IJSEAT International Journal of Science Engineering and Advance Technology

Scheduling Cached-Content in Wireless Networks With Elastic and Inelastic Traffic

Syed Ajmal Pasha C[1], Sayeed Yasin[2]
[1] M.Tech (CSE), Nimra College of Engineering & Technology, A.P., India.
[2]Assoc.profissor & HOD of Computer Science & Engineering, Nimra College of Engineering & Technology, A.P., India.

Abstract — With the exponential growth of wireless content access, the need for content placement and scheduling at wireless base stations increased rapidly. We study a system under which users are divided into clusters based on their channel conditions, and their requests are represented by different queues at logical front ends. Requests might be elastic (implying no hard delay constraint) or inelastic (requiring that a delay target be met). Correspondingly, we have request queues that indicate the number of elastic requests, and deficit queues that indicate the deficit in inelastic service. Caches are of finite size and can be refreshed periodically from a media vault. We consider two cost models that correspond to inelastic requests for streaming stored content and real-time streaming of events, respectively. We design provably optimal policies that stabilize the request queues (hence ensuring finite delays) and reduce average deficit to zero [hence ensuring that the quality-of-service (QoS) target is met] at small cost. We illustrate our approach through simulations.

Keywords — **Content distribution network (CDN), delay-sensitive traffic, prediction, quality of service (QoS), queuing.**

I. Introduction

The rapid rise of smart handheld wireless devices as a means of content consumption has seen in the past few years. Content might include streaming applications in which chunks of the file must be received under hard delay constraints, as well as file downloads such as software updates that do not have such hard constraints. The core of the Internet is well provisioned, and network capacity constraints for content delivery are at the media vault (where content originates) and at the wireless access links at end-users. Hence, a natural location to place caches for a content distribution network (CDN) would be at the wireless gateway, which could be a cellular base station through which users obtain network access. Furthermore, it is natural to try to take advantage of the inherent broadcast nature of the wireless medium to satisfy multiple users simultaneously.

An abstraction of such a network is illustrated in Fig. 1. There are multiple cellular base stations (BSs), each of which has a cache in which to store content. The content of the caches can be periodically refreshed through accessing a media vault. We divide users into different clusters, with the idea that all users in each cluster are geographically close such that they have statistically similar channel conditions and are able to access the same base stations. Note that multiple clusters could be present in the same cell based on the dissimilarity of their channel conditions to different base stations. The requests made by each cluster are aggregated at a logical entity that we call a front end (FE) associated with that cluster. The front end could be running on any of the devices in the cluster or at a base station, and its purpose is to keep track of the requests associated with the users of that cluster. The following constraints affect system operation: 1) the wireless network between the caches to the users has finite capacity; 2) each cache can only host a finite amount of content; and 3) refreshing content in the caches from the media vault incurs a cost.

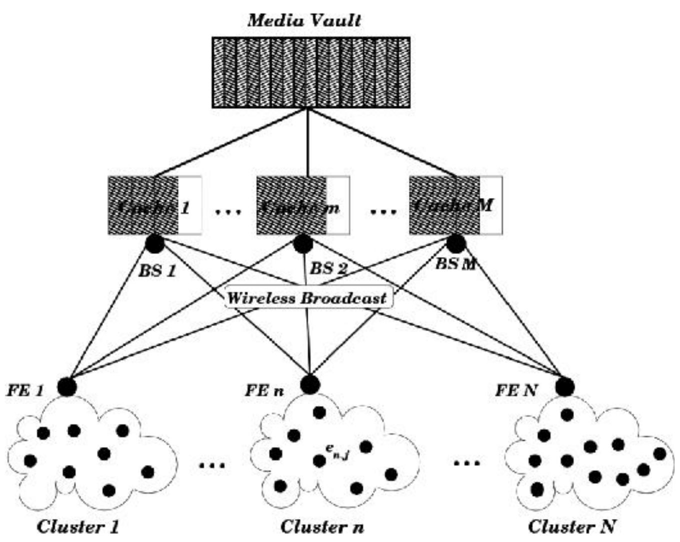

Fig. 1: Wireless content distribution. A media vault is used to place content in caches at wireless BSs, which can

broadcast content. Users are grouped into clusters, each of whose requirements are aggregated at FEs.

