ACHIEVING QUALITY OF SERVICE GUARANTEES FOR DELAY SENSITIVE APPLICATIONS IN WIRELESS NETWORKS

A Dissertation

by

NAVID ABEDINI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2012

Major Subject: Computer Engineering

ACHIEVING QUALITY OF SERVICE GUARANTEES FOR DELAY SENSITIVE APPLICATIONS IN WIRELESS NETWORKS

A Dissertation

by

NAVID ABEDINI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Srinivas Shakkottai
Committee Members,	Jean-Francois Chamberland-Tremblay
	Natarajan Gautam
	Panganamala Ramana Kumar
	Narasimha Annapareddy
Head of Department,	Costas N. Georghiades

August 2012

Major Subject: Computer Engineering

ABSTRACT

Achieving Quality of Service Guarantees for Delay Sensitive Applications in Wireless Networks. (August 2012)

Navid Abedini, B.S., Sharif University of Technology, Iran

Chair of Advisory Committee: Dr. Srinivas Shakkottai

In the past few years, we have witnessed the continuous growth in popularity of delay-sensitive applications. Applications like live video streaming, multimedia conferencing, VoIP and online gaming account for a major part of Internet traffic these days. It is also predicted that this trend will continue in the coming years. This emphasizes the significance of developing efficient scheduling algorithms in communication networks with guaranteed low delay performance. In our work, we try to address the delay issue in some major instances of wireless communication networks.

First, we study a wireless *content distribution network* (CDN), in which the requests for the content may have service deadlines. Our wireless CDN consists of a media vault that hosts all the content in the system and a number of local servers (base stations), each having a cache for temporarily storing a subset of the content. There are two major questions associated with this framework: (i) content caching: which content should be loaded in each cache? and (ii) wireless network scheduling: how to appropriately schedule the transmissions from wireless servers? Using ideas from queueing theory, we develop provably optimal algorithms to jointly solve the caching and scheduling problems.

Next, we focus on *wireless relay networks*. It is well accepted that *network coding* can enhance the performance of these networks by exploiting the broadcast nature of the wireless medium. This improvement is usually evaluated in terms of the number of required transmissions for delivering flow packets to their destinations.

In this work, we study the effect of delay on the performance of network coding by characterizing a trade-off between latency and the performance gain achieved by employing network coding. More specifically, we associate a holding cost for delaying packets before delivery and a transmission cost for each broadcast transmission made by the relay node. Using a Markov decision process (MDP) argument, we prove a simple threshold-based policy is optimal in the sense of minimum long-run average cost.

Finally, we analyze delay-sensitive applications in wireless *peer-to-peer* (P2P) networks. We consider a hybrid network which consists of (i) an expensive *base station-to-peer* (B2P) network with unicast transmissions, and (ii) a free broadcast P2P network. In such a framework, we study two popular applications: (a) a content distribution application with service deadlines, and (b) a multimedia live streaming application. In both problems, we utilize random linear network coding over finite fields to simplify the coordination of the transmissions. For these applications, we provide efficient algorithms to schedule the transmissions such that some *quality of service* (QoS) requirements are satisfied with the minimum cost of B2P usage. The algorithms are proven to be throughput optimal for sufficiently large field sizes and perform reasonably well for finite fields.

To

My wife, Sonia

and my beloved family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Professor Srinivas Shakkottai. Only his continuous support, brilliant ideas and encouragement could make this work possible. I had such a joyful experience working with him, since he was always open for questions and extremely enthusiastic for discussions. Besides all his guidance and great supervision in the work, his patience and exemplary positive attitude were indispensable factors in helping me find my interests and love my research. He has been an ideal advisor, a remarkable collaborator and a great teacher.

I am also very thankful to my thesis committee members: Professor Jean-Francois Chamberland, Professor Natarajan Gautam, Professor P. R. Kumar and Professor Narasimha Reddy. Their questions, feedback and novel ideas helped me a lot in completing this dissertation. It was a great honor to be their student and have them in my PhD committee.

I am also deeply grateful to Professor Serap Savari, for the opportunity of working with her in my first year of PhD. I had also the great fortune of collaborating with Professor Sunil Khatri and Professor Alex Sprintson. My special thanks to all my wonderful teachers including Professor Robert Cui, Professor Scott Miller, Professor Krishna Narayanan and Professor Tie Liu, from whom I learned a lot, and I really enjoyed their courses at Texas A&M University. I would also like to sincerely thank Carolyn Warzon for all her assistance and effort in the Computer Engineering and Systems Group.

During my four years of PhD studies at Texas A&M University, I have found the most admirable friends and colleagues. To name a few of them who had a crucial impact on my academic life, I would like to acknowledge Meghana Amble, Babak Faryabi, Yu-Pin Hsu, Mayank Manjrekar, Parimal Parag, Vinod Ramaswamy, Solairaja Ramasamy, Sadegh Tabatabaei and Golanz Vahedi. I would also like to specially thank Swetha Sampath for all the great effort she put into implementing our analytical media streaming results on an Android framework. I also had the nicest and most peaceful officemates: Suming Lai, Sangwhan Moon and Sandeep Yadav.

Finally, I would like to express my true love and gratitude to my encouraging father Ghodrat, my adorable mother Mahdokht, my lovely sisters (in-laws) Leila, Sara and Mahsa and my awesome brothers (in-laws) Babak, Pezhman and Mahziar. My achievements (if any) were only because of their great support and motivation. Many thanks to my fantastic parents-in-law, Reza and Farzaneh, not only for all their caring and inspiration, but more importantly for giving me the sweetest creature, Sonia. She has been always around to give excitement and meaning to my life, and I cannot imagine where I would be without her love and support.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

We are entering an era in which wireless devices are the major sources of Internet traffic. Many of the applications running on these devices are sensitive to delay. A recent report on analyzing Internet traffic [1] declares that more than 49% of Internet traffic during the peak period in North America is generated by real-time entertainment applications. In another report, conducted by Cisco [2], it is predicted that the traffic due to online gaming and video calling applications will increase about 300% by end of 2015.

Our main goal in this research is to study the delay issue in wireless communication networks. We will consider different situations in wireless networks, where low delay performance is required. For such instances, we show how to take latency into account and develop efficient scheduling algorithms. More specifically, we study the following problems:

1. **Wireless content distribution networks: caching and scheduling**

   The rapid growth of wireless content access implies the need for content placement and scheduling at wireless base stations. We study a system under which clients are divided into clusters based on their channel conditions, and their requests are represented by different queues at logical frontends. 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

The journal model is *IEEE/ACM Transactions on Networking*.

a media vault. We design provably optimal policies that stabilize the request queues (hence ensuring finite delays) and reduce average deficit to zero (hence ensuring that the QoS target is met).

2. **Wireless relay networks: delay versus network coding performance** [1]

   It has been well established that reverse-carpooling based network coding can significantly improve the efficiency of multi-hop wireless networks. However, in a stochastic environment when there are no opportunities to code because of packets without coding pairs, should these packets wait for a future opportunity or should they be transmitted without coding? To help answer that question we formulate a stochastic dynamic program with the objective of minimizing the long-run average cost per unit time incurred due to transmissions and delays. In particular, we develop optimal control actions that would balance between costs of transmission against those of delays. In that process we seek to address a crucial question: what should be observed as the state of the system? We analytically show that just the queue lengths is enough if it can be modeled as a Markov process. Subsequently we show that a stationary policy based on queue lengths is optimal and describe a procedure to find such a policy.

3. **Wireless broadcast P2P networks: timely synchronization of data** [2]

   We consider a group of cooperative wireless peer devices that desire to receive the same content within some deadline. The block(s) of content is divided into chunks, which are received via two methods that can be used simultaneously (i) the B2P (base-station-to-peer) network: each peer has an unreliable, expensive,

[1]This work was done in collaboration with other students, Yu-Pin Hsu and Solairaja Ramasamy

[2]Parts of this work were done in collaboration with other students, Mayank Manjrekar and Swetha Sampath

unicast channel to a cellular base station, and (ii) the P2P (peer-to-peer) network: peers can share the content over a free, lossless internal wireless broadcast network. Chunks are coded using random linear codes to alleviate the duplicate chunk reception issue, and the state of each peer can be associated with the rank of the matrix of chunk vectors that it possesses.

We study two problems in this framework. First, a content distribution problem, in which the QoS metric is that peers are all required to receive a single common block with a certain target probability by a fixed deadline. Second, a multimedia live streaming problem, where peers are interested in a common stream of media content which is generated as a long sequence of blocks, each requiring a fixed deadline for delivery. The QoS metric requires that each peer individually receives a minimum long-run average fraction of the blocks within their deadlines.

We seek efficient algorithms that can attain these QoS metrics at the lowest cost of using B2P network. We transform the problem into the two questions of (i) deciding which peer should broadcast on the P2P channel at each time, and (ii) how long B2P transmissions should take place. For each scenario, we provide policies for coordinating P2P transmissions which are shown to be optimal for large field sizes, and determine the stopping time for B2P transmissions in an offline manner. We also provide performance bounds for finite field sizes.

In the following chapters, we will separately study each of these problems in detail. We review the previous work that has been done in each area, present our analytical and simulation results, and give some ideas for the future work.

CHAPTER II

CACHING AND SCHEDULING OF ELASTIC AND INELASTIC TRAFFIC

The past few years have seen the rise of smart hand-held wireless devices as a means of content consumption. 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. Since the core of the Internet is far less bandwidth constrained than access wireless networks, a natural location to implement a content distribution network (CDN) would be at the wireless gateway, which could be a cellular base-station through which users obtain network access. Further, it is natural to try to take advantage of the inherent broadcast nature of the wireless medium to satisfy multiple clients simultaneously.

An abstraction of such a network is illustrated in Figure 1. There are multiple cellular *base stations* (BS), 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 clients into different *clusters*, with the idea that all clients in each cluster are geographically close such that they have 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. Thus, we have N clusters indexed by $n = 1, 2, ..., N$. The requests made by each cluster are aggregated at a logical entity that we call a *front end* 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 clients of that cluster. The following constraints affect system operation (i) the wireless broadcast network between the caches to the clients has finite capacity, (ii) each cache can only host a finite amount of content, and (iii) refreshing content in the caches from the media vault incurs a cost.

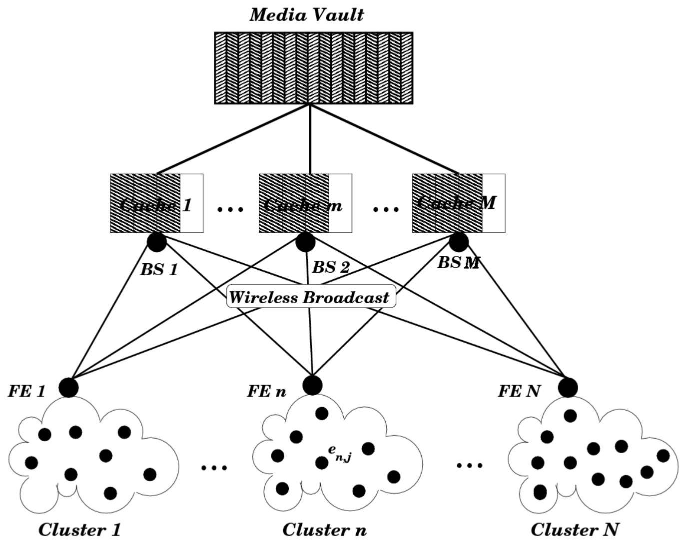


Fig. 1. Wireless broadcast content distribution. A media vault is used to place content in caches at wireless base stations (BS), which can broadcast content. Clients are grouped into clusters, each of whose requirements are aggregated at frontends (FE).

Clients can make two kinds of requests, namely (i) elastic requests that have no delay constraints, and (ii) 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 [3] wherein clients 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 playout”. 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 research, we are interested in solving the joint content placement and scheduling problem for both elastic and inelastic traffic in wireless broadcast networks. While the classical caching literature does not deal with this problem, several papers on switch scheduling are related to this question. Tassiulas *et al.* proposed the Max Weight scheduling algorithm for switches and multihop wireless networks in their seminal work [4]. 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 [5–8]. These papers explore the delays in the system for single down-link with variable connectivity, multi-rate links and multi-hop wireless flows. However, [5–8] do not consider content distribution with its attendant question of content placement, while [9] only considers elastic traffic. We use some of the analytical techniques of these papers in the context of content distribution.

A. Main Results

In this research, 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.

- We then consider the case of periodic refresh under which content may be placed in the caches at finite intervals of time. We show that a similar algorithm to the max-weight scheme is optimal here. Further, in contrast to our earlier work [9] which only considers elastic traffic, joint scheduling and eviction is essential to stabilize the system.

- 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 [10], we develop a joint content placement and scheduling algorithm which 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.

B. Problem Definition

Consider the content distribution network depicted in Figure 1. There are a fixed number of inelastic clients in cluster n who make inelastic requests and are denoted by $e_{n,j}$ for $j = 1, 2, ..., N_n$. The elastic requests in this cluster are made by other clients who may temporarily exist in the system. Each of these clients is interested in at most one elastic chunk during his existence. We aggregate all elastic requests at the frontend (denoted by $e_{n,0}$ in cluster n). For simplicity, we assume the elastic chunks must be delivered to the frontends. The results will still hold without this assumption. The system consists of M caches which are all connected to a media vault. Each cache m has a finite capacity of v_m pieces of content (we assume that all have the same size), for $m = 1, 2, ..., M$. The media vault stores $\mathbf{F}$, the set of all content. $\mathbf{F}$ is partitioned into two disjoint sets $\mathbf{I}$ and $\mathbf{E}$ which are respectively the set of inelastic content and elastic content: $\mathbf{F} = \mathbf{I} \cup \mathbf{E}$.

We assume that service from the base stations is by means of broadcast transmissions. Since we associate each base station with a cache, we will use the same notation for a cache and a base station. Suppose that time is slotted. The channel $l_{n,j}^m$ between cache m and the client $e_{n,j}$ is modeled as a stochastic On-OFF process whose state $c_{n,j}^m(t)$ at time t is unknown to the scheduler. We assume the channels of the clients in the same cluster have similar average characteristics. During each time slot link $l_{n,j}^m$ is ON with probability $\bar{c}_n^m$ and OFF with probability $1 - \bar{c}_n^m$, for $j = 0, 1, ..., N_n$. When a channel is ON, it can be used to transmit at most one chunk (per slot) which is being broadcast by the corresponding cache. We divide time into *frames* consisting of D time slots.

At the beginning of each frame k, each inelastic client makes at most one request. The idea is that an inelastic request must either be satisfied by the end of the frame or dropped. Let $a^I_{n,j}(k) = (a^I_{n,j}(i,k) : i \in \mathbf{I})$ be the inelastic request vector that client $e_{n,j}$ makes at frame k. We model this request by a Bernoulli process with the mean value $\lambda^I_{n,j}$, that is

$$\begin{aligned} &\textstyle\sum_{i\in\mathbf{I}} a^I_{n,j}(i,k) = 1 \quad \text{with probability } \lambda^I_{n,j} \\ &\textstyle\sum_{i\in\mathbf{I}} a^I_{n,j}(i,k) = 0 \quad \text{with probability } 1-\lambda^I_{n,j} \end{aligned} \tag{2.1}$$

Note that while the Bernoulli process models an inelastic request for each client, 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).

Also in each cluster, there may be some requests for each chunk of elastic content $e \in \mathbf{E}$ at the beginning of a frame. $a^E_n(k) = (a^E_n(e,k) : e \in \mathbf{E})$ is used to denote the elastic request vector. $a^E_n(e,k) \leq \mathbf{a}^e_n < \infty$ is a bounded random variable with mean λ^e_n and variance $\sigma^2_{e,n}$. We further assume that arrivals are independently distributed over frames.

As mentioned above, an inelastic request made at the beginning of a frame is valid only till the end of that frame. In other words, if an inelastic request is not served during a frame, it will be dropped. This implies two important results: (1) there are at most D time slots delay in serving the inelastic requests, (2) because of the limited resources in the system, all of the requests cannot be served. While the former provides a hard limit on the maximum service delay for inelastic traffic, the latter suggests that for providing enough service to each client, we need to declare a minimum *delivery ratio* for inelastic requests. Thus, the delivery ratio is the proportion of the inelastic requests which are served. Therefore, the expected inelastic service of client $e_{n,j}$ is $\eta_{n,j}\lambda^I_{n,j}$, in which $\eta_{n,j}$ is the minimum acceptable delivery ratio. This model follows that of [3], 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.

On the other hand, an elastic request which does not get served during a frame will be enqueued and wait for the service during next frames. However, we need to make sure that the request queue lengths in each cluster remain bounded as time passes so that the delay does not become unboundedly large. Thus we require that the expected elastic service for a content e in cluster n is λ_n^e.

We will consider three variations of this problem: (1) unit period cache refresh, (2) periodic cache refresh and (3) inelastic caching with expiry and cost of replacement. In the first case, we can reload caches with completely new content at the beginning of each frame. However, in practice, reloading a cache requires connecting to the media vault and fetching chunks, which can incur a cost. Hence, we model cost in two different ways in the following two cases. In the second variation of the problem, we define the notion of *cache refresh period* to model the cache reloading cost. A cache refresh period consists of R frames. The content of a cache can be completely refreshed only at the beginning of each refresh period. However, in the inter-refresh frames we may only partially refresh the cache content (i.e., replace at most r_m out of v_m chunks of cache m). Thus, to formulate higher refresh costs we can consider a larger refresh period (i.e., larger R value) or smaller r_m values. In the third case, we assume that the content of the caches expires and will not be useful at the end of each frame. This case is similar to that of real-time streaming of ongoing events. 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.

In the next section, we will consider the unit period scenario. First, the set of all allowable requests is found, then we show that a max-weight type algorithm is *throughput optimal* in the sense that it can satisfy any set of allowable requests. The periodic cache refresh case and the inelastic caching with expiry will be addressed in Sections D and E respectively.

C. Unit Period Cache Refresh

In this section we first find the *feasibility region*, which is the set of all allowable schedules. Next, we will determine the *capacity region*, which is the set of all allowable requests by the clients.

1. Feasibility and Capacity Regions

Let $p_m(k) = (p_{m,f}(k) : f \in \mathbf{F})$ be the chunk presence vector at cache m and frame k, that is $p_{m,f}(k) = 1$ if f is present in cache m at frame k and $p_{m,f}(k) = 0$ otherwise. The cache capacity constraint requires each cache m to satisfy

$$\sum_{f \in \mathbf{F}} p_{m,f}(k) \leq v_m \quad \text{for each frame } k. \tag{2.2}$$

For a given frame k, we denote the constituent time slots by $\{t : t \in k\}$. During a time slot each link can transmit one chunk when it is ON. Let $s_f(m,t) \in \{0,1\}$ denote the scheduled service to chunk $f \in \mathbf{F}$ from cache m and during time slot t, that is $s_f(m,t) = 1$ if cache m broadcasts chunk f at time t and $s_f(m,t) = 0$ otherwise. Without loss of generality, we assume that each cache broadcasts at most one chunk during each slot. Clearly this chunk must be present in the cache at that time. Therefore, the following conditions need to hold for each cache m and $t \in k$:

$$\sum_f s_f(m,t) \leq 1 \quad \text{and} \quad s_f(m,t) \leq p_{m,f}(k) \tag{2.3}$$

Since the states of the channels are not known a priori, the scheduler assumes that all channels are ON to determine the schedule. Also for each inelastic chunk, we have at most one valid request from each client during each frame. This suggests that each inelastic chunk i needs to be broadcast at most once in the system during a frame k:

$$\sum_m \sum_{t \in k} s_i(m,t) \leq 1 \quad \text{for each } i \in \mathbf{I} \tag{2.4}$$

Now, we define the feasibility region *FeasReg(V)* for the cache capacity vector $V = (v_m : 1 \leq m \leq M)$, as the set of all vectors $S = (s_f(m,t), p_{m,f}(k) : f \in \mathbf{F}, 1 \leq m \leq M, t \in k)$ satisfying (2.2), (2.3) and (2.4) in frame k.

A set of requests is said to be *allowable*, if there exists a policy to schedule the network such that (i) each client receives enough expected service and (ii) the scheduled service lies in the feasibility region *FeasReg(V)*.

Suppose that S is a feasible schedule. The actual service $\mu_{n,j}(f,k)$ received by client $e_{n,j}$ for chunk f during frame k depends on the realization of the channels $c^m_{n,j}(t)$ for all m and t and can be written as

$$\mu_{n,j}(f,k) = \sum_{t \in k} \sum_m c^m_{n,j}(t) s_f(m,t). \tag{2.5}$$

Since the inelastic requests have strict deadlines, the provided inelastic service $\mu^I_{n,j}(k)$ depends on whether there is a request arrival for the delivered chunk in that frame. That is

$$\begin{aligned} \mu^I_{n,j}(k) &= \textstyle\sum_{i \in I} \min(\mu_{n,j}(i,k), a^I_{n,j}(i,k)) \\ &= \textstyle\sum_{i \in I} a^I_{n,j}(i,k) \mu_{n,j}(i,k) \end{aligned} \tag{2.6}$$

where the second equality follows from $a^I_{n,j}(i,k), \mu_{n,j}(i,k) \in \{0,1\}$.

Since the states of the wireless links and the arrivals are identically and independently distributed over frames, following the same argument as in [11] we can formally define the capacity region based on the existence of a randomized stationary policy which can fulfill the requests:

Definition 1. (capacity region of the unit period scenario) *Consider a system with the cache capacities* $V = (v_m : 1 \leq m \leq M)$ *and the average channel capacities* $C = (\bar{c}_n^m : 1 \leq n \leq N, 1 \leq m \leq M)$. *A set of request arrivals* $\Lambda = (\lambda_n^e, \lambda_{n,j}^I, \eta_{n,j} : n, j, e)$ *is allowable if there exists a policy* $\mathbf{P}^*$ *which during each frame* k *with the arrival profile* $A(k) = (a_{n,j}^I(i,k), a_n^E(e,k) : n, j, i, e)$, *chooses a feasible schedule* $S \in \text{FeasReg(V)}$ *with probability* $\mathbb{P}(S|A(k))$, *such that* $\sum_{S \in \text{FeasReg(V)}} \mathbb{P}(S|A(k)) = 1$. *And*

$$\begin{aligned} \eta_{n,j}\lambda_{n,j}^I &\leq \bar{\mu}_{n,j}^I \qquad \textit{for each client } e_{n,j} \\ \lambda_n^e &\leq \bar{\mu}_n^e \qquad \textit{for each } e \in \mathbf{E} \textit{ and } n \end{aligned} \tag{2.7}$$

where the average services $\bar{\mu}_{n,j}^I$ *and* $\bar{\mu}_n^e$ *are defined as*

$$\begin{aligned} \bar{\mu}_{n,j}^I &= \mathbb{E}_{A(k)}\left[\mathbb{E}_C\left[\mathbb{E}_S\left[\sum_{i,m,t} a_{n,j}^I(i,k)c_{n,j}^m(t)s_i(m,t)\right]\right]\right] \\ \bar{\mu}_n^e &= \mathbb{E}_{A(k)}\left[\mathbb{E}_C\left[\mathbb{E}_S\left[\sum_{m,t} c_{n,0}^m(t)s_e(m,t)\right]\right]\right]. \end{aligned}$$

Note that $\mathbb{E}_S[.]$ *is the expectation over all feasible schedules in the* $\text{FeasReg}(V)$ *with respect to the probability distribution* $\mathbb{P}(S|A(k))$ *implied by* $\mathbf{P}^*$. *Also* $\mathbb{E}_A[.]$ *and* $\mathbb{E}_C[.]$ *are expectations over respectively arrival processes and channels states. A set of requests is said to be* strictly allowable, *if (2.7) holds with strict inequalities. The capacity region is defined as the convex hull of all allowable sets of requests.*

2. Max-Weight Algorithm: Throughput Optimality and Implementation

In this section, we show that the max-weight algorithm is throughput optimal. In other words, the max-weight algorithm can fulfill any set of allowable requests. We consider two types of queues: (i) *request queues* for the elastic traffic and (ii) *deficit queues* for the inelastic traffic.

The elastic requests in cluster n go through a set of request queues whose lengths at frame k are denoted by $q_n^e(k)$ for each chunk e, and follow the dynamic below:

$$q_n^e(k+1) = q_n^e(k) + a_n^E(e,k) - \mu_n^e(k). \tag{2.8}$$

where $\mu_n^e(k) = \min(\mu_{n,0}(e,k), q_n^e(k^+))$ and $q_n^e(k^+) = q_n^e(k) + a_n^E(e,k)$.

Next we define the deficit queue for each client $e_{n,j}$ which captures the accumulated unhappiness of the client about the provided inelastic service. $d_{n,j}(k)$ denotes the length of the corresponding deficit queue at frame k and follows

$$d_{n,j}(k+1) = d_{n,j}(k) + \sum_{i \in \mathbf{I}} \tilde{a}_{n,j}^I(i,k) - \mu_{n,j}^I(k). \tag{2.9}$$

where $\tilde{a}_{n,j}^I(i,k) = a_{n,j}^I(i,k)$ with probability $\eta_{n,j}$ and it is zero otherwise. Note that the deficit queue is a virtual queue whose length can be negative. A negative length shows up when the provided inelastic service is greater than the expected service. We also define $d_{n,j}(k^+) = d_{n,j}(k) + \sum_{i \in \mathbf{I}} \tilde{a}_{n,j}^I(i,k)$.

The evolution of these queues can be studied by considering a Markov chain whose states are vectors of all request and deficit queue lengths. Using the Lyapunov stability criterion, we show that a max-weight algorithm implies that this Markov chain is positive recurrent and $\lim_{K\to\infty} \mathbb{E}\left[\max(d_{n,j}(K), 0)\right]$ and $\lim_{K\to\infty} \mathbb{E}\left[q_n^e(K)\right]$ are some finite values. As a result, we will have the following conditions which are sufficient for satisfying the elastic and inelastic requests,

$$\lim_{K\to\infty} \mathbb{E}\left[\frac{1}{K} d_{n,j}(K)\right] = \eta_{n,j}\lambda_{n,j}^I - \bar{\mu}_{n,j}^I \leq 0$$

and for each e,

$$\lim_{K\to\infty} \mathbb{E}\left[\frac{1}{K} q_n^e(K)\right] = \lambda_n^e - \bar{\mu}_n^e = 0.$$

We first present the Foster-Lyapunov stability criterion that will enable us to derive the future stability results.

Theorem 1. *(Foster-Lyapunov stability criterion) Let* $\mathbf{Q}$ *be a countable state-space, and let* $\vec{\mathcal{Q}}[k]$ *be an irreducible, aperiodic, countable-state Markov chain. Suppose there exists a Lyapunov function* $L : \mathbf{Q} \to \mathbb{R}^+$, *and* $\hat{\mathbf{Q}}$, *which is a finite subset of* $\mathbf{Q}$. *If* $\delta > 0$ *and* b *is a constant such that the drift*

$$\Delta L[k] = \mathbb{E}\left[L[k+1] - L[k] \,|\vec{\mathcal{Q}}[k]\right] \leq -\delta + b I_{\hat{\mathbf{Q}}}, \tag{2.10}$$

then $\vec{\mathcal{Q}}[k]$ *is positive recurrent.*

The following theorem summarizes our result on the throughput optimality of the max-weight algorithm.

Theorem 2. *For any set of strictly allowable requests, the max-weight algorithm (in* **Algorithm 1***) stabilizes the request and deficit queues and hence is throughput optimal.*

In Theorem 2, we show that by applying the max-weight algorithm, the Markov chain whose states are the vectors of deficit and request queue lengths is positive recurrent. Therefore, this Markov chain converges to a unique steady state. The following corollary provides a bound on the queue lengths at the steady state.

Corollary 1. *Sum of the average request and the positive part of the deficit queue lengths at the steady state satisfies*

$$\begin{gathered}\textstyle\sum_{n,j} \mathbb{E}\left[\{d_{n,j}\}^+\right] + \sum_{n,e} \mathbb{E}\left[q_n^e\right] \leq \\ (\sum_n N_n + 5D^2M^2N + \textstyle\sum_{n,e} (\mathbf{a}_n^e)^2)/2\epsilon.\end{gathered} \tag{2.11}$$

where $\{d_{n,j}\}^+ = \max(d_{n,j}, 0)$ *and for some* $\epsilon > 0$ *which is the heavy traffic parameter,* i.e., ϵ *determines how close to the boundary of the capacity region the requests are.*

In the next section, we study the periodic cache refresh case as a generalization of the unit-period scenario. The proof of the above theorem and the corollary is a special case of the one for the periodic case (Theorem 3 and Corollary 2) and hence is omitted for the brevity.

In **Algorithm 1**, $s_f(m) = \sum_{t \in k} s_f(m,t)$ is the total scheduled service that cache m provides for chunk f during frame k. The solution to the maximization (2.13), $(s_i^*(m), s_{e^*(m)}^*(m), p_{m,e^*(m)}^* : m, i)$, will give the optimal placement: for each cache m and $i \in \mathbf{I}$ and $e \in \mathbf{E}$

$$p_{m,i}^* = s_i^*(m) \quad \text{and} \quad p_{m,e}^* = 0 \quad \text{for each } e \neq e^*(m) \tag{2.12}$$

During the corresponding frame, each cache m transmits the loaded inelastic chunks first. If there are some time slots left, the loaded elastic chunk $e^*(m)$ will be broadcast for the remaining of the frame.

D. Periodic Cache Refresh

In the periodic cache refresh scenario we model the cost incurred by fetching content from the media vault using R, the periodicity of cache reloading, and $r_m \leq v_m$, the number of chunks that can be fetched by cache m in an inter-refresh frame.

Each client $e_{n,j}$ is interested in a sequence of chunks during a cache refresh period. For example, a client who is watching a video makes a request for a sequence of video frames for the next couple of time units (this corresponds to inelastic traffic, since there are some QoS constraints associated with a streaming video). On the other hand, a client who wants to download a software update requires a set of chunks.

Algorithm 1 Unit-Period Scenario: Max-Weight Algorithm

At the beginning of each frame k: given the queue lengths $d_{n,j}(k), q_n^e(k)$, and the arrivals $A(k)$ compute,

$d_{n,j} = d_{n,j}(k) + \sum_{i \in \mathbf{I}} \tilde{a}_{n,j}^I(i,k)$

$q_n^e = q_n^e(k) + a_n^E(e,k)$

$w_{m,i}^I(k) = \sum_{n,j} \{d_{n,j}\}^+ a_{n,j}^I(i,k)\bar{c}_n^m$

$w_{m,e}^E = \sum_n q_n^e \bar{c}_n^m$

$e^*(m) = RAND\{arg \max_{e'}(w_{m,e'}^E)\}$ (i.e., one of the content chosen uniformly at random among the elastic content with largest weights)

Solve the following maximization problem:

$$\begin{array}{c} \max \sum_m w_{m,e^*(m)}^E p_{m,e^*(m)} s_{e^*(m)}(m) + \sum_{i,m} w_{m,i}^I(k) s_i(m) \\ \text{subject to} \\ \begin{array}{ll} (i)\forall m: & \sum_f s_f(m) \leq D \\ (ii)\forall i \in \mathbf{I}: & \sum_m s_i(m) \leq 1 \\ (iii)\forall m: & p_{m,e^*(m)} + \sum_{i \in \mathbf{I}} s_i(m) \leq v_m \\ (iv)\forall m, f: & s_f(m) \in \{0, 1, ..., D\} \\ (v)\forall m: & p_{m,e^*(m)} \in \{0, 1\} \end{array} \end{array} \tag{2.13}$$

There is no tight delay constraint like the streaming case for this traffic.

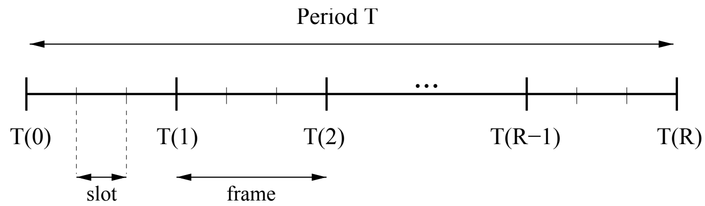


Fig. 2. Different time scales.

Let T be a cache refresh period. We denote the frames in T by $\{l : l \in T\}$ or $l = T(0), ..., T(R-1)$ (see Figure 2). $A = (a^I_{n,j}(l), a^E_n(l) : l \in T$ and $n, j)$ refers to the arrivals of the requests during this period, where the inelastic request $a^I_{n,j}(l)$ and the elastic request $a^E_n(l)$ vectors are defined as in the unit-period case.

1. Feasibility and Capacity Regions

As before, we first find the set of feasible schedules. In order to simplify the argument, we change the notation used in the previous case and define *FeasRegP(P)* (read *feasiblity region subject to presence*) to be the set of all feasible schedules, when the content of the caches is given in $P = (p_{m,f} : m, f)$:

$$S = (s_f(m,t) : f \in \mathbf{F}, 1 \leq m \leq M, 1 \leq t \leq D)$$

$$\text{such that:}$$

$$\begin{aligned} \textstyle\sum_f s_f(m,t) \leq 1 &\qquad \text{for each } m \text{ and } t \\ \textstyle\sum_m \sum_t s_i(m,t) \leq 1 &\qquad \text{for each } i \in \mathbf{I} \\ s_f(m,t) \leq p_{m,f} &\qquad \text{for each } m, t \text{ and } f \end{aligned} \tag{2.14}$$

For the presence vector P, in addition to the cache capacity constraint, we need to formulate the partial refresh scheme at the intermediate frames. Therefore for each cache m and frame $l \in T$:

$$\begin{aligned} \textstyle\sum_f p_{m,f}(l) &\leq v_m \\ \textstyle\sum_f |p_{m,f}(l) - p_{m,f}(l-1)| &\leq 2r_m \quad \text{for } l \neq T(0), \end{aligned} \tag{2.15}$$

where r_m is the maximum number of chunks, cache m can replace in an inter-refresh frame.

For the presence pattern P, in addition to the cache capacity constraint, we need to formulate the partial refresh scheme at the intermediate frames. Therefore for each cache m and frame $l \in T$:

$$\begin{aligned} \textstyle\sum_f p_{m,f}(l) &\leq v_m \\ \textstyle\sum_f |p_{m,f}(l) - p_{m,f}(l-1)| &\leq 2r_m \quad \text{for } l \neq T(0). \end{aligned} \tag{2.16}$$

where r_m is the maximum number of chunks, cache m can replace in an inter-refresh frame.

Capacity region, the convex hull of allowable requests, is defined in a similar way to the unit-period case with a key difference. In the previous case, placement and service scheduling are done for each frame independently of the previous and future frames. Note that in the periodic case, because of the partial refresh scheme, at each inter-refresh frame only a subset of all possible placements is feasible. Therefore, what is loaded in the cache at a time will directly impact the service of the next frames. Based on this observation, the capacity region of the periodic refresh scenario is defined as follows.

Definition 2. (capacity region of the periodic cache replacement scenario) *For the given vectors of cache capacities V and average channel capacities C, a set of requests $\Lambda = (\lambda_n^e, \lambda_{n,j}^I$ [chunks/frame]$, \eta_{n,j} : n, j, e)$ is said to be allowable if the following holds: there exists a policy $\mathbf{P}^*$ which, according to the arrivals profile A in each cache period*

T, provides a service that is greater than the expected service on average:

$$\begin{aligned} \eta_{n,j}\lambda^I_{n,j} \le \bar{\mu}^I_{n,j} \quad & \textit{for each client } e_{n,j} \\ \lambda^e_n \le \bar{\mu}^e_n \quad & \textit{for each } e \in \mathbf{E} \textit{ and } n \end{aligned} \tag{2.17}$$

The average inelastic service $\bar{\mu}^I_{n,j} =$

$$\tfrac{1}{R}\mathbb{E}_A\left[\sum_{l\in T}\mathbb{E}_{P(l)}\left[\mathbb{E}_{S(l)}\left[\sum_{i,m,t\in l} a^I_{n,j}(i,l)\bar{c}^m_n s_i(m,t)\right]\right]\right]$$

and the average elastic service $\bar{\mu}^e_n$ can be written as

$$\bar{\mu}^e_n = \frac{1}{R}\mathbb{E}_A\left[\sum_{l\in T}\mathbb{E}_{P(l)}\left[\mathbb{E}_{S(l)}\left[\sum_{m,t\in l}\bar{c}^m_n s_e(m,t)\right]\right]\right].$$

In which, $\mathbb{E}_{S(l)}[.]$ and $\mathbb{E}_{P(l)}[.]$ are expectations over all feasible schedules and placements with respect to the probability distributions implied by $\mathbf{P}^*$.

A set of requests is said to be *strictly allowable*, if (2.17) holds with strict inequalities. In what follows, we will show that the max-weight algorithm can stabilize the request and deficit queues for any set of strictly allowable requests.

2. Max-Weight Algorithm: Throughput Optimality and Implementation

Similar to the unit-period case, we attain results on the throughput optimality of a max-weight algorithm (presented in **Algorithm 2**) for the periodic scenario.

Theorem 3. *In the periodic replacement scheme, for any set of strictly allowable requests, the max-weight algorithm (in* **Algorithm 2***) stabilizes the request and deficit queues and hence is throughput optimal.*

Proof. Recall that the unit-period scenario is a special case of the periodic scheme (for $R = 1$), hence the following argument can be used to prove Theorem 2 as well.

We will use the Lyapunov technique, Theorem 1 (for further details see [12]), to prove the throughput optimality. Let the Lyapunov function be

$$L(k) = 1/2 \sum_{n,e} q_n^{e\,2}(k^+) + 1/2 \sum_{n,j} (\max(d_{n,j}(k^+), 0))^2.$$

The max-weight algorithm results in an expected drift

$$\mathbb{E}\left[L(k+R) - L(k) \mid \text{state of the system at time } k\right]$$

which is negative except in a finite subset of the state space. Therefore, by using the Lyapunov stability criterion, the max-weight algorithm stabilizes both the request and the deficit queues.

The expected drift over a period T of R frames can be written as

$$\begin{aligned}
&\mathbb{E}_{A,C}\left[L(k+R) - L(k) \mid q_n^e(k^+) = q_n^e, d_{n,j}(k^+) = d_{n,j}\right] \\
&= \tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_{n,e} q_n^{e\,2}(k+R^+) - q_n^{e\,2}\right] + \tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_{n,j}(\{d_{n,j}(k+R^+)\}^+)^2 - (\{d_{n,j}\}^+)^2\right] \\
&= \mathbb{E}\left[\textstyle\sum_{n,e} q_n^e \sum_{l\in T} a_n^E(e,l+1)\right] - \mathbb{E}\left[\textstyle\sum_{n,e} q_n^e \sum_{l\in T} \min(\mu_{n,0}(e,l), q_n^e(l^+))\right] \\
&+\tfrac{1}{2}\mathbb{E}\left[\sum_{n,e}(\sum_{l\in T} a_n^E(e,l+1) - \min(\mu_{n,0}(e,l), q_n^e(l^+)))^2\right] \\
&+\tfrac{1}{2}\mathbb{E}\left[\sum_{n,j}(\{d_{n,j} + \sum_{i,l\in T} \tilde{a}_{n,j}^I(i,l+1) - \sum_{l\in T}\mu_{n,j}^I(l)\}^+)^2\right] - \tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_{n,j}(\{d_{n,j}\}^+)^2\right] \\
&\overset{(a)}{\leq} R\textstyle\sum_{n,e} q_n^e\lambda_n^e - \mathbb{E}\left[\textstyle\sum_{n,e,l} q_n^e\mu_{n,0}(e,l)\right] + B_1 \\
&+\tfrac{1}{2}\mathbb{E}\left[\sum_{n,j}(\{d_{n,j}\}^+ + \sum_{i,l\in T} \tilde{a}_{n,j}^I(i,l+1) - \sum_{l\in T}\mu_{n,j}^I(l))^2\right] - \tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_{n,j}(\{d_{n,j}\}^+)^2\right] \\
&= R\textstyle\sum_{n,e} q_n^e\lambda_n^e - \mathbb{E}\left[\textstyle\sum_{n,e,l} q_n^e\mu_{n,0}(e,l)\right] + B_1 \\
&+\mathbb{E}\left[\textstyle\sum_{n,j}\{d_{n,j}\}^+(\sum_{i,l\in T}\tilde{a}_{n,j}^I(i,l+1) - \sum_{l\in T}\mu_{n,j}^I(l))\right] \\
&+\tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_{n,j}(\sum_{i,l\in T}\tilde{a}_{n,j}^I(i,l+1) - \sum_{l\in T}\mu_{n,j}^I(l))^2\right] \\
&= R\textstyle\sum_{n,e} q_n^e\lambda_n^e - \mathbb{E}\left[\textstyle\sum_{n,e,l} q_n^e\mu_{n,0}(e,l)\right] \\
&+R\textstyle\sum_{n,j}\{d_{n,j}\}^+\eta_{n,j}\lambda_{n,j}^I - \mathbb{E}\left[\textstyle\sum_{n,j,l}\{d_{n,j}\}^+\mu_{n,j}^I(l)\right] + B,
\end{aligned} \tag{2.18}$$

in which (a) follows since $(\{X+Y\}^+)^2 \leq (\{X\}^+ + Y)^2$, $B_1 = B_{11} + B_{12}$ and $B =$

$B_1 + B_2$, where,

$$B_{11} = \frac{1}{2}\mathbb{E}\left[\sum_{n,e}(\sum_{l\in T} a_n^E(e,l+1) - \min(\mu_{n,0}(e,l), q_n^e(l^+)))^2\right]$$

$$B_{12} = \mathbb{E}\left[\sum_{n,e} q_n^e \sum_{l\in T}(\mu_{n,0}(e,l) - \min(\mu_{n,0}(e,l), q_n^e(l^+)))\right]$$

$$B_2 = \frac{1}{2}\mathbb{E}\left[\sum_{n,j}(\sum_{l\in T}\sum_{i} \tilde{a}_{n,j}^I(i,l+1) - \mu_{n,j}^I(l))^2\right].$$

The following lemma shows that B has a finite bounded value,

Lemma 1. $B \leq 0.5R^2\sum_n N_n + RD^2M^2N(1+1.5R) + 0.5R^2\sum_{n,e}(\mathbf{a}_n^e)^2$.

Proof. For B_{11}, we have

$$\begin{aligned}
&B_{11} = \tfrac{1}{2}\mathbb{E}\left[\sum_{n,e}(\sum_{l\in T} a_n^E(e,l+1) - \min(\mu_{n,0}(e,l), q_n^e(l^+)))^2\right] \\
&\leq \tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_{n,e}(\sum_{l\in T} a_n^E(e,l+1))^2\right] + \tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_{n,e}(\sum_{l\in T}\min(\mu_{n,0}(e,l), q_n^e(l^+)))^2\right] \\
&\overset{(b)}{\leq} \tfrac{1}{2}\textstyle\sum_{n,e} R^2(\mathbf{a}_n^e)^2 + \tfrac{1}{2}\mathbb{E}\left[\textstyle\sum_n(\sum_{l\in T,e}\min(\mu_{n,0}(e,l), q_n^e(l^+)))^2\right] \\
&\overset{(c)}{\leq} \tfrac{1}{2}\textstyle\sum_{n,e} R^2(\mathbf{a}_n^e)^2 + \tfrac{1}{2}R^2D^2M^2N
\end{aligned}$$

where (b) follows since $a_n^E(e,k) \leq \mathbf{a}_n^e$ for all frames k and the last inequality (c) holds because the total elastic service provided for each frontend during a frame is not greater than the total number of transmissions that can be made (there are M caches and each one can make at most D transmissions during a frame).

