Impact of Traffic Burstiness on Optimal Batching Policy for Energy-Efficient Video-on-Demand Services

Xiaolei Wang[1], Yanan Bao[2], Congshi Hu[1], Sheng Zhou[1], and Zhisheng Niu[1]
[1]Tsinghua National Laboratory for Information Science and Technology,
Dept. of Electronic Engineering, Tsinghua Univ., Beijing 100084, P.R. China
[2]Dept. of Computer Science, University of California, Davis 95616, USA
Email: wang-xl09@mails.tsinghua.edu.cn

***Abstract*—In this paper, we consider the impact of traffic burstiness on optimal batching policy for energy-efficient Video-on-Demand (NVoD) services. By introducing batching technology, multiple users can be served by one multicast transmission. The more users in one multicast transmission, the less transmissions are needed, which induces less channel occupancy and energy consumption. However, to expect more users coming leads to longer queueing time. Hence, there is a trade-off between the number of transmissions and the average queueing time of users. We prove that N-Policy is optimal among all feasible policies when the arrivals of users follow a Poisson process. When the inter-arrival time of users follows Gamma distribution, i.e., the arrival process is smooth, it is shown that N-Policy is still optimal. But, when the inter-arrival time of users follows hyper-exponential distribution, i.e., the arrival process is bursty, the N-policy is no more optimal and the energy cost can be further reduced by a Generalized Impatient (GI) policy, especially when the coefficient of variance is large.**

I. Introduction

Recent years have witnessed rapid progress in providing video services throughout all kinds of networks, such as Internet, mobile communication networks and digital television networks. Since video services normally require high transmission rate and long channel occupation time, they are now the major resource consumer in today's networks. Meanwhile, the bandwidth resources in wireless communication are highly limited, which challenges the capacity and energy efficiency of mobile networks. To utilize the advantage of broadcast nature of the wireless medium, the industry standard MBMS (Multimedia Broadcast Multicast Service) [1] and BCMCS (Broadcast and Multicast Service) [2] have been proposed to realize multimedia transmission in mobile communication networks.

Batching is a technique to collect enough requests to multicast. In the batching scenario, the first user who requests a video downloading triggers one multicast session. Instead of starting the transmission immediately, the server postpones the service and waits for potential users who request the same content. This period is called batching window. In the end of the batching window, if some condition is satisfied (e.g. N requests have arrived or after time T), the multicast session is set up by the server to transmit one video stream to multiple users.

Intuitively, the longer the batching window is, the more users can be served together by one multicast, which means more energy efficiency and spectrum efficiency. However, this will cause longer delay. Therefore, there is a trade-off between resource efficiency and average delay. Traditionally, this trade-off is studied by analysing heuristic policies [3]. However, how to decide when to start the multicast session? Are there any policies which are optimal in terms of the trade-off?

In this work, we propose a T-Policy that waits for a certain time, N-Policy that waits for a certain number of requests, and a Generalized Impatient (GI) policy stops waiting if no arrival comes within a period of time. We try to give the optimal solutions along with the performance analysis when the burstiness of the user arrivals varies. We show that the efficiency of multicast can be increased when the burstiness increases.

The contributions of this paper include:

- We prove the optimality of N-Policy when the requests arrive as a Poisson process;
- We give a way to find the optimal numerical solution when the requests arrive as a renewal process;
- We compare different heuristic policies and show that N-Policy is optimal for smooth arrivals, and GI-Policies can achieve higher efficiency for bursty arrivals;
- We show the impacts of traffic burtiness on the efficiency of multicast, i.e., the average number of requests gathered by one multicast session increases with burstiness with the same average waiting time constraint.

II. System Model and Batching Policies

In this paper, we consider a base station (BS) that provides near Video-on-Demand (NVoD) services, and serves multiple users requesting videos successively. If sufficient users request the same video simultaneously, the BS can multicast the video to the users thus reduce the bandwidth occupation and energy consumption. In order to brought more chance to multicast, we assume the user can tolerant some delay. Accordingly, the

This work is sponsored in part by the National Basic Research Program of China (973 Program: 2012CB316001), the National Science Foundation of China (NSFC) under grant No. 61201191, the Creative Research Groups of NSFC under grant No. 61321061, and Hitachi R&D Headquarter.

BS can collect more users to serve by one multicast session, i.e., the multicast is more efficient.

