QoE Aware and Energy Efficient Online Scheduling for Video Streaming in Cellular Networks

Xiao Li, Zhilong Zhang, and Danpu Liu
Beijing Laboratory of Advanced Information Network
Beijing Key Laboratory of Network System Architecture and Convergence
Beijing University of Posts and Telecommunications, Beijing, P.R. China 100876
Email:{lx_2015, zhangzhilong, dpliu}@bupt.edu.cn

***Abstract*—In recent years, mobile video streaming is gaining overwhelming popularity and occupying an increasing proportion in mobile data traffic. Video streaming is sensitive to viewing interruptions and hungry for energy, so it is of great significance to mitigate the effect of viewing interruptions and reduce energy cost while enjoying video streaming services in cellular networks. In this paper, considering both channel quality and user buffer state, we propose a QoE aware and energy efficient online scheduling scheme based on Lyapunov optimization framework. In the scheme, a novel method by dynamically controlling the weighting value of energy consumption is introduced to obtain more energy saving and guarantee QoE performance. Simulation results demonstrate that our proposed scheme can reduce both rebuffering delay and energy consumption compared with the state-of-art method.**

I. Introduction

Video streaming services are becoming more and more popular in mobile networks with the rapid increase of hardware capability of mobile devices and the transmission bandwidth of cellular networks these years. According to Visual Networking Index [1] from CISCO, more than half of the mobile data traffic comes from video streaming services and it will increase to 78% by 2021. DASH is a promising video streaming technology and has attracted increasing attention. It is able to cope with the heterogeneities of mobile device and wireless network conditions and allows mobile viewers to select an appropriate video version according to network conditions and buffer states.

The majority of previous work is aiming at improving QoE metrics including how to reduce initial delay, rebuffering delay, the number of interruptions, etc [2]. However, energy cost should be also taken into account because video streaming is hungry for energy and battery capacity of mobile device is limited. In fact, the fraction of energy consumed by wireless interface occupies a large proportion in the total mobile energy consumption.

Energy consumed by wireless interface is composed of transmission energy and tail energy [3]. Transmission energy is used to receive data for mobile device and tail energy is the wasted energy which will be explained in detail in the following part. In typical cellular technologies (3G or LTE), there exists a radio resource control (RRC) protocol which claims that the radio of mobile device should stay in high power state until an inactivity timeout instead of switching to a low power state immediately when data transmission is completed [4]. If there is no transmission during that period, there will be a considerable part of energy wasted during that time which is also called tail time. The tail time is introduced to reduce the frequency of mode switching so that high signaling overhead could be avoided. However, once a large amount of tail time is introduced, a large amount of tail energy would be produced.

In consideration of such characteristics of energy consumption in mobile devices, significant efforts have been put into the research of reducing energy cost of mobile devices from different aspects. F. Qian et al. [5] applied traffic prediction and fast dormancy to optimize the inactivity timer to reduce tail energy. S.Herrerła-Alonso et al. [6] proposed an adaptive DRX scheme based on DRX technologies to improve energy efficiency with a bounded packet delay in LTE networks. These efforts are not targeting at streaming service. A.Schulman et al. [7] utilized a signal strength prediction method to prefetch video content when channel condition is good so that energy consumption for receiving data could be reduced. Matti Siekkinen et al. [8] used viewing statistics to predict user behavior so as to determine the chunk size for the purpose of cutting down the tail energy and traffic overhead.

However, traditional energy efficient methods based on prediction are not well adapted to the dynamical mobile environment. To break the bottleneck, recent efforts [9] [10] [11] have introduced Lyapunov optimization framework based on which a non-predication online scheduling scheme is always designed. SALSA [9] considers both delay and wireless link quality and designs a multi-interface online scheduler based on Lyapunov optimization without considering the impact of tail energy. Authors in [10] [11] add tail energy into consideration but don't take QoE metrics such as rebuffering delay or viewing interruptions into account. Neither of them considers resource competitions among users. The work [12] closest to our research proposes a scheduling framework with two complementary modes of which one is called EMA scheduling algorithm aiming at reducing energy consumption of video streaming users while guaranteeing their performance based on Lyapunov optimization. However, the rebuffering delay queue they construct for recursion relation cannot reflect the actual rebuffering delay and they use a static weighted value of energy consumption that is hard to work well in the complex

MOBIMEDIA 2017, July 13-14, Chongqing, People's Republic of China

DOI 10.4108/eai.13-7-2017.2270266

mobile environment.