Users can make two kinds of requests, namely: 1) elastic requests that have no delay constraints, and 2) inelastic requests that have a hard delay constraint. Elastic requests are stored in a request queue at each front end, with each type of request occupying a particular queue. Here, the objective is to stabilize the queue, so as to have finite delays. For inelastic requests, we adopt the model proposed in [1] [2] wherein users request chunks of content that have a strict deadline, and the request is dropped if the deadline cannot be met. The idea here is to meet a certain target delivery ratio, which could be something like "90% of all requests must be met to ensure smooth play out." Each time an inelastic request is dropped, a deficit queue is updated by an amount proportional to the delivery ratio. We would like the average value of the deficit to be zero.

In this paper, we are interested in solving the joint content placement and scheduling problem for both elastic and inelastic traffic in wireless networks. In doing so, we will also determine the value of predicting the demand for different types of content and what impact it has on the design of caching algorithms.

II. SYSTEM MODEL

Consider the content distribution network depicted in Fig. 1.

There is a set of base stations and each base station M is associated with a cache. The caches are all connected to a media vault that contains all the content. The users in the system are divided into clusters based on their geographical positions, and we let N denote the set of these clusters. Also, as discussed in the Introduction, there are front ends in each cluster, also denoted by $n \in N$ whose purpose is to aggregate requests from the users. Time is slotted, and we divide time into frames consisting of D time-slots. Requests are made at the beginning of each frame. There are two types of users in this system—inelastic and elastic—based on the type of requests that they make. Requests made by inelastic users must be satisfied within the frame in which they were made. Elastic users do not have such a fixed deadline, and these users arrive, make a request, are served, and depart.

The base stations employ multiple access schemes (e.g., OFDMA), and hence each base station can support multiple simultaneous unicast transmissions, as well as a single broadcast transmission. It is also possible to study other scenarios (e.g., multicast transmissions to subsets of users) using our framework. We adopt a slow-fading packet erasure model for the wireless channels. Accordingly, the channel between cache m and user u (or front end n) is modeled as a stochastic ON–OFF process c^m_u (or c^m_n), which is *i.i.d.* over frames, and the state $c^m_u(K) \in \{0, 1\}$ (or $c^m_n(K)$) does not change during frame and is known to the scheduler. We assume that all pieces of content have the same size, and we call the unit of storage and transmission as a chunk. When a channel is ON, it can be used to transmit at most one chunk (per slot).

Note that while the Bernoulli process models an inelastic request for each user, the distribution of the requests over different content types can be chosen arbitrarily (e.g., following a Zipf's law that captures the varied popularity of different types). Since there are limited resources in the system, all requests cannot be served. In order to provide enough service to each user, we need to decide on a minimum delivery ratio for inelastic users. The delivery ratio is the proportion of inelastic requests that are served, and hence the expected service required by user u is $\eta_u\lambda_u$, in which η_u is the minimum acceptable delivery ratio. This model follows that of [2] and is consistent with the idea that streaming media can tolerate a fraction of chunk losses, but has hard delay constraints on the received chunks.

Each cache m has a finite capacity of v_m chunks of content. In what follows, we assume for simplicity that $v_m = v$ for all $m \in M$. Reloading a cache requires connecting to the media vault and fetching new chunks, which is subject to the capacity constraint between the media vault and the caches. One way to model this constraint is to consider different timescales for cache reloading and request arrivals. Hence, lower capacity can be modeled as a slower timescale for refreshing cache contents. For simplicity, we use the same timescale (i.e., frames) for request arrivals and refreshing cache contents. Hence, base stations can reload their caches with new content at the beginning of each frame. The same framework can be used to study the general case at the expense of more computational complexity.

We explicitly model the reloading cost for a variation of our caching model. For this model, we assume the content of the caches expires and will not be useful at the end of each frame. However, placing each chunk in a cache induces a cost. Therefore, in order to reduce the cost, we may occasionally choose to reload a cache partially and not utilize the whole available capacity. For this variation, we will only consider inelastic traffic, which is consistent with the idea of real-time streaming of live events.

Algorithm 1: Optimal Content Placement and Scheduling of Elastic Requests

At the beginning of each frame k: given the queue lengths $q_{n,e}(k)$, and the arrivals $a_{n,e}(k)$, let $q_{n,e} = q_{n,e}(k) + a_{n,e}(k)$.

Content placement:

At each cache m, solve the following maximization problem to find the optimal placement $(p_e^m)^*$:

$$\max \sum_{e,n} c_n^m q_{n,e} \alpha_{n,e}$$
$$\text{s.t.} \quad \alpha_{n,e} \le p_e^m \qquad \forall n, e$$
$$\sum_e \alpha_{n,e} \le 1 \qquad \forall e$$
$$p_e^m \in \{0,1\} \qquad \forall e$$
$$\sum_e p_e^m \le v \qquad (6)$$

in which c_n^m values denote the channel states during this frame and are given.

Service scheduling:

For each cache m and front end n, determine the optimal schedule $(s_{n,e}^m)^*$ as follows:

$$(s_{n,e}^m)^* = \begin{cases} D, & \text{if } e = rand\Big(\arg\max_{f \in \mathbf{E}} \big((p_f^m)^* c_n^m q_{n,f}\big)\Big) \\ 0, & \text{otherwise.} \end{cases}$$

Thus, the capacity of the link between cache m and front end n is completely devoted to serve one of the contents (randomly chosen) that maximizes $(p_f^m)^* c_n^m q_{n,f}$.

III. RELATED WORK

The problem of caching and content scheduling has earlier been studied for online Web caching and distributed storage systems. A commonly used metric is a competitive ratio of misses, assuming an adversarial model. Examples of work in this context are [3]–[5]. Load balancing and placement with linear communication costs is examined in [6] and [7]. Here, the objective is to use distributed and centralized integer programming approaches to minimize the costs. However, this work does not take account for network capacity constraints, delay-sensitive traffic, or wireless aspects.

The techniques that we will employ are based on the literature on scheduling schemes. Tassiulas et al. proposed the MaxWeight scheduling algorithm for switches and wireless networks in their seminal work [8]. They proved that this policy is throughput-optimal and characterized the capacity region of the single-hop networks as the convex hull of all feasible schedules. Various extensions of this work that followed since are [9]–[12]. These papers explore the delays in the system for single downlink with variable connectivity, multirate links, and multihop wireless flows. However, these do not consider content distribution with its attendant question of content placement. Closest to our work is [13], which, however, only considers elastic traffic and has no results on the value of prediction.

Main Results

In this paper, we develop algorithms for content distribution with elastic and inelastic requests. We use a request queue to implicitly determine the popularity of elastic content. Similarly, the deficit queue determines the necessary service for inelastic requests. Content may be refreshed periodically at caches. We study two different kinds of cost models, each of which is appropriate for a different content distribution scenario. The first is the case of file distribution (elastic) along with streaming of stored content (inelastic), where we model cost in terms of the frequency with which caches are refreshed. The second is the case of streaming of content that is generated in real-time, where content expires after a certain time, and the cost of placement of each packet in the cache is considered.

- We first characterize the capacity region of the system and develop feasibility constraints that any stabilizing algorithm must satisfy. Here, by stability we mean that elastic request queues have a finite mean, while inelastic deficit values are zero on average.

- We develop a version of the max-weight scheduling algorithm that we propose to use for joint content placement and scheduling. We show that it satisfies the feasibility constraints and, using a Lyapunov argument, also show that it stabilizes the system of the load within the capacity region. As a by-product, we show that the value of knowing the arrival rates is limited in the case of elastic requests, while it is not at all useful in the inelastic case.