We define the "service rate" of a video as the average number of multicast sessions for the video per unit time. From the aspect of BSs, users may request a large amount of videos. If we assume that the number of videos is sufficiently large and the arriving processes of the requests for different videos are independent, the overall process of multicast sessions can be considered as a Poisson process. The service rate of the overall process is the summation of all the service rates for each video. Therefore, if we minimize the service rate of each video, we can minimize the total number of multicast sessions for NVoD service, which leads to lower energy consumption and bandwidth requirement in terms of higher efficiency.

In practice, multicast causes extra cost such as overhead and higher transmitting power. As a result, if we cannot collect enough users within a batching window, we should unicast the content to the users. In a typical scenario, we should use unicast if we collect less than 5 users [7], [8]. In this paper, we assume all transmissions are multicasts and investigate how to collect more users in a batching window with the average waiting time constraint. In the future work, we can extend our result to the scenario that unicast and multicast coexist.

The problem can be modeled as a special queueing system. Here we only consider the requests for one video. We assume the arrivals of requests follow a renewal process, which means that the interval between two successive requests for one video are independent and identically distributed. In contrast with conventional queueing model, we assume the server serves all the users waiting in the queue at the time specified by the policy. Since when the multicast session starts, all the customers waiting in the queue can be served instantly at the same time with a constant cost in total. We need to derive a policy to determine when to serve the customers.

The performance of a policy is measured by two parameters: the average waiting time and the service rate. Intuitively, the more often to serve the users waiting in the queue, the less time customers have to wait. The objective is to minimize the service rate while keeping the average waiting time under a certain value.

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, P)$ be the underlying probability space. The arriving time epoch of the i-th request is denoted by τ_i^{a}. The arrival process of the requests can be formulated by a counting process:

$$A(t) \triangleq |\{\tau_i^{\mathrm{a}} | 0 < \tau_i^{\mathrm{a}} < t\}| \tag{1}$$

that represents the number of arrivals from time 0 to time t. The departing time epoch of the i-th request τ_i^{d} and the departure process $D(t)$ is defined similarly:

$$D(t) \triangleq \left|\left\{\tau_i^{\mathrm{d}} | 0 < \tau_i^{\mathrm{d}} < t\right\}\right|. \tag{2}$$

Let λ denote the arrival rate:

$$\lambda = \lim_{k\to\infty} \frac{k}{\tau_k^{\mathrm{a}}}. \tag{3}$$

Let τ_j^{s} denote the service time epoch of the j-th multicast session. We assume the policy could only take the causal information, i.e., τ_j^{s} is stopping time of $(\mathcal{F}_t)_{t\geq 0}$. All the users arriving after time τ_{j-1}^{s} and before time τ_j^{s} is served by the j-th multicast session. Accordingly, τ_i^{d} can be determined by the sequence of τ_j^{s}:

$$\tau_i^{\mathrm{d}} = \min_j \left\{\tau_j^{\mathrm{s}} | \tau_j^{\mathrm{s}} \geq \tau_i^{\mathrm{a}}\right\}. \tag{4}$$

Let batch size b_j denote the number of users being served by the j-th multicast session. The batch size is equal to the number of requests arrived between τ_{j-1}^{s} and τ_j^{s}:

$$b_i \triangleq \left|\left\{\tau_i^{\mathrm{a}} | \tau_{j-1}^{\mathrm{s}} < \tau_i^{\mathrm{a}} \leq \tau_j^{\mathrm{s}}\right\}\right|. \tag{5}$$

The average batch size denoted by $\overline{b}$ is:

$$\overline{b} = \lim_{k\to\infty} k^{-1} \sum_{i=1}^{k} b_i. \tag{6}$$

Let $\overline{D}$ denote the average waiting time in terms of the average delay caused by batching:

$$\overline{D} \quad = \quad \lim_{k\to\infty} k^{-1} \sum_{j=1}^{k} (\tau_j^{\mathrm{d}} - \tau_j^{\mathrm{a}}). \tag{7}$$

The average service rate denoted by $\overline{C}$ indicates how often the server starts a multicast session:

$$\overline{C} \quad = \quad \lim_{k\to\infty} \frac{k}{\tau_k^{\mathrm{s}}}. \tag{8}$$

From (3) (6) (8), we have:

$$\overline{C} \quad = \quad \lambda/\overline{b}. \tag{9}$$

The objective of the problem is to minimize the average service rate $\overline{C}$ that corresponds to the rate of resource consumption, e.g. bandwidth occupation or energy consumption. From (9), to minimize the average service rate is to maximize the average batching size in terms of the efficiency of multicasts. On the other hand, lower service rate may cause larger delay. As a result, we should guarantee that the average delay for the batching users is within a reasonable threshold D_{th}. The optimization problem can be formulated by:

$$\begin{aligned} \min \quad & \overline{C} \\ \text{s.t.} \quad & \overline{D} \leq D_{\mathrm{th}}. \end{aligned} \tag{10}$$

In this paper, we define the *feasible policy* as follows:

Definition 1. A *feasible point* is a way to determine when to start a multicast session. By this way, the service rate is equal to C and the average waiting time is equal to D. Thus a feasible point corresponds to a pair of real number (D, C). A feasible point should also satisfies:

1) The policy only takes the information no later than time t into account.