Based on the previous work, we adopt Lyapunov framework and design an online scheduling scheme considering energy cost and users' QoE metrics for video streaming service in mobile network. Our contributions are twofold:

• We construct user buffer queue mechanism to reflect buffer state, based on which we utilize Lyapunov optimization to reduce energy consumption while keeping rebuffering delay in a bounded range.

• We design a dynamic method to control the weighted value of energy consumption to obtain more energy saving and guarantee QoE performance.

The remainder of the paper is organized as follows. Section II presents our system model and problem formulation. In Section III, we introduce our proposed online scheduling scheme embeded with two sub-algorithms and provide correlative analysis. We simulate our method in Section IV and conclude our work in Section V.

II. System Model

In this section, we elaborate the models of wireless transmission, energy consumption and performance metrics of video streaming over cellular networks.

A. Transmission Model

We assume that downlink traffic is scheduled to transmit in the unit of time slot that is composed of τ seconds. The LTE protocols provide that data is transmitted in frames. Data transmitted per frame is denoted as δ and it is related to signal strength. Then we define a parameter $\omega_i(n)$ to indicate the number of frames transmitted to user i in slot n. In LTE network, the total number of frames in a certain slot is limited and we cannot serve infinite users at the same time so we should satisfy the following restraint:

$$\sum_{i=1}^{N} \omega_i(n) \leqslant S, \tag{1}$$

where S is denoted as the number of frames in a slot and N is the number of users.

In addition, the length of users' buffers is always limited and data transmitted to users cannot exceed the limit. We utilize $V_i(n)$ to represent the maximum transmittable frames for user i in slot n due to buffer length limit, $V_i(n) = \left\lfloor \frac{(Buffer\_limit - buff_i(n)) \times rate_i(n)}{\delta(n)} \right\rfloor$, So $\omega_i(n)$ should satisfy another inequality as follows:

$$\omega_i(n) \leqslant V_i(n), \tag{2}$$

where $buff_i(n)$ and $rate_i(n)$ are indicated as the current buffering length and the current video bitrate respectively.

B. Energy Model

First we need to realize the entire energy consumption for mobile device is composed of transmission energy and tail energy. The transmission energy denoted as $E_i^{tran}(n)$ is consumed for receiving data , which can be expressed by

$$E_i^{tran}(n) = P_i(n) \times \tau, \tag{3}$$

from above we can see if we transmit as much data as possible when the channel quality is good then the transmission energy will be greatly reduced. However, if there is no data transmission, mobile device will suffer from tail energy consumption until the inactivity timeout. The tail energy is determined by tail time and power consumption of different modes [13]. In LTE networks, there are two kinds of modes in mobile device which are RRC_CONNECTED (high power state) and RRC_IDLE (low power state) respectively. In RRC_CONNECTED mode there are short DRX cycles and long DRX cycles and their timer periods are denoted as T1 and T2. Their average power consumption are expressed by p1 and p2 respectively. If there exists a long period when there is no data transmission mobile device will switch to RRC_IDLE state. We utilize Δt to represent the time interval between two transmission, and tail energy can be expressed as follows:

$$E_{tail}(\Delta t) = \begin{cases} p_1 \cdot \Delta t & 0 \leqslant \Delta t \leqslant T_1 \\ p_1 \cdot T_1 + p_2 \cdot (\Delta t - T_1) & T_1 < \Delta t \leqslant T_1 + T_2 \\ p_1 \cdot T_1 + p_2 \cdot T_2 & otherwise. \end{cases} \tag{4}$$

Combining equation (3) and (4), we can conclude the energy consumption of user i in slot n could be expressed as follows:

$$E_i(n) = \begin{cases} E_i^{tran}(n) & \omega_i(n) \neq 0 \\ E_i^{tail}(n) & \omega_i(n) = 0, \end{cases} \tag{5}$$

from above we can get the time-average energy consumption of all users in the following equation:

$$\widehat{PE(T)} = \frac{1}{T} \sum_{n=0}^{T-1} \sum_{i=1}^{N} E_i(n), \tag{6}$$

where T is denoted as the number of scheduling slots.