Note that $q_n^e(l) \leq q_n^e(l+1) + \mu_{n,0}(e,l) \leq q_n^e(l+1) + DM$, therefore in order to find an upperbound on B_{12} we have

$$\begin{aligned}
&B_{12} = \mathbb{E}\left[\sum_{n,e,l} q_n^e \max(0, \mu_{n,0}(e,l) - q_n^e(l^+))\right] \\
&\leq \mathbb{E}\left[\sum_{n,e,l} q_n^e(l^+) \max(0, \mu_{n,0}(e,l) - q_n^e(l^+))\right] \\
&+\mathbb{E}\left[\sum_{n,e,l} \max(0, \mu_{n,0}(e,l) - q_n^e(l^+))(l - T(0))DM\right] \\
&\overset{(d)}{\leq} \mathbb{E}\left[\sum_{n,e,l} (\mu_{n,0}(e,l))^2\right] + RDM \times \mathbb{E}\left[\sum_{n,e,l} \mu_{n,0}(e,l)\right] \\
&\leq \mathbb{E}\left[\sum_{l,n} (\sum_e \mu_{n,0}(e,l))^2\right] + R^2D^2M^2N \\
&\leq RD^2M^2N(1+R)
\end{aligned}$$

where (d) follows since $l - T(0) \leq R$ for any $l \in T$, $\max(0, \mu_{n,0}(e,l) - q_n^e(l^+)) \leq \mu_{n,0}(e,l)$ and $q_n^e(l^+) \max(0, \mu_{n,0}(e,l) - q_n^e(l^+)) \leq (\mu_{n,0}(e,l))^2$.

Also we have

$$-1 \leq \sum_i \tilde{a}_{n,j}^I(i, l+1) - \mu_{n,j}^I(l) \leq 1$$

which results in

$$B_2 = \frac{1}{2}\mathbb{E}\left[\sum_{n,j} (\sum_{l \in T} \sum_i \tilde{a}_{n,j}^I(i, l+1) - \mu_{n,j}^I(l))^2\right] \leq \frac{1}{2}\sum_{n,j} R^2 \leq \frac{R^2}{2}\sum_{n=1}^{N} N_n.$$

Therefore, $B = B_1 + B_2$ is upperbounded by the value in the lemma. □

Now that we have seen B has a finite value, in order to show the negativeness of the expected drift (in (2.18)), we need to show

$$\begin{aligned}
R\sum_{n,e} q_n^e \lambda_n^e + R\sum_{n,j} \{d_{n,j}\}^+ \eta_{n,j} \lambda_{n,j}^I < \\
\mathbb{E}_{A,C}\left[\sum_{n,e,l} q_n^e \mu_{n,0}(e,l) + \sum_{n,j,l} \{d_{n,j}\}^+ \mu_{n,j}^I(l)\right]
\end{aligned} \tag{2.19}$$

From (2.5) and (2.6), the right hand side of (2.19) can be rephrased as:

$$\mathbb{E}_A\left[\sum_{n,e,l \in T} q_n^e \sum_{m,t \in l} \bar{c}_n^m s_e(m,t)\right] + \mathbb{E}_A\left[\sum_{n,j,l \in T} \{d_{n,j}\}^+ \sum_{i,m,t \in l} a_{n,j}^I(i,l) \bar{c}_n^m s_i(m,t)\right]$$

At the beginning of each refresh period T, given the queue lengths $q_n^e(T(0)^+) =$

q_n^e, $d_j^n(T(0)^+) = d_j^n$ and the arrivals A, the max-weight algorithm solves the following:

$$\begin{aligned} &\max \textstyle\sum_{n,e,l} q_n^e \sum_{m,t\in l} \bar{c}_n^m s_e(m,t) + \sum_{n,j,l}\{d_{n,j}\}^+ \sum_{i,m,t\in l} a_{n,j}^I(i,l)\bar{c}_n^m s_i(m,t) \\ &\text{such that for each } l \in T: \\ &S(l) = (S_f(m,t) : f,m,t \in l) \in \textit{FeasRegP(P(l))} \\ &P(l) \text{ satisfies the conditions in (2.16)} \end{aligned} \tag{2.20}$$

to find an optimal placement $P^*(l)$ and schedule $S^*(l)$ for each $l \in T$. Note that the above maximization is done over all feasible placements $P(l)$ and schedules $S(l) \in$ *FeasRegP(P(l))* for each frame l. Therefore,

$$\begin{aligned} &\textstyle\sum_{n,e,l} q_n^e \sum_{m,t\in l} \bar{c}_n^m s_e^*(m,t) + \sum_{n,j,l}\{d_{n,j}\}^+ \sum_{i,m,t\in l} a_{n,j}^I(i,l)\bar{c}_n^m s_i^*(m,t) \geq \\ &\textstyle\sum_{n,e,l} q_n^e \mathbb{E}_{P(l)}\left[\mathbb{E}_{S(l)}\left[\sum_{m,t\in l} \bar{c}_n^m s_e(m,t)\right]\right] + \\ &\textstyle\sum_{n,j,l}\{d_{n,j}\}^+ \mathbb{E}_{P(l)}\left[\mathbb{E}_{S(l)}\left[\sum_{i,m,t\in l} a_{n,j}^I(i,l)\bar{c}_n^m s_i(m,t)\right]\right]. \end{aligned}$$

in which $\mathbb{E}_{S(l)}[.]$ and $\mathbb{E}_{P(l)}[.]$ are expectation with respect to the probability distributions implied by the fulfilling policy $\mathbf{P}^*$ (in Definition 2). By taking expectation over the sequences of the arrivals A on both sides of the previous inequality, we will get:

$$\begin{aligned} &\mathbb{E}_A\left[\textstyle\sum_{n,e,l} q_n^e \sum_{m,t\in l} \bar{c}_n^m s_e^*(m,t)\right] + \mathbb{E}_A\left[\sum_{n,j,l}\{d_{n,j}\}^+ \sum_{i,m,t\in l} a_{n,j}^I(i,l)\bar{c}_n^m s_i^*(m,t)\right] \\ &\geq \textstyle\sum_{n,e} q_n^e \mathbb{E}_A\left[\sum_{l\in T} \mathbb{E}_{P(l)}\left[\mathbb{E}_{S(l)}\left[\sum_{m,t\in l} \bar{c}_n^m s_e(m,t)\right]\right]\right] + \\ &\textstyle\sum_{n,j}\{d_{n,j}\}^+ \mathbb{E}_A\left[\sum_{l\in T} \mathbb{E}_{P(l)}\left[\mathbb{E}_{S(l)}\left[\sum_{i,m,t\in l} a_{n,j}^I(i,l)\bar{c}_n^m s_i(m,t)\right]\right]\right] \\ &> R\textstyle\sum_{n,e} q_n^e \lambda_n^e + R\sum_{n,j}\{d_{n,j}\}^+ \eta_{n,j}\lambda_{n,j}^I \end{aligned}$$

where the last inequality follows from the assumption of the strict allowablity of the requests ((2.17) in Definition 2). Therefore, one can find a small enough $\epsilon > 0$ such that

$$\begin{aligned} &\mathbb{E}_A\left[\textstyle\sum_{n,e,l} q_n^e \sum_{m,t\in l} \bar{c}_n^m s_e^*(m,t)\right] + \mathbb{E}_A\left[\sum_{n,j,l}\{d_{n,j}\}^+ \sum_{i,m,t\in l} a_{n,j}^I(i,l)\bar{c}_n^m s_i^*(m,t)\right] \\ &\geq R\textstyle\sum_{n,e} q_n^e \lambda_n^e(1+\epsilon) + R\sum_{n,j}\{d_{n,j}\}^+ \eta_{n,j}\lambda_{n,j}^I(1+\epsilon) \end{aligned}$$

Considering the above inequality in (2.18) results in

$$\begin{aligned}&\mathbb{E}\left[L(k+R)-L(k)\middle|q_n^e(k^+)=q_n^e, d_{n,j}(k^+)=d_{n,j}\right]\\&\leq B-R\epsilon(\sum_{n,e}q_n^e+\sum_{n,j}\{d_{n,j}\}^+).\end{aligned} \tag{2.21}$$

Therefore the max-weight algorithm (in (2.20)) results in a negative Lyapunov drift (except in a finite subset of the state space) over a refresh period. This concludes that the max-weight algorithm stabilizes both the deficit and the request queues and is throughput optimal. □

Similar to Corollary 1, we can bound the queue lengths at the steady state for the periodic case as follows,

Corollary 2. *Sum of the average request and the deficit queue lengths at the steady state is bounded by (for some $\epsilon > 0$, the heavy traffic parameter)*

$$\sum_{n,j}\mathbb{E}\left[\{d_{n,j}\}^+\right]+\sum_{n,e}\mathbb{E}\left[q_n^e\right]\leq (R\sum_n N_n + D^2M^2N(2+3R)+R\sum_{n,e}(\mathbf{a}_n^e)^2)/2\epsilon.$$

Proof. Note that the proof of Corollary 1, which is a special case ($R = 1$) of this corollary, follows the same argument. By taking expectation from both sides of (2.21) with respect to the queue lengths, we get the following,

$$\mathbb{E}[L(k+R)]-\mathbb{E}[L(k)]\leq B-R\epsilon(\sum_{n,j}\mathbb{E}[\{d_{n,j}(k^+)\}^+]+\sum_{n,e}\mathbb{E}[q_n^e(k^+)]).$$

Since the Markov chain is shown to be positive recurrent, it converges to the steady state and as $k\to\infty$ we have $\mathbb{E}[L(k+R)]=\mathbb{E}[L(k)]$. Hence at the steady state,

$$\sum_{n,j}\mathbb{E}\left[\{d_{n,j}\}^+\right]+\sum_{n,e}\mathbb{E}\left[q_n^e\right]\leq \frac{B}{R\epsilon} \tag{2.22}$$

□

In what follows, we will show that the max-weight algorithm which solves (2.20) can be implemented as the policy presented in **Algorithm 2** for the periodic case and **Algorithm 1** for the unit-period scenario.

Algorithm 2 Periodic Scenario: Max-Weight Algorithm

At the beginning of a period T: given $d_{n,j}(T(0)), q_n^e(T(0))$ **and** $A = (A_{n,j} : n, j)$ **compute**,

$d_{n,j}$, q_n^e, $w_{m,e}^E$, $e^*(m)$ and $w_{m,i}^I(l)$ (for each frame $l \in T$) as in Algorithm 1

Solve the following maximization problem:

$$\max \sum_{l\in T} \sum_m w^E_{m,e^*(m)} p_{m,e^*(m)}(l) \hat{s}_{e^*(m)}(m,l) + \sum_{l\in T} \sum_{i,m} w^I_{m,i}(l) p_{m,i}(l) \hat{s}_i(m,l)$$

such that for each $l \in T$:

$$
\begin{array}{ll}
(i)\forall m : & \sum_f \hat{s}_f(m,l) \le 1 \\
(ii)\forall i \in \mathbf{I} : & \sum_m \hat{s}_i(m,l) \le 1 \\
(iii)\forall m : & p_{m,e^*(m)}(l) + \sum_{i\in\mathbf{I}} p_{m,i}(l) \le v_m \\
(iv)\forall l > T(0) : & \sum_i |p_{m,i}(l) - p_{m,i}(l-1)| + |p_{m,e^*(m)}(l) - p_{m,e^*(m)}(l-1)| \le 2r_m \\
(v)\forall m, f : & \hat{s}_f(m,l) \in \{0,1,...,D\} \\
(vi)\forall m, f : & p_{m,f}(l) \in \{0,1\}
\end{array}
$$

Below, we repeat the maximization in (2.20) with all its constraints,

$$\max \sum_{n,e,l} q_n^e \sum_{m,t\in l} \bar{c}_n^m s_e(m,t) + \sum_{n,j,l} \{d_{n,j}\}^+ \sum_{i,m,t\in l} a^I_{n,j}(i,l) \bar{c}_n^m s_i(m,t)$$

such that for each $l \in T$:

$$
\begin{array}{ll}
(i)\forall (t\in l), m : & \sum_f s_f(m,t) \le 1 \\
(ii)\forall i \in \mathbf{I} : & \sum_m \sum_{t\in l} s_i(m,t) \le 1 \\
(iii)\forall (t\in l), m, f : & s_f(m,t) \le p_{m,f}(l) \\
(iv)\forall m : & \sum_f p_{m,f}(l) \le v_m \\
(v)\forall l > T(0) : & \sum_f |p_{m,f}(l) - p_{m,f}(l-1)| \le 2r_m \\
(vi)\forall (t\in l), m, f : & s_f(m,t), p_{m,f}(l) \in \{0,1\}
\end{array}
$$

In a couple of steps, we transfer the above maximization problem into a simpler form.

Observation 1. *From* (iii) *for* $t \in l$*:* $s_f(m,t) = 1$ *only if* $p_{m,f}(l) = 1$*. Therefore we can replace* $s_f(m,t)$ *by* $p_{m,f}(l)s_f(m,t)$ *in the objective function. Also since the objective is a linear function of* $s_f(m,t)$ *and it is a maximization problem, the condition* (iii) *will be redundant.*

Observation 2. *Let* $\hat{s}_f(m,l) = \sum_{t \in l} s_f(m,t)$*. The constraints in the max-weight problem can be modified to be expressed in terms of* $\hat{s}_f(m,l)$*:*
(i) *requires that each cache transmits at most one chunk in each slot. This is equivalent to the following condition which says at most* D *transmissions can be made by each cache during a frame:*

$$\sum_f \hat{s}_f(m,l) \leq D \quad \textit{for each } m$$

(ii) *can be accordingly replaced by*

$$\sum_m \hat{s}_i(m,l) \leq 1 \quad \textit{for each } i \in \mathbf{I}$$

The previous two observations suggest the following equivalent formulation of the max-weight problem:

$$\max \sum_{e,m,l} w^E_{m,e} p_{m,e}(l)\hat{s}_e(m,l) + \sum_{i,m,l} w^I_{m,i}(l) p_{m,i}(l)\hat{s}_i(m,l)$$

$$\text{such that for each } l \in T:$$

$$\begin{aligned}
&(i)\forall m: && \sum_f \hat{s}_f(m,l) \leq D \\
&(ii)\forall i \in \mathbf{I}: && \sum_m \hat{s}_i(m,l) \leq 1 \\
&(iii)\forall m: && \sum_f p_{m,f}(l) \leq v_m \\
&(iv)\forall l > T(0): && \sum_f |p_{m,f}(l) - p_{m,f}(l-1)| \leq 2r_m \\
&(v)\forall m,f: && \hat{s}_f(m,l) \in \{0,1,...,D\}, p_{m,f}(l) \in \{0,1\}
\end{aligned}$$

where $w^E_{m,e} = \sum_n q^e_n \bar{c}^m_n$ and $w^I_{m,i}(l) = \sum_{n,j} \{d_{n,j}\}^+ a^I_{n,j}(i,l)\bar{c}^m_n$.

Observation 3. *Suppose* $S^* = (s_f^*(m,l) : m, f, l)$ *and* $P^* = (p_{m,f}^*(l) : m, f, l)$ *are the optimal solutions to the above max-weight problem. If* $p_{e,m}^*(l) = p_{e',m}^*(l') = 1$ *and* $s_e^*(m,l), s_{e'}^*(m,l') > 0$ *for some* m*, elastic chunks* e *and* e' *and frames* l *and* l'*, then we require* $w_{m,e}^E = w_{m,e'}^E$*. Because otherwise providing* $s_e^*(m,l) + s_{e'}^*(m,l')$ *service for* $\hat{e} = argmax(w_{m,e}^E, w_{m,e'}^E)$ *would result in a greater objective value and hence* (S^*, P^*) *would not be optimal. Therefore for large enough queue lengths, at each frame* l *each cache* m *suffices to load at most one elastic content* $e^*(m) = RAND\{arg\max_{e'}(w_{m,e'}^E)\}$*, which is chosen uniformly at random among the elastic content with the largest weight,*

$$p_{m,e}(l) = \hat{s}_e(m,l) = 0, \quad \text{for each } e \neq e^*(m).$$

After applying further simplifications suggested by the above observation, the max-weight problem changes to

$$\max \textstyle\sum_{m,l\in T} w_{m,e^*(m)}^E p_{m,e^*(m)}(l)\hat{s}_{e^*(m)}(m,l) + \sum_{i,m,l\in T} w_{m,i}^I(l) p_{m,i}(l)\hat{s}_i(m,l)$$

$$\text{such that for each } l \in T:$$

$$\begin{aligned}
&(i)\forall m: && \textstyle\sum_f \hat{s}_f(m,l) \leq D \\
&(ii)\forall i \in \mathbf{I}: && \textstyle\sum_m \hat{s}_i(m,l) \leq 1 \\
&(iii)\forall m: && p_{m,e^*(m)}(l) + \textstyle\sum_{i\in\mathbf{I}} p_{m,i}(l) \leq v_m \\
&(iv)\forall l > T(0): && \textstyle\sum_i |p_{m,i}(l) - p_{m,i}(l-1)| + \\
& && |p_{m,e^*(m)}(l) - p_{m,e^*(m)}(l-1)| \leq 2r_m \\
&(v)\forall m, f: && \hat{s}_f(m,l) \in \{0,1,...,D\}, p_{m,f}(l) \in \{0,1\}
\end{aligned}$$

Observation 4. *For the unit-period case, it is correct to assume* $s_i^*(m,l) = 1$ *when* $p_{m,i}^*(l) = 1$*. Because otherwise* $(s_i^*(m,l), p_{m,i}^*(l)) = (0,1)$ *would result in the same objective value as* $(s_i^*(m,l), p_{m,i}^*(l)) = (0,0)$*, i.e., loading chunk* i *in cache* m *is not beneficial. This suggests to consider* $\hat{s}_i(m) = p_{m,i}(l)$ *in our problem, which simplifies the max-weight algorithm even more for the unit-period case.*

Next, we present a more efficient way to implement the max-weight algorithm, based on the idea of dynamic programming. First we define two new notations for each $l \in T$:

1.

$$w(P(l)) = \max \sum_m w^E_{m,e^*(m)} p_{m,e^*(m)}(l) \hat{s}_{e^*(m)}(m,l) + \sum_{i,m} w^I_{m,i}(l) p_{m,i}(l) \hat{s}_i(m,l)$$

$$\text{such that:}$$

$$\begin{aligned} \forall m: &\quad \textstyle\sum_f \hat{s}_f(m,l) \leq 1 \\ \forall i \in \mathbf{I}: &\quad \textstyle\sum_m \hat{s}_i(m,l) \leq 1 \\ \forall m, f: &\quad \hat{s}_f(m,l) \in \{0, 1, ..., D\} \end{aligned}$$

Note that $w(P(l))$ can be found in polynomial time, since the corresponding maximization is an assignment problem [13].

2. $W(P(l))$: the maximum total weight that can be achieved during frames $k \geq l$ given $P(l)$.

Therefore, the objective of the max-weight algorithm is to find

$$\begin{gathered} \max_{P(T(0))} W(P(T(0))) \\ \text{such that for each } m: \\ \textstyle\sum_f p_{m,f}(T(0)) \leq v_m \\ p_{m,e}(T(0)) = 0 \text{ for } e \neq e^*(m) \end{gathered} \tag{2.23}$$

It is straightforward to verify that

$$\begin{gathered} W(P(l)) = w(P(l)) + \max_{P(l+1)} W(P(l+1)) \\ \text{such that for each } m: \\ \textstyle\sum_f p_{m,f}(l+1) \leq v_m \\ \textstyle\sum_f |p_{m,f}(l+1) - p_{m,f}(l)| \leq 2r_m. \end{gathered} \tag{2.24}$$

The modified max-weight algorithm is presented below. This algorithm returns the

Algorithm 3 DP-based Max-weight Algorithm

step 1: Compute $w(P(T(R-1)))$ for all feasible $P(T(R-1))$

step 2:

for $l = T(R-2)$ down to $T(0)$ **do**

Compute $W(P(l))$ for all feasible $P(l)$'s as in (2.24).

end for

step 3: Return $W^*(P(T(0)))$ as in (2.23)

maximum value of the max-weight algorithm (in **Algorithm 2**). By keeping track of *argmax* values in steps 2 and 3, the corresponding optimal placements and service at each frame are provided as well.

E. Inelastic Caching Problem with Content Expiry

In this section, we consider an inelastic caching problem where the chunks expire after some time. We assume that the life time of an inelastic chunk is equal to the length of a frame. Hence, we can cache a chunk only for the duration of a frame after which the content will not be useful any longer. This model is compatible with real-time streaming of live events. We consider a new model for cache refresh cost that is consistent with this scenario, in which loading a cache incurs some cost proportional to the number of fetched chunks. Our objective is to find a policy which minimizes the long-time average expected cost of cache replacement while stabilizing the deficit queues in the system. First, we formally define our model and express the corresponding capacity region. The optimal max-weight algorithm will be presented next.

1. Model and Capacity Region

In this model, we only consider inelastic traffic. For simplicity, we aggregate all requests in each cluster n at its corresponding frontend and use $a_i^n(k)$ to denote the number of requests for chunk i in cluster n, during frame k. η_i^n and $\mu_i^n(k)$ respectively denote the delivery ratio associated with and the service during frame k to chunk i in cluster n. For the sake of argument, assume a Bernoulli arrival model, i.e., $a_i^n(k) \in \{0,1\}$ and $\mathbb{E}[a_i^n(k)] = \lambda_i^n$. Like before, for the channels, an ON-OFF stochastic model with unknown states is considered. We assume loading cache m during frame k incurs a unit cost $\gamma_m(k)$ per chunk. $\gamma_m(k)$ is a random variable identically and independently distributed over frames, with the average of $\mathbb{E}[\gamma_m(k)] = \bar{\gamma}_m$ for all k. The total cost of replacing new chunks in the caches, at frame k, is denoted by $\Gamma(k)$, where

$$\Gamma(k) = \sum_{m,i} \gamma_m(k) p_{m,i}(k) \tag{2.25}$$

and $p_{m,i}(k)$ denotes the presence of a fresh chunk of content i in cache m for the k^{th} frame.

The feasibility region, set of all feasible schedules, is defined in the same way as in Section 1. The capacity region can be found by considering the following lemma.

Lemma 2. *For the i.i.d arrival processes $A(k)$ and i.i.d loading costs $\{\gamma_m(k):\ 1 \leq m \leq M\}$, there exists a randomized stationary policy P^* such that*

- *At frame k, based only on the realization of $A(k), \{\gamma_m(k):\ 1 \leq m \leq M\}$, it chooses a feasible schedule S with probability*

$$\mathbb{P}^*(S|A(k), \{\gamma_m(k):\ 1 \leq m \leq M\})$$

(maintaining $\sum_S \mathbb{P}^*(S|A(k), \{\gamma_m(k):\ 1 \leq m \leq M\}) = 1$).

- *On average P^* provides enough service, that is*

$$\mathbb{E}^*[\mu_i^n(k)] = \mathbb{E}^*[\sum_{m,t\in k} a_i^n(k)c_n^m(t)s_i(m,t)] \geq \eta_i^n\lambda_i^n \tag{2.26}$$

  in which the expectation is with respect to the distributions of the arrivals, loading costs, channels and $\mathbb{P}^*(S|A(k),\{\gamma_m(k):\ 1\leq m\leq M\})$.

- *P^* also achieves the minimum average expected cost Γ^*:*

$$\mathbb{E}^*[\Gamma(k)] = \Gamma^* \tag{2.27}$$

The proof follows the same argument as in [10] and is omitted for brevity.

Now the capacity region is the convex hull of all sets of requests for which such randomized stationary policies exist.

2. Optimal Policy

To keep track of inelastic requests, we use deficit queues as described before. The deficit queue length $d_i^n(k)$ associated with inelastic chunk i in cluster n follows the dynamic below:

$$d_i^n(k+1) = d_i^n(k) + \tilde{a}_i^n(k) - \mu_i^n(k). \tag{2.28}$$

Our framework to find an optimal policy is to minimize an upperbound on expected (*Lyapunov drift + cost*) at each frame. The following lemma gives this upperbound.

Lemma 3. *For the Lyapunov function $L(k) = \frac{1}{2}\sum_{n,i}(\{d_i^n(k)\}^+)^2$, we have:*

$$\begin{aligned}&\mathbb{E}[L(k+1)-L(k)|d_i^n(k)] + Y\mathbb{E}[\Gamma(k)|d_i^n(k)]\\ &\leq B + Y\mathbb{E}[\Gamma(k)|d_i^n(k)] + \textstyle\sum_{n,i}\{d_i^n(k)\}^+\eta_i^n\lambda_i^n\\ &-\textstyle\sum_{n,i}\{d_i^n(k)\}^+\mathbb{E}[\mu_i^n|d_i^n(k)]\end{aligned} \tag{2.29}$$

where $Y > 0$ is some control parameter and $B = \frac{N|\mathbf{I}|}{2}$.

For brevity and since this lemma is a special case of the analysis in Section 2, the proof is omitted.

At each frame k, given the arrivals and the costs $\gamma_m(k)$, we minimize

$$Y \sum_{m,i} \gamma_m(k) p_{m,i}(k) - \sum_{n,i} \{d_i^n(k)\}^+ \mu_i^n(k) \tag{2.30}$$

over the feasibility region to get the max-weight schedule $\hat{p}_{m,i}(k)$ and $\hat{\mu}_i^n(k)$. Therefore, we will have

$$\begin{aligned} B + Y\mathbb{E}[\textstyle\sum_{m,i} \gamma_m(k)\hat{p}_{m,i}(k)|d_i^n(k)] \\ + \textstyle\sum_{n,i} \{d_i^n(k)\}^+ \eta_i^n \lambda_i^n - \sum \{d_i^n(k)\}^+ \mathbb{E}[\hat{\mu}_i^n | d_i^n(k)] \\ \leq B + Y\mathbb{E}^*[\Gamma(k)] + \textstyle\sum_{n,i} \{d_i^n(k)\}^+ (\eta_i^n \lambda_i^n - \mathbb{E}^*[\mu_i^n]) \end{aligned} \tag{2.31}$$

where the right hand side is what the randomized stationary policy P^* achieves. If the requests are strictly allowable (please refer to Definition 1), then for all n and i, $\exists \epsilon > 0$ such that $\mathbb{E}^*[\mu_i^n] \geq \eta_i^n \lambda_i^n + \epsilon$. Considering this fact and $\mathbb{E}^*[\Gamma(k)] = \Gamma^*$ in (2.31) and (2.29) gives:

$$\begin{aligned} \mathbb{E}[\hat{L}(k+1) - \hat{L}(k)|d_i^n(k)] + Y\mathbb{E}[\textstyle\sum_{m,i} \gamma_m(k)\hat{p}_{m,i}(k)|d_i^n(k)] \\ \leq B + Y\Gamma^* - \epsilon \textstyle\sum_{n,i} \{d_i^n(k)\}^+ \end{aligned} \tag{2.32}$$

where $\hat{L}(k)$ is the value of the Lypaunov function at frame k when we use the max-weight schedule.

Corollary 3. *The max-weight schedule, derived from minimizing (2.30), results in a negative Lyapunov drift (except in a finite subset of the state space) and hence stabilizes the deficit queues.*

Corollary 4. *The max-weight schedule, derived from minimizing (2.30), results in a long-time average expected cost $\hat{\Gamma}$ which deviates from the minimum cost Γ^* by an* *amount less than B/Y:*

$$\hat{\Gamma} = \lim_{K\to\infty} \frac{1}{K+1} \sum_{k=0}^{K} \mathbb{E}[\hat{\Gamma}(k)] \leq \Gamma^* + \frac{B}{Y} \tag{2.33}$$

Observation 5. *Note that by increasing the control parameter Y, we can achieve an average expected cost which is arbitrarily close to the minimum cost. However, this will potentially lead to larger expected deficit queue lengths. Hence, there is a trade-off between the cost and the average deficit queue lengths.*

Proof. (Corollary 4) Note that (2.32) holds for any frame k. From both sides of this inequality, we take expectation with respect to the distribution of the deficit queues $d_i^n(k)$ to get:

$$\mathbb{E}[\hat{L}(k+1) - \hat{L}(k)] + Y\mathbb{E}[\sum_{m,i} \gamma_m(k)\hat{p}_{m,i}(k)] \leq B + Y\Gamma^* \tag{2.34}$$

Assume the initial deficit queue lengths are zero, i.e., $\hat{L}(0) = 0$. Now sum both sides of (2.34) from $k = 0$ to $k = K$ and divide by $K + 1$ to get

$$\frac{\mathbb{E}[\hat{L}(K+1)]}{K+1} + \frac{Y}{K+1} \sum_{k=0}^{K} \mathbb{E}[\sum_{m,i} \gamma_m(k)\hat{p}_{m,i}(k))] \leq B + Y\Gamma^* \tag{2.35}$$

By letting K tend to infinity and noting that $\mathbb{E}[\hat{L}(K+1)]$ is a bounded positive value for each K, the desired result in (2.33) is derived. □

The previous arguments in implementing the max-weight algorithm (Section 2) can also be applied to this problem and will result in **Algorithm 4**.

F. Simulation Results

In this section, we evaluate the performance of the proposed max-weight algorithm, a distributed greedy policy and an iterative version of the max-weight by simulating a wireless content distribution network using Matlab. We assume that the caches

Algorithm 4 Inelastic Traffic with Expiry: Max-Weight

At the beginning of each frame k: given the deficit queue lengths $d_i^n(k)$, arrivals $A(k)$ and refresh costs $\gamma_m(k)$ compute,

$w_i(m) = \sum_n \{d_i^n(k)\}^+ a_i^n(k)\bar{c}_n^m - Y\gamma_m(k)$

Solve the following maximization problem:

$$\max \sum_{m,i} w_i(m) s_i(m)$$

$$\text{subject to}$$

$$(i)\forall m: \quad \sum_i s_i(m) \le \min(D, v_m)$$

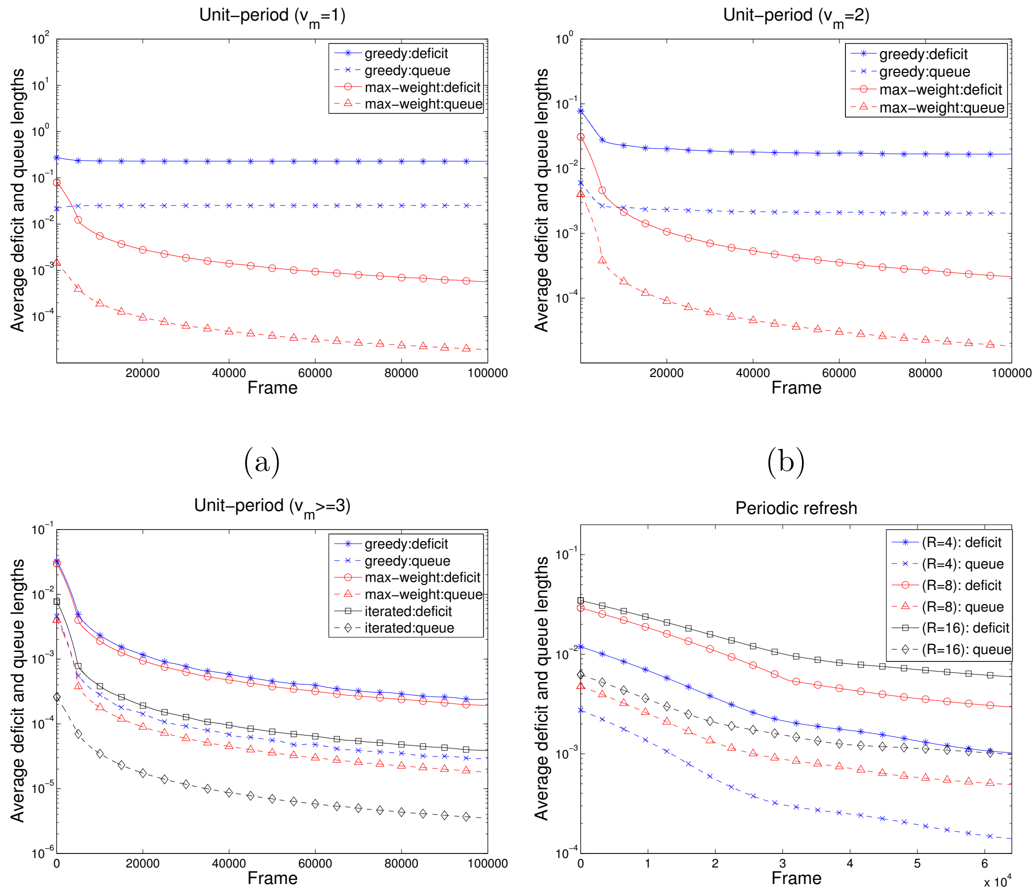
$$(ii)\forall i: \quad \sum_m s_i(m) \le 1$$

operate in distinct frequency spectra and hence there is no interference issue. We further assume that the popularity of the inelastic chunks is distributed uniformly.

In Figures 3(a)-3(d), the evolution of the normalized average queue lengths ($\frac{1}{k\sum_n N_n}\sum_{n,j} d_{n,j}(k)$ and $\frac{1}{|\mathbf{E}|k\sum_n N_n}\sum_{n,e} q_n^e(k)$) are displayed over the frames k. These values must converge to zero for a stabilizing algorithm.

For the large networks, the max-weight algorithm can be very hard to implement. Therefore, the performance of a distributed greedy algorithm is also simulated. In this algorithm each cache, independent of the others, loads and serves the chunks based on their weights (as computed in Algorithm 1). The simulation results suggest that although the greedy algorithm is not throughput optimal, it can provide enough service if the requests lie sufficiently inside the capacity region (e.g. in Figure 3(c)).

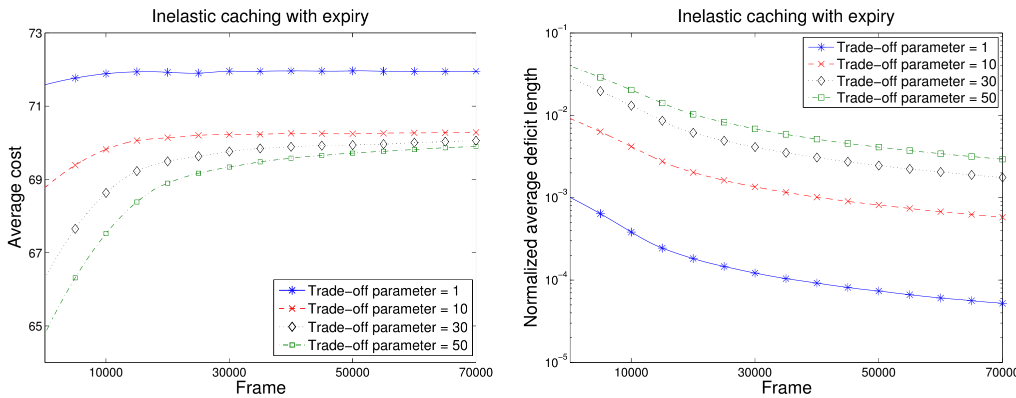
An iterative version of the max-weight, with the possibility of rescheduling at each time slot instead of each frame, is also simulated. It can be shown that with a non-zero probability, the iterative max-weight algorithm can achieve strictly smaller average queue lengths compared to the original algorithm (Figure 3(c)).



(a) (b)

(c) (d)

Fig. 3. Evolution of the average queue lengths. (a)-(c): unit-refresh with ($|\mathbf{E}| = |\mathbf{I}| = 5$, $N = M = 4$, $N_n = 3$, $\bar{c}_n^m = 0.9$, $\eta_{n,j} = 0.5$, $\lambda_{n,j}^I = 0.9$, $a_n^E(e,k)$~$B(N_n, 0.9)$, $D = 3$ and different cache capacity values v_m), (d): max-weight in a periodic case with ($|\mathbf{I}| = 4$, $|\mathbf{E}| = 3$, $N = M = 3$, $N_n = 3$, $\bar{c}_n^m = 0.9$, $\eta_{n,j} = 0.5$, $\lambda_{n,j}^I = 0.6$, $a_n^E(e,k)$~$B(N_n, 0.6)$, $D = 2$, $r_m = 1$, $v_m = 2$ and different periods R)



(a) (b)

Fig. 4. Inelastic traffic with expiry, the evolution of the average deficit queue length and cost with respect to different trade-off parameter values ($|\mathbf{I}| = 12$, $N = M = 3$, $\bar{c}_n^m = 0.9$, $\eta_i^n = 0.6$, $\lambda_i^n = 0.9$, $D = 3$, $v_m = 3$, γ_m~$\mathcal{N}(10, 1)$)

Since the service to a deficit queue is subject to having a new request arrival, in general the average length of the deficit queues is smaller than the request queues. Also it is intuitive that by increasing the capacity of the caches, more service can be provided (Figures 3(a)-3(c)). Also as shown in corollary 2, the average queue lengths increase by the refresh period R. (Figures 3(d))

The performance of **Algorithm 4** that is aimed towards real-time streaming is evaluated in Figures 4(a) and 4(b). As expected, the average deficit queue length increases by the trade-off parameter Y, while increasing Y lets the average cost decrease and tend to its minimum value.

G. Summary and Future Work

In this chapter, we studied algorithms for content placement and scheduling in wireless broadcast networks. 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, and also showed that iterative versions of the algorithm can enhance performance still further. We also incorporated the cost of loading caches in our problem with considering two different models. First, periodic cache refresh, in which the cost is modeled as the period of refreshing the caches. Similar results to the basic model, unit-period case, were derived for this variation. Second, inelastic caching with expiry, in which 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 achieve an average cost which is arbitrarily close to the minimum cost.

For future work, one can think of extending these results to a multi-hop caching network. It may also be worthwhile to consider more general QoS requirements like heterogeneous deadlines. In Section F, we proposed a simple decentralized greedy algorithm whose performance was evaluated by simulations. Finding more efficient algorithms, that can be implemented in a distributed fashion, seems to be another appealing direction for extending this work.

In the content distribution problem studied in this chapter, we did not consider any type of coding over the transmitted packets. We might be able to enhance the performance of our CDN by utilizing the broadcast nature of the system more efficiently. Note that a packet broadcast for a particular user will be heard by its neighboring nodes. In the current framework, we just ignored the received packets which were not requested. However, we could potentially exploit these overheard transmissions by employing coding over contents. Network coding is a popular technique to efficiently use such overheard transmissions, especially in wireless networks. In the next chapter, we will study the effect of network coding in a simple wireless relay network. We will specifically look at the performance of network coding from a delay perspective and propose a simple algorithm to optimally trade off delay and throughput in network coding-enabled relay networks.

CHAPTER III

OPTIMAL DELAY-THROUGHPUT TRADE-OFF IN THE NETWORK CODING-ENABLED RELAY NETWORKS

In recent years, there has been a growing interest in the applications of network coding techniques in wireless communication networks. It was shown that the network coding can result in significant improvements in the performance of multi-hop wireless networks. For example, consider a wireless network coding scheme depicted in Fig. 5(a). Here, wireless nodes 1 and 2 need to exchange packets x_1 and x_2 through a relay node (node 3). A simple *store-and-forward* approach needs four transmissions. However, the network coding solution uses a *store-code-and-forward* approach in which the two packets x_1 and x_2 are combined by means of a bitwise XOR operation at the relay and broadcast to nodes 1 and 2 simultaneously. Nodes 1 and 2 can then decode this coded packet to obtain the packets they need.

[14] introduces the strategy of reverse carpooling that allows two information flows traveling in opposite directions to share a path. Fig. 5(b) shows an example of two connections, from n_1 to n_4 and from n_4 to n_1 that share a common path (n_1, n_2, n_3, n_4). The wireless network coding approach results in a significant (up to 50%) reduction in the number of transmissions for two connections that use reverse carpooling. In particular, once the first connection is established, the second connection (of the same rate) can be established in the opposite direction with little additional cost.

In this work [1], we focus on design and analysis of scheduling protocols that exploit the fundamental trade-off between the number of transmissions and delay in

[1]This work was done in collaboration with other students, Yu-Pin Hsu and Solairaja Ramasamy

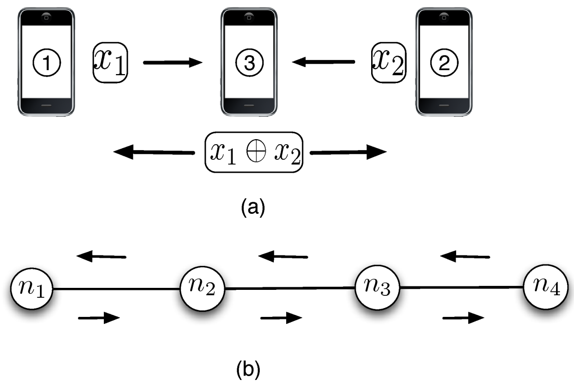


Fig. 5. (a) Wireless Network Coding (b) Reverse carpooling.

the reverse carpooling schemes. In particular, to cater to delay-sensitive applications, the network must be aware that savings achieved by coding may be offset by delays incurred in waiting for such opportunities. Accordingly, we design delay-aware controllers that use local information to decide whether or not to wait for a coding opportunity, or to go ahead with an uncoded transmission. By sending uncoded packets we do not take advantage of network coding, hence are not energy-efficient. However, by waiting for packets to code, we might be able to achieve energy efficiency at the cost of packets getting delayed further.

Consider a relay node that transmits packets between two of its adjacent nodes that have flows in opposite directions, as depicted in Fig. 6. The relay maintains two queues q_1 and q_2, such that q_1 and q_2 store packets that need to be delivered to node 2 and 1, respectively. If both queues are not empty, then it can relay two packets from these queues by performing an XOR operation. However, what should the relay do if one of the queues has packets to transmit, while the second queue is empty? Should the relay wait for a coding opportunity or just transmit a packet from a non-empty queue without coding? This is the fundamental question we seek to answer. In essence we would like to trade off efficiently transmitting the packets against high quality of service (i.e. low delays).

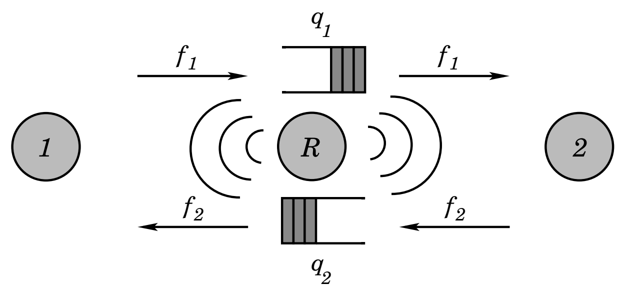


Fig. 6. 3-Node Relay Network.

Network coding research was initiated by the seminal work [15] and since then attracted major interest from the research community. Network coding technique for wireless networks has been considered in [16]. They propose an architecture called COPE, which contains a special network coding layer between the IP and MAC layers. In [17] an opportunistic routing protocol is proposed, referred to as MORE, that randomly mixes packets that belong to the same flow before forwarding them to the next hop. Moreover, several works, e.g. [18–23], investigate the scheduling or/and routing problems in the network coding enabled networks. [18] focuses on the network coding in the tandem networks and formulates a related cross-layer optimization problem, while [19] devises a joint coding-scheduling-rate controller when the pairwise intersession network coding is allowed. In an earlier work [20], it is shown how to design coding-aware routing controllers that would maximize coding opportunities in multihop networks. [21] and [22] attempt to schedule the network coding between multiple-session flows. A distributed algorithm is proposed in [23] for the minimum cost transmission of the multicast session.

However, in contrast to all the above literature, we consider the stochastic arrival process and see if the packet desires to wait for a coding opportunity. Our objective is therefore to study the delicate trade-off between the energy exhaustion and the queueing delay when network coding is an option. For that we formulate and solve a stochastic dynamic program, in particular a Markov decision process, to determine the optimal control actions in various states. The literature is extremely rich for finite state-space problems. However, although there have been several excellent books (e.g. [24–27]) on MDPs, there are relatively few articles that provide a methodology to find optimal policies for problems that are infinite horizon, average cost optimization with a countably infinite state space. In fact, [27] specifically says that such problems are difficult to analyze and obtain optimal policies.

