

일반논문 (Regular Paper)

방송공학회논문지 제22권 제5호, 2017년 9월 (JBE Vol. 22, No. 5, September 2017)

<https://doi.org/10.5909/JBE.2017.22.5.618>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

성능 향상을 위한 퍼지 논리 기반 DASH 알고리즘의 수정

김 현 준^{a)}, 손 예 슬^{a)}, 김 준 태^{a)†}

A Modification of The Fuzzy Logic Based DASH Adaptation Algorithm for Performance Improvement

Hyun-Jun Kim^{a)}, Ye-Seul Son^{a)}, and Joon-Tae Kim^{a)†}

요 약

본 논문에서는 시변 네트워크 상황에서 끊김 없는 미디어 서비스를 제공 할 수 있는 퍼지 논리 기반 DASH 적용 알고리즘(FDASH)의 수정을 제안한다. 제안하는 알고리즘은 퍼지 논리 제어부(FLC : Fuzzy Logic Controller)의 수정을 통하여 다음 요청 할 세그먼트의 비트율에 대해 최적의 판단을 하도록 하고, 세그먼트 비트율 필터링 모듈(SBFM : Segment Bit-rate Filtering Module)을 적용하여 비디오 화질의 변화 횟수를 줄인다. 또한, 스트리밍 서비스를 시작 할 때 사용자들이 일정시간 저화질의 비디오를 시청해야 하는 상황을 막는 시작 메커니즘(Start Mechanism)과 버퍼의 오버플로우를 방지하는 대기 메커니즘(Sleeping Mechanism)을 포함한다. 최종적으로 제안된 알고리즘이 FDASH에 비해 좋은 성능을 가짐을 NS-3를 이용한 모의실험을 통해 검증한다. 모의실험 결과, 제안된 방식이 FDASH에 비해 제한된 버퍼크기 상황 하에서도 버퍼 언더플로우/오버플로우가 발생하지 않음을 확인하였다. 또한 점대점(Point-to-Point) 환경과 Wi-Fi환경에서 거의 동일 화질 성능을 보이면서도 비디오 화질 변화 횟수를 50% 이상 줄일 수 있음을 확인하였다.

Abstract

In this paper, we propose a modification of fuzzy logic based DASH adaptation algorithm(FDASH) for seamless media service in time-varying network conditions. The proposed algorithm selects more appropriate bit-rate for the next segment by the modification of the Fuzzy Logic Controller(FLC) and reduces the number of video bit-rate changes by applying Segment Bit-rate Filtering Module(SBFM). Also, we apply the Start Mechanism for clients not to watch the low quality videos in the very beginning stage of streaming service and add the Sleeping Mechanism to avoid any buffer overflow expected. Ultimately, we verified by using NS-3 Network Simulator that the proposed method shows better performance compared to FDASH. According to the experimental results, there is no buffer underflow/overflow within the limited buffer size, which is not guaranteed in FDASH on the other hand. Also, we confirmed that mFDASH has almost the same level of average video quality against FDASH and reduces about 50% of number of video bit-rate changes compared to FDASH in Point-to-Point network and Wi-Fi network.

Keyword : DASH, Fuzzy Logic, Segment Bit-rate Filtering Module(SBFM), Adaptive streaming

Copyright © 2017 Korean Institute of Broadcast and Media Engineers. All rights reserved.

"This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered."

I. 서 론

DASH^{[1][2][3]}는 시변 네트워크 상황과 서로 다른 단말기의 특성을 고려하여 끊김 없는 미디어 서비스를 제공하기 위한 적응적 스트리밍 표준이며 HTTP/TCP를 기반으로 하여 기존 RTP/UDP 기반 스트리밍 프로토콜의 방화벽에 차단 될 수 있다는 문제점을 해결 할 수 있고, 이미 존재하는 HTTP 서버를 사용 할 수 있어 비용 측면에서도 효율적이다. DASH에서는 미디어 콘텐츠를 여러 가지 비트율 버전으로 인코딩 한 뒤 각각의 버전을 수 초에서 수 십초 단위의 세그먼트로 분할하여 HTTP 서버에 저장하고, DASH 서비스 사용자는 시변 네트워크 상황에 적응적으로 알맞은 버전의 세그먼트를 HTTP 서버에 요청해서 서비스를 받는다.

최근까지 DASH를 이용한 다양한 서비스에 대한 많은 연구가 진행 되었으며^{[4][5]}, 이와 더불어 시간에 따라 변하는 네트워크 환경과 단말기의 특성을 고려하여 최적의 사용자 체감 품질(QoE)을 제공하기 위한 적응적 스트리밍 기법에 관한 많은 연구가 진행 되었다. [6]에서 제안하는 mDASH는 Markov Theory^{[7][8]}를 기반으로 하여 적절한 세그먼트 비트율을 결정하는 기술이다. [6]의 저자는 DASH 사용자들에게 현재와 미래의 버퍼 상황, 평균 세그먼트 비트율, 비트율 변화 빈도에 대한 보상 값을 통해 최적의 QoE를 제공하였다. [9]에서는 Scalable Video Coding(SVC)^[10]기법과 물리층의 채널 정보, 응용층의 비디오 정보, 버퍼정보를 이용하여 보다 더 높은 QoE를 제공하는 알고리즘을 소개하였다. Agile and Smooth Video Adaptation Algorithm (SVAA)^[11]은 현재 버퍼 점유율, 버퍼의 트랜드, 그리고

History-Based TCP Throughput Estimation(HBTTE)기법^[12]을 이용하여 측정한 사용가능 대역폭을 통해 다음 세그먼트의 비트율을 결정하는 기술이다.

최근에는 퍼지 논리^{[13][14]}를 이용한 적응형 스트리밍 기법인 FDASH^[15]가 소개 되었다. 그럼 1은 FDASH 알고리즘의 구조이다. 먼저 DASH 사용자는 서버로부터 세그먼트를 받아 Playback 버퍼에 저장을 한 뒤, 버퍼 점유율과 버퍼 점유율의 차이의 정보를 FLC에 전달한다. FLC는 두 가지 정보를 입력으로 받아서 다음 세그먼트의 비트율을 계산하고, FS (FDASH Scheme)에서 FLC가 계산한 뒤, 세그먼트의 비트율과 스루풋 측정값을 이용해 FLC가 계산한 비트율로 다음 세그먼트를 요청할지, 또는 이전 세그먼트의 비트율로 다음 세그먼트를 요청할지를 결정한다. 결정된 비트율은 HTTP Request를 통해 DASH 서버로 전달된다. FDASH는 미디어 콘텐츠의 품질, 미디어 재생의 끊김 현상에 대해서는 다른 알고리즘에 비해 좋은 특성을 보였지만 버퍼의 오버플로우에 대한 고려를 하지 않으며 비디오 화질 변화가 많이 발생하는 문제점을 보였다. QoE는 비디오 화질 변화 빈도, 버퍼 언더플로우/오버플로우 등의 요인에 영향을 받기 때문에 버퍼 오버플로우의 발생과 비디오 화질 변화가 빈번히 발생하는 것은 비디오 품질에 부정적인 영향을 미친다^{[16][17]}. 따라서 FDASH기법을 이용해 DASH 서비스 사용자에게 최적의 QoE를 제공하기 위해서는 비디오 화질 변화 횟수를 줄여야 하고 버퍼 오버플로우는 발생하지 않도록 해야 한다.

본 논문에서는 시변 네트워크 상황에서 끊김 없는 미디어 서비스를 제공 할 수 있는 FDASH의 수정을 제안한다. 제안된 알고리즘(mFDASH)에서는 FLC^{[18][19]}가 최적의 출력을 낼 수 있도록 수정하고, FLC의 출력이 적합한지를 판단하는 SBFM을 적용하여 비디오 화질 변화 횟수를 줄인다. 또한 mFDASH에서는 FDASH에서 사용하는 스루풋 측정 기법인 FHTTE보다 더 정확하면서 평활도(smoothness)는 적절히 유지하는 스루풋 측정 기법을 사용하여 SBFM가 더 정확하게 동작할 수 있도록 한다. 마지막으로, 시작 메커니즘을 적용하여 스트리밍 서비스를 시작할 때 버퍼가 언더플로우가 나지 않게 하면서 기존보다 더 높은 화질의 비디오 서비스를 제공하고, 대기 메커니즘을 적용하여 버퍼 오버플로우가 발생하는 것을 방지한다.