2) The policy is the same for each multicast session. In other words, the arrival and departure processes follow the same distribution after the first request of each multicast session.

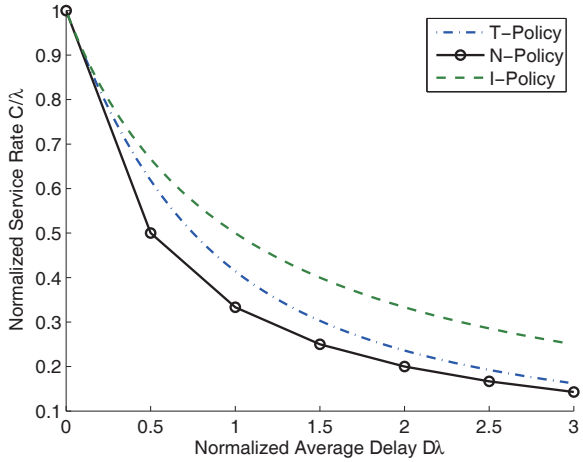

Fig. 1. The resource-delay trade-off of T-Policy, N-Policy, and I-Policy when the users arrive as a Poisson process.

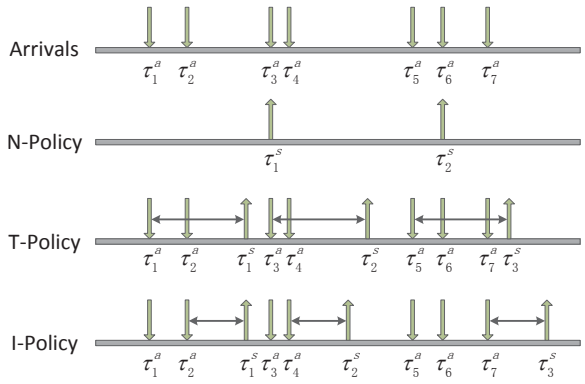

Fig. 2. An illustration of N-Policy, T-Policy and I-Policy. The N-Policy always waits for N users, here $N = 3$. The T-Policy always waits for time T since the first arrival. The I-Policy waits for time T since the last arrival.

3) The policy makes the mean and the second moment of batch size b_i converge, i.e.,

$$\bar{b} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} b_i < \infty \tag{11}$$

$$v = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} b_i^2 < \infty. \tag{12}$$

Definition 2. For any given $D_{\text{th}} > 0$, a *feasible policy* should give a feasible point corresponding to (D, C) that satisfies $D \le D_{\text{th}}$.

A feasible point corresponds to (D, C) can be presented on a performance trade-off graph such as Fig. 1. A policy thus can be presented as a curve comprising the feasible points given by all $D_{\text{th}} > 0$. We derive three heuristic policies: T-Policy, N-Policy and I-Policy illustrated by Fig. 2, and give the formulation of $(\overline{D}, \overline{C})$ curves when the arrivals follow a Poisson process.

- A policy that waits for time T after the first arrival to an empty queue before the service is called a T-Policy with parameter T. The average waiting time $D_T(T)$ and the service rate $C_T(T)$ are given by:

$$D_T(T) = \frac{T(\lambda T + 2)}{2(\lambda T + 1)} \tag{13}$$

$$C_T(T) = \frac{1}{T + \lambda^{-1}}. \tag{14}$$

With given D_{th}, the parameter T is solved from (13):

$$T = \frac{(\lambda D_{\text{th}} - 1) + \sqrt{(\lambda D_{\text{th}} - 1)^2 + 2\lambda D_{\text{th}}}}{\lambda}. \tag{15}$$

- A policy that always waits for and serves N_w customers is called an N-Policy with parameter N_w. The average waiting time $D_N(N_w)$ and the service rate $C_N(N_w)$ are given by:

$$D_N(N) = \frac{(N-1)\lambda^{-1}}{2} \tag{16}$$

$$C_N(N) = \frac{\lambda}{N}. \tag{17}$$

When $2\lambda D_{\text{th}} + 1$ is an integer, we take the parameter $N = 2\lambda D_{\text{th}} + 1$. When $2\lambda D_{\text{th}} + 1$ is not an integer, let $\hat{N} = \lfloor 2\lambda D_{\text{th}} + 1 \rfloor$ be the integer part and α be the decimal part. For each multicast session, we randomly take $N = \hat{N}$ with probability

$$p = \frac{\alpha(N+1)}{\alpha N + (1-\alpha)(N+1)} \tag{18}$$

and $N = \hat{N} + 1$ with probability $(1 - p)$.