A. Main Results

We first consider the case of a relay node that can serve at most 1 packet for each time, and assume that the arrivals into both queues are independent and identically distributed. We can think of the system state as the queue lengths and explore the effect of including the vector of waiting times associated with each of the packets, to design a cost-minimizing controller. We first claim that although waiting time information might be available, we do not actually need to use it. Since our system is Markovian, the problem is essentially that of a Markov Decision Process. In general, we could have a controller that belongs to one of the following sets [24]: Π^{HR} (randomized history dependent policies), Π^{MR} (randomized Markov policies), Π^{SR} (randomized stationary policies), and Π^{SD} (deterministic stationary policies). The complexity of the algorithms exactly decreases from left to right above. In what regime does the solution to our problem lie? We find that the optimal policy is a simple queue-length threshold policy with one threshold for each queue at the relay, and whose action is simply: if a coding opportunity exists, code and transmit; else transmit a packet if the threshold for that queue is reached. We then show how to find the optimal thresholds.

Two general models are examined afterwards. In the first model, the service capacity of the relay is not restricted to 1 packet per time unit. Then if the relay can serve a batch of packets, the optimal controller is found to be of the threshold type for one queue when the length of the other queue is fixed. Moreover, the arrivals with memory is also studied. The optimal policy for Markov-modulated arrival process is discovered to have multiple thresholds. We then numerically study a number of policies, based on waiting time and queue length, waiting time only, randomized thresholds, and the optimal deterministic queue-length threshold policy to indicate the potential of our approach. Finally, the performance of the deterministic queue length based policy in the line network topology is illustrated via simulations.

B. System Overview

Consider a multi-hop wireless network operating a time-division multiplexing scheme to store and forward packets from various sources to destinations. Time is slotted into small intervals and in each interval relay node gets a chance to transmit a number of packets of a flow. These packets are transmitted during a "mini-slot" that the node has been assigned. We assume that this mini-slot is instantaneous for all practical purposes. Also, in this work we will not consider any scheduling issues and assume that we have scheduled mini-slots assigned to each node for each flow where nodes have opportunities to transmit if they choose to. With that said, we will now describe the scenario from the perspective of a single node, especially a relay that has the potential for network coding packets from flows in opposing directions.

1. Scenario from a Relay's Perspective

Consider Fig. 6. We call two of the adjacent nodes to the relay R as nodes 1 and 2. Say there is a flow f_1 that goes from node 1 to 2 and another flow f_2 from node 2 to 1, both of which are through the relay under consideration. The packets (type 1 and type 2) from both flows respectively go through separate queues, q_1 and q_2, at node R. With respect to the relay we now define a *slot* as the time between successive opportunities for the relay to transmit. The arrival processes to both flows are assumed to be independent of each other and also independently and identically distributed ($i.i.d$) over time, with the random variables $\mathcal{A}_i$ for $i \in \{1, 2\}$ respectively. In each slot, n packets arrive at q_i with the probability $\mathbb{P}(\mathcal{A}_i = n) = p_n^{(i)}$ for $n \in \mathbb{N} \cup \{0\}$. At the end of a slot, the relay gets an opportunity to transmit. First we consider that at the end of a slot, the relay can transmit a maximum of one packet. When both queues are non-empty, one packet from q_1 and one from q_2 can be transmitted together as a single packet using XOR coding. This scenario, in which transmitting a combination of packets results in decreasing the required number of transmissions, is referred as a *coding opportunity.* Whenever such a coding opportunity exists between the packets of two flows, the relay encodes the packets and transmits the coded packet back to the adjacent nodes. However, if there is only one type of packet at the end of a slot, there are two options: (i) one of those packets gets transmitted without coding or (ii) we wait for a future slot to receive a packet in the other queue to utilize the coding opportunity. We assume that transmissions within a type are according to a first-in-first-out service discipline. It is unclear as to what is the best course of action, do nothing (thus worsening delay) or transmit without coding (thus being inefficient). In other words, to wait or not to wait, that is the question.

2. Markov Decision Process Model

To develop a strategy for the relay to decide at every transmission opportunity, its best course of action, we use a Markov decision process (MDP) model. For $i = 1, 2$ and $t = 0, 1, 2, \cdots$, let $Q_t^{(i)}$ be the number of packets of type i in the relay at the end of time slot t just before an opportunity to transmit. Let A_t be the action chosen at the end of the t^{th} time slot with $A_t = 0$ implying the action is to do nothing and $A_t = 1$ implying the action is to transmit. As we described before, if $Q_t^{(1)} + Q_t^{(2)} = 0$, then $A_t = 0$ because that is the only feasible action. Also, if $Q_t^{(1)} Q_t^{(2)} > 0$, then $A_t = 1$ because the best option is to transmit a coded XOR packet as it reduces both the number of transmissions as well as latency. However, when exactly one of $Q_t^{(1)}$ and $Q_t^{(2)}$ is non-zero, it is not immediately clear what action we should choose.

We first define costs for latency and transmission. Let C_T be the cost for transmitting a packet and C_H be the cost for holding a packet for a length of time equal to one slot. Without loss of generality, we assume that if a packet is transmitted in the same slot it arrived, its latency is zero. Also, the cost of transmitting a coded packet is the same as that of a non-coded packet. That said, our objective is to derive an optimal policy that minimizes the long-run average cost per slot. For that we define the MDP$\{(Q_t, A_t), t \geq 0\}$ where $Q_t = (Q_t^{(1)}, Q_t^{(2)})$ is the state of the system and A_t the control action chosen at time t. The state space (*i.e.*, all possible values of Q_t) is the set $\{(i, j) : i \geq 0, j \geq 0\}$.

Let $C(Q_t, A_t)$ be the *immediate cost* incurred at time t if action A_t is taken, when the system is in state Q_t. Therefore,

$$C(Q_t, A_t) = C_H([Q_t^{(1)} - A_t]^+ + [Q_t^{(2)} - A_t]^+) + C_T A_t, \tag{3.1}$$

where $[x]^+ = \max(x, 0)$. The long-run average cost for some policy θ is given by

$$V(\theta) = \lim_{K\to\infty} \frac{1}{K+1} \mathbb{E}_\theta \left[\sum_{t=0}^{K} C(Q_t, A_t) | Q_0 = (0,0) \right], \tag{3.2}$$

where $\mathbb{E}_\theta$ is the expectation operator taken for the system under policy θ. Notice that our initial state is an empty system, although the average cost would not depend on it. Define *average-optimal policy* as the policy that minimizes $V(\theta)$. Our goal is to characterize and obtain the average-optimal policy. For that we first describe the probability law for our MDP and then in subsequent section develop a methodology to obtain the average-optimal policy.

For the MDP$\{(Q_t, A_t), t \geq 0\}$, let $P_a(Q_t, Q_{t+1})$ be the transition probability from state Q_t to Q_{t+1} associated with action $a \in \{0, 1\}$. Then the probability law can be derived as $P_0\left((i,j),(k,l)\right) = p^{(1)}_{k-i} p^{(2)}_{l-j}$ for all $k \geq i$ and $l \geq j$. Also, $P_1\left((i,j),(k,l)\right) = p^{(1)}_{k-[i-1]^+} p^{(2)}_{l-[j-1]^+}$ for all $k \geq [i-1]^+$ and $l \geq [j-1]^+$.

C. Analysis

As we described in the previous section, our goal is to obtain the average-optimal policy. To that end, we first find the space of possible policies and then identify the average-optimal policy within this space. Our first question is: what is the appropriate state space: is it just queue length, or should we also consider waiting time?

1. Should We Maintain Waiting Time Information?

Intuition tells us that if a packet has not been waiting long enough then perhaps it could afford waiting a little more but if a packet has waited too long, it may be better to just transmit it. That seems logical considering that we try our best to code but we cannot wait too long because it hurts in terms of holding costs. Also, one could get waiting time information from time-stamps on packets. Let $T^{(i)}$ be the arrival time of i^{th} packet and $\mathcal{D}_\theta^{(i)}$ be its delay (i.e. the waiting time before it is transmitted) while policy θ is applied. $\mathcal{T}_{t,\theta}$ is the number of transmission by time t under policy θ. Then (3.2) can be written as

$$V(\theta) = \lim_{K\to\infty} \frac{1}{K+1} \mathbb{E}_\theta \left[\sum_{i:T^{(i)}\leq K} C_H \mathcal{D}_\theta^{(i)} + C_T \mathcal{T}_{K,\theta} \right]. \tag{3.3}$$

Given that, would we be making better decisions by also keeping track of waiting times of each packet? We answer that question by means of a theorem which requires the following lemma for a generic MDP$\{(X_t, D_t), t \geq 0\}$, where X_t is the state of the MDP and D_t is the action at time t.

Lemma 4 (Theorem 5.5.3 [24])**.** *For an MDP $\{(X_n, D_n), n \geq 0\}$, given any history dependent policy and starting state, there exists a randomized Markov policy with the same long-run average cost.*

Using the above lemma we show next that it is not necessary to maintain waiting time information.

Theorem 4. *For the MDP$\{(Q_t, A_t), t \geq 0\}$, if there exists a randomized history dependent policy that is average-optimal, then there exists a randomized Markov policy θ^* that minimizes $V(\theta)$ defined in (3.2). Further, one cannot find a policy which also uses waiting time information that would yield a better solution than $V(\theta^*)$.*

Proof. From Lemma 4, if the MDP$\{(Q_t, A_t), t \geq 0\}$ has a history dependent policy that is average-optimal, then we can construct a randomized Markov policy that yields the same long-run average cost given $Q_0 = (0, 0)$. Therefore, if there exists a randomized history dependent policy that is average-optimal, then there exists a randomized Markov policy θ^* that minimizes $V(\theta)$ defined in (3.2).

Knowing the entire history of states and actions one can always determine the history of waiting times as well as the current waiting times of all packets. Therefore the average-optimal policy θ' that uses waiting time information is equivalent to a history dependent policy. From Lemma 4, we can always find a randomized Markov policy that yields the same average-optimal solution as $V(\theta')$. □

2. Average-Optimal Policy: Stationary and Deterministic

In the previous section, we showed that there exists an average-optimal policy that does not include the waiting time in the state of the system. Afterward, we focus on queue length-based policies and determine the structure of the average-optimal policy. In the MDP literatures (e.g. [28]), the conditions for the structure and location of average-optimal policy usually rely on the results of the infinite horizon α-discounted cost case, where $0 < \alpha < 1$. Accordingly, for our MDP$\{(Q_t, A_t), t \geq 0\}$, the total expected discounted cost incurred by a policy θ is

$$V_{\theta,\alpha}(i,j) = \mathbb{E}_\theta \left[\sum_{t=0}^{\infty} \alpha^t C(Q_t, A_t) | Q_0 = (i,j) \right]. \tag{3.4}$$

In addition, we define $V_\alpha(i,j) = \min_\theta V_{\theta,\alpha}(i,j)$ as well as $v_\alpha(i,j) = V_\alpha(i,j) - V_\alpha(0,0)$. Define the α-*optimal policy* as the policy that minimizes (3.4).

Proposition 5. *If* $\mathbb{E}[\mathcal{A}_i] < \infty$ *for* $i \in \{1,2\}$, *then* $V_\alpha(i,j) < \infty$ *for every state* (i,j) *and discounted factor* $0 < \alpha < 1$.

Proof. By $\tilde{\theta}$, we denote the stationary policy of always doing nothing for each time slot. Hence, $V_\alpha(i,j) \le V_{\tilde{\theta},\alpha}(i,j)$. Furthermore, we note that

$$\begin{aligned} V_{\tilde{\theta},\alpha}(i,j) &= \mathbb{E}_{\tilde{\theta}}\Big[\sum_{t=0}^{\infty} \alpha^t C(Q_t, A_t) | Q_0 = (i,j)\Big] \\ &= \sum_{t=0}^{\infty} \alpha^t C_H \left(i + j + t\mathbb{E}[\mathcal{A}_1 + \mathcal{A}_2]\right) \\ &= \frac{C_H(i+j)}{1-\alpha} + \frac{\alpha C_H}{(1-\alpha)^2}\mathbb{E}[\mathcal{A}_1 + \mathcal{A}_2] < \infty. \end{aligned}$$

□

Lemma 5 (Proposition 1 and 3 [28])**.** *Proposition 5 (i.e. $V_\alpha(i,j)$ is finite) implies that $V_\alpha(i,j)$ satisfies the optimality equation as following,*

$$V_\alpha(i,j) = \min_{a\in\{0,1\}}[C_H([i-a]^+ + [j-a]^+) + C_T a + \alpha \sum_{k,l} P_a\big((i,j),(k,l)\big)V_\alpha(k,l)] \quad (3.5)$$

The stationary policy that realizes the minimum of right hand side of (3.5) is α-optimal policy. Moreover, let $V_{\alpha,0}(i,j) = 0$ and for $n \ge 1$,

$$V_{\alpha,n}(i,j) = \min_{a\in\{0,1\}}[C_H([i-a]^+ + [j-a]^+) + C_T a + \alpha \textstyle\sum_{k,l} P_a\big((i,j),(k,l)\big)V_{\alpha,n-1}(k,l)]. \quad (3.6)$$

Then, $V_{\alpha,n}(i,j) \to V_\alpha(i,j)$ as $n \to \infty$ for every i, j, and α.

We use the second part of Lemma 5 to prove that $V_\alpha(i,j)$ is a non-decreasing function.

Lemma 6. *$V_\alpha(i,j)$ is non-decreasing with respect to i and j respectively.*

Proof. Prove by induction on n in (3.6). It is obvious for $V_{\alpha,0}(i,j)$. Assume that $V_{\alpha,n}(i,j)$ is non-decreasing. Note that

$$\begin{aligned} &C_H([i-a]^+ + [j-a]^+) + C_T a + \alpha \textstyle\sum_{k,l} P_a\big((i,j),(k,l)\big)V_{\alpha,n}(k,l) \\ &= C_H([i-a]^+ + [j-a]^+) + C_T a + \alpha \textstyle\sum_{r,s} p_r^{(1)} p_s^{(2)} V_{\alpha,n}([i-a]^+ + r, [j-a]^+ + s) \end{aligned}$$

is non-decreasing for all $a \in \{0, 1\}$ since C_H is non-negative. Therefore, $V_{\alpha,n+1}(i, j)$ is non-decreasing because the minimum operator in (3.6) keeps the non-decreasing property. □

The next lemma relates the α-discounted cost and the long-run average cost cases.

Lemma 7 (Theorem (i) [28]). *There exists a stationary and deterministic policy that is average-optimal for the MDP$\{(Q_t, A_t), t \geq 0\}$ if the following conditions are satisfied: (i) $V_\alpha(i, j)$ is finite for all i, j, and discount factor α; (ii) there exists a nonnegative N such that $v_\alpha(i, j) \geq -N$ for all i, j, and α; and (iii) there exists a nonnegative $M_{i,j}$ such that $v_\alpha(i, j) \leq M_{i,j}$ for every i, j, and α. Moreover, for each state (i, j) there is an action $a(i, j)$ such that $\sum_{k,l} P_{a(i,j)}\big((i, j), (k, l)\big) M_{k,l} < \infty$.*

Using this lemma, we will show in Theorem 6 that the our MDP has an average-optimal policy that is stationary and deterministic. Please refer to Appendix A for the proof.

Theorem 6. *For the MDP$\{(Q_t, A_t), t \geq 0\}$, there exists a stationary and deterministic policy θ^* that minimizes $V(\theta)$ if $\mathbb{E}[\mathcal{A}_i^2] < \infty$ and $\mathbb{E}[\mathcal{A}_i] < 1$ for $i \in \{1, 2\}$.*

Now that we know the average-optimal policy is stationary and deterministic, the question is how we find it. In the next section, we investigate this problem and will show that the average-optimal policy is a threshold-based policy.

3. Structure of Optimal Policy: Threshold Based

The standard methodology to obtain stationary policy for infinite-horizon average cost minimization problem is to use a linear program (LP) [25]. However, the method assumes the finite states, so we cannot directly apply to our MDP$\{(Q_t, A_t), t \geq 0\}$, as our MDP has infinite states and the Markov chain under some policy is not irreducible (for example if we always transmit, it is not possible to reach some of the states). To circumvent that, we might construct a finite size LP with N states and force it to be irreducible by creating dummy transitions with probability $\epsilon > 0$ between some states. Let us call this LP(N, ϵ). By letting $N \to \infty$ and $\epsilon \to 0$ we argue that our MDP would have an average-optimal deterministic policy.

LP(N, ϵ) provides a method to obtain the stationary and deterministic suboptimal policy for our MDP$\{(Q_t, A_t), t \geq 0\}$. We are still interested in how to find the average-optimal policy. If we know that the average-optimal policy satisfies the structural properties then it is possible to search through the space of stationary deterministic policies and obtain the optimal one. We will study the α-optimal policy first and then discuss how to extend to the average-optimal policy.

a. Structure of α-Optimal Policy

In this subsection, we will investigate the properties of an α-optimal policy. These results will be useful in the subsequent section in order to find the average-optimal scheme. Before examining the general *i.i.d.* arrival model, first we consider a special case of Bernoulli arrival processes.

***i.i.d.* Bernoulli Arrival Processes:**

The following lemma summarizes our result for this special case, where the arrivals to the queues are independently distributed as Bernoulli random variables.

Lemma 8. *Consider the i.i.d. Bernoulli arrival process. For the MDP$\{(Q_t, A_t), t \geq 0\}$, the α-optimal policy is of threshold type. There exist the optimal thresholds $L^*_{\alpha,1}$ and $L^*_{\alpha,2}$ so that the optimal deterministic action in state $(i, 0)$ is to wait if $i < L^*_{\alpha,1}$, and to transmit without coding if $i \geq L^*_{\alpha,1}$; while in state $(0, j)$ is to wait if $j < L^*_{\alpha,2}$,* *and to transmit without coding if* $j \geq L^*_{\alpha,2}$.

Proof. Referring to (3.5), let

$$\mathcal{V}_\alpha(i,0,a) = C_H([i-a]^+) + C_T a + \alpha \sum_{k,l} P_a\big((i,0),(k,l)\big) V_\alpha(k,l)$$

and define $V_\alpha(i,0) = \min_{a\in\{0,1\}} \mathcal{V}_\alpha(i,0,a)$. Let $L^*_{\alpha,1} = \min\{i \in \mathbb{N}\cup\{0\} : \mathcal{V}_\alpha(i,0,1) \leq \mathcal{V}_\alpha(i,0,0)\}$. Then the optimal stationary and deterministic action (for the total expected α-discounted cost) is $A_n = 0$ for the states $(i,0)$ with $i < L^*_{\alpha,1}$, and $A_n = 1$ for the state $(L^*_{\alpha,1}, 0)$. Note that we do not care about the states $(i,0)$ with $i > L^*_{\alpha,1}$ since they are not accessible as $(L^*_{\alpha,1},0)$ only transits to $(L^*_{\alpha,1}-1,0)$, $(L^*_{\alpha,1},0)$, $(L^*_{\alpha,1}-1,1)$, and $(L^*_{\alpha,1},1)$. The similar argument is applicable for the states $(0,j)$. Consequently, there exists a policy of threshold type that is α-optimal. □

For the *i.i.d.* Bernoulli arrival process, we have shown that the the α-optimal policy can be threshold based. The next question is the possibility to extend to any *i.i.d.* arrival process.

General *i.i.d.* Arrival Processes:

We shall see that a threshold based α-optimal policy exists for the case of general *i.i.d.* arrivals as well. Define

$$\mathcal{V}_\alpha(i,j,a) = C_H\left([i-a]^+ + [j-a]^+\right) + C_T a + \alpha\mathbb{E}\Big[V_\alpha\left([i-a]^+ + \mathcal{A}_1, [j-a]^+ + \mathcal{A}_2\right)\Big],$$

and

$$\mathcal{V}_{\alpha,n}(i,j,a) = C_H\left([i-a]^+ + [j-a]^+\right) + C_T a + \alpha\mathbb{E}\Big[V_{\alpha,n}\left([i-a]^+ + \mathcal{A}_1, [j-a]^+ + \mathcal{A}_2\right)\Big].$$

Then, (3.5) can be expressed as $V_\alpha(i,j) = \min_{a\in\{0,1\}} \mathcal{V}_\alpha(i,j,a)$, while (3.6) is written as $V_{\alpha,n}(i,j) = \min_{a\in\{0,1\}} \mathcal{V}_{\alpha,n-1}(i,j,a)$. For every discounted factor α, we want to show that there exists an α-optimal policy that is of threshold type. To be precise, let the α-optimal policy for the first dimension $a^*_{\alpha,i} = \arg\min_{a\in\{0,1\}} \mathcal{V}_\alpha(i,0,a)$, we will show that $a^*_{\alpha,i}$ is non-decreasing as i increases, and so is the case for the second dimension.

In our analysis, we investigate some important properties of function $\mathcal{V}_{\alpha,n}(i,j,a)$ like *subadditivity*, *submodularity* and *subconvexity*. These properties are properly defined as follows.

Definition 3 (Subadditivity [24])**.** *A function $f : (\mathbb{N}\cup\{0\})^2 \to \mathbb{R}$ is subadditive if for all $i^- \le i^+$ and $a^- \le a^+$,*

$$f(i^+,a^+) - f(i^+,a^-) \le f(i^-,a^+) - f(i^-,a^-). \tag{3.7}$$

Definition 4 (Submodularity [29])**.** *A function $f : (\mathbb{N}\cup\{0\})^2 \to \mathbb{R}$ is submodular if for all $i,j \in \mathbb{N}\cup\{0\}$*

$$f(i,j) + f(i+1,j+1) \le f(i+1,j) + f(i,j+1). \tag{3.8}$$

Definition 5 ($\mathcal{K}$-Convexity)**.** *A function $f : (\mathbb{N}\cup\{0\})^2 \to \mathbb{R}$ is $\mathcal{K}$-convex (where $\mathcal{K}\in\mathbb{N}$) if for every $i,j \in \mathbb{N}\cup\{0\}$*

$$f(i+\mathcal{K},j) - f(i,j) \le f(i+\mathcal{K}+1,j) - f(i+1,j) \tag{3.9}$$

$$f(i,j+\mathcal{K}) - f(i,j) \le f(i,j+\mathcal{K}+1) - f(i,j+1). \tag{3.10}$$

Definition 6 ($\mathcal{K}$-Subconvexity)**.** *A function $f : (\mathbb{N}\cup\{0\})^2 \to \mathbb{R}$ is $\mathcal{K}$-subconvex (where $\mathcal{K}\in\mathbb{N}$) if for all $i,j \in \mathbb{N}\cup\{0\}$*

$$f(i+\mathcal{K},j+\mathcal{K}) - f(i,j) \le f(i+\mathcal{K}+1,j+\mathcal{K}) - f(i+1,j) \tag{3.11}$$

$$f(i+\mathcal{K},j+\mathcal{K}) - f(i,j) \le f(i+\mathcal{K},j+\mathcal{K}+1) - f(i,j+1). \tag{3.12}$$

Remark 7. *If a function $f : (\mathbb{N}\cup\{0\})^2 \to \mathbb{R}$ is submodular and $\mathcal{K}$-subconvex, then* *it is $\mathcal{K}$-convex, and for every $r \in \mathbb{N}$ with $1 \le r < \mathcal{K}$,*

$$f(i+\mathcal{K}, j+r) - f(i,j) \le f(i+\mathcal{K}+1, j+r) - f(i+1,j) \tag{3.13}$$

$$f(i+r, j+\mathcal{K}) - f(i,j) \le f(i+r, j+\mathcal{K}+1) - f(i,j+1). \tag{3.14}$$

For simplicity, we will ignore $\mathcal{K}$ in Definitions 5 and 6 when $\mathcal{K} = 1$. Suppose it is true that

$$\mathcal{V}_\alpha(i+1,0,1) - \mathcal{V}_\alpha(i+1,0,0) \le \mathcal{V}_\alpha(i,0,1) - \mathcal{V}_\alpha(i,0,0). \tag{3.15}$$

Now if the α-optimal polity for state $(i,0)$ is $a^*_{\alpha,i} = 1$, *i.e.*, $\mathcal{V}_\alpha(i,0,1) \le \mathcal{V}_\alpha(i,0,0)$, then the α-optimal policy for state $(i+1,0)$ is also $a^*_{\alpha,i+1} = 1$. On the other hand, the α-optimal policy for state $(i,0)$ is $a^*_{\alpha,i} = 0$ once the α-optimal polity for state $(i+1,0)$ is $a^*_{\alpha,i+1} = 0$. Therefore in order to show that α-optimal policy is monotonic, it suffices to show that (3.15) holds. Equivalently, it is enough to show that $\mathcal{V}_\alpha(i,0,a)$ and $\mathcal{V}_\alpha(0,j,a)$ are subadditive for $a \in \{0,1\}$.

Our approach to show the subadditivity property of $\mathcal{V}_\alpha(i,0,a)$ (and $\mathcal{V}_\alpha(0,j,a)$) is by inductively verifying this property for $\mathcal{V}_{\alpha,n}(i,j,a)$. The following useful lemma provides a set of sufficient conditions for $\mathcal{V}_{\alpha,n}(i,j,a)$ to be subadditive.

Lemma 9. *Given $0 < \alpha < 1$ and $n \in \mathbb{N} \cup \{0\}$. If $V_{\alpha,n}(i,j) = \min_{a\in\{0,1\}} \mathcal{V}_{\alpha,n-1}(i,j,a)$ is non-decreasing, submodular, and subconvex, then $\mathcal{V}_{\alpha,n}(i,j,a)$ is subadditive for i and a when j is fixed, and vice versa. Consequently, $\min\{a : \arg\min_{a\in\{0,1\}} \mathcal{V}_{\alpha,n}(i,j,a)\}$ is non-decreasing w.r.t. i for fixed j, and vice versa.*

Proof. Define $\Delta\mathcal{V}_{\alpha,n}(i,j) = \mathcal{V}_{\alpha,n}(i,j,1) - \mathcal{V}_{\alpha,n}(i,j,0)$. We claim $\Delta\mathcal{V}_{\alpha,n}(i,j)$ is non-increasing, more precisely $\Delta\mathcal{V}_{\alpha,n}(i,j)$ is a non-increasing function *w.r.t.* i while j is fixed, and vice versa. In what follows, we will prove this claim by focusing on the prior part. Notice that $\Delta\mathcal{V}_{\alpha,n}(i,j) =$

$$C_H([i-1]^+ + (j-1)^+) + C_T + \alpha\mathbb{E}[V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)]$$
$$-C_H(i+j) - \alpha\mathbb{E}[V_{\alpha,n}(i+A_1, j+A_2)]$$

To be precise for $i \geq 1$, if $j = 0$ we have

$$\Delta\mathcal{V}_{\alpha,n}(i,0) = C_T - C_H + \alpha\mathbb{E}[V_{\alpha,n}(i-1+\mathcal{A}_1, \mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, \mathcal{A}_2)], \tag{3.16}$$

otherwise, for $j \geq 1$, we get

$$\begin{aligned} &\Delta\mathcal{V}_{\alpha,n}(i,j) = \\ &\quad C_T - 2C_H + \alpha\mathbb{E}[V_{\alpha,n}(i-1+\mathcal{A}_1, j-1+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)]. \end{aligned} \tag{3.17}$$

Fix $j \geq 1$ in (3.17), according to the subconvexity of $V_{\alpha,n}(i,j)$, $\Delta\mathcal{V}_{\alpha,n}(i,j)$ does not increase as i increases for $i \geq 1$. While $j = 0$ is given, that is also true due to the convexity of $V_{\alpha,n}(i,j)$. Furthermore, the boundary conditions are considered in the following. Let $j \geq 1$ fixed, then

$$\begin{aligned} \Delta\mathcal{V}_{\alpha,n}(1,j) &= C_T - 2C_H + \alpha\mathbb{E}[V_{\alpha,n}(\mathcal{A}_1, j-1+\mathcal{A}_2) - V_{\alpha,n}(1+\mathcal{A}_1, j+\mathcal{A}_2)] \\ \Delta\mathcal{V}_{\alpha,n}(0,j) &= C_T - C_H + \alpha\mathbb{E}[V_{\alpha,n}(\mathcal{A}_1, j-1+\mathcal{A}_2) - V_{\alpha,n}(\mathcal{A}_1, j+\mathcal{A}_2)]. \end{aligned} \tag{3.18}$$

Note that $\mathbb{E}[V_{\alpha,n}(1+\mathcal{A}_1, j+\mathcal{A}_2)] \geq \mathbb{E}[V_{\alpha,n}(\mathcal{A}_1, j+\mathcal{A}_2)]$ according to non-decreasing property of $V_{\alpha,n}(i,j)$ and then $\Delta\mathcal{V}_{\alpha,n}(1,j) \leq \Delta\mathcal{V}_{\alpha,n}(0,j)$. Finally, for $j = 0$ we have

$$\begin{aligned} \Delta\mathcal{V}_{\alpha,n}(1,0) &= C_T - C_H + \alpha\mathbb{E}[V_{\alpha,n}(\mathcal{A}_1, \mathcal{A}_2) - V_{\alpha,n}(1+\mathcal{A}_1, \mathcal{A}_2)] \\ \Delta\mathcal{V}_{\alpha,n}(0,0) &= C_T. \end{aligned} \tag{3.19}$$

$\Delta\mathcal{V}_{\alpha,n}(1,0) \leq \Delta\mathcal{V}_{\alpha,n}(0,0)$ results from $\mathbb{E}[V_{\alpha,n}(\mathcal{A}_1, \mathcal{A}_2) - V_{\alpha,n}(1+\mathcal{A}_1, \mathcal{A}_2)] \leq 0$ as $V_{\alpha,n}(i,j)$ is non-decreasing. Consequently, $\Delta\mathcal{V}_{\alpha,n}(i,j)$ is a non-increasing function *w.r.t.* i while j is fixed. □

In the previous lemma, we found some conditions on $V_{\alpha,n}(i,j)$ which imply the subadditivity of $\mathcal{V}_{\alpha,n}(i,j,a)$. If we show that these conditions hold for all values of n, then one can easily conclude that $\mathcal{V}_{\alpha}(i,0,a)$ (similarly $\mathcal{V}_{\alpha}(0,j,a)$) is subadditive and as a result the α-optimal policy is monotonic.

The following lemma declares that if the mentioned conditions hold for $V_{\alpha,n}(i,j)$, then they will also hold for $V_{\alpha,n+1}(i,j)$. Please refer to Appendix A for the proof.

Lemma 10. *Given $0 < \alpha < 1$ and $n \in \mathbb{N} \cup \{0\}$. If $V_{\alpha,n}(i,j)$ is non-decreasing, submodular, and subconvex, then $V_{\alpha,n+1}(i,j)$ is non-decreasing, submodular, and subconvex.*

Using all the previous preliminary lemmas, we can now present our main result for the structure of α-optimal policies in the following lemma.

Theorem 8. *For the MDP$\{(Q_t, A_t), t \geq 0\}$ with any i.i.d. arrival processes to both queues, there exists an α-optimal policy that is of threshold type. Given Q_2, the α-optimal policy is monotone w.r.t. Q_1, and vice versa.*

Proof. Prove by induction. Note that $V_{\alpha,0}(i,j) = 0$ is non-decreasing, submodular, and subconvex, that leads to $\min\{a : \arg\min_{a\in\{0,1\}} \mathcal{V}_{\alpha,0}(i,j,a)\}$ being non-decreasing based on Lemma 9. These properties propagate as n goes to infinity according to Lemma 10. □

In the next subsection, we extend these results to average-optimal policies.

b. Structure of Average-Optimal Policy

Thus far, we have characterized the α-optimal policy. The following lemma describes a useful relation between the average-optimal policy and the α-optimal policy.

Lemma 11 (Lemma and Theorem (i) [28])**.** *Consider the MDP$\{(Q_t, A_t), t \geq 0\}$. Let $\{\alpha_n\}$, converging to 1, be any sequence of discount factors associated with the*

α-optimal policy $\{\theta_{\alpha_n}(i,j)\}$. There exists a subsequence $\{\beta_n\}$ and a stationary policy $\theta^*(i,j)$ that is the limit point of $\{\theta_{\beta_n}(i,j)\}$. If the three conditions in Lemma 7 are satisfied, $\theta^*(i,j)$ is the average-optimal policy for (3.2).

Using this lemma, it is straightforward to conclude that a threshold based average-optimal policy exists. The following theorem formally expresses this result.

Theorem 9. *Consider any i.i.d. arrival processes to both queues. The average-optimal policy for the MDP$\{(Q_t, A_t), t \geq 0\}$ is of threshold type. There exist the optimal thresholds L_1^* and L_2^* so that the optimal deterministic action in states $(i,0)$ is to wait if $i \leq L_1^*$, and to transmit without coding if $i > L_1^*$; while in state $(0,j)$ is to wait if $j \leq L_2^*$, and to transmit without coding if $j > L_2^*$.*

Proof. Let $(\tilde{i},0)$ be any state, at which the average-optimal policy is to transmit, *i.e.*, $\theta^*(\tilde{i},0) = 1$ in Lemma 11. Since there is a sequence of discounted factors $\{\beta_n\}$ such that $\theta_{\beta_n}(i,j) \to \theta^*(i,j)$, then there exists $N > 0$ so that $\theta_{\beta_n}(\tilde{i},0) = 1$ for all $n \geq N$. Due to the monotone of α-optimal policy in Theorem 8, $\theta_{\beta_n}(i,0) = 1$ for all $i \geq \tilde{i}$ and $n \geq N$. Therefore, $\theta^*(i,0) = 1$ for all $i \geq \tilde{i}$. To conclude, the average-optimal policy is of threshold type. □

In the next subsection, we briefly comment about how to find the best thresholds for an average-optimal policy.

4. Obtaining the Optimal Deterministic Stationary Policy

We have shown in the previous subsection that the average-optimal policy is stationary, deterministic and threshold type, so we only need to consider the subset of deterministic stationary policies. Given the thresholds of both queues, the MDP is reduced to a Markov chain. The next step is to find the optimal threshold.

First note that the condition $\mathbb{E}[\mathcal{A}_i] < 1$ might not be sufficient for the stability of the queues since the threshold based policy leads to a lower average service rate. In the following theorem, we claim that the conditions $\mathbb{E}[\mathcal{A}_i^2] < \infty$ and $\mathbb{E}[\mathcal{A}_i] < 1$ for $i \in \{1, 2\}$ are enough for the stability of the queues. For the proof, please refer to Appendix A.

Theorem 10. *For the MDP$\{(Q_t, A_t), t \geq 0\}$ with $\mathbb{E}[\mathcal{A}_i^2] < \infty$ and $\mathbb{E}[\mathcal{A}_i] < 1$ for $i \in \{1, 2\}$. The reduced Markov chain from applying the stationary and deterministic threshold based policy to MDP is positive recurrent, i.e. the stationary distribution exists.*

We realize that if $\mathbb{E}[\mathcal{A}_i^2] < \infty$ and $\mathbb{E}[\mathcal{A}_i] < 1$ for $i \in \{1, 2\}$, then there exists a stationary threshold type policy that is average-optimal and can be obtained from the reduced Markov chain. The following theorem gives an example of how to compute the optimal thresholds for the case of Bernoulli arrivals.

Theorem 11. *Consider the Bernoulli arrival process. The optimal thresholds L_1^* and L_2^* are*

$$(L_1^*, L_2^*) = \arg\min_{L_1, L_2} C_T \mathcal{T}(L_1, L_2) + C_H \mathcal{H}(L_1, L_2), \tag{3.20}$$

where,

$$\mathcal{T}(L_1, L_2) = p_1^{(1)} p_1^{(2)} \pi_{0,0} + p_1^{(2)} \textstyle\sum_{i=1}^{L_1} \pi_{i,0} + p_1^{(1)} \sum_{j=1}^{L_2} \pi_{0,j} + p_1^{(1)} p_0^{(2)} \pi_{L_1,0} + p_0^{(1)} p_1^{(2)} \pi_{0,L_2}$$
$$\mathcal{H}(L_1, L_2) = \textstyle\sum_{i=1}^{L_1} i\pi_{i,0} + \sum_{j=1}^{L_2} j\pi_{0,j},$$

in which,

$$\begin{aligned}
\pi_{0,0} &= \frac{1}{\left(\frac{1-\alpha^{L_1+1}}{1-\alpha}\right) + \left(\frac{1-1/\alpha^{L_2+1}}{1-1/\alpha}\right) - 1} \\
\pi_{i,0} &= \alpha^i \pi_{0,0} \\
\pi_{0,j} &= \pi_{0,0}/\alpha^j \\
\alpha &= \frac{p_1^{(1)} p_0^{(2)}}{p_0^{(1)} p_1^{(2)}}.
\end{aligned}$$

Proof. Let $Y_t^{(i)}$ be the number of type i packets at the t^{th} slot after transmission. It is crucial to note that this observation time is different from when the MDP is observed. Then the bivariate stochastic process $\{(Y_t^{(1)}, Y_t^{(2)}), t \geq 0\}$ is a discrete-time Markov chain which state space is smaller than the original MDP, i.e. $(0,0)$, $(1,0)$, $(2,0)$, $\cdots$, $(L_1, 0)$, $(0,1)$, $(0,2)$, $\cdots$, $(0, L_2)$. Define α as a parameter such that

$$\alpha = \frac{p_1^{(1)} p_0^{(2)}}{p_0^{(1)} p_1^{(2)}}.$$

Then, the balance equations for $0 < i \leq L_1$ and $0 < j \leq L_2$ are:

$$\begin{aligned}
\pi_{i,0} &= \alpha \pi_{i-1,0} \\
\alpha \pi_{0,j} &= \pi_{0,j-1}.
\end{aligned} \tag{3.21}$$

Since $\pi_{0,0} + \sum_{i,j} \pi_{i,0} + \pi_{0,j} = 1$, we have

$$\pi_{0,0} = \frac{1}{\left(\frac{1-\alpha^{L_1+1}}{1-\alpha}\right) + \left(\frac{1-1/\alpha^{L_2+1}}{1-1/\alpha}\right) - 1}. \tag{3.22}$$

The expected number of transmissions per slot is

$$\mathcal{T}(L_1, L_2) = p_1^{(1)} p_1^{(2)} \pi_{0,0} + p_1^{(2)} \sum_{i=1}^{L_1} \pi_{i,0} + p_1^{(1)} \sum_{j=1}^{L_2} \pi_{0,j} + p_1^{(1)} p_0^{(2)} \pi_{L_1,0} + p_0^{(1)} p_1^{(2)} \pi_{0,L_2}.$$

The average number of packets in the system at the beginning of each slot is

$$\mathcal{H}(L_1, L_2) = \sum_{i=1}^{L_1} i\pi_{i,0} + \sum_{j=1}^{L_2} j\pi_{0,j}. \tag{3.23}$$

Thus upon minimizing we get the optimal thresholds L_1^* and L_2^*. □

Whenever $C_H > 0$, it is relatively straightforward to obtain L_1^* and L_2^*. Since it costs C_T to transmit a packet and C_H for a packet to wait for a slot, it would be better to transmit a packet than make a packet wait for more than C_T/C_H slots. Thus L_1^* and L_2^* would always be less than C_T/C_H. Hence by completely enumerating between 0 and C_T/C_H for both L_1 and L_2, we can obtain L_1^* and L_2^*. One could perhaps find faster techniques than complete enumeration, but it certainly serves the purpose.

Subsequently, we study a special case, $p_1^{(1)} = p_1^{(2)} \triangleq p$, in Theorem 11. Note that $L_1 = L_2 \triangleq L$ as both arrival processes are the same, and $\alpha = 1$ and $\pi_{i,j} = 1/(2L+1)$ for all i, j. Hence, we have

$$\mathcal{T}(L) = \frac{2pL + 2p - p^2}{2L+1}, \tag{3.24}$$

$$\mathcal{H}(L) = \frac{L^2 + L}{2L+1}. \tag{3.25}$$

Define $R = C_T/C_H$. Then the optimal threshold is

$$L^*(p, R) = \arg\min_L \frac{R(2pL + 2p - p^2) + L + L^2}{2L+1}. \tag{3.26}$$

By taking the derivative, we obtain that $L^* = 0$ if $R < 1/(2p - 2p^2)$ and otherwise,

$$L^*(p, R) = \frac{-1 + \sqrt{1 - 2(1 - 2Rp + 2Rp^2)}}{2}. \tag{3.27}$$

We can observe that $L^*(p, R)$ is a concave function *w.r.t.* p. Given R fixed, $L^*(1/2, R) = (\sqrt{R-1} - 1)/2$ is the largest optimal threshold among various values of p. When $p < 1/2$, the optimal-threshold decreases as there is a relatively lower probability for packets in one queue to wait for a coding pair in another queue. When $p > 1/2$, there will be a coding pair already in the relay node with a higher probability, and therefore the optimal-threshold also decreases. Moreover, $L^*(1/2, R) = \mathcal{O}(\sqrt{R})$, so the maximum optimal threshold grows with the square root of R, but not linearly. When p is very small, $L^*(p, R) = \mathcal{O}(\sqrt{Rp})$ grows slower than $L^*(1/2, R)$.

D. Numerical Studies

In this section we present several numerical results to demonstrate the analytical formulation as well as its extensions. We study the performance of a number of policies:

1. Opportunistic Coding: when a packet arrives, coding is performed if a compatible packet is available, otherwise transmission takes place immediately.

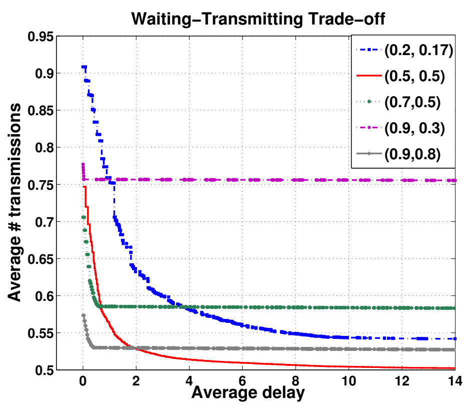
2. Queue-length-based threshold: a Stationary Deterministic policy that our analysis suggests it should be optimal for i.i.d arrival processes.

3. Randomized-Queuelength-based threshold: a Stationary policy that Randomizes over deterministic policies. We expect that it would not perform any better than deterministic queue-length-based policies.

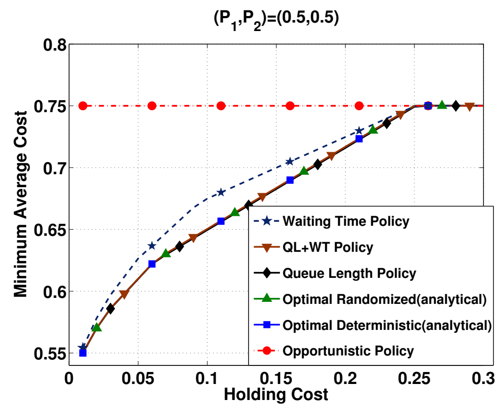
4. Queue-length-plus-Waiting-time-based thresholds: a History Dependent policy which is likely to give the best possible performance.

5. Waiting-time-based thresholds: an HR policy that we create for the purpose of comparison to illustrate that history on its own is only of limited value.

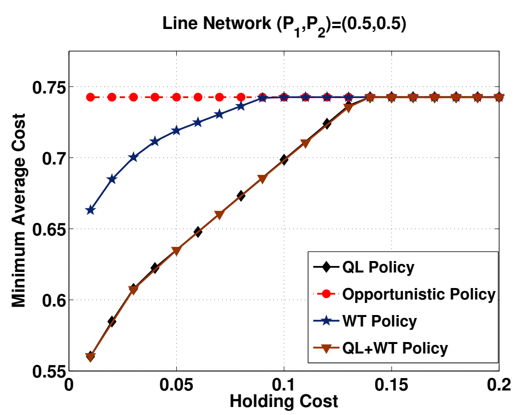
We simulate these policies on two different cases: (i) the single relay with Bernoulli arrivals (Figures 7(a) and 7(b)) and (ii) a line network with 4 nodes, in



(a) Single relay network, queue-length threshold policy, Bernoulli arrival rates (p_1, p_2)



(b) Single relay network, Bernoulli arrival rates $(0.5, 0.5)$, the costs are normalized by the transmission cost



(c) Line network with two intermediate nodes, two Bernoulli flows with mean arrival rates $(0.5, 0.5)$

Fig. 7. The performance of network coding enabled relay networks in terms of transmission and holding costs

which the sources are Bernoulli (Figure 7(c)). Note that in this case, since the departures from one queue determine the arrivals into the other queue, the arrival processes are significantly different from Bernoulli. Our simulations are done in Java and for each scenario we report the average results of 10^5 iterations.