본 논문에서는 NS-3(Network Simulator)^[20]를 이용한 네

a) 건국대학교 전자·정보통신공학과(Department of Electronic Engineering, Konkuk University)

‡ Corresponding Author : 김준태(Joon-Tae Kim)
E-mail: jtkim@konkuk.ac.kr
Tel: +82-31-670-6734

ORCID: <http://orcid.org/0000-0001-6953-5482>

※ 본 연구는 미래창조과학부 및 정보통신기술 연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R0101-16-0189, 네트워크가 결합된 매체 독립형 차세대 응합방송 시스템 및 모니터링 시스템 개발]

※ 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. R0101-16-0189, 네트워크가 결합된 매체 독립형 차세대 응합방송 시스템 및 모니터링 시스템 개발)

Manuscript received June 26, 2017; Revised August 25, 2017; Accepted August 25, 2017.

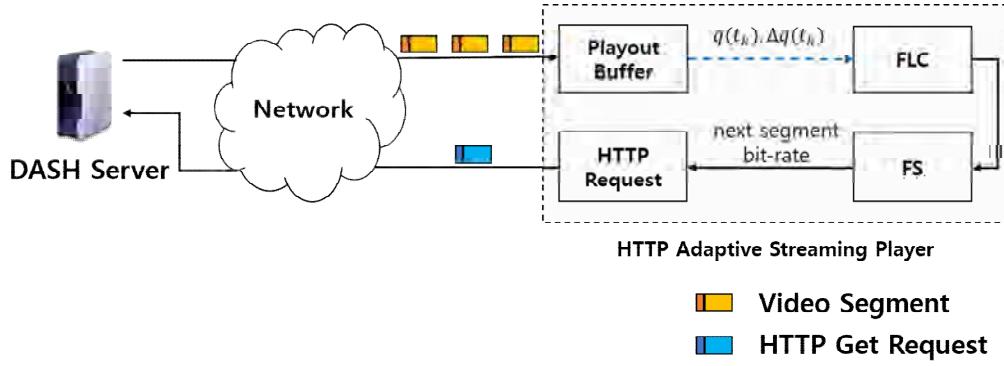


그림 1. FDASH 알고리즘 구조
Fig. 1. Architecture of FDASH Algorithm

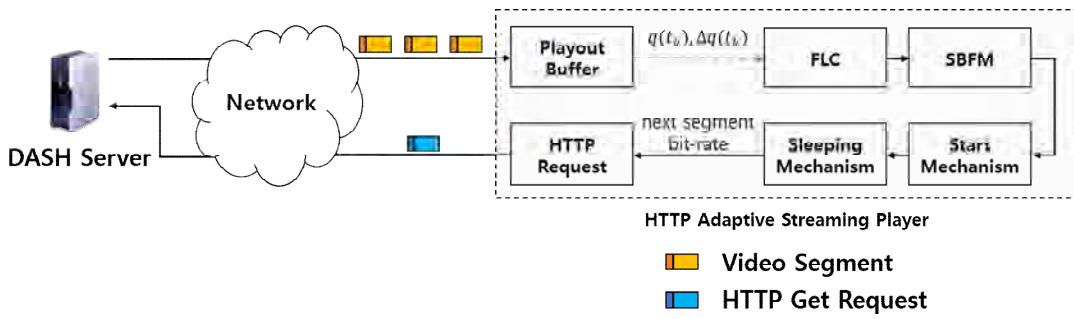


그림 2. mFDASH 알고리즘 구조
Fig. 2. Architecture of mFDASH Algorithm

트워크 모의실험 결과를 통해 제안된 방식이 FDASH 방식에 비하여 제한된 버퍼크기 상황 하에서도 버퍼 오버플로우 현상과 버퍼 언더플로우로 인한 재생 끊김 현상이 발생하지 않음을 보인다. 또한 점대점 상황에서는 거의 동일 화질 성능을 보이면서도 비디오 화질 변화 횟수를 50% 이상 줄일 수 있음과 일반 Wi-Fi 환경에서도 평균적으로 거의 동일한 화질성능을 보이면서 비디오 화질 변화 횟수는 53.1% 정도 줄일 수 있음을 보여준다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 FLC에 대하여 기술하고, Ⅲ장에서는 mFDASH 알고리즘에 대하여 설명한다. Ⅳ장에서는 모의실험 환경설정 및 결과에 대해 기술하고, 마지막으로 Ⅴ장에서 본 논문의 결론을 맺는다.

II. 퍼지 논리 제어부

mFDASH의 기본적인 구조는 그림2와 같다. 먼저 DASH

사용자는 서버로부터 세그먼트를 받아 Playback 버퍼에 저장을 한 뒤, 버퍼 점유율과 버퍼 점유율의 차이의 정보를 FLC에 전달한다. FLC는 두 가지 정보를 입력으로 받아서 다음 세그먼트의 비트율을 계산하고, SBFM에서 다음 세그먼트의 비트율을 현재 세그먼트의 비트율로 유지 할 것인지, 아니면 FLC가 계산한 세그먼트 비트율로 바꿀 것인지를 판단한다. 이때, 시작 메커니즘이 동작하고 있으면 SBFM에서 판단한 비트율은 무시하고 시작 메커니즘에서 결정한 비트율을 다음 세그먼트의 비트율로 결정한다. 마지막으로, 대기 메커니즘을 통해 다음 세그먼트 요청까지 특정 시간만큼 대기하면서 버퍼 오버플로우를 방지한다. 본 장에서는 mFDASH 기법의 구성요소중 하나인 FLC에 대하여 기술한다. FLC는 퍼지 논리를 이용한 제어 시스템으로 그림 3과 같이 퍼지화(Fuzzification) 인터페이스, 정보 베이스(Knowledge Base), 결정 논리(Decision-Making Logic), 역 퍼지화(De-Fuzzification)인터페이스 단계로 구성된다.

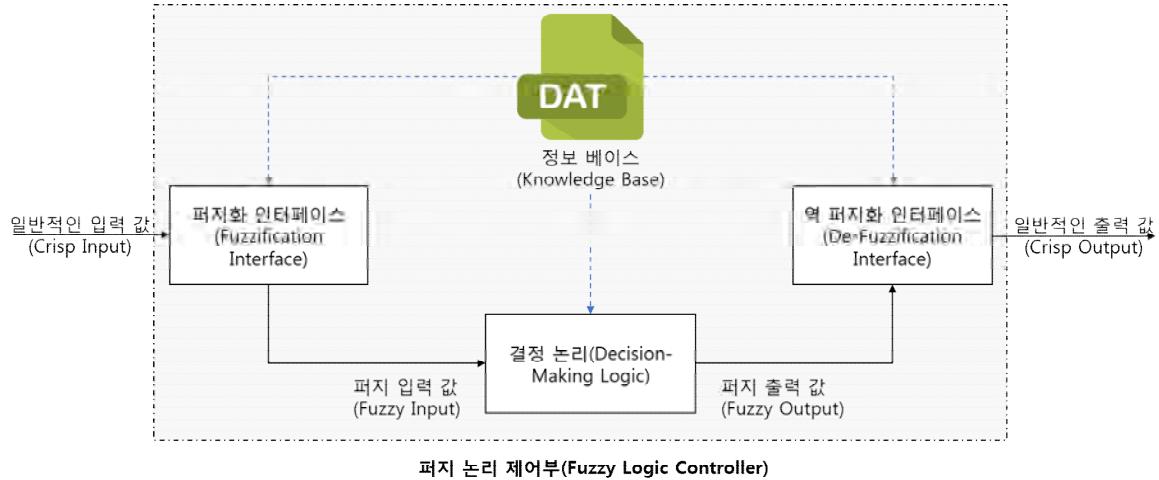


그림 3. FLC 기본 구조
 Fig. 3. Fundamental Architecture of FLC

1) 퍼지화 인터페이스

퍼지화 인터페이스에서는 일반적인 입력 값(Crisp Input)을 받고 입력 값을 FLC의 논의영역으로 대응시키며, 대응된 값과 정의된 소속 함수에 의해 각 언어학적 값인 퍼지 입력 값(Fuzzy Input)이 결정된다.

