Optimal Rate Allocation and Scheduling in Cooperative Streaming
Mohammad Reza Zakerinasab, Mea Wang
Department of Computer Science
University of Calgary
{mrzakeri, meawang}@ucalgary.ca

Abstract—The demand for multimedia streaming, especially from mobile devices, is growing rapidly, mostly due to the increasing deployment of broadband cellular technologies and the growing computing power on modern mobile devices. While cellular carriers strive to meet the demand, mobile devices are challenged by the variable transmission rate in wireless channels and limited battery capacity. To this end, we propose an energy-efficient cooperative streaming system. In the proposed system, mobile devices collectively stream a copy of the multimedia content from the source over cellular links. The devices form a cooperative group and share received content with each other over short-range links. The design of the system is guided by the optimal rate allocation and scheduling (RAS) algorithm that determines the amount of data and the data to be transmitted on each link. The actual data scheduled to be transmitted on each link is delivered in coded form for efficient loss detection and recovery. Overall, the system minimizes both the streaming traffic in the cellular network and the energy consumed by streaming applications on mobile devices. Our experimental results show that significant energy saving is achieved in the proposed system. Moreover, our RAS algorithm prolongs the streaming session for the entire cooperative group.

I. INTRODUCTION

The demand for high-quality video streaming on mobile devices grows along with the deployment of 3G/4G technologies and the increasing computing power on mobile devices. Currently, video streaming traffic accounts for more than 53% of the cellular Internet traffic, and is expected to reach 66% in the next four years [1]. While cellular carriers strive to meet the expected growth, mobile devices are also challenged by the variable transmission rate in wireless channels and limited battery capacity. Reducing the loss rate and bandwidth fluctuation is a straightforward, but not easy, way to improve cellular networks for multimedia streaming. The alternative approach is to encourage cooperation among mobile devices by utilizing short-range links such as WiFi and Bluetooth [2]. The use of short-range links in addition to cellular links not only alleviates the workload in cellular networks, but also increases the overall link capacity around individual mobile devices.

The cooperative system is formed by groups of mobile devices that are within the same WiFi network and are interested in viewing the same video at the same time. As shown in Fig. 1, there are groups of mobile devices, where within each group the mobile devices communicate over short-range (WiFi, in this case) links and form a swarming session to exchange multimedia content received from the cellular network [2]. In this cooperative streaming system, mobile devices of the same group contact the source regarding the video of their interest. The video source prepares and serves the multimedia content over cellular links. Upon receiving the multimedia content, a mobile device broadcasts the content in the local short-range network. Due to higher capacity of short-range links, this cooperation leads to higher data rates and shorter delays in the streaming session. Without loss of generality, in this paper, we assume that the cellular network operates on 3G links and the short-range network operates on WiFi links.

Keller et al. argue that such a cooperative setup is useful in many scenarios [3]. For example, a group of users (either located at the same site or remote sites) watch a live event or a movie together; family members view the same video on their own mobile devices when travelling. In general, this setup is suitable for any occasion that involves streaming the same video to a group of mobile devices within proximity of each other. Fitzek et al. conclude that this setup leads to higher data rate and shorter delays. It also improves the energy efficiency on mobile devices, as the energy consumed by short-range links is less than that by cellular links per transmitted bit [4].

In this paper, we seek to minimize both the streaming traffic in the cellular network and the energy consumed by streaming applications on mobile devices. To do so, we propose an energy-efficient cooperative streaming system. The design of the system is guided by the optimal rate allocation and scheduling (RAS) algorithm that determines the amount of data and the data to be transmitted on each link. The actual data scheduled to be transmitted on each link is delivered in coded form for efficient loss detection and recovery. The use of network coding also simplifies the cooperation protocol. However, the coding operations introduce additional delays and leads to extra energy consumption on mobile devices. To address these shortcomings, we apply the two-level systematic networking coding scheme proposed in [5] so that coding is performed only if necessary and the coding operations on mobile devices are greatly simplified to conserve energy. Compared to different cooperative mobile streaming systems, including Microcast [3], our proposed system achieves significant energy saving on mobile devices. Moreover, our RAS algorithm prolongs the streaming session for the entire cooperative group. Furthermore, our RAS algorithm supports mobility in the cellular network networks by simply sharing the state information of the network between base stations. Since focus and the object of this work is on optimal rate allocation and scheduling, we do not consider security and privacy issues, as they are orthogonal to this work.

The remainder of this paper is organized as follows. Sec. II reviews existing proposals on combining multiple wireless channels and the use of network coding in these systems. The design of the cooperative streaming system and the RAS algorithm are presented in Sec. III. Sec. IV compares the performance of the proposed system with other cooperative streaming systems. Finally, Sec. V concludes the paper.

