

확률론적 조율 전략 기반의 DASH 시스템

권도현*, 김종현^o

Stochastic Tuning Strategy Based DASH System

Dohyun Kwon^{*}, Joongheon Kim^o

요약

본 논문에서는 최근 미디어 스트리밍 플랫폼들이 채택하고 있는 MPEG-DASH (DASH) 기반의 미디어 스트리밍 방법을 Markov Decision Process (MDP) 형태로 재해석하고, 이에 확률론적 조율 전략을 추가한 미디어 스트리밍 알고리즘을 제안한다. 제안하는 알고리즘에서는 이를 위해 MDP의 구성요소인 상태, 행동, 보상, 감쇠 상수, 마지막으로 전이 확률을 정의 및 결정하고, 비디오 재생의 가장 핵심적이며 실질적 데이터인 비디오 청크를 요청하는 세션들을 진행하면서 큐의 안정성과 품질 변화 빈도, 재생 정지현상과 같은 다양한 문제를 고려하고 클라이언트가 최적의 품질을 서비스 받을 수 있도록 설계한다. 이어지는 소단원들에서는 DASH의 개요를 소개하고, 및 제안하는 DASH 알고리즘에 대한 전반적 설계 사항을 소개한다. 마지막으로 이전 연구로 제안된 DASH 알고리즘들과 제안하는 알고리즘의 성능을 평균 수신 품질 관점에서 비교하는 실험을 통해 제안하는 알고리즘의 우수성을 검증한다.

Key Words : Markov Decision Process, Dynamic Adaptive Streaming over HTTP, Reinforcement Learning, State Transition Matrix

ABSTRACT

In this article, we reinterpret MPEG-DASH (MPEG Dynamic Adaptive Streaming over HTTP, or DASH) media streaming strategy, which is widely adopted by recent media streaming platforms, and transform sequential media requesting episodes utilizing Markov Decision Process (MDP) and propose a media streaming algorithm with stochastic tuning strategy. Firstly, the algorithm designs and decides elements of MDP, which consists of state, action, reward, discount factor, and transition probability. Plus, along with multiple scenarios of video chunk requesting sessions, which are the most essential and realistic data transmission procedure of entire media streaming, we carefully consider many issues such as queue stability, frequency of quality variation, and freezing phenomenon. Also, we design algorithm for client to experience optimal service under possibly drastically varying network bandwidth. In following subsections, we introduce an overview of DASH and elaborate design features of our proposed DASH algorithm. Lastly, we conduct experiments with proposed algorithm and previous MDP-based DASH system related researches by investigating experiments in terms of average quality of video chunk which impacts quality of experience of users.

※ 본 연구는 한국연구재단 신진연구자 지원사업(2016R1C1B1015406)과 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2018-0-00170, 이동체간 가상현실을 위한 5G 이동통신 기술 연구).

• First Author : (ORCID:0000-0002-9501-707X) Chung-Ang University School of Computer Science and Engineering, kdh1102@cau.ac.kr, 학생회원

o Corresponding Author : (ORCID:0000-0003-2126-768X) Chung-Ang University School of Computer Science and Engineering, joongheon@gmail.com, 중신회원

논문번호 : 201806-C-130-SE, Received February 2, 2018; Revised June 28, 2018; Accepted September 10, 2018

1. 서 론

Cisco에서 발표한 자료에 따르면, 2016년에 매 월 평균적으로 발생한 전 세계 모바일 데이터 트래픽은 2015년 대비 약 63% 증가한 7.2 엑사바이트인 것으로 집계되었다. 또한, 2015년 대비 2016년에 새롭게 추가된 모바일 디바이스 및 인터넷 연결은 약 5억 개이며, 2016년 기준 전체 모바일 트래픽의 약 60%가 모바일 비디오 트래픽인 것으로 조사되었다. 이러한 증가 추세의 모바일 트래픽은 2021년까지 약 7배 증가할 것으로 추산된다^[9]. 급증하는 모바일 데이터 트래픽 추세와 새롭게 생겨나는 모바일 기기들과 그 응용 분야가 생겨나는 추세로 미루어 보았을 때, 모바일 트래픽 전체의 대부분을 차지하는 모바일 비디오 트래픽은 앞으로도 증가할 것으로 예상할 수 있다.

1998년 표준화된 실시간 비디오 전송 프로토콜로서 RTSP 및 RTP는 각각 온 디맨드 방식으로 미디어 전송에 대한 제어 및 데이터 전송에 관여하는 동기 방식의 프로토콜이다. 하지만, 이러한 RTSP 및 RTP의 경우 out-of-band 방식의 동작으로 서로 다른 네트워크 연결을 통해 데이터를 교환하여 Network Address Translator (NAT) 이나 방화벽을 많이 사용하는 환경에서는 서비스의 품질이 떨어질 수 있다. 또한, 사용자의 대역폭과 CPU 자원의 가용성 등을 고려한 적응적 미디어 서비스보다 QoE가 저하될 수 있다는 문제점이 있다.

이러한 특징으로 인해 최근 YouTube, Netflix와 같은 HTTP Adaptive Streaming (HAS) 방식의 DASH 비디오 스트리밍 플랫폼들이 점차 많이 사용되고 있다^[10]. DASH는 사용자 측의 네트워크의 대역폭과 미디어 플레이어의 버퍼 길이, 사용자 경험을 최대한 고려하여 서비스할 비디오의 품질을 스트리밍 에피소드의 흐름에 따라 가변적으로 비디오 품질을 바꾸는 정책을 통해 스트리밍 서비스를 제공 한다.