2) 정보 베이스

정보 베이스는 데이터 기반(Data Base), 규칙 기반(Rule Base) 두 가지로 구성된다. 데이터 기반에서는 FLC에서 사용되는 퍼지 값과 퍼지 규칙을 다루기 위한 정의들을 제공한다. 규칙기반의 퍼지 규칙(fuzzy rules)은 간단한 IF-THEN 규칙으로 언어 변수로 구성이 되며, 정해진 제어 목표에 의해 퍼지 출력 값을 도출하는데 사용된다.

3) 결정 논리

결정 논리 단계에서는 퍼지화 인터페이스에서 얻은 퍼지 입력 값들과 퍼지 규칙을 이용해 퍼지 출력 값을 도출한다.

4) 역 퍼지화 인터페이스

역 퍼지화 인터페이스에서는 퍼지 출력 값을 논의영역으로 대응시킨 다음, 일반적인 출력 값(Crisp Output)을 도출해낸다. 역 퍼지화 인터페이스 기법에는 5가지의 방법이 존재한다. i) 중심 기법(The centroid), ii) 이등분선 기법(The bisector), iii) 최댓값 중심 기법(The middle of max-

imum), iv) 최댓값 최대화 기법(The largest of maximum), v) 최댓값 최소화 기법(The smallest of maximum)^[21]. mFDASH에서 사용하는 FLC에서는 간단하면서도 보편적으로 사용되며, FDASH의 FLC에서 사용한 중심 기법을 사용한다.

mFDASH에서 사용하는 FLC는 베퍼 점유율, 베퍼 점유율의 차이를 입력으로 받고, f 를 출력한다. 이때, f 는 다음 세그먼트의 비트율을 스루풋 측정값보다 얼마나 증가/감소 시킬 것인지를 나타내는 값이다. 예를 들어, f 가 2면, mFDASH에서 사용하는 스루풋 측정 방법으로 측정한 스루풋 측정값의 2배가 다음 세그먼트의 비트율이 된다.

그림 4는 입력, 출력의 소속 함수들을 나타낸다. 그림 4(a)는 베퍼 점유율에 대한 소속 함수들을 나타내며, 이 소속 함수들의 값은 베퍼 점유율이 해당 소속 함수를 나타내는 언어 변수에 어느 정도 속하는지를 나타낸다. 베퍼 점유율은 가장 최근에 요청한 k 번째 세그먼트가 완전히 수신 되었을 때 베퍼가 가지고 있는 데이터를 초 단위로 나타낸 값으로, $q(t_k)$ 로 표기한다. $q(t_k)$ 의 상태를 나타내기 위해 3개의 언어 변수 [Short(S), Close(C), Long(L)]가 사용되며, Short는 $q(t_k)$ 가 부족한 경우, Close는 $q(t_k)$ 가 이상적인 베퍼 점유율에 가까운 경우, 그리고 Long은 $q(t_k)$ 가 베퍼의 최대 용량에 가까운 경우를 뜻한다. mFDASH에서는 FDASH의 $q(t_k)$ 소속 함수 범위를 다음과 같이 설정하였다.

표 1. FDASH FLC와 mFDASH FLC의 차이점

Table 1. Difference Between FDASH FLC and mFDASH FLC

	수정 여부	수정 이유 (Reason of Modification)
입력 값(Input Variable), 출력 값(Output Variable)	×	FDASH의 FLC에서 사용한 입력 값, 출력 값을 그대로 사용하였다. (We use the same input variables and the same output variable of FLC of FDASH.)
입력 소속 함수(Input Membership Function)	○	입력 소속 함수의 범위의 수정을 통해 적절한 버퍼 상태에서 세그먼트 비트율의 스위칭이 발생하도록 하였다. (We let the segment bit-rate switched at the appropriate buffer status by the modification of the ranges of input membership functions.)
출력 소속 함수(Output Membership Function)	○	출력 언어 변수를 5개에서 3개로 줄여 불필요한 비트율 스위칭이 일어나지 않도록 하였다. 또한, 출력 소속 함수의 범위를 수정하여 적절한 출력 값(f)을 선택하도록 하였다. (We reduce the number of output linguistic variables from 5 to 3 to prevent unnecessary bit-rate switching. Also, we modified the ranges of the output membership functions to obtain optimal output variable(f)).
퍼지 규칙(Fuzzy Rule)	○	출력 언어 변수, 출력 소속 함수가 변경되면서 퍼지 규칙도 변경 되었다. (The Fuzzy Rules are also modified as the output linguistic variables and output membership functions are modified.)
역 퍼지화 기법(De-Fuzzification Method)	×	FDASH에서 사용한 The Centroid 방식을 사용하였다. (We use the The Centroid method which is also used in FDASH.)

1) 이상적인 $q(t_k)$ 값을 T 라고 정의할 때, FDASH에서 Short가 1이 되는 $q(t_k)$ 의 범위는 $0 \sim \frac{2T}{3}$ 가 된다. 이 범위로 인해 버퍼가 최대한으로 활용되기 전에 퍼지 규칙에 의해 다음 세그먼트의 비트율이 낮아 질 수 있다. 따라서 mFDASH에서는 해당 범위를 $0 \sim \frac{T}{3}$ 으로 수정하여 $q(t_k)$ 을 보다 효율적으로 사용 할 수 있도록 하였다.

2) 또한, Long이 1이 되는 $q(t_k)$ 의 범위를 $4T$ 이상에서 $2T$ 이상으로 수정하면서 버퍼 오버플로우가 발생하기 전에 세그먼트 비트율이 증가하게 하였다.

그림 4(b)는 버퍼 점유율의 차이에 대한 소속 함수들을 나타내며, 버퍼 점유율의 차이는 k 번째 세그먼트를 완전히 수신 했을 때의 버퍼 점유율과 k-1 번째 세그먼트를 완전히 수신 했을 때 버퍼 점유율의 차이로, $\Delta q(t_k) = q(t_k) - q(t_{k-1})$ 로 표기한다. 이때, $q(t_k)$ 는 버퍼에 들어있는 미디어 데이터의 실제 크기가 아니고, 미디어 데이터의 재생 시간이기 때문에, $\Delta q(t_k)$ 또한 초 단위가 된다. $q(t_k)$ 와 마찬가지로 $\Delta q(t_k)$ 의 상태를 나타내기 위해 3 가지 언어 변수 [Falling(F), Steady(S), Rising(R)]가 사용된다. Falling은 $q(t_k)$ 가 감소하는 경우, Steady는 $q(t_k)$ 가 거의 변하지 않는 경우, 그리고 Rising은 $q(t_k)$ 가 증가하는 경우를 뜻한다. 두 번째 입력 $\Delta q(t_k)$ 에 대해서는 다음과 같은 수정을 적용하였다.

1) 모의실험을 진행하면서 분석한 결과, 네트워크 상황이 아무리 혼잡하여도 $\Delta q(t_k)$ 값은 $-\frac{T}{3}$ 초에서 세그먼트 재생 시간(τ) 사이를 벗어나지 않음을 확인 하였

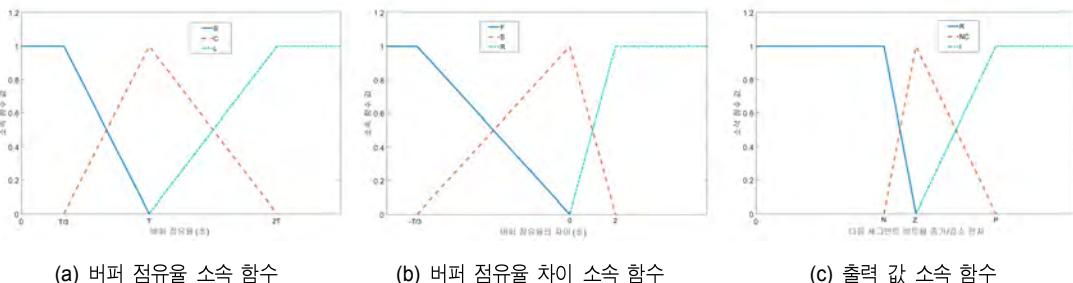


그림 4. 입력 출력 소속 함수
Fig. 4. Membership Functions of Inputs and Output

- 다. 따라서 Falling이 1이 되는 $\Delta q(t_k)$ 의 범위를 $-\frac{2T}{3}$ 이하에서 $-\frac{T}{3}$ 이하로 수정하여 현재 버퍼 점유율의 변화 상태가 잘 반영되도록 하였다.
- 2) 또한, 버퍼 점유율의 차이는 네트워크 상황이 아무리 좋아도 τ 보다 짧을 수밖에 없으므로 Rising이 1이 되는 $\Delta q(t_k)$ 의 범위를 $4T$ 이상에서 τ 이상으로 수정하여 네트워크 상황이 좋을 때 $\Delta q(t_k)$ 가 f 에 적절히 반영되도록 하였다.