II. RELATED WORKS

The use of short-range (e.g., WiFi) communication to offload traffic in cellular networks was originally proposed in [2]. Since then, this idea has been adapted in different proposals in wireless networks. To reduce delays, multiple
copies of data are shared within the cooperative network that operates on the short-range links [6]. To recover errors from cellular links, error correction code produced by devices that receive good copies of the content are shared among smartphones over short-range links [7]. To increase receiving throughput of mobile devices, multiple copies of data are sent over multi-hop unicast connections in a WiFi network [8]. To reduce traffic in the cellular network, smartphones and tablets may store popular video files and serve them later via device-to-device localized transmissions [9]. For multimedia streaming, Microcast adapts this cooperative arrangement and utilizes the overhearing property in wireless channels to share content received from cellular links in a local WiFi network [3]. Furthermore, Microcast employs network coding to ensure that the content broadcasted in the WiFi network is useful to as many smartphones as possible.

Network coding (NC) was originally proposed in the field of information theory to achieve optimal communication throughput [10]. After the introduction of random linear NC (RLNC) [11], the concept of NC has been widely applied in practical content distribution systems [12]. With RLNC, the to-be-disseminated content is divided into \( k \) original blocks of the same size. Coded blocks are then produced as linear combinations of the original blocks using random coefficients in Galois Field GF(256). Each node in the network receives and recodes the coded blocks using random coefficients to increase the linear independency of any \( k \) coded blocks in the network. The intended receiver can recover the original content after receiving any \( k \) linearly independent coded blocks. The concept of RLNC was first applied in wireless networks in [13] to improve the communication throughput. Since then this idea has been used for many practical applications, including error recovery in wireless P2P video broadcasting [7], and unequal error protection in multi-layered video streaming [14].

One of the challenges of applying NC is the computational complexity of the encoding and decoding operations. This challenge has prevented the application of NC on mobile devices due to limited computing and battery power until recent advancement in processing power on mobile devices. The first implementation of NC on mobile devices was presented in [15], followed by an implementation of NC on iPhone in [16]. These studies have shown that it is now feasible to perform coding operations on mobile devices, which leads to investigations on various applications of NC in mobile networks. For instance, Pedersen et al. proposed Pictureviewer, a mobile application that utilizes NC to transfer pictures among mobile devices over WiFi links [17]. Furthermore, network coding is utilized as an effective mechanism to recover losses and errors with minimum communication overhead in wireless networks [18]. Towards multimedia streaming, Vingelmann et al. applied NC to stream video content among a group of iPhone devices [19].

Recently, network coding has been applied in cooperative streaming systems (exemplified by system in Fig. 1) to reduce traffic in the cellular network and to simplify the cooperation mechanism in the WiFi network [3]. Microcast performs RLNC in GF(256) to encode the video content. The coded content is transmitted from the source to smartphones over a cellular network and is then shared among smartphones in local WiFi network [3]. However, this design trade higher power consumption on mobile devices for less traffic in the cellular network and better system throughput. Although it has been shown that modern smartphones can perform coding operations in GF(256) at a decent rate [15], [16], the operations still consume a noticeable amount of energy. To this end, a two-level coding scheme has been proposed to reduce the computational complexity on mobile devices [5].

In this paper, for the first time, we formulate the power consumption minimization problem in the NC-based cooperative streaming system and propose an optimal rate allocation and scheduling (RAS) algorithm. Our cooperative streaming system, designed based on the RAS algorithm, minimizes both the streaming traffic in the cellular network and the energy consumed by streaming applications on mobile devices. The system carefully employs NC only when necessary to minimize the communication and computational overhead introduced by coding operations. At last, the system enforces fairness in battery drainage among mobile devices so that the system can support longer streaming sessions.

### III. ENERGY-EFFICIENT COOPERATIVE STREAMING

#### A. Cooperative Streaming

A video source \( S \), hosted in a cloud, provides the streaming service at the rate of \( r \) bps to a set of cooperating mobile devices \( N \). The video is divided into segments, representing a short duration of the playback. Each mobile device maintains a playback buffer storing segments that are due for playback in the immediate future. The size of this buffer also marks the window of interest of each mobile device. To achieve smooth playback on the mobile devices, each segment must be received and decoded prior to its playback deadline. We now present the two layers of the cooperative streaming system: the cellular network and the cooperative network.

1) In the Cellular Network: The connection between mobile device \( i \), referred to as node \( n_i \), hereon, and the video source is a 3G link \( g_i \) with capacity \( c_i \) and loss rate \( p_i \). We further categorize nodes into two groups: active nodes and passive nodes. An active node communicates with the video source to download video segments with link has capacity \( c_i > 0 \). A passive node relies on active nodes in the cooperative network for receiving the content, maybe due to limited battery power, lack of data plan, or weak cellular signals. To keep the network model simple, we assume that a link with \( c_i = 0 \) still exist between the video source and the passive nodes. Since interference management among cellular links is not the focus of this work, we assume that the base station properly manages the wireless channels and assigns sub-channels to connections.

In this system, the source serves only one copy of the video to active nodes, and the nodes cooperatively exchange received segments in the network to deliver missing segments on all devices. Hence, the source schedules the transmission of segments according to their playback deadline and the rate allocation on each cellular link. The video source collects channel state and energy usage information from all active nodes, based on which the RAS algorithm determines to which nodes the video segment will be pushed. Hence each cellular link transmits different complete or partial segments within the current window of interest. For clarity, we assume a segment represents one time unit of the playback. Over time, exactly one copy of the video is streamed over the cellular network. Here, we assume that an algorithm for estimating the next state of channel, based on the channel state history, is in place.