- An I-Policy in terms of *impatient policy* with parameter T waits for time T and starts the service if there is no arrival during the time. Otherwise, it waits for another period of time T. In other words, if the inter-arrival time between the i-th and $(i+1)$-th customers $\tau_{i+1}^{\text{a}} - \tau_i^{\text{a}}$ is larger than T, the service will start at $\tau_i^{\text{a}} + T$.

$$D_I(T) = \frac{1 - e^{-\lambda T}}{\lambda e^{-\lambda T}} \tag{19}$$

$$C_I(T) = \lambda e^{-\lambda T}. \tag{20}$$

The parameter T can be solved from (19):

$$T = \lambda^{-1} \ln(\lambda D_{\text{th}} + 1) \tag{21}$$

- A generalized impatient policies (GI-Policies) is a series of generalized policies. The parameter of feasible points given by GI-Policies is a series $\{T_i\}_{i\ge 1}, T_i \in \mathbb{R}^+ \cup \{0, +\infty\}$. After the i-th request in a multicast session arrives, it waits for time T_i and starts the service if no request arrives within T_i. Otherwise, it waits for another T_{i+1} after the arrival of the $(i+1)$-th request. In other words, if the inter-arrival time between the i-th and $(i+1)$-th customers $\tau_{i+i}^{\text{a}} - \tau_i^{\text{a}}$ is larger than T_i, the service will start at $\tau_i^{\text{a}} + T_i$. Specifically, if $T_i = +\infty$, it always

waits for the $(i+1)$-th arrival; if $T_i = 0$, it waits at most i requests.

Remark 1. The N-Policy is a special case of GI-Policies by taking the parameter $T_i = +\infty$ for $i < N_w$, and $T_i = 0$ for $i \geq N_w$.

Remark 2. When the arrival rate λ varies, the shape of the trade-off graph stays the same except that $\overline{D}$ scales with λ^{-1} and $\overline{C}$ scales with λ. Therefore on Fig. 2, the x axis is $\lambda\overline{D}$ and the y axis is $\overline{C}/\lambda$, thus the graph remains the same for arbitrary $\lambda > 0$.

Remark 3. The GI-Policies give a structure of policies that does not specify how to choose T_i. However, in the next section, we prove that the optimal solution can be achieved by GI-Policy. As a result, although we cannot give the closed-form solution of the optimal solution for generalized cases, we can find out the optimal trade-off curve by numerical studies.

III. Optimal Policies for Poisson, Smooth and Bursty Arrivals

In this section, we assume the arrivals of the requests are Poisson, smooth and bursty respectively, and discuss the influence of the burstiness on the optimal policy.

A policy is said to be optimal if it gives the feasible point with the minimum C while $D \leq D_{\text{th}}$ for any $D_{\text{th}} > 0$, i.e., it is the solution of problem (10). Illustrated on the trade-graph, all the feasible points should be on the upper-right of the curve corresponds to the optimal policy.

A. Poisson Arrivals

Here we assume the arrival process of requests is a Poisson process. We propose the following theorem to show the optimal $\overline{D}$ and $\overline{C}$ trade-off. According to Definition 2, the heuristic policies in Section II are all feasible policies. We claim that N-Policy is optimal since its performance is the lower bound of average service rate on the resource-delay trade-off curve in Fig. 1. This is supported by the following theorem:

Theorem 1. *If the arrival process is a Poisson process, under any feasible policy, the average waiting time $\overline{D}$ and the service rate $\overline{C}$ satisfies:*

$$\overline{D} + \frac{\hat{b}^2 + \hat{b}}{2\lambda^2}\overline{C} \geq \frac{\hat{b}}{\lambda}, \tag{22}$$

where $\overline{C}$ equals to $\frac{\lambda}{\hat{b}}$, $\hat{b}$ denotes the integer part of $\overline{b}$:

$$\hat{b} = \lfloor \lambda/\overline{C} \rfloor. \tag{23}$$

The proof of this theorem is omitted here due to the space limitation, and will be included in the journal version.