출력을 위한 언어 변수에는 [Reduce(R), No change(NC), Increase(I)]가 있다. 그림 4(c)는 f 의 범위를 N, Z, P로 나누어 R, NC, I의 소속 함수의 구간을 나눈 그림이다. N은 세그먼트의 비트율을 줄이는 인자로 $N \in [0,1]$ 이고, Z는 세그먼트 비트율을 유지시키는 인자로 1이다. 마지막으로 P는 세그먼트 비트율을 높이는 인자로 $P \in [1, \infty]$ 이다.

본 알고리즘에서 사용되는 퍼지 규칙은 아래와 같이 정의 하였다.

Fuzzy Rules

- 1: if short and falling then R : $r_1 = \min(S, F)$
- 2: if close and falling then R : $r_2 = \min(C, F)$
- 3: if short and steady then R : $r_3 = \min(S, S)$
- 4: if long and falling then NC : $r_4 = \min(L, F)$
- 5: if close and steady then NC : $r_5 = \min(C, S)$
- 6: if short and rising then NC : $r_6 = \min(S, R)$
- 7: if long and steady then I : $r_7 = \min(L, S)$
- 8: if close and rising then I : $r_8 = \min(C, R)$
- 9: if long and rising then I : $r_9 = \min(L, R)$

앞에서 설명한 규칙에 의해 출력 소속 함수 R, NC, I는 식 (1), (2), (3)으로, 출력 값 f 는 The Centroid 방식을 사용하여 식(4)로 나타낼 수 있다.

$$R = \sqrt{r_1^2 + r_2^2 + r_3^2} \quad (1)$$

$$NC = \sqrt{r_4^2 + r_5^2 + r_6^2} \quad (2)$$

$$I = \sqrt{r_7^2 + r_8^2 + r_9^2} \quad (3)$$

$$f = \frac{N \times R + Z \times NC + P \times I}{R + NC + I} \quad (4)$$

지금까지 mFDASH FLC와 FDASH FLC의 중요한 차이점에 대해 기술 하였고, 전체적인 FLC의 수정 사항은 표 1에 정리 되어있다.

III. mFDASH 알고리즘

본 장에서는 그림 2의 구성 요소 중 앞서 설명한 FLC를 제외한 대역폭 측정 기법(HBTTE), SBFM, 시작 메커니즘, 대기 메커니즘에 대하여 기술한다.

1. 대역폭 측정 기법

FDASH에서는 과거 S_{window} 초안에 포함되는 스루풋 값들의 평균값을 대역폭 측정값(스루풋 측정값)으로 사용하여 FLC가 계산한 다음 세그먼트 비트율($v(k+1)$)이 현재 네트워크 상황에 적합한지를 판단한다. FHBTE 기법은 측정한 대역폭 값이 평활하다는 장점이 있지만 대역폭의 단계 변화(level shift)와 이상점(outlier)을 탐지하지 못해서 정확한 대역폭 예측이 불가능 하다는 단점이 있다. 따라서 mFDASH는 단계 변화, 이상점을 감지한 뒤 대역폭을 측정하는 HBTTE 기법을 사용한다.

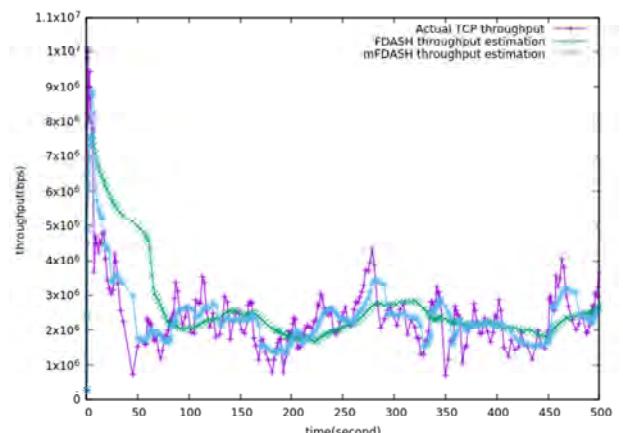


그림 5. 스루풋 측정 기법 비교
 Fig. 5. Comparison of Throughput Estimation

그림 5은 Wi-Fi환경에서 FHBTE 기법과 HBTTE 기법의 차이점을 보여주고 있다. FHBTE 기법은 초반 0~70초 사이에서 단계 변화를 감지하지 못하여 실제 TCP 스루풋을 잘 따라가지 못하는 것을 확인 할 수 있다. 반면에 HBTTE 기법은 단계 변화뿐만 아니라, 일정 범위에서 벗어난 스루풋 값인 이상점을 감지하여 무시하기 때문에 HBTTE 기법이 실제 스루풋을 더 잘 따라가면서, 적절한 평활도를 가지는 것을 확인 할 수 있다.

2. 세그먼트 비트율 필터링 모듈(Segment Bit-rate Filtering Module)

FLC의 출력 값 f 는 $q(t_k)$ 와 $\Delta q(t_k)$ 에 의해 결정된다. 따라서 비디오 화질이 변하게 되면, 버퍼 점유율과 버퍼 점유율이 순간적으로 변하게 되고, 그 때문에 다음 f 값은 다시 바뀐 입력에 대한 값을 출력하게 된다. 이러한 현상 때문에 f 만으로 다음 세그먼트의 비트율을 결정하게 되면 세그먼트 비트율이 오실레이션(Oscillation) 현상을 겪게 된다. 따라서 mFDASH에서는 SBFM을 추가적으로 적용하여 세그먼트 비트율의 오실레이션 현상을 없애면서 불필요한 비트율 변화 횟수를 줄이고, 현재 네트워크와 버퍼 상황에 최고로 적합한 세그먼트를 서버에 요청한다.

알고리즘1은 SBFM 알고리즘의 의사코드이다. q_{high} 는 버퍼의 최대 점유율(버퍼 사이즈)이고, q_{low} , q_{min} 은 세그먼트 비트율의 급격한 저하와 버퍼의 언더플로우를 방지하기 위한 2개의 한계점(threshold)이다. 일단 FLC의 출력 값인 f 는 HBTTE 기법으로 측정한 값인 T_k 와 곱해져서 $v(k+1)$ 이 된다. $Q(\cdot)$ 는 양자화 함수로, 선택 가능한 세그먼트 비트율 중 \hat{v} 보다 작은 가장 큰 비트율이 $v(k+1)$ 이 된다.

$v(k+1)$ 이 $v(k)$ 보다 큰 경우에는 T_k 와 $q(t_k)$ 를 고려하여 버퍼의 오버플로우와 세그먼트 비트율의 오실레이션 현상을 없앤다. 즉, T_k 가 $v(k+1)$ 보다 일정수준 이상 높거나, $q(t_k)$ 가 q_{high} 이상이면 다음 세그먼트의 비트율을 바꾸고, 그 외에는 비트율을 바꾸지 않는다. 반대로 $v(k+1)$ 이 $v(k)$ 보다 작은 경우에도 T_k 와 $q(t_k)$ 를 고려하여 버퍼 언더플로우와 세그먼트 비트율의 오실레이션 현상을 없앤다. 즉, T_k 가 $v(k+1)$ 보다 일정수준 이상 낮거나, $q(t_k)$ 이 q_{low} 보다 작은 경우 다음 세그먼트의 비트율을 바꾼다.