Before pushing video segments, the source first divides a segment into \( k \) original blocks of the same size and serves the segment using systematic code in Galois Field GF(256). The coded blocks, a random linear combination of the original blocks, are sent along with the original blocks to node \( n_i \) that is selected by the source. Please note that there is an extensive body of research on how to divide the video segment into blocks or how to encode them together to minimize...
signal distortion caused by packet losses. We assume such an algorithm is in place and focus on determining the number of blocks to be transmitted. The number of coded blocks required to ensure the segment can be successfully recovered by node \( n_i \) is given in Eqn. 1:

\[
\epsilon_i = \frac{E[p_i]}{1 - E[p_i]} k 
\]  

(1)

, where \( E[p_i] \) is the expected packet loss rate of the 3G link \( g_i \). In other words, the source sends \( k \) original blocks and \( \epsilon_i \) coded blocks, and node \( n_i \) receives \( 0 \leq \epsilon_i \leq k \) coded blocks and \( 0 \leq k_2 \leq k \) original blocks over link \( g_i \) and \( k_1 + k_2 = k \). If \( k_1 \leq k \), i.e., not all of the \( k \) received blocks are original blocks, the missing original blocks can be recovered by solving the linear system formed by the \( k_1 \) original blocks and the \( k_2 \) coded blocks. To better characterize the effectiveness of the coding scheme, we define delivery rate \( d_i \) as the ratio of the streaming data among all received data, including control messages, sequence numbers, coding information, etc. Hence, in the cellular network, node \( n_i \) receives segments at rate \( s_i \) over \( g_i \), and the received data is recovered at rate \( d_i * s_i \).

2) In the Cooperative Network: In the cooperative network each pair of nodes \((n_i, n_j)\), \( n_i, n_j \in \mathcal{N} \), are reachable to each other over a WiFi link \( \{w_{i,j}\} \), with capacity \( c_{i,j} > 0 \) and packet loss rate \( p_{i,j} \geq 0 \). Hence, the cooperative network is a fully connected network. We assume that each WiFi link is bi-directional, i.e., \( w_{i,j} \approx w_{j,i}, \) with \( c_{i,j} = c_{j,i} \) and \( p_{i,j} = p_{j,i} \). In the WiFi network, we consider the time division multiple access (TDMA) model for the following reason. While IEEE 802.11 networks are based on orthogonal frequency-division multiplexing (OFDM) and can transmit the information on multiple carrier frequencies, the usual smartphones are equipped with just one cellular and one WiFi antenna. This means that when broadcasting in the WiFi network, each smartphone is in either the sending mode or the receiving mode, but not both. To avoid collision in the broadcast session, only one node can send at a time, while all other nodes are in the receiving mode. Hence, the broadcasting node can use all the available frequency range of the WiFi channel. Nodes in our system take advantage of this property and employ a broadcast mechanism to share received segments. Upon receiving a complete or partial segment, node \( n_i \) becomes the seed of this segment in the cooperative network, and is responsible to disseminate it to all other nodes. Before doing so, node \( n_i \) first reconstructs the \( k \) original blocks and serves the segment using systematic code in Galois Field GF(2). The coded blocks, XOR of a random subset of the original blocks, are broadcasted in the WiFi network. Our experiments on the Galaxy Nexus phone indicate that encoding and decoding in GF(2) is almost 100 times more energy efficient than that in GF(256). Since the loss rate in the WiFi network is expected to be much lower than that in the cellular network, the field size of 2 is sufficient. The number of coded blocks required to ensure the segment can be successfully recovered by all other nodes is given in Eqn. 2.

\[
\epsilon_i = \frac{E[p_i^w]}{1 - E[p_i^w]} k 
\]  

(2)

, where \( E[p_i^w] \) is the expected packet loss rate when node \( n_i \) broadcasts. The loss rate \( p_i^w \) is the maximum of all WiFi links incident to node \( n_i \), i.e., \( p_i^w = \max_{j} p_{i,j} \forall n_j \in \mathcal{N}_{n_i} \). In other words, node \( n_i \) broadcasts \( k \) original blocks and \( \epsilon_i \) coded blocks to ensure that all nodes in the WiFi network receive at least \( k \) (coded or original) blocks. To be energy efficient, a node may stop receiving once it has \( k \) linearly independent blocks. Up to now, every node in the WiFi network should have a copy of this segment, and its streaming is completed. In fact, node \( n_i \) broadcasts a segment at rate \( b_i \) in the cooperative network, and node \( n_j \in \mathcal{N}_{n_i} \) receives the segment at rate \( b_i(1 - p_{i,j}) \). The segment is recovered at rate \( d_{i,j}b_i(1 - p_{i,j}) \), where \( d_{i,j} \) is the delivery rate for transmitting coded blocks over \( w_{i,j} \).

Due to the use of network coding over cellular and WiFi links, all blocks of the same segment are equally useful in recovering the segment. This feature simplifies the application of the RAS algorithm, to be proposed in Sec. III-C, since no data reconciliation is needed to reconstruct a video segment. Interested readers may refer to [5] for more detail about the two-level coding mechanism used in the streaming system. For clarity, we summarize the notations used in this section in Table I, in which some are already introduced and some will be defined in the energy usage minimization problem (Sec. III-B).

