ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

A DASH Based Scheme for Video Streaming over Multiple Wireless Access Networks: A Review

Ashwini N. Nikumbhe[1], Chhaya Nayak[2]

M.Tech Student, Dept. of CSE, B. M. Technology, RGPV University, Bhopal, MP, India[1]

Professor, Dept. of CSE, B. M. Technology, RGPV University, Bhopal, MP, India[2]

ABSTRACT: As Video streaming is very popular among mobile users. So there is need to efficiently and cost-effectively use multiple links to improve video streaming quality. So in this paper, the optimal video streaming process with multiple links is formulated as an Markov Decision Process (MDP) to maintain the high video streaming quality with less wireless service cost The reward function is designed to consider the quality of service (QoS) requirements for video traffic, such as the startup latency, playback fluency, average playback quality, playback smoothness and wireless service cost. To solve the MDP in real time, we propose an adaptive, best-action search algorithm to obtain a sub-optimal solution. To assess the performance of the proposed adaptation algorithm, we implemented a testbed using the Android mobile phone and the Scalable Video Coding (SVC) codec.

KEYWORDS: DASH; Video streaming; MDP; SVC; multiple links.

I. INTRODUCTION

As Internet access is becoming most important thing on mobile devices like smart phones, smartbooks, connected netbooks and laptops the Mobile Internet use is increasing. At the same time mobile users expect high-quality video experience in terms of video quality. So the Media streaming applications over wireless environments have drawn the attention of the research community.

As mobile devices have limited storage capacity and energy supply, and the wireless channels are highly dynamic, it is very challenging to provide high quality video streaming services for mobile users consistently. So it is assuring way to use multiple wireless network interfaces with different wireless communication techniques for mobile devices. For example, smart phones and tablets are usually equipped with cellular, WiFi and Bluetooth interfaces. So by applying multiple links concurrently can improve video streaming in several aspects: the aggregated higher bandwidth can support video of higher bit rate; when one wireless link suffers poor link quality or congestion, the others can reward for it [1]. So high flexibility to bandwidth variation and easy deployment are both important requirements for video streaming applications.

In this paper, we proposed dynamic adaptive streaming over HTTP (DASH) strategy. Videos encoded in different versions & multiple copies are stored into small segments. After the client receives one segment, it has a chance to decide which version of the video to request for the next segment, based on the current network condition. Thus, rate adaptation is performed at the client side & client can request the appropriate quality version based on its screen resolution, current available bandwidth, and buffer occupancy status.

Specifically, we focus on the rate adaptation algorithm for streaming scalable video in wireless networks. We model the rate adaptation problem as a Markov Decision Process (MDP), and also implemented depth-first real-time search algorithm aiming to find an optimal streaming strategy in terms of user-perceived quality of experience (QoE) such as playback interruption, average playback quality and playback smoothness . We then obtain the optimal MDP solution

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

using dynamic programming [2]. In this paper, we formulate the multi-link video streaming process as a reinforcement learning task. Also we used reward function to carefully designed to consider the video QoS requirements, such as the interruption rate, average playback quality, and playback smoothness, as well as the service costs. DASH, as an extension of the classic HTTP streaming, is able to automatically switch the video quality level according to the current network conditions [1]. The main contributions of this paper are-

First, we formulate the video streaming process over multiple links as an MDP problem. To achieve smooth and high quality video streaming, we define several actions and reward functions for each state. Second, we propose a depth-first real-time search algorithm. The proposed adaptation algorithm will take several future steps into consideration to avoid playback interruption and achieve better smoothness and quality. Last, we implement a realistic testbed using an Android phone and Scalable Video Coding (SVC) encoded videos to evaluate the performance [1].

II. LITERATURE SURVEY

DASH has been a hot topic in recent years. There are many products that have implemented DASH in different ways, such as Apple HTTP Live Streaming and Microsoft Smooth streaming [1].

