제안한 스킴과 현재 CCN과의 평균 비트레이트 비교  
Fig. 8. Comparison of average bit-rate between proposed scheme & current CCN

트를 나타낸다. 제안된 알고리즘에서의 클라이언트가 제공받은 스트리밍 콘텐츠의 평균 화질이 약 18%정도 향상된 것을 볼 수 있다. 이것은 그림 7에서 나타난 것처럼, 클라이언트가 요청한 데이터를 보다 빠르게 수신할 수 있기 때문에, 클라이언트는 더 높은 화질의 콘텐츠를 수신할 수 있다고 판단하여 높은 비트레이트를 요청하게 되고 이에 관련된 콘텐츠를 수신하기 때문이다.

그림 9는 스트리밍 서비스 시 발생한 단말들의 평균 버퍼링 시간을 나타낸다. 일반적인 콘텐츠 중심 네트워킹의 경우에 비해 정규표현식을 활용한 경우 클라이언트는 약 21% 정도의 낮은 버퍼링 시간을 보여준다. 이는 클라이언트가 보다 안정적으로 스트리밍 서비스를 이용한다는 것을 의미한다.

위의 결과들을 종합하면, 콘텐츠 중심 네트워킹 상에서 정규표현식을 활용하여 스트리밍 서비스를 할 때, 네트워크의 트래픽 감소로 인한 부하 감소, 클라이언트의 스트리밍 안정성 확보 및 높은 화질의 서비스가 모두 가능하다는 것을 확인할 수 있다.

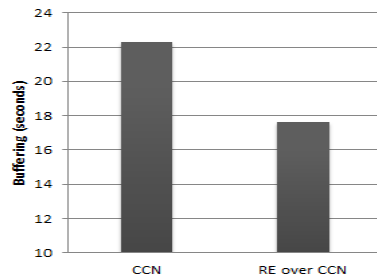


그림 9. 제안한 스킴과 현재 CCN과의 평균 버퍼링 시간 비교  
Fig. 9. Comparison of average buffering between proposed scheme & current CCN

## V. 결론

본 논문에서는 CCN에서 정규표현식을 활용하여 사용자의 상황에 따라 달라지는 요구사항을 고려한 비디오 스트리밍 프레임워크 및 알고리즘 선택 전략을 제시하였다. 정규표현식을 이용하여 콘텐츠 네이밍을 하면 CCN 전송 프로토콜에 부합하면서도 애니캐스트 기반 전송을 통해 캐시 활용도를 높일 수 있다. 따라서 단말의 입장에서 기존 스트리밍 알고리즘을 사용하였을 때 보다 더 높은 대역폭 확보가 가능하고, 또한 표현식을 달리 정의하여 사용자의 상황에 맞도록 안정적이면서도 높은 화질의 비디오 스트리밍 서



비스를 가능하게 한다. 콘텐츠 중심 네트워크에서 단말의 상황에 맞도록 정규표현식을 정의하여 활용을 하였을 때 평균 비트레이트와 버퍼링 시간 측면에서 큰 개선의 효과가 있는 것을 시뮬레이션을 통해 확인하였다.

향후에는 다른 경험 품질의 요소 중 비트레이트 변화 횟수 및 끊김 횟수, 그리고 데이터 요금을 고려하여 콘텐츠 중심 네트워크에서 비디오 스트리밍 사용자의 만족도를 높이는 정규표현식 활용 방안과 콘텐츠의 인기도를 고려한 CCN 라우터 내의 CS의 캐시 친화적인 전략에 대해 추가 연구를 진행할 계획이다.

