

Double DQN 기반의 끊김 없는 적응형 비디오 스트리밍 기법

최민제*, 정성욱*, 임경식*

A Seamless Adaptive Video Streaming Technique Based on the Double DQN

Minje Choi*, Sungwook Jung*, Kyungshik Lim*

요약

최근 스마트폰의 보급으로 HTTP Adaptive Streaming(HAS) 기반 비디오 스트리밍 서비스가 보편화되고 있다. 이에 자연스럽게 네트워크의 지연시간과 손실률의 변화 폭이 큰 모바일 환경에서도 끊김 없는 서비스를 제공하기 위한 다양한 연구가 진행되고 있다. HAS 기반 스트리밍 기술의 핵심은 동적으로 변화하는 네트워크 환경과 다양한 단말 상태에 적합한 비디오 품질을 선택하는 것인데 절차적 알고리즘만을 사용하는 경우 다양한 환경 변수를 고려하기에 어려움이 있다. 이를 보완하기 위하여 인공지능 기법을 도입한 다양한 방법이 시도되고 있는데 본 논문에서는 Double DQN(DDQN) 기반의 비디오 품질 선택 알고리즘을 제안한다. 이 방법은 기존 Deep Q-Network(DQN) 기반 연구가 비록 Quality of Experience(QoE)를 효율적으로 증가시키지만 학습모델을 생성하는데 긴 시간이 소요되며 학습과정 중 과대평가(overestimation) 문제가 발생한다는 단점을 해결한다. 또한 보다 적절한 비디오 품질을 선택하면서도 학습모델 생성 시간을 약 21% 정도 줄일 수 있음을 보인다.

Key Words : Adaptive Video Streaming, Q-learning, Reinforcement Learning, DASH

ABSTRACT

With the recent spread of smartphones, HTTP Adaptive Streaming (HAS)-based video streaming services are becoming more common. The key to HAS-based streaming technology is to choose the video quality that is suitable for dynamic changing network environments and various terminal conditions, making it difficult to consider various environmental variables if only procedural algorithms are used. To compensate for this, various methods of introducing artificial intelligence techniques are being tried, and this paper proposes a Double DQN(DDQN)-based video quality selection algorithm. This method addresses the disadvantages of existing Deep Q-Network(DQN)-based research that although it effectively increases Quality of Experience(QoE), it takes a long time to create a learning model and over-estimation problems occur during the learning process. It also shows that learning model creation time can be reduced by about 21% while choosing more appropriate video quality.

* First Author : Kyungpook National University, School of Computer Science and Engineering, alswp25@gmail.com, 학생회원

° Corresponding Author : Kyungpook National University, School of Computer Science and Engineering kslim@knu.ac.kr, 정회원

* Kyungpook National University, School of Computer Science and Engineering, squirrelsuj@gmail.com, 학생회원

논문번호 : 202006-125-B-RE, Received June 11, 2020; Revised July 12, 2020; Accepted July 28, 2020

I. 서론

최근 스마트폰의 보급으로 유튜브와 넷플릭스 등 비디오 스트리밍 서비스가 확대되고 있으며, 네트워크의 지연시간과 손실율의 변화폭이 큰 모바일 환경에 적합한 비디오 스트리밍 서비스에 대한 요구가 증가하고 있다.

모바일 환경(LTE)에서 네트워크 환경의 변화는 비디오 스트리밍 서비스에 심각한 문제를 초래한다. 동적 네트워크 환경에서 비디오 스트리밍 서비스를 제공하기 위해 많은 연구가 진행되었다. 적응형 비디오 스트리밍 기법 중 하나인 HTTP Adaptive Streaming(HAS)은 서비스 제공 업체가 가능한 최상의 품질의 비디오를 전송하고 적응을 통해 자원 활용과 QoE를 향상시킬 수 있도록 한다. 사용자 장치의 성능, 사용 가능한 비디오 품질 수준, 현재 네트워크 상태 및 현재 서버 상태를 고려할 수 있다. 그러나 매우 가변적인 모바일 환경에서 모든 네트워크 상태를 고려하여 적절한 비디오 품질을 선택하기에는 기존 알고리즘의 한계가 존재한다. 이 단점을 해결하기 위하여 Normal DQN(NDQN) 기반의 적응형 비디오 스트리밍 메커니즘이 연구되었다. 기존 방식보다 현재 네트워크 상태에 적합한 비디오 품질을 선택할 확률을 높임으로써 사용자의 QoE를 향상시켰다.^[1] 이 메커니즘은 시스템이 제 성능을 발휘하기 위해서 적절한 비디오 재생을 통해 학습모델을 생성한다. 이 과정은 많은 시간이 소요된다. 또한, NDQN은 Q값을 높게 평가하여 안정적인 학습을 저해하는 과대평가 문제가 발견되었다.^[2]

본 논문에서는 학습모델을 생성하는 시간을 줄이고 NDQN의 과대평가 문제를 해결하여 안정적으로 학습할 수 있는 적응형 비디오 스트리밍 시스템을 제안한다. 또한, 이 시스템은 적은 학습 횟수로도 제 성능을 발휘할 수 있고 네트워크 상태에 따른 비디오 품질 변화에도 더 효율적으로 대처하는 특징을 가진다.