The authors in [2] proposed optimal rate adaptation algorithm for streaming over SVC over HTTP using MDP. In SVC each video frame is encoded into a base layer & several enhancement layers. This paper consist only single link case and in this work, we consider the more challenging case with multiple access links. T. Stockhammer introduced the 3GPP & specification draft version of MPEG in [3]. In addition it adds an informative description on how a DASH client may use the provided information to establish a streaming service for the user. There are many authors who have been done research efforts on Adaptive video streaming over HTTP. In [4] S. Akhshabi designed an evaluation method to test the performance & compared several existing commercial DASH products, such as two commercial players-Smooth Streaming & Netflix, and one open source player OSMF. In [5] K.P.Mok presented a QoE aware DASH system to improve the user perceived quality of video. They integrated the available bandwidth measurement by probing with the video data. To enhance the quality experience their algorithm will switch the video quality levels according to network conditions. In [6], T. Kupka proposed to evaluate the performance of live DASH under on/off traffic and tested four different methods to improve the performance. How to efficiently deliver video in heterogeneous wireless networks has been extensively studied, and different layer approach have been proposed in [6,7]. Recently, a few approaches have appeared to extend the DASH technique to support multiple links. Kaspar et al. proposed an approach to implementing DASH over multiple links [8]. In their algorithm, each segment will be transmitted over one link. Thus multiple segments can be transmitted at the same time. To reduce the overhead, they used HTTP pipelining to improve the performance. This approach may lead to the “last-segment problem” due to the link transmission speed difference. To overcome this disadvantage, in [9],[10], Evensen et al. suggested to divide each segment into small sub-segments, and these sub-segments can be downloaded through different links. Their algorithm estimates the available bandwidth according to the throughput of the previous segment, and selects the video quality version most close to the estimated bandwidth.

III. METHODOLOGY

A *System Architecture*

How to utilize multiple wireless access networks together for video streaming like g a combination of cellular, WiFi, and/or Bluetooth simultaneously. In this paper we are going to study about Bluetooth and WiFi access networks. As a wireless channel can suffer from time-varying fading, shadowing, interference and congestion, the available bandwidth of a wireless link may vary all the time and also the different smart phones or tablets may have different screen size and resolution. Taking these two things into consideration, the server should store many different quality copies of a video. As the videos are encoded with SVC into a base layer and several enhancement layers and chopped into segments and each segment can be played with a fixed duration [1].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

The rate adaptation agent will request a video segment of appropriate quality version based on the current queue length and estimated available bandwidth. Once the request decision is completed, HTTP requests over both WiFi and Bluetooth and the video segment is downloaded. This process will repeat until the termination of the video streaming by the user

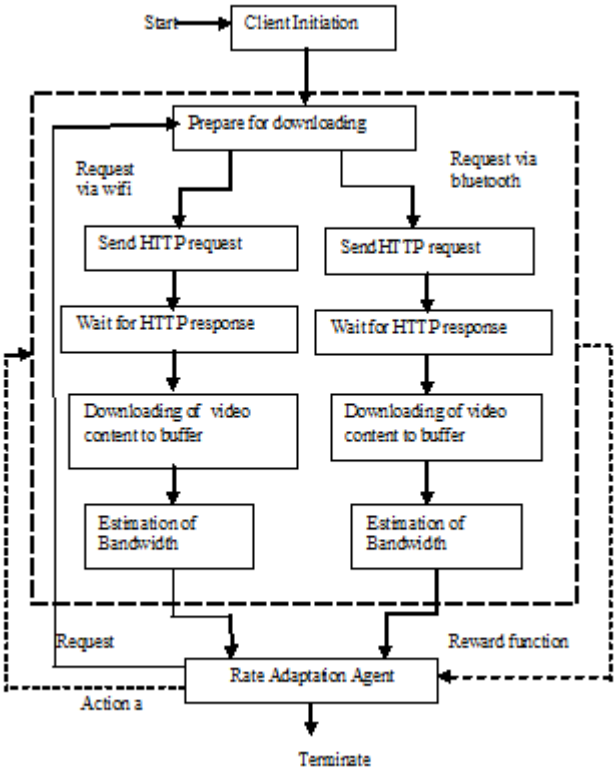

Fig. 1. System Architecture

B *Streaming Process Formulation*

In the above figure downloading & estimation steps at the top viewed as an integrated environment module, and the rate adaptation agent can be viewed as an agent module. The environment sends a state signal for each video segment to the agent, and the agent will determine the best action correspondingly and in each action, the environment replies a reward to the agent. As the system states having the Markov Property , a Markov Decision Process (MDP) can be formulated for the streaming process & needs to devise the state transition model of the Markov process.