C. Performance Model

For video streaming service, we usually utilize quality of experience (QoE) metrics to evaluate performance of service, which are embodied by viewing interruptions and rebuffering delay etc. Rebuffering delay means the time period for resuming playback when the playback gets stuck. If we define $buff_i(n)$ as the current buffering length of user i in slot n, then rebuffering delay denoted as $rebuff_i(n)$ can be derived from

$$rebuff_i(n) = \max\{\tau - buff_i(n), 0\}. \tag{7}$$

If data in the buffer could not satisfy the need of user playback in a certain slot, viewing interruption event would take place and rebuffering delay would be produced. We define $\widehat{PR}$ as the average rebuffering delay of all users in the total scheduling period which can be expressed by

$$\widehat{PR(N)} = \frac{1}{N} \sum_{n=0}^{T-1} \sum_{i=1}^{N} rebuff_i(n). \tag{8}$$

III. QEOS Scheme

In this section, we introduce our QoE aware and energy efficient online scheduling (QEOS) scheme which is mainly composed of two sub-algorithms. First, we consider QoE requirements of users. We adopt rebuffering delay as the QoE metric which has been proved to be one of the most influential factors of user experience. As we discussed in the previous sections, viewing interruption event is highly related with the state of user buffer. Once the current buffering length is shorter than the length of a slot, viewing interruption event will occur and rebuffering delay is produced. The buffering length is changing all the time and the variation can be expressed by

$$buff_i(n+1) = buff_i(n) + t_i^{sc}(n) - \tau, \tag{9}$$

$$t_i^{sc}(n) = \frac{\omega_i(n) \times \delta}{rate_i(n)}, \tag{10}$$

where $rate_i(n)$ is the required video bitrate of user i in slot n and $t_i^{sc}(n)$ is denoted as playback time maintained by received data of user i in the slot n. The total resource is limited and there exist resource competitions among users. If too much resource is allocated to a certain user, its buffer will be adequate but other users may suffer from viewing interruption events. If we can ensure the buffering length of users stay stable, namely not too much or too little, we can achieve better overall QoE performance in multiple users scenario.

Based on Eq. (9) and Eq. (10), Lyapunov optimization framework is employed to control the state of user buffer. First we set α as the threshold of buffer. When the current buffering length is shorter than α, it indicates video player enters into a dangerous state of interruption. Then we define Q as a Lyapunov function factor which is equal to $buff - \alpha$. Namely Q can be regarded as equivalent buffer. From Eq. (9), we can get the recursion relation of Q as follows:

$$Q_i(n+1) = Q_i(n) + t_i^{sc}(n) - \tau. \tag{11}$$

Then we define a Lyapunov function $L(n)$ which represents a overall metric of queue congestion [14] for reflecting buffer states of users:

$$L(n) = \frac{1}{2}\sum_{i=1}^{N}(Q_i(n))^2. \tag{12}$$

As we discussed above, in order to guarantee QoE performance we need to keep the queues stable by pushing the Lyapunov function towards a lower congestion state.

Next we introduce Lyapunov drift $\Delta(L)$ as follows:

$$\begin{aligned}\Delta(L) &= \boldsymbol{E}\{L(n+1) - L(n) \mid \boldsymbol{Q(n)}\} \\ &\leqslant B + \sum_{i=1}^{N}\boldsymbol{E}\{Q_i(n) \times (t_i^{sc} - \tau) \mid \boldsymbol{Q(n)}\},\end{aligned} \tag{13}$$

where $\boldsymbol{Q(n)}$ represents the vector $(Q_1(n), ..., Q_N(n))$ and

$$B = \frac{1}{2}\sum_{i=1}^{N}(\tau^2 + (t_{max}^{sc})^2), \tag{14}$$

where t_{max}^{sc} is denoted as the maximum playback time maintained by any user in a slot. Referring to the Lyapunov optimization approach [14], we take both performance metrics and energy consumption into consideration, which leads to *drift-plus-penalty* term below:

$$\Delta(L) + \rho \cdot \boldsymbol{E}\{E(n) \mid \boldsymbol{Q(n)}\}, \tag{15}$$

where $E(n) = \sum_{i=1}^{N} E_i(n)$ is described as the total energy consumption of all users and $\boldsymbol{E}\{E(n) \mid \boldsymbol{Q(n)}\}$ represents the average energy consumption. The parameter ρ is a weighted factor which is used to balance $\Delta(L)$ and energy consumption. Combining Eq. (13) with Eq. (15), we can deduce the upper bound of the *drift-plus-penalty* problem into

$$\begin{aligned}&\Delta(L) + \rho \cdot \boldsymbol{E}\{E(n) \mid \boldsymbol{Q(n)}\} \\ \leqslant &B + \sum_{i=1}^{N}\boldsymbol{E}\{Q_i(n) \times (t_i^{sc}(n) - \tau) \mid \boldsymbol{Q(n)}\} \\ &+ \rho \cdot \boldsymbol{E}\{E(n) \mid \boldsymbol{Q(n)}\}.\end{aligned} \tag{16}$$