알고리즘1 Segment Bit-Rate Filtering Module

```

1:    $\hat{v} = f \times T_k;$ 
2:    $v(k+1) = Q(\hat{v})$ 
3:   if  $v(k+1) > v(k)$  then
4:     if  $T_k / v(k+1) > a$  and  $q(t_k^{(e)}) < q_{high}$  then
5:        $v(k+1) = v(k);$ 
6:     end if
7:   else if  $v(k+1) < v(k)$  then
8:     if  $T_k / v(k+1) < b$  and  $q(t_k^{(e)}) > q_{low}$  then
9:       if  $q_{low\_flag} \equiv true$  then
10:         $q_{low\_flag} \equiv false;$ 
11:         $v(k+1) = v(k);$ 
12:      end if
13:    end if
14:    if  $q(t_k^{(e)}) < q_{low}$  and  $q(t_k^{(e)}) > q_{min}$  and
15:       $q_{low\_flag} \equiv false$  then
16:         $q_{low\_flag} = true;$ 
17:      else if  $q(t_k^{(e)}) < q_{low}$  and  $q(t_k^{(e)}) > q_{min}$  and
18:         $q_{low\_flag} \equiv true$  then
19:           $v(k+1) = v(k);$ 
20:        end if
21:      end if

```

$q(t_k)$ 가 감소하는 상황에서 처음으로 q_{low} 밑으로 내려가게 되면 세그먼트 비트율이 급격하게 감소 할 가능성이 있으므로, q_{low_flag} 를 사용하여 세그먼트 비트율의 급격한 감소를 막는다. 일단 q_{low_flag} 는 시작과 동시에 false로 초기화 되고, $q(t_k)$ 가 q_{low} 와 q_{min} 사이에 존재하고 q_{low_flag} 가 false면 세그먼트 비트율을 감소시키고 q_{low_flag} 를 true로 반전시킨다. 이와 반대로, $q(t_k)$ 가 q_{low} 와 q_{min} 사이에 존재하면서 q_{low_flag} 가 true면 세그먼트 비트율을 유지시키면서 불필요한 비트율 변화를 줄이고 비트율이 필요 이상으로 낮아지는 것을 방지한다. 앞선 세그먼트 비트율의 감소에도 불구하고 $q(t_k)$ 가 계속해서 감소하여 q_{min} 보다 낮아지는 경우에는 버퍼 언더플로우를 방지하는 것이 최우선 사항이기 때문에 세그먼트 비트율을 충분히 감소시킨다. 세그먼트 비트율이 충분히 감소되어 $\Delta q(t_k)$ 가 양수가 되고 $v(k+1)$ 이 $v(k)$ 보다 큰 경우에는 q_{low_flag} 를 다시 false로 반전시켜 $q(t_k)$ 가 감소할 때 앞서 설명한 과정들을 반복한다.

3. 시작 메커니즘

DASH 스트리밍 서비스를 시작할 때 SBFM로 인해 DASH 서비스 사용자가 일정시간동안 저화질의 비디오를 시청해야 하는 현상이 발생한다. 따라서 mFDASH에서는 시작 메커니즘을 추가하여 스트리밍 서비스를 시작할 때 언더플로우가 나지 않으면서 더 높은 화질로 스트리밍 서비스를 받을 수 있게 한다.

알고리즘2 시작 메커니즘

```

1:  $v_{BEGIN} = \hat{Q}(T_k/c);$ 
2: if  $Begin-flag \equiv false$  then
3:   if  $T_k > T_{k-1}$  then
4:      $v(k+1) = v_{BEGIN};$ 
5:   else
6:      $Begin-flag = true;$ 
7:   end if
8: end if

```

알고리즘2는 시작 메커니즘에 대한 의사 코드이다. 스트리밍 서비스를 시작하면 시작 비트율인 v_{BEGIN} 을 $\hat{Q}\left(\frac{T_k}{c}\right)$ 로 설정한다. $\hat{Q}(\cdot)$ 는 양자화 함수로, 선택 가능한 세그먼트 비트율 중 $\frac{T_k}{c}$ 보다 큰 가장 작은 비트율이 v_{BEGIN} 이 되며, c 는 시작 세그먼트 비트율을 조절하는 계수이다. 시작 메커니즘은 T_k 가 처음으로 T_{k-1} 보다 작은 값을 가지기 전까지 $v(k+1)$ 을 증가 시키며 T_k 가 T_{k-1} 보다 작아지는 순간 시작 메커니즘을 빠져나간다.

4. 대기 메커니즘

보통 모바일 기기의 버퍼 사이즈는 한계가 정해져 있다. 따라서 DASH 스트리밍 서비스가 진행이 되는 도중에 $q(t_k)$ 가 q_{high} 를 넘어가면 대기 메커니즘이 동작하여 $q(t_k) - q_{high}$ 초 만큼 대기한 후에 다음 세그먼트를 요청한다.

IV. 모의실험 환경 설정 및 결과

본 장에서는 NS-3를 이용한 점대점 네트워크와 Wi-Fi 환

경에서의 모의실험 결과를 분석하여 mFDASH의 성능을 평가한다. 점대점 네트워크 환경에서는 DASH 서버와 DASH 서비스 사용자가 점대점 링크로 연결되어 있으며, 링크 용량의 장기(long-term) 변화가 존재하는 환경과 주기적인 단기(short term) 변화가 존재하는 환경에서 모의실험을 진행하였다. Wi-Fi 네트워크 환경에는 하나의 서버와 하나의 DASH 서비스 사용자, 그리고 5개의 백그라운드 트래픽(Background traffic)이 존재하며, 백그라운드 트래픽의 주기를 다르게 하여 총 6개의 서로 다른 Wi-Fi 환경에서 모의실험을 진행하였다. 적응적 스트리밍 기법의 성능 평가 지표에는 대표적으로 세그먼트 비트율의 크기, 세그먼트 비트율 변화 빈도, 세그먼트 비트율의 변화 폭(세그먼트 비트율의 표준편차), 시작 지연, 재생 끊김이 있다[22]. 그 중에서 세그먼트 비트율의 크기, 세그먼트 비트율 변화 빈도, 재생 끊김, 시작 지연은 큰 영향을 미치지만, 세그먼트 비트율의 변화 폭은 사용자의 QoE에 큰 영향을 미치지 않는다[23][24]. 또한, mFDASH와 FDASH는 모두 가장 작은 비트율의 세그먼트를 맨 처음 세그먼트로 요청하기 때문에, 시작 지연을 성능 평가 지표로 고려하지 않았다. 따라서 본 논문의 모의실험 결과에서는 세그먼트 비트율의 크기, 세그먼트 비트율 변화 빈도, 재생 끊김 지표를 사용해서 성능을 평가 하였다.

세그먼트 재생시간(τ)은 2초로 설정하였고, q_{high} 는 현재 보편적으로 사용되는 모바일 기기의 성능을 고려하여 30초로 설정하였으며, 이상적인 버퍼 점유율(T)은 20초로 설정하였다. 또한, q_{low} , q_{min} 은 각각 10초, 7초로 설정하였고, SBFM에서 사용된 계수 a, b 는 각각 0.8, 1.5로 설정하였으며, 시작 메커니즘에서 사용한 계수 c 는 3으로 정하였다. DASH서버에 있는 세그먼트의 비트율 종류는 총 20가지이며, 각각의 비트율은 45Kbps, 89Kbps, 131Kbps, 178Kbps, 221Kbps, 263Kbps, 334Kbps, 396Kbps, 522Kbps, 595Kbps, 791Kbps, 1.033Mbps, 1.245Mbps, 1.547Mbps, 2.134Mbps, 2.484Mbps, 3.079Mbps, 3.527Mbps, 3.840Mbps, 4.220Mbps이다.