Consider m step as to download segment m, For each state at m as- $s_m = \{q_m, \Delta q_m, v_m, \Delta v_m, t_m, \Delta t_m, bw_m, bt_m, d_m\}$, Where q_m is the queue length as the number of buffered (unplayed) segments, q_m is in the range of (0 to F). v_m is the SVC video version layer index of the last received m^{th} segment. In this work, there are F SVC video layer & F represents higher quality layer. Δq_m is the queue length variation i.e. $\Delta q_m = q_m - q_{m-1}$, it will determine wheather the requested video's bit rate matches the available bandwidth. Δv_m is the difference of video versions where $\Delta v_m = v_m - v_{m-1}$. To download the previous segment the total traffic of Bluetooth is recorded by t_m, and $\Delta t_m = t_m - t_{m-1}$. The current bandwidth states of the WiFi link is bw_m Bluetooth link is denoted by bt_m, d_m is the current requested segment which is in the range of $[0, N_T]$. Where N_T is the total number of segments the clients needs to request.

The rate adaptation agent receives input of state s_m from the environment, it will make the decision of which action to take. So here we define four types of actions, A_c, A_u, A_w, and A_p. The two actions A_c is used for both Wifi &

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Bluetooth links and A_u is used to only for Wifi link for downloading the next segment at the quality determined by v, and Δv is used to determine whether to upgrade, downgrade or maintain the current quality. Once the quality v is decided, the base layer and all enhancement layers belong to quality v will be downloaded. As a Bluetooth is short distance wireless communication technique, using Bluetooth the client connect to the web server indirectly via tethering cellular data services. And also bluetooth connection is not that much reliable as it is easy to be use, So sometimes we only prefer Wifi connection. As the Aw represents the waiting action, and the client will wait for W seconds, equal to the duration of one segment. Since SVC video is encoded into different layers, in that higher enhancement layers can be requested to improve the streaming quality and smoothness of video. Therefore for smooth action As is used. So there are three principles use to follow for the smooth action. First, as there is a high probability of playback freezing due to queue length is shorter, the smooth action will not be taken. So to overcome this , the queue length is must be sufficiently larger, then the smooth action will be invoked. Second, the segment which will be played soon, we will not take the smooth action for it, because the requested enhancement layers may miss the playback deadline. Therefore we only take the smooth action for a few number of buffered segments which are stored in the tail part of the buffer [1]. The last principle of the smooth action is to ensure that it will not bring in any additional layer variation [1].

To obtain the optimal rate adaptation the most challenging issue is to obtain the wireless link bandwidth accurately. For most wireless streaming scenarios, Markov channel models are useful to describe the variations of wireless channels [12]. Thus, we use two discrete-time finite-state Markov models to describe the available bandwidth of the WiFi and Bluetooth links, respectively and also to capture the variation of the bandwidth with the state transition probabilities obtained from the measurement from the wireless channel model [12]. According to the Markov property, the state at any time instance only depends on its immediately previous state. So for any state s and action a, the transition probability of the MDP can be given as

$P^a_{ss'} = Pr\{s_m+1 = s' | s_m = s, a_m = a\}$. eq. (1)

As in our model we have two wireless links and considering they are independent, the transition probability is calculated as- $P^a_{ss'} = Pr\{bw'|bw\} \cdot Pr\{bt'|bt\}$. eq. (2)

For any state s = {q, Δq, v, Δv, t, Δt, bw, bt, d}, if action A_c is selected then the new state s' = {q',Δq', v',Δv', t',Δt', bw', bt', d'} can be derived as-

$q' = q - [XZ^{v'}_{d+1} / (bw' + bt')] + 1$,
$\Delta q' = q' - q$, $v' = v + m$, $\Delta v' = n$,
$\Delta t' = XZ^{v'}_{d+1} \cdot bw'/(bw' + bt')$,
$t' = t' + \Delta t'$, $d' = d + 1$, eq. (3)

Similarly we can derive other state transition probabilities.