According to Lyapunov optimization approach, in order to minimize the energy consumption with performance guaranteed, we minimize the upper bound of *drift-plus-penalty* problem. Since B and τ are constants, the problem Eq. (16) can be transformed into

$$\begin{aligned}&\min \rho \cdot E(n) + \sum_{i=1}^{N}\{Q_i(n) \times (t_i^{sc} - \tau)\} \\ \triangleq &\min \sum_{i=1}^{N}\{\rho \cdot E_i(n) + Q_i(n) \times (t_i^{sc}(n) - \tau)\} \\ \triangleq &\min \sum_{i=1}^{N} F(i, \omega_i(n)) \\ &s.t. (1)\&(2).\end{aligned} \tag{17}$$

From Eq. (17), if $Q_i(n)$ is negative, which means the buffer of user i enters into the risk area of interruption, we should allocate more resource to the corresponding user to make $t_i^{sc}(n) - \tau$ become a positive value. However, if the user buffer is sufficient, which is embodied by a great positive value of $Q_i(n)$, then we can schedule less resource to the corresponding user in that slot to keep $t_i^{sc}(n) - \tau$ negative. Thus we can wait for a better chance to transmit more data in the latter case because energy could be saved if data were transmitted in good channel quality. Our aim is to find the optimal $\omega_i(n)$ in each slot. We can adopt dynamic programming method to solve the minimization problem. We define C as a two-dimensional array which indicates the minimal accumulation of $F(i, \omega_i(n))$. Specifically, $C[i][M]$ is denoted as the sum of $F(i, \omega_i(n))$ of the previous i users when we decide to schedule M data units to them. Hence $C[i][M]$ can be derived from the following formula:

$$C[i][M] = \min\{C[i-1][M - \omega_i(n)] + F(i, \omega_i(n))\}, \tag{18}$$

where M is the number of data units scheduled to the previous i users and $\omega_i(n)$ is the number of data units we decide to

schedule to user i in slot n. Based on Eq. (17) and Eq. (18), we design an online resource allocation algorithm (ORA) to find the solution of problem (17).

Algorithm 1 ORA

Input: User number N, Slot length τ, The number of frames per slot S, The transmittable data per frame $\delta(n)$, The current buffer length $buff_i(n)$, Required video encoding rate $rate_i(n)$, Lyapunov parameter ρ.

Output: Resource allocation $\omega_i(n)$, $i \in [1, N]$

1: Update $Q_i(n)$ by Eq. (11)
2: Initiate $\omega_i(n) : \omega_i(n) \leftarrow 0, i \in [1, N]$
3: **for** $M = 0 \rightarrow V_1(n)$ **do**
4: $C[1][M] = F(1, M)$
5: $Bd[1][M] = M$
6: **end for**
7: **for** $i = 2 \rightarrow N$ **do**
8: **for** $M = 0 \rightarrow S$ **do**
9: **for** $m = 0 \rightarrow \min\{V_i(n), M\}$ **do**
10: $C[i][M] = \min\{C[i-1][M-m] + F(i, m)\}$
11: **end for**
12: $Bd[i][M] = arg\min_m C[i][M]$
13: **end for**
14: **end for**
15: $Z_N = arg\min_M C[N][M], \omega_N(n) = Bd[N][Z_N]$
16: **for** $i = N - 1 \rightarrow 1$ **do**
17: $Z_i = Z_{i+1} - \omega_{i+1}(n), \omega_i(n) = Bd[i][Z_i]$
18: **end for**
19: **return** $\omega_i(n), i \in [1, N]$

As algorithm 1 illustrates, ORA first updates the queue Q and initiates the allocated resource set to zeros (steps 1-2). By solving the dynamic programming problem (steps 3-14), we obtain all the possible resource allocation traces. Then we select the best trace and use iteration method to obtain the allocated resource set (steps 15-19).