### References

- [1] Cisco Visual Networking Index Mobile(2015), [http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white\\_paper\\_c11-520862.pdf](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf)
- [2] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proc. ACM CoNEXT*, pp. 1-12, Rome, Italy, Dec. 2009.
- [3] T. Stockhammer, "Dynamic adaptive streaming over http--: standards and design principles," in *Proc. ACM MMSys*, pp. 133-144, Feb. 2011.
- [4] A. Perkis, S. Munkcby, and O. Hillestad, "A model for measuring quality of experience," in *Proc. NORSIG*, pp. 198-201, Rejkjavik, Jun. 2006.
- [5] Y. Ito and S. Tasaka, "Quantitative assessment of user-level QoS and its mapping," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 572-584, Jun. 2005.
- [6] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *Proc. CoNEXT*, pp. 97-108, Nice, France, Dec. 2012.
- [7] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. MMSys*, pp. 169-174, Feb. 2011.
- [8] D. Suh, I. Jang, and S. Pack, "A video bitrate adaptation algorithm for DASH-Based multimedia streaming services to enhance user QoE," *J. KICS*, vol. 39, no. 6, pp. 341-349, Jun. 2014.
- [9] J. Lee and J.-H. Lee, "Secure routing scheme in CCN-Based mobile Ad-Hoc networking environments," *J. KICS*, vol. 39B, no. 5, pp. 304-308, May 2014.
- [10] J. Shin, J. Lee, and J. Lee, "Secure routing in CCN-Based mobile Ad-Hoc networking environments," *J. KICS*, vol. 39, no. 12, pp. 817-821, Dec. 2014.
- [11] F. Yu, Z. Chen, Y. Diao, T. V. Lakshman, and R. H. Katz, "Fast and memory-efficient regular expression matching for deep packet inspection," in *Proc. ANCS '06*, pp. 93-102, CA, USA, Dec. 2006.
- [12] H. Choi, J. Yoo, T. Chung, N. Choi, T. Kwon, and Y. Choi, "CoRC: Coordinated routing and caching for named data networking," *ACM/IEEE Symp. ANCS 2014*, pp. 161-172, CA, USA, Oct. 2014.
- [13] D. Saucez, L. A. Grieco, and C. Barakat, "AIMD and CCN: past and novel acronyms working together in the future internet," in *Proc. CSWS '12*, pp. 21-26, Nice, France, Dec. 2012.
- [14] H. T. Le, D. V. Nguyen, N. P. Ngoc, A. T. Pham, and T. C. Thang, "Buffer-based bitrate adaptation for adaptive HTTP streaming," in *Proc. ATC'13*, pp. 33-38, Ho Chi Minh City, Vietnam, Oct. 2013.
- [15] Microsoft Silverlight. <https://www.microsoft.com/silverlight/>
- [16] Alexander Afanasyev, Ilya Moiseenko, Lixia Zhang, *ndnSIM: NDN simulator for NS-3*, Technical Report NDN-0005, NDN Project, July 2012.

손 동 현 (Donghyun Son)



2013년 2월 : 숭실대학교 컴퓨터학부 학사 졸업  
2013년 3월~현재 : 서울대학교 컴퓨터공학 석박사 통합과정  
<관심분야> 미래인터넷, 비디오 스트리밍

송 정 환 (Junghwan Song)



2012년 2월 : 고려대학교 정보통신대학 졸업  
2012년 3월~현재 : 서울대학교 컴퓨터공학 석박사 통합과정  
<관심분야> 미래인터넷, 네트워크 가상화

최 대 진 (Daejin Choi)



2012년 8월 : 고려대학교 정보통신대학 졸업  
2012년 9월~현재 : 서울대학교 컴퓨터공학 석박사 통합과정  
<관심분야> 네트워크, 데이터 사이언스

권 태 경 (Ted "Taekyoung" Kwon)



1993년 : 서울대학교 컴퓨터 공학 학사  
1995년 : 서울대학교 컴퓨터 공학 석사  
2000년 : 서울대학교 컴퓨터 공학 박사  
2004년~서울대학교 컴퓨터 공학부 교수

<관심분야> 미래인터넷, IoT, Web/SNS Analysis, 인터넷보안, 무선네트워크

최 낙 중 (Nakjung Choi)



2009년 8월 : 서울대학교 컴퓨터공학박사  
2009년 9월~2010년 4월 : 서울대학교 박사후연구원  
2010년 4월~현재 : Member of Technical Staff, Bell Labs  
<관심분야> 미래인터넷, 모바일 네트워크 클라우드, 네트워크 가상화

일 네트워크 클라우드, 네트워크 가상화