Harnessing Multiple Wireless Interfaces for Guaranteed QoS in Proximate P2P Networks

Navid Abedini, Mayank Manjrekar, Srinivas Shakkottai
Dept. of ECE, Texas A&M University
Email: {novid_abed, mayankm, sshak} @tamu.edu

Libin Jiang
Dept. of CS, California Institute of Technology
Email: libinj@caltech.edu

Abstract—We consider the problem of content distribution to a group of cooperative wireless peer devices that desire the same block of information. The QoS metric is that peers are all required to receive the block by a fixed deadline, with a certain target probability. The block 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, 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 encoded using random linear codes to alleviate the duplicate chunk reception issue. We seek an algorithm that can attain the QoS metric at the lowest cost of using B2P network.

We transform the problem into 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. We use dynamic programming and queueing ideas to show that for large field sizes, a combination of Max-Rank-First and Non-min-Rank-First policies for P2P transmissions is optimal, and determine the stopping time for B2P transmissions using a Markov chain model. We provide performance bounds for finite field sizes, and illustrate our insights using simulations.

I. INTRODUCTION

The past few years have seen the rise in popularity of smart, handheld devices with multiple wireless communication interfaces. For example, smart phones usually 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 such as WiFi may be used for ad-hoc peer-to-peer broadcast transmissions, and are usually less expensive.

The objective of this work 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 a block of information within a specified deadline and at lowest total cost. Essentially, the problem that we are interested in is the timely synchronization of data on multiple wireless devices with minimal infrastructure support. Applications include 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). Also live streaming applications (which require hard deadlines for delivering the multimedia content) can be developed based on the framework studied in this paper.

Our setup is illustrated in Figure 1, 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. The block of information is divided into chunks 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.

![Figure 1. Hybrid P2P wireless network. Each device can utilize both base-station-to-peer (B2P) as well as broadcast peer-to-peer (P2P) communication.](image)

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 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 [3]–[5]. 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 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. In [6], we comment on some challenges of considering lossy P2P transmissions. 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 [3] to a larger class of algorithms.

The problem of managing multiple interfaces for the purpose of cooperatively sharing content over a P2P network has been also recently investigated in [7]. Unlike our delay sensitive model, they try to maximize a utility function of the average information flow (video) rate achieved by the peers. However delay sensitive communication in P2P networks has attracted significant recent interest. For example, [8] develops 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 trade-offs in choosing one or the other wireless interface [2]. However, they do not consider P2P communications.

Main Results

In Section III, 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 (NnRF) scheme for P2P transmissions, and 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 Section IV. Here, our quality of service (QoS) target is to ensure that all peers obtain the block with a target probability \( \eta \) 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 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.

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

Because of space limitations, we omit most of the proofs which can be found in the technical report [6].

II. System Model

In our system, 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. 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 \( p \). We denote the total number of chunks delivered to peer \( i \) via the B2P network by the beginning of slot \( t \) using \( e_i[t] \).

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

\[
\sum_t x_i[t] - \sum_t x_i[t-1] \leq 1 \quad \forall t > 0. \tag{1}
\]
Also, 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 chunk selection we use random linear network coding. Each coded chunk is a random linear combination of the \( N \) original chunks, with coefficients from \( F_q \), a finite field of size \( q \). Therefore, we can associate a vector \( \gamma = (\gamma_1, ..., \gamma_N) \) with 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 \( \gamma \), when there is no ambiguity.

Let \( S_i[t] \) be the set of vectors possessed by peer \( i \) at time \( t \), with cardinality \( n_i[t] \). Thus,

\[
\hat{n}_i[t] = e_i[t] + r_i[t] = e_i[t] - x_i[t] + \sum_j x_j[t].
\] (2)

We represent the dimension of the subspace spanned by \( S_i[t] \) using \( n_i[t] \), which is called the rank of peer \( i \) at time \( t \).