II. 관련 연구

2.1 HTTP Adaptive Streaming(HAS)

HAS는 현재 네트워크 상태에 맞춰 비디오 품질을 조정함으로써 네트워크 상태의 변화에 따라 생길 수 있는 동영상 끊김을 개선하기 위해 채택된 기술이다. 고전적인 HTTP 비디오 스트리밍과 비교하여 HAS는 간헐적으로 발생하는 비디오 중단의 횟수를 줄이고 더 높은 대역폭을 활용하여 비디오 품질을 향상시키

는 장점을 갖는다. 이는 일반적으로 더 높은 QoE를 제공하는데, 이 기술의 핵심은 클라이언트 측에서 네트워크 및 클라이언트 단말기의 환경에 적합한 비디오 품질을 선택하는 것이다.

HAS 시스템은 비디오 세그먼트를 제공하는 서버와 비디오를 재생하는 클라이언트로 나뉜다. 서버는 비디오 콘텐츠를 해상도, 비트율(bitrate), 초당 프레임(frame) 수, 압축률 등 다양한 기준으로 품질을 나누고 세그먼트 단위로 인코딩된 비디오를 저장한다. 세그먼트는 네트워크 환경의 변화에 빠르게 반응할 수 있을 만큼 충분히 짧아야 한다. 또한 이에 대한 정보는 Media Presentation Description(MPD)에 저장된다. 클라이언트는 자신의 네트워크 환경에 적합한 비디오 품질을 결정하고 비디오 세그먼트를 서버에 요청한다. 적응 알고리즘은 QoE를 최대화하기 위해 적절한 품질을 선택해야 한다. 가장 일반적인 접근방식은 즉각적인 채널 대역폭을 추정하여 의사결정 기준으로 사용하는 것이다.^[3]

HAS의 비디오 품질 결정 알고리즘은 네트워크 상태를 처리량(throughput), 지연시간(latency), 버퍼 길이(buffer length), 프레임 드롭 횟수(dropped frames)를 가지고 네 가지 규칙을 통해 품질을 선택한다. 하지만 질차적 알고리즘으로 품질을 선택하기 때문에 네트워크의 모든 상태를 고려하여 품질을 선택할 수 없다. 이 단점을 해결하기 위해 NDQN 기반 비디오 품질 결정 메커니즘이 연구되었다.^[1]

2.2 Double Q-learning

강화학습(reinforcement learning)은 일련의 행동(action)을 하고 보상(reward)을 받는 과정을 통해, 미래에 수행할 행동에 대한 최적의 정책(policy)을 학습하는 방법이다.

Q-learning은 강화학습에서 학습을 위한 방법 중 하나이다. 강화학습의 과정은 샘플링을 통해 에피소드를 모으고, 이들을 기반으로 Q값과 정책값을 업데이트한다. 하지만 최근의 문제들은 상태 공간이 커지며 Q값이 수렴하는데 많은 에피소드를 필요로 한다. 상태공간이 큰 문제에서 Q값을 수렴하기 위해서 Q-learning에 deep neural network를 적용하는 Deep Q-network(DQN)이 제안되었다.^[5] Q-learning은 인기 있는 강화학습 알고리즘 중 하나이지만 특정 조건에 따라 행동(action)값을 과소평가된 값보다 과대평가를 선호하는 것으로 알려져 있다. Q값을 과대평가하게 되면 최적의 행동이 아닌 다른 행동을 취하도록 학습이 될 우려가 있으며, 이는 최적의 정책을 학습하는데