The reward in MDP for the particular action is taken at a state is $r_m = R(s_m-1)$.i.e. is determined by the previous state. As shown is table, rewards are defined for the different states. In the reward function table I, * means any value for the state [1]. The reward of a state can be looked up in the table from the top to bottom, until one entry is matched to the current state. When d = NT, all the segments are downloaded, and the reward 0 is assigned to them. The reward functions are highly associated to the streaming quality of service i.e. QoS. Therefore by giving the minimum reward when the buffer is empty, we can minimize playback interruption by giving a negative reward to the state when the number of buffered frames is larger than the maximum value, to avoid buffer overflow [2]. The streaming policy π is a mapping of the possible action at each step .The long term reward $v^{\pi}(s)$ under policy π can be computed as-

$V^{\pi}(s) = P^a_{ss'}[R^a(s)+\gamma V^{\pi}(s')]$ eq. (4)

Where, R is the next reward of taking action a at state s. γ is the discount rate & $0<\gamma<1$.

So, finding optimal strategy policy $\pi^*(s)$ should maximize the state value function in the long run, So our multilink video streaming task can be formulated as an optimization problem:

$\pi^*(s) = \text{argmax} \sum_{s'} P^a_{ss'}[R^a(s)+\gamma V^*(s')]$ eq. (5)

Where $V^*(s)$ is the optimal value function.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

State s_t=s	Reward R(s)
(*,*,*,*,*,*,*,*,*,N_T)	0
(q, *,*,*,*,*,*,*,*,*), if q<T_{qmin}	-(Q_F-q)+ Δq
(q, *,*,*,*,*,*,*,*,*), if q>T_{qmax}	-q-Δq
(*,Δq,*, Δv, t, Δt,*,*,*)	min(-|Δv |,-|Δq |)- ΔtR_t
Other states	0
Any state with smooth action	R_s

Table I : Rewards Associated with states

IV. ALGORITHM DESIGN

As in the above section, we formulated an optimization problem for video streaming process. But this approach is not suitable for real-time adaptive streaming. To overcome this problem, we aim to develop a real-time best streaming action search algorithm to find a sub-optimal solution for the optimization problem formulated in the previous section [1].

A. *Bandwidth Estimation*

In this work, a heterogeneous and Time-varying Markov model is used to estimate the future bandwidth. The bandwidth of each link will be divided into several regions. Each region will represent a state of the Markov channel model, and the total number of the states is equal to the number of regions. Assume that there are p states, then an p*p transition matrix X will be used for the Markov channel model. Each element xij is the transition probability from state i to j. To obtain the transition probability, another p*p matrix C is used to count the number of transitions for each state. Once a segment has been successfully downloaded, the transmission bandwidth can be calculated by dividing the total size of the data transmitted over the total transmission time. Then the bandwidth region can be determined and we will increase the corresponding cij by one. x_{ij} is updated by the following equation-

$$X_{ij} = c_{ij} + 1/ \sum_{j=1}^{p} c_{ij} + p \quad \text{eq. (6)}$$

Initially, if there is no history data available, $c_{ij} = 0$, and x_{ij} is set to 1/p. The transition matrix will be updated after each segment has been successfully downloaded, so the transition matrix can better predict the future bandwidth variations with the recent measurements [1].

B. *Real-time Search Algorithm*

In this work, we develop a real-time recursive best-action search algorithm, to obtain a sub-optimal solution for the future steps consideration to avoid long computation time which is shown in Algorithm 1[1].