DASH 미디어 스트리밍 서비스는 그림 1과 같이 이루어진다. 먼저 DASH 미디어 서버는 XML 형식의 Media Presentation Description (MPD) 문서를 제공함으로써 사용자가 가용한 미디어 품질 정보를 제시하고, 사용자는 미디어 플레이어의 알고리즘에 따라 서비스 받을 품질을 선택하여 해당 정보를 서버 측으로 전송한다. 이러한 일련의 과정이 반복되면서, DASH 클라이언트는 자신이 연결되어있는 네트워크 환경 수준에 따라 가용한 비디오 품질 범위를 관측하고, 재생 정지 문제와 지나친 품질 변화 현상을 지양하도록 다양한 변수들을 고려하여 적응적으로 비디오



그림 1. DASH 미디어 스트리밍 동작
Fig. 1. Operation of DASH media streaming

서비스를 요청한다. 이 때, 요청하는 정보는 비디오 전체가 아니라, 세그먼트화 된 비디오인 청크라는 것에 유의해야 한다. 미디어 서버는 서비스할 비디오를 짧은 재생시간을 갖는 청크들로 세분화 하고, 각 청크 별로 다양한 품질의 청크들을 저장하여 클라이언트의 요청에 따라 선택적으로 청크를 전송한다. 즉, 그림 1을 통해 비디오 스트리밍 서비스의 예상 시나리오로 시간의 흐름에 따라 가변적인 비디오 청크 요청이 가능함을 예상할 수 있다.

본 논문은 이러한 일련의 미디어 요청 시나리오를 MDP로 나타내고 이 속에서 더 나은 상태로 미디어 요청 에이전트가 전이할 수 있도록 상태 전이에 대한 확률 보정 과정을 도입한 새로운 DASH 시스템인 Quality-Time aware MDP (QTMDP) 을 제시한다. QTMDP는 이름에서 알 수 있듯이, 단순히 최적의 품질 요청을 위한 알고리즘이 아니라, 해당 시간에 최적의 품질을 요청할 수 있도록 시간-품질을 동시에 고려하여 설계하였다. 이를 위해 클라이언트가 행하는 일련의 미디어 요청을 MDP로 나타내고, 클라이언트가 미디어 요청이라는 행동에 따라 상태전이를 할 때, 최적의 상태로 전이할 확률을 조율해 나가는 알고리즘을 적용하였다. 또한, 이를 기존 DASH 알고리즘인 [6]의 Panda, Festive 알고리즘과의 비교를 통해 기존 연구결과 대비 개선사항 및 차이점을 소개한다.

2장에서는 QTMDP의 세부적 동작 시나리오를 그림과 함께 설명하여 데이터 채널과 컨트롤 채널이 상존하는 구조를 설명하고, 미디어 요청 시나리오를 MDP를 활용해 재표현하여 각 변수들의 정의 및 설계에 대한 설명을 제시한다. 또한 MDP외에 알고리즘의

설계에 포함된 각종 변수들과 만족되어야 할 수식들의 내용도 제시된다. 3장에서는 DASH 시스템을 위한 알고리즘들 간의 실험 환경을 도표를 활용하여 제시하고 실험 결과와 이에 대한 분석내용이 제시된다. 실험에 인용된 비디오는 다양한 압축률인 Quantization Parameter (QP) 가 적용되어 같은 비디오라고 할 지라도 압축률에 따라 매우 다른 크기를 갖는 비디오 청크들이 수신되는 것과 같은 효과를 갖는다. 이를 통해 클라이언트에게 수신된 비디오 청크 품질이 동적으로 변화할 때 알고리즘이 얼마나 유연하게 대응하는지 확인할 수 있다. 마지막으로 4장은 QTMDP의 전체적 요약과 실험을 통해 도출된 결과를 바탕으로 QTMDP가 갖는 장점을 역설하며 본 논문을 맺는다.

II. 제안 알고리즘 및 기존 연구활동

본 장에서는 먼저 QTMDP와 같이 DASH 알고리즘들에 대한 기존의 연구들을 살펴보고, QTMDP의 전체적 개요를 그림을 통해 설명한다. 또한, 활용된 수식 및 의사코드와 변수들에 대한 설계 및 정의를 세부적 설명과 함께 제시한다. 앞서 설명한 바와 같이, QTMDP는 미디어 서버로부터 DASH방식으로 비디오 청크들을 서비스 받는 시나리오를 MDP 형식으로 재표현하여 스트리밍 서비스를 실시한다. 이를 위해 이어지는 세부 소단원에서는 QTMDP를 구성하는 MDP 세부 요소들을 설계, 정의하고, 각각에 대한 세부적 설명이 제공된다. 또한, QTMDP의 행동방식을 결정하는 정책 (Policy) 을 QTMDP를 구성하는 핵심 요소인 STM (State Transition Matrix) 조율 전략으로 표현하여 일련의 에피소드의 진행에 따라 각 STM을 변화시켜 최적의 미디어 요청을 수행해 나가는 강화 학습 전략을 QTMDP의 정책에 적용함을 보인다.

2.1 관련 연구 활동

Kreuzberger 등은 미디어 서버측에서 선택하는 품질에 따라 사용자 측에서 경험하는 만족도에 얼마나 영향을 미치는지 사용자의 종류에 따라 실험한 결과를 제시한다^[5]. 플랫폼으로는 YouTube, Netflix, 그리고 Apple 등을 활용하여 다양한 DASH 비디오 서비스 플랫폼 상에서 청크의 품질을 다변화하여 사용함으로써 실제 사용자 특성을 포함한 연구 결과를 제시하였다. 이 논문에서는 하루에 걸쳐 주어진 대역폭 환경에서 사용자의 평균적 만족도와 goodput에 대한 값을 관측하였다.