From Theorem 1, if $\overline{b}$ is an integer, $\overline{C} = \lambda/\overline{b}$, $\overline{D} \geq (\overline{b} - a)\lambda^{-1}/2$. If $\overline{b}$ is not an integer, the lower bound of $\overline{D}$ is the linear combination of successive integer points. As a result, the optimal service rate and average waiting time trade-off curve is given as the solid line on Fig. 1 which corresponds to N-Policy.

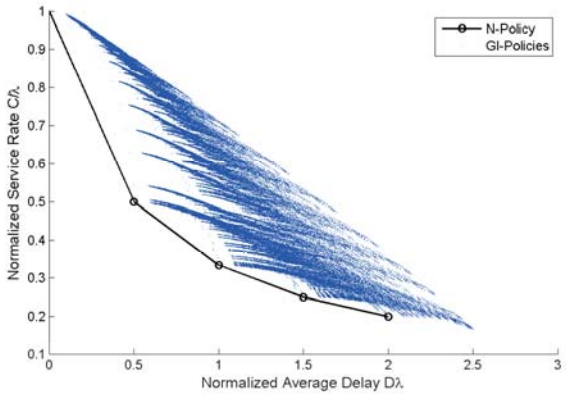

Fig. 3. Brute force search result for different parameters of GI-Policies when the inter-arrival time of smooth requests follows Gamma distribution $\Gamma(5, 0.2)$. From theorem 2, all the feasible points are linear combinations of feasible points in GI-Policies. Therefore N-Policy is still optimal.

B. Smooth and Bursty Arrivals

In this subsection, we assume the arrivals follow a renewal process. We will show the numerical results under two cases where the inter-arrival time follows gamma distribution and hyper-exponential distribution that denotes smooth and bursty arrivals, respectively.

In contrary to the Poisson arrivals, we cannot prove that N-Policy is optimal, nor find out a closed-form solution of the optimal policy. However, we can prove that any feasible point is a linear combination of feasible points in GI-Policies:

Theorem 2. *For any feasible point, the corresponding (D, C) is a linear combination of the feasible points in GI-Policies.*

The proof of this theorem is omitted here due to the space limitation, and will be included in the journal version.

From Theorem 2, we can find out the optimal trade-off curve by brute-force searching for the lower bound of feasible points in GI-Policies.

First, we assume that the intervals between users' arrivals follow Gamma distribution $\Gamma(k, 1/(k\lambda))$. This process is used to denote a smooth arrival process since the coefficient of variance is $1/k < 1$.

Fig. 3 shows the result of brute force search for different parameters of GI-Policies with $k = 5$. All the feasible points of GI-Policies are not lower than N-Policy, thus N-Policy is still optimal. The reason will be discussed in the next subsection.

Then we consider bursty arrivals. We assume the inter-arrival time follows hyper-exponential distribution:

$$F(t) = 1 - \sum_{i=1}^{r} \alpha_i e^{-\lambda_i t} \tag{24}$$

Fig. 4 shows the result of brute force search. Here $\alpha_1 = \alpha_2 = 0.5$, $\lambda_1 = 19.48$, $\lambda_2 = 0.5132$. The result shows that N-Policy is no longer optimal, i.e., the service rate can be significantly increased.

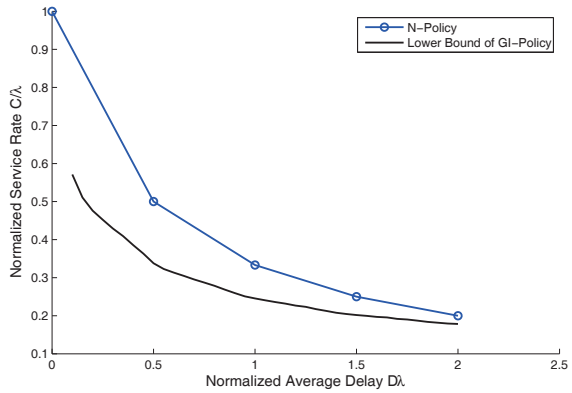

Fig. 4. Brute force search result for different parameters of GI-Policies when the inter-arrival time of bursty requests follows hyper-exponential distribution. The efficiency of Multicast can be significantly increased when the burstiness is large.