- We next study another version of our content distribution problem with only inelastic traffic, in which each content has an expiration time. We assume that there is a cost for replacing each expired content chunk with a fresh one. For this model, we first find the feasibility region and, following a similar technique to [14], develop a joint content placement and scheduling algorithm that minimizes the average expected cost while stabilizing the deficit queues.

- We illustrate our main insights using simulations on a simple wireless topology and show that our algorithm is indeed capable of stabilizing the system. We also propose two simple algorithms, which are easily implementable, and compare their performance to the throughput-optimal scheme.

IV. Conclusion

In this paper, we studied algorithms for content placement and scheduling in wireless broadcast networks. While there has been significant work on content caching algorithms, there is much less on the interaction of caching and networks. Converting the caching and load balancing problem into one of queueing and scheduling is hence interesting. We considered a system in which both inelastic and elastic requests coexist. Our objective was to stabilize the system in terms of finite queue lengths for elastic traffic and zero average deficit value for the inelastic traffic. We showed how an algorithm that jointly performs scheduling and placement in such a way that Lyapunov drift is minimized is capable of stabilizing the system. In designing these schemes, we showed that knowledge of the arrival process is of limited value to taking content placement decisions. We incorporated the cost of loading caches in our problem with considering two different models. In the first model, cost corresponds to refreshing the caches with unit periodicity. In the second model relating to inelastic caching with expiry, we directly assumed a unit cost for replacing each content after expiration.

A max-weight-type policy was suggested for this model, which can stabilize the deficit queues and achieves an average cost that is arbitrarily close to the minimum cost.

V. References

[1] N. Abedini and S. Shakkottai, “Content caching and scheduling in wireless broadcast networks with elastic and inelastic traffic,” in Proc. IEEE WiOpt, 2011, pp. 125–132.

[2] I. Hou, V. Borkar, and P. Kumar, “A theory of QoS for wireless,” in Proc. IEEE INFOCOM, Rio de Janeiro, Brazil, Apr. 2009, pp. 486–494.

[3] R. M. P. Raghavan, Randomized Algorithms. NewYork,NY,USA: Cambridge Univ. Press, 1995.

[4] P. Cao and S. Irani, “Cost-awareWWWproxy caching algorithms,” in Proc. USENIX Symp. Internet Technol. Syst., Berkeley, CA, Dec. 1997, p. 18.

[5] K. Psounis and B. Prabhakar, “Efficient randomized Web-cache replacement schemes using samples from past eviction times,” IEEE/ACM Trans. Netw., vol. 10, no. 4, pp. 441–455, Aug. 2002.

[6] N. Laoutaris, O.T. Orestis, V.Zissimopoulos, and I. Stavrakakis, “Distributed selfish replication,” IEEE Trans. Parallel Distrib. Syst., vol. 17, no. 12, pp. 1401–1413, Dec. 2006.

[7] S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithms for content distribution networks,” in Proc. IEEE INFOCOM, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[8] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Trans. Autom. Control, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[9] X. Lin and N. Shroff, “Joint rate control and scheduling in multihop wireless networks,” in Proc. 43rd IEEE CDC, Paradise Islands, Bahamas, Dec. 2004, vol. 2, pp. 1484–1489.

[10] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” Queueing Syst. Theory Appl., vol. 50, no. 4, pp. 401–457, 2005.

[11] A. Eryilmaz and R. Srikant, “Joint congestion control, routing, and MAC for stability and fairness in wireless networks,” IEEE J. Sel. Areas Commun., vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[12] J. Jaramillo and R. Srikant, “Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic,” in Proc. IEEE INFOCOM, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[13] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, “Content-aware caching and traffic management in content distribution networks,” in Proc. IEEE INFOCOM, Shanghai, China, Apr. 2011, pp. 2858–2866.

[14] M. Neely, “Energy optimal control for time-varying wireless networks,” IEEE Trans. Inf. Theory, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.