Claeys 등이 제시한 알고리즘은 강화학습 기반의

DASH 에이전트 모델을 다루며 MDP를 활용한 적응적 미디어 요청 전략에 대한 조사를 실시하였다^[2, 3]. 이 알고리즘은 보상 및 페널티 값을 포함한 MDP의 구성요소들을 수식화 하고, MDP 에이전트가 따르는 정책에 의해 주어진 상태에서 최적의 행동을 지속적으로 관측하여 행동에 대한 비디오 청크를 요청한다. 또한, DASH 에이전트의 학습 경향의 수렴과 강화학습 기반의 DASH 에이전트는 다른 여타 DASH 에이전트의 성능을 상회한다는 결과도 제시한다. 논문 [2, 3]의 저자는 공통적으로 Q-learning 기법을 활용해 강화학습 기반의 미디어 품질 요청 알고리즘을 제시하였고, 청크의 품질을 학습하는 것에 초점을 맞추었다면, 본 논문은 이와 더불어 더 좋은 품질의 청크를 요청하는 것과 동시에 비디오 큐의 상태를 종합적으로 고려한 강화학습 기반의 미디어 요청 알고리즘이라고 할 수 있다.

[4]는 마찬가지로 MDP모델을 DASH 클라이언트에 적용하여 적응적 비디오 요청 알고리즘의 성능을 제시한 것이다. 저자는 k-MDP, s-MDP, x-MDP라는 세 개의 서로 다른 전략을 사용한 MDP모델을 DASH 클라이언트에 적용하여 각각의 성능을 제시하고, 상태 전이에 대한 함수와 전체 보상 함수를 상세히 수식화 하였다. 또한 이러한 함수들과 MDP를 구성하는 각각의 요소들 및 다섯 개의 품질단계를 갖는 비디오 청크들을 활용하여 정지 현상과 관련이 깊은 Deadline Miss (DM), 빈번한 품질변화로 인한 사용자 경험 저하와 연관이 깊은 Quality Change (QC), 마지막으로 평균 청크 품질인 Average chunk Quality (AQ) 관점에서 각각의 DASH 알고리즘들의 성능을 평가한다.

반면, QTMDP와 비교하고자 하는 DASH 시스템의 알고리즘인 Panda 알고리즘은 대역폭의 상황에 따라 점진적 품질개선과 급진적 품질 하락이라는 Additive-increase multiplicative-decrease (AIMD) 특성을 바탕으로 클라이언트의 대역폭과 수신한 청크의 품질 정보를 활용하여 비교적 보수적인 행동을 취하는 알고리즘이다. 또한, Festive 알고리즘은 요청된 청크별 계산된 버퍼의 포화 정도로부터 버퍼의 효율성, 안정성을 고려하기 위해 제안된 알고리즘이다. 이들 알고리즘은 새로운 DASH 알고리즘 연구주제와 관련해 널리 사용되고 있는 비교군이며, 추후 성능 평가 비교를 위한 실험에서 이 두 알고리즘을 제시하는 알고리즘과 비교하기 위한 비교군으로 활용함을 밝혀둔다.

[3]는 DASH 클라이언트의 행동을 결정짓는 최적의 미디어 품질 요청에 대한 정책의 분석이 제외되었고, [2-5] 모두 MPD와 데이터가 축적되는 큐의 안정

성에 대한 문제를 고려하지 않았다. 지속적으로 안정적인 비디오 스트리밍 서비스가 이루어지기 위해서는 큐의 안정성이 보장되는 시스템을 설계하는 것이 중요하기 때문에, 스트리밍의 핵심 데이터인 MPD와 비디오 청크 데이터를 저장하는 큐에 대한 상세한 설계가 필요하다. 따라서, 본 논문에서는 큐의 안정성과 DASH 에이전트의 행동 정책을 동시에 고려한 DASH 클라이언트 알고리즘인 QTMDP를 제안하고자 한다.

2.2 QTMDP 개요

본 소단원에서는 논문을 통해 제시하고자 하는 DASH 알고리즘인 QTMDP의 전체적 구조와 동작 원리에 대하여 서술한다. 개략적인 QTMDP의 특징은 다음과 같다. 먼저, QTMDP는 알고리즘^{3, 4}과 같이 강화학습을 활용하고, 일련의 미디어 요청을 MDP 모델을 도입하여 DASH 클라이언트가 스트리밍 서비스를 쾌적하게 받을 수 있도록 한다는 점에서 비슷하다. 좀 더 자세히 말하자면, 상태, 행동(다음 에피소드에서 요청할 비디오 청크의 품질), 상태 전이 확률, 보상, 감쇠 상수 등의 MDP 요소들을 정의하여 에피소드의 연속을 통해 DASH 클라이언트가 취할 행동을 결정하는 정책을 강화학습을 통해 발전시켜 나간다. 이 때, 상태는 이전 에피소드에서 요청한 비디오 청크의 품질 (q)과 수신 결과 버퍼의 여유 공간을 시간 영역으로 표현한 변수 (e)의 쌍으로 구성된다. 이와 같이 DASH 클라이언트의 상태를 설정한 이유는 우선 품질을 상태의 요소로 설정함으로써 다음에 요청할 청크의 품질 선택시 지나치게 가변적인 품질 변화를 발생시키지 않도록 할 수 있기 때문이다. 또한, 버퍼 공간을 시간 영역으로 재해석하여 상태를 표현하는데 활용한 이유는 동일 품질에 대한 요청을 지속적으로 수행하였을 때, 버퍼의 여유 공간에 대해 더욱 명확한 예측이 가능하여 지속적인 스트리밍 서비스를 받는데 더욱 유리하기 때문이다. 다른 알고리즘과는 달리, QTMDP는 상태 전이 함수 STM에 대한 학습을 진행하는데, 이를 통해 주어진 환경에서 최적의 품질을 갖는 청크를 요청할 확률을 증가시킨다. 즉, 최적 행동으로 전이될 확률을 각 에피소드가 진행됨에 따라 지속적으로 갱신하여 DASH 클라이언트가 취할 수 있는 행동에 대해 최적의 상태로 도달할 확률을 향상시켜 나간다. 이를 위해 상태 전이 행렬인 STM을 이전 상태(S(q, e))와 다음 상태(S(q', e'))를 활용하여 정의하고, 지속적으로 갱신하여 최적의 정책을 만들어 나간다. 이 때, STM은 3차원 좌표축으로 표현되며, 이