### B. The Power Consumption Minimization Problem

To minimize the energy consumption in the cooperative network described in Sec. III-A, we need an optimal rate allocation and segment scheduling algorithm in the cellular network and the WiFi network. To do so, we first formulate the power consumption minimization problem. We define \( P_i(r) \) (in Watt or W) as the energy consumed by a cooperative node \( n_i \in \mathcal{N} \) to receive the video stream at rate \( r \), and \( P_i(r) = \sum_{n_j \in \mathcal{N}} P_j(r) \) as the objective energy function. According to [20], we consider the energy consumption of data transmission and coding operations as a linear function of transmission and coding rate. The objective function is defined as follow:

\[
P_i(r) = \alpha_i s_i + \beta_i b_i + \gamma_i \sum_{j} d_{i,j} b_j (1 - p_{j,i}), \tag{3} 
\]

\( \forall n_i \in \mathcal{N}, n_j \notin \mathcal{N}_{n_i} \)

, where \( \alpha_i, \beta_i, \) and \( \gamma_i \) are energy efficiency factors (in J/bit) of node \( n_i \) when receiving and decoding each bit of data over a 3G link, encoding and broadcasting each bit over the WiFi network, and receiving and decoding each bit in the WiFi network. Note that the idle power is not included in this formula, since the objective here is to minimize the power consumption due to data transmission. However, the idle may affect lifetime of mobile devices, and we will discuss this in Sec. III-D. According to the setup in Sec. III-A, the optimization problem can be formulated as follows:

\[
\min_{s_i \geq \beta_i d_{i,j}} \sum_{n_i \in \mathcal{N}} P_i(r), \forall n_i \in \mathcal{N} \tag{4} 
\]

s.t. \( (1) \leq \sum_{i} s_i, \forall n_i \in \mathcal{N} \)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{N} )</td>
<td>a set of cooperating mobile devices</td>
</tr>
<tr>
<td>( g_i )</td>
<td>3G downlink for node ( n_i \in \mathcal{N} )</td>
</tr>
<tr>
<td>( w_{i,j} )</td>
<td>WiFi link between nodes ( n_i ) and ( n_j )</td>
</tr>
<tr>
<td>( c_{i,j} )</td>
<td>link capacity (in bps) of ( g_i ) and ( w_{i,j} )</td>
</tr>
<tr>
<td>( p_{i,j} )</td>
<td>packet loss rate of links ( g_i ) and ( w_{i,j} )</td>
</tr>
<tr>
<td>( d_{i,j} )</td>
<td>delivery rate of coded data on links ( g_i ) and ( w_{i,j} )</td>
</tr>
<tr>
<td>( r )</td>
<td>the streaming rate (in bps)</td>
</tr>
<tr>
<td>( P_i(r) )</td>
<td>power consumption (in Watt or W) of node ( n_i )</td>
</tr>
<tr>
<td>( \alpha_i, \beta_i, \gamma_i )</td>
<td>energy efficiency factors (in J/bit) of node ( n_i )</td>
</tr>
<tr>
<td>( s_i )</td>
<td>3G download rate (in bps) of node ( n_i )</td>
</tr>
<tr>
<td>( b_i )</td>
<td>broadcast rate (in bps) of node ( n_i ) in WiFi network</td>
</tr>
<tr>
<td>( \tau_i )</td>
<td>timeshare (in sec) of node ( n_i ) in WiFi network</td>
</tr>
<tr>
<td>( \psi )</td>
<td>shared session elongation coefficient</td>
</tr>
<tr>
<td>( l_i )</td>
<td>battery level of node ( n_i ) (in J)</td>
</tr>
<tr>
<td>( \lambda, \eta, \mu, \xi )</td>
<td>Lagrange multipliers</td>
</tr>
</tbody>
</table>
(2) \( r \leq s_i + \sum_j d_{ij} b_j (1 - p_{i,j}) \), \( \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \),
(3) \( s_i \leq b_i \min_j (d_{ij} (1 - p_{i,j})) \), \( \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \)
(4) \( s_i \leq d_{ci} (1 - p_c) \), \( \forall n_i \in \mathcal{N} \)
(5) \( b_i \leq \tau_i \min_j c_{i,j} \), \( \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \)
(6) \( \sum_i \tau_i \leq 1, \forall n_i \in \mathcal{N} \)

The first constraint requires the accumulative download rate over 3G links to be larger than the streaming rate \( r \). This is inevitable as the source must send at least one copy of the video into the cooperative network. The second constraint implies that for smooth playback at each node, the accumulative receiving rate from the cellular link and the WiFi broadcast must be larger than the streaming rate \( r \). The third constraint enforces the flow conservation in the WiFi network, and requires each node to broadcast any packet received over the cellular link in the WiFi network. The fourth constraint is the capacity constraint in the cellular network, and specifies an upper bound for the receiving rate of node \( n_i \). The last two constraints define the broadcast rate of each node in the WiFi network. The fifth constraint indicates that a node cannot broadcast at a rate higher than the lowest capacity among its outgoing WiFi links in order to ensure that all nodes can receive the broadcast data during the allocated time slot. As discussed in Sec. III-A2, broadcasting in the WiFi network is a TDMA model. According to this model, \( \tau_i \geq 0 \) in the last constraint is the time share that node \( n_i \) may use the WiFi channel. This formulation minimizes power consumption (in Watt) instead of total energy consumption (in Joule). By definition, \( J = W \cdot s \), i.e., energy consumption grows proportionally with the length of a streaming session. Therefore, for any streaming session, minimizing power consumption will lead to minimized energy consumption.