1. 장기 변화 점대점 네트워크 환경

그림 6는 장기 변화 점대점 네트워크 환경에서의 모의실

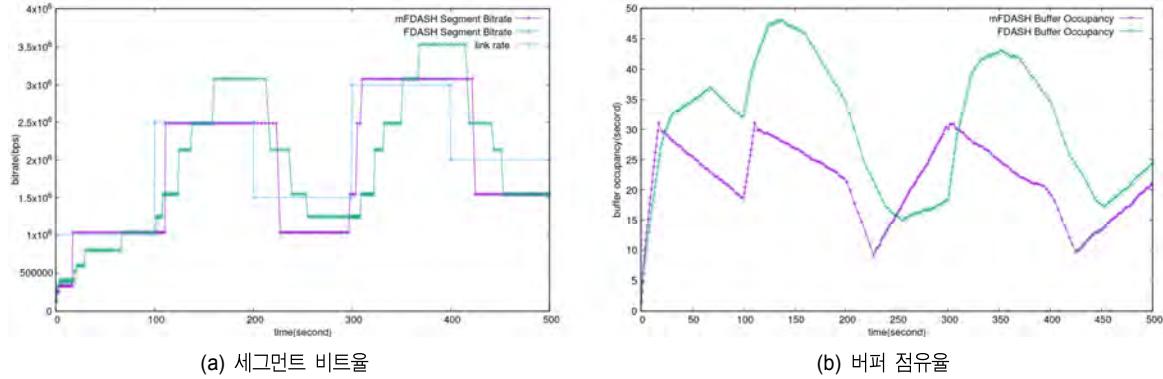


그림 6. 장기 대역폭 변화 점대점 네트워크 환경 시뮬레이션 결과
Fig. 6. Simulation result in long-term bandwidth variation Point-to-Point network

표 2. 장기 대역폭 변화 점대점 네트워크 환경 성능 지표

Table 2. Performance index in long-term bandwidth variation Point-to-Point network

적응적 스트리밍 기법 Adaptive Streaming Algorithm	평균 세그먼트 비트율 Average Segment Bit-rate	평균 비트율 변화 횟수 Average number of Bit-rate Changes	재생 끊김 횟수 Number of Interruptions
FDASH	1.721Mbps	24	0
mFDASH	1.708Mbps	11	0

험 결과를 나타낸다. 그림 6(a)는 mFDASH와 FDASH의 세그먼트 비트율과 점대점 링크의 용량을 나타내며 링크 용량은 100초 간격으로 1Mbps, 2.5Mbps, 1Mbps, 3Mbps, 1.5Mbps이다. 그림 6(b)는 mFDASH와 FDASH의 $q(t_k)$ 를 나타낸다. FDASH의 경우 $q(t_k)$ 가 50초 근처까지 올라가기 때문에 버퍼 오버플로우의 위험이 있지만, mFDASH의 경우에는 $q(t_k)$ 가 q_{high} (버퍼 사이즈)보다 높이 올라가지 않기 때문에 버퍼 오버플로우가 발생할 위험이 없다는 것을 확인 할 수 있다.

표 2에는 장기 변화 점대점 환경에서의 평균 세그먼트 비트율, 세그먼트 비트율 변화 횟수, 재생 끊김 횟수가 나타나 있다. mFDASH가 FDASH에 비해 평균 세그먼트 비트율이 0.7% 정도 낮지만, 비트율 변화 횟수면에서는 54% 정도 좋은 성능을 내는 것을 확인 할 수 있다. 재생 끊김 횟수는 두 스트리밍 기법 모두 0번으로 좋은 성능을 보인다.

2. 주기적 단기 변화 점대점 네트워크 환경

주기적 단기 대역폭 변화 점대점 네트워크 환경에서의

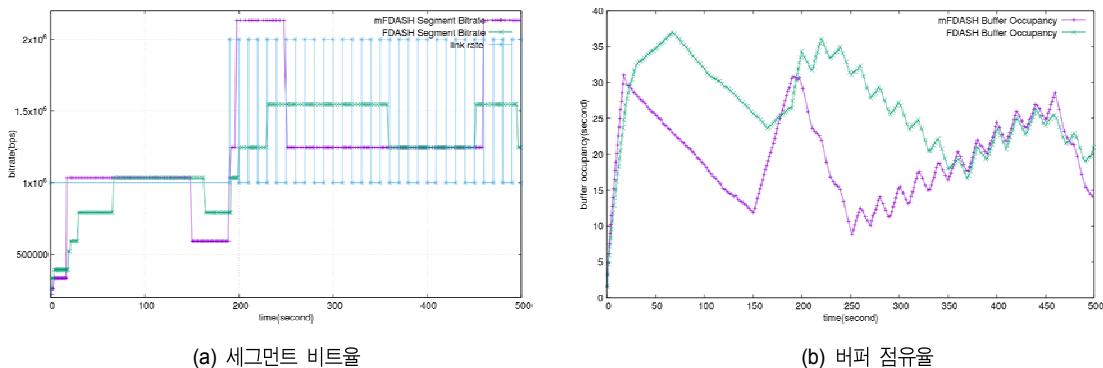


그림 7. 주기적 단기 대역폭 변화 점대점 네트워크 환경 시뮬레이션 결과
Fig. 7. Periodic short-term bandwidth variation Point-to-Point network

표 3. 주기적 단기 대역폭 변화 점대점 네트워크 환경 성능 지표

Table 3. Performance index in periodic short-term bandwidth variation Point-to-Point network

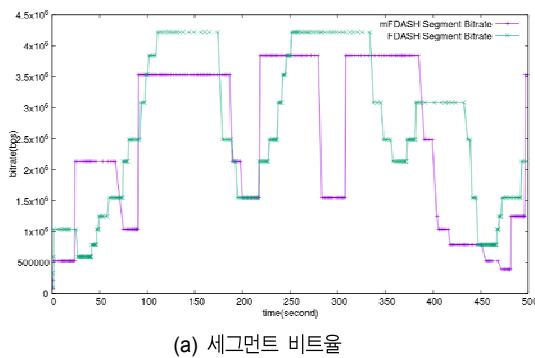
적응적 스트리밍 기법 Adaptive Streaming Algorithm	평균 세그먼트 비트율 Average Segment Bit-rate	평균 비트율 변화 횟수 Average number of Bit-rate Changes	재생 끊김 횟수 Number of Interruptions
FDASH	1.116Mbps	15	0
mFDASH	1.107Mbps	11	0

시뮬레이션 결과는 그림 7에 나타나있다. 그림 7(a)는 mFDASH와 FDASH의 세그먼트 비트율과 점대점 링크의 용량을 나타내며, 링크 용량은 0초~190초까지는 1Mbps이고, 이후에는 10초 주기로 1Mbps와 2Mbps 사이에서 흔들린다. 그림 7(b)는 mFDASH와 FDASH의 $q(t_k)$ 를 나타내며, 이를 통해서 FDASH의 $q(t_k)$ 가 30초를 초과하여 35초 이상까지 버퍼를 사용하지만 mFDASH의 $q(t_k)$ 는 30초를 초과하면 대기 메커니즘의 동작으로 대기상태에 들어가므로 버퍼를 많이 사용하지 않는 것을 확인 할 수 있고, 이는 mFDASH를 사용했을 때 버퍼 오버플로우가 발생할 가능성이 더 낮음을 의미한다.

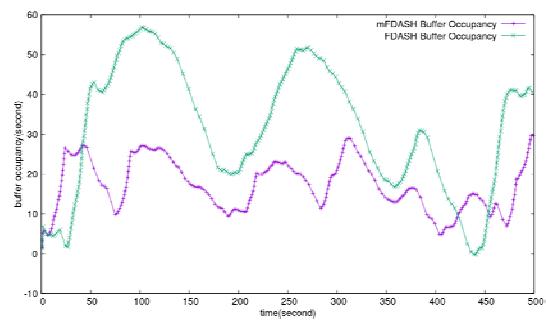
표 3은 주기적 단기 변화 점대점 네트워크 환경에서의 평균 세그먼트 비트율, 세그먼트 비트율 변화 횟수, 그리고 재생 끊김 횟수가 정리되어 있다. mFDASH가 FDASH에 비해 평균 세그먼트 비트율은 1.6%정도 낮지만, 비트율 변화 횟수는 mFDASH가 26.7% 더 낮게 가져가면서 좋은 성능을 내는 것을 확인 할 수 있다.