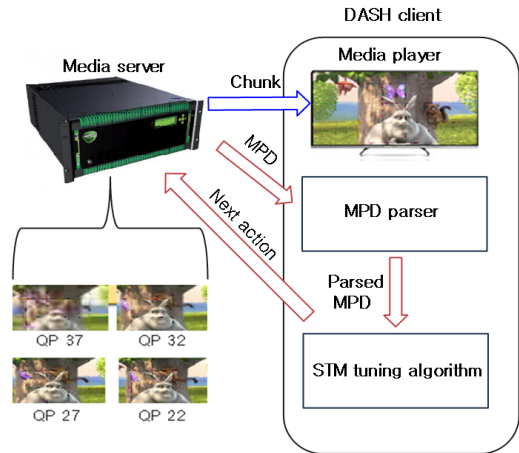


그림 2. QTMDP의 동작과정
Fig. 2. Operational mechanism of QTMDP

전 에피소드에서 시행한 요청에 의해 서비스 받은 비디오 청크의 품질 (q) 에 따른 버퍼 가용값 (e) 와 다음 시행 가능한 행동들 (q') 에 대한 버퍼 가용값 (e') 을 각각 종축과 횡축으로 갖고, 그에 따른 확률을 누적 분포 함수 (CDF) 로 표현한다.

그림 2는 이러한 QTMDP의 동작과정을 묘사한 것이다. 스트리밍 서비스의 개시는 클라이언트의 접속 및 미디어 요청에 대한 DASH 서버의 MPD 파일 전송을 시작으로 한다. MPD 파일을 전송받은 클라이언트는 MPD 파싱 프로세스를 통해 클라이언트가 이용 가능한 품질의 비디오 청크들에 대한 정보를 얻고, 클라이언트 측의 대역폭 정보 등 동영상 파일을 재생하는 데 필요한 모든 정보를 얻게 된다. 이러한 정보를 바탕으로 클라이언트는 다음 에피소드때 어떤 품질

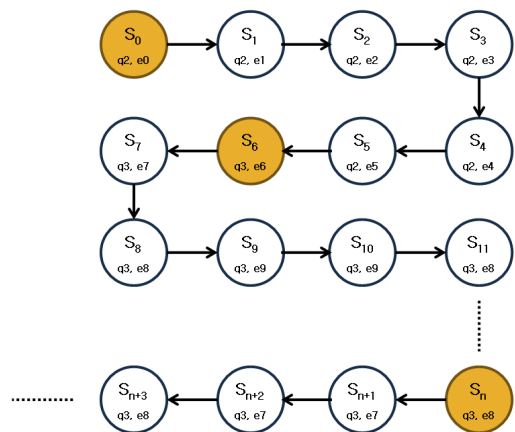


그림 3. QTMDP의 상태 변화 시나리오
Fig. 3. State transition scenario of QTMDP

(q')의 비디오를 요청할 것인지 결정할 수 있다. 이러한 일련의 스트리밍 서비스가 진행되는 동시에 QTMDP의 STM 조율도 병행적으로 이루어져 최적의 품질 선택이 가능하도록 동작한다.

2.3 QTMDP 설계 세부사항

앞서 밝힌 바와 같이, DASH 클라이언트의 일련의 미디어 요청들은 Markov Chain으로 표현된다. 시간의 흐름에 따라 가변적인 클라이언트의 상태와 행동, 누적 보상값이 발산하지 않도록 설정한 감쇠 상수, 상태 전이 확률과 마지막으로 보상값이 모두 고려되어 미디어 스트리밍이 이루어지게 된다. 본 소단원에서는 이러한 MDP 구성요소들에 대한

세부 설정 사항과 동작에 필요한 변수들 및 알고리즘을 의사 코드를 통해 설명하고, 이외에 충족되어야 할 제한사항을 소개한다.

먼저, 클라이언트의 상태는 $s(q, e)$ 와 같이 표현된다. 초기 상태의 품질 정보 q 는 최초 DASH 서버로부터 전달받은 MPD를 통해 이용 가능한 품질 가운데 중간 품질을 요청하고 해당 품질에 대한 정보를 상태의 품질 값으로 설정한다. 이를 제외한 나머지 에피소드의 상태를 구성하는 품질 값은 지속적인 MPD 문서의 참조, STM 조율 전략을 통한 최적 행동 추출, 지나치게 자주 품질 변화를 일으키지 않는 역치 변수 등의 영향을 통해 결정된다. 클라이언트의 행동은 특정 품질을 요청한다는 의미를 가지며, 본 논문에서 사용한 품질은 동일 청크에 대하여 압축률 QP를 달리한 총 네 가지(QP22, QP27, QP32, QP37)로 구성된다. 또한, 상태를 구성하는 또 다른 변수인 e 는 동일 상태

에서 수신 받은 품질에 대한 데이터 큐에 가용한 여유 공간을 시간 영역으로 재표현한 변수이다. 즉, 같은 환경에서 동일 품질을 지속적으로 e 초만큼 수신할 수 있다는 정보를 의미한다. 이를 통해 쉽게 동일 품질의 비디오 청크를 지속적으로 요청하였을 때 플레이어가 얼마나 안정적인 스트리밍 서비스를 받을 수 있는지에 대한 척도를 얻을 수 있으며, 이는 결국 사용자 경험인 QoE의 저하를 방지할 수 있다. 만일 대역폭이 더욱 고품질의 비디오 청크를 요청할 수 있는 환경이 된다면, 클라이언트는 일정 에피소드 이후(역치 변수만큼) 고품질의 비디오 청크를 요청할 수 있게 되어 빈번한 품질 변화를 하지 않으면서 재생 중지 현상도 방지하고 고품질의 비디오 청크를 서비스 받을 수 있는 안정적인 스트리밍 서비스를 경험할 수 있다.