Algorithm 1 Real-Time Best-Action Search Algorithm
1: **Start procedure** GETBESTACTION(s)
2: Initialize action←−1, Qmax ←−∞
3: Generate all possible actions A(s) for state s
4: for all Action a ∈ A(s) do
q ← REWARDSEARCH(s, a, 0)
Check if q > Qmax then
Qmax ← q, action ← a
end if and for loop.
5: return action
6: **end procedure**

7: **Start procedure** REWARDSEARCH(s, a, d)
8: q ← reward of (s, a)
9: if d ≥ D then

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

return q and end if statement.
10: Generate all possible next states S' of (s, a)
11: for all s' from S' do
12: Qmax ←−∞
13: Generate all possible actions A'(s) for state s'
14: for all Action a'∈A' (s) do
Q_t ← REWARDSEARCH (s', a', d+ 1)
Check if $Q_t > Q_{max}$ then $Q_{max} \leftarrow Q_t$
end if
end for
15: q ← q + γ Pss' Qmax
16: end for
17: return q
18: **end procedure**

To meet the requirement of the real-time search, an important thing is to reduce the search duration for each state to an acceptable value. So for this we used small search depth D to invoke the search algorithm. For the current state s, all the possible actions A(s) will be enumerated. The recursive reward search algorithm is invoked to obtain the reward of state s with action a by enumerating all the possible future states S' and their associated actions A'(s) [1].

C.*Adaptive Search Depth*

Search depth is an important issue in our work. The search depth can determine how good the search result is, and a larger value of depth will achieve a better result [1]. From several preliminary experiment results, when the search depth D is larger than three, it will take more than two seconds to obtain a decision on the test Android smart phone [1]. We divide the buffer queue into three regions, [(0, q1), (q1, q2) and (q2, qL)]. For each state, the search depth D is determined according to its queue length q as follows [1]:

D =1 if q ∈[0, q1]
2 if q ∈[q1, q2]
3 if q ∈[q2, qL]

Generally, the space complexity of our algorithm is bound by O(bD).

V. PERFORMANCE EVALUATION

A. *Testbed Implementation*

A smart phone with Android OS is used as the client, to evaluate the performance of our proposed video streaming algorithm. We established a realistic testbed to measure the performance with a real video stream and an Android mobile phone.

The smart phone is integrated with WiFi, 3G and Bluetooth wireless interfaces. WiFi and Bluetooth are used together to transmit the video segments simultaneously according to our multi-link streaming. We implemented a video streaming application which can request the video contents from the web server through the HTTP/1.1 protocol. The video streaming application contains three main components: the streaming action search module, WiFi connection management module and Bluetooth connection management module. When the video streaming process begins, the streaming action search module will make the decision on how many enhancement layers to be requested and how to assign the transmission load to each link. Once the decision is made, WiFi and Bluetooth modules send HTTP/1.1 requests to the server to fetch the corresponding video segments. This process will continue until the last segment is successfully fetched. As there is no available SVC decoder for Android OS yet, we only use the smart phone to request the video segments and record the transmission trace.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

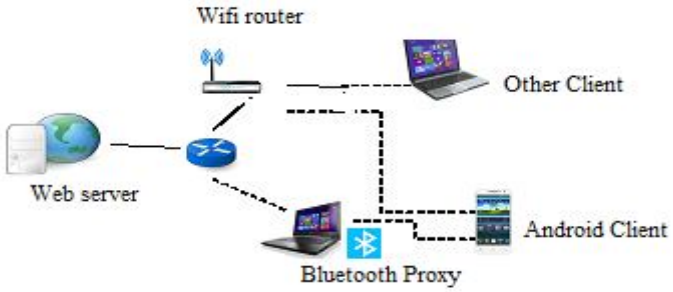

Fig. 2. Testbed Network Topology.

With the trace, we can decode the SVC video on a PC to evaluate the experiment results.
The video information is stored in a simplified manifest file in our implementation. In the manifest file, first the general video information is listed: the total number of segments, the duration of a segment, the total number of enhancement layers and the correspondent resolutions. Then the detailed bit rate of each segment and the size of each layer for each segment are listed. The manifest file will be transmitted to the client first before the stream begins [1].

As Bluetooth cannot connect the client to the web server directly, we implemented a Bluetooth proxy to let the client access the web server. The Bluecove Java Bluetooth library is used to implement the proxy on a laptop. When the client sends an HTTP/1.1 request, the proxy will forward the request to the web server, fetch the video segments from the server and also forward the fetched segments to the client [1].

B. *Experiment Settings*
The network topology of our testbed is shown in Fig. 2.
The video streaming server is deployed on a web server, which is deployed on a desktop by Glassfish HTTP Server with Windows 7 OS. Both the wireless router and the Bluetooth proxy are connected with the web server through wired links. In our experiment, the WiFi network is configured to be in the IEEE 802.11b mode and the wireless transmission rate is set to 1 Mbps or 2 Mbps. The Bluetooth proxy runs on a laptop which is equipped with a core-duo 2.00 GHz Intel CPU and 2 GB memory. We used a Nexus S(4.0") smart phone with Android 4.4 OS as the client. The Nexus S is equipped with a ARM 1.2 GHz CPU and 343 MB RAM. The open-source SVC codec JSVM [13] is used to encode a test video "Big Buck Bunny" (available at http://www.bigbuckbunny.org), and we did not pack the encoded video with any container [1].

For the Markov channel model, the available bandwidth for each link is divided into four regions, and thus two 4×4 transition matrices were used for the WiFi and the Bluetooth links, respectively. To reduce the search time of our real-time best-action search algorithm, we implemented it in a non-recursive way [1].

VI. CONCLUSION

In this paper, we proposed a real-time adaptive best-action search algorithm for video streaming over multiple wireless access networks like Wifi & bluetooth. First, we formulated the video streaming process as an MDP and to achieve smooth video streaming with high quality, we carefully designed the reward functions. Second, with the proposed rate adaptation algorithm, we can solve the MDP to obtain a sub-optimal solution in real time. Last, we explained the proposed algorithm. The proposed solution can give a lower startup latency, higher video quality and better smoothness and established a realistic testbed to measure the performance with a real video stream and an Android mobile phone. There are still some issues to investigate in the future. Like how to better allocate the loads between several links with finer granularity should be investigated.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

REFERENCES

1. Min Xing, Siyuan Xiang and Lin Cai, "A Real-Time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks" in IEEE journal on selected areas Iin communications, Vol. 32, NO. 4, APRIL 2014 795
2. S. Xiang, L. Cai and J. Pan, "Adaptive scalable video streaming in wireless networks," in ACM MMSys'12, 2012, pp. 167–172.
3. T. Stockhammer, "Dynamic adaptive streaming over HTTP –: standards and design principles," in ACM MMSys'11, 2011, pp. 133–144.
4. S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," Signal Processing: Image Communication, vol. 27, no. 4, pp. 271–287, 2012.
5. R. Mok, X. Luo, E. Chan, and R. Chang, "QDASH: a QoE-aware DASH system," in ACM MMSys'12, 2012, pp. 11–22.
6. L. Cai, S. Xiang, Y. Luo, and J. Pan, "Scalable modulation for video transmission in wireless networks," IEEE Trans. Veh. Technol., vol. 60, no. 9, pp. 4314–23, 2011
7. M. Kobayashi, H. Nakayama, N. Ansari, and N. Kato, "Robust and efficient stream delivery for application layer multicasting in heterogeneous networks," IEEE Trans. Multimedia, vol. 11, no. 1, pp. 166–176, 2009.
8. D. Kaspar, K. Evensen, P. Engelstad, and A. Hansen, "Using HTTP pipelining to improve progressive download over multiple heterogeneous interfaces," in IEEE ICC'10, 2010, pp. 1–5.
9. K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. Hansen, and P. Engelstad, "Improving the performance of quality-adaptive video streaming over multiple heterogeneous access networks," in ACM MMSys' 11, 2011, pp. 57–69.
10. K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. F. Hansen, and P. Engelstad, "Using bandwidth aggregation to improve the performance of quality-adaptive streaming," Signal Processing: Image Communication, vol. 27, no. 4, pp. 312–328, 2012.
11. R. Sutton and A. Barto, Reinforcement learning: An introduction.Cambridge Univ Press, 1998, vol.
12. H. Wang and N. Moayeri, "Finite-state Markov channel-a useful model for radio communication channels," IEEE Trans. Veh. Technol., vol. 44, no. 1, pp. 163–171, 1995.
13. J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model 11 (JSVM 11)," Joint Video Team, Doc. JVT- X, 2007.