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

\[
x_i[t] \leq e_i[t] \quad \text{for all } t \geq 0.
\] (3)

Observation 1: Given a sequence of transmissions \( (\gamma_1, ..., \gamma_l) \) for P2P dissemination, any permutation \( \Pi(\gamma_1, ..., \gamma_l) \) is also equally effective, i.e., the order of P2P transmissions is not relevant.

Note that a feasible P2P scheme \( \{x_i[t]\}_{i=0}^{T} \) must satisfy (1) and (3). Also, in order to decode the original \( N \) chunks at time \( T \), each peer \( i \) must have \( n_i[T] = N \).

III. 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. [3] 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., \( u_i[t] \in \{ \text{arg max } n_i[t] \} \). 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, and we show that it performs equally well as MRF.

Non-min-Rank-First (NmRF) algorithm: At any time \( t \), one of the peers, except those with the minimum rank, transmits, i.e., \( u_i[t] \in \{1, ..., M\} \setminus \{\text{arg min } n_i[t]\} \). 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 1: 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 \( T[t] = (\gamma_1, ..., \gamma_{t + \text{Dim}(T[t]) - 1}) \)

\[
\gamma_i \in \text{Span}(S_i[0]) \quad \forall l \geq t
\]

\[
\text{Dim}(T[t] \cup S_i[t]) = N \quad \forall i
\] (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 \( \text{Dim}(T[t]) + \text{Dim}(S_i[t]) \geq \text{Dim}(T[t] \cup S_i[t]) = N \), for all \( i \), which implies

\[
\text{Dim}(T[t]) \geq N - \min \text{Dim}(S_i[t]) = N - \min_{i} n_i[t].
\] (5)

Also, it is shown [3] that when all the ranks are equal, i.e., \( n_i[t] = n \) for all \( i \), then

\[
\text{Dim}(T[t]) \geq N - n + 1.
\] (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 \( \gamma_j \in T[t] \) which is a linear combination of the vectors in \( S_j[t] \). Otherwise, we would have

\[
\text{Dim}(T[t] \cup S_j[t]) = \text{Dim}(T[t]) + \text{Dim}(S_j[t])
\]

\[
\text{and from (4) Dim}(T[t]) = N - \text{Dim}(S_j[t]),
\]

which contradicts (5) for the non-minimum rank peer \( j \) (or (6) when all the ranks are equal). Consequently, peer \( j \) can generate \( \gamma_j \) and transmit it at time \( t \).

For the proof of the probability bound \( 1 - \frac{NM}{q} \), we can follow a similar argument as in [3], which we omit due to space constraints.

Corollary 2: For the infinite field size, NmRF 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 at time \( t \), we let \( u[t] \) transmit 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. If we assume that \( q \) is large enough, then every new randomly coded chunk is indeed useful (i.e., \( n_i[t] = \hat{n}_i[t] \)). 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 3:** 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

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

Note that \( \tau^* \) follows from minimizing the number of transmissions such that each peer receives enough DoF. In Theorem 5, we will see that \( \tau^* \) implies intuitive conditions on the whole system as well as each individual peer, for a QoS metric to be achievable.

For the finite field case, the following Theorem evaluates the performance of \( \text{NmRF} \) algorithm. The proof follows from a simple matrix rank argument. For details see [6].

**Theorem 4:** When the random linear coding coefficients are drawn from a finite field \( F_q \), \( \text{NmRF} \) completes dissemination of the block to all peers by time \( \tau^* \) with probability at least \( 1 - \frac{M}{q^T} \).