In addition, according to Lyapunov optimization principles [14], for any energy consumption weight $\rho > 0$, if we increases ρ, more energy will be saved. However, performance cost is increasing at the same time. So for any performance cost constraint Ω, if we can find a certain ρ which ensures performance cost close to but within Ω, then we can obtain more energy saving while satisfying performance requirements.

Traditionally, we always choose a static ρ based on some heuristic information but it is hard to work well in the long run that we apply these traditional ways in a complex mobile environment. As a result, we refer to the congestion avoidance algorithm derived from TCP protocol and design a dynamic scheme called D-ρ-A (dynamic ρ algorithm). We increase ρ by σ gradually as long as performance cost does not exceed the constraint so that we can obtain more energy saving. Once the performance cost constraint is not met, we cut ρ to a half to satisfy the performance requirements. We calculate average performance cost $\bar{R}$ every Γ seconds. The work [11] has utilized this method in Lyapunov optimization problem and proved its high convergence speed but it is not targeting at video streaming service. Here we introduce this method into QEOS to obtain more energy saving guarantee QoE performance for video streaming service and the results demonstrate the effectiveness of this method.

Algorithm 2 D-ρ-A

Input: Ω, n, Γ

Output: ρ

if n mod Γ equals 0 **then**
  caculate $\bar{R}$, $\bar{R} = \frac{1}{N \cdot n} \sum_{k=0}^{n-1} \sum_{i=1}^{N} rebuff_i(k)$
  if $\bar{R} < \Omega$ **then**
    $\rho = \rho + \sigma$
  else
    $\rho = \rho / 2$
  end if
else
  ρ keeps the same.
end if
Call EERA to get $\omega_i(n)$

IV. Simulations

A. Parameters Setting

Our algorithms are implemented in Matlab simulation environment. The total simulation time T is set to 1000 seconds and each slot τ lasts for one second. We consider a circular area with a radius of 500m where a BS is located in the center. The transmit power and bandwidth of the BS are set to 46dBm and 5MHz respectively. The corresponding path loss is $L(d) = 34 + 40log(d)$ where d is the distance between UE and BS. The lognormal shadowing with a standard deviation is set to 8dB. The noise power is assumed as -106dBm. We assume UE is moving and the distance d follows a sine function fluctuating within 500m. Since the initial position of UEs may be different so we add different initial phases. The performance cost constraint Ω is 0.1 and its statistic period Γ is set to 50 seconds. The number of users is 50 and the number of frames S is set to 100 for all the slots according to LTE protocols. As for user requested video bitrate in each slot, we introduce buffer-based bitrate adaptation mechanism and UE would select bitrate adaptively according to the current buffering length. The power consumption of UE is set to 1680mW and the average power consumption for short DRX cycles (p_1) and long DRX cycles (p_2) are set to 1091.0mW and 1075.5mW respectively. The corresponding timer periods T_1 and T_2 are set to 3.82s and 7.64s.

B. Simulation Results

In this subsection we simulate our proposed QEOS scheme. For comparison, we implement the state-of-art energy efficient scheduling algorithm called EMA [12] which is also targeting at video streaming service and has proved its advantage over other non-prediction online scheduling algorithms such as SALSA [9] and EStreamer [15]. Hence we only give the comparison between QEOS and EMA in our simulation. In

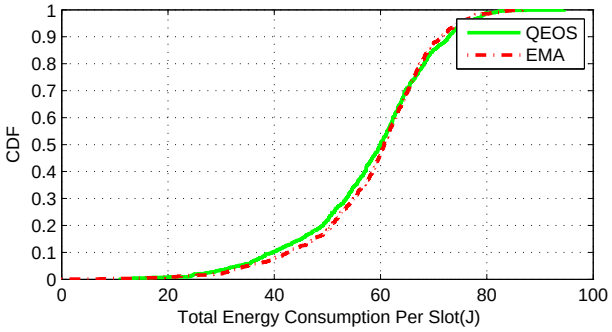

Fig. 1. Energy consumption

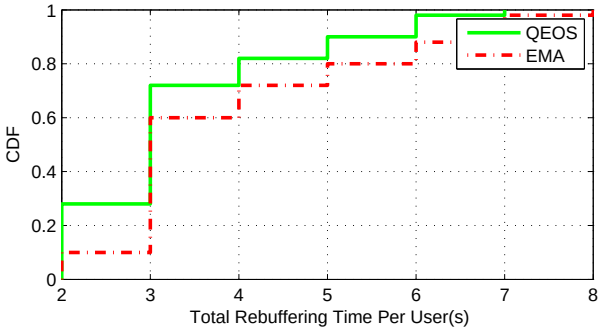

Fig. 2. Rebuffering delay

the following analysis we will present the comparison results mainly from two aspects: energy consumption and rebuffering delay.