방해요소가 된다. NDQN을 학습 시키다보면 네트워크의 성능이 불안정하게 증가하는 것을 볼 수 있다. 이를 과대평가 문제라고 한다. 이 문제를 해결하기 위해 DDQN이 제안되었는데, 이 기법은 두 개의 네트워크를 사용하여 안정적으로 Q값을 학습할 수 있다. Q값이 가장 높은 행동을 선택하는 과정과 해당 행동의 값을 추정하는 과정으로 분리하여 서로 다른 네트워크를 사용하여 학습한다. 그리고 네트워크를 업데이트 하는 속도를 각각 다르게 하여 Q값이 커지지 않도록 한다. Hado van Hasselt는 DDQN을 사용하여 Atari 2600 도메인에서 NDQN의 과대평가 문제를 해결하여 더 나은 정책을 학습하는 것을 확인하였다.^[6]

2.3 NDQN 기반 적응형 비디오 스트리밍

NDQN 기반 적응형 비디오 스트리밍은 기존의 HTTP Adaptive Streaming(HAS)의 비디오 품질 선택 알고리즘을 개선하기 위해 제안되었다. HAS의 비디오 품질 선택 알고리즘은 절차식 알고리즘으로 구현이 되어있어 가변적인 모바일 네트워크의 상태를 모두 고려하여 비디오 품질을 선택하기에는 한계가 존재한다. 이 문제를 해결하기 위하여 NDQN을 비디오 품질 선택 알고리즘에 적용하여 비디오 품질을 선택할 수 있도록 하였다.

NDQN 기반 적응형 비디오 스트리밍 시스템에서 에이전트는 미디어 플레이어로부터 네트워크 상태를 받아와 가장 높은 보상을 받을 수 있는 비디오 품질을 예측한다. 비디오 재생이 종료될 때까지 특정 네트워크 상태에서 결정한 품질과 상태, 보상을 재생 버퍼에 저장한다. 그리고 재생 버퍼에 저장된 데이터를 가지고 학습하여 네트워크를 업데이트한다. 이후 업데이트된 네트워크를 가지고 비디오 품질을 예측하도록 한다.

기존의 HAS에서 NDQN 기반의 비디오 품질 선택 알고리즘으로 변경한 후 사용자의 QoE 증가에 큰 영향을 미치는 평균 비디오 품질과 재생 중단 횟수 등 여러 부분에서 성능을 증진시켰다.^[1]

III. Double DQN 기반의 적응형 비디오 스트리밍 시스템

3.1 학습 파라미터 정의

DDQN 기반의 비디오 품질 선택 메커니즘의 학습 모델을 생성하기 위해서 neural network의 input 값인 학습 파라미터를 정의해야 한다. 표 1은 학습 파라미터의 이름과 가질 수 있는 값들을 나타낸다. 학습 파라미터에는 네트워크 상태를 나타내는 처리량

표 1. 학습 파라미터
Table 1. Learning Parameters

Parameters	Value
Throughput	0~40 Mbps
Latency	10ms, 30ms, 50ms, 100ms
Buffer target	0 or 10
Buffer length	0~60
Prev Action(Quality)	0~9
Past Action(Quality)	0~9
Rebuffering	True or False

(throughput)과 지연시간(latency)과 단말기의 환경을 나타내는 버퍼길이(buffer length), 버퍼타겟(buffer target), 리버퍼링(rebuffering)이 있다.^[7] 그리고 최근 선택한 비디오 품질, 이전에 선택한 비디오 품질, 동영상 세그먼트 크기(segment size)를 학습 파라미터에 포함시켰다.

3.2 시스템 구조

그림 1은 DDQN 기반 비디오 품질 선택 메커니즘이 적용된 시스템의 구조를 나타낸다. 비디오 세그먼트를 전송하는 비디오 서버, 비디오 프레젠테이션 모듈이 포함된 클라이언트 그리고 학습 모델을 생성하