3. Wi-Fi 네트워크 환경

일반 Wi-Fi 네트워크 환경에서의 모의실험 결과는 그림

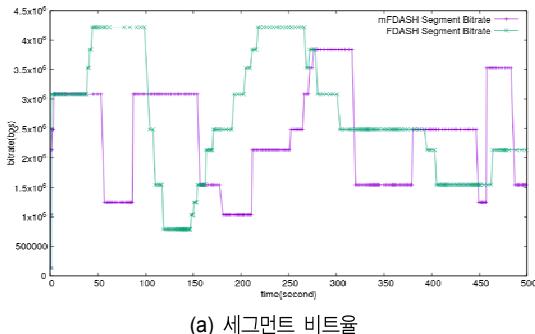


(a) 세그먼트 비트율

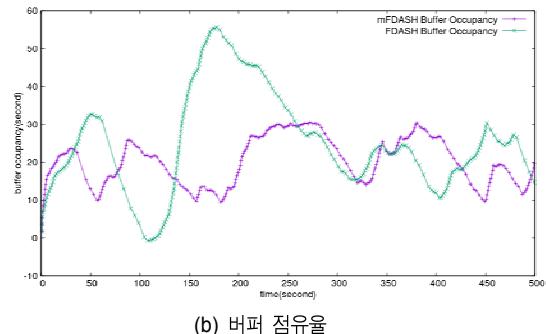


(b) 버퍼 점유율

그림 8. 일반 Wi-Fi 네트워크 환경 첫 번째 시뮬레이션 결과
 Fig. 8. First simulation result in Wi-Fi network



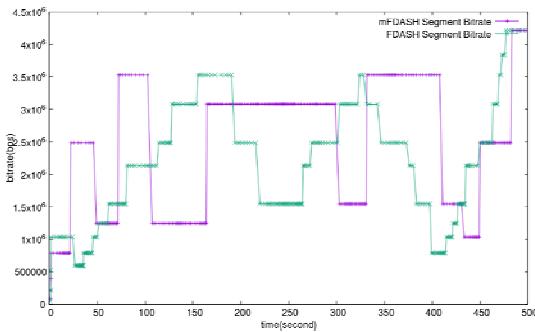
(a) 세그먼트 비트율



(b) 버퍼 점유율

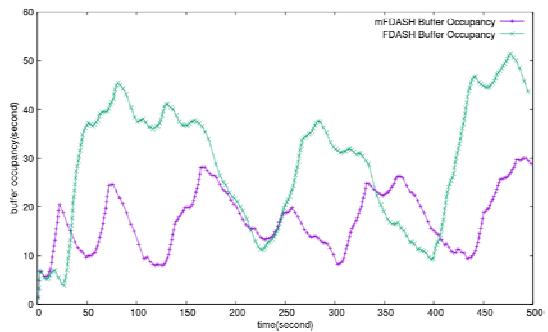
그림 9. 일반 Wi-Fi 네트워크 환경 두 번째 시뮬레이션 결과
 Fig. 9. Second simulation result in Wi-Fi network

8~그림 13에 나타나있고, 성능지표는 표 4에 나타나있다. Wi-Fi 네트워크 환경에서는 백그라운드 트래픽의 주기성을 달리하여 총 6번의 시뮬레이션을 진행 하였다.



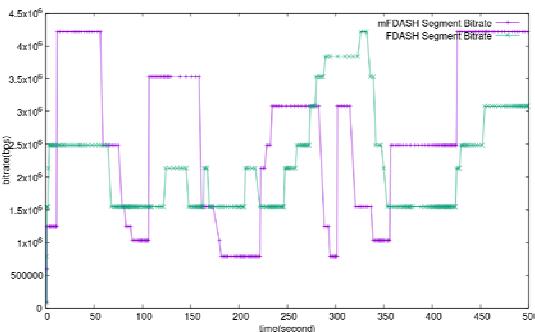
(a) 세그먼트 비트율

그림 8~그림 13의 (b)를 보면 FDASH의 경우 버퍼에 50초 정도까지 데이터가 쌓이는 것을 확인 할 수 있다. 반대로, mFDASH의 경우 q_{high} 이상 올라가지 않는 안정적인

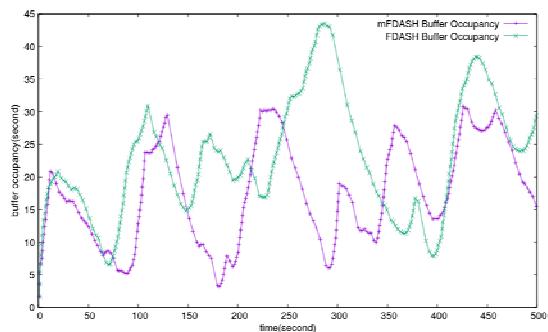


(b) 버퍼 점유율

그림 10. 일반 Wi-Fi 네트워크 환경 세 번째 시뮬레이션 결과
Fig. 10. Third simulation result in Wi-Fi network

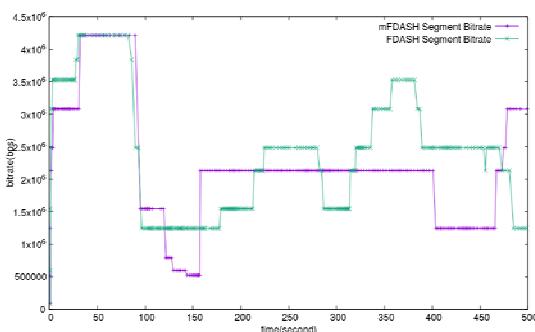


(a) 세그먼트 비트율

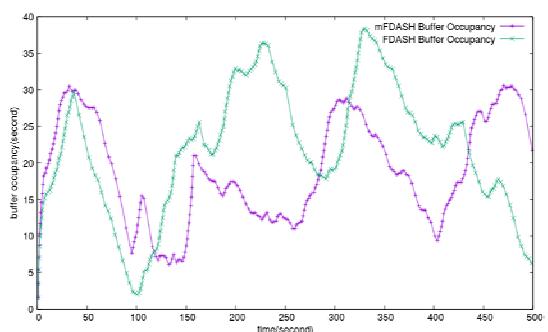


(b) 버퍼 점유율

그림 11. 일반 Wi-Fi 네트워크 환경 네 번째 시뮬레이션 결과
Fig. 11. Fourth simulation result in Wi-Fi network



(a) 세그먼트 비트율



(b) 버퍼 점유율

그림 12. 일반 Wi-Fi 네트워크 환경 다섯 번째 시뮬레이션 결과
Fig. 12. Fifth simulation result in Wi-Fi network

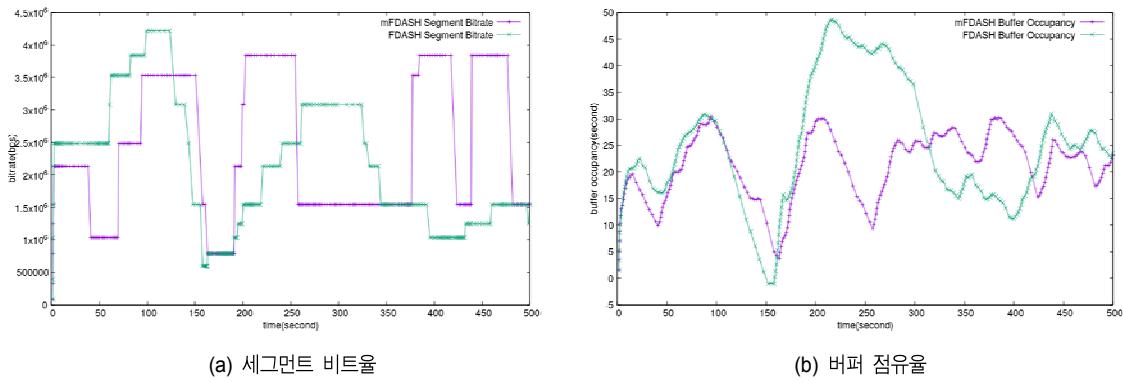


그림 13. 일반 Wi-Fi 네트워크 환경 여섯 번째 시뮬레이션 결과
Fig. 13. Sixth simulation result in Wi-Fi network

모습을 보인다. 즉, mFDASH가 오버플로우 문제에 관하여 더 안정적인 모습을 보여준다. 또한, 그림 9과 그림 13의 (b)을 보면, FDASH의 경우 $q(t_k)$ 가 0이 되는 버퍼 언더플로우 현상이 발생하는 것을 확인 할 수 있다. 표 4를 통해 FDASH가 버퍼 언더플로우에 의해 재생 끊김 현상이 총 76번이 발생하고, 그에 반해 mFDASH는 버퍼 언더플로우가 발생하지 않아 재생 끊김 현상이 일어나지 않는 것을 확인 할 수 있다. 즉, 버퍼 오버플로우/언더플로우 관점에서 mFDASH가 FDASH에 비해 좋은 QoE을 제공하는 것을

확인 할 수 있다.

표 4를 보면, mFDASH는 6번의 시뮬레이션에서 평균적으로 2.153Mbps 세그먼트 비트율을 가지고, FDASH는 2.127Mbps의 세그먼트 비트율을 가진다. 즉, mFDASH가 약 1.2%정도 높은 평균 세그먼트 비트율을 가진다. 또한, 평균 비트율 변화 횟수는 mFDASH가 18.5번, FDASH가 32번으로 mFDASH가 42%정도 좋은 성능을 보이는 것을 확인 할 수 있다.