**Proof:** In the \( \text{NmRF} \) algorithm, \( u[t] \in \{1, \ldots, M\} \setminus \{\arg \min n_i[t]\} \). First, we show that this theorem holds even if \( u[t] \in \{1, \ldots, M\} \setminus \{\arg \min \hat{n}_i[t]\} \), i.e., at each time \( t \), a peer whose total number of available 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 3, both result in \( \hat{n}_i[t + \tau^*] \geq N \) for all \( i \). Hence, we are interested in computing the following probability

\[
\mathbb{P}(\forall i : n_i[t + \tau^*] < N | \forall i : \hat{n}_i[t + \tau^*] \geq N) = 1 - \mathbb{P}(\exists i : n_i[t + \tau^*] < N | \forall i : \hat{n}_i[t + \tau^*] \geq N) \geq 1 - \sum_i \mathbb{P}(n_i[t + \tau^*] < N | \hat{n}_i[t + \tau^*] \geq N),
\]

where the last inequality follows from the union bound. Note that \( \hat{n}_i[t + \tau^*] \geq N \) implies that peer \( i \) has received at least \( N \) randomly generated chunks by time \( t + \tau^* \), and \( \mathbb{P}(n_i[t + \tau^*] < N | \hat{n}_i[t + \tau^*] \geq N) \) is the probability that the corresponding \( \hat{n}_i[t + \tau^*] \times N \) matrix is not full-rank. Therefore, we have (from Lemma 5, [6])

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

As a result, we obtain

\[
\mathbb{P}(\forall i : n_i[t + \tau^*] < N | \forall i : \hat{n}_i[t + \tau^*] \geq N) \geq 1 - \frac{M}{q - 1}.
\]

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

**Observation 2:** Note that unlike the bound for the uncoded initialization case, \( 1 - \frac{N}{q^T} \), the performance guarantee found in Theorem 4 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.

IV. 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. We define our QoS metric \((\eta, T)\), as a requirement that all peers must receive the whole block with probability \( \eta \) 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 III-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. 2. Two-phase scheme. Both B2P and P2P transmissions take place in Phase 1, whereas only P2P transmissions occur in Phase 2.

The time-line in our hybrid system is illustrated in Figure 2. We have two phases, where both B2P and P2P methods are used in Phase 1, while only P2P is used in Phase 2. Further if we assume \( N \gg M \), due to symmetry of the B2P channels,
all B2P channels will be turned off simultaneously at $T_1$. We first consider the problem in the case of infinite field size.

A. Infinite Field Size

From Figure 2, we observe that the evolution of information state in Phase 2 is identical to the pure P2P broadcast system in Section III. Theorem 3 can be used to specify the initial conditions needed to attain any target completion time (deterministically) in this situation. Thus, from Theorem 3, we define the class of states $C_\tau$ that can finish within time $\tau$ purely using P2P broadcasts by

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

Note that the peer selection algorithm during Phase 2 would follow $NnRF$. Then the optimal stopping time $T_1$ must be the smallest value (least 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 $\eta$.

Now, consider the evolution of information state during Phase 1. P2P broadcast can only happen 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 to P2P idleness in slot $s$ by $I_s \in \{0, 1\}$. Hence, for all work conserving P2P schemes $I_s = 1 (\sum_{t=0}^{s} I_s) = 0$.

Denote the cumulative P2P idle time within the time interval $[0, t]$ by $I(t)$, i.e., $I(t) = \sum_{s=0}^{t-1} I_s$. The following theorem expresses the conditions for $(\eta, T)$ to be achievable.

**Theorem 5:** A target QoS $(\eta, T)$ can be achieved if and only if there exists a stopping time $T_1 \leq T$, and a feasible P2P transmission schedule, $\{x_i[t]\}_{t=1}^{T_1}$, such that the following conditions hold with probability at least $\eta$:

\begin{align*}
(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.
\end{align*}

Note that (C1) simply requires the whole system to be full-rank. Also From Theorem 3, $T - T_1 \geq \tau^*$ which leads to conditions (C2) and (C3). The intuition is that in order to achieve the QoS, the idleness should not be too much ((C2)) and each individual peer should receive enough chunks in Phase 1 ((C3)).

The next theorem declares that the MRF algorithm is an optimal scheme for Phase 1.

**Theorem 6:** For any $T, T_1$ and $\eta$, MRF maximizes $\mathbb{P}((C1), (C2), (C3))$ over all work conserving schemes.

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 [6].