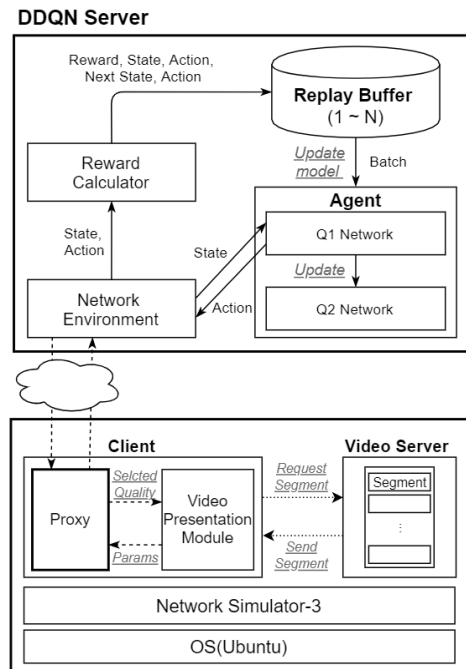


그림 1. DDQN 기반 비디오 품질 선택 메커니즘의 구조
Fig. 1. System Architecture

```

1 Initialize replay buffer  $M[state, action, reward, next\_state]$  to capacity  $N_M$ 
2 Initialize  $Q_\theta$  and the  $Q_{\theta'}$ ,  $\gamma \ll 1$ 
3 Obtain an initial  $s$  which is composed of network metrics
4 Repeat
5     Download the first segment with random action  $a_1$ 
6     Obtain network state  $s_1$ 
7     For  $i=2$  to  $i=N_{segment}$  do
8         Select random action  $a_i$ 
9         otherwise  $a_i = \text{argmax}(Q_\theta(s_{i-1}))$ 
10        Download  $i$ th segment with a corresponding quality  $a_i$ 
11        Update network state  $s_i$ 
12        Calculate reward  $R_{i-1}$  with state  $s_{k-1}$  and  $s_k$ 
13        Store replay information  $\langle s_{i-1}, a_{i-1}, R_{i-1}, s_k \rangle$  to  $M$ 
14    End For
15    For each update step do
16        Obtain a predefined number of randomly selected replay information
17        Sample  $\langle s_{i-1}, a_{i-1}, R_{i-1}, s_k \rangle$  from  $M$ 
18         $Q^*(s_t, a_t) = r_t + \gamma^* Q_\theta(s_{t+1}, \text{argmax}_{a'}(Q_{\theta'}(s_{t+1}, a')))$ 
19        Perform gradient descent step on  $(Q^*(s_t, a_t) - Q_\theta(s_t, a_t))^2$ 
20        Update  $Q_{\theta'}$  target network parameters
21    End For

```

그림 2. Double DQN 기반 메커니즘
Fig. 2. The mechanism based on Double DQN

고 비디오 품질을 선택하는 서버로 구성된다. 비디오 서버는 비디오 프레젠테이션 모듈이 비디오 세그먼트를 요청하면 해당 세그먼트를 클라이언트로 전송해주는 역할을 한다. 비디오 프레젠테이션 모듈은 현재 자신의 네트워크 상태를 추정하고 그 데이터를 네트워크 환경 모듈로 전송한다(그림 2. - 5~6행). 이때 전송하는 데이터는 표 1에 정의된 바와 같이 처리량, 지연시간, 버퍼타겟, 버퍼길이, 리버퍼링이다. Q1 Network는 네트워크 환경 모듈이 전송한 네트워크 상태를 가지고 비디오 품질(action)을 추정한다(그림 2. - 8~9행). 그 후 Q1 Network가 추정한 비디오 품질과 네트워크 상태 가지고 보상(reward)을 계산하고 재생버퍼(replay buffer)에 저장한다(그림 2. - 12~13행). 재생버퍼에 저장된 데이터는 랜덤으로 일정 수만큼 추출하여 Q1 Network의 업데이트에 사용된다. Q2 Network는 학습 과정 중 Q값을 구하기 위해 사용되는 네트워크이다(그림 2. - 16~20행). 본 모의실험에서는 batch size를 2000으로 설정하고 데이터를 쌓았다. 이 과정이 DDQN의 학습 과정이며 이를 training

step으로 정의하였다.

IV. 모의실험 및 평가

4.1 모의실험 환경

제안된 시스템은 비디오 스트리밍 서버, 클라이언트와 기존 HAS의 비디오 품질 선택 알고리즘 대신 DDQN, NDQN 기반의 비디오 품질 선택 메커니즘이 구현된 서버로 구성된다. 다양한 네트워크 환경에서 비디오 스트리밍을 하기 위해 비디오 서버와 클라이언트는 network simulator-3(ns-3)를 이용하여 실험하였다.¹⁸⁾ 실험에 사용된 ns-3 시뮬레이션 모델은 Jonathan Kua, Grenville Armitage과 Philip Branch가 개발하였고 실제 HAS의 비디오 품질 선택 로직이 포함되어 있다.¹⁹⁾ 기존 HAS의 성능을 측정하기 위해 이 모델을 그대로 사용하여 실험하였고, NDQN과 DDQN 기반 메커니즘을 구현하고 이 모델의 비디오 품질 선택 로직을 변경하여 실험하였다. NDQN과 DDQN이 구현된 서버는 오픈 소스 머신러닝 시스템

표 2. 품질별 세그먼트 평균 크기
Table 2. Segment Size by Quality

Quality	Average Segment Size(byte)
0	96,310
1	177,987
2	272,615
3	359,870
4	431,460
5	633,126
6	1,016,876
7	1,574,997
8	2,990,699
9	4,433,153

인 텐서플로우(tensorflow) 기반에서 실행할 수 있는 케라스(keras) 신경망 라이브러리를 이용하여 구현하였다. 비디오 서버와 클라이언트는 ns-3 시뮬레이션 환경에서 동작하고 클라이언트와 DDQN 서버는 소켓으로 연결하였다. 그리고 기존의 NDQN 기반 메커니즘과 DDQN 기반 메커니즘의 학습 성능을 비교하였고 비디오 품질, 리버퍼링을 가지고 3가지 품질 선택 기법의 성능을 비교하였다.