As shown in Figure 1, QEOS can be more energy efficient than EMA because QEOS takes both buffer state and channel quality into account. It would wait for better channel condition to transmit data as long as user buffer is not in the risk area of interruptions. From Figure 2, we can prove that QEOS outperforms EMA in rebuffering delay because QEOS would not allocate too much resource to a user when the user buffer is adequate. Instead, it would spare that part of resource to users whose buffers are in the risk area so that we could achieve better overall performance. Meanwhile, the dynamic weighted value mehtod embedded in QEOS has played an important role in obtaining more energy saving and guaranteeing QoE performance. We have also calculated the average energy consumption ($\widehat{PE(T)}$) and rebuffering delay ($\widehat{PR(N)}$) under the same simulation conditions and find that QEOS could reduce energy consumption and rebuffering delay by 1.2% and 15.8% respectively.

V. Conclusion

In this paper, we propose an QoE aware and energy efficient online scheduling (QEOS) scheme for video streaming service which embeds two sub-algorithms including ORA and D-ρ-A. Different from existing work, our focuses lie in how to keep user buffer staying in a safe area in multiple users scenario, and how to allocate wireless resource to save energy and enhance performance. Simulation results show that our proposed algorithm could save more energy while also reducing rebuffering delay in cellular networks.

Acknowledgment

This work was funded by the NSFC under Grant 61271257, 61171107, and supported by the Fundamental Research Funds for the Central Universities.

References

[1] Cisco visual networking index: Global Mobile Data Traffic Forecast Update, 2016-2021 White Paper, 2017.
[2] Vinay Joseph, and G. De Veciana. "NOVA: QoE-driven Optimization of DASH-based Video Delivery in Networks." INFOCOM, 2014 Proceedings IEEE IEEE, 2013:82-90.
[3] A. Pathak, Y. Hu, and M. Zhang, Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof, in Prof. of ACM EuroSys, 2012.
[4] 3GPP, "System impact of poor proprietary fast dormancy," 3GPP discussion and decision notes RP-090941, 2009.
[5] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Top: Tail optimization protocol for cellular radio resource allocation," in IEEE International Conference on Network Protocols (ICNP) 2010.
[6] S.Herrerła-Alonso, M.Rodrłguez-Prez, M.Fernndez-Veiga and C.Lpez-Garcła. "Adaptive DRX scheme to improve energy efficiency in LTE networks with bounded delay," IEEE Journal on Selected Areas in Communications, vol.33, no.12, pp.2963-2973, 2015.
[7] A.Schulman et al. Bartendr: a practical approach to energy-aware cellular data scheduling. In MOBICOM, 2010.
[8] Matti SiekkinenMohammad Ashraful Hoque and Jukka K. Nurminen. "Using viewing statistics to control energy and traffic overhead in mobile video streaming, "IEEE/ACM Transaction on Networking, vol.24, no.3, pp.1489-1503, 2015.
[9] M.Ra et al. Energy-delay tradeoffs in smartphone applications. In MOBISYS, 2010.
[10] P.Shu et al. etime: energy-efficient transmission between cloud and mobile devices. In INFOCOM, 2013.
[11] Y.Cui et al. Performance-aware energy optimization on mobile devices in cellular network. In INFOCOM, 2014.
[12] Zeqi Lai et al. Joint Media Streaming Optimization of Energy and Rebuffering Time in Cellular Networks. In ICPP, 2015.
[13] N.Balasubramanian et al. Energy consumption in mobile phones: a measurement study and implications for network applications. In IMC, 2009.
[14] M. Neely, "Stochastic network optimization with application to communication and queueing systems," Morgan & Claypool Publishers, 2010.
[15] M.Hoque et al. Saving energy in mobile devices for on-demand multimedia streaming-a cross-layer approach. ACM TOMCCAP, 2014.