그림 3은 일련의 QTMDP 상태변화에 대한 도식이다. 각 원은 에피소드마다 갖는 QTMDP 클라이언트의 상태를 나타내며, 화살표를 따라 상태 전이, 다시 말해 에피소드의 전환이 이루어진다. 에피소드의 시작에 위치한 노란 원은 MPD를 통해 더욱 높은 수준의 비디오 청크를 요청할 수 있음을 감지하기 시작한 상태로, 다음 노란 원까지 지속적으로 해당 품질에 대한 네트워크 대역폭이 충분하다면 바로 상위단계의 품질의 비디오 청크를 요청하지 않고, 다음 노란원에서 비디오 청크의 품질을 상향해 DASH 서버로 요청한다. 이는 빈번한 비디오 품질 변화로 인해 사용자의 QoE가 저하되지 않게 하기 위함이다. QTMDP에서 설정한 이러한 ‘역치 변수’는 기본적으로 5로 설정되어 있으며, 각 청크의 재생시간이 2초임을 고려해 품질의 상향 조절이 필요하다고 판단되면 약 10초 이후 품질의 향상을 경험할 수 있다. 이 때, e 는 전달받은 비디오 청크의 품질에 대한 버퍼의 여유 공간을 시간 영역으로 재표현한다. e 에 대한 식은 다음과 같다.

표 1. QTMDP 상태 연관 변수
Table 1. Variables of relating QTMDP state

Name	Value
QP37 (q_0)	1,600~4,900 (kbps)
QP32 (q_1)	4,900~9,600 (kbps)
QP27 (q_2)	9,600~21,300 (kbps)
QP22 (q_3)	21,300 ~ (kbps)
b_s (buffer unit size)	100Kb
M (Maximum number of buffer unit)	2,400
m (intervals in a second)	2

$$e = R(s) \times \frac{b_s}{br_q^t} \times m, 0 \leq e \leq \left[\frac{b_s}{br_q^t} \times M \times m \right] \quad (1)$$

수식 (1)의 $R(s)$ 는 버퍼에 가용한 영역을 버퍼 유닛의 개수로 표현한 변수이다. $R(s)$ 는 상태 s 일 때 가용한 버퍼 유닛 개수를 의미하고, b_s 는 버퍼 유닛의 크기를 의미하며, QTMDP의 시뮬레이션을 위해 100Kb로 설정하였다. br_q^t 는 에피소드 t 에서 특정 품질 q 에 대한 비트 레이트(bps) 값이며, 이는 표 1의 상단에 위

치한 네 개의 변수들에 해당한다.

또한, 변수 m 은 1초를 m 개의 간격을 갖도록 설정한 것으로 1초 내에서 다변화적일 수 있는 비디오 청크의 재생 시간을 처리할 수 있게 하기 위한 값이다. 마지막으로 M 은 버퍼가 모두 빈 상태의 버퍼 유닛의 총 개수로, $R(s)$ 가 가질 수 있는 최대값이다. 즉 QTMDP 클라이언트는 앞서 소개한 바와 같이 품질 정보와 버퍼 여유 공간에 대한 시간 영역의 값을 쌍으로 상태정보를 유지하여 일련의 특정 미디어 품질 요청을 지속적으로 수행해 스트리밍 서비스를 받게 된다.

이러한 시간 영역의 여유 공간을 의미하는 변수 e 는 그림 4와 같은 추이로 변화할 수 있음을 보일 수 있다. 일정한 Departure Process (DP) 로 버퍼의 데이터가 읽어진다 가정할 때, 그림 4의 하단 세 개의 에피소드의 경우 DP보다 더욱 많은 데이터가 들어온다면 ($\lambda_2 > \mu$) e_0 부터 e_2 까지 그 값이 감소하며, $S_{n+2}(q_2, e_2)$ 에 도달해 역치 값 이상의 감지가 일어났을 경우, 요청 품질을 저하시켜 $S_{n+3}(q_1, e_3)$ 상태로 상태 전이를 하고 버퍼의 여유 공간을 확보한다.

QTMDP 클라이언트는 이러한 상태 변수와 적절 품질의 비디오 청크를 지속적으로 요청하는데, 에피소드의 진행에 따라 앞서 언급한 상태 전이 확률 행렬들의 값을 조율한다. 이러한 작업을 지속적으로 반복하여 클라이언트의 행동을 결정짓는 정책인 상태 전이 확률 행렬들의 분포가 최적 상태로의 전이가 이루어질 수 있도록 수정된다. 상태 전이에 대한 확률은 인용한 [4]의 식을 변형한 것을 사용하였으며 자세한 수식은 아래와 같다.

$$P_{ee'}^n(t) = \begin{cases} 1 - F(e' - e), & \text{for } e' > e, t = 1 \\ F(\frac{e' - e}{t}) - F(\frac{e' - e}{t'}) & \text{for } e' > e, t > 1 \\ 0 & \text{for } e' < e \end{cases} \quad (2)$$

수식 (2)의 F 함수는 QTMDP 클라이언트의 대역폭에 대한 CDF이다. 현재 상태에서 선택하는 행동에 따른 다음 상태의 e' 값과 현재 상태의 e 값을 활용해 다음 상태로의 상태 전이 확률을 계산할 수 있다. 이때 t 는 현재 상태의 시간을, t' 는 현재 상태 이후 다음 청크를 수신할 때의 시간을 의미하며 두 시간 사이의 누적 확률 분포 차이값을 통해 $t > 1$ 일 때의 확률을 계산할 수 있다.