본 모의실험에 사용된 동영상은 약 640초 분량의 Big buck bunny 컴퓨터 단편 애니메이션을 사용하였다. 동영상은 170개의 세그먼트로 분할되어있고, 한 개의 세그먼트의 길이는 3.5~4초이다. 비디오 품질은 총 10단계(0~9품질)로 구분하고 품질의 숫자가 높을 수록 비디오의 화질이 좋으며 용량이 크다. 표 2는 비디오 품질별 세그먼트의 크기를 나타낸다. 학습 파라미터에는 세그먼트 크기가 포함되어 있어 1개의 동영상에 1개의 학습모델이 생성된다.

ns-3에서 구현된 비디오 서버와 클라이언트 간의 대역폭은 모바일 환경(LTE)의 평균 대역폭인 40Mbps로 설정하였다.^[10] 지연시간을 10ms, 30ms, 50ms, 100ms 그리고 손실률을 0.001, 0.01, 0.1로 다양하게 조합한 12개의 시뮬레이션 환경을 설정하여 학습모델을 생성하였다. 골고루 분포된 데이터를 재생 버퍼(replay buffer)에 쌓기 위해서 12번의 비디오 재생이 필요하다. 즉, 학습 모델 생성에서 한 번 학습 하는데 걸리는 최소 시간(1 training steps)은 비디오 재생 시간의 12배인 약 7680초(128분)가 소요된다. 모의실험에서 NDQN과 DDQN의 성능을 비교하기 위해 각각 학습 모델을 생성하여 학습시켰다. 충분히 학습된 모델을 생성하였고, 학습 횟수마다 테스트를 수행하여 로그를 쌓았다.

4.2 성능 평가

비디오 스트리밍 시스템에서 비디오 품질의 저하와 동영상 끊김 현상은 사용자의 QoE에 큰 영향을 준다. 따라서 본 모의실험에서는 평균 비디오 품질(average quality)과 리버퍼링 횟수(rebuffering)를 가지고 성능을 나타냈다.^[6]

4.2.1 대역폭 40Mbps, 손실률 0.001, 지연시간 10ms

그림 3은 학습 횟수에 따른 평균 품질 변화와 리버퍼링 횟수 변화를 나타낸다. 이 환경은 비교적 손실률이 매우 낮으며 지연시간도 적은 환경이므로 비디오 서버가 높은 품질의 비디오 세그먼트를 전송할 수 있다. 따라서 리버퍼링 횟수가 적으면서도 높은 비디오 품질을 선택할 수 있도록 학습이 이루어져야 한다. 학습 초기에는 NDQN과 DDQN 둘 다 네트워크 상태에 적합한 비디오 품질을 선택하지 못하고 있다. 하지만 일정 횟수 학습을 하면 점점 안정적으로 네트워크 상태에 적합한 비디오 품질을 선택하게 된다. NDQN의 학습 곡선인 붉은색 실선을 보면 학습이 어느 정도 이루어져 안정적인 비디오 품질을 선택하기 시작하는 학습 횟수는 100회이다. 100회를 학습하기 위해 필요한 비디오 재생 시간은 213시간이다. 그리고 DDQN의 학습 곡선인 푸른색 실선을 보면 안정적인 비디오 품질을 선택하기 시작하는 학습 횟수는 87회이다. 87

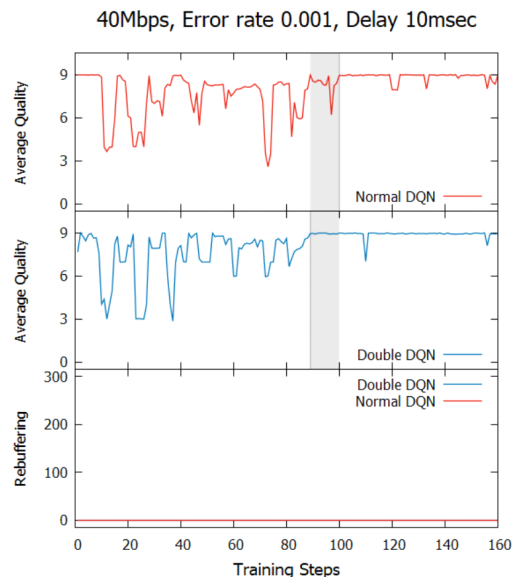


그림 3. 학습 횟수에 따른 평균 비디오 품질과 리버퍼링 횟수
Fig. 3. Average Quality and Rebuffering in 40Mbps, Error rate 0.001, Delay 10ms