Our numerical studies illustrate that, as expected, a deterministic queue-length based policy is optimal for different network scenarios. The results are intriguing as they suggest that achieving a near-perfect tradeoff between waiting and transmission costs is possible using simple policies; and coupled with optimal network-coding aware routing policies like the one in [20], have the potential to exploit the positive externalities that network coding offers.

E. Further Discussions and Extensions

We understand that the average-optimal policy is stationary and threshold based for the *i.i.d.* arrival process with the service rate 1 packet per time unit. Two more general models are discussed here.

1. Batched Service

Assume that the relay R can serve a group of packets with the size of $\mathcal{K}$ at end of the time slot. R decides to transmit, $A_t = 1$, or to wait $A_t = 0$ at the end of every time slot. The holding cost per unit time for a packet is C_H, while C_T is the cost to transmit a batched packet. Then the immediate cost at time t is

$$C^{(\mathcal{K})}(Q_t, A_t) = C_H([Q_t^{(1)} - A_t\mathcal{K}]^+ + [Q_t^{(2)} - A_t\mathcal{K}]^+) + C_T A_t, \tag{3.28}$$

We also want to find the optimal policy θ^* that minimizes the long-time average cost $V^{(\mathcal{K})}(\theta)$, called $\mathcal{K}$-MDP$\{(Qt, At), t \geq 0\}$ problem,

$$V^{(\mathcal{K})}(\theta) = \lim_{K\to\infty} \frac{1}{K+1} \mathbb{E}_\theta \left[\sum_{t=0}^{K} C^{(\mathcal{K})}(Q_t, A_t) | Q_0 = (0,0) \right]. \tag{3.29}$$

Notice that the best policy might not just transmit when both queues are non-empty. When $\mathcal{K} > 1$, R might also want to wait even if $Q_t^{(1)} Q_t^{(2)} > 0$ because the batched service of size less than $\mathcal{K}$ has the same transmission cost C_T. The optimality equation of the expected α-discounted cost is revised as

$$\begin{aligned} V_\alpha^{(\mathcal{K})}(i,j) = \quad & \min_{a\in\{0,1\}} \Big[C_H([i - a\mathcal{K}]^+ + [j - a\mathcal{K}]^+) + C_T a + \\ & \mathbb{E}[V_\alpha^{(\mathcal{K})}([i - a\mathcal{K}]^+ + \mathcal{A}_1, [j - a\mathcal{K}]^+ + \mathcal{A}_2)] \Big] \end{aligned} \tag{3.30}$$

We can get the following results.

Theorem 12. *Given α and $\mathcal{K}$, $V_\alpha^{(\mathcal{K})}(i,j)$ is non-decreasing, submodular, and $\mathcal{K}$-subconvex. Moreover, there is an α-optimal policy that is of threshold type. For a fixed j, the α-optimal policy is monotone w.r.t. i, and vice versa.*

Theorem 13. *Consider any i.i.d. arrival processes to both queues. For the $\mathcal{K}$-MDP$\{(Q_t, A_t), t \geq 0\}$, the average-optimal policy is of threshold type. Given $j = \tilde{j}$ fixed, there exists the optimal threshold $L_{\tilde{j}}^*$ such that the optimal stationary and deterministic policy in state $(i, \tilde{j})$ is to wait if $i \leq L_{\tilde{j}}^*$, and to transmit if $i > L_{\tilde{j}}^*$. Similar argument is true for the other queue.*

2. Markov-Modulated Arrival Process

So far we only considered *i.i.d.* arrival processes, here we will study a specific arrival process with memory, *i.e.,* Markov-modulated arrival process (MMAP). The service capacity of R is focused on $\mathcal{K} = 1$ packet. Let $\mathcal{N}^{(i)} = \{0, 1, \cdots, N^{(i)}\}$ be the state space of MMAP at node i, with the transition probability $p^{(i)}_{k,l}$ where $k, l \in \mathcal{N}^{(i)}$. The number of packets generated by the node i at time t is denoted by $\mathcal{N}^{(i)}_t \in \mathcal{N}^{(i)}$. Then the decision of R is made based on the observation of $(Q^{(1)}_t, Q^{(2)}_t, \mathcal{N}^{(1)}_t, \mathcal{N}^{(2)}_t)$. Similarly, the objective is to find the optimal policy that minimizes the long-term average cost, named MMAP-MDP$\{(Q^{(1)}_t, Q^{(2)}_t, \mathcal{N}^{(1)}_t, \mathcal{N}^{(2)}_t) : t \geq 0\}$ problem. The optimality equation of the expected α-discounted cost becomes

$$\begin{aligned} V^{\text{MMAP}}_\alpha(i, j, n_1, n_2) = \quad & \min_{a \in \{0,1\}} [C_H([i-a]^+ + [j-a]^+) + C_T a + \\ & \alpha \sum_{k,l} p^{(1)}_{n_1,k} p^{(2)}_{n_2,l} V^{\text{MMAP}}_\alpha([i-a]^+ + k, [j-a]^+ + l, k, l)]. \end{aligned}$$

Then we can conclude the following results.

Theorem 14. *Given $n_1 \in \mathcal{N}^{(1)}$ and $n_2 \in \mathcal{N}^{(2)}$, $V^{MMAP}_\alpha(i, j, n_1, n_2)$ is non-decreasing, submodular, and subconvex w.r.t. i and j. Moreover, there is an α-optimal policy that is of threshold type. Fixed n_1 and n_2, the α-optimal policy is monotone w.r.t. i when j is fixed, and vice versa.*

Theorem 15. *Consider any MMAP arrival process. The average-optimal policy for the MMAP-MDP$\{(Q^{(1)}_t, Q^{(2)}_t, \mathcal{N}^{(1)}_t, \mathcal{N}^{(2)}_t) : t \geq 0\}$ is of multiple thresholds type. There exists a set of optimal thresholds $\{L^*_{1,n_1,n_2}\}$ and $\{L^*_{2,n_1,n_2}\}$, where $n_1 \in \mathcal{N}^{(1)}$ and $n_2 \in \mathcal{N}^{(2)}$, so that the optimal stationary decision in states $(i, 0, n_1, n_2)$ is to wait if $i \leq L^*_{1,n_1,n_2}$, and to transmit without coding if $i > L^*_{1,n_1,n_2}$, while in state $(0, j, n_1, n_2)$ is to wait if $j \leq L^*_{2,n_1,n_2}$, and to transmit without coding if $j > L^*_{2,n_1,n_2}$.*

F. Summary and Future Work

In this chapter, we investigated the delicate trade-off between waiting and transmitting using network coding. We started with the idea of exploring the whole space of history dependent policies, but showed step-by-step how we could move to simpler regimes, finally culminating in a stationary deterministic queue-length threshold based policy. The policy is attractive because its simplicity enables us to characterize the thresholds completely, and we can easily illustrate its performance on multiple networks. We showed by simulation how the performance of the policy is optimal in the *i.i.d* (Bernoulli) arrival scenario, and how it also does well in other situations such as for line networks.

An immediate extension to this work will be investigating a similar trade-off in general queuing networks with shared resources. Also one can check how the results will change for more general and possibly heterogeneous cost functions.

The aim of this chapter was to develop an analytical framework for studying the performance of network coding when there are constraints on the service delay. In the next chapter, we will study a practical situation in which a simple random linear network coding scheme plays a fundamental role. We will consider a peer-to-peer network that supports delay sensitive applications like content distribution with delay constraints on the delivery time and multimedia live streaming which naturally requires strict delay guarantees. We shall see how using network coding can (i) simplify the construction of efficient algorithms, and (ii) enhance the performance of the system.

CHAPTER IV

WIRELESS BROADCAST P2P NETWORKS

Delay sensitive applications are beginning to dominate Internet applications [30], accompanied by the rising popularity of smart, handheld devices. Such devices usually have multiple communication interfaces. For example, smart phones have 3G (or 4G), WiFi, and Bluetooth interfaces. Cellular data interfaces are currently designed for unicast communication between a base station to a device, and are expensive, both in terms of energy and dollar-cost of usage. Other interfaces may be used for ad-hoc peer-to-peer broadcast transmissions, and are usually less expensive. The hardware and applications to exploit one or both interfaces for proximate P2P communication are already making an appearance [31].

The objective of this work [1] is to design algorithms to exploit both base-station to peer (B2P) and peer-to-peer (P2P) wireless interfaces simultaneously such that a group of proximate peer devices can all obtain some common information within a specified deadline and at lowest cost. Essentially, the problem that we are interested in is the *timely synchronization* of data on multiple wireless devices with minimal infrastructure support. Applications can be divided into two categories, namely, (i) *content distribution* and (ii) *multimedia streaming*. Some examples of the content distribution application might be emergency response situations (here, cellular data support is limited), as well as ensuring that purchases such as software and media files are simultaneously available on multiple smart devices (here, cellular data support is expensive). Furthermore, multimedia streaming applications are becoming increasingly popular. Based on our framework, we can develop new efficient schemes to

[1] Parts of this work were done in collaboration with other students, Mayank Manjrekar and Swetha Sampath.

stream a common multimedia channel to a group of cooperative peers. The schemes that we develop will be able to reduce the traffic load on the cellular network and the media server, and also increase the quality of experience of the peers while decreasing their cost.

Our setup is illustrated in Figure 8, where peers use both their cellular and ad-hoc interfaces simultaneously. All information enters the system from a media server through the cellular interfaces, and is then re-distributed proximately via P2P broadcast. Each block of information is divided into clunks for transmission. The questions that we need to answer are (i) how long should peers utilize their cellular data interfaces? and (ii) which peer should broadcast what chunk at each time? In order to meet a strict time deadline, there are two intuitive requirements on the information state of the system at the time of turning off cellular data, namely, (i) each peer individually, and (ii) the system as a whole should have received sufficient B2P transmissions. We seek to arrive at an appropriate information state by using both expensive unicast B2P transmissions and low-cost P2P broadcasts for some duration of time, such that the P2P broadcasts acting alone can make up the balance afterward.

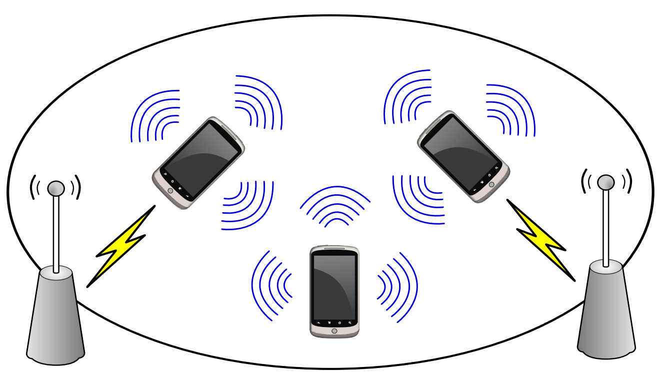


Fig. 8. Hybrid P2P wireless network. Each device can utilize both base-station-to-peer (B2P) and peer-to-peer (P2P) communication.

If chunks are sent in an uncoded fashion, every base station and every peer would have to coordinate with each other so as to send an optimal sequence of chunks that would minimize the number of required B2P and P2P transmissions for a given deadline. Furthermore, if any chunk is lost, it would require the whole optimal sequence to be re-calculated. To avoid this complexity, each transmitted chunk is created using random linear coding [32, 33] so that chunk identities can be ignored. Here, each coded chunk is a linear combination of the original chunks, with the coefficients drawn randomly from a finite field. Thus, each coded chunk can be thought of as an element in a vector space with the scalars in a finite field. The information available with each peer can then be represented by a matrix that contains all the vectors that it has received thus far, and the block of information can be decoded when this matrix is of full rank. Each coded chunk received can either increase the rank of the matrix by one (adds a *degree of freedom (DoF)*), or might have no impact if it can be recreated as a linear combination of existing vectors with the peer.

The problem of efficiently exchanging common information over a broadcast P2P network (without existence of the external B2P network) was recently studied in [34–36]. The common objective in these papers is to deliver a block of information to a group of peers, with some initial conditions, in the minimum number of broadcast transmissions. The problem of scheduling the P2P network (*i.e.,* finding the right peer to broadcast at each slot) in our content distribution framework has some similarities with the problem studied in these papers. All of the aforementioned works consider a reliable model for the broadcast links. For analytical purposes, we also assume that our wireless broadcast network is lossless. Indeed we have been conducting experiments on an Android smart phone testbed, and have empirically found that P2P WiFi broadcast transmissions essentially succeed with probability one due to the proximity of the devices. We will comment on the challenges of considering lossy

P2P transmissions in Section C. Despite the similarities of our model and the ones mentioned above, we face a new challenge of dealing with a hybrid network (*i.e.*, simultaneous B2P and P2P transmissions). Also our QoS metric, minimum cost timely synchronization, is essentially different from their objective. To solve this new challenge, we use ideas from queueing theory and develop some results on the performance of the *longest-queue-first* algorithm, which will be used in scheduling our hybrid network. We further generalize the results of the broadcast P2P case in [34] to a larger class of algorithms.

In the context of P2P multimedia streaming applications, the literature is very rich and it is well accepted that network coding can indeed simplify the implementation and yet improve the performance. A few examples of systems built based on the idea of network coding are Lava [37], CoolStreaming [38], PPLive [39], [40] and UUSee [41]. While all the above models only consider unicast transmissions among peers, our P2P framework is one of the few models in this context that allows having broadcast transmissions over the P2P network. Although in wireless networks it is very reasonable and also efficient to consider broadcast transmissions, the coordination of transmissions is much more challenging compared to the systems with only unicast sessions.

Closest to our multimedia streaming model is [42] that investigates the problem of managing multiple interfaces for the purpose of cooperatively sharing video content over a broadcast P2P network. Unlike our delay sensitive model, they try to maximize a utility function of the average information flow (video) rate achieved by the peers. Moreover, we will specifically study a *live streaming* application as opposed to *on-demand streaming*. In this model, we need to download and play out the stream of multimedia in a timely manner in order to meet fixed service deadlines while guaranteeing some predefined minimum quality requirements.

Delay sensitive communication in P2P networks has attracted significant recent interest. For example, [43, 44] develop analytical models on the trade-off between the steady state probability of missing a chunk and buffer size, under different block selection policies in a streaming situation. Unlike our model, they consider live streaming with a single deterministic channel between each pair of peers. There is also work on characterizing cost vs. quality tradeoffs in choosing one or the other wireless interface [33, 45]. However, neither of these consider P2P communications.

In what follows, we first study the content distribution application in Section A. The live multimedia streaming system will be discussed in Section B. Section C will conclude this chapter by summarizing our results and presenting some ideas for future work.

A. Content Distribution with Service Deadline

In this section, we shall consider the problem of delay sensitive content distribution over a P2P network. We organized this section as follows.

We begin our study by formally describing the system model in Subsection 1. Our overall objective is to develop algorithms for content distribution to a set of cooperative wireless users which require some service deadline.

In Subsection 2, we study the content distribution problem in a pure P2P network (no B2P) with an arbitrary initialization of chunks. Our objective is to find an algorithm whereby all peers would possess full rank matrices after the smallest number of broadcast P2P transmissions. We propose a *Non-min-Rank-First (NmRF)* scheme for P2P transmissions, wherein any one peer *except* the one (or ones) with smallest matrix rank may broadcast a random linear combination. We evaluate the algorithm for two cases: (i) uncoded initialization, where peers initially possess uncoded packets and (ii) coded initialization. We show that the proposed scheme is optimal for large field sizes, and obtain performance bounds for finite field sizes.

The next step consists of integrating both P2P and B2P transmissions, presented in Subsection 3. Here, our quality of service (QoS) target is to ensure that all peers obtain the block with a target probability η by a deadline T. We pose this problem as an offline stopping time problem for the B2P transmissions. Thus, we have two phases whose durations are determined in an offline manner. In the first phase, both B2P and P2P networks are used, while in the second phase only P2P broadcasts are used to complete the dissemination of the content. We show that the appropriate P2P broadcast schemes during phase 1 and 2 are, respectively, *Max-Rank-First (MRF)* and *Non-min-Rank-First* for any stopping time. We map the question of selecting the appropriate peer to broadcast in phase 1 to that of choosing the right queue to serve in a system of multiple queues. A basic insight that forms the core of the proof is that the *longest-queue-first* service regime actually *maximizes the minimum queue length* (the usual result is that it minimizes the maximum queue length), which results in roughly equal ranks for all peers. This in turn ensures that all peers can meet the delay target. We then show how to compute the optimal stopping time T_1 that would achieve the QoS targets with the minimum cost in the case of large field sizes, and compute performance bounds of the algorithm in the finite field case.

In Subsection 4, we show how we can implement the peer selection algorithms, *MRF* and *NmRF* schemes, in a decentralized fashion. We will see that with a small overhead in the proposed distributed scheme, peers can individually perform the algorithm without the need for a central scheduler or excessive communications to coordinate the transmissions.

Finally, we illustrate the main insights using the simulations on a hybrid network in Subsection 5, and verify the performance of our suggested algorithms.

1. System Model

Consider Figure 8 again. There are M peers, denoted by $i \in \{1, \ldots, M\}$, all interested in receiving the same block of information. The block is further divided into N chunks. Each peer has two communication interfaces—broadcast P2P and unicast B2P. We assume that each B2P transmission has unit cost, while P2P broadcasts are free. We assume that the time is slotted and the duration of a time slot is enough for transmitting at most one chunk over either interface.

A transmitted chunk over a B2P channel may not be successfully delivered due to the unreliability of the channel. We model each B2P unicast channel as an independent Bernoulli random variable with parameter p. Since the peers are all proximate to each other, they have statistically similar B2P channels. Thus, peers access base stations with identical probabilities of success at any particular time. We denote the total number of chunks delivered to peer i via the B2P network by the beginning of slot t using $e_i[t]$. We assume that each B2P transmission has a unit cost.

The common P2P broadcast channel can only be used by at most one peer, $u[t] \in \{1, \ldots, M\}$, in each time slot t. We assume that the peers are near enough to each other that all of them receive each broadcast successfully. Therefore, the average data rate over the broadcast P2P channel is $\frac{1}{p}$ of the rate over each B2P channel In practice, the 3G channel is much less reliable than WiFi, and further, the ratio of data rates achievable using WiFi vs. 3G is of the order of $5:1$.

$x_i[t]$ and $r_i[t]$ are used to denote respectively the total number of transmitted and received chunks by peer i via P2P by the beginning of slot t. Note that according to the model

$$\sum_i x_i[t] - \sum_i x_i[t-1] \leq 1 \quad \text{for all } t > 0 \tag{4.1}$$

and because of the lossless nature of the broadcast P2P network, we have

$$\sum_{j \neq i} x_j[t] = r_i[t].$$

To avoid the problem of piece selection we use random linear network coding. Each coded chunk is a random linear combination of the N original chunks in the block. The linear combinations are carried out with coefficients in F_q, a finite field of size q. Therefore, we can associate a vector $\gamma = (\alpha_1, ..., \alpha_N)$ to each coded chunk, where $\alpha_i \in F_q$ is the coefficient of the i^{th} original chunk in the corresponding combination. We will interchangeably use a coded chunk and its corresponding vector γ, when there is no ambiguity.

Let $S_i[t]$ be the set of vectors possessed by peer i at time t, with cardinality $\hat{n}_i[t]$. Thus,

$$\hat{n}_i[t] = e_i[t] + r_i[t] = e_i[t] - x_i[t] + \textstyle\sum_j x_j[t]. \tag{4.2}$$

Further, we represent the dimension of the subspace spanned by $S_i[t]$ using $n_i[t]$, which is accordingly called the *rank* of peer i at time t.

Since the P2P transmissions are heard by everybody, chunks that have already been broadcast via P2P may not be used in future transmissions. Therefore, without loss of generality, we assume that at time t, $u[t]$ transmits a chunk which is a linear combination of the chunks received from the B2P network. This implies that

$$x_i[t] \leq e_i[t] \quad \text{for all } t \geq 0. \tag{4.3}$$

Observation 6. *Given a sequence of transmissions* $(\gamma_1, \ldots, \gamma_t)$ *for P2P dissemination, any permutation* $\Pi(\gamma_1, \ldots, \gamma_t)$ *is also equally effective, i.e., the order of P2P transmissions is not relevant.*

We define QoS metric (η, T) in the form of a requirement that all peers are able to decode the block by the end of time T with probability at least equal to η. We seek a scheme that would achieve this target at the lowest cost of using B2P transmissions. Note that a feasible P2P scheme $\{(x_1[t], ..., x_M[t])\}_{t=0}^{T}$ must satisfy (4.1) and (4.3). Also, in order to decode the original N chunks at time T, each peer i must have $n_i[T] = N$.

2. P2P Broadcast Network

In our P2P broadcast model, only one peer can transmit at a time. In this section, we attempt to obtain a clear understanding of which peer this should be. Hence, we study the case of pure P2P broadcasts, assuming that each peer i was initialized with a set of chunks $S_i[0]$ at time $t = 0$ (*i.e.,* $e_i[t] = e_i[0] := e_i$ for $t \geq 0$). If $\bigcup_i S_i[0]$ spans the whole space of dimension N, then the P2P broadcasts can be used to ensure that all peers can eventually recover the block. We would like to find algorithms that do so with the smallest number of broadcasts. We will study the problem under two cases, namely, (i) uncoded initialization, and (ii) coded initialization.

a. Uncoded Initialization

In this variation of the problem, we study the basic case in which each peer initially has a subset of uncoded original chunks. Sprintson *et al.* [34] study this case, and show that a *Max-rank-first* algorithm will achieve the minimum number of transmissions with probability at least $1 - \frac{NM}{q}$.

Max-Rank-First (MRF) algorithm: At any time t, one of the peers with the maximum rank transmits, *i.e.,* $u[t] \in \{\arg\max_i n_i[t]\}$. Further, if the ranks of all peers are equal and smaller than N, an arbitrary peer transmits.

In what follows, we propose a new scheme, *Non-min-Rank-First (NmRF)*, and we show that it performs equally well as the *MRF* algorithm.

Non-min-Rank-First (NmRF) algorithm: At any time t, one of the peers, except those with the minimum rank, transmits, *i.e.,* $u[t] \in \{1, \ldots, M\} \backslash \{\arg\min_i n_i[t]\}$. Further, if the ranks of all peers are equal and smaller than N, an arbitrary peer transmits.

Note that the *MRF* algorithm is a special case of the *NmRF* algorithm. The main value of the *NmRF* scheme is that it provides a larger decision space which can be utilized in different ways. For example, improving the level of fairness in the system is possible by letting a larger group of peers attend in the transmissions. Also since there are more candidates for transmitting at each slot, the system is less vulnerable to unexpected changes in the set of peers. More interestingly, in the distributed implementation of this system, we will need less number of coordinating signals, because finding a non-min-rank peer is much easier than one of the max-rank peers.

We now present results on the performance of *NmRF*.

Lemma 12. *If the peers are initialized with uncoded chunks,* NmRF *disseminates all N degrees of freedom to M peers using the minimum number of transmissions with probability at least* $1 - \frac{NM}{q}$.

Proof. At time t, given $S_i[t]$ for all $i \in \{1, ..., M\}$, we denote an optimal set of transmissions to disseminate all degrees of freedom by $\mathcal{T}[t] = (\gamma_t, \ldots, \gamma_{t+Dim(\mathcal{T}[t])-1})$ such that

$$\gamma_l \in Span(S_{u[l]}[0]) \quad \forall l \geq t \quad \text{and} \quad Dim(\mathcal{T}[t] \cup S_i[t]) = N \quad \forall i, \tag{4.4}$$

where $Span(S)$ is the subspace spanned by the vectors in the set S, and $Dim(S)$ is the dimension of this subspace. Note that $Dim(\mathcal{T}[t]) + Dim(S_i[t]) \geq Dim(\mathcal{T}[t] \cup S_i[t]) = N$, for all i, which implies

$$Dim(\mathcal{T}[t]) \geq N - \min_i Dim(S_i[t]) = N - \min_i n_i[t]. \tag{4.5}$$

Also, it is shown [34] that when all the ranks are equal, *i.e.*, $n_i[t] = n < N$ for all i, then

$$Dim(\mathcal{T}[t]) \geq N - n + 1. \tag{4.6}$$

This suggests that for any non-minimum rank peer j (or any peer j when all the ranks are equal), there exists at least one $\gamma_{\hat{t}} \in \mathcal{T}[t] \cap Span(S_j[t])$. Because otherwise $Dim(\mathcal{T}[t]) + Dim(S_j[t]) = Dim(\mathcal{T}[t] \cup S_j[t]) = N$, which contradicts (4.5) for the non-minimum rank peer j (or (4.6) when all the ranks are equal). Consequently, peer j can generate $\gamma_{\hat{t}}$, and transmit it at time t.

The proof of probability bound $1 - \frac{NM}{q}$ follows a similar approach as in [34] and is omitted for brevity. □

Corollary 5. *For the infinite field size q, the* NmRF *algorithm is optimal.*

Note that the above results also hold for the case where the chunks are initially coded. In what follows, we will specifically study this latter case and show that by initially providing randomly coded chunks, better performance guarantees can be achieved.

b. Coded Initialization

In this subsection we assume that the media server performs the network coding initially and transmits coded chunks instead of the original chunks. Therefore, peers are assumed to possess randomly coded chunks at the beginning. We immediately see that combining the initial randomly coded chunks still further for P2P broadcasts cannot improve the performance on average. Therefore, we assume that at time t, $u[t]$ transmits any one of its initial chunks that has not yet been transmitted. This will alleviate the need for encoding procedure at the peer devices.

Suppose that a peer has rank $n < N$, *i.e,* it possesses n linearly independent coded chunks. For a new coded chunk to be useful to this peer, it has to be linearly independent of the previous n chunks. However, there is always some non-zero probability that the new random linearly coded chunk is not useful. An upper bound on this probability is $\frac{1}{q}$, which does not depend on n (see [32]). If we assume that q is large enough, then every new randomly coded chunk is indeed useful (*i.e,* $n_i[t] = \hat{n}_i[t]$). This suggests one can simply count the number of arrivals to a peer to compute its rank, *i.e,* $n_i[t] = \hat{n}_i[t]$. Hence, we can define the state of the system at time t by giving the number of chunks from external (B2P) sources possessed by each peer, $(e_1[t], \ldots, e_M[t])$, and the amount of shared content between them, $(x_1[t], \ldots, x_M[t])$.

The following theorem provides an expression for the minimum number of required transmissions to disseminate all degrees of freedom, in case of infinite field size.

Theorem 16. *Let the system state at time* $t \geq 0$ *be* $((e_1, \ldots, e_M), (x_1[t], \ldots, x_M[t]))$ *and* $\sum_i e_i \geq N$. *The minimum number of slots required to complete the dissemination using only P2P broadcasts is*

$$\tau^* = \left\lceil N - \sum_i x_i[t] - \min\left(\frac{\sum_i e_i - N}{M-1}, \min_i (e_i - x_i[t])\right)\right\rceil.$$

Proof. Note that in this proof, we deal with infinite field size q. Let $i^* \in \{\arg\max_i n_i[t]\}$ and $n_{i^*}[t] = n_{max}$. In any trajectory of P2P transmissions, peer i^* needs to receive $N - n_{max}$ broadcasts. Since the order of the transmissions is irrelevant, let the first

$N - n_{max}$ transmissions be those needed by i^*. Therefore after $\tau_1 = N - n_{max}$ slots, i^* will become full-rank, *i.e.*, $n_{i^*}[t + N - n_{max}] = N$. At this time, i^* can take care of all remaining P2P transmissions, whose minimum number is $\tau_2 = N - \min_i n_i[t + N - n_{max}]$.

Let y_i be the number of times peer i transmits in the first τ_1 slots. Then we require

$$\textstyle\sum_{i \neq i^*} y_i = N - n_{max} \quad \text{and} \quad y_i \leq e_i - x_i[t], \tag{4.7}$$

and we have, $n_i[t + N - n_{max}] = n_i[t] + N - n_{max} - y_i$.

Consequently, the minimum time, $\tau^* = \min_{(y_i)_i}(\tau_1 + \tau_2)$, to deliver all chunks is

$$\tau^* = N - \max_{(y_i)_i} \left(\min_{i \neq i^*} (n_i[t] - y_i) \right) \tag{4.8}$$

subject to the constraints given in (4.7). From (4.2), τ^* can be rewritten as

$$\tau^* = N - \sum_i x_i[t] - \max_{(y_i)_i} \left(\min_{i \neq i^*} (e_i - x_i[t] - y_i) \right). \tag{4.9}$$

Solving the above maximization gives the following solution to the minimum completion time:

$$\tau^* = \left\lceil N - \sum_i x_i[t] - \min \left(\frac{\sum_i e_i - N}{M - 1}, \min_i (e_i - x_i[t]) \right) \right\rceil.$$

□

Note that although the solution to the maximization in (4.9) will provide an optimal P2P scheme in an offline manner, from Corollary 5 the *NmRF* online scheme achieves the same solution.

Next, we consider the finite field case. The following Theorem provides a lower bound on the probability that the *NmRF* algorithm completes dissemination by time

τ^*.

Theorem 17. *When the random linear coding coefficients are drawn from a finite field* F_q*, the* NmRF *algorithm completes dissemination of the block to all peers by time* τ^* *with probability at least* $1 - \frac{M}{q-1}$.

In the proof, we will use the following useful lemma.

Lemma 13. *A matrix of dimension* $R \times N$*, whose elements are drawn uniformly at random from a finite field* F_q*, is full-rank with probability at least* $1 - \frac{1}{q^{|N-R|}(q-1)}$*, where* q *is the field size.*

Proof. Assume $R \leq N$. The probability of having R linearly independent rows is

$$\frac{(q^N-1)(q^N-q)(q^N-q^2)\ldots(q^N-q^{R-1})}{q^{RN}}$$
$$= \prod_{i=N+1-R}^{N}(1-q^{-i}) \geq 1 - \sum_{i=N+1-R}^{\infty}\frac{1}{q^i} = 1 - \frac{1}{q^{N-R}(q-1)},$$

in which the denominator is the number of all matrices of dimension $R \times N$ in the field F_q. Also the r^{th} term in the numerator is the number of all possibilities for the r^{th} row, such that it is linearly independent of the previous $r-1$ rows. For $R > N$, it can be shown similarly that the probability of obtaining N linearly independent columns is at least $1 - \frac{1}{q^{R-N}(q-1)}$. □

Proof. (Theorem 17) Recall that $n_i[t]$ and $\hat{n}_i[t]$ are the rank and the number of chunks, respectively, available with i. In the *NmRF* algorithm, $u[t] \in \{1, \ldots, M\} \backslash \{\arg\min_i n_i[t]\}$. First, we show that this theorem holds even if $u[t] \in \{1, \ldots, M\} \backslash \{\arg\min \hat{n}_i[t]\}$, *i.e.*, at each time t, a peer whose total number of received chunks is not minimum can transmit.

For the infinite field size case, $n_i[t] = \hat{n}_i[t]$, *i.e.*, both the above schemes are identical, and, as shown in Theorem 16, both result in $\hat{n}_i[t+\tau^*] \geq N$ for all i. Hence, we want to compute

$$\begin{gathered}\mathbb{P}\left(\forall i: n_i[t+\tau^*] = N \middle|\ \forall i: \hat{n}_i[t+\tau^*] \geq N\right) = \\ 1 - \mathbb{P}\left(\exists i: n_i[t+\tau^*] < N \middle|\ \forall i: \hat{n}_i[t+\tau^*] \geq N\right) \geq \\ 1 - \sum_i \mathbb{P}\left(n_i[t+\tau^*] < N \middle|\ \hat{n}_i[t+\tau^*] \geq N\right),\end{gathered} \tag{4.10}$$

where the last inequality follows from the union bound. Note that $\hat{n}_i[t+\tau^*] \geq N$ means peer i has received at least N randomly generated chunks by time $t+\tau^*$, and the probability that the corresponding $\hat{n}_i[t+\tau^*] \times N$ matrix is not full-rank is $\mathbb{P}\left(n_i[t+\tau^*] < N | \hat{n}_i[t+\tau^*] \geq N\right)$. From Lemma 13,

$$\mathbb{P}\left(n_i[t+\tau^*] < N \middle|\ \hat{n}_i[t+\tau^*] \geq N\right) \leq 1/(q-1). \tag{4.11}$$

Consequently, $\mathbb{P}\left(\forall i: n_i[t+\tau^*] = N \middle|\ \forall i: \hat{n}_i[t+\tau^*] \geq N\right) \geq 1 - \frac{M}{q-1}$.

We observe that for a given state at time t, if the *NmRF* algorithm fails to finish the dissemination by time $t+\tau^*$, then the algorithm which chooses $u[t] \in \{1,\ldots,M\} \backslash \{\arg\min \hat{n}_i[t]\}$ also fails. Therefore, the *NmRF* algorithm performs better than the other scheme, and the same lower bound is still valid. □

Observation 7. *Note that unlike the bound for the uncoded initialization case,* $1 - \frac{NM}{q}$*, the performance guarantee found in Theorem 17 is independent of* N*, the number of chunks. Thus, the probability of failure with a finite field size is likely to be much lower on average when we start with randomly coded chunks.*

To summarize, it is highly preferred to implement the encoding procedure at the media server. The advantage of this setup is two-fold: (i) need for implementing a much simpler application at the peers, since they do not need to perform the encoding process and (ii) achieving potentially a much better performance.

3. Hybrid Network

In this section, we consider the problem when the B2P network coexists with the P2P network. We assume that at $t = 0$ peers have no chunks, and can use both B2P and P2P transmissions to obtain chunks of the block. Recall the QoS constraint (η, T), which requires that all peers must receive the whole block with probability η by deadline T. All peers have identical Bernoulli B2P channels with equal (unit) cost and success probability p, which they use to receive randomly coded chunks from base stations. As discussed in Section b, there is no performance loss if these chunks are transmitted via P2P broadcasts without performing any further coding. The questions that we must answer are: (i) how long should the B2P channels be used? and (ii) which peer should transmit using P2P broadcast at each time? We will seek an offline solution for the first question. Therefore, it suffices to consider only those schemes in which we use the B2P channels for the first few time slots and stop using them afterward.

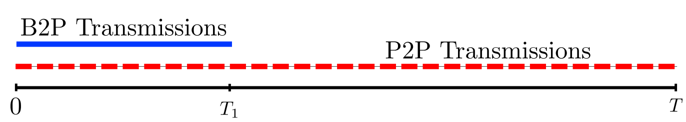


Fig. 9. Two-phase scheme. Both B2P and P2P transmissions take place in Phase 1, whereas only P2P transmissions occur in Phase 2.

If we assume $N \gg M$, since the B2P channels are symmetric, our objective will be to determine a time T_1 such that the B2P channels are *all* turned off at this time. We will choose this stopping time in an offline fashion in such a way that the QoS target can be met with lowest cost. The time-line in our hybrid system is illustrated in Figure 9. We have two phases, where both B2P and P2P methods are used in Phase 1, while only P2P is used in Phase 2. Further, due to symmetry of the B2P channels, *all* B2P channels will be turned off simultaneously at T_1, with the total cost being MT_1. We first consider the problem in the case of infinite field size.

a. Infinite Field Size

From Figure 9, we observe that the evolution of information state in Phase 2 is identical to the pure P2P broadcast system in Subsection 2. Theorem 16 can be used to specify the initial conditions needed to attain any target completion time (deterministically) in this situation. Thus, from Theorem 16, we define the class of states C_τ that can finish within time τ purely using P2P broadcasts by

$$C_\tau = \Big\{((e_1, ..., e_M), (x_1, ..., x_M)) \;\Big|\; \sum_i e_i \geq N,\; x_i \leq e_i,$$
$$\tau \geq N - \sum_i x_i - \min\left(\frac{\sum_i e_i - N}{M-1}, \min_i(e_i - x_i)\right)\Big\}$$

Note that the peer selection algorithm during Phase 2 would follow *NmRF.* Then the optimal stopping time T_1 must be the smallest value (lowest cost) such that the probability that the system state $((e_1[T_1], \ldots, e_M[T_1]), (x_1[T_1], \ldots, x_M[T_1]))$ is in C_{T-T_1} at the end of Phase 1 is at least η.

Now, consider the evolution of information state during Phase 1. All B2P channels are used during this interval. However, P2P broadcast can only happen at time t if at least one peer has a chunk that has not yet been broadcast via P2P. If there is no such peer, P2P broadcast must remain idle in that time slot. We call all P2P transmission policies that transmit whenever it is possible to do so as *work conserving.* Denote the event of P2P idleness in time slot s by $I_s \in \{0, 1\}$. Hence, for all work conserving P2P schemes

$$I_s = 1 \quad \text{only if } \textstyle\sum_{i=1}^{M} (e_i[s] - x_i[s]) = 0. \tag{4.12}$$

Denote the cumulative P2P idle time within the time interval $[0, t-1]$ by $I[t]$, *i.e.,*

$I[t] = \sum_{s=0}^{t-1} I_s$. The following theorem expresses the conditions for (η, T) to be achievable.

Theorem 18. *A target QoS (η, T) is achievable if and only if there exists a stopping time $T_1 \leq T$, and a feasible P2P schedule, $\{(x_1[t], ..., x_M[t])\}_{t=1}^{T_1}$, such that the following conditions hold with probability at least η:*

$$(C1)\ \sum_i e_i[T_1] \geq N$$
$$(C2)\ \sum_i e_i[T_1] - (M-1)I[T_1] \geq MN - (M-1)T$$
$$(C3)\ e_i[T_1] - x_i[T_1] - I[T_1] \geq N - T \quad \forall i.$$

Proof. The whole system needs to receive at least N chunks from the B2P network in Phase 1 in order to be able to disseminate all N original chunks to all peers. Therefore, condition $(C1)$ must hold.

For a work conserving scheme, we have $T_1 = \sum_i x_i[T_1] + I[T_1]$. Further, from Theorem 16 we have

$$T_2 - T_1 \geq N - \sum_i x_i[T_1] - \min\left(\frac{\sum_i e_i[T_1] - N}{M-1}, \min_i(e_i[T_1] - x_i[T_1])\right).$$

Therefore, $T_2 \geq N + I[T_1] - \min\left(\frac{\sum_i e_i[T_1]-N}{M-1}, \min_i(e_i[T_1] - x_i[T_1])\right)$. For a successful delivery of the block we need $T_2 \leq T$, which implies the constraints $(C2)$ and $(C3)$. Hence, in order to meet the QoS (η, T), all three conditions in Theorem 18 must hold with a probability at least equal to η. □

From the previous theorem, the optimal stopping time is the smallest value for T_1, for which we have $\mathbb{P}((C1), (C2), (C3)) \geq \eta$. The next theorem declares that *MRF* is an optimal scheme for Phase 1.

Theorem 19. *For any T, T_1 and η, the* MRF *algorithm maximizes $\mathbb{P}((C1), (C2), (C3))$ over all work conserving schemes.*

Proof. To prove this theorem, we model our system as a system of M queues and a single server, in which $e_i[t]$ and $x_i[t]$ are respectively the accumulated arrival and service to some queue i, at time t. The arrival rate is p for all queues, and the server can serve one queue in each slot. Note that $(C1)$ and $(C2)$ are independent of the P2P scheme, and $(C3)$ implies some minimum length constraint on the shortest queue in this model. We show that the *Longest-Queue-First (LQF) algorithm* (equivalent to *MRF* in the P2P network), due to its equalizing effect, results in a longer minimum queue length compared to all other work conserving policies. The optimality of *MRF* follows from this result. For details, see Appendix B. $\square$

Now, it remains to find the minimum T_1 for which $\mathbb{P}((C1),(C2),(C3)) \geq \eta$, when the *MRF* scheme is used. To this end, we define a Markov chain $\mathcal{M}$ whose state at time t is $(I[t], Z[t]) = (I[t], z_i[t] : 1 \leq i \leq M)$, where $Z[t]$ is a vector of $z_i[t] = e_i[t] - x_i[t]$ elements. For this Markov chain, Theorem 18 determines a set of desirable states, $C(N, M, T, t)$, to be hit at time $t = T_1$. Lemma 14 provides this set and the transition probabilities of $\mathcal{M}$, which can be used to find the minimum T_1 (Algorithm 5).

Lemma 14. *For the Markov chain $\mathcal{M}$, the one step transition probability is as follows,*

$$\mathbb{P}_{(\mathbf{I},\mathbf{Z})}((I + 1_{\{\sum_i z_i=0\}}, \{z_i - 1_{\{i=i^*\}}\}^+ + m_i : 1 \leq i \leq M)|(I, z_i : 1 \leq i \leq M))$$
$$= \begin{cases} p^{\sum_{i=1}^M m_i}(1-p)^{M-\sum\limits_{i=1}^{M} m_i} & \text{if } \forall i : m_i \in \{0,1\} \\ 0 & \text{otherwise} \end{cases}$$

where $i^* = \min\arg\max_i z_i$ *and* $\{a\}^+ = \max(a, 0)$. *Also the set of desirable states at* *time t,*

$$C(N,M,T,t) = \{(I,Z) | \qquad \min_i z_i - I \geq N - T, \\ \sum_i z_i + (t-I) \geq N + (M-1)\max(0, N-T+I)\}.$$

Proof. Note that for all i, $z_i[t+1] = z_i[t] - (x_i[t+1] - x_i[t]) + m_i$, where $m_i \in \{0,1\}$ and $\mathbb{P}(m_i = 1) = p$. Without loss of generality, we assume that at time t *MRF* algorithm chooses $i^* = \min \arg\max_i n_i[t] = \min \arg\max_i z_i[t]$ (from (4.2)) to make a P2P transmission. Therefore for all $i \neq i^*$, $x_i[t+1] = x_i[t]$. From (4.12), if $\sum_i z_i[t] = 0$ then $I[t+1] = I[t] + 1$. Otherwise, $z_{i^*}[t] > 0$ and $x_{i^*}[t+1] - x_{i^*}[t] = 1$. Therefore, the state (I,Z) changes with the transition probability presented in the lemma.

Also it can be seen that $\sum_i e_i[t] = \sum_i z_i[t] + t - I[t]$, because $\sum_i x_i[t] + I[t] = t$ for all work conserving policies. Hence, it is straightforward to see that the conditions in Theorem 18 imply the set of desirable states $C(N,M,T,t)$ at time $t = T_1$. □

In Algorithm 5, we present an algorithm to achieve the QoS target and summarize its performance as follows.

Theorem 20. *Algorithm 5 achieves (η, T) with the minimum cost, when the field size is infinite.*

b. Finite Field Size

In this subsection, we take into account the effect of the finiteness of the field size.

In Theorem 20 we showed that the suggested scheme provides each peer with at least N coded chunks by the deadline T, with a probability at least equal to η. Since the field size was assumed to be unboundedly large, N coded chunks were enough to decode the original ones. For the finite field case, the received chunks by a peer are not linearly independent with a non-zero probability. In this case, the corresponding peer

Algorithm 5 Optimal B2P and P2P schemes (infinite field)

Given M, N, p, T **and** η**:**

1) Find T_1^* **such that:**

$T_1^* = \min\{0 \leq t \leq T : \mathbb{P}\left((I[t], Z[t]) \in C(N, M, T, t) \mid (I[0], Z[0]) = \mathbf{0}\right) \geq \eta\}$.

2) For $0 \leq t < T_1^*$**:** turn all B2P channels on and use the *MRF* algorithm to schedule the P2P network.

3) At $t = T_1^*$**:** turn all B2P channels off.

4) For $T_1^* \leq t \leq T$**:** use the *NmRF* algorithm to schedule the P2P network.

fails to decode the original chunks. The following theorem declares the probability of such an event.