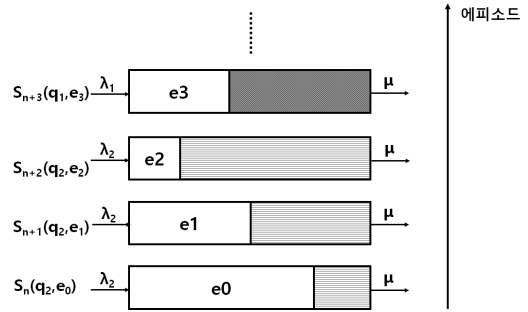


그림 4. 미디어 스트리밍 진행에 따른 e 값 추이
Fig. 4. Tendency of e value transition

그림 5는 초기 품질 별 상태 전이 확률 분포를 CDF 형태로 표현한 것으로, QTMDP 클라이언트가 미디어 스트리밍을 최초에 시작할 때 사용하며 미디어 스트리밍과 동시에 지속적으로 조율 및 개선해 나가는 정책의 일부이다. 종축과 횡축은 각각 특정 품질을 통해 가질 수 있는 버퍼 여유 시간(e 와 e')을 의미하며, 양 축과 직교하는 축은 CDF 값을 나타낸다. 즉, 본 논문에서 제시하는 비디오 청크의 품질을 QP 22, QP 27, QP 32, QP 37임을 생각하면 총 16개의 확률 분포를 얻을 수 있으나, 그림 5는 그 중 일부인 동일 품질을 재요청하였을 때의 상태 전이 확률 분포를 표현한 것이다. 그림 5의 (a)는 QP 37품질의 청크를 요청했던 클라이언트가 다시 QP 37품질의 청크를 요청할 경우의 초기 누적 확률 분포를 의미하며, (d)는 QP 22품질의 청크를 요청했던 클라이언트가 다시 QP 22품질의 청크를 요청할 경우의 누적 확률 분포를 의미한다. 나머지 (b)와 (c)는 각각 QP 32, QP 27의 경우에 해당된다.

앞서 언급한 상태 전이 확률을 조율하는 방향은 최

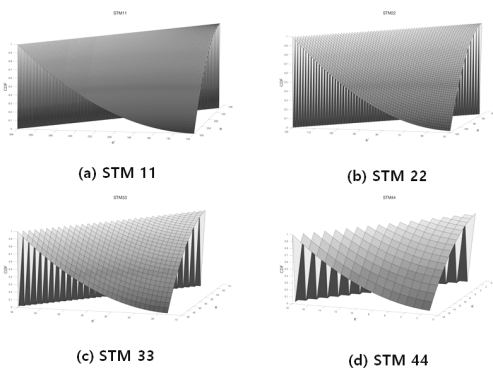


그림 5. 품질 별 초기 상태 전이 확률 분포
Fig. 5. Initial state transition probability distribution for each chunk quality

적의 행동을 결정하여 도달하고자 하는 상태에 도달하기 위한 확률의 값을 향상시키는데 있다. 이를 위해서는 최적의 행동을 판별하는 기준이 있어야 하는데, 이를 위해 가치 함수, 보상 함수, 그리고 페널티 함수를 정의할 필요가 있다. 먼저, 매 에피소드마다 클라이언트는 다음의 식들을 계산해 행동에 대한 예상 가치를 판단할 수 있다.

$$Q_{\pi}(s, a) = \sum_{\pi} T(s, a, s') [R(s, a, s') + dV_{\pi}(s')] \quad (3)$$

$$V_{\pi}(s') = \begin{cases} 0, & \text{streaming is end} \\ Q_{\pi}(s', a'), & \text{else} \end{cases} \quad (4)$$

$$R_{tot}^q(q, e) = R(q') - C(q, q') - f(e, q') \quad (5)$$

QTMDP는 수식 (4)와 수식 (5)를 수식 (3)에 적용하여 매 에피소드 별 요청 가능한 범위의 품질에 대한 청크들에 대하여 그 가치를 평가하고, 역치 변수를 고려하여 최종적으로 어떤 품질의 청크를 요청할지 결정한다. 수식 (3)의 T함수는 상태 전이 확률값이며, R함수는 수식 (5)에서 제시되는 총 보상함수, V함수는 수식 (4)의 가치함수를 의미하며, 마지막으로 d는 감쇠 상수이다. 수식 (3)을 통해 결정한 다음 청크의 품질은 클라이언트의 요청에 의해 미디어 서버로 전송되고, 그에 상응하는 응답을 받은 클라이언트는 자신의 상태와 이상적 상태의 차를 확인해 필요에 따라 상태 전이 확률 행렬을 조율하게 된다. 수식 (5)는 품질 q에 대한 비디오 청크를 수신하여 버퍼 여유 공간에 대한 시간영역 값이 e의 상태일 때 품질 q'을 요청하여 얻을 수 있는 총 보상값을 계산하는데 사용된다. 이는 q' 품질을 요청하는데서 오는 순수한 이득값에 품질 변화에 대하여 감소되어야 할 페널티 값 C(q, q')과 q' 품질을 요청하였을 때 발생될 수 있는 정지 현상에 대한 페널티 값 f(e, q')을 때 계산한다. 품질 변화에 대한 페널티 값은 인용한 [4]에서 사용한 값을 활용하며, 품질 별 순수 보상 값과 정지 현상에 대한 페

표 2. 품질 별 보상 값
Table 2. Reward values for each QP

Name	Bitrate(Kbps)	Reward value
QP37 (q_0)	1,600 ~ 4,000	1
QP32 (q_1)	4,000 ~ 9,600	2
QP27 (q_2)	9,600 ~ 26,400	5
QP22 (q_3)	26,400 ~ 40,000	10

표 3. 정지 현상에 대한 페널티 값
Table 3. Penalty values of freezing phenomenon

Freezing(sec)	Penalty value
1 to 2	1
3 to 4	2
5 to 10	4
11 to 20	8

널티 값은 각각 표 2와 표 3을 활용한다. 이 때, 비트 레이트 항목에서 볼 수 있듯이, 4K 기준 청크의 품질을 활용하여 같은 크기의 비디오라 할 지라도 편차가 큰 압축률을 기준으로 청크를 구분하여 동적으로 변화하는 네트워크의 상황에 대한 클라이언트의 유동성을 평가할 수 있다. 표 3의 페널티 값은 단계를 나눠 지수적으로 페널티 값을 증가하도록 설정하였고, 품질 별 보상 값은 품질에 대한 비트 레이트를 기준으로 보상 값을 비례적으로 정의하였다.

설계 마지막 사항으로, MPD 큐 및 데이터 큐를 유지하여 클라이언트가 요청 가능한 범위의 비디오 품질 정보 및 청크 데이터를 저장하도록 하였고, 지속적인 안정성을 위해 수식 (6)과 수식 (7)과 같은 제한 사항을 설정하였다. 수식 (6)과 수식 (7) 및 [1]를 활용하여 클라이언트 측에 존재하는 MPD 큐와 데이터 큐에 적용하여 지속적인 스트리밍 세션 동안 큐에서 오버 플로우 및 언더 플로우가 발생하지 않도록 큐의 안정성을 확보할 수 있다.

$$g(X) \cong \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau}^{t-1} \Pr[Q(\tau) > X] \quad (6)$$

$$\lim_{X \rightarrow \infty} g(X) = 0 \quad (7)$$

III. 모의 실험 및 성능 평가

본 장은 이전 장에서 살펴본 설계 사항을 직접 시뮬레이션 한 결과를 살펴보고 그에 대한 분석 결과를 다룬다. 결과를 살펴보기에 앞서 시뮬레이션 환경에 대한 소개가 먼저 제시된다.

3.1 시뮬레이션 환경 설정

설계한 QTMDP와 Panda, Festive 클라이언트의 성능 비교를 위하여 본 논문에서는 Wowza 미디어 엔진^[7]을 활용하여 대역폭과 그에 따른 비디오 청크의 품질을 나타내는 비트 레이트 정보를 확보할 수 있었다. 이 엔진을 사용함으로써 Panda와 Festive 플레이어의 대역폭 및 GUI를 통해 표현 되는 비트 레이트를

표 4. 항목별 성능평가
Table 4. Performance evaluation for each item

Item	QTMDP	Panda	Festive
Avg. bandwidth (Kbps)	13,955	11,020	11,020
Avg. chunk reception ratio (Kbps)	12,754	8,322	9,831
Reception efficiency of given bandwidth (%)	91.3	75.5	89.2
Avg. quality	QP27	QP32	QP27

통해 지속적으로 변화하는 환경에서 각 클라이언트가 어떻게 행동하는지 관찰할 수 있었다. 또한, 이러한 행동 양식을 시간에 따라 대역폭과 비트 레이트를 동시에 나타내어 Panda와 Festive 클라이언트가 어떻게 유동적으로 행동하는지 알 수 있었다.

반면 QTMDP는 Matlab을 활용하여 모든 품질의 비디오 청크를 요청할 수 있는 범위 내에서 임의로 대

역폭을 생성하였다. 또한, 이에 따라 QTMDP 클라이언트가 비교적 얼마나 우수하게, 적응적으로 대역폭 범위 내에서 요청할 비디오의 품질을 선택하는지 행동 방식을 관찰하였다. 세 클라이언트 모두 기본적으로 4K 품질의 Big Buck Bunny 비디오^[8]를 활용하였으며, 비디오를 구성하는 청크는 2초의 재생시간을 갖고 500초 동안 비디오를 재생하였다. 각 클라이언트들은 공통적으로 필요에 따라 압축률을 가변적으로 선택해 상황에 맞는 품질의 비디오 청크를 요청하였다. QTMDP와 Panda, Festive의 성능 비교는 그림 6, 7, 8을 통해 제시되며, 평균 대역폭, 평균 청크 수신율, 대역폭 대비 수신효율 및 평균 수신 품질에 대한 산술적 비교는 표 4와 같다.

3.2 시뮬레이션 결과

그림 6, 그림 7, 그림 8, 표 4를 통해 QTMDP와 Panda, Festive의 성능 평가에 대한 결과를 항목별로 살펴볼 수 있었다. Panda와 Festive는 Wowza 미디어 엔진을 통해 동일한 대역폭 상에서 동작하였으며, 평균적으로 QP32, QP27 품질의 비디오 청크를 재생할 수 있었음을 확인할 수 있다.

반면, 임의로 생성한 QTMDP 클라이언트의 평균 대역폭은 13,955(Kbps)로 HDS와 HLS보다 다소 높았으나, 수신한 평균 비디오 청크의 품질은 QP27로 더욱 우수하거나 비슷함을 알 수 있다. 또한, 네트워크 트래픽의 효율성으로 대역폭 대비 수신 효율을 비교하였을 때, 표 4와 같이 QTMDP의 경우 91.3%의 효율을, Panda와 Festive는 각각 75.5%, 89.2%의 효율을 보여 QTMDP가 더욱 우수함을 알 수 있다.