회를 학습하기 위한 비디오 재생 시간은 185시간이다. 즉, 대역폭 40Mbps, 손실률 0.001, 지연시간 10ms 환경에서 DDQN 기반 메커니즘이 NDQN 기반 메커니즘 보다 13% 빠르게 학습하고, 비디오 재생 시간의 차이는 28시간이다.

4.2.2 대역폭 40Mbps, 손실률 0.01, 지연시간 10ms 이 환경은 비교적 높은 비디오 품질을 허용할 수 있다. 그림 4를 보면 NDQN은 학습 횟수 110회, DDQN은 87회부터 안정적인 성능을 나타냈다. 즉, 대역폭 40Mbps, 손실률 0.01, 지연시간 10ms 환경에서 DDQN 기반 메커니즘이 NDQN 기반 메커니즘 보다 21% 빠르게 학습하고, 차이는 49시간이다.

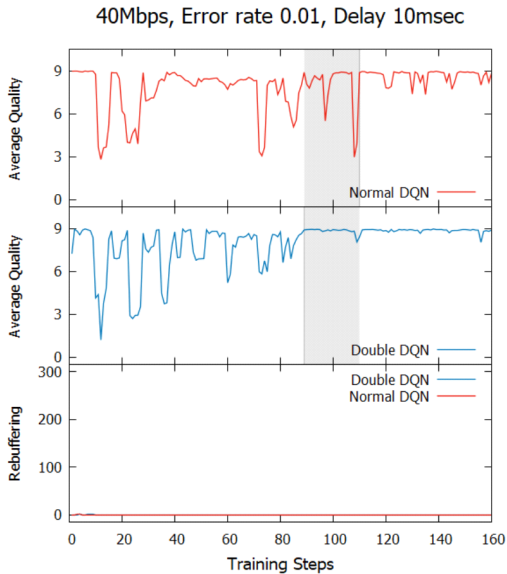


그림 4. 학습 횟수에 따른 평균 비디오 품질과 리버퍼링 횟수 Fig. 4. Average Quality and Rebuffering in 40Mbps, Error rate 0.01, Delay 10ms

4.2.3 대역폭 40Mbps, 손실률 0.1, 지연시간 30ms

이 환경은 손실률이 매우 높고 지연시간도 30ms로 비교적 네트워크 환경이 좋지 않은 경우이다. 따라서 낮은 비디오 품질을 선택할 수 있도록 학습이 되어야 한다. 그림 5를 보면 DDQN과 NDQN 둘 다 학습 횟수 약 10회부터 비디오 품질을 낮춰 리버퍼링 횟수가 줄어든다. 이 환경에서는 차이를 볼 수 없었다.

그림 3과 4같이 비교적 네트워크 환경이 좋은 상태에서 NDQN과 DDQN의 학습 횟수의 차이가 나는 것을 볼 수 있었고 그 이 외의 환경에서는 차이를 볼 수

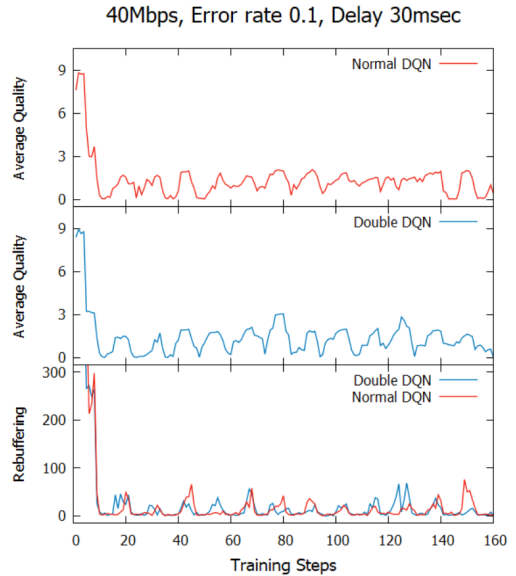


그림 5. 학습 횟수에 따른 평균 비디오 품질과 리버퍼링 횟수 Fig. 5. Average Quality and Rebuffering in 40Mbps, Error rate 0.1, Delay 30ms

없었다. 동적인 모바일 환경에서는 좋은 네트워크 환경에서 좋지 않은 환경까지 전체적으로 학습이 완료되어야 안정적인 성능을 낼 것이다. 따라서 전체적으로 안정적인 성능을 내기 위해서 DDQN은 최소 학습 횟수 87회, NDQN은 110회 정도 이루어져야 한다.

그림 6과 7은 학습 횟수가 90회일 때, HAS의 비디오 품질 선택 알고리즘(DASH), NDQN 기반 메커니즘 그리고 DDQN 기반 메커니즘 3가지의 네트워크 환경별 평균 비디오 품질과 리버퍼링 횟수를 나타낸다. 그림 6에서 손실률이 0.001일 때, DDQN이 다른 메커니즘보다 미세하게 높은 비디오 품질을 선택하는 것을 볼 수 있다. 더 높은 비디오 품질을 선택함에도 그림 7에서 리버퍼링 횟수가 0회이다. 그리고 손실률이 0.01일 때 지연시간이 낮을 경우에는 DDQN이 가장 좋은 비디오 품질을 선택한다. 손실률이 0.1일 때, 3가지 메커니즘이 각각 선택한 비디오 품질의 차이는 미세하게 보일 수 있으나 지연시간과 손실률이 매우 큰 환경이기 때문에 작은 차이에도 리버퍼링이 일어날 수 있다. DDQN이 다른 메커니즘에 비해 미세하게 낮은 품질을 선택하여 리버퍼링 횟수를 많이 낮출 수 있었다. 따라서 비디오를 시청하는 사용자의 QoE를 크게 향상시킬 수 있다.

본 모의실험에서 학습이 완료된 NDQN 기반의 메커니즘이 기존 HAS의 비디오 품질 선택 알고리즘보다 높은 비디오 품질을 선택하고 리버퍼링 횟수를 낮

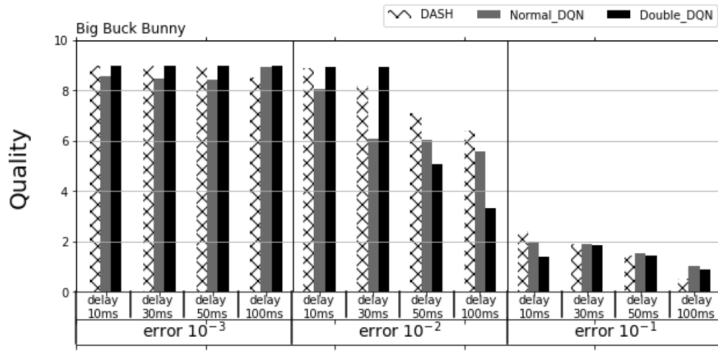


그림 6. 평균 비디오 품질
Fig. 6. Average Video Quality

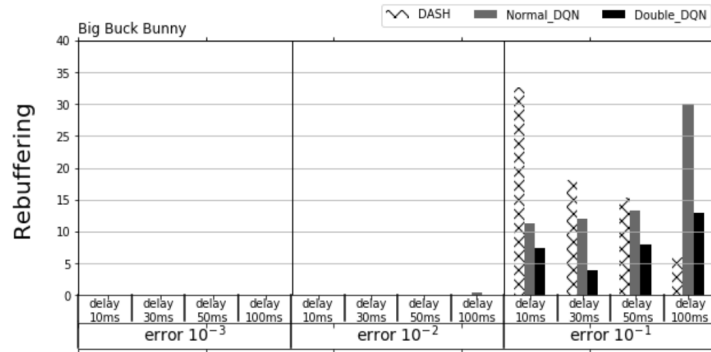


그림 7. 리버퍼링 횟수
Fig. 7. Rebuffering

출 수 있었다. 하지만 최소 213시간 비디오를 재생하며 학습모델을 생성해야함을 확인했다. 그리고 DDQN 기반의 메커니즘은 NDQN 메커니즘과 비교하여 비디오 품질을 선택하는 성능 면에서도 우수하며 학습 모델 생성 시간을 약 21% 정도 줄일 수 있었다.

V. 결 론

본 논문에서는 기존 NDQN 기반 비디오 품질 선택 메커니즘의 성능을 높일 수 있는 DDQN 기반 비디오 품질 선택 메커니즘을 제안하였다. NDQN은 학습 과정 중 Q값을 과대평가하여 안정적인 학습을 저해하는 과대평가 문제가 있었고 그 문제를 해결한 DDQN을 적용하였다. NDQN 기반 비디오 스트리밍 시스템에서 과대평가 문제가 자주 발생하지 않았으나 DDQN은 특정 환경에서 NDQN보다 안정적이고 빠른 학습을 할 수 있었다. 그 결과로 시스템이 안정적으로 성능을 나타내기까지의 시간을 약 21% 줄일 수 있었다.

NDQN은 한명의 사용자가 10분 길이의 비디오를 1408회 시청해야 그 비디오에 가장 적합한 학습모델이 생성되었다. DDQN에서는 학습모델 생성까지의 비디오 시청 횟수가 1110회로 횟수를 298회를 줄이면서 시스템의 학습 성능을 크게 개선할 수 있었다. 또한, 다수의 사용자가 서로 다른 환경에서 비디오를 시청할 경우 더욱 효율적이고 빠르게 학습모델을 생성할 수 있을 것이다. 그리고 네트워크 상태가 좋을 때 더 높은 비디오 품질을 선택하면서도 리버퍼링 횟수를 낮추었고 상태가 좋지 않을 때는 미세하게 더 낮은 품질을 선택하여 리버퍼링 횟수를 낮추었다. 이러한 결과는 사용자들이 비디오 재생 끊김과 화질 저하가 적은 비디오 스트리밍을 할 수 있다는 것을 증명한다. 추후 연구에서 이 시스템에서 학습 횟수를 더 줄일 방법을 모색할 것이고 수많은 클라이언트에서 스트리밍을 하며 각각 다른 환경에서 오는 데이터를 비동기적으로 학습하여 학습 성능을 높일 수 있는 연구를 진행할 것이다.

References

[1] C. Seong, S. Hong, and K. Lim, "An intelligent video streaming mechanism based on a deep Q-Network for QoE enhancement," *J. Korea Multimedia Soc.*, vol. 21, no. 2, pp. 188-198, 2018.

[2] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," in *Proc. Thirtieth AAAI Conf. Artificial Intell.*, pp. 2094-2100, 2016.

[3] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surv. and Tuts.*, vol. 17, no. 1, pp. 469-492, 2015.

[4] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learning*, vol. 8, no. 3-4, pp. 279-292, 1992.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529-533, 2015.

[6] H. van Hasselt, "Double Q-learning," *Advances in Neural Info. Process. Syst.*, vol. 23, pp. 2613-2621, 2010.

[7] P. Juluri, V. Tamarapalli, and D. Medhi, "QoE management in DASH systems using the segment aware rate adaptation algorithm," in *Proc. IEEE/IFIP Network Operations and Manag. Symp.*, pp. 129-136, 2016.

[8] T. R. Henderson, M. Lacage, and G. F. Riley, "Network simulations with the ns-3 simulator," in *Proc. SIGCOMM'08*, p. 527, Aug. 2008.

[9] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," *IEEE Commun. Surv. & Tuts.*, vol. 19, pp. 1842-1866, 2017.

[10] C. Seong, S. Hong, and K. Lim, "A context-based adaptive multimedia streaming scheme in IoT environments," *J. Korea Multimedia Soc.*, vol. 19, no. 7, pp. 1166-

1178, 2016.

[11] T. Stockhammer, "Dynamic adaptive streaming over HTTP - Standards and design principles," in *Proc. Second Annu. Assoc. for Computing Machinery Conf. Multimedia Syst.*, pp. 133-144, 2011.

최민제 (Minje Choi)



2019년 2월 : 대구대학교 컴퓨터정보공학부 (공학사)
 2019년 3월~현재 : 경북대학교 컴퓨터학부 석사과정
 <관심분야> 미디어 스트리밍, 모바일 컴퓨팅, 인공지능

[ORCID:0000-0003-4975-8069]

정성욱 (Sungwook Jung)



2017년 2월 : 안동대학교 멀티미디어공학과 (공학사)
 2017년 3월~현재 : 경북대학교 컴퓨터학부 석사과정
 <관심분야> 무선인터넷, 클라우드 컴퓨팅

[ORCID:0000-0001-6379-1290]

임경식 (Kyungshik Lim)



1982년 : 경북대학교 전자공학과 (공학사)
 1985년 : 한국과학기술원 전산학과 (공학석사)
 1994년 : University of Florida 전산학과 (공학박사)
 1985년~1998년 : 한국전자통신연구원 책임연구원, 실장

1998년 3월~현재 : 경북대학교 컴퓨터학부 교수
 <관심분야> 무선인터넷, 모바일 컴퓨팅, 사물인터넷
 [ORCID:0000-0003-3876-9571]