Theorem 21. *For the field size q, if the proposed scheme in Algorithm 5 is applied,*

$$\mathbb{P}\left(\exists i : n_i[T] < N\right) \leq 1 - \eta + \frac{M}{q-1}. \tag{4.13}$$

Proof. We have,

$$\begin{aligned}\mathbb{P}\left(\exists i : n_i[T] < N\right) = \mathbb{P}\left(\exists i : n_i[T] < N \mid \exists i : \hat{n}_i[T] < N\right) \mathbb{P}\left(\exists i : \hat{n}_i[T] < N\right) + \\ \mathbb{P}\left(\exists i : n_i[T] < N \mid \forall i : \hat{n}_i[T] \geq N\right) \mathbb{P}\left(\forall i : \hat{n}_i[T] \geq N\right)\end{aligned}$$

Note that the first probability in the RHS of the above equation is 1. Also it was shown in Theorem 20

$$\mathbb{P}\left(\exists i : \hat{n}_i[T] < N\right) \leq 1 - \eta. \tag{4.14}$$

Therefore, we get

$$\mathbb{P}\left(\exists i : n_i[T] < N\right) \leq \; 1 - \eta + \mathbb{P}\left(\exists i : n_i[T] < N \mid \forall i : \hat{n}_i[T] \geq N\right) \leq \; 1 - \eta + \tfrac{M}{q-1}, \tag{4.15}$$

where the last inequality follows from Theorem 17. $\square$

We can see that the probability of failure has two components. One stems from the unreliability of the B2P channels, upper bounded by some δ_1 (here $\delta_1 = 1-\eta$), and the other from the finiteness of the field size, upper bounded by δ_2 (here $\delta_2 = \frac{M}{q-1}$). In order to meet the QoS constraint, it suffices to have $\delta_1 + \delta_2 \leq 1 - \eta$. The following lemma shows that for decreasing δ_2, each peer requires to receive more coded chunks, which means that B2P usage and the associated cost increase.

Lemma 15. *The bound* $\delta_2 \in (0, \frac{M}{q-1})$ *is achievable, if for all* i *we have* $\hat{n}_i(T) \geq N + \Delta(\delta_2)$, *where* $\Delta(\delta_2) = \left\lceil \log_q(\frac{M}{q-1}) - \log_q(\delta_2) \right\rceil$ *is the number of additional coded chunks each peer needs on average, in order to reduce the probability of winding up with a non-full rank matrix to* δ_2.

Proof. The proof follows directly from Lemma 13. □

4. Distributed Implementation of Peer Selection Algorithms

In the previous sections, we presented two simple peer selection algorithms—*MRF* and *NmRF*. These algorithms indicate which peer is the appropriate one to broadcast at each time. Although we saw that *MRF* and *NmRF* perform well (indeed they are optimal for large field sizes), coordination of the P2P broadcasts requires a central entity which has complete knowledge about the rank of all peers. The overhead of having a central scheduler in a wireless network can enormously degrade the performance. In this section, we present some ideas to decentralize the peer selection schemes with a small overhead.

Recall our model, in which we assumed the peers are close to each other and the P2P broadcasts are always successful. For such a model, it is reasonable to further assume that the delay of sensing a broadcast transmission is negligible. That is, when a peer broadcasts over the P2P network, every other peer will be aware of it almost instantaneously, as in the model considered in [46].

Now, we divide each time slot to two parts; a control slot (with length T_c) and a data slot (with length T_d). The purpose of a control slot is to find an appropriate peer to broadcast a message during the following data slot. We assume that each peer includes its current rank in the header of the data message that it transmits. Hence, after a P2P broadcast, all the peers get to know the rank of the transmitter. We use this rank as a reference in order to find a maximum-rank (or non-minimum-rank) peer in the *MRF* (or *NmRF*) algorithm for the next time slot. The generic form of our distributed peer selection algorithm is as follows:

Distributed peer selection algorithm: At the beginning of time slot t, each peer i decides whether to contend for a transmission during the data slot based on its current rank $n_i[t]$ and the rank of the previous transmitter $u[t-1]$ at the previous slot $n_{u[t-1]}[t-1]$. If i decides to contend, it will wait for some random back-off time during the control slot. The decision rule for MRF and NmRF, and the method of choosing the back-off times will be described later in this section. If during the waiting interval, peer i hears a transmission from another peer, it does not contend to transmit. Otherwise, when peer i finishes waiting for its determined back-off time, it will broadcast a short control message informing its intension to transmit over the data slot. If no peer attempts to transmit during the control slot, the peer that transmitted in the previous time slot, peer $u[t-1]$, will transmit in slot t as well.

Clearly, due to the assumption of zero sensing delay, no collision will happen in either the control slot or the data slot. Also, the overhead of the control slot can be significantly small, because in general a control message is much shorter than a data message. In what follows, we show how peers choose the back-off time in case of *MRF* and *NmRF* algorithms.

First, we note that from (4.2), we have

$$\hat{n}_i[t] = \hat{n}_i[t-1] + (e_i[t] - e_i[t-1]) + \left(\sum_{j \neq i} x_j[t] - \sum_{j \neq i} x_j[t-1] \right), \tag{4.16}$$

where $(e_i[t] - e_i[t-1])$, $(\sum_{j \neq i} x_j[t] - \sum_{j \neq i} x_j[t-1]) \in \{0, 1\}$ determine if peer i received a transmission during slot $t-1$ respectively from the B2P and the P2P networks. Thus, the rank of any peer can increase in a slot by at most two.

MRF Algorithm

Following *MRF* algorithm, for each peer i we have $n_i[t-1] \leq n_{u[t-1]}[t-1]$ and from (4.16), $\hat{n}_i[t] \leq \hat{n}_i[t-1] + 2$ and consequently $n_i[t] \leq n_i[t-1] + 2$ hold. As a result, $n_i[t] \leq n_{u[t-1]}[t-1] + 2$ for all $i \in \{1, ..., M\}$. In other words, any peer that contends for transmission in slot will have a rank either one or two greater than the peer that transmitted in the previous slot.

Back-off time in distributed MRF *algorithm:* At the beginning of time slot t, peer i chooses to contend for a transmission if $n_i[t] \geq n_{u[t-1]}[t-1] + 1$. Now, there are only two possibilities, and if $n_i[t] = n_{u[t-1]}[t-1] + 2$, peer i waits for a random time uniformly distributed in $[0, T_c/2]$, else if $n_i[t] = n_{u[t-1]}[t-1] + 1$ it chooses a uniformly random time in $[T_c/2, T_c]$.

Note that the proposed distributed *MRF* algorithm can also be applied to decentralize the *NmRF* algorithm. However, we will present a slightly different scheme which is more suited for the *NmRF* scheme in pure P2P systems (especially, in Phase 2 of Algorithm 5).

NmRF Algorithm

In a pure P2P system (no B2P transmissions), from (4.16), we have $\hat{n}_i[t] = \hat{n}_i[t-1] + (\sum_{j \neq i} x_j[t] - \sum_{j \neq i} x_j[t-1])$, and $n_{u[t-1]}[t] = n_{u[t-1]}[t-1]$.

Back-off time in distributed NmRF *algorithm:* At the beginning of time slot t, peer i chooses to contend for a transmission only if $n_i[t] \geq n_{u[t-1]}[t-1]+1$. In this case, it will wait for a random amount of time uniformly distributed in $[0, T_c]$ in the control slot.

It is straightforward to check that in the proposed distributed schemes, an appropriate peer (maximum-rank for *MRF* and non-minimum-rank for *NmRF*) succeeds in transmitting during the data slot.

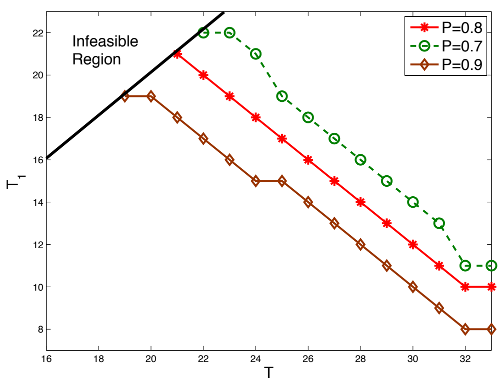
5. Simulation Results

We now evaluate the performance of Algorithm 5 for a simple hybrid network with $M = 4$ users and different values of N.

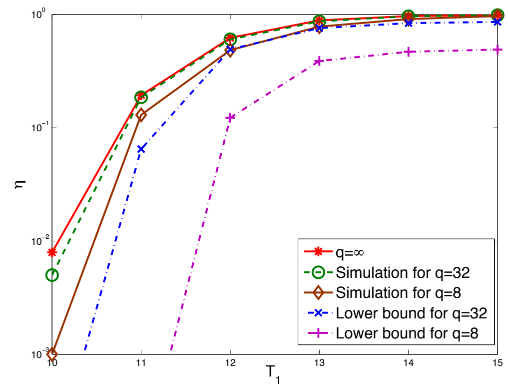
Figure 10(a) illustrates the tradeoff between the minimum cost, T_1, and the deadline T. By increasing the B2P channel capacity p, lower costs are achievable for the same T. Also as T increases, peers have more opportunities to disseminate chunks in Phase 2. Hence, we can achieve the QoS target with lower costs. However, T_1 should be large enough to satisfy conditions $(C1)-(C3)$, which is why it does not decrease beyond $T = 32$. The infeasible region indicates those values for T which are not achievable (Theorem 18).

The effect of finite field sizes is evaluated in Figure 10(b). Clearly, increasing the field size results in a better performance.

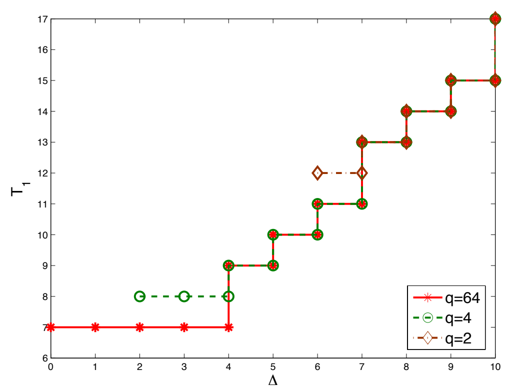
Figure 10(c) displays the minimum cost to achieve the QoS versus Δ, which is the minimum number of extra chunks each peer i would receive eventually. These extra chunks can compensate the failure probability due to finiteness of q. However, a larger Δ incurs more cost of the B2P usage. Note that as the field size decreases, a larger Δ is required. For example for $q = 4$, we need at least $\Delta_{\min} = 2$ extra chunks in order to meet the QoS.



(a) Minimum cost vs. deadline trade-off ($N = 30$, $\eta = 0.9$, $q = \infty$)



(b) Probability of success vs. minimum cost ($N = 20$, $T = 15, q = 8, 32$)



(c) The effect of extra delivered chunks on achievability ($N = 20$, $T = 25$, $\eta = 0.9$)

Fig. 10. The performance of Algorithm 5

B. Live Multimedia Streaming

In this section, we study a new model for streaming applications and propose simple optimal algorithms to achieve the required QoS. The structure of this section is as follows.

In Subsection 1, we adopt a new model, suitable for live streaming applications, and modify the QoS metric. Unlike the content distribution model, here we have a continuous stream of multimedia content that can be thought of as a long sequence of blocks. Each block has a fixed deadline to be delivered from the time it is generated. As before, we let peers utilize both of their B2P and P2P interfaces simultaneously. However, we will show how we can separate the decisions on the B2P and the P2P networks in this new model. According to the new QoS metric, each peer individually requires a minimum average delivery ratio in order to guarantee smooth playout.

As mentioned above, we can separately study the P2P part of the system. Hence, we will focus on the P2P network in Subsection 2, and find a set of necessary and sufficient conditions for achievability of a given QoS metric. Then we use ideas from queueing theory and Foster-Lyapunov stability criterion to find an optimal P2P scheme for sufficiently large field sizes. We shall see that the proposed P2P scheme has a simple and intuitive form.

In Subsection 3, the problem of coordinating the B2P network will be investigated. Based on the necessary and sufficient conditions found in Subsection 2, we propose a general framework to find the optimal B2P usage times for any given cost criterion.

We specifically study a symmetric system, in which all peers have similar QoS requirements, in Subsection 4. We discover new properties of this system, and it will be shown how the results found in the previous subsections can be simplified for this special case.

In all the above subsections, we assume the coding is performed in the fields of sufficiently large size to ignore the degradation effect of receiving useless packets. We turn our attention to the finite fields in Subsection 5. We will provide some bounds on the performance of the proposed B2P-P2P algorithm in this case. It will be seen that this degradation effect decreases inversely proportional to the field size q and is independent of other parameters like the number of peers or the number of chunks in a block.

In Subsection 6, we extend our results to a number of interesting situations which can occur in practice. For example, in order to improve the QoS in the system we can add some peers (called *boosters*) whose role is to merely help the other peers. The effect of having boosters is evaluated in this subsection. Also, an important issue that arises in multi-user systems is the fairness problem. We will also show how we can incorporate fairness into our model. Many practical media streaming systems use some erasure protection coding schemes in order to take care of the failures and hence improve the quality. We will briefly point out to some examples of these techniques and suggest a new QoS model suitable for such applications. An optimal P2P scheme will also be proposed for achieving the new QoS metric.

Some simulation results will be presented in Subsection 7, which will show the capabilities of our framework and the optimality of the proposed algorithms.

1. System Model

Consider Figure 8 again. There are M peers, denoted by $i \in \{1, \ldots, M\}$, all interested in receiving the same stream of data (*e.g.* a video file). The data source generates the stream in the form of a sequence of blocks. Each block is further divided into N chunks. We consider a slotted-time system, where at most one chunk can be transmitted over either of B2P or P2P interfaces during a time slot.

These blocks need to be played out at the peer devices in sequence and within some specific deadlines. Suppose the playing time of a block consists of T slots, we call this time scale a *frame*. We require all peers to synchronously play out the k^{th} block during the k^{th} frame, for $k \geq 1$. However a peer i is able to do so, only if he has received all N chunks of the k^{th} block by the beginning of frame k. Otherwise, i will be idle during this frame. Let $I_i[k] \in \{0, 1\}$ denote the idleness of peer i during the k^{th} frame.

We are interested in *live* streaming applications as opposed to *on-demand* or *stored content* streaming. That is the k^{th} block is available at the servers to disseminate to peers only within some short fixed time before the k^{th} frame. We assume that the k^{th} block is available at base stations not before frame $k - 2$.

Each peer i has a QoS metric $\eta_i \in (0, 1]$, which requires him to successfully play out at least η_i fraction of blocks. η_i is called the delivery ratio of peer i and is the minimum acceptable long-run average number of frames peer i is busy playing out,

$$\eta_i \leq 1 - \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[I_i[k]]. \tag{4.17}$$

Similar to our model in Section A, we assume two communication interfaces (free, reliable broadcast P2P and costly, unreliable unicast B2P) for each peer. We model each B2P unicast channel as an independent Bernoulli random variable with parameter p. By $u[t] \in \{1, \ldots, M\}$, we denote the peer who broadcasts over P2P channel in time slot t.

Our objective is to find a scheme that would satisfy the delivery ratio requirement of all peers at the lowest cost of using B2P transmissions. Figure 11 illustrates our timing structure. Note that peers receive chunks of block k from the B2P network during frame $k - 2$. These chunks may be disseminated over the broadcast P2P network during frame $k-1$. Those peers who could successfully receive enough number of chunks of block k, before the beginning of frame k, will be able to recover and play out this block during the k^{th} frame.

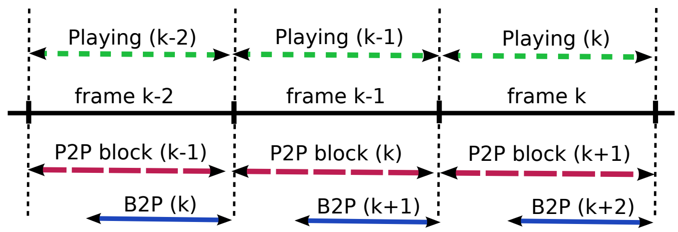


Fig. 11. Sequence of transmissions over B2P and P2P networks and the playout time of each block

We would like to emphasize the differences between the content distribution application, studies in the previous section, with the current live streaming problem. In the former problem, our objective was to disseminate a *single* common block of information to a group of peers. We considered a *system-wide* QoS constraint which requires the whole group of peers to receive the common block, with some target probability, within some fixed deadline. In our proposed scheme, we had simultaneous transmissions of the block over two interfaces.

In the current live streaming application, our goal is to enable multicasting a live multimedia channel to a group of peers. Hence unlike the previous problem, we deal with a much longer time horizon which consists of many subsequent frames. Also the QoS is defined for each peer individually based on the notion of delivery ratio, which is in essence a long-run average parameter. In the proposed framework, the transmission of block k is decomposed to two separate phases; B2P transmissions during frame $k-2$ and P2P broadcasts during frame $k-1$. We realized if we do not decompose the P2P and B2P transmissions, then we will need an excessive amount of feedbacks and computations to efficiently coordinate the P2P transmissions. However, we shall see that the optimal scheme for the proposed timing structure is quite simple and easily implementable in a distributed fashion without any need for exclusive feedback from peers.

For now let us focus on a single block (say block k). We denote the total number of chunks of block k delivered to peer i via the B2P network by the beginning of slot t using $e_i^{(k)}[t]$. Also $x_i^{(k)}[t]$ and $r_i^{(k)}[t]$ are used to denote respectively the total number of transmitted and received chunks of block k by peer i via P2P by the beginning of slot t. Note that according to the model

$$\sum_{i,k} x_i^{(k)}[t] - \sum_{i,k} x_i^{(k)}[t-1] \leq 1 \quad \text{for all } t > 0 \tag{4.18}$$

that is only one peer can broadcast over the P2P network at each slot, and because of the lossless nature of the broadcast P2P network, we have $\sum_{j \neq i} x_j^{(k)}[t] = r_i^{(k)}[t]$. Also our timing structure, displayed in Figure 11, implies the following conditions,

$$\begin{aligned}
&e_i^{(k)}[t] = 0 && \text{for all } t \leq (k-2)T,\ k \geq 2 \\
&e_i^{(k)}[t] = e_i^{(k)}[(k-1)T] && \text{for all } t \geq (k-1)T \\
&r_i^{(k)}[t] = 0 && \text{for all } t \leq (k-1)T \\
&r_i^{(k)}[t] = r_i^{(k)}[kT] && \text{for all } t \geq kT.
\end{aligned} \tag{4.19}$$

For $k = 1$, we assume that at time 0 peers are initialized with some buffered chunks from the B2P network. Like before, we use random linear network coding over the chunks of each block in order to avoid the problem of piece selection. Each coded chunk is now a random linear combination (with coefficients in finite field F_q) of the N original chunks in the corresponding block.

Let $S_i^{(k)}[t]$ be the set of vectors corresponding to the coded chunks of block k

possessed by peer i at time t, with cardinality $\hat{n}_i^{(k)}[t]$. Thus,

$$\hat{n}_i^{(k)}[t] = e_i^{(k)}[t] + r_i^{(k)}[t] = e_i^{(k)}[t] - x_i^{(k)}[t] + \textstyle\sum_j x_j^{(k)}[t]. \tag{4.20}$$

We denote by $n_i^{(k)}[t]$ (called the k^{th} *rank* of peer i at time t) the dimension of the subspace spanned by $S_i^{(k)}[t]$.

As seen in the previous section, the reliability of P2P transmissions will result in two important points. First, we can assume without loss of generality that at time t, $u[t]$ transmits a chunk which is a linear combination of the chunks received from the B2P network (during the previous frame). This implies that

$$x_i^{(k)}[t] \leq e_i^{(k)}[t] \quad \text{for all } k \text{ and } t \geq 0. \tag{4.21}$$

Second, given a sequence of P2P transmissions, any changes in the order of transmissions will not affect the performance.

Note that a feasible P2P scheme $\{(x_1^{(k)}[t], ..., x_M^{(k)}[t])\}_{t=(k-1)T}^{kT}$ must satisfy (4.18) and (4.21). Also, in order to decode the original N chunks of block k at the beginning of frame k, peer i must have $n_i^{(k)}[kT] = N$. One can immediately verify that $I_i[k] = 1_{\{n_i^{(k)}[kT]<N\}}$.

Consider Figure 11 again. For each block k, peers have T time slots (duration of frame $k-1$) to further exchange the chunks, received from the B2P network, over the broadcast P2P network. Hence, the B2P network should deliver enough coded chunks to peers during frame $k-2$ such that they can meet their QoS constraints using the T P2P broadcasts in frame $k-1$.

In what follows, we assume that the field size q is large enough that we can ignore the effect of its finiteness on the performance of the linear coding (i.e., $n_i^{(k)}[t] = \min\{N, \hat{n}_i^{(k)}[t]\}$). We are also interested in the case where $N > T$, because otherwise there are enough number of time slots in each frame for peers to broadcast all N

degrees of freedom and hence the optimal P2P scheme becomes trivial. We will consider the finite field case in Section 5.

2. Coordination of the P2P Broadcast Network

In this subsection, we seek an algorithm to pick the right peer to broadcast over the P2P network at each time. Since we decomposed the B2P and P2P transmissions, we can study a pure P2P broadcast network for each block k. The effect of the B2P transmissions can be modeled as a set of stochastic arrivals of coded chunks to the peers at the beginning of each frame. That is for each block k, peer i receives a number $e_i^{(k)}$ of randomly coded chunks at the beginning of frame $k-1$. These chunks will be used during frame $k-1$ over the P2P network such that peers can satisfy their QoS metrics. As seen in Section A, it suffices to transmit initial chunks, received from B2P network, without any further coding or combining.

Note that the $e_i^{(k)}$ values are random due to unreliability of B2P channels. Following our result in the previous section, if we fix the number of slots each peer attempts to receive a chunk from the B2P network in each frame, then $e_i^{(k)}$ values will be independent over peers, and for each peer i, it is independently and identically distributed over blocks k. We denote the arrival process to peer i using $\mathbf{e}_i$.

Our objective is to find a P2P scheme which can satisfy the QoS requirements $(\eta_1, ..., \eta_M)$ of the peers for a given set of arrival processes $(\mathbf{e}_1, ..., \mathbf{e}_M)$. First, we need to determine whether the QoS metric $(\eta_1, ..., \eta_M)$ is achievable for the given arrivals $(\mathbf{e}_1, ..., \mathbf{e}_M)$ or not.

a. Achievability of QoS Metric

Definition 7. *We say the QoS $(\eta_1, ..., \eta_M)$ is achievable, if there exists a feasible policy to coordinate the P2P transmissions such that, on average each peer i successfully* *receives η_i fraction of the blocks before their deadlines.*

In what follows, we will present some necessary and sufficient conditions for QoS metrics to be achievable.

We denoted the number of arrived B2P chunks of block k to peer i by $e_i^{(k)}$ and further assumed that $e_i^{(k)}$ values, for $k \geq 1$, are independently and identically distributed as a random variable $\mathbf{e}_i$.

Peers have T slots in each frame to exchange the received B2P chunks. Hence, each peer i can potentially recover block k, only if (i) he has received enough B2P chunks initially (i.e, $e_i^{(k)} \geq N - T$) and (ii) the whole system is full-rank at the beginning of the frame (i.e, $\sum_j e_j^{(k)} \geq N$). We let the idleness of peer i at frame k be one (i.e., $I_i[k] = 1$), when i is not able to recover block k. Therefore for each peer i we have,

$$1 - I_i[k] \leq 1_{\{e_i^{(k)} \geq N-T,\ \sum_j e_j^{(k)} \geq N\}} \tag{4.22}$$

From (4.17) and since $(e_1^{(k)}, ..., e_M^{(k)})$ is assumed to be identically and independently distributed over frame k with $(\mathbf{e}_1, ..., \mathbf{e}_M)$, we get the following necessary condition on achievability of η_i

$$\eta_i \leq \mathbb{P}\left(\mathbf{e}_i \geq N - T,\ \sum_j \mathbf{e}_j \geq N\right). \tag{4.23}$$

Note that even if $\sum_j e_j^{(k)} \geq N$ holds, we may not be able to deliver the whole block k to all peers i who satisfy $e_i^{(k)} \geq N - T$. Let $N_s(e_1^{(k)}, ..., e_M^{(k)}) = \sum_i (1 - I_i[k])$ be the number of peers who can successfully receive the whole block k, given the B2P arrivals $(e_1^{(k)}, ..., e_M^{(k)})$. The following lemma provides an upperbound on $N_s(e_1^{(k)}, ..., e_M^{(k)})$.

Lemma 16. *Given the B2P arrivals* $(e_1^{(k)}, ..., e_M^{(k)})$*, we have*

$$N_s(e_1^{(k)}, ..., e_M^{(k)}) \leq N_s^*(e_1^{(k)}, ..., e_M^{(k)}) = \frac{\min\left(|\mathcal{S}|(N-T), \left[\sum_i e_i^{(k)} - T\right]^+\right)}{N-T}, \quad (4.24)$$

where $\mathcal{S} = \{i \in \{1, ..., M\} : \ N - e_i^{(k)} \leq T, \sum_j e_j^{(k)} \geq N\}$ *and* $[a]^+ = \max\{a, 0\}$.

Proof. We have $1 - I_i[k] = 1_{\{n_i^{(k)}[kT] \geq N\}} \leq 1_{\{\hat{n}_i^{(k)}[kT] \geq N\}} = 1_{\{e_i^{(k)} - x_i^{(k)}[kT] + \sum_j x_j^{(k)}[kT] \geq N\}}$. Therefore, we can solve the following maximization problem in order to find an upperbound on $N_s(e_1^{(k)}, ..., e_M^{(k)})$,

$$\begin{aligned} &\max \ \textstyle\sum_i 1_{\{e_i^{(k)} - x_i^{(k)}[kT] + \sum_j x_j^{(k)}[kT] \geq N\}} \\ &\qquad \text{subject to} \\ &x_i^{(k)}[kT] \leq e_i^{(k)} \qquad \text{for all } i \\ &\textstyle\sum_j x_j^{(k)}[kT] \leq T \end{aligned} \quad (4.25)$$

The first constraint implies $\sum_j x_j^{(k)}[kT] \leq \sum_j e_j^{(k)}$, hence to achieve the maximum objective we can let $\sum_j x_j^{(k)}[kT] = \min(\sum_j e_j^{(k)}, \ T)$.

Lets partition the set of peers $\{1, ..., M\}$ into sets $\mathcal{S}$ and $\mathcal{S}^c = \{1, ..., M\} \backslash \mathcal{S}$. Note that $\mathcal{S}^c$ is the set of peers who, either individually or collectively, have not received enough number of B2P arrivals and no feasible P2P scheme can help them to recover the block before its deadline (i.e., $I_i[k] = 1$ for $i \in \mathcal{S}^c$). Hence, $N_s(e_1^{(k)}, ..., e_M^{(k)}) \leq |\mathcal{S}|$.

Suppose $\sum_j e_j^{(k)} < N$, then we have $N_s(e_1^{(k)}, ..., e_M^{(k)}) = |\mathcal{S}| = 0$. Otherwise $\sum_j e_j^{(k)} \geq N$ and with our assumption of $T < N$, the optimization in (4.25) can be modified to

$$\begin{aligned} &\max \ \textstyle\sum_{i \in \mathcal{S}} 1_{\{e_i^{(k)} - x_i^{(k)}[kT] \geq N - T\}} \\ &\qquad \text{subject to} \\ &x_i^{(k)}[kT] \leq e_i^{(k)} \qquad \text{for all } i \\ &\textstyle\sum_j x_j^{(k)}[kT] = \min(\sum_j e_j^{(k)}, \ T) = T \end{aligned} \quad (4.26)$$

Note that we can increase $x_i^{(k)}[kT]$ upto $e_i^{(k)}$ for $i \in \mathcal{S}^c$ without affecting the objective value. However, we need to make sure that the second constraint is satisfied. Considering this observation will result in the following modification of our maximization problem

$$\begin{array}{ll} \max \ \sum_{i\in\mathcal{S}} 1_{\{e_i^{(k)}-x_i^{(k)}[kT]\geq N-T\}} & \\ \text{subject to} & \\ x_i^{(k)}[kT] \leq e_i^{(k)} & \text{for all } i \in \mathcal{S} \\ \sum_{j\in\mathcal{S}} x_j^{(k)}[kT] = [T - \sum_{j\in\mathcal{S}^c} e_j^{(k)}]^+ & \end{array} \tag{4.27}$$

where the optimal value will be $|\mathcal{S}| - \left\lceil \frac{\left[[T-\sum_{j\in\mathcal{S}^c} e_j^{(k)}]^+ - \sum_{i\in\mathcal{S}} e_i^{(k)} + |\mathcal{S}|(N-T)\right]^+}{N-T} \right\rceil$. Note that $\sum_{i\in\mathcal{S}} e_i^{(k)} \geq |\mathcal{S}|(N-T)$ and $[[X]^+ + Y]^+ = [X+Y]^+$ for any X and any $Y \leq 0$. Therefore,

$$\begin{array}{l} |\mathcal{S}| - \left\lceil \frac{\left[[T-\sum_{j\in\mathcal{S}^c} e_j^{(k)}]^+ - \sum_{i\in\mathcal{S}} e_i^{(k)} + |\mathcal{S}|(N-T)\right]^+}{N-T} \right\rceil = |\mathcal{S}| - \left\lceil \frac{\left[[T-\sum_i e_i^{(k)} + |\mathcal{S}|(N-T)\right]^+}{N-T} \right\rceil \\ = \left\lfloor \frac{\min\left(|\mathcal{S}|(N-T), \sum_i e_i^{(k)} - T\right)}{N-T} \right\rfloor. \end{array} \tag{4.28}$$

Note that the above optimal value is subject to the condition $\sum_j e_j^{(k)} \geq N$. However, it is straightforward to check that we can get a general form for the optimal value of (4.25) by a slight modification as follows,

$$\left\lfloor \frac{\min\left(|\mathcal{S}|(N-T), \left[\sum_i e_i^{(k)} - T\right]^+\right)}{N-T} \right\rfloor.$$

Consequently, we get

$$N_s(e_1^{(k)}, ..., e_M^{(k)}) \leq \frac{\min\left(|\mathcal{S}|(N-T), \left[\sum_i e_i^{(k)} - T\right]^+\right)}{N-T}.$$

□

Once again from (4.17) and since $(e_1^{(k)}, ..., e_M^{(k)})$ is *i.i.d* over frame k, the following necessary condition on $\sum_i \eta_i$ will be resulted

$$\sum_i \eta_i \le \mathbb{E}\left[\lfloor N_s^*(\mathbf{e}_1, ..., \mathbf{e}_M)\rfloor\right]. \tag{4.29}$$

The following theorem summarizes our result,

Theorem 22. *The QoS metric* $(\eta_1, ..., \eta_M)$ *is achievable with respect to* i.i.d *B2P arrivals* $(\mathbf{e}_1, ..., \mathbf{e}_M)$ *if and only if the following conditions are satisfied*

$$\begin{aligned} &(C1) \quad \eta_i \le \mathbb{P}\left(\mathbf{e}_i \ge N - T,\ \textstyle\sum_j \mathbf{e}_j \ge N\right) \qquad \textit{for all } i \\ &(C2) \quad \textstyle\sum_i \eta_i \le \mathbb{E}\left[\lfloor N_s^*(\mathbf{e}_1, ..., \mathbf{e}_M)\rfloor\right]. \end{aligned} \tag{4.30}$$

Proof. The necessity part was shown in (4.23) and (4.29). To prove the sufficiency of these conditions, we will propose an algorithm in the next subsection which can fulfill any QoS constraints satisfying $(C1)$ and $(C2)$. □

Corollary 6. *For the symmetric case, when* $\eta_i = \eta$ *and* $\mathbf{e}_i$ *are identically distributed for all peers* i*, the condition* $(C1)$ *is dominated by* $(C2)$ *and the necessary and sufficient condition on the achievability of the QoS metric* $(\eta, ..., \eta)$ *reduces to*

$$(C2') \quad \eta \le \tfrac{1}{M}\mathbb{E}\left[\lfloor N_s^*(\mathbf{e}_1, ..., \mathbf{e}_M)\rfloor\right]. \tag{4.31}$$

Proof. From (4.24) and $(C2)$ in (4.30), we have

$$\begin{aligned} M\eta &\le \mathbb{E}\left[\left\lfloor \frac{\min\left(|\mathcal{S}|(N-T), \left[\sum_i e_i^{(k)} - T\right]^+\right)}{N-T} \right\rfloor\right] \le \mathbb{E}\left[\frac{\min\left(|\mathcal{S}|(N-T), \left[\sum_i e_i^{(k)} - T\right]^+\right)}{N-T}\right] \\ &\overset{(a)}{\le} \tfrac{1}{N-T} \min\left(\mathbb{E}\left[|\mathcal{S}|(N-T)\right], \mathbb{E}\left[\left[\textstyle\sum_i e_i^{(k)} - T\right]^+\right]\right) \\ &\le \mathbb{E}\left[|\mathcal{S}|\right] = \mathbb{E}\left[\textstyle\sum_i 1_{\{\mathbf{e}_i \ge N-T, \sum_j \mathbf{e}_j \ge N\}}\right] = M\mathbb{P}\left(\mathbf{e}_i \ge N - T,\ \textstyle\sum_j \mathbf{e}_j \ge N\right), \end{aligned} \tag{4.32}$$

where (a) follows since $\mathbb{E}[\min(X, Y)] \le \min(\mathbb{E}[X], \mathbb{E}[Y])$. Hence, condition $(C2)$ implies condition $(C1)$ for the symmetric case. □

In the next subsection, we will use some ideas from queueing theory in order to find a simple algorithm which is able to meet any achievable QoS metric.

b. Optimal P2P Scheme

In this subsection, we propose a simple algorithm which can fulfill any QoS metric satisfying the conditions in (4.30). As a result, $(C1)$ and $(C2)$ are sufficient conditions on achievability of a QoS metric (Theorem 22). Also the proposed algorithm is throughput optimal in the sense that it can satisfy any achievable QoS metric.

In order to keep track of peers' quality of experience, we define a virtual deficit queue for each peer. The length of this queue $d_i[k]$, for peer i at frame k, follows the dynamic below

$$d_i[k] = d_i[k-1] + \eta_i - (1 - I_i[k]).$$

Recall that $I_i[k] = 1$ means peer i is not successful in receiving the block k before its deadline and hence is idle in frame k. In this case the deficit value of this peer increases by an amount equal to its delivery ration η_i. Otherwise, $I_i[k] = 0$ and the deficit value decreases by $1 - \eta_i$. Therefore, the deficit queue length captures the accumulated unhappiness of a peer about the QoS experienced so far,

$$d_i[k] = k\eta_i + \sum_{l=1}^{k} I_i[l] - k.$$

The evolution of these deficit queues can be studied by a Markov chain $\mathcal{D}$ whose state at each step k is $([d_1[k]]^+, ..., [d_M[k]]^+)$. Our objective is to find a P2P scheme which leads to this Markov chain being stable (positive recurrent). If $\mathcal{D}$ is stable, then $\mathbb{E}[d_i[k]]^+$ converges to some finite value. Consequently, we will have

$$\lim_{K\to\infty} \frac{1}{K}\mathbb{E}[d_i[k]] \leq \lim_{K\to\infty} \frac{1}{K}\mathbb{E}[[d_i[k]]^+] = 0.$$

Hence,

$$\eta_i \leq 1 - \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[I_i[k]]$$

which implies that the QoS requirement of peer i is satisfied. Next, we will provide a P2P scheme in Algorithm 6 whose performance is summarized in Theorem 23.

Algorithm 6 Optimal P2P scheme (infinite field)

At the beginning of each frame $k-1$, given the arrivals $(e_1^{(k)}, ..., e_M^{(k)})$:

1) Partition the peers into sets $\mathcal{S} = \{i \in \{1, ..., M\} :\ N - e_i^{(k)} \leq T, \sum_j e_j^{(k)} \geq N\}$ and $\mathcal{S}^c$.

If $\mathcal{S} = \emptyset$, nobody can get full-rank. Otherwise,

2) Let all the peers in $\mathcal{S}^c$ transmit all they have initially received for $t = (k-1)T$ to $(k-1)T + T_1 - 1$, where $T_1 = \min\{\sum_{i\in\mathcal{S}^c} e_i^{(k)}, T\}$.

3) **While** $t < kT - 1$ **and** $\exists i \in \mathcal{S} :\ n_i^{(k)}[t] < N$**, do:**

Let $\mathcal{S}'(t) = \{i \in \mathcal{S} :\ n_i^{(k)}[t] \geq N + t - kT\}$.

If $\mathcal{S}' \neq \emptyset$, then let one of the peers in $\mathcal{S}'$ broadcast a chunk.

Otherwise, let $\hat{i} \in \arg \min_{i\in\mathcal{S}:\ n_i^{(k)}[t] > t-(k-1)T} d_i$ transmit.

$t \leftarrow t + 1$.

Theorem 23. *The P2P scheme in Algorithm 6 is throughput optimal, in the sense that it can satisfy* all *achievable QoS metrics* $(\eta_1, ..., \eta_M)$.

Proof. In this proof, we will use the Lyapunov criterion [12] to show the stability of Markov chain $\mathcal{D}$. Let $V[k] = \frac{1}{2}\sum_i([d_i[k]]^+)^2$ be the Lyapunov function at frame k. We will show that for any achievable QoS, the proposed P2P algorithm results in an expected drift

$$\Delta V[k] = \mathbb{E}\left[V[k] - V[k-1] \mid \text{state of the system at frame } k-1\right]$$

which is negative except in a finite subset of the state space and hence the Lyapunov Theorem implies that the Markov chain $\mathcal{D}$ is stable.

$$\begin{aligned}
&\Delta V[k] = \mathbb{E}\left[V[k] - V[k-1] \mid [d_i[k-1]]^+ = d_i : \forall i\right] \\
&= \tfrac{1}{2}\mathbb{E}\left[\sum_i ([d_i[k]]^+)^2 - ([d_i[k-1]]^+)^2 \mid [d_i[k-1]]^+ = d_i : \forall i\right] \\
&= \tfrac{1}{2}\mathbb{E}\left[\sum_i \left([d_i[k-1] + \eta_i - 1 + I_i[k]]^+\right)^2 - (d_i)^2 \mid [d_i[k-1]]^+ = d_i : \forall i\right] \\
&\overset{(a)}{\leq} \tfrac{1}{2}\mathbb{E}\left[\sum_i (d_i + \eta_i - 1 + I_i[k])^2 - (d_i)^2\right] \\
&= \mathbb{E}\left[\sum_i d_i (\eta_i - 1 + I_i[k])\right] + \tfrac{1}{2}\mathbb{E}\left[\sum_i (\eta_i - 1 + I_i[k])^2\right] \\
&\overset{(b)}{\leq} M/2 + \sum_i d_i\eta_i - \mathbb{E}\left[\sum_i d_i(1 - I_i[k])\right]
\end{aligned} \tag{4.33}$$

where (a) follows from $([X+Y]^+)^2 \leq ([X]^+ + Y)^2$, and (b) holds since $(\eta_i - 1 + I_i[k])^2 \leq \max((\eta_i)^2, (1-\eta_i)^2) \leq 1$.

In order to get a negative drift (except in a finite subset), we minimize the above upperbound. That is at the beginning of each frame $k-1$, for given deficit values $([d_1[k-1]]^+, ..., [d_M[k-1]]^+) = (d_1, ..., d_M)$ and any realization of the B2P arrivals $(e_1^{(k)}, ..., e_M^{(k)})$, we solve the following optimization problem,

$$\begin{aligned}
\max \ \sum_i d_i(1 - I_i[k]) = \sum_i d_i 1_{\{e_i^{(k)} + \sum_{j\neq i} x_j^{(k)}[kT] \geq N\}} \\
\text{subject to} \\
x_i^{(kT)}[t] \leq e_i^{(k)} \qquad \text{for all } i \\
\sum_i x_i^{(k)}[kT] \leq T
\end{aligned} \tag{4.34}$$

where the second constraint comes from $x_i^{(k)}[(k-1)T] = 0$ and

$$\sum_i x_i^{(k)}[t] - \sum_i x_i^{(k)}[t-1] \leq 1 \quad \text{for all } (k-1)T \leq t \leq kT.$$

It is seen that the optimization in (4.34) is very similar to the one in (4.25). Hence, we can apply the same argument and verify that the following optimization problem is equivalent to (4.34),

$$\begin{gathered} \max \ \sum_i d_i z_i \\ \text{subject to} \\ z_i \leq 1_{\{e_i^{(k)} \geq N-T, \ \sum_j e_j^{(k)} \geq N\}} \quad \text{for all } i \\ \sum_i z_i \leq N_s^*(e_1^{(k)}, ..., e_M^{(k)}) \end{gathered} \tag{4.35}$$

where $N_s^*(e_1^{(k)}, ..., e_M^{(k)})$ is defined in (4.24).

Considering the solution to the above maximization in (4.33) will result in

$$\begin{aligned} &\Delta V[k] \leq B_2 + \sum_i d_i \eta_i - \mathbb{E}\left[\max \sum_i d_i z_i\right] \\ &\leq B_2 + \sum_i d_i \eta_i - \max \sum_i d_i \mathbb{E}\left[z_i\right] \end{aligned} \tag{4.36}$$

From (4.35), we notice that $\mathbb{E}[z_i]$ must satisfy

$$\begin{gathered} \mathbb{E}[z_i] \leq \mathbb{P}\left(e_i^{(k)} \geq N-T, \ \sum_j e_j^{(k)} \geq N\right) \\ \sum_i \mathbb{E}[z_i] \leq \mathbb{E}\left[\left\lfloor N_s^*(e_1^{(k)}, ..., e_M^{(k)})\right\rfloor\right] \end{gathered} \tag{4.37}$$

In Subsection a, we have shown that an achievable QoS metric $(\eta_1, ..., \eta_M)$ needs to satisfy the above conditions. This suggests that for a strictly achievable QoS metric $(\eta_1, ..., \eta_M)$, for which these conditions hold with strict inequalities, there exists some $\epsilon > 0$ such that

$$\max \sum_i d_i \mathbb{E}\left[z_i\right] \geq \sum_i d_i \eta_i (1+\epsilon) \tag{4.38}$$

Consequently the drift in (4.36) reduces to

$$\Delta V[k] \leq B_2 - \epsilon \sum_i d_i \eta_i \tag{4.39}$$

So for large enough deficit values d_i, the drift is negative. This means the P2P scheme implied by solving (4.34), at each frame, can satisfy any achievable QoS metric and hence is optimal. Also it proves the sufficiency of the conditions in Theorem 22 on the achievability of a QoS metric.

The following lemma, whose proof can be found in Appendix B, completes the proof of Theorem 23,

Lemma 17. *For each block k, given the B2P arrivals $(e_1^{(k)}, ..., e_M^{(k)})$, the P2P scheme in Algorithm 6 solves the optimization problem in (4.34).*

□

c. Distributed Implementation of the P2P Scheme

It is easy to see that the proposed P2P scheme in Algorithm 6 can be efficiently implemented in a decentralized fashion. Lets assume $\sum_j e_j^{(k)} \geq N$. Because first, gathering the information about the total initial rank of the system (*i.e.*, $\sum_j e_j^{(k)}$) requires some extra coordination and transmissions, which may not be practically appealing. Second, if $\sum_j e_j^{(k)} < N$ (there is not enough degrees of freedom in the system), then no matter what we do, the peers cannot recover the original block.

With this assumption, we will show how the three phases of the P2P algorithm can be implemented in a distributed way:

Phase 1- transmissions by peers in $\mathcal{S}^c$: each peer i can individually determine whether he should transmit in the first phase by checking the number of his initial chunks. If $e_i^{(k)} < N - T$, then i attempts to broadcast his $e_i^{(k)}$ initial chunks. Note that coordinating the order of transmissions can be done in a decentralized manner by using the idea of having short control slots and random back-off times to avoid collisions.