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 $M$ whose state at time $t = (I[t], Z[t])$, where $Z[t]$ is a vector of $z_i[t] = e_i[t] - x_i[t]$ elements. For brevity, we may also use $(I[t], z_i[t] : i)$ to denote $(I[t], Z[t])$. For this Markov chain, Theorem 5 determines a set of desirable states, $C(N, M, T, t)$, to be hit at time $t = T_1$.

**Lemma 7:** For the Markov chain $M$, we have

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

where $i^* = \arg \max z_i$ and $\{a\}^+ = \max(a, 0)$, and $C(N,M,T,t) = \{ (I,Z) \mid \min_i z_i - I \geq N - T, \sum_i z_i + (t-I) \geq N + (M-1) \max(0,N-T+I) \}$.

In Algorithm 1, we present an algorithm to achieve the QoS target and summarize its performance as follows.

**Theorem 8:** Algorithm 1 achieves $(\eta, T)$ with the minimum cost, when the field size is infinite.

**Algorithm 1 Optimal B2P and P2P schemes (infinite field)**

Given $M, N, p, T$ and $\eta$:

1. Find $T_1^*$ such that:

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

2. For $0 \leq t < T_1^*$:

   - turn all B2P channels on and use MRF to schedule the P2P network.

3. At $t = T_1^*$:

   - turn all B2P channels off.

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

B. Finite Field Size

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

We saw that Algorithm 1 provides each peer with at least $N$ coded chunks by the deadline $T$, with a probability at least equal to $\eta$. For the finite field case, the received chunks by a peer may be linearly dependent. In this case, the corresponding peer fails to decode the original chunks. The following theorem, whose proof follows from an argument similar to Theorem 4, declares the probability of such an event.

**Theorem 9:** For the field size $q$, if the proposed scheme in Algorithm 1 is applied,

$$\mathbb{P}(\exists i : n_i[T] < N) \leq 1 - \eta + \frac{M}{q-1}. \quad (9)$$
We can see that the probability of failure has two components. One stems from the unreliability of the B2P channels, upper bounded by some $\delta_1$, and the other from the finiteness of the field size, upper bounded by $\delta_2$. 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 $\delta_2$, each peer requires to receive more coded chunks, which means that B2P usage and the associated cost increase.

**Lemma 10:** The bound $\delta_2 \in (0, \frac{M}{q^2})$ is achievable, if for all $i$ we have $n_i(T) \geq N + \Delta(\delta_2)$, where $\Delta(\delta_2) = \left[ \log_q \left( \frac{M}{q^2} \right) - \log_q(\delta_2) \right]$ 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 $\delta_2$.

**V. SIMULATION**

We now evaluate the performance of Algorithm 1 for a simple hybrid network with $M = 4$ users and different $N$.

Figure 3(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 5).

The effect of finite field sizes is evaluated in Figure 3(b). Clearly, increasing the field size results in a better performance.

Figure 3(c) displays the minimum cost to achieve the QoS versus $\Delta$, 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 $\Delta$ incurs more cost of the B2P usage. Note that as the field size decreases, a larger $\Delta$ is required. For example for $q = 4$, we need at least $\Delta_{\text{min}} = 2$ extra chunks in order to meet the QoS.

**VI. CONCLUSION**

In this paper, we studied the problem of content distribution in a wireless P2P network, where all peers desire a common block of content within a deadline with a target probability. We utilized random linear coding over finite fields in our algorithms to simplify coordinating transmissions. We considered two variations of this problem: (i) a pure P2P broadcast network, and (ii) a hybrid network, in which an expensive B2P network assists the P2P network. In the pure P2P case, our objective was to find optimal P2P schemes that minimize the total number of transmissions to disseminate the content.

We proposed the NmRF algorithm, and showed it is optimal for large field sizes. For the hybrid network, we presented a two-phase algorithm which minimizes the cost of the B2P usage while meeting the QoS for large field sizes. We also evaluated the performance of this algorithm both analytically and by simulation.

**REFERENCES**