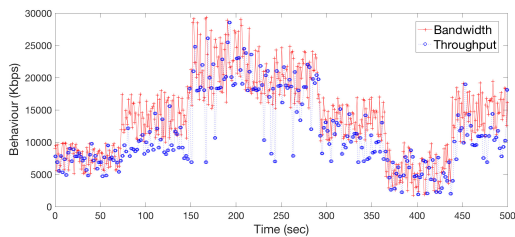


그림 6. QTMDP 미디어 요청 추이
Fig. 6. Media request tendency of QTMDP

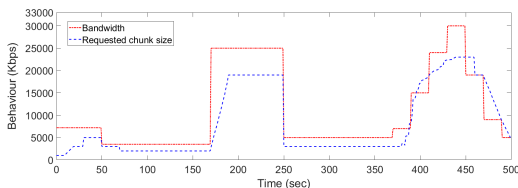


그림 7. Panda 미디어 요청 추이
Fig. 7. Media request tendency of Panda

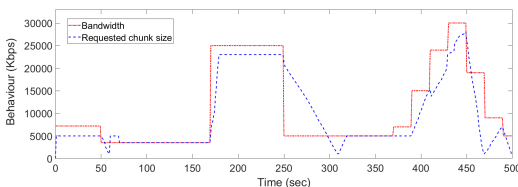


그림 8. Festive 미디어 요청 추이
Fig. 8. Media request tendency of Festive

IV. 결 론

본 논문은 HTTP 기반의 적응적 미디어 스트리밍 방식인 DASH 시스템을 일련의 미디어 요청이 반복되는 MDP로 재표현하여 기존의 DASH 시스템을 개선하도록 상태 전이 확률 조율 방식을 도입하였다. 이를 통해 점진적으로 클라이언트가 최적의 상태로 전이될 확률을 향상시킬 수 있었으며, 결과적으로 다른 DASH 클라이언트보다 우수한 성능을 보인 것을 확인할 수 있었다. 또한 제시한 수식들을 변수들에 적용하여 세부적 동작 제어가 가능하도록 하고 QTMDP 전체적으로 단순히 이용 가능한 비디오 청크의 품질 가운데 최상의 것을 무조건적으로 요청하는 것이 아니라, 빈번한 비디오 청크의 품질 변화를 지양하면서

최적의 품질을 요청하고 큐의 안정성을 동시에 고려하는 것을 지향하도록 설계하였다. 그 결과, QTMDP는 실험을 통해 Panda와 Festive보다 평균적으로 수신하는 비디오 청크의 품질이 더욱 우수할 뿐 아니라, 대역폭 대비 효율적인 미디어 요청을 수행하는 것을 확인할 수 있었다. 본 논문의 후속 연구는 초기 지연 시간을 다르게 설정하여 미디어 서버에 요청된 청크가 사용자에게 전달되어 디스플레이에 출력될 때 까지 소요된 시간을 측정하고, 사용자가 기존 DASH 미디어 플레이어보다 더욱 빠르게 미디어를 시청할 수 있도록 지연시간 단축 개선에 대한 연구를 진행할 것이다. 또한, 동일 화질 내에서 다른 압축률을 적용한 실험과는 별도로 서로 다른 화질의 비디오 샘플들 (SD, HD)을 통해 더욱 큰 dynamic range를 갖는 자료들을 실험에 적용하여 DASH 클라이언트의 행동에 영향을 미치는 추이를 실험할 것이다.

References

[1] H. Lee and D. Kim, "Combination of token bucket and AMP schemes to solve buffer underflow and overflow of video streaming in wireless communication," *J. KICS*, vol. 40, no. 7, pp. 1330-1338, Jul. 2015.

[2] M. Claeys, S. Latre, J. Famaey, and F. Turck, "Design and evaluation of a self-learning HTTP adaptive video streaming client," *IEEE Commun.s Lett.*, vol. 18, no. 4, pp. 716-719, Apr. 2014.

[3] M. Claeys, S. Latre, J. Famaey, T. Wu, W. V Leekwijck, and F. Turck, "Design and optimisation of a (FA) Q-learning-based HTTP adaptive streaming client," *Connection Sci.*, vol. 26, no. 1, pp. 25-43, Mar. 2014.

[4] A. Bokani, M. Hassan, and S. Kanhere, "Http-based adaptive streaming for mobile clients using markov decision process," *Packet Video Workshop (PV 2013)*, pp. 1-8, San Jose, USA, Dec. 2013.

[5] C. Kreuzberger, B. Rainer, H. Hellwagner, L. Toni, and P. Frossard, "A comparative study of dash representation sets using real user characteristics," *NOSSDAV '16*, no. 4, Klagenfurt, Austria, May 2016.

[6] D. Cicco, L. Caldaralo, V. Palmisano, and S.

Mascolo, "Elastic: a client-side controller for dynamic adaptive streaming over http (dash)," *Packet Video Workshop (PV 2013)*, pp. 1-8, San Jose, USA, Dec. 2013.

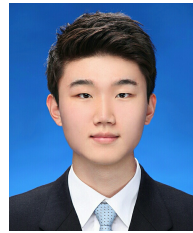
[7] Wowza Engine, *Wowza Streaming Engine*, Retrieved Apr. 20, 2018, from <https://www.wowza.com/products/streaming-engine>

[8] Big Buck Bunny, *Download Guide: Big Buck Bunny, Sunflower version*, Retrieved Apr. 20, 2018, from <http://bbb3d.renderfarming.net/download.html>

[9] Cisco, *Cisco Visual Networking Index 2016-2021 White Paper*, Retrieved Apr., 20, 2018, from <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>

[10] Cisco Visual Networking Index 2016-2021 White Paper, *Select Over-the-Top (OTT) Video Service Users, by Provider, 2015-2020 % of total OTT video service users*, Retrieved Apr. 20, 2018, from <http://www.telecompetitor.com/report-amazon-prime-subscription-growth-surgin-netflix-and-hulus-slip/>

권도현 (Dohyun Kwon)



2018년 2월 : 중앙대학교 컴퓨터공학부 졸업
 <관심분야> 차세대 무선 통신, 네트워크 최적화, 머신러닝 알고리즘

김중헌 (Joongheon Kim)



2004년 2월 : 고려대학교 컴퓨터학과 졸업
 2006년 2월 : 고려대학교 컴퓨터학과 석사
 2014년 8월 : University of Southern California Computer Science 박사
 <관심분야> 네트워크 최적화, 머신러닝 알고리즘