Although Eqn. 4 minimizes power consumption in the streaming session described in Sec. III-A, it suffers from short lifetime of the shared streaming session. The solution of Eqn. 4 will favour high-capacity nodes, and undoubtedly quickly drains battery of these nodes, leading to lower accumulative capacity in the WiFi network. Once these nodes consume all of their energy and quit from the cooperative network, the energy consumption will increase, as the system now consists of only low-capacity nodes. Our experimental results in Sec. IV also confirm this phenomena. Therefore, we introduce a new constraint to prolong the shared time in the cooperative streaming session as follows:

\[ l_i \geq \psi l_i, \forall n_i \in \mathcal{N} \]  

(5)

where \( l_i \) is the battery power (in Watt) of node \( n_i \), \( \psi \geq 0 \) is a global shared session elongation constant, and \( l \) is the average battery power of cooperating nodes. Clearly, \( \psi = 0 \) turns off the shared session elongation control, \( \psi = 1 \) invites low-capacity nodes to contribute. We then complete the formulation by adding Eqn. 5 to Eqn. 4 and replacing \( P_i(r) \) with Eqn. 3. Moreover, the symmetric property on each link (\( w_{i,j} \approx w_{j,i} \)) allows us to further simplify the model by replacing \( c_{j,i}, p_{j,i}, \) and \( d_{j,i} \) with \( c_{i,j}, p_{i,j}, \) and \( d_{i,j} \), respectively. The standard form of the final problem formulation is presented in Table II.

Since the second order partial derivative of the objective function and all the constraints in Table II equals to zero, this problem is a convex optimization problem. Hence, exhaustive search algorithms will be too complex due to size of the search space. Thus, we solve the problem through its Lagrangian dual function. We note that the fourth constraint in Table II specifies the upper bound for rate allocation in 3G network, and the fifth and sixth constraints specify the upper bound for rate control in the WiFi network. Since these constraints are defined by the network capacity, we keep these three and relax the remaining constraints using a Lagrange multiplier \( \lambda \) for constraint (1) and three Lagrange multiplier vectors \( \eta, \mu, \xi \) for constraints (2), (3) and (7) respectively. Then the Lagrangian of the problem can be reformulated by expanding and reordering the terms towards rate allocation variables, i.e., \( s_i \) and \( b_i \). Thus, we have:

\[ L(r, \lambda, \eta, \mu, \xi) = \sum_i \left[ s_i (\alpha_i - \lambda - \eta_i + \mu_i) + \right. \]
\[ \left. + \sum_i b_i (\beta_i - \mu_i \min_j (d_{i,j} (1 - p_{i,j}))) \right] \]
\[ + \sum_i (\gamma_i - \eta_i) \min_j d_{i,j} b_j (1 - p_{i,j}) \]
\[ + r (\lambda + \sum_i \eta_i) \]
\[ + \sum_i \xi_i (\psi l_i - l_i), \quad \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \]  

(6)

Now, we can define the Lagrange dual function of the energy usage minimization problem as:

\[ D(\lambda, \eta, \mu, \xi) = \min_{s_i, b_i} L(r, \lambda, \eta, \mu, \xi) \]  

(7)

s.t. (1) \( s_i \leq d_{ci} (1 - p_c), \forall n_i \in \mathcal{N} \)
(2) \( b_i \leq \tau_i \min_j c_{i,j} \), \( \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \)
(3) \( \sum_i \tau_i \leq 1, \quad \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \)

To minimize the energy consumption on mobile devices, each node \( n_i \in \mathcal{N} \) should broadcast data at the highest possible rate so that it can keep the antennas of other nodes in the sleep mode as long as possible, i.e., reducing the idle power. This allows us to remove the second constraint in Eqn. 7 and replace \( b_i \) with \( \tau_i \min_j c_{i,j} \) in the Lagrangian \( L \). Now, we have only two variables: \( s_i \) and \( \tau_i \), and the Lagrangian dual problem can be decomposed into two simpler problems to find the optimal rate allocation in the cellular network and the WiFi network. Hence, the optimization problem for the cellular links and WiFi links rate control can be respectively written as:

\[ \min_{s_i} \sum_i \left[ s_i (\alpha_i - \lambda - \eta_i + \mu_i) + r (\lambda + \sum_i \eta_i) \right] \]
\[ + \sum_i \xi_i (\psi l_i - l_i) ; \quad \forall n_i \in \mathcal{N} \]

s.t. \( s_i \leq d_{ci} (1 - p_c), \forall n_i \in \mathcal{N} \)