표 4. Wi-Fi 환경 성능 지표
Table 4. Performance index in Wi-Fi network

적응적 스트리밍 기법 Adaptive Streaming Algorithm	평균 세그먼트 비트율 Average Segment Bit-rate		평균 비트율 변화 횟수 Average Number of Bit-rate Changes		재생 끊김 횟수 Number of Interruptions	
	1회	2.27261Mbps	1회	36	1회	0
FDASH	2회	2.38311Mbps	2회	26	2회	25
	3회	1.84099Mbps	3회	32	3회	0
	4회	2.02917Mbps	4회	25	4회	0
	5회	2.29307Mbps	5회	26	5회	0
	6회	1.94372Mbps	6회	29	6회	51
	평균	2.12700Mbps	평균	32	합계	76
mFDASH	1회	2.20100Mbps	1회	19	1회	0
	2회	2.19932Mbps	2회	19	2회	0
	3회	2.16935Mbps	3회	15	3회	0
	4회	2.19115Mbps	4회	22	4회	0
	5회	2.02663Mbps	5회	16	5회	0
	6회	2.13292Mbps	6회	20	6회	0
	평균	2.15300Mbps	평균	18.5	합계	0

V. 결 론

본 논문에서는 DASH 스트리밍 서비스를 위한 수정된 FDASH 기법을 제안 하였다. 제안된 기법은 FDASH의 FLC를 수정하고, SBFM, HBTTE 기법, 시작 메커니즘, 대기 메커니즘을 적용하였다. 4장에서 언급한 인자들($a, b, c, q_{min}, q_{low}, T$)을 적절히 조절하여 평균 세그먼트 비트율 변화 빈도에 대한 성능을 높였다. 또한 NS-3를 이용하여 제안된 기법을 FDASH와 비교하여 비슷한 비디오 화질 성능을 보임과 동시에 세그먼트 비트율 변화 빈도 측면에서는 mFDASH가 훨씬 좋은 성능을 보임을 검증하였고, mFDASH의 버퍼 점유율이 q_{high} 를 넘기지 않는 것을 확인하면서 버퍼 오버플로우가 발생 하지 않음을 확인 하였다. 제안하는 기법을 DASH 사용자가 사용할 경우, 기존의 FDASH 알고리즘에 비해 더 높은 QoE를 제공 받을수 있을것으로 기대된다. 또한, 향후에는 하나의 DASH 사용자에 대한 DASH 알고리즘뿐만 아니라, 하나의 네트워크 안에 존재하는 다수의 DASH 사용자들에게 최적의 QoE를 제공하는 DASH 알고리즘에 대한 연구를 진행 할 예정이다.

참 고 문 헌 (References)

- [1] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet." IEEE MultiMedia Vol.18, No.4, pp.62-67, April 2011.
- [2] T. Stockhammer, "Dynamic adaptive streaming over HTTP--: standards and design principles," Proceedings of the second annual ACM conference on Multimedia systems, San Jose, CA, USA, pp.133-144, 2011.
- [3] M. Park, and Y. Kim, "MMT-based Broadcasting Services Combined with MPEG-DASH," Journal of Broadcast Engineering, Vol.20, No.6, pp.283-299, March 2015.
- [4] G. Park, G. Lee, J. Lee, and K. Kim, "HTTP Adaptive Streaming Method for Service-compatible 3D Contents Based on MPEG DASH," Journal of Broadcast Engineering, Vol.17, No.2, pp.207-222, March 2012.
- [5] Y. Kim, and M. Park, "MPEG-DASH Services for 3D Contents Based on DMB AF," Journal of Broadcast Engineering, Vol.18, No.1, pp.115-121, January 2013.
- [6] C. Zhou, Lin, C. W., and Guo, Z., "mDASH: A markov decision-based rate adaptation approach for dynamic HTTP streaming." IEEE Transactions on Multimedia, Vol.18, No.4, pp.738-751, January 2016.
- [7] D. L. Isaacson, and W. M. Richard, Markov chains, theory and applications. Vol. 4. New York: Wiley, 1976.
- [8] R. M. Blumenthal, and R. K. Getoor, Markov processes and potential theory. Courier Corporation, 2007.
- [9] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "Scheduling and resource allocation for wireless dynamic adaptive streaming of scalable videos over HTTP." Communications (ICC), Sydney, NSW, Australia, pp. 1681-1686, 2014.
- [10] H. Schwarz, D. Marpe, and T. Wiegand. "Overview of the scalable video coding extension of the H. 264/AVC standard." IEEE Transactions on circuits and systems for video technology Vol.17, No.9, pp.1103-1120, September 2007.
- [11] G. Tian, and Y. Liu. "Towards agile and smooth video adaptation in dynamic HTTP streaming." Proceedings of the 8th international conference on Emerging networking experiments and technologies, Nice, France, pp.109-120, 2012.
- [12] Q. He, C. Dovrolis, and M. Ammar. "On the predictability of large transfer TCP throughput." ACM SIGCOMM Computer Communication Review, Vol. 35, No. 4, ACM, October 2005.
- [13] G. Klir, and B. Yuan. Fuzzy sets and fuzzy logic, New Jersey: Prentice hall, 1995.
- [14] L. A. Zadeh, "Fuzzy logic." Computer Vol.21, No.4, pp.83-83, April 1988.
- [15] D. J. Vergados, et al, "FDASH: A Fuzzy-Based MPEG/DASH Adaptation Algorithm." IEEE Systems Journal Vol.10, No.2, pp.859-868, December 2015.
- [16] R. K.P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: a QoE-aware DASH system." Proceedings of the 3rd Multimedia Systems Conference, New York, NY, USA, pp.11-22, 2012.
- [17] SG12, I. T. U. T. "Definition of quality of experience." TD 109rev2 (PLEN/12), Geneva, Switzerland, pp.16-26, 2007.
- [18] H. R. Berenji, "Fuzzy logic controllers." An Introduction to Fuzzy Logic Applications in Intelligent Systems. Springer US, pp. 69-96, 1992.
- [19] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I." IEEE Transactions on systems, man, and cybernetics, Vol.20, No.2, pp.404-418, March/April, 1990.
- [20] The network simulator - ns-3, <http://www.nsnam.org/> (accessed May. 25, 2017).
- [21] Z. Bingül, and O. Karahan, "A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control," Expert Systems with Applications, Vol.38, No.1, pp.1017-1031, January 2011.
- [22] M. Seufert, S. Egger, M. Slanina, and T. Zinner, "A survey on quality of experience of HTTP adaptive streaming," IEEE Communications Surveys & Tutorials, Vol.17, No.1, pp.469-492, March 2015.
- [23] K. Xiao, S. Mao, and J. K. Tugnait, "QoE-Driven Resource Allocation for DASH over OFDMA Networks," Proceedings of Global Communications Conference (GLOBECOM), Washington, DC, USA, pp.1-6, 2016.
- [24] S. Egger, B. Gardlo, M. Seufert, and R. Schatz, "The impact of adaptation strategies on perceived quality of http adaptive streaming," Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming. Sydney, Australia, pp.31-36, 2014.

저자소개

김현준



- 2017년 : 건국대학교 전자공학과 학사과정 졸업
- 2017년 ~ 현재 : 건국대학교 전자정보통신공학과 석사과정
- ORCID : <http://orcid.org/0000-0001-5457-8957>
- 주관심분야 : 무선 이동통신 네트워크, 멀티미디어 스트리밍 프로토콜, 최적화 이론

손예슬



- 2017년 : 건국대학교 전자공학과 학사과정 졸업
- 2017년 ~ 현재 : 건국대학교 전자정보통신공학과 석사과정
- ORCID : <http://orcid.org/0000-0001-8048-9966>
- 주관심분야 : 무선 이동통신 네트워크, 멀티미디어 스트리밍 프로토콜

김준태



- 1990년 : 한국과학기술원 학사과정 졸업
- 1993년 : 한국과학기술원 석사과정 졸업
- 1998년 : 한국과학기술원 박사과정 졸업
- 1998년 ~ 2003년 : LG전자 DTV 연구소 책임 연구원
- 2003년 ~ 현재 : 건국대학교 전자공학부 교수
- ORCID : <http://orcid.org/0000-0001-6953-5482>
- 주관심분야 : 차세대 방송 통신 & 디지털 TV 시스템, 통신 시스템