Phase 2- transmissions by peers in $\mathcal{S}$ upto some threshold: at each time t peers can separately evaluate their ranks $n_i^{(k)}[t]$ to see whether $n_i^{(k)}[t] \geq N + t - kT$. The peers whose ranks are greater than this threshold can compete, in a distributed way as described above, to broadcast during time slot t.

Phase 3- transmissions based on a Min-Deficit-First discipline: if all the peers hit the threshold in Phase 2, the remaining transmissions should be performed with respect to a Min-Deficit-First scheme. For this purpose, each peer must be able to determine whether he has the minimum deficit or not without exchanging exclusive feedback messages. In order to solve this issue, we propose peers to include their deficit values in the header of the messages they broadcast. Note that this will introduce some overhead which can be neglected if the size of the chunks is reasonably large. In this framework, the deficit values of the peers who transmitted during the first two phases are known by everybody. What if some of the peers did not attempt to transmit in the first two phases? How do the other peers get to know their deficit values? The key point is that each peer can maintain an approximation of the deficit values of the other peers, by storing the last reported values in the most recent transmissions by each of the other peers. Since the deficit values change by at most 1 unit in each frame, these approximations should be within some finite range of the actual values. On the other hand, we have seen in the proof of Theorem 23 that a finite difference in the drift (4.33) will not change the results. Therefore, peers can randomly back off based on their deficit values compared to the approximations of the other values, and if someone achieves the free channel, he can safely broadcast over the P2P network.

3. Coordination of the B2P Network

In the previous section, we proposed a simple P2P scheme to disseminate the chunks received from the B2P network over the broadcast P2P interface. We have shown that the proposed scheme is able to satisfy any achievable long-term QoS metrics and hence is throughput optimal in this sense. In this section, we focus on the B2P part of the system.

Recall our objective was to achieve the QoS metric $(\eta_1, ..., \eta_M)$ at the minimum cost of B2P usage. All peers have identical Bernoulli B2P channels with equal (unit) cost and success probability p, which they use to receive randomly coded chunks from base stations. As discussed in Subsection 2, there is no performance loss if these chunks are transmitted via P2P broadcasts without performing any further coding. The question that we investigate in this section is how long should the B2P channels be used? We will seek an offline solution for this question. In other words, we want to find $0 \leq T_{B2P}(i) \leq T$ which is the number of times peer i attempts to receive a chunk from the B2P channel in a frame.

We will choose these B2P usage time intervals in an offline fashion in such a way that the QoS target can be met with the lowest average cost $C(T_{B2P}(1), ..., T_{B2P}(M)) = \sum_i T_{B2P}(i)$. One can consider any general cost criterion $C(T_{B2P}(1), ..., T_{B2P}(M))$ to be minimized.

For the B2P model described above, we can verify that the B2P arrivals $e_i^{(k)}$ to each peer i are independently and identically distributed over frames k as a Binomial random variable $Bin(p, T_{B2P}(i))$ with parameters p and $T_{B2P}(i)$:

$$\mathbb{P}(e_i^{(k)} = a) = 1_{\{0 \leq a \leq T_{B2P}(i)\}} \binom{T_{B2P}(i)}{a} p^a (1-p)^{T_{B2P}(i)-a}.$$

In order to achieve a given QoS metric $(\eta_1, ..., \eta_M)$, $T_{B2P}(i)$ values must be large enough such that the conditions $(C1)$ and $(C2)$ in Theorem 22 are satisfied. Hence in general, the optimal values of $T^*_{B2P}(i)$ can be derived by solving the following problem,

$$\begin{aligned} &\min C(T_{B2P}(1), ..., T_{B2P}(M)) \\ &\text{s.t.} \\ &0 \leq T_{B2P}(i) \leq T \quad \text{for all } i \\ &\mathbf{e}_i = Bin(p, T_{B2P}(i)) \quad \text{for all } i \\ &(C1) \text{ and } (C2) \text{ in (4.30) are satisfied.} \end{aligned} \tag{4.40}$$

How can one solve for the optimal $T^*_{B2P}(i)$ values?

Note that the minimization problem in (4.40) does not have a known simple form that can be solved efficiently. However since the region for feasible $T_{B2P}(i)$ values is finite (*i.e.,* $\{0, 1, ..., T\}^M$), we can simply adopt an exhaustive search to find the optimal $T^*_{B2P}(i)$ values. However, one can improve the search algorithm by considering lower bounds suggested by $(C1)$ in (4.30) and adopting more efficient search mechanisms like a *branch-and-bound* algorithm. It is noteworthy that this search process needs to be performed only once for the given set of system parameters.

As a special case, we may assume $T_{B2P}(i) = T_{B2P}$ for all i. That is all the peers use their B2P channels for the same duration of time. For this case, the optimal T^*_{B2P} value can be found much faster, since the search region reduces to at most T values. More specifically, if we are interested in the average cost criterion $C(T_{B2P}(1), ..., T_{B2P}(M)) = \sum_i T_{B2P}(i)$, the problem in (4.40) is simplified to

$$\min\Big\{0 \le t \le T : \quad \textstyle\sum_j \eta_j \le \mathbb{E}\left[N^*_s(\mathbf{e}_1, ..., \mathbf{e}_M)\right], \\ \max_i \eta_i \le \mathbb{P}\left(\mathbf{e}_i \ge N - T,\ \sum_j \mathbf{e}_j \ge N\right)\Big\} \tag{4.41}$$

where $\mathbf{e}_i = Bin(p, t)$ are distributed identically for all i and N^*_s is defined as in (4.24).

The following observation declares that we can always find the optimal B2P usage times (even if the achievability region is unknown) with a small overhead, which is the cost to learn the system behavior.

Observation 8. *We can consider two time scales, (i) a short time scale for the dynamic of the queues in the system, and (ii) a larger time scale for scheduling the B2P network. We assume the latter time scale is large enough for the queues in the system to converge to their steady states (if they are stable). Hence starting from $T_{B2P}(i) = T$ for all i, we can gradually modify and search for the optimal set of*

$T_{B2P}(i)$ values over the larger time scale. Since the space of $T_{B2P}(i)$ values is finite, it is guaranteed that in finite time (with respect to the larger time scale) our search algorithm will converge. Note that since the QoS and costs are defined as long-run average parameters, the first few iterations to find the optimal B2P scheme should not cause a major problem in the performance of the system.

In what follows, we consider the special case of symmetric QoS metric $(\eta, ..., \eta)$ and derive some interesting results.

4. Symmetric QoS Constraints

Throughout this section, we focus on average cost criterion and further assume $N \gg M$ and $\eta_i = \eta$ for all i (*i.e.,* symmetric QoS metric). This complete symmetry of the system plus the assumption of $N \gg M$ suggests that the solution to (4.40) should be also symmetric, *i.e.,* $T^*_{B2P}(i) = T^*_{B2P}$ for all i. This immediately implies that $\mathbf{e}_i$ are identically distributed for all peers i. Corollary 6 studies this case and declares the necessary and sufficient condition for achievability of $(\eta, ..., \eta)$ reduces to the constraint $(C2')$,

$$\eta \leq \frac{1}{M}\mathbb{E}\left[N_s^*(\mathbf{e}_1, ..., \mathbf{e}_M)\right].$$

Hence, the optimal $T^*_{B2P(M)}$ value can be found as follows,

$$\min\left\{0 \leq t \leq T : \quad \eta \leq \tfrac{1}{M(N-T)}\mathbb{E}\left[\min\left(|\mathcal{S}|(N-T), [\textstyle\sum_i \mathbf{e}_i(t) - T]^+\right)\right]\right\}, \quad (4.42)$$

where $\mathcal{S} = \{i \in \{1, ..., M\} : \ N - \mathbf{e}_i(t) \leq T, \sum_j \mathbf{e}_j(t) \geq N\}$ and $\mathbf{e}_i(t) = Bin(p, t)$.

Note that we used $T^*_{B2P(M)}$ notation to explicitly emphasize the dependence of the optimal T^*_{B2P} value on the number of peers M, similarly $\mathbf{e}_i(t)$ reminds that the random variable $\mathbf{e}_i$ depends on the time parameter t. We can rephrase (4.42) as follows,

$$T^*_{B2P(M)} = \min\Big\{0 \le t \le T : \\ M\eta \le \mathbb{E}\left[\min\left(1_{(\sum_1^M \mathbf{e}_j(t) \ge N)} \sum_{i=1}^{M} 1_{(\mathbf{e}_i(t) \ge N-T)}, [\sum_{j=1}^{M} \mathbf{e}_j(t) - T]^+/(N-T)\right)\right]\Big\}, \tag{4.43}$$

Observation 9. *Note that in Corollary 6, we have shown for the symmetric case that condition* $(C2)$ *implies* $(C1)$ *in (4.30). That is in order to meet the QoS constraints we also need to have* $\eta \le \mathbb{P}(\mathbf{e}_i(t) \ge N-T, \sum_1^M \mathbf{e}_j(t) \ge N)$. *Hence* $\underline{T}^*_{B2P(M)} = \min\{0 \le t \le T : \eta \le \mathbb{P}(\mathbf{e}_i(t) \ge N-T, \sum_1^M \mathbf{e}_j(t) \ge N)\}$ *provides a lower bound on* $T^*_{B2P(M)}$.

In the following lemma, we show that $T^*_{B2P(M)}$ is always a non-increasing function of M.

Lemma 18. *Suppose that* $t \in [0, T]$ *satisfies the following inequality,*

$$M\eta \le \mathbb{E}\left[\min\left(1_{(\sum_1^M \mathbf{e}_j(t) \ge N)} \sum_{i=1}^{M} 1_{(\mathbf{e}_i(t) \ge N-T)}, [\sum_{j=1}^{M} \mathbf{e}_j(t) - T]^+/(N-T)\right)\right], \tag{4.44}$$

then the same t *satisfies the inequality for* $M+1$ *as well,*

$$(M+1)\eta \le \mathbb{E}\left[\min\left(1_{(\sum\limits_{1}^{M+1} \mathbf{e}_j(t) \ge N)} \sum_{i=1}^{M+1} 1_{(\mathbf{e}_i(t) \ge N-T)}, \frac{[\sum\limits_{j=1}^{M+1} \mathbf{e}_j(t) - T]^+}{(N-T)}\right)\right]. \tag{4.45}$$

Consequently, we have $T^*_{B2P(M)} \ge T^*_{B2P(M+1)}$.

Proof. First notice that

$$\begin{aligned}
&\min\left(1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}\sum_{i=1}^{M+1}1_{(\mathbf{e}_i(t)\geq N-T)},\frac{[\sum\limits_{j=1}^{M+1}\mathbf{e}_j(t)-T]^+}{(N-T)}\right)\\
&\overset{(a)}{=}1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}\min\left(\sum_{i=1}^{M+1}1_{(\mathbf{e}_i(t)\geq N-T)},\frac{\sum\limits_{j=1}^{M+1}\mathbf{e}_j(t)-T}{(N-T)}\right),\\
&=1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}\min\left(\sum_{i=1}^{M}1_{(\mathbf{e}_i(t)\geq N-T)}+1_{(\mathbf{e}_{M+1}(t)\geq N-T)},\frac{\sum\limits_{j=1}^{M}\mathbf{e}_j(t)-T}{(N-T)}+\frac{\mathbf{e}_{M+1}}{(N-T)}\right)\\
&\overset{(b)}{\geq}1_{(\sum_1^{M}\mathbf{e}_j(t)\geq N)}\min\left(\sum_{i=1}^{M}1_{(\mathbf{e}_i(t)\geq N-T)},\frac{\sum\limits_{j=1}^{M}\mathbf{e}_j(t)-T}{(N-T)}\right)\\
&+1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}\min\left(1_{(\mathbf{e}_{M+1}(t)\geq N-T)},\frac{\mathbf{e}_{M+1}}{(N-T)}\right),
\end{aligned}\tag{4.46}$$

where (a) holds, because if $1_{(\sum\limits_1^{M+1}\mathbf{e}_j(t)\geq N)}=1$, then $\sum\limits_{j=1}^{M+1}\mathbf{e}_j(t)\geq N>T$ and both sides will be the same. Otherwise when $1_{(\sum\limits_1^{M+1}\mathbf{e}_j(t)\geq N)}=0$, both sides are identically zero. The following argument proves the validity of inequality (b), by considering different cases:

(1) Let $1_{(\sum_1^{M}\mathbf{e}_j(t)\geq N)}=1$: consequently we have $1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}=1$ as well and also note that $\min(X+c,Y+d)\geq\min(X,Y)+\min(c,d)$ always holds. Using these two observations, one can easily verify inequality (b) for this case.

(2) Let $1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}=0$: consequently we have $1_{(\sum_1^{M}\mathbf{e}_j(t)\geq N)}=0$ and both sides of inequality (b) are identically zero.

(3) Let $1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}=1$ but $1_{(\sum_1^{M}\mathbf{e}_j(t)\geq N)}=0$: for this case inequality (b) reduces to

$$\begin{aligned}
&\min\left(\sum_{i=1}^{M}1_{(\mathbf{e}_i(t)\geq N-T)}+1_{(\mathbf{e}_{M+1}(t)\geq N-T)},\frac{\sum\limits_{j=1}^{M}\mathbf{e}_j(t)-T}{(N-T)}+\frac{\mathbf{e}_{M+1}}{(N-T)}\right)\\
&\geq\min\left(1_{(\mathbf{e}_{M+1}(t)\geq N-T)},\frac{\mathbf{e}_{M+1}}{(N-T)}\right).
\end{aligned}\tag{4.47}$$

Note that if $\sum_{j=1}^{M} \mathbf{e}_j(t) - T \geq 0$, the we have

$$\begin{aligned}
&\min\left(\sum_{i=1}^{M} 1_{(\mathbf{e}_i(t)\geq N-T)} + 1_{(\mathbf{e}_{M+1}(t)\geq N-T)}, \frac{\sum_{j=1}^{M} \mathbf{e}_j(t)-T}{(N-T)} + \frac{\mathbf{e}_{M+1}}{(N-T)}\right)\\
&\geq \min\left(\sum_{i=1}^{M} 1_{(\mathbf{e}_i(t)\geq N-T)}, \frac{\sum_{j=1}^{M} \mathbf{e}_j(t)-T}{(N-T)}\right) + \min\left(1_{(\mathbf{e}_{M+1}(t)\geq N-T)}, \frac{\mathbf{e}_{M+1}}{(N-T)}\right)\\
&\geq \min\left(1_{(\mathbf{e}_{M+1}(t)\geq N-T)}, \frac{\mathbf{e}_{M+1}}{(N-T)}\right).
\end{aligned}$$

Otherwise when $\sum_{j=1}^{M} \mathbf{e}_j(t) - T < 0$, since we have considered $1_{(\sum_1^{M+1} \mathbf{e}_j(t)\geq N)} = 1$ for this case, we have

$$\sum_{1}^{M+1} \mathbf{e}_j(t) - T \geq N - T,$$

and

$$\mathbf{e}_{M+1} + \sum_{1}^{M} \mathbf{e}_j(t) \geq N.$$

Therefore, $\mathbf{e}_{M+1} \geq N - \sum_1^M \mathbf{e}_j(t) \geq N - T$. We can now verify that the right hand side of (4.47) is 1, while the left hand side is

$$\min\left(\sum_{i=1}^{M} 1_{(\mathbf{e}_i(t)\geq N-T)} + 1, \frac{\sum_{j=1}^{M+1} \mathbf{e}_j(t) - T}{(N-T)}\right) \geq 1.$$

This completes the verification of inequality (b) in (4.46). Now take an expectation from both sides of (4.46) to get,

$$
\begin{aligned}
&\mathbb{E}\left[\min\left(1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}\sum_{i=1}^{M+1}1_{(\mathbf{e}_i(t)\geq N-T)},\frac{[\sum_{j=1}^{M+1}\mathbf{e}_j(t)-T]^+}{(N-T)}\right)\right]\\
&\geq \mathbb{E}\left[1_{(\sum_1^{M}\mathbf{e}_j(t)\geq N)}\min\left(\sum_{i=1}^{M}1_{(\mathbf{e}_i(t)\geq N-T)},\frac{\sum_{j=1}^{M}\mathbf{e}_j(t)-T}{(N-T)}\right)\right]\\
&+\mathbb{E}\left[1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}\min\left(1_{(\mathbf{e}_{M+1}(t)\geq N-T)},\frac{\mathbf{e}_{M+1}}{(N-T)}\right)\right]\\
&\overset{(c)}{\geq} M\eta+\mathbb{E}\left[1_{(\sum_1^{M+1}\mathbf{e}_j(t)\geq N)}1_{(\mathbf{e}_{M+1}(t)\geq N-T)}\right]\\
&= M\eta+\mathbb{P}\left(\sum_1^{M+1}\mathbf{e}_j(t)\geq N,\mathbf{e}_{M+1}(t)\geq N-T\right)\overset{(d)}{\geq}(M+1)\eta,
\end{aligned}
\tag{4.48}
$$

where (c) follows from (4.44) and the fact that $1_{(\mathbf{e}_{M+1}(t)\geq N-T)}\leq\frac{\mathbf{e}_{M+1}(t)}{N-T}$, and (d) holds by Observation 9.

Therefore, if (4.44) holds, then so does (4.45), and as a result $T^*_{B2P(M)}$ is non-increasing by M. □

The following lemma completes the characterization of $T^*_{B2P(M)}$ as a function of number of peers M.

Lemma 19. *Let $T^*_{B2P(M)}$ be the optimal B2P usage time, defined as in (4.42). The followings are true,*

1. *For each value of M,*

   $T^*_{B2P(M)}\geq\underline{T}^*_{B2P(M)}=\min\{0\leq t\leq T:\eta\leq\mathbb{P}(\mathbf{e}_i(t)\geq N-T,\sum_1^M\mathbf{e}_j(t)\geq N)\}.$

2. *$\{\underline{T}^*_{B2P(M)}\}_M$ is a monotone sequence converging to some value $\underline{T}^*_{B2P}$ from above, where $\underline{T}^*_{B2P}=\min\{0\leq t\leq T:\eta\leq\mathbb{P}(\mathbf{e}_i(t)\geq N-T)\}$.*

3. *$\{T^*_{B2P(M)}\}_M$ is also a monotone sequence converging to the same value $\underline{T}^*_{B2P}$ from above.*

Proof. Note that the lower bound $\underline{T}^*_{B2P(M)}$ was found in Observation 9. Also it is clearly seen that

$$\mathbb{P}(\mathbf{e}_i(t) \geq N - T, \sum_1^M \mathbf{e}_j(t) \geq N) \leq \mathbb{P}(\mathbf{e}_i(t) \geq N - T, \sum_1^{M+1} \mathbf{e}_j(t) \geq N).$$

Therefore, $\underline{T}^*_{B2P(M)}$ is monotone non-increasing and converges to some value $\underline{T}^*_{B2P}$. However as M goes to infinity, $\sum_1^M \mathbf{e}_j(t) \geq N$ holds true with probability 1, and consequently $\underline{T}^*_{B2P} = \min\{0 \leq t \leq T : \eta \leq \mathbb{P}(\mathbf{e}_i(t) \geq N - T)\}$.

The monotonicity of $T^*_{B2P(M)}$ was proved in Lemma 18, so it remains to show that $T^*_{B2P(M)} \to \underline{T}^*_{B2P}$.

Note that as M grows large, $1_{(\sum_1^M \mathbf{e}_j(t) \geq N)} = 1$ and by *the strong law of large numbers*

$$\frac{1}{M} \sum_{i=1}^M 1_{(\mathbf{e}_i(t) \geq N - T)} \overset{a.s.}{\to} \mathbb{P}\left(\mathbf{e}_i(t) \geq N - T\right)$$

and

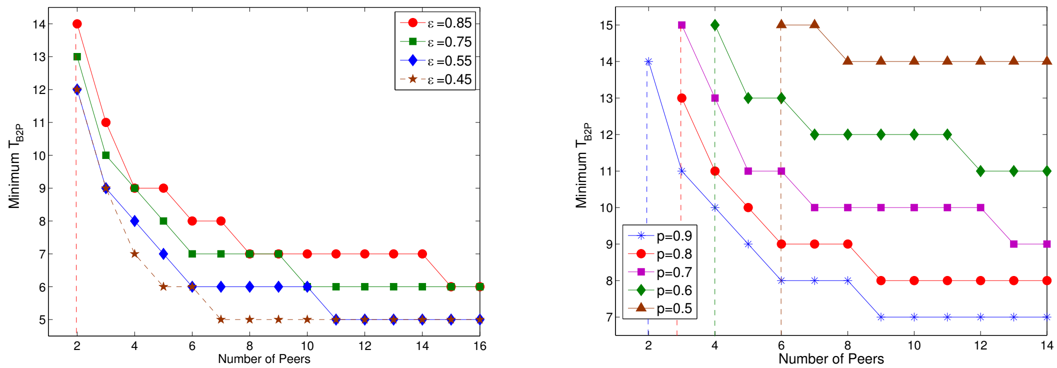
$$\frac{1}{M} [\sum_{j=1}^M \mathbf{e}_j(t) - T]^+ / (N - T) \overset{a.s.}{\to} \mathbb{E}[\mathbf{e}_j(t)] / (N - T),$$

in which $a.s.$ stands for *almost surely convergence* of random variables. On the other hand, by *the Markov's inequality* for the non-negative random variable $\mathbf{e}_i(t)$, we have

$$\mathbb{P}\left(\mathbf{e}_i(t) \geq N - T\right) \leq \frac{\mathbb{E}[\mathbf{e}_j(t)]}{(N - T)}.$$

Hence, as M tends to infinity, $T^*_{B2P(M)}$ in (4.42) reduces to $\min\{0 \leq t \leq T : \eta \leq \mathbb{P}(\mathbf{e}_i(t) \geq N - T)\} = \underline{T}^*_{B2P}$. □

The variation of $T^*_{B2P(M)}$ as M changes is shown in Figures 12(a) and 12(b). The non-increasing nature of $T^*_{B2P(M)}$ is clearly seen in these figures. Also we observe that with an intermediate (not very large) number of peers in the system, we can achieve the minimum possible cost. This number is at most $M = 15$ for the configurations plotted in these figures. Figure 12(a) displays the variation of the minimum cost as



(a) Variation of η for $p = 0.9$

(b) Variation of p for $\eta = 0.9$

Fig. 12. Minimum cost $T^*_{B2P(M)}$ vs. number of peers M ($N = 20$, $T = 15$, $q = \infty$)

the delivery ratio η changes. As expected, satisfying a larger delivery ratio incurs more cost. The variation of cost as a function of B2P channel quality (probability of success p) is demonstrated in Figure 12(b). By increasing success probability p, lower costs are sufficient to achieve a given QoS ($\eta = 0.9$ for this case).

There are two noteworthy points that can be inferred from these graphs. First, as indicated by the dotted lines, there is a limit, on the minimum number of peers in the system, for a QoS to be achievable. For example, if $p = 0.6$ we need at least $M = 4$ collaborative peers in the system to achieve $\eta = 0.9$. Second, for a fixed channel quality p, the smaller η is, the smaller number of peers M is required to achieve minimum possible cost $\underline{T}^*_{B2P}$. On the other hand, if we fix QoS η and change the channel quality p, we cannot get such a monotone behavior. For example in Figure 12(b), the minimum number of peers to achieve the minimum cost for $p = 0.5, 0.7$ and 0.9 is respectively $M = 8, 7$ and 9.

5. Finite Field Case

So far we only focused on infinite field size case, where we assumed the field size q is large enough such that all randomly coded chunks are linearly independent with a probability almost equal to 1. With this assumption, it was sufficient for each peer to receive N distinct coded chunks in order to recover the original block. We found some necessary and sufficient conditions on the achievability of a given QoS metric $(\eta_1, ..., \eta_M)$ in Subsection 2. An optimal scheme was proposed in Algorithm 6 for coordinating the P2P broadcast transmissions, and in Subsection 3 we have shown how to choose the B2P usage times optimally in order to minimize a general cost criterion.

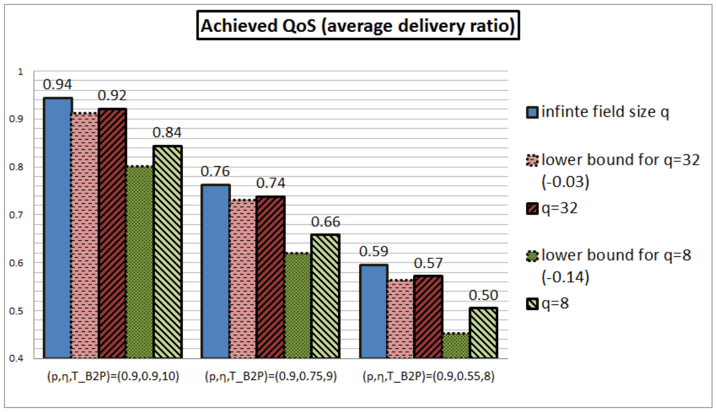
In this section, we turn our attention into the finite field case, where the field size $q < \infty$ and there is a non-zero probability that randomly coded chunks are linearly dependent. More specifically, we are interested in evaluating the performance of our proposed B2P and P2P schemes in the finite fields.

Theorem 24. *Suppose the coefficients for coding the chunks are drawn uniformly at random from a field of size $q \geq 2$. If we apply the Algorithm 6 for the P2P broadcasts and choose the B2P usage times according to the optimization problem in (4.40), then for each peer i we have*

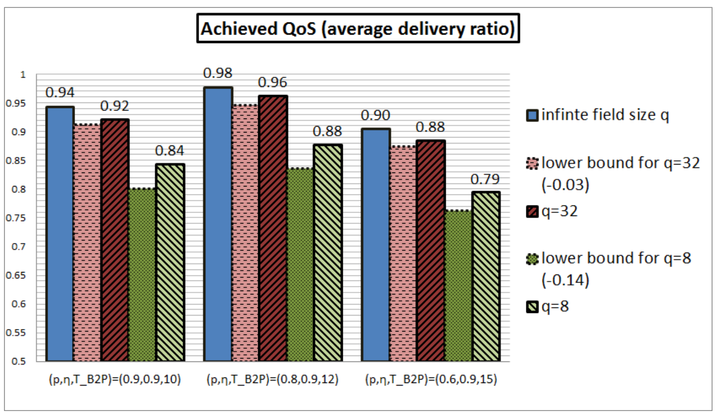
$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[I_i[k]] \leq 1 - \eta_i + \frac{1}{q-1}. \tag{4.49}$$

That is on average each peer i can successfully receive and play out the multimedia stream at least $\eta_i - \frac{1}{q-1}$ fraction of time.

Note that the effect of finiteness of the field size is limited by the value $\frac{1}{q-1}$, for example for field size $q = 32$, there is only around %3 reduction in the quality of service. In the simulation results presented in Figures 13(a) and 13(b), we see the actual reduction in the QoS is even less than this value.



(a) Variation of η for $p = 0.9$



(b) Variation of p for $\eta = 0.9$

Fig. 13. Achievable delivery ratio with finite field sizes ($N = 20$, $T = 15$, $M = 4$, with symmetric peers)

Proof. Lets define $\hat{I}_i[k] = 1_{\{\hat{n}_i^{(k)}[kT]<N\}}$, where $\hat{n}_i^{(k)}[kT]$ is the number of coded chunks of block k peer i has received by the end of its deadline. Note that in our previous arguments for the infinite field size, we had $n_i^{(k)}[t] = \min\{N, \hat{n}_i^{(k)}[t]\}$ and $\hat{I}_i[k] = I_i[k]$.

In the discussion followed by Algorithm 6, and the corresponding result presented in Theorem 23, we have shown that

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[\hat{I}_i[k]] \leq 1 - \eta_i \tag{4.50}$$

holds true, when we employ the P2P scheme in Algorithm 6. For $I_i[k]$, we have

$$\begin{aligned} &\mathbb{E}[I_i[k]] = \mathbb{E}[I_i[k]|\hat{I}_i[k] = 1]\mathbb{P}(\hat{I}_i[k] = 1) + \mathbb{E}[I_i[k]|\hat{I}_i[k] = 0]\mathbb{P}(\hat{I}_i[k] = 0) \\ &\overset{(a)}{\leq} \mathbb{E}[\hat{I}_i[k]] + \mathbb{E}[I_i[k]|\hat{n}_i^{(k)}[kT] \geq N] \\ &= \mathbb{E}[\hat{I}_i[k]] + \mathbb{P}[n_i^{(k)}[kT] < N|\hat{n}_i^{(k)}[kT] \geq N] \\ &\overset{(b)}{\leq} \mathbb{E}[\hat{I}_i[k]] + \tfrac{1}{q-1} \end{aligned} \tag{4.51}$$

where (a) follows since $\mathbb{E}[I_i[k]|\hat{I}_i[k] = 1] = 1$ and $\mathbb{P}(\hat{I}_i[k] = 1) = \mathbb{E}[\hat{I}_i[k]]$, and (b) holds by Lemma 13 in Section A.

If we sum both sided of (4.51) from $k = 1$ to K, divide the result by K and let K tend to infinity, then the desired bound in (4.49) will be achieved using (4.50). □

6. Further Discussions and Extensions

This section addresses other features which can be added into our framework. In Subsection a, we will discuss how the system performance can be enhanced by adding some extremely altruistic peers (called *boosters*). The problem of maintaining the fairness among the peers will be studied in Subsection b. Finally, we consider using some erasure protection coding schemes in order to improve the peers' quality of experience in Subsection c. We propose a new throughput optimal P2P scheme which can support applications in which such coding techniques are employed.

a. The Effect of Boosters

In order to improve the performance of our system or even make the system capable of satisfying an unachievable QoS metric, we can add a number of *boosters*. The role of boosters in a system is to help delivering more coded chunks to the peers. We can easily incorporate the effect of boosters into our framework by adding some peers with zero delivery ratio.

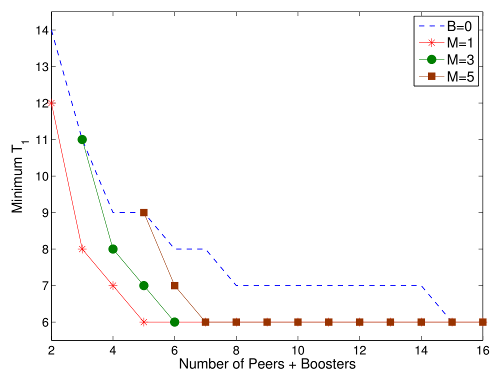
If we assume there are B boosters in the system who all individually use their B2P channels T_b times per frame, then they can collectively contribute $\mathbf{e}_b(T_b)$ number of coded chunks over the P2P network. $\mathbf{e}_b(T_b) = Bin(p, BT_b)$ is a Binomial random variable with parameters p and BT_b. Now in the presence of these B boosters, we need to satisfy the following conditions in order to guarantee the QoS requirements of the peers,

$$\begin{aligned} &(C1B) \quad \eta_i \leq \mathbb{P}\left(\mathbf{e}_i \geq N-T,\ \textstyle\sum_j \mathbf{e}_j + \mathbf{e}_b \geq N\right) \qquad \text{for all } i \\ &(C2B) \quad \textstyle\sum_i \eta_i \leq \\ &\qquad \mathbb{E}\left[\min\left(1_{(\sum_1^M \mathbf{e}_j + \mathbf{e}_b \geq N)} \sum_{i=1}^{M} 1_{(\mathbf{e}_i \geq N-T)}, [\sum_{j=1}^{M} \mathbf{e}_j + \mathbf{e}_b - T]^+/(N-T)\right)\right]. \end{aligned} \tag{4.52}$$

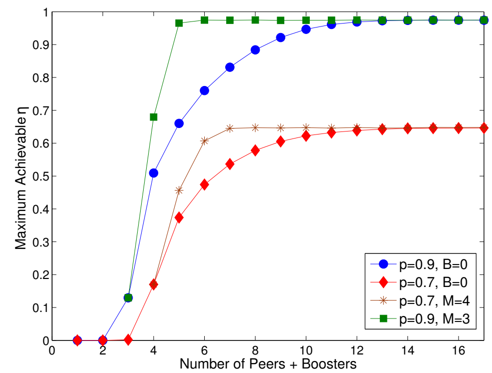
A similar optimization problem as (4.40), with constraints $(C1B)$ and $(C2B)$ and a cost criterion which is also a function of T_b, can be considered to find the optimal B2P usage times. In order to coordinate the P2P transmissions, we first let the boosters broadcast all the $\mathbf{e}_b$ chunks they have received from the B2P network, and then follow the Algorithm 6.

Figures 14(a) and 14(b) depict the effect of boosters on the minimum B2P cost and the achieved delivery ratio respectively. We consider a symmetric configuration in which all the peers and boosters use their B2P channels for the same duration.

In Figures 14(a), the dotted line shows the variation of the minimum B2P usage time versus the number of peers M, when there is no booster. In the other curves, we show how the B2P usage time changes if we fix the number of peers ($M = 1, 3, 5$ for this plot) and add boosters to the system. It is interesting to note that a small number of boosters are sufficient to help the peers achieve their minimum possible



(a) Minimum Cost vs. Number of Entities (Peers+Boosters) for $p = 0.9$



(b) Maximum Achievable Delivery Ratio vs. Number of Entities for Fixed $T_{B2P} = 7$

Fig. 14. Effect of boosters ($N = 20$, $T = 15$, $q = \infty$)

B2P usage time. For example if $M = 5$, with adding only two boosters we will achieve the minimum $\underline{T}^*_{B2P}$, while without boosters we need to add 10 more peers in order to get the same B2P usage time.

We can observe a similar effect in Figure 14(b). That is if we fix the B2P usage time, by increasing the number of peers (red and blue curves with $B = 0$), higher delivery ratios are achievable by the peers. However, if we fix the number of peers and add boosters to the system the delivery ratio will increase much faster. For example, if $p = 0.7$ and $M = 4$, the maximum quality is achievable by adding only 3 boosters.

b. The Fairness Problem

Fairness is one of the most important factors that should be considered in designing practical systems. The problem of guaranteeing some level of fairness in the systems with multiple entities has been extensively studied in literature [47], [48], [49].

In our model, there are a number of peers, with potentially different QoS requirements, who need to individually download information from the external servers (*i.e.,* B2P network) and cooperatively share this information over a broadcast P2P network. Our objective in this subsection is to investigate the fairness issue among these peers. Note that downloading content from the B2P network incurs some cost to the peers. Hence, it may not be fair if we require a peer with a small delivery ratio requirement to use its B2P interface for the same duration as another peer who has a much larger QoS requirement. In what follows, we show how we can incorporate the concept of fairness into out framework.

Fairness with respect to B2P download rate: Consider a group of peers with different QoS constraints $\eta_1, ..., \eta_M$. One way to introduce fairness to our model is to let peers use their B2P network proportionally to their delivery ratios. Our model is general enough to easily handle this case, by modifying the definition of the cost function $C(T_{B2P}(1), ..., T_{B2P}(M))$ in (4.40). Hence, one can define different cost criteria in order to meet different desired levels of fairness. For example in order to maintain proportional fairness among peers, we can let $T_{B2P}(i) = \alpha\eta_i$ for all peers i. Therefore, $C(T_{B2P}(1), ..., T_{B2P}(M)) = \alpha \sum_i \eta_i$ and we can solve (4.40) to find the smallest value of α.

c. Improving Robustness by Employing Erasure Protection Techniques

Consider an application of streaming a multimedia channel. In many conventional systems, the multimedia content is coded as a single stream and peers need to receive all the constituent packets of the stream in order to playout the multimedia at their own devices. This coding method is used in video standards like MPEG-1/2/4 and H.261/3.

One of the proposed techniques for robust multimedia streaming is *Multiple Description Coding (MDC)* [50]. It is well understood that using MDC we can improve the streaming quality. The idea in MDC is to code a multimedia channel into a number of descriptions and in order to achieve the highest quality a peer needs to receive all the descriptions. However, the quality degrades gradually as the number of received descriptions reduces. For example consider an MDC scheme which codes a stream into two descriptions. By receiving one description, one can decode the media stream at the basic quality and for the best media experience he needs to get another description as well. It can be seen how MDC is useful in heterogeneous systems with various bandwidths and quality requirements. The combination of MDC and random linear network coding has been also studied [51] and shown to have advantages in increasing robustness and decreasing the complexity of the construction. In this model, the media server codes each generation of the stream using MDC into C descriptions. These C descriptions are further coded using random linear coding before transmissions. At the receivers, the number of linearly independent descriptions will determine the rate at which the source information can be decoded.

A similar erasure protection technique, based on *Priority Encoding Transmissions (PET)* [52], has been considered in [53]. The idea in PET is to partition the multimedia content into layers of different importance. The level of protection of different layers increases by their importance. A peer who receives more layers, will achieve a better quality. However, decoding a layer requires receiving the previous (more important) layers. Now we can use network coding together with PET (as mentioned in [53]) to get a better performance. In this scenario, rank of the received matrix of encoding vectors determines the quality of media playout.

In our proposed framework, we did not consider such erasure protection schemes and assumed that there exists a single stream in the form of a sequence of blocks and each peer needs to successfully receive all N chunks in a block in order to be able to play it out. To clarify the value of these schemes, note that each peer requires to receive N linearly independent coded chunks within T slots in order to be able to recover the corresponding block. Now suppose by the deadline T, a peer i has received only $N-1$ independent chunks. In the current framework, peer i wastes all he has received in this frame because no useful information can be extracted out of $N-1$ coded chunks. However peer i could potentially playout the corresponding block with a reasonable quality if the stream was appropriately precoded using one of the mentioned schemes.

In this subsection, we shall study the application of these schemes in our live streaming problem. We will slightly modify our model and present a new P2P scheme for this extended variation.

We assume the media server precodes the blocks of the multimedia stream in such a way that peers can still decode the original blocks by partially receiving them. Combining this technique with random linear network coding, we consider the following general QoS metric for our problem.

We assume there are L levels of satisfaction, directly associated with the achievable qualities, with respect to the number of linearly independent chunks a peer receives. More specifically lets assume $0 = N_0 < N_1 < N_2 < ... < N_L = N$ and $0 \le w_L \le w_{L-1} \le ... \le w_1 \le 1$, such that $\sum_l w_l = 1$ and w_l is the marginal happiness of a peer when his rank increase from N_{l-1} to N_l. With this notation, the satisfaction level $s_i^{(k)}$ of peer i in the k^{th} frame (corresponding to the service of the k^{th} block) is

$$s_i^{(k)} = \sum_{l=1}^{L} w_l 1_{\{n_i^{(k)} \ge N_l\}}. \tag{4.53}$$

The new QoS metric is defined as a minimum requirement on the long-run average satisfaction of each peer i:

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} s_i^{(k)} \ge \eta_i \quad \text{for all } i = 1, ..., M. \tag{4.54}$$

In what follows we assume that the B2P usage times are appropriately chosen such that the QoS metric $(\eta_1, ..., \eta_M)$ is achievable for a given set of N_l and w_l values. We shall see how the P2P scheme proposed in Algorithm 6 should change in order to address the new QoS model.

Using the same methodology as in Subsection 2, we can consider the following deficit queue dynamics in order to keep track of satisfaction level of each peer i:

$$d_i[k] = d_i[k-1] + \eta_i - s_i^{(k)}. \tag{4.55}$$

One can easily verify that stability of the vector $([d_1[k]]^+, ..., [d_M[k]]^+)$, as k grows, implies satisfying the QoS metric.

Note that the B2P arrivals are assumed to be independently and identically distributed over frames. Hence, similar to the argument in [11], the achievability of the new QoS metric can be defined based on the existence of a randomized stationary P2P policy as follows,

Definition 8. *Given the B2P arrival processes* $(\mathbf{e}_1, ..., \mathbf{e}_M)$*, the QoS metric* $(\eta_1, ..., \eta_M)$ *is (strictly) achievable for a given set of* N_l *and* w_l *values if and only if there exists a policy* $\mathbf{P}^*$*, to coordinate the P2P transmissions, with the following properties.*
Given the realization of the B2P arrivals $e^{(k)} = (e_1^{(k)}, ..., e_M^{(k)})$ *for each block* k*,* $\mathbf{P}^*$ *chooses a feasible vector of P2P transmissions* $x^{(k)} = (x_1^{(k)}, ..., x_M^{(k)})$ *(satisfying* $\sum_i x_i^{(k)} \leq T$ *and* $x_i^{(k)} \leq e_i^{(k)}$ *for all* i*) with probability* $\mathbb{P}\left(x^{(k)}|e^{(k)}\right)$*, such that for each peer* i*,*

$$\mathbb{E}_{\mathbf{e}}\left[\sum_{x^{(k)}} s_i^{(k)} \mathbb{P}\left(x^{(k)}|e^{(k)}\right) \mid e^{(k)}\right] > \eta_i \tag{4.56}$$

where $\mathbb{E}_{\mathbf{e}}[.]$ *is expectation with respect to the B2P arrival processes.*

Like before, we will utilize the Lyapunov stability criterion to find a throughput optimal P2P scheme for this case. Lets consider the same Lyapunov function $V[k] =$

$\frac{1}{2}\sum_i([d_i[k]]^+)^2$ as in Subsection 2. Our approach is to minimize an upperbound on the expected drift $\Delta V[k]$ to guarantee the stability of the deficit queues.

It is straightforward to verify the following bound on the drift,

$$\begin{aligned}&\Delta V[k] = \mathbb{E}\left[V[k] - V[k-1] \mid [d_i[k-1]]^+ = d_i : \forall i\right] \\ &\leq M/2 + \textstyle\sum_i d_i\eta_i - \mathbb{E}\left[\textstyle\sum_i d_i s_i^{(k)}\right].\end{aligned} \tag{4.57}$$

Minimizing the above upperbound suggests the following framework.
At the beginning of each frame $k-1$, for given deficit values $([d_1[k-1]]^+, ..., [d_M[k-1]]^+) = (d_1, ..., d_M)$ and any realization of the B2P arrivals $(e_1^{(k)}, ..., e_M^{(k)})$, solve the following problem,

$$\begin{aligned}\max\ W(x^{(k)}) = \textstyle\sum_i d_i s_i^{(k)} = \textstyle\sum_i d_i(\sum_{l=1}^{L} w_l 1_{\{n_i^{(k)} \geq N_l\}})& \\ \text{subject to}\qquad\qquad& \\ n_i^{(k)} = e_i^{(k)} + \textstyle\sum_{j\neq i} x_j^{(k)} \quad \text{for all } i& \\ 0 \leq x_i^{(k)} \leq e_i^{(k)} \qquad\quad \text{for all } i& \\ \textstyle\sum_i x_i^{(k)} \leq T\qquad\qquad\qquad&\end{aligned} \tag{4.58}$$

Let $W(x^{(k)^*})$ be the maximum objective value achieved by solving the above maximization problem. Note that policy $\mathbf{P}^*$ randomly chooses a feasible $x^{(k)}$ according to some distribution $\mathbb{P}\left(x^{(k)}|e^{(k)}\right)$, hence one can easily verify that

$$W(x^{(k)^*}) \geq \sum_{x^{(k)}} W(x^{(k)})\mathbb{P}\left(x^{(k)}|e^{(k)}\right).$$

Taking expectation from both sides of the above inequality with respect to the arrival processes results in

$$\begin{aligned}&\mathbb{E}[\textstyle\sum_i d_i s_i^{(k)^*}] = \mathbb{E}[W(x^{(k)^*})] \geq \mathbb{E}[\sum_{x^{(k)}} W(x^{(k)})\mathbb{P}\left(x^{(k)}|e^{(k)}\right)] \\ &\overset{(a)}{>} \textstyle\sum_i d_i\eta_i \geq \sum_i d_i(\eta_i + \epsilon)\end{aligned} \tag{4.59}$$

for some small enough $\epsilon > 0$, where (a) follows from the definition of $W(x^{(k)})$ and

(4.56). By considering (4.59) and (4.57), we conclude that using the P2P schedule $x^{(k)^*}$ will result in

$$\Delta V[k] \leq \frac{M}{2} - \epsilon \sum_i (d_i),$$

which is negative for large enough d_i queue lengths, and hence can stabilize the queues. Therefore, the solution to (4.58) will provide an optimal scheme to coordinate P2P transmissions at each frame. In what follows, we investigate this optimal P2P scheme further to find a more simple and intuitive form of this algorithm.

Note that if $\sum_i e_i^{(k)} \leq T$, the optimal solution to the above problem is trivially $x_i^{(k)} = e_i^{(k)}$ for all i. Otherwise, the whole system has received enough DoFs to utilize all the T time slots in the frame, *i.e,* we can safely let $\sum_i x_i^{(k)} = T$, and we have

$$\begin{aligned} &\max \ \textstyle\sum_i d_i \sum_{l=1}^{L} w_l z_{il} \\ &\text{subject to} \\ &z_{il} \leq 1_{\{e_i^{(k)} - x_i^{(k)} \geq N_l - T\}} \quad \text{for all } i \text{ and } l \\ &0 \leq x_i^{(k)} \leq e_i^{(k)} \quad \text{for all } i \\ &\textstyle\sum_i x_i^{(k)} = T \end{aligned} \tag{4.60}$$

We define $l_{min} = \min\{l : N_l - T \geq 0\}$ and $l_i = \max\{l : N_l - T \leq e_i^{(k)}\}$ for each peer i. Note that since $N_L = N > T$, l_{min} is well defined, also we let $N_0 = 0$ that makes sure $l_i \geq 0$ is clearly defined. Also it should be clear that

$$\begin{aligned} &z_{il} \geq z_{i(l+1)} \\ &z_{il} = 0 \quad \text{for all } l > l_i \text{ and } i \\ &z_{il} = 1 \quad \text{for all } l < l_{min} \text{ and } i. \end{aligned} \tag{4.61}$$

We shall see the problem (4.60) can be solved in two rounds.

Round 1: Initially we can let $x_i^{(k)} = \min\{e_i^{(k)} - (N_{l_i} - T), e_i^{(k)}\}$ without affecting the objective function and as a result $\hat{T} = [T - \sum_i(\min\{e_i^{(k)} - (N_{l_i} - T), e_i^{(k)}\})]^+$

transmissions will be left for this frame. If $\hat{T} = 0$, then no further transmissions need to be done. Otherwise $\hat{T} > 0$ and we find the best choice for the remaining transmissions in Round 2.

Round 2: For each peer i (with $l_i \geq l_{min}$) we need to determine the following decision variables,

$$z_{il} \in \{0,1\} \qquad \text{for } l = l_i, l_i - 1, ..., l_{min}.$$

Note that if $l_i < l_{min}$, then in the first round of transmissions we should have chosen $x_i^{(k)} = e_i^{(k)}$ and no further transmissions can be done by this peer. Let $\hat{M} = |\{i : l_i \geq l_{min}\}|$ be the number of peers whose transmissions should be determined in this round, and without loss of generality we use indices $\{1, ..., \hat{M}\}$ to denote these peers. We further assume $d_1 \geq d_2 \geq ... \geq d_{\hat{M}}$.

Suppose that a candidate vector $(z_{il} : i = 1, ..., \hat{M},\ l = l_{min}, ..., l_i)$ is given. It can be verified that for this vector to be feasible, we require

$$\begin{array}{ll} (1) & \sum_{i=1}^{\hat{M}} \Big((1 - z_{il_{min}})(N_{l_{min}} - T) + \sum_{l=l_{min}+1}^{l_i} (1 - z_{il})(N_l - N_{l-1}) \Big) \geq \hat{T} \\ (2) & z_{il} \geq z_{i(l+1)}. \end{array} \tag{4.62}$$

Hence we get the following problem (which is equivalent to (4.60), after considering the transmissions in Round 1),

$$\begin{array}{ll} \max \ \sum_{i=1}^{\hat{M}} \sum_{l=l_{min}}^{l_i} v_{il} z_{il} & \\ \text{subject to} & \\ \sum_{i=1}^{\hat{M}} \sum_{l=l_{min}}^{l_i} (\gamma_l z_{il}) \leq \tilde{T} & \\ z_{il} \in \{0,1\} & \text{for all } i \text{ and } l \\ z_{il} \geq z_{i(l+1)} & \text{for all } i \text{ and } l \end{array} \tag{4.63}$$

where $v_{il} := d_i w_l$, $\gamma_l = 1_{\{l > l_{min}\}}(N_l - N_{l-1}) + 1_{\{l = l_{min}\}}(N_l - T)$ and $\tilde{T} = \sum_{i:l_i \geq l_{min}} (N_{l_i} - T) - \hat{T} = \sum_i e_i^{(k)} - T \geq 0$. We can observe that the above problem is a special form of the *Knapsack problem* with an additional constraint ($z_{il} \geq z_{i(l+1)}$) which essentially requires some order for choosing the items z_{il}.

Lets define $t_i := \sum_{l=l_{min}}^{l_i} (\gamma_l z_{il})$ and suppose $(t_1, ..., t_{\hat{M}})$ is provided such that

$$\sum_{i=1}^{\hat{M}} t_i \leq \tilde{T}$$

and

$$t_i \leq \sum_{l=l_{min}}^{l_i} \gamma_l.$$

Note that considering the constraint $z_{il} \geq z_{i(l+1)}$, we can easily verify that the maximum value ($\sum_{l=l_{min}}^{l_i} v_{il} z_{il}$) that can be achieved, when t_i is given, has the following form

$$V(t_i, i) := \sum_{l=l_{min}}^{l(t_i)} v_{il} = d_i \sum_{l=l_{min}}^{l(t_i)} w_l$$

where $l(t_i) = \max\{l \geq l_{min} - 1 : \sum_{j=l_{min}}^{l} \gamma_l \leq t_i\}$. It can be seen that $V(t_i, i) = \frac{d_i}{d_1} V(t_i, 1)$.

Now it should be straightforward to realize that problem (4.63) can be equivalently expressed as follows,

$$\begin{array}{c} \max \ \sum_{i=1}^{\hat{M}} \frac{d_i}{d_1} V(t_i, 1) \\ \text{subject to} \\ t_i \leq \sum_{l=l_{min}}^{l_i} \gamma_l \quad \text{for all } i \\ \sum_{i=1}^{\hat{M}} t_i \leq \tilde{T} \end{array} \tag{4.64}$$

The above problem can be easily solved using *Dynamic programming* in $O(\hat{M}\tilde{T})$ time.

Algorithm 7 summarizes our optimal P2P scheme which can support any achievable QoS metric, introduced in this subsection.

Algorithm 7 Optimal Robust P2P scheme (infinite field)

At the beginning of each frame $k-1$**, given the arrivals** $(e_1^{(k)}, ..., e_M^{(k)})$ **and the deficit values** $([d_1[k-1]]^+, ..., [d_M[k-1]]^+) = (d_1, ..., d_M)$**:**

1) If $\sum_i e_i^{(k)} \leq T$: let each peer i broadcast his initial $e_i^{(k)}$ coded chunks.

2) If $\sum_i e_i^{(k)} > T$: let $l_i = \max\{l : N_l - T \leq e_i^{(k)}\}$ and each peer i transmit up to $\min\{e_i^{(k)} - (N_{l_i} - T), e_i^{(k)}\}$ chunks.

2.1) If $\sum_i \min\{e_i^{(k)} - (N_{l_i} - T), e_i^{(k)}\} \geq T$: there is no time slot left for further transmissions.

2.2) If $\sum_i \min\{e_i^{(k)} - (N_{l_i} - T), e_i^{(k)}\} < T$: consider peers $\{i : l_i \geq l_{min}\}$, and define

$$\begin{gathered} v_{il} := d_i w_l, \\ \gamma_l = 1_{\{l > l_{min}\}}(N_l - N_{l-1}) + 1_{\{l = l_{min}\}}(N_l - T), \\ \tilde{T} = \sum_i e_i^{(k)} - T, \\ l(t) = \max\{l \geq l_{min} - 1 : \sum_{j=l_{min}}^{l} \gamma_l \leq t\}, \\ V(t,1) := \sum_{l=l_{min}}^{l(t)} v_{1l}. \end{gathered} \tag{4.65}$$

Solve the following dynamic programming to get optimal t_i^* values,

$$\begin{gathered} \max \ \sum_i \frac{d_i}{d_1} V(t_i, 1) \\ \text{subject to} \\ t_i \leq \sum_{l=l_{min}}^{l_i} \gamma_l \quad \text{for all } i \\ \sum_i t_i \leq \tilde{T}, \end{gathered} \tag{4.66}$$

Now let each peer $i \in \{j : l_j \geq l_{min}\}$ broadcast $1_{\{l(t_i^*) = l_{min}-1\}}(N_{l_i} - T) + 1_{\{l(t_i^*) \geq l_{min}\}}(N_{l_i} - N_{l(t_i^*)})$ of his chunks that have not been transmitted yet.

7. Simulation Results

In this section, we evaluate the performance of our proposed B2P-P2P scheme for a number of scenarios.

(1) coding over finite fields versus infinite field: we have already considered this case in Subsection 5. It is depicted in Figure 13 that the proposed P2P scheme, which is shown to be optimal for the infinite field size, still performs quite well in the finite field case. Indeed for a field size of $q = 32$, the degradation the quality experienced by the peers is shown to be only %2.

(2) coding versus non-coding solutions: we utilize random linear network coding in our algorithms to simplify coordinating transmissions. However, it is well understood that in a broadcast system like our P2P network, using network coding can potentially enhance the performance. In our proposed P2P policy, we assumed that the media server transmits randomly coded chunks to the peers via B2P network and peers further disseminate these chunks over the P2P network. We also observed that peers do not need to perform the encoding process and can only forward the received chunks from the B2P network, whenever they get a chance to broadcast.

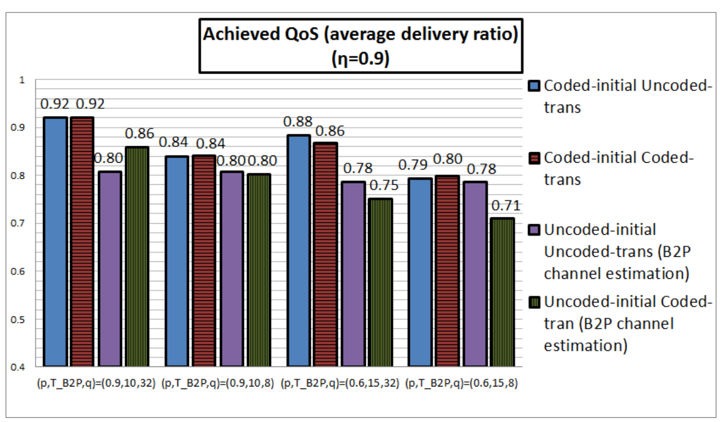


Fig. 15. Benefits of random linear coding at the media server ($N = 20$, $T = 15$, $M = 4$, with symmetric peers)

Figure 15 compares our proposed random linear coding scheme with three other cases for a target $\eta = 0.9$: (i) using random linear coding both in media server and P2P transmissions: it is seen that by performing another layer of encoding at the peer devices, the performance does not change. (ii) no coding: in this scheme the original chunks are transmitted over both B2P and P2P interfaces without any coding. We employ a *Max-Rank-First* scheme to coordinate the P2P transmissions and make sure no chunks is transmitted more than once over the P2P network. Note that for this non-coding scheme, even without receiving all N constituent chunks of a block, peers can partially recover the original block. Hence, we measure the QoS as the average number of chunks received by a peer for each block. For different configurations evaluated in Figure 15, we see that by utilizing coding over chunks, a better performance is achievable. (iii) sending uncoded chunks from the media server and performing random linear coding at P2P transmissions: for this scheme the number of DoFs that a peer can offer to the system is smaller compared to the first scenario, because peers start with uncoded chunks and a pair of peers have a non-negligible chance of having common chunks. Hence, it is seen that, compared to the first two scenarios, the quality will drop significantly for this scheme as well. It should be noted that for this case, like the non-coding case, the original chunks received from the B2P network will be immediately available for playing out. Hence, the QoS metric is defined based on the average number of chunks available for playout at the peer devices at the end of each frame.

Note that in Figure 15 and for the last two scenarios (no coding scheme and the case of coding at peer devices only), we assumed the media server is able to estimate the B2P channel conditions (On or Off) before performing any transmissions. This information can help distributing the initial uncoded chunks to the peers more efficiently. For example, for the last scenario in which we employ coding only at peer

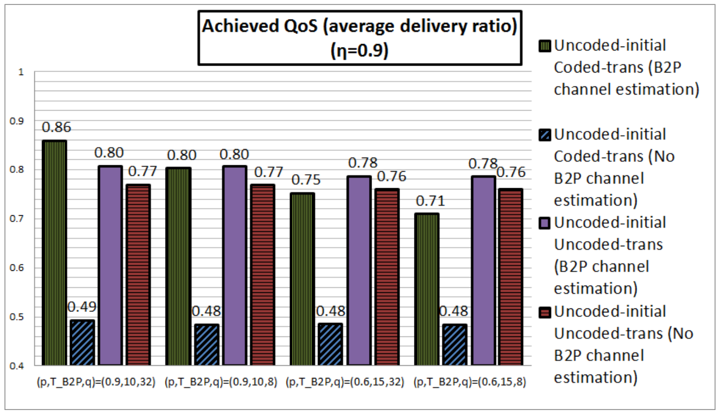


Fig. 16. Effect of B2P channel estimation ($N = 20$, $T = 15$, $M = 4$, with symmetric peers)

devices, we need to make sure that initially all N original chunks are available in the system in order to have useful P2P transmissions. Hence, knowing channels states helps the server to make better decisions regarding what chunks to transmit to each peer. In practice, it may not be possible to achieve such information about the states of B2P channels. In this case, the server may randomly choose some uncoded chunks to transmit to the peers. In Figure 16, the performance of the system is compared when we have complete channel estimation and when such knowledge does not exist. It is seen that the achieved QoS of the latter case is worse than the former and the quality drop for the scheme, in which we perform coding at peer devices, is more significant.

Figure 17 demonstrates the performance of coding and non-coding schemes when the target QoS metric is $\eta = 0.55$. Note that if the quality metric is defined based on the average number of received chunks which are available for playing out, then for some configurations it is better to employ a non-coding scheme. For example for the case shown in this figure, an *MRF* P2P algorithm with no coding employed can deliver a larger number of playable chunks to the peers compared to the scheme which

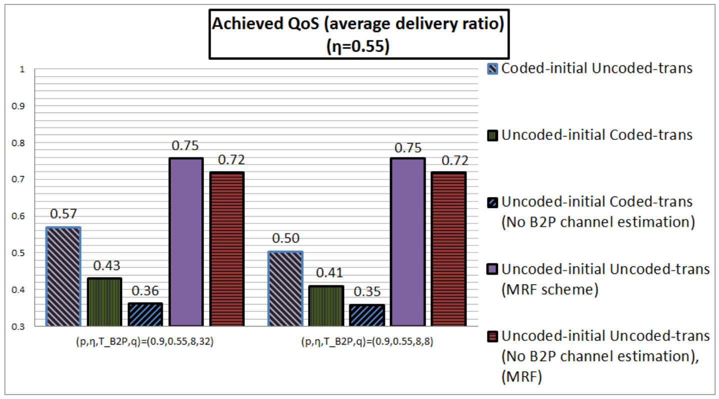


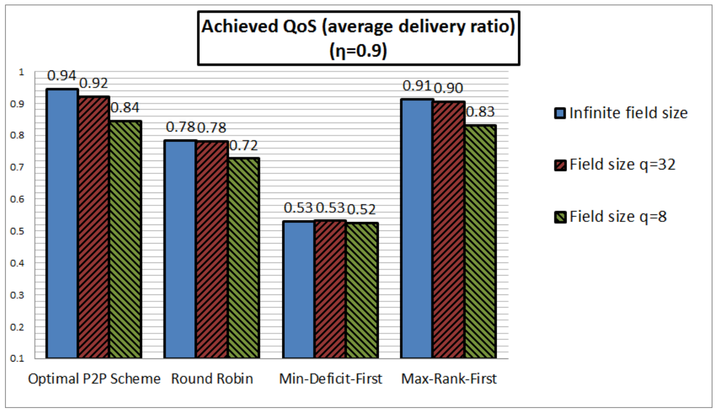
Fig. 17. Non-coding scheme outperforms coding solution for some cases($N = 20$, $T = 15$, $M = 4$, with symmetric peers)

utilizes random coding.

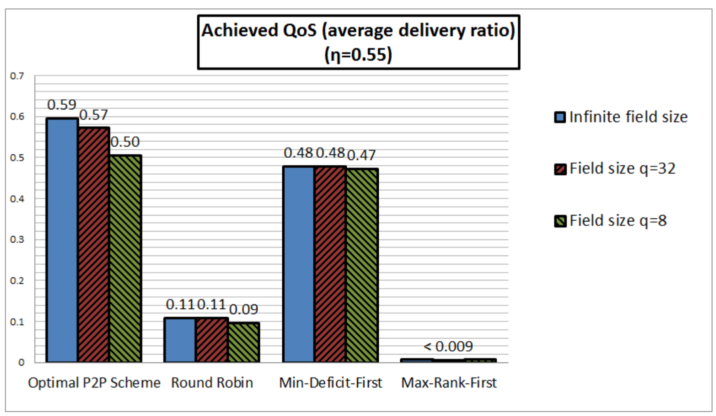
(3) proposed P2P policy versus other greedy policies: we proposed a P2P scheme in Algorithm 6 which consists of three phases and showed that such a simple algorithm is throughput optimal. Figures 18(a) and 18(b) compare the performance of our algorithm with three other algorithms: Round Robin scheme, Min-Deficit-First discipline and Max-Rank-First algorithm. We can observe that none of the two greedy algorithms can achieve the required QoS and our P2P scheme proves the best performance.

Note that in the case of $\eta = 0.55$, the Round Robin algorithm performs much worse. The reason is that peers try to transmit the same number of times during each frame and most often this does not let any of them receive enough chunks to be able to recover the original block.

(4) playout smoothness: note that we model our QoS metric in terms the long-run average number of frames a peer is busy playing out the multimedia content. Both analytically and by simulations, we proved our proposed algorithm is capable of supporting such a QoS metric. However in reality, the average playout time alone cannot completely model the quality a peer experiences. For example assume peer i



(a) $\eta = 0.9$



(b) $\eta = 0.55$

Fig. 18. Comparison of Algorithm 6 with greedy algorithms (Round Robin, Min-Deficit-First, and Max-Rank-First) ($N = 20$, $T = 15$, $M = 4$, with symmetric peers)

requires a minimum delivery ratio $\eta_i = 0.5$. Although our algorithm makes sure that peer i will be able to successfully receive at least 0.5 of the stream, we do not know in what fashion he receives these blocks. As two extreme examples, suppose peer i receives the stream periodically with (1) a very large period and (2) a short period. Which one is more suitable? In case (1), peer i periodically looses a big portion of the stream, while in case (2) the lost blocks are more uniformly distributed over the stream. In practice case (2) is more preferred because by coding over the blocks, peer i will be able to recover the dropped blocks to some extent. Hence we prefer to receive blocks almost periodically with small periods.

We consider four states $s_i[k] \in \{1, 0, -1, -2\}$ for a peer i at the beginning of frame k as follows,

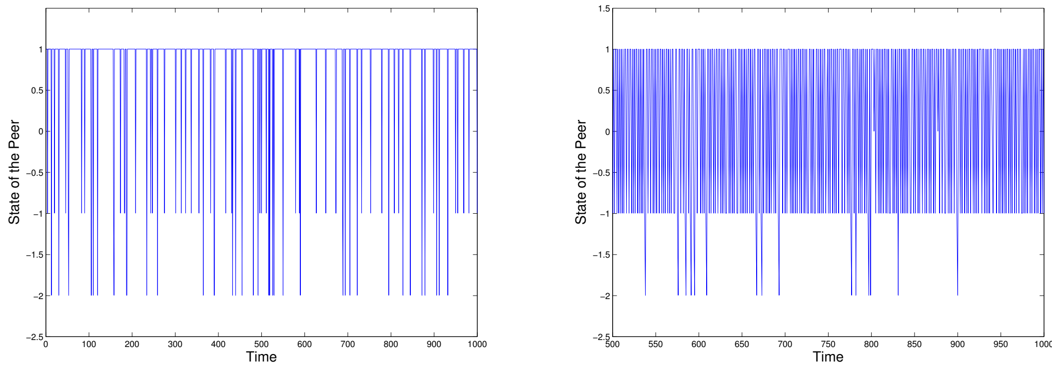
$$s_i[k] = \begin{cases} 1 & \text{if } n_i^{(k)} = N \\ 0 & \text{if } e_i^{(k)} < N - T \\ -1 & \text{if } e_i^{(k)} \geq N - T, \text{ but } \hat{n}_i^{(k)} < N \\ -2 & \text{if } \hat{n}_i^{(k)} \geq N, \text{ but } n_i^{(k)} < N \end{cases} \tag{4.67}$$

Note that only if $s_i[k] = 1$, peer i is able to successfully recover block k. Figures 19(a) and 19(b) display the variation of $s_i[k]$ for two delivery ratios $\eta = 0.9$ and 0.55 respectively.

Consider peer i and let $T_l = j_l - k_l$ for $l \geq 1$, where j_l and k_l are defined as follows

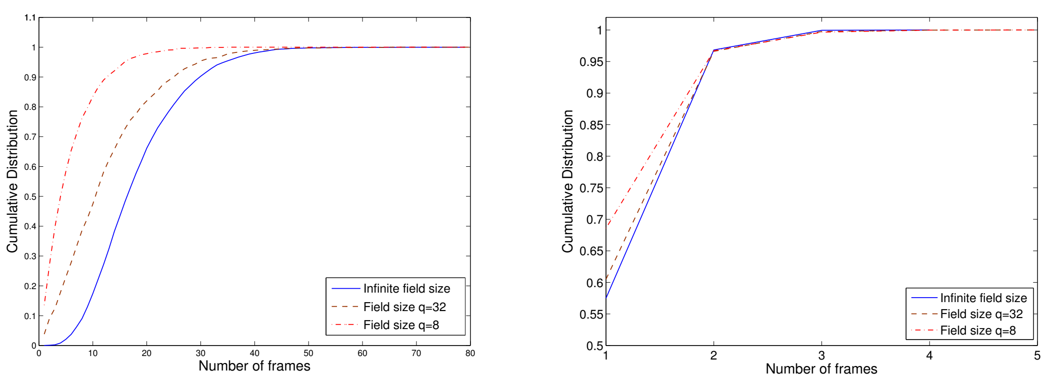
$$\begin{aligned} j_0 &= 0 \\ k_l &= \min\{k \geq j_{l-1} : s_i[k] = 1\} \quad \text{for } l \geq 1 \\ j_l &= \min\{j \geq k_l : s_i[j] = 0\} \quad \text{for } l \geq 1. \end{aligned} \tag{4.68}$$

We call t_l the l^{th} *smooth playout time*, that is the length of the l^{th} longest sequence of frames during which i continuously receives the blocks. As mentioned above, we



(a) $\eta = 0.9$ (b) $\eta = 0.55$

Fig. 19. Variations of the state $s_i[k]$ ($N = 20$, $T = 15$, $M = 4$, $p = 0.9$, $q = 32$, with symmetric peers)



(a) $\eta = 0.9$ (b) $\eta = 0.55$

Fig. 20. Distribution of smooth playout times T_l ($N = 20$, $T = 15$, $M = 4$, $p = 0.9$, $q = 32$, with symmetric peers)

would like the sequence $(T_1, T_2, T_3, ...)$ to have a small variance (that is almost periodic reception) and a small average (to prevent the loss of big portions).

The distribution of the smooth playout time for two different delivery ratios $\eta = 0.9$, and 0.55 is demonstrated in Figures 20(a) and 20(b) respectively. From these plots and the actual state variation shown in Figures 19(a) and 19(b), we observe our algorithm performs desirably, since peer i receives blocks almost periodically with a small variance and no big portion of subsequent blocks is lost.

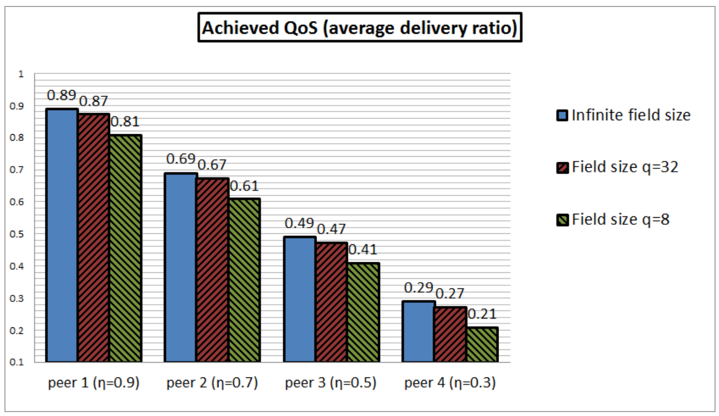
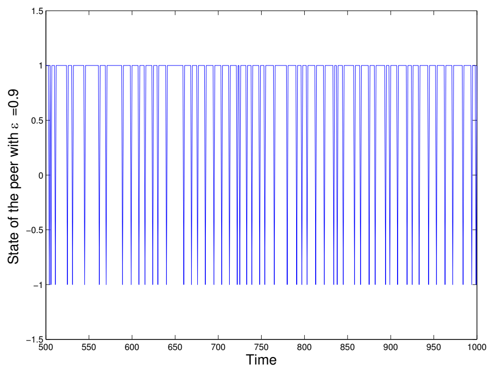
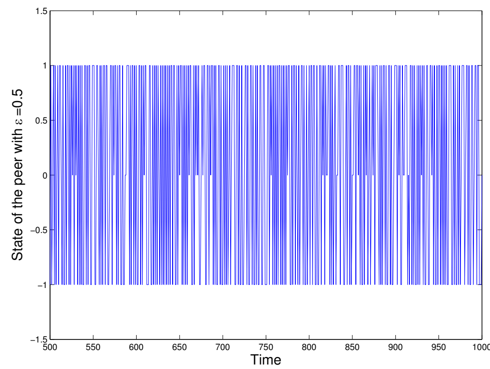


Fig. 21. Performance of Algorithm 6 in finite fields ($N = 20$, $T = 15$, $M = 4$, with different QoS)

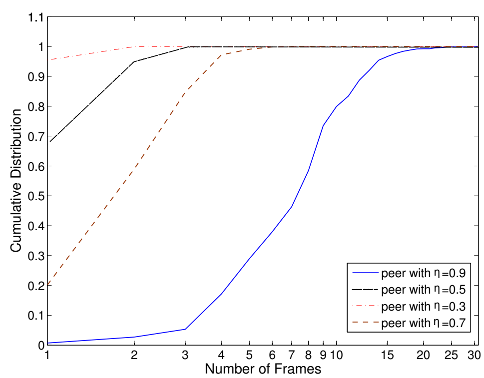
(5) peers with different delivery ratios: all the above plots and simulations have been done for a completely symmetric system in which peers have similar QoS requirements. In this part, we simulate a system with $M = 4$ peers with delivery ratios $(0.9, 0.7, 0.5, 0.3)$ and show all the previous results hold for the asymmetric case as well. Figure 21 displays the performance of our proposed algorithm in the finite fields. It can also be seen in Figures 22(a), 22(b) and 22(c) that the desirable performance of the algorithm in terms of playout smoothness still holds even when we have peers with different requirements.



(a) State variation of the peer with $\eta = 0.9$



(b) State variation of the peer with $\eta = 0.5$



(c) Distribution of smooth playout times

Fig. 22. Playout pattern of the algorithm on an asymmetric system($N = 20$, $T = 15$, $M = 4$, $p = 0.9$, $q = \infty$, $(\eta_1, ..., \eta_4) = (0.9, 0.7, 0.5, 0.3)$)

C. Summary and Future Work

In this chapter, we studied the problems of content distribution and multimedia live streaming in a wireless hybrid network, which consists of an expensive unicast B2P network and a free broadcast P2P network. In the content distribution problem, we defined our QoS metric as a requirement that all the peers successfully receive a single common block within a deadline with a target probability. In the streaming variation, we considered a long sequence of common blocks and defined individual QoS metrics such that each peer individually needs to successfully receive some minimum fraction of the blocks within their deadlines.

We utilized random linear coding over finite fields in our algorithms to simplify coordinating transmissions. In the content distribution problem, we presented a two-phase algorithm. In the first phase, peers receive packets from both B2P and P2P (using *MRF,* which appears naturally from the QoS constraints), while in the second phase, only P2P (using *NmRF*) is used. For the live streaming application, we adopted a new framework in which the decisions on P2P and B2P networks can be decomposed and proposed a three-phase P2P scheme which was derived from a Lyapunov stability argument. We further showed that our framework is capable of handling a general B2P cost criterion as well. We evaluated the performance of these algorithms both analytically and by simulation.

In our model, we assumed that P2P network is fully connected and reliable. Two natural extensions for both problems is to consider lossy P2P transmissions and/or multihop P2P networks. We will shortly comment about the challenges of these extensions. For future work, one can also study *on-demand* streaming applications in which we do not have fixed playout intervals and can buffer the content well in advance. We briefly mentioned the possibility of finding the optimal B2P scheme by applying an iterative search and considering two time scales. It is worthwhile thinking how this algorithm can be made more efficient and whether it can be implemented in a distributed fashion. Finally, these P2P algorithms, which work based on cooperation of the peers, show many interesting aspects from a game theoretic point of view. For example, it is appealing to develop more efficient mechanisms that naturally lead peers to some equilibria at which the QoS metrics are satisfied.

1. Extension to Unreliable P2P Network

An interesting direction for the future work is extending the results of this paper to lossy P2P networks. In this subsection, we shall point out some of the challenges of this extension.

For the case of unreliable wireless channels, one can consider different models. In general we can divide the models, based on the capability of peers to estimate the channels, into two categories: (i) peers can determine the state (ON or OFF) of their channels at each slot, and (ii) peers do not have any information about the realization of their channels and only know some average statistics (*e.g.* the probability of being ON in an *i.i.d.* channel model).

Note that channel estimation requires feedback messages and as indicated in [54], the large overhead of gathering this feedback information makes channel estimation in broadcast networks practically inefficient. However in a very slow fading scenario, where the state of the channels does not change during the transmission interval (T slots), it might be possible to initially determine the realization of the channels. This model will most often reduce to a multihop broadcast network (represented by a partially connected graph). [36] studies this model to find the minimum number of transmissions required to disseminate a common block over a pure P2P network. There, it was shown that finding the optimal set of transmissions is an NP-hard problem in general. With the existence of the B2P network, the problem can only get harder to solve.

In the other model, we assume peers do not have any knowledge about their channels' state at the time of transmission. It should be noticed that mostly acknowledgement messages (ACK) are not available in broadcast networks, hence peers may not realize whether their transmissions are successful even after broadcasting. That means in practice, we cannot keep track of the states of the peers (*i.e.,* the matrices corresponding to the received coded chunks).

The only option left to practically model the system is to assume that there is no feedback or ACK messages and the only available information is the probabilities of different channel states (as in [54]). For this case, despite the model considered in this paper, the peers require to randomly combine all their available chunks at the time of transmission in order to increase the chance of its usability. Therefore clearly the number of transmissions done by a peer can be larger than its B2P receptions and also the order of transmissions is now important. Note that the usefulness probability of a transmission depends on the history of all the previous transmissions. Hence, it is not clear how to efficiently solve for the best sequence of transmissions and this problem remains unanswered.

CHAPTER V

CONCLUSION

In this dissertation, we studied different scenarios in wireless communication networks where delay needs to be taken care of. We considered a wireless content distribution network in which both inelastic requests (requiring service deadlines) and elastic requests (delay tolerant) exist. Optimal algorithms were proposed for this framework, which jointly solve content caching and link scheduling problems.

We also evaluated the effect of delay in relay networks which utilize network coding. Relay nodes can decrease the number of required transmissions for exchanging flow packets by employing network coding. However, they may need to delay the service to the enqueued packets, in order to find coding opportunities. We investigated this trade-off between delay and the number of transmissions, and proposed a simple threshold-based algorithm to optimally balance between latency and efficiency.

Finally, we looked at a broadcast cooperative P2P network and showed how simple (sub)optimal algorithms can be constructed to support delay-sensitive applications. We specifically focused on a content distribution problem and a multimedia live streaming application.

We discussed about several possible extensions to the above problems at the end of each chapter, and occasionally pointed to some challenges and obstacles on these extensions.

The objective of the work presented in this dissertation was first to understand how delay can be studied in a couple of popular wireless networks, and then develop provably optimal algorithms to satisfy the corresponding delay requirements. We tried to concentrate on some instances of the wireless networks and applications that are currently very popular and are predicted to have even more appearance in future.

REFERENCES

[1] Sandvine, "Global internet phenomena report, spring 2011." [Online]. Available: http://www.wired.com/images_blogs/epicenter/2011/05/SandvineGlobalInternetSpringReport2011.pdf

[2] C. Index, "Forecast and methodology, 2010–2015," *White paper, CISCO, June*, 2011.

[3] I. Hou, V. Borkar, and P. Kumar, "A theory of QoS for wireless," in *IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, April 2009.

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

[5] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proc. 43rd IEEE Conference on Decision and Control (CDC 2004)*, Paradise Islands, Bahamas, Dec. 2004.

[6] 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.

[7] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and mac for stability and fairness in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[8] J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," in *IEEE INFOCOM 2010*, San Diego, CA, March 2010.

[9] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-aware caching and traffic management in content distribution networks," in *Proc. of IEEE Infocom*, Shanghai, China, 2011.

[10] M. J. Neely, "Optimal backpressure routing for wireless networks with multi-receiver diversity," in *Proc. Conf. on Information Sciences and Systems (CISS)*, March 2006, pp. 18–25.

[11] M. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Information Theory*, vol. 52, no. 2, pp. 2915–2934, July 2006.

[12] F. Foster, "On the stochastic matrices associated with certain queuing processes," *The Annals of Mathematical Statistics*, vol. 24, no. 3, pp. 355–360, 1953.

[13] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[14] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," *Proc of IEEE ITW*, pp. 62–66, 2006.

[15] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[16] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 243–254, 2006.

[17] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *Proc. of ACM SIGCOMM*, pp. 169–180, 2007.

[18] Y. Sagduyu and A. Ephremides, "Cross-layer optimization of mac and network coding in wireless queueing tandem networks," *Information Theory, IEEE Transactions on*, vol. 54, no. 2, pp. 554–571, 2008.

[19] A. Khreishah, C. Wang, and N. Shroff, "Cross-layer optimization for wireless multihop networks with pairwise intersession network coding," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 5, pp. 606–621, 2009.

[20] V. Reddy, S. Shakkottai, A. Sprintson, and N. Gautam, "Multipath wireless network coding: a population game perspective," in *Proc. of INFOCOM*, San Diego, CA, March 2010.

[21] A. Eryilmaz, D. Lun, and B. Swapna, "Control of multi-hop communication networks for inter-session network coding," *Information Theory, IEEE Transactions on*, vol. 57, no. 2, pp. 1092–1110, 2011.

[22] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *Information Theory, IEEE Transactions on*, vol. 55, no. 2, pp. 797–815, 2009.

[23] Y. Xi and E. Yeh, "Distributed algorithms for minimum cost multicast with network coding," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 2, pp. 379–392, 2010.

[24] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* New York, NY, USA: John Wiley & Sons, Inc., 1994.

[25] S. Ross, *Introduction to stochastic dynamic programming: Probability and mathematical.* Orlando, FL, USA: Academic Press, Inc., 1983.

[26] L. I. Sennott, *Stochastic Dynamic Programming and the Control of Queueing Systems.* New York, NY, USA: Wiley-Interscience, 1999.

[27] D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[28] L. I. Sennott, "Average Cost Optimal Stationary Policies in Infinite State Markov Decision Processes with Unbounded Costs," *Operations Research*, vol. 37, pp. 626–633, 1989.

[29] G. Koole, *Monotonicity in Markov Reward and Decision Chains: Theory and Applications.* Hanover, MA, USA: Now Publishers Inc, 2007.

[30] C. Labovitz, D. McPherson, and S. Iekel-Johnson, "2009 Internet Observatory report," in *NANOG-47*, October 2009.

[31] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashlinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," in *Proc. of Allerton*, 2010.

[32] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2486–2507, 2006.

[33] A. ParandehGheibi, M. Medard, A. Ozdaglar, and S. Shakkottai, "Avoiding interruptions-a QoE reliability function for streaming media applications," *IEEE Journal on Selected Areas in Communications*, To appear in 2011.

[34] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *Proc. of IEEE ISIT*, 2010, pp. 1888–1892.

[35] N. Milosavljevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramchandran, "An optimal divide-and-conquer solution to the linear data exchange problem," in *Proc. of IEEE ISIT*, 2011.

[36] T. Courtade and R. Wesel, "Coded cooperative data exchange in multihop networks," *Arxiv preprint arXiv:1203.3445v1*, 2012.

[37] M. Wang and B. Li, "Lava: A reality check of network coding in peer-to-peer live streaming," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE.* IEEE, 2007, pp. 1082–1090.

[38] X. Zhang, J. Liu, B. Li, and Y. Yum, "Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. Ieee, 2005, pp. 2102–2111.

[39] "PPLive," http://www.pplive.com/, 2011.

[40] G. Huang, "Pplive: A practical p2p live system with huge amount of users," in *Proc. of the ACM SIGCOMM Workshop on Peer-to-Peer Streaming and IPTV Workshop*, 2007.

[41] Z. Liu, C. Wu, B. Li, and S. Zhao, "Uusee: Large-scale operational on-demand streaming with random network coding," in *INFOCOM, 2010 Proceedings IEEE.* Ieee, 2010, pp. 1–9.

[42] H. Seferoglu, L. Keller, B. Cici, A. Le, and A. Markopoulou, "Cooperative video streaming on smartphones," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on.* IEEE, 2011, pp. 220–227.

[43] Y. P. Zhou, D. M. Chiu, and J. C. S. Lui, "A simple model for analyzing P2P streaming protocols," in *Proc. of IEEE ICNP 2007*, Beijing, China, October 2007.

[44] S. Shakkottai, R. Srikant, and L. Ying, "The asymptotic behavior of minimum buffer size requirements in large P2P streaming networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, May 2011.

[45] M. Ra, J. Paek, A. Sharma, R. Govindan, M. Krieger, and M. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proc. of ACM MobiSys*, 2010, pp. 255–270.

[46] L. Jiang and J. Walrand, "Distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proc. Ann. Allerton Conf. Communication, Control and Computing*, 2008.

[47] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[48] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *Proc. of IEEE INFOCOM*, Miami, FL, 2005.

[49] G. Veciana and X. Yang, "Fairness, incentives and performance in peer-to-peer

networks," in *Proc. Forty-First Annual Allerton Conference on Control, Communications and Computing*, Monticello, IL, USA, October 2003.

[50] V. Goyal, "Multiple description coding: Compression meets the network," *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 74–93, 2001.

[51] A. Ramasubramonian and J. Woods, "Multiple description coding and practical network coding for video multicast," *Signal Processing Letters, IEEE*, vol. 17, no. 3, pp. 265–268, 2010.

[52] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *Information Theory, IEEE Transactions on*, vol. 42, no. 6, pp. 1737–1744, 1996.

[53] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Annual Allerton Conference on Communication, Control, and Computing*, 2003.

[54] P. K. I-H. Hou, "Broadcasting delay-constrained trafc over unreliable wireless links with network coding," in *Proc. of MobiHoc 2011*, 2011.

APPENDIX A

OPTIMAL DELAY-THROUGHPUT TRADE-OFF IN THE NETWORK CODING-ENABLED RELAY NETWORKS

Proof of Theorem 6

In the proof of this theorem, we use the following result which enhances applying Lemma 7.

Lemma 20 (Proposition 5 [28])**.** *Assume there exists a stationary policy θ inducing an irreducible and ergodic Markov chain with the following properties: there exists a nonnegative function $F(i,j)$ and a finite nonempty subset G with $G \subseteq \mathbb{R}^2$ such that for $(i,j) \in \mathbb{R}^2 - G$,*

$$\sum_{k,l} P_\theta((i,j),(k,l))F(k,l) - F(i,j) \leq -C((i,j),\theta), \tag{A.1}$$

and $\sum_{k,l} P_\theta((i,j),(k,l))F(k,l) < \infty$ for $(i,j) \in G$. Then the condition (iii) in Lemma 7 holds.

As described earlier it is sufficient to show that the three conditions in Lemma 7 are met. Proposition 5 results in the condition (i), while the condition (ii) is satisfied based on Lemma 6 (e.g. set $N = 0$). Three cases are considered to show that the condition (iii) holds.

Case (1) $p_0^{(i)} + p_1^{(i)} < 1$ for $i = 1,2$:

Let $F(i,j) = B(i^2 + j^2)$ for some positive B in Lemma 20. By $\tilde{\theta}$ we denote the stationary policy of always transmitting, that induces an irreducible and ergodic

Markov chain from MDP. Then, for all state $(i,j) \in \mathbb{R}^2 - \{(0,0),(0,1),(1,0)\}$, we can compute that

$$\begin{aligned}
&\sum_{k,l} P_{\tilde{\theta}}\left((i,j),(k,l)\right)\left[F(k,l) - F(i,j)\right] \\
=&\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} P_1\left((i,j),([i-1]^+ + r,[j-1]^+ + s)\right)\left[F([i-1]^+ + r,[j-1]^+ + s) - F(i,j)\right] \\
=&\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} p_r^{(1)} p_s^{(2)} B\left[2i(r-1) + (r-1)^2 + 2j(s-1) + (s-2)^2\right] \\
=&2B\Big(i(\mathbb{E}[\mathcal{A}_1]-1) + j(\mathbb{E}[\mathcal{A}_2]-1)\Big) + B\Big(\mathbb{E}[(\mathcal{A}_1-1)^2] + \mathbb{E}[(\mathcal{A}_2-1)^2]\Big). \qquad \text{(A.2)}
\end{aligned}$$

Note that $\mathbb{E}[\mathcal{A}_i] < 1$, so $2B(\mathbb{E}[\mathcal{A}_i]-1) < -C_H$ for sufficiently large B. Moreover, since $\mathbb{E}[\mathcal{A}_i^2] < \infty$ for $i \in \{1,2\}$, (A.1) can be guaranteed. That is

$$\sum_{k,l} P_{\tilde{\theta}}\left((i,j),(k,l)\right)\left[F(k,l) - F(i,j)\right] \le -C((i,j),\tilde{\theta}),$$

when i,j are large enough, where $C((i,j),\tilde{\theta}) = C_H([i-1]^+ + [j-1]^+) + C_T$. Accordingly, we define G as a finite set such that for $(i,j) \in \mathbb{R}^2 - G$, the above inequality is satisfied. We also let G include the states $\{(0,0),(0,1),(1,0)\}$.

Finally, for $(i,j) \in G$,

$$\begin{aligned}
&\sum_{k,l} P_{\tilde{\theta}}\left((i,j),(k,l)\right) F(k,l) \\
=&B\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} p_r^{(1)} p_s^{(2)}\left[([i-1]^+ + r)^2 + ([j-1]^+ + s)^2\right] \\
=&B\Big\{(i-1)^2 + 2[i-1]^+\mathbb{E}[\mathcal{A}_1] + \mathbb{E}[\mathcal{A}_1^2] + (j-1)^2 + 2[j-1]^+\mathbb{E}[\mathcal{A}_2] + \mathbb{E}[\mathcal{A}_1^2]\Big\} < \infty,
\end{aligned}$$

since all the terms are finite. Therefore, the condition (iii) in Lemma 7 is verified according to Lemma 20.

Case (2) $p_0^{(1)} + p_1^{(1)} = 1$ and $p_0^{(2)} + p_1^{(2)} < 1$:

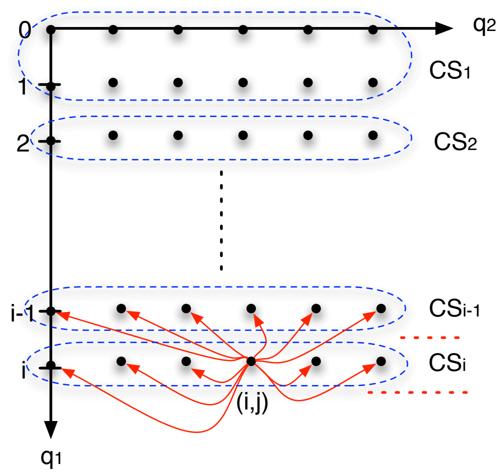


Fig. 23. In case (iii), state (i, j) can only transit to states in the communicating classes CS_i and CS_{i-1}

We shall see that under $\tilde{\theta}$ in case (1), the Markov chain is not irreducible and we cannot use Lemma 20, hence we will directly verify the condition (iii) in Lemma 7.

Consider Figure 23 and define sets $CS_1 = \{(a, b) : a = 0, 1 \text{ and } b \in \mathbb{N} \cup \{0\}\}$ and $CS_i = \{(a, b) : a = i, b \in \mathbb{N} \cup \{0\}\}$ for $i \geq 2$. Then all CS_i are communicating classes under policy $\tilde{\theta}$. The states in CS_1 can be shown to be positive-recurrent, but in CS_i with $i \geq 2$ are transient. For $i \geq 2$, let $\overline{C}_{i,j}$ be the expected cost for a passage from state (i, j) (in class CS_i) to the next class CS_{i-1}. Note that state (i, j) has the probability of $p_0^{(1)}$ to escape to class CS_{i-1} and $p_1^{(1)}$ to remain in class CS_i. By considering all the possible paths to escape from state (i, j), we can compute $\overline{C}_{i,j}$ as following, where $C(i, Q_t^{(2)}) = C_T + C_H([i-1]^+ + [Q_t^{(2)} - 1]^+)$,

$$\begin{aligned}
\overline{C}_{i,j} &= \mathbb{E}\left[\sum_{k=0}^{\infty}(p_1^{(1)})^k p_0^{(1)} \sum_{t=0}^{k} C(i, Q_t^{(2)}) | (Q_0^{(1)}, Q_0^{(2)}) = (i, j)\right] \\
&= p_0^{(1)} \mathbb{E}\left[\sum_{t=0}^{\infty} C(i, Q_t^{(2)}) \sum_{k=t}^{\infty} (p_1^{(1)})^k | (Q_0^{(1)}, Q_0^{(2)}) = (i, j)\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty}(p_1^{(1)})^t C(i, Q_t^{(2)}) | (Q_0^{(1)}, Q_0^{(2)}) = (i, j)\right] \qquad \text{(A.3)}
\end{aligned}$$

Hence, $\overline{C}_{i,j}$ can be viewed as the total expected $p_1^{(1)}$-discounted cost of the system, i.e. one can imagine that there is the arrival process $\mathcal{A}_1$ with $\mathbb{P}(\mathcal{A}_1 = 1) = 1$ to q_1, the same process $\mathcal{A}_2$ to q_2, and the relay always transmits. Therefore, $\overline{C}_{i,j} < \infty$ with the similar proof to Proposition 5. By $\overline{C}_{(i,j),(k,l)}$ we denote the expected cost of a first passage from state (i,j) to (k,l). Note that $\overline{C}_{(1,j),(0,0)} < \infty$ for any j by using the Proposition 4 in [28] (intuitively, the expected traveling time from state $(1,j)$ to $(0,0)$ is finite due to the positive recurrence of CS_1). Let $T_0 = \min\{t \geq 1 : (Q_t^{(1)}, Q_t^{(2)}) = (0,0)\}$ and for $i \geq 1$, $T_i = \min\{t \geq 1 : Q_t^{(1)} = i\}$ with the corresponding state $(Q_{T_i}^{(1)}, Q_{T_i}^{(2)}) = (i, \tilde{j}_i)$. Then $\overline{C}_{(i,j),(0,0)} < \infty$, since

$$\overline{C}_{(i,j),(0,0)} = \overline{C}_{i,j} + \sum_{k=1}^{i-2} \overline{C}_{i-k,\tilde{j}_{i-k}} + \overline{C}_{(1,\tilde{j}_1),(0,0)}. \tag{A.4}$$

By $\widehat{\theta}$, we define a policy of always transmitting until T_0 after which the α-optimal policy is employed. $V_\alpha(i,j)$ can be bounded by the discounted cost incurred by $\widehat{\theta}$,

$$\begin{aligned}
&V_\alpha(i,j) \leq \\
&\mathbb{E}_{\widehat{\theta}}\left[\sum_{t=0}^{T_{i-1}-1} \alpha^t C(Q_t, A_t) | Q_0 = (i,j)\right] + \sum_{k=1}^{i-2} \mathbb{E}_{\widehat{\theta}}\left[\sum_{t=T_{i-k}}^{T_{i-k-1}-1} \alpha^t C(Q_t, A_t) | Q_0 = (i,j)\right] + \\
&\mathbb{E}_{\widehat{\theta}}\left[\sum_{t=T_1}^{T_0-1} \alpha^t C(Q_t, A_t) | Q_0 = (i,j)\right] + \mathbb{E}_{\widehat{\theta}}\left[\sum_{t=T_0}^{\infty} \alpha^t C(Q_t, A_t) | Q_0 = (i,j)\right] \\
&\leq \overline{C}_{(i,j),(0,0)} + V_\alpha(0,0)
\end{aligned}$$

Let $M_{i,j} = \overline{C}_{(i,j),(0,0)}$ in Lemma 7. Then $M_{i,j} < \infty$ and $v_\alpha(i,j) = V_\alpha(i,j) - V_\alpha(0,0) \leq M_{i,j}$. Moreover,

$$\sum_{k,l} P_1\big((i,j),(k,l)\big) M_{k,l} = \sum_{k,l} P_1\big((i,j),(k,l)\big) \overline{C}_{(k,l),(0,0)} \leq \overline{C}_{(i,j),(0,0)} < \infty.$$

Therefore, the condition (iii) of Lemma 7 is met.

Case (3) $p_0^{(i)} + p_1^{(i)} = 1$ for $i = 1, 2$ (i.e. Bernoulli arrivals to both queues): Note that $\tilde{\theta}$ in *case*(1) cannot define an irreducible Markov chain for this case either.

By a similar argument to case (2), we can define $M_{i,j} = \overline{C}_{(i,j),(0,0)}$, and show that $\overline{C}_{(i,j),(0,0)}$ is finite for this case. It is true since there is a finite expected cost for state (i,j) to escape to $\{(i-1,j-1),(i-1,j),(i,j-1)\}$.

Proof of Lemma 10

Note that in the proof of Lemma 6, we have shown $V_{\alpha,n+1}(i,j)$ is non-decreasing, if so is $V_{\alpha,n}(i,j)$. Therefore, we only need to show the submodularity and the subconvexity properties of $V_{\alpha,n+1}(i,j)$.

In what follows, we first prove the submodularity part. The subconvexity property will be shown subsequently.

Submodularity

We intend to show that for all $i,j \in \mathbb{N} \cup \{0\}$,

$$V_{\alpha,n+1}(i+1,j+1) - V_{\alpha,n+1}(i+1,j) \leq V_{\alpha,n+1}(i,j+1) - V_{\alpha,n+1}(i,j).$$

According to Lemma 9, we are only interested in 6 cases of $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i,j+1}, a^*_{i+1,j+1})$, where $a^*_{i,j} = \min\{a : \arg\min_{a\in\{0,1\}} \mathcal{V}_{\alpha,n}(i,j,a)\}$.

Case (i) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i,j+1}, a^*_{i+1,j+1}) = (1,1,1,1)$*:* We claim that

$$\begin{aligned}&\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)] \\ &\leq \mathbb{E}[V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)].\end{aligned} \tag{A.5}$$

When $i,j \neq 0$, it is true according to submodularity of $V_{\alpha,n}(i,j)$. Otherwise, both sides are 0.

Case (ii) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i,j+1}, a^*_{i+1,j+1}) = (0,0,0,0)$*:* We claim that

$$\begin{aligned}&\mathbb{E}[V_{\alpha,n}(i+1+\mathcal{A}_1, j+1+\mathcal{A}_2) - V_{\alpha,n}(i+1+\mathcal{A}_1, j+\mathcal{A}_2)]\\&\leq \mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+1+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)].\end{aligned} \tag{A.6}$$

This is obvious from the submodularity of $V_{\alpha,n}(i,j)$.

Case (iii) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i,j+1}, a^*_{i+1,j+1}) = (0,0,0,1)$*:* We claim that

$$\begin{aligned}&C_T - C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+1+\mathcal{A}_1, j+\mathcal{A}_2)]\\&\leq C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+1+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)].\end{aligned} \tag{A.7}$$

From the submodularity of $V_{\alpha,n}(i,j)$, we have

$$\begin{aligned}&V_{\alpha,n}(i,j) - V_{\alpha,n}(i+1,j) + V_{\alpha,n}(i,j) - V_{\alpha,n}(i,j+1)\\&\leq V_{\alpha,n}(i,j) - V_{\alpha,n}(i+1,j) + V_{\alpha,n}(i+1,j) - V_{\alpha,n}(i+1,j+1)\\&= V_{\alpha,n}(i,j) - V_{\alpha,n}(i+1,j+1).\end{aligned} \tag{A.8}$$

Since $a^*_{i+1,j+1} = 1$, we have $\Delta\mathcal{V}_{\alpha,n}(i+1,j+1) \leq 0$, *i.e.*,

$$C_T - 2C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+1+\mathcal{A}_1, j+1+\mathcal{A}_2)] \leq 0. \tag{A.9}$$

The claim is true because

$$\begin{aligned}&C_T - 2C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+1+\mathcal{A}_1, j+\mathcal{A}_2)\\&+V_n(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+1+\mathcal{A}_2)]\\&\leq \Delta\mathcal{V}_{\alpha,n}(i+1,j+1) \leq 0.\end{aligned} \tag{A.10}$$

Case (iv) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i,j+1}, a^*_{i+1,j+1}) = (0,0,1,1)$*:* We claim that

$$\begin{aligned}&-C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+1+\mathcal{A}_1, j+\mathcal{A}_2)]\\&\leq C_H([i-1]^+ - i) + \alpha\mathbb{E}[V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)].\end{aligned} \tag{A.11}$$

When $i \neq 0$, it is satisfied because $V_{\alpha,n}(i,j)$ is convex. Otherwise, it is true since $V_{\alpha,n}(i,j)$ is non-decreasing.

Case (v) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i,j+1}, a^*_{i+1,j+1}) = (0,1,0,1)$*:* We claim that

$$\begin{aligned}&C_H(j-[j-1]^+) + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)] \\ &\le C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+1+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)].\end{aligned} \tag{A.12}$$

When $j \neq 0$, it holds since $V_{\alpha,n}(i,j)$ is convex. It is true for other cases because of the non-decreasing property of $V_{\alpha,n}(i,j)$.

Case (vi) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i,j+1}, a^*_{i+1,j+1}) = (0,1,1,1)$*:* We claim that

$$\begin{aligned}&C_H(j-[j-1]^+) + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)] \\ &\le C_T + C_H([i-1]^+ - i) + \alpha\mathbb{E}[V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)].\end{aligned} \tag{A.13}$$

Based on the submodularity of $V_{\alpha,n}(i,j)$, we have

$$\begin{aligned}&V_{\alpha,n}([i-1]^+, j) - V_{\alpha,n}(i,j) + V_{\alpha,n}(i, [j-1]^+) - V_{\alpha,n}(i,j) \\ &\ge V_{\alpha,n}([i-1]^+, [j-1]^+) - V_{\alpha,n}(i, [j-1]^+) + V_{\alpha,n}(i, [j-1]^+) - V_{\alpha,n}(i,j) \\ &= V_{\alpha,n}([i-1]^+, [j-1]^+) - V_{\alpha,n}(i,j).\end{aligned} \tag{A.14}$$

It is noted that $a^*_{i,j} = 0$ and hence $\Delta\mathcal{V}_{\alpha,n}(i,j) \ge 0$, *i.e.*,

$$\begin{aligned}&C_T + C_H([i-1]^+ + [j-1]^+ - i - j) + \\ &\alpha\mathbb{E}[V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, [j-1]^+ + \mathcal{A}_1) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_1)] \ge 0.\end{aligned} \tag{A.15}$$

Therefore, it can be acquired that

$$\begin{aligned}&C_T + C_H([i-1]^+ + [j-1]^+ - i - j) + \alpha\mathbb{E}[V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, j+\mathcal{A}_2) - \\ &V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) + V_{\alpha,n}(i+\mathcal{A}_1, [j-1]^+ + \mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)] \\ &\ge \Delta\mathcal{V}_{\alpha,n}(i,j) \ge 0.\end{aligned} \tag{A.16}$$

Subconvexity

We want to show that for all i and j,

$$V_{\alpha,n+1}(i+1, j+1) - V_{\alpha,n+1}(i,j) \le V_{\alpha,n+1}(i+2, j+1) - V_{\alpha,n+1}(i+1, j).$$

There will be only 5 cases of $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i+1,j+1}, a^*_{i+2,j+1})$ that need to be considered.

Case (i) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i+1,j+1}, a^*_{i+2,j+1}) = (1,1,1,1)$*:* We claim that

$$\begin{aligned} &C_H(i-[i-1]^+) + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}([i-1]^+ + \mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)] \\ &\le C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+1+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)]. \end{aligned} \tag{A.17}$$

When $i, j \neq 0$, it is true according to the subconvexity of $V_{\alpha,n}(i,j)$. The argument is satisfied for $i = 0, j \neq 0$ due to $V_{\alpha,n}(i,j)$ being non-decreasing, and for the case $i \neq 0, j = 0$ due to the convexity of $V_{\alpha,n}(i,j)$. Otherwise, it holds according to the non-decreasing property.

Case (ii) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i+1,j+1}, a^*_{i+2,j+1}) = (0,0,0,0)$*:* We claim that

$$\begin{aligned} &\mathbb{E}[V_{\alpha,n}(i+1+\mathcal{A}_1, j+1+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)] \\ &\le \mathbb{E}[V_{\alpha,n}(i+2+\mathcal{A}_1, j+1+\mathcal{A}_2) - V_{\alpha,n}(i+1+\mathcal{A}_1, j+\mathcal{A}_2)]. \end{aligned} \tag{A.18}$$

This is obvious from the subconvexity of $V_{\alpha,n}(i,j)$.

Case (iii) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i+1,j+1}, a^*_{i+2,j+1}) = (0,0,0,1)$*:* We claim that

$$2C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+1+\mathcal{A}_1, j+1+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2)] \le C_T. \tag{A.19}$$

Since $a^*_{i+1,j+1} = 0$, we have $\Delta\mathcal{V}_{\alpha,n}(i+1, j+1) \ge 0$, *i.e.*,

$$C_T - 2C_H + \alpha\mathbb{E}[V_{\alpha,n}(i+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+1+\mathcal{A}_1, j+1+\mathcal{A}_2)] \ge 0. \tag{A.20}$$

Hence the claim is verified.

Case (iv) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i+1,j+1}, a^*_{i+2,j+1}) = (0,0,1,1)$*:* It is trivial, since both sides are equal to zero.

Case (v) $(a^*_{i,j}, a^*_{i+1,j}, a^*_{i+1,j+1}, a^*_{i+2,j+1}) = (0,1,1,1)$*:* We claim that

$$\begin{aligned} &C_T \le C_H(1+j-[j-1]^+) \\ &+\alpha\mathbb{E}[V_{\alpha,n}(i+1+\mathcal{A}_1, j+\mathcal{A}_2) - V_{\alpha,n}(i+\mathcal{A}_1, [j-1]^+ + \mathcal{A}_2)]. \end{aligned} \tag{A.21}$$

Notice that $a^*_{i+1,j} = 1$, so $\Delta\mathcal{V}_{\alpha,n}(i+1,j) \leq 0$, *i.e.*,

$$\begin{aligned}&C_T - C_H(1 + j - [j-1]^+) + \\ &\alpha\mathbb{E}[V_{\alpha,n}(i + \mathcal{A}_1, [j-1]^+ + \mathcal{A}_2) - V_{\alpha,n}(i + 1 + \mathcal{A}_1, j + \mathcal{A}_2)] \leq 0.\end{aligned} \tag{A.22}$$

Therefore, the argument is confirmed.

Proof of Theorem 10

The proof is based on Foster-Lyapunov criterion [12] associated with the Lyapunov function $\mathcal{L}(x,y) = x^2 + y^2$. Notice that

$$Q^{(i)}_{t+1} = [Q^{(i)}_t - A_t]^+ + \mathcal{A}_i = Q^{(i)}_t - A_t + U^{(i)}_t + \mathcal{A}_i, \tag{A.23}$$

where,

$$U^{(i)}_t = \begin{cases} 0 & \text{if } Q^{(i)}_t - A_t \geq 0 \\ 1 & \text{if } Q^{(i)}_t - A_t = -1. \end{cases} \tag{A.24}$$

Then it can be observed that

$$
\begin{aligned}
&\mathbb{E}\left[\mathcal{L}(Q_{t+1}^{(1)}, Q_{t+1}^{(2)}) - \mathcal{L}(Q_t^{(1)}, Q_t^{(2)}) | Q_t^{(1)} = x, Q_t^{(2)} = y\right] \\
&= \mathbb{E}\left[\textstyle\sum_{i=1}^{2} (Q_t^{(i)} - A_t + U_t^{(i)} + \mathcal{A}_i)^2 | Q_t^{(1)} = x, Q_t^{(2)} = y\right] - (x^2 + y^2) \\
&= \sum_{i=1}^{2} \mathbb{E}\left[(Q_t^{(i)} - A_t + \mathcal{A}_i)^2 | Q_t^{(1)} = x, Q_t^{(2)} = y\right] + \sum_{i=1}^{2} \mathbb{E}\left[(U_t^{(i)})^2 | Q_t^{(1)} = x, Q_t^{(2)} = y\right] \\
&+ \sum_{i=1}^{2} \mathbb{E}\left[2U_t^{(i)}(Q_t^{(i)} - A_t + \mathcal{A}_i) | Q_t^{(1)} = x, Q_t^{(2)} = y\right] - (x^2 + y^2) \\
&\overset{(a)}{\leq} \sum_{i=1}^{2} \mathbb{E}\left[(Q_t^{(i)} - A_t + \mathcal{A}_i)^2 | Q_t^{(1)} = x, Q_t^{(2)} = y\right] + 2 + 2\mathbb{E}[\mathcal{A}_1] + 2\mathbb{E}[\mathcal{A}_2] - (x^2 + y^2) \\
&= 2x\mathbb{E}\left[\mathcal{A}_1 - A_t | Q_t^{(1)} = x, Q_t^{(2)} = y\right] + 2y\mathbb{E}\left[\mathcal{A}_2 - A_t | Q_t^{(1)} = x, Q_t^{(2)} = y\right] \\
&+ \mathbb{E}\left[(\mathcal{A}_1 - A_t)^2 | Q_t^{(1)} = x, Q_t^{(2)} = y\right] + \mathbb{E}\left[(\mathcal{A}_2 - A_t)^2 | Q_t^{(1)} = x, Q_t^{(2)} = y\right] \\
&+ 2\mathbb{E}[\mathcal{A}_1] + 2\mathbb{E}[\mathcal{A}_2] + 2 \\
&\overset{(b)}{\leq} 2x\mathbb{E}\left[\mathcal{A}_1 - A_t | Q_t^{(1)} = x, Q_t^{(2)} = y\right] + 2y\mathbb{E}\left[\mathcal{A}_2 - A_t | Q_t^{(1)} = x, Q_t^{(2)} = y\right] \\
&+ \mathbb{E}[\mathcal{A}_1^2] + 1 + \mathbb{E}[\mathcal{A}_2^2] + 1 + 2\mathbb{E}[\mathcal{A}_1] + 2\mathbb{E}[\mathcal{A}_2] + 2 \\
&= \mathbb{E}[\mathcal{A}_1^2] + \mathbb{E}[\mathcal{A}_2^2] + 2\mathbb{E}[\mathcal{A}_1] + 2\mathbb{E}[\mathcal{A}_2] + 4 \\
&+ \begin{cases} 2x(\mathbb{E}[\mathcal{A}_1] - 1) + 2y(\mathbb{E}[\mathcal{A}_2] - 1) & \text{if } (x, y) \in \mathcal{B}^c \\ 2x\mathbb{E}[\mathcal{A}_1] + 2y\mathbb{E}[\mathcal{A}_2] & \text{if } (x, y) \in \mathcal{B} \end{cases}
\end{aligned}
$$

where $\mathcal{B} = \{(x, y) : (x = 0, y \leq L_2) \text{ or } (x \leq L_1, y = 0)\}$. The inequality (a) comes from $(U_t^{(i)})^2 \leq 1$ and $(Q_t^{(i)} - A_t)U_t^{(i)} \leq 0$, while $\mathbb{E}[A_t] \leq 1$ results in (b). Since $\mathbb{E}[\mathcal{A}_i^2] < \infty$ and $\mathbb{E}[\mathcal{A}_i] < 1$ for $i \in \{1, 2\}$, the result follows from Forster-Lyapunov theorem.

APPENDIX B

WIRELESS BROADCAST P2P NETWORKS

Result on Longest-Queue-First (LQF) Algorithm

Consider a system of M queues with *i.i.d* Bernoulli arrivals of rate p packets per time slot, which are commonly served by a single server with a service rate of 1 packet/slot. The length of the i^{th} queue at the beginning of time slot t is denoted by $b_i[t]$, and $B[t] = (b_1[t], ..., b_M[t])$ is the state of the system at this time. We also use $A[t] = (a_1[t], ..., a_M[t])$ to denote the arrivals to the queues during time slot t.

It is obvious that the *LQF* algorithm, which schedules the longest queue at each time, is *work conserving.* Denote the event that the server is idle at time s by $I_s \in \{0, 1\}$. In a work conserving scheme, idleness happens only when all the queues are empty

$$I_s = 1 \text{ only if } \textstyle\sum_i b_i[s] = 0. \tag{B.1}$$

Let $I[t] = \sum_{s=0}^{t-1} I_s$ be the total number of idle slots in interval $[0, t-1]$ and $I[0] = 0$. The following Theorem declares that *LQF*, on average, results in longer minimum queue length, as compared to all other work conserving policies.

Theorem 25. *Let μ be any causal work conserving scheduling policy for the queueing system described above. Then for any values of $T > 0$, $B[0]$, $I[T]$, $\sum\limits_i \sum\limits_{t=0}^{t=T-1} a_i[t]$ and $\beta(I[T]) \geq 0$,*

$$\mathbb{P}^{LQF}\left(\min_i b_i[T] \geq \beta(I[T]) \middle| B[0], I[T], \sum_i \sum_{t=0}^{t=T-1} a_i[t]\right)$$
$$\geq \mathbb{P}^{\mu}\left(\min_i b_i[T] \geq \beta(I[T]) \middle| B[0], I[T], \sum_i \sum_{t=0}^{t=T-1} a_i[t]\right).$$

Proof. Throughout this proof, we assume all policies are work conserving and we may drop the time index $[t]$, when there is no ambiguity. We also use $\min X$, $\max X$ and $sum(X)$ to denote respectively the minimum, the maximum and the sum of elements in the vector X. In our queueing system for each $t \in [0, T]$, we say a set of values $\{B[t],\ I[t],\ I[T],\ \sum_{\tau=0}^{T-1} sum(A[\tau])\}$ is *feasible*, if there exists a realization of the Bernoulli arrivals and a service scheme for time $\tau = 0$ to T which result in the above values. We assume all these values in this proof are feasible.

For any work conserving policy, we have $sum(B[t]) = \sum_{\tau=0}^{t-1} sum(A[\tau]) - (t - I[t])$, which implies, given $B[t]$ and $I[t]$, the total number of arrivals up to time t, $\sum_{\tau=0}^{t-1} sum(A[\tau])$, is deterministically known.

For given $T,\ I[T],\ \sum_{\tau=0}^{T-1} sum(A[\tau]),\ \beta = \beta(I[T])$ and any $t < T$, we define the success probability of a policy μ at time t, $\mathbb{P}^{(\mu)}(B[t], I[t], t)$, as

$$\mathbb{P}\Big(\min B[T] \geq \beta \ \Big| B[t],\ I[t],\ I[T],\ \sum_{\tau=0}^{T-1} sum(A[\tau]),\ \mu \text{ is used from } t \text{ to } T\Big).$$

We will prove that, given any initial state $B[0]$ and $I[0] = 0$, the *LQF* algorithm is optimal in the sense that its success probability is not less than any other policy μ. We do this using dynamic programming, and use induction to order states on their probability of failure. Note that because of the symmetry in the system, if $B[t]$ is a permutation of $\hat{B}[t]$, then $\mathbb{P}^{(\mu)}(B[t], I[t], t) = \mathbb{P}^{(\mu)}(\hat{B}[t], I[t], t)$. We denote this case by $B[t] \sim \hat{B}[t]$.

Induction Hypothesis:

1. For each $t < T$ and any state $B[t]$ and $I[t]$, we have

$$\mathbb{P}^{(LQF)}(B[t], I[t], t) \geq \mathbb{P}^{(\mu)}(B[t], I[t], t) \quad \forall \mu. \tag{B.2}$$

2. Let $B_1 = (b_1, ..., b_{\hat{i}}, ..., b_{\hat{j}}, ..., b_M)$ and $B_2 = (b_1, ..., (b_{\hat{i}}+1), ..., (b_{\hat{j}}-1), ..., b_M)$ and

$0 \le b_{\hat{i}} < b_{\hat{j}} - 1$. Then for any $t < T$ and $I[t]$, the following holds

$$\mathbb{P}^{(LQF)}(B_1[t], I[t], t) \le \mathbb{P}^{(LQF)}(B_2[t], I[t], t). \tag{B.3}$$

First we show that the hypothesis is valid for $t = T - 1$.

<u>Case for $t = T - 1$:</u> Let $u[T-1] \in \{1, ..., M\}$ be the served queue at time $T-1$. Recall that given $B[T-1]$, $I[T-1]$ and $\sum_{\tau=0}^{T-1} sum(A[\tau])$, the total number of arrivals during the last time slot $sum(A[T-1])$ is known. Then the probability of success, $\mathbb{P}(\min B[T] \ge \beta | B[T-1], I[T-1], I[T], u[T-1], \sum_{\tau=0}^{T-1} sum(A[\tau])) =$

$$\begin{cases} 0 & \text{if } (\min_k b_k \le \beta - 2) \text{ or } \left(b_{u[T-1]} \le \beta - 1\right) \text{or } (\max_k b_k \le \beta - 1) \\ P(l) & \text{otherwise} \end{cases} \tag{B.4}$$

where $P(l)$ is the probability that each queue in the set $\{k : b_k = \beta - 1\} \cup \{u[T-1] : b_{u[T-1]} = \beta\}$, of cardinality l, receives one packet at this time. Since the arrival processes are assumed to be identically distributed, given the total number of arrivals at this time, $sum(A[T-1])$, we have

$$P(l) = 1_{\{sum(A[T-1]) \ge l\}} \frac{\binom{M-l}{sum(A[T-1])-l}}{\binom{M}{sum(A[T-1])}}.$$

It can be verified that $P(l)$ decreases with l, therefore choosing $u[T-1]$ such that $b_{u[T-1]} > \beta$ (if possible) maximizes the probability of success. *LQF* is hence an optimal strategy at time $T - 1$.

Next we prove (B.3) holds for $t = T - 1$. Note that for *LQF*, we have $b_{u[T-1]} = \max_k b_k$. Based on the success probability in (B.4), we consider the following cases:

(1) $\min B_2 \le \beta - 2$: From (B.4) and since $(\min B_1 \le \min B_2)$, we have $\mathbb{P}^{(LQF)}(B_1, I[T-1], T-1) = \mathbb{P}^{(LQF)}(B_2, I[T-1], T-1) = 0$.

(2) $\max B_2 \le \beta - 1$: From (B.4) and since $(\min B_1 \le b_{\hat{i}} \le \max B_2 - 1 \le \beta - 2)$,

we have $\mathbb{P}^{(LQF)}(B_1, I[T-1], T-1) = \mathbb{P}^{(LQF)}(B_2, I[T-1], T-1) = 0$ for this case as well.

(3) $\min B_2 \geq \beta - 1$, $\max B_2 \geq \beta$: For this case, we have $b_{\hat{i}} \geq \beta - 2$. Now if $b_{\hat{i}} = \beta - 2$, then $\mathbb{P}^{(LQF)}(B_1, I[T-1], T-1) = 0$. Otherwise, $b_{\hat{i}} \geq \beta - 1$ and the parameters $l(B_1)$ and $l(B_2)$ of the success probability in (B.4), when the LQF is applied, for B_1 and B_2 respectively, are

$$l(B_1) = \hat{l} + 1_{\{b_{\hat{i}} = \beta - 1\}} \quad \text{and} \quad l(B_2) = \hat{l} + 1_{\{\max B_2 = \beta\}} \tag{B.5}$$

where $\hat{l} = card(\{k \neq \hat{i}, \hat{j} : b_k = \beta - 1\})$. Since $\beta \leq b_{\hat{i}} + 1 \leq \max B_2$, it can be seen that $l(B_1) \geq l(B_2)$. Consequently, (B.3) holds for $T-1$.

Inductive step: Suppose the hypothesis holds for $t+1, ..., T-1$. To show (B.2) at time t, given $B[t]$ and $I[t]$, note that if $b_i \neq 0$ for at most one $i \in \{1, ..., M\}$, then all work conserving policies will take the same action. Now suppose there are at least two non-empty queues. LQF chooses $i^* = \arg\max_k b_k$ to serve and suppose another policy μ chooses g, where $0 < b_g < b_{i^*}$. The system states at time $t+1$ corresponding to these two policies are respectively $B_{LQF}[t+1] = (b_1+a_1, ..., b_g+a_g, ..., b_{i^*}-1+a_{i^*}, ..., b_M+a_M)$ and $B_\mu[t+1] = (b_1 + a_1, ..., b_g - 1 + a_g, ..., b_{i^*} + a_{i^*}, ..., b_M + a_M)$, for the arrivals $A[t] = (a_1, ..., a_M)$. Also $I[t+1] = I[t]$ for both policies, since we assumed there exist non-empty queues at time t.

Note that $b_g + a_g \leq b_{i^*} + a_{i^*}$ for any Bernoulli arrivals. Now if $b_g + a_g = b_{i^*} + a_{i^*}$, then $B_{LQF}[t+1] \sim B_\mu[t+1]$. Otherwise, it can be easily verified that

$$\begin{aligned}\mathbb{P}^{(LQF)}(B_{LQF}[t+1], I[t+1], t+1|A[t]) &\geq \mathbb{P}^{(LQF)}(B_\mu[t+1], I[t+1], t+1|A[t]) \\ &\geq \mathbb{P}^{(\mu)}(B_\mu[t+1], I[t+1], t+1|A[t])\end{aligned}$$

where the first and the second inequalities follow respectively from (B.3) and (B.2) for $t+1$. By taking expectation over all arrivals $A[t]$, we conclude that (B.2) holds at time t as well.

Now, we show that the inequality in (B.3) holds at time t. Consider states $B_1[t]$ and $B_2[t]$ as defined in the induction hypothesis. Using LQF at this time, these states will lead respectively to $B_1[t+1]$ and $B_2[t+1]$. Without loss of generality, we assume LQF always chooses queue $\hat{j}$ to serve, if this queue has the maximum length. Let $k^* = \arg\max_{k \neq \hat{j}} b_k$ and consider the following cases:

(1) $b_{\hat{j}} < b_{k^*}$: It can be verified that $k^* \neq \hat{i}$ and the LQF algorithm chooses k^* to serve in both $B_1[t]$ and $B_2[t]$ states. Hence, we have $B_1[t+1] = (b_1 + a_1, ..., b_{\hat{i}} + a_{\hat{i}}, ..., b_{k^*} - 1 + a_{k^*}, ..., b_{\hat{j}} + a_{\hat{j}}, ..., b_M + a_M)$ and $B_2[t+1] = (b_1 + a_1, ..., b_{\hat{i}} + 1 + a_{\hat{i}}, ..., b_{k^*} - 1 + a_{k^*}, ..., b_{\hat{j}} - 1 + a_{\hat{j}}, ..., b_M + a_M)$.

Note that $b_{\hat{j}} + a_{\hat{j}} - 1 \geq b_{\hat{i}} + a_{\hat{i}}$ for any $a_{\hat{j}}, a_{\hat{i}} \in \{0, 1\}$. If $b_{\hat{j}} + a_{\hat{j}} - 1 = b_{\hat{i}} + a_{\hat{i}}$, then $B_1[t+1] \sim B_2[t+1]$. Otherwise, from the induction step at time $t+1$, (B.3) holds for $B_1[t+1]$ and $B_2[t+1]$ at time $t+1$. By taking expectation over all arrivals $A[t]$, (B.3) is shown to be true for t as well.

(2) $b_{\hat{j}} = b_{k^*}$: It can be seen that $k^* \neq \hat{i}$ and LQF chooses $\hat{j}$ and k^* to serve at states $B_1[t]$ and $B_2[t]$ respectively. So, we have $B_1[t+1] = (b_1 + a_1, ..., b_{\hat{i}} + a_{\hat{i}}, ..., b_{k^*} + a_{k^*}, ..., b_{\hat{j}} - 1 + a_{\hat{j}}, ..., b_M + a_M)$ and $B_2[t+1] = (b_1 + a_1, ..., b_{\hat{i}} + 1 + a_{\hat{i}}, ..., b_{k^*} - 1 + a_{k^*}, ..., b_{\hat{j}} - 1 + a_{\hat{j}}, ..., b_M + a_M)$. As in the previous case, since $b_{k^*} + a_{k^*} - 1 \geq b_{\hat{i}} + a_{\hat{i}}$ for any $a_{k^*}, a_{\hat{i}} \in \{0, 1\}$, (B.3) holds for this case at time t as well.

(3) $b_{\hat{j}} > b_{k^*}$: LQF chooses $\hat{j}$ to serve in both states $B_1[t]$ and $B_2[t]$. We have $B_1[t+1] = (b_1 + a_1, ..., b_{\hat{i}} + a_{\hat{i}}, ..., b_{\hat{j}} - 1 + a_{\hat{j}}, ..., b_M + a_M)$ and $B_2[t+1] = (b_1 + a_1, ..., b_{\hat{i}} + 1 + a_{\hat{i}}, ..., b_{\hat{j}} - 2 + a_{\hat{j}}, ..., b_M + a_M)$.

If $b_{\hat{j}} - 2 + a_{\hat{j}} \geq b_{\hat{i}} + a_{\hat{i}}$, then the same argument as before will follow. Otherwise, $b_{\hat{j}} - 1 + a_{\hat{j}} = b_{\hat{i}} + a_{\hat{i}}$ and for $(a_{\hat{i}}, a_{\hat{j}}) \in \{(0,0), (1,1)\}$, both states are similar and for $(a_{\hat{i}}, a_{\hat{j}}) \in \{(0,1), (1,0)\}$ they are symmetric. Observation 10 will show that the probability of the two latter cases is the same for $B_1[t]$ and $B_2[t]$. Hence, by taking expectation over different realizations of $A[t]$, we will have (B.3) with equality at t in this case.

Observation 10. *Since* $sum(B_1[t]) = sum(B_2[t])$*, given* $\sum_{\tau=0}^{T-1} sum(A[\tau])$*, the total number of arrivals in the remaining time slots is the same for both states. We assumed that the arrival processes are identically distributed and are independent over time slots. Therefore, it is seen that*
$\mathbb{P}((a_{\hat{i}}[t], a_{\hat{j}}[t]) = (a, 1-a) | B_w[t], I[t], I[T], \sum_{\tau=0}^{T-1} sum(A[\tau]))$ *has the same value for all* $a \in \{0, 1\}$ *and* $w = 1, 2$*.*

$\square$

Proof of Optimality of *MRF* Scheme (Theorem 19)

We will prove the optimality of the *MRF* scheme in Phase 1, based on the results developed in Appendix B on the performance of the *Longest-Queue-First (LQF) algorithm* in the context of queueing systems.

Note that $\mathbb{P}((C1), (C2), (C3)) = \mathbb{P}((C1), (C2)) \, \mathbb{P}((C3)|(C1), (C2))$ and for a given T_1, $\mathbb{P}((C1), (C2))$ is independent of the P2P scheme in Phase 1. Because, the values of $e_i[T_1]$ and $I[T_1]$ are independent of any work conserving scheme and only depend on the realization of the arrival processes. We will show that *MRF* maximizes $\mathbb{P}((C3)|(C1), (C2))$ and hence is optimal.

$$\mathbb{P}((C3)|(C1), (C2)) = \mathbb{P}(\min_i e_i[T_1] - x_i[T_1] \geq \beta(I[T_1]) | (C1), (C2)) =$$
$$\sum_d \mathbb{P}(\min_i e_i[T_1] - x_i[T_1] \geq \beta(d) | (C1), (C2), I[T_1] = d) \times \mathbb{P}(I[T_1] = d | (C1), (C2))$$

where $\beta(I[T_1]) = N - T + I[T_1]$. Note that $((C1), (C2), I[T_1] = d)$ is equivalent to $(\sum_i e_i[T_1] \geq \tilde{N}, I[T_1] = d)$, where $\tilde{N} = N + (M-1)\max(0, N + d - T)$.

Hence, $\mathbb{P}\left((C3)|(C1),(C2)\right) =$

$$\sum_{d}\sum_{\hat{N}\geq\tilde{N}} \mathbb{P}\left(\min_{i} e_i[T_1]-x_i[T_1]\geq\beta(d)|\sum_{i}e_i[T_1]=\hat{N}, I[T_1]=d\right)\times$$

$$\mathbb{P}\left(I[T_1]=d|(C1),(C2)\right)\mathbb{P}\left(\sum_{i}e_i[T_1]=\hat{N}|(C1),(C2),I[T_1]=d\right).$$

We can interpret $e_i[t]$ and $x_i[t]$ respectively as the accumulated arrival and service to some queue i, at time t, in which the arrival rate is p for all i and the total service rate is 1. For such a queueing system, Theorem 25 declares that the LQF maximizes $\mathbb{P}\left(\min_i e_i[T_1]-x_i[T_1]\geq\beta(I[T_1])|\sum_i e_i[T_1], I[T_1]\right)$, for all values of T_1, $I[T_1], \sum_i e_i[T_1]$ and $\beta(I[T_1])$ and any initial state.

From (4.2), $\arg\max_i e_i[T_1]-x_i[T_1]=\arg\max_i \hat{n}_i[T_1]$ and since for the infinite field size $n_i[t]=\hat{n}_i[t]$, LQF is translated to the MRF algorithm. Accordingly this implies that MRF maximizes $\mathbb{P}\left((C1),(C2),(C3)\right)$.

Proof of Lemma 17

Suppose $\sum_i e_i^{(k)} \geq N$, because otherwise the whole system does not have enough degrees of freedom to recover the block and hence the objective value in (4.34) is identically equal to 0, for any P2P scheme. Also recall our assumption $N>T$. (Note that if $N\leq T$, any *work conserving* scheme will achieve the optimal value of (4.34). More precisely, the duration of the frame T is large enough to broadcast at least N degrees of freedom and any peer will be able to recover the block after these N broadcasts.)

Since we have assumed $\sum_i e_i^{(k)}\geq N$ and $N>T$, there are enough arrived chunks such that we can have a new broadcast transmission at each time during frame $k-1$. That is we can indeed assume $\sum_i x_i^{(k)}=T$ in (4.34). Note that based on the state of the system, peers may all get full-rank before the deadline and we do not have to make T transmissions. However, for the sake of analyzing the optimization problem in (4.34), it does not hurt if we consider $\sum_i x_i^{(k)} = T$ instead of the second condition in (4.34).

Given the B2P arrivals $(e_1^{(k)}, ..., e_M^{(k)})$, we partition the set of peers $\{1, ..., M\}$ into sets $\mathcal{S}$ and $\mathcal{S}^c = \{1, ..., M\}\backslash\mathcal{S}$ as defined in Algorithm 6. Note that $\mathcal{S}^c$ is the set of peers who have not received enough number of B2P arrivals and no feasible P2P scheme can help them to recover the block before its deadline (i.e., $1_{\{e_i^{(k)}+\sum_{j\neq i} x_j^{(k)} \geq N\}} = 0$). Therefore since the peers are assumed to be cooperative, it is *socially beneficial* to let the peers in $\mathcal{S}^c$ broadcast all they have received over the P2P network. Let $T_1 = \min\{\sum_{i\in\mathcal{S}^c} e_i^{(k)}, \text{T}\}$. So we can devote the first T_1 slots of the current frame to transmissions from the peers in $\mathcal{S}^c$.

Now we can rephrase the optimization in (4.34) in a simpler form as follows,

$$\begin{array}{ll} \max \ \sum_{i\in\mathcal{S}} d_i 1_{\{x_i^k[kT] \leq e_i^k + T - N\}} & \\ \text{subject to} & \\ 0 \leq x_i^k[kT] \leq e_i^k & \text{for all } i \in \mathcal{S} \\ \sum_{i\in\mathcal{S}} x_i^k[kT] = T - T_1 & \end{array} \tag{B.6}$$

Note that if $T_1 = T$, for each $i \in \mathcal{S}$ we have $N - e_i^k \leq T = T_1$ and hence $1_{\{e_i^k+\sum_{j\neq i} x_j^k[T] \geq N-T_1\}} = 1$. That is during this frame only the peers in $\mathcal{S}^c$ will transmit and their transmissions are sufficient to deliver the whole block to the peers in $\mathcal{S}$.

Recall that from the assumption we maid here, at each time we can have a new broadcast transmission,

$$\sum_i x_i^{(k)}[t] = t - (k-1)T \qquad \text{for any } (k-1)T \leq t \leq kT.$$

On the other hand from (4.2) and for large field sizes, we have $n_i^{(k)}[t] = \hat{n}_i^{(k)}[t] = e_i^{(k)} - x_i^{(k)}[t] + \sum_j x_j^{(k)}[t]$.

To achieve the maximum objective value in (B.6), each peer $i \in \mathcal{S}$ can transmit upto $e_i^{(k)}+T-N$ chunks without hurting the objective value (i.e., $x_i^{(k)}[t] \leq e_i+T-N$). This will imply the following threshold on the rank of peers who can potentially broadcast at each time t,

$$n_i^{(k)}[t] = e_i^{(k)} - x_i^{(k)}[t] + t - (k-1)T \geq N + t - kT.$$

If all the peers hit the above threshold before the deadline T, then the remaining transmissions should be done by the peers subsequently in an increasing order of their deficit values. That is the peer with the smallest deficit (say $\hat{i}$) makes more transmissions while $x_{\hat{i}}^{(k)}[t] < e_{\hat{i}}^{(k)}$ or equivalently $n_{\hat{i}}^{(k)}[t] > t - (k-1)T$. If there is a need for more transmissions, the peer with the next smallest deficit value transmits its remaining chunks and so on until all peers get full-rank or we hit the deadline T.

VITA

Navid Abedini graduated with a B.S. Degree in Electrical Engineering from Sharif University of Technology, Iran in 2008. He is currently a Ph.D. candidate in Electrical and Computer Engineering, Texas A&M University, where he is expected to graduate in December 2012. He is the recipient of the Capocelli prize (the best student paper award of IEEE Data Compression Conference 2010). He has also received IEEE WiOpt 2011 travel grant, and Texas A&M Graduate Student Research and Presentation grant in 2010. In 2007 he was awarded in the First National Congress of the Iranian Talented Youth and also in the National Undergraduate entrance exam (Iran, 2004) was ranked 7 (among 360,000).

His research mainly focuses on scheduling and optimization of the communication networks, queueing theory, coding and information theory. His recent projects deal with design and analysis of algorithms to manage Peer-to-Peer and content distribution networks.

Email : navid.abed@gmail.com