(8)

\[ \min_{\tau_i} \sum_i \tau_i \min_j c_{i,j} (\beta_i - \mu_i \min_j (d_{i,j} (1 - p_{i,j}))) \]
\[ + \sum_j \tau_j \sum_i (\gamma_i - \eta_i) \min_j d_{i,j} b_j (1 - p_{i,j}) \]
\[ + \sum_i \xi_i (\psi l_i - l_i) ; \quad \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \]

s.t. \( \sum_i \tau_i \leq 1, \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} \)  

(9)

**TABLE II: Energy consumption optimization problem for video streaming in a cooperative network**

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) ( r )</td>
<td>( \sum_i s_i \leq \sum_i \Psi l_i, \forall n_i \in \mathcal{N} )</td>
</tr>
<tr>
<td>(2) ( r )</td>
<td>( \sum_i s_i \leq d_{ci} \sum_i (1 - p_c), \forall n_i \in \mathcal{N} )</td>
</tr>
<tr>
<td>(3) ( s_i )</td>
<td>( \sum_i s_i \leq b_i \sum_j (d_{ij} (1 - p_{i,j})), \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} )</td>
</tr>
<tr>
<td>(4) ( s_i )</td>
<td>( s_i \leq d_{ci} (1 - p_c), \forall n_i \in \mathcal{N} )</td>
</tr>
<tr>
<td>(5) ( b_i )</td>
<td>( b_i \leq \tau_i \sum_j c_{i,j}, \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} )</td>
</tr>
<tr>
<td>(6) ( \sum_i \tau_i )</td>
<td>( \sum_i \tau_i \leq 1, \forall n_i \in \mathcal{N}, n_j \in \mathcal{N}_{n_i}^{-} )</td>
</tr>
</tbody>
</table>
Because the proposed Lagrange dual function does not provide a strong duality, regardless of the value of $\lambda$, $\eta$, $\mu$, and $\xi$, there is a gap between the optimums of the primal problem (Table II) and its Lagrange dual function (Eqn. 7). We use a two-level iterative optimization method to set the values of the Lagrange multipliers such that this gap is minimized, as will be described in the RAS algorithm (Sec. III-C).

### C. The Rate Allocation and Scheduling (RAS) Algorithm

Based on the solution for the power consumption minimization problem, we propose the optimal rate allocation and scheduling (RAS) algorithm for our cooperative streaming system. Before pushing each video segment, the video source checks and updates (if necessary) information about each node $n_i \in \mathcal{N}_i$ including channel state information (CSI), partial WiFi links state information (i.e., $c_{i,j}$ and $p_{i,j}$, if $i < j$), energy consumption coefficients (i.e., $\alpha_i$, $\beta_i$, and $\gamma_i$), and the remaining battery power (i.e., $l_i$). It then formulates the power consumption minimization problem using the information collected from all nodes. To solve the energy consumption minimization problem, we start from a chosen set of values for $\lambda$, $\eta$, $\mu$, and $\xi$ (e.g., by initializing all multipliers to one), and solve Eqs. 8 and Eqs. 9 using a linear solver. We then use subgradient optimization method to update the Lagrange multipliers as follows:

$$\lambda^{t+1} = \max \{0, \lambda^t + \rho^t \left( r - \sum_i s_i^t \right) \}$$

$$\eta^{t+1} = \max \{0, \eta^t + \rho^t_i \left( r - g_i^t - \sum_j d_{i,j} \beta_j^t (1 - p_{i,j}) \right) \}$$

$$\mu^{t+1} = \max \{0, \mu^t_i + \rho^t \left( s_i^t - b_i^t \min_i (d_{i,j} (1 - p_{i,j}) \right) \}$$

$$\xi^{t+1} = \max \{0, \xi^t_i + \rho^t \left( c_i - l_i \right) \}$$

.. where $n_i \in \mathcal{N}$ and $n_j \in \mathcal{N}_i^{-}$, and $\rho^t$ is the step size series. According to Eqn. 10, the Lagrange multiplier $\lambda$ is updated according to the difference between the streaming rate and the accumulative receive rate over 3G links. So $\lambda$ is the size of the source queue that buffers segments to be sent by the source in the 3G network. The Lagrange multiplier $\eta$ specifies the number of missing blocks at node $n_i$ as it is updated according to the number of blocks needed to reconstruct a segment. The Lagrange multiplier $\mu$ is updated according to the difference between the receiving rate on 3G link $g_i$ and the broadcast rate at node $n_i$, so it can be used as the output queue size at node $n_i$. Finally, the Lagrange multiplier $\xi$ has an inverse relation with the battery conservation of node $n_i$, i.e., smaller $\xi$ encourages node $n_i$ to contribute in the download process in lower battery level conditions.

After each round of update on $\lambda$, $\eta$, $\mu$, $\xi$ for all nodes, we update Eqn. 8 and Eqn. 9 with the new values and feed them to the linear solver again. This iterative process repeatedly improves the values of $s_i$ and $b_i$. To make the algorithm converge, we must have $\sum_{i=0}^{\infty} \rho^t = \infty$ and $\lim_{i \to \infty} \rho^t = 0$. One example for such a sequence is $\rho^t = \frac{1}{n^t}$. For faster convergence, we utilize the step size adaptation formula proposed by Held and Karp [21]. We believe that such a centralized solution is feasible in a Cloud-assisted streaming service due to the computing power of the Cloud.