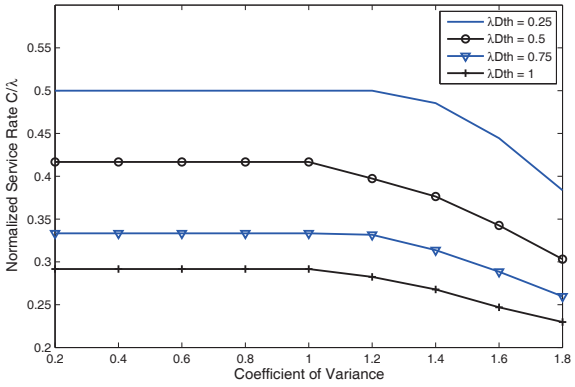

Fig. 5. The impact of burstiness on the efficiency of multicast. When the arrivals are bursty, more user will be served by one multicast session.

The burstiness of requests' arrivals can be denoted by the coefficient of variance of the inter-arrival time. Fig. 5 shows that, as the burstiness varies, the optimal service rate varies with the same average delay tolerance. For smooth arrivals, we should always wait for a constant number of requests, and the service rate remains the same. However, as the burstiness increases, the service rate reduces, which leads the efficiency of multicast increases significantly.

C. Discussions

In this subsection, we discuss the results shown in previous sections. Instead of mathematical rigour and proof, we qualitatively discuss the physical meaning of previous results.

When the arrival process is Poisson, due to the memoryless feature, the probability of future arrivals within a unit time remains the same at any time. From micro perspective, if there are n requests waiting in the queue, the probability that new request arrives within time $\mathrm{d}t$ stays $\lambda \mathrm{d}t$. Instead of immediately starting a multicast session, waiting for another $\mathrm{d}t$ increase the total waiting time $n\mathrm{d}t$. Therefore, for Poisson arrivals, whether to start transmission or not is determined by the number of requests waiting in the queue, which implies that N-Policy is optimal.

For the smooth case, the probability of future arrival within time $\mathrm{d}t$ grows with the time since the previous request arrives. Accordingly, the benefit of waiting grows until the next arrival while the cost in terms of increment of total waiting time remains a constant growth rate $n\mathrm{d}t$. If we should wait right after the n-th arrival of a multicast session, we should wait till the next arrival. As a result, N-Policy remains optimal. In contrast, for the bursty case, the probability of future arrival decreases with the time since the previous request arrives. If it has been a sufficient long time since the previous request arrived, the probability is small enough for us to stop waiting and start the multicast immediately.

IV. Conclusion

In this paper, we investigated the impact of traffic burstiness on a batching optimization problem. We first give the optimal policy when the requests follow Poisson process. Then, we derive the structure of optimal policy for smooth and bursty arrivals that follow a renewal process. The results of numerical studies show that N-Policy is optimal for smooth and Poisson arrivals. For bursty arrivals, the N-Policy is not optimal. Instead of always waiting for a constant number of requests, the optimal policy should be "impatient". Furthermore, the efficiency of multicast grows significantly with the burstiness of requests' arrivals.

References

[1] 3GPP TS 23.246, *Multimedia Broadcast/Multicast Service (MBMS) Architecture and functional description*.

[2] 3GPP2 X S022-A v1.0, *Broadcast and multicast service in cdma2000 wireless IP network, revision A*.

[3] Yanan Bao, Xiaolei Wang, Sheng Zhou, and Zhisheng Niu, "Energy-efficient multicast with deadlines in wireless networks via lazy rate scheduling," *2012 1st IEEE International Conference on Communications in China (ICCC)*, pp.393-398, Aug. 2012.

[4] Asit Dan, Dinkar Sitaram, and Perwez Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia systems*, vol. 4, no. 3, pp. 112-121, 1996.

[5] Charu C. Aggarwal, Joel L. Wolf, and Philip S. Yu, "On optimal batching policies for video-on-demand storage servers," *3rd IEEE International Conference on Multimedia Computing and Systems*, pp. 253-258, 1996.

[6] Simon Sheu, Kien A. Hua, and Wallapak Tavanapong, "Chaining: A generalized batching technique for video-on-demand systems," *IEEE International Conference on Multimedia Computing and Systems*, pp. 110-117, 1997.

[7] J. De Vriendt, I. Grimez Vinagre, and A. Van Ewijk, "Multimedia broadcast and multicast services in 3g mobile networks," Alcatel Telecommunications Review, no.4-1, pp.17–24, 2003.

[8] IST-2003-507607 (B-BONE), Deliverable of the project (D2.5), Final Results with combined enhancements of the Air Interface.