After finding the optimal rate allocation, the source now prepares the video segment by dividing it into $k$ blocks. It then produces coded blocks in GF(2^56) according to the loss rate on the 3G link connecting the chosen node $n_i$ with download rate $s_i$. The coded blocks are sent along with the original blocks over this 3G link. The rate allocation and scheduling information for the WiFi network is also sent to the selected node. Upon receiving $k$ linearly independent blocks of a video segment, the node decodes the original blocks and encodes them again in GF(2) according to its broadcast loss rate. The coded blocks are broadcast along with the original blocks over this WiFi network at rate $b_i$ provided by the RAS algorithm. Upon receiving or overhearing the broadcasted segment, nodes decode the segment and buffer it for playback.

### D. Overhead Analysis

The proposed RAS algorithm requires each active node to send battery usage and battery information to the streaming source. This may be done periodically or just once. If this information is provided to the source only once, the rate allocation on 3G links will not consider power drained due to video decoding, rendering, and playback on the screen. If this information is sent to the source periodically, the rate allocation will be dynamically updated according to the current status of mobile devices. However, if bandwidth is scarce in the network or signals are weak, the one-time update is preferred to reduce communication overhead in the cellular network.

When providing update to the source, a node $n_i$, in a cooperative network consisting of $N$ nodes, collects information $(c_{i,j}, g_{i,j}, p_{i,j}, d_{i,j})$ of WiFi links between itself and each of the $N - 1$ nodes, the information $(c_{i}, p_{i}, d_{i}, s_{i})$ on the 3G link $g_{i}$ from the source to itself, as well as its broadcast rate $b_i$ in the WiFi network and its battery level $l_i$. Each of these values can be stored in 4 bytes. Hence, the size of an update message is $4 \times (4(N - 1) + 6)$ bytes. For instance, in a cooperative network that consists of 10 nodes and all nodes are active nodes, an update from a node is 168 bytes in size. Overall, there are 1680 bytes flowing from the cooperative network to the source for each update period. Assume that the update period is 10 seconds, which is frequent enough. The update shares 1680 Bps of the bandwidth. If we assume the bit rate for a typical streaming session 1 Mbps, the communication overhead is 0.13% if the source serves only one copy of the video to the cooperative network. Bear in mind that without the RAS algorithm, the source will stream more than one copy of the video to the active nodes over the cellular network. Therefore, we argue that this overhead is negligible. Furthermore, we note that the periodic update will not lead to extra transmission delays, as the streaming source can use the previous RAS results before receiving any new update.

At last, we consider mobility of the mobile devices. Over time, the cooperative group may move from cell to another cell in the cellular network. For quick handover, the old base station sends the new base station link information $(w_{i,j}, c_{i,j}, p_{i,j}, d_{i,j})$ for each of the $N^2$ WiFi links, the Lagrange multipliers $(\eta, \mu, \xi_i)$ for each of the $N$ nodes and $\lambda$, as well as the broadcast rate $b_i$, the share time $\tau_i$, and the battery level $l_i$ of each node in the WiFi network. Each of these values needs 4 bytes. Hence, the size of an update message is $4N^2 + 6N + 1$ bytes. For instance, in a cooperative network that consists of 10 nodes and all nodes are active nodes, the old base station sends the new base station 1844 bytes during the handover. Again, this overhead is considered negligible compared to the large volume of the streaming traffic.

### IV. Performance Evaluation

In this section, we evaluate the performance of the proposed system, especially the effectiveness of the RAS algorithm. In all experiments, unless specified otherwise, we assume the packet loss rate of each 3G link and the WiFi broadcast channels are $p_l = 0.2$ and $p_w = 0.2$, respectively. The average round trip time in the 3G network and the WiFi network are 237 ms and 89 ms, respectively.
To evaluate our system in a close-to-reality setting, we use the energy profile from a real smartphone, Samsung Galaxy Nexus I9250, to drive the simulation. This phone is equipped with a dual-core 1.2 GHz processor, 1 GB of memory, and a 3.7 V 1750 mAh Li-Ion battery that provides 23.31 KJ of energy. The operating system is Android 4.2 (Jelly Bean). To collect the energy profile, we transmit (upload and download) a large file using this phone in different network settings and monitor the energy use. The energy consumption is measured three times using a fully charged phone, and the average is reported in Table III. For transmissions in 3G network, we control the download rate of the phone using a software. We will use the energy profile of the phone with different download rates to simulate different types of nodes in the cooperative network. According to Table III, type I node achieves the maximum throughput, and the throughput lowers from type II to type IV. We note that nodes with lower throughput tend to be less energy efficient, as it will take longer to transmit the same file. Type V node represents a passive node that does not have any 3G connection, and relies on other types of nodes for the streaming service. From Table III, we also observe that data transmission over WiFi has higher throughput and consumes less energy, which justifies the benefits of offloading the cellular transmission to the WiFi cooperative network formed by mobile devices.

<table>
<thead>
<tr>
<th>Throughput</th>
<th>Energy Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>GF(2), Encoding</td>
<td>393.58</td>
</tr>
<tr>
<td>GF(2), Decoding</td>
<td>408.16</td>
</tr>
<tr>
<td>GF(256), Encoding</td>
<td>5.03</td>
</tr>
<tr>
<td>GF(256), Decoding</td>
<td>7.04</td>
</tr>
</tbody>
</table>

To measure the energy consumption of coding operations over GF(2) and GF(256), we use NCUutils [22], a network coding library written in Java. When profiling the coding throughput and the energy efficiency on the phone, file size and block size are chosen such that the coding throughput and the linear independence among the coded blocks are maximized. Table III shows that coding operations in GF(2) are faster and more energy efficient due to their linear computational complexity, whereas the coding operations in GF(256) have quadratic complexity. The communication overheads (coding coefficients and extra coded blocks) of network coding in GF(2) and GF(256) are 7.03% and 3.125%, respectively. The higher overhead in GF(2) is the result of high dependency of coded blocks, i.e., extra coded blocks must be generated and sent to guarantee 99.990% of decodability. Based on the energy profile in Table III, even with the extra overhead, sharing coded blocks in GF(2) is still a green choice as coding operations are significantly more energy efficient than that of GF(256).

In our simulation, we measure the energy consumption (in kJ) instead of power consumption (in kW), as it presents the total energy consumed throughout a streaming session. To provide a clear view on the energy saving on mobile devices, we take the average of the energy consumption consumed by each mobile device in the system. We also measure the streaming delay (in ms), defined as the time taken to have all nodes a segment ready for playback, starting from the time a segment is scheduled by the RAS algorithm for transmission.

### A. Cooperative Streaming using Different Coding Strategies

We begin the study with an investigation on the effect of different coding strategies on the power consumption in a cooperative network of homogeneous devices, i.e., the cooperating nodes are all from type I, as defined in Table III. We turn off the shared session elongation constraint (i.e., $\psi = 0$) in order to focus on the impact of different coding strategies and cooperative arrangements. We feed the simulated source a high-definition video of length 6077 seconds with bitrate 4.36 Mbps. For comparison purpose, we implement four different cooperation schemes listed below.

- **Streaming over 3G**: In this scheme, all mobile devices download the video directly over 3G downlinks. There is no cooperation among mobile devices.
- **Cooperation without network coding**: In this scheme, mobile devices cooperate over WiFi, but no coding is employed, i.e., $d_{i,j} = d_{i,j} = 1, \forall i \in N, j \in N_i$. Without network coding, all nodes must perform data reconciliation by contacting individual nodes for particular missing segments.
- **Cooperation using RLNC**: In this scheme, random linear network coding is employed over both cellular and WiFi links and coded blocks are always coded in GF(256). This scheme is proposed in [3].
- **Cooperation using two-level NC**: In this scheme, systematic Reed-Solomon codes and systematic Fountain codes are used in the cellular and WiFi networks, respectively [5]. Our system uses this scheme.

In this experiment, we first vary the size of the cooperative network from 1 node to 20 nodes. Fig. 2 shows that when streaming over 3G, the energy used by each node is constant. Compared to cooperation using RLNC, noticeable energy saving is offered by our system, primarily due to the smaller field size and the systematic network coding utilized in two-level NC scheme. The cooperation with RLNC employed in Microcast [3] consumes much more energy due to its coding complexity, while our system consumes almost no extra energy compared to the no-coding scheme with 10 or less nodes. If the system consists of more than 10 nodes, in the absence of network coding, the limited WiFi capacity forces the nodes to use 3G downlinks to receive missing packets, leading to increased battery consumption of no coding scheme.

Fig. 3 shows a break down of the average energy consumption. This confirms that the cooperation among mobile devices greatly reduces the energy usage due to 3G transmissions. Both RLNC and two-level NC minimize the traffic and the respective energy usage in the cellular and WiFi networks. The simplified coding operations in two-level NC consume very small amount of energy without consuming considerable extra energy due to slightly increased WiFi transmissions.

### B. Effectiveness of the optimal RAS Algorithm

In this experiment, we compare the RAS algorithm with heuristic algorithms suggested in [3] and [5]:

- **Aggressive collaboration**: Nodes download from 3G downlink whenever possible.
- **Equal collaboration**: Each node downloads the same amount of video content over the cellular links.
- **Battery centric**: The amount of data downloaded by a node over 3G links is proportional to its remaining battery lifetime.
The experiments are conducted in various heterogeneous cooperative networks consisting of 20 nodes of types II—V defined in Table III with equal probability. To imitate the real-world model of battery charges, we use a normal distribution with $\mu = 67$ and $\sigma^2 = 10$ [23] to model the remaining battery lifetime. Again, we turn off the shared session elongation constraint ($i.e., \psi = 0$) in order to focus on the impact of different rate allocation algorithms. In order to ensure that almost all nodes have sufficient bandwidth on their cellular links to sustain the streaming rate, we feed the simulated source a high-definition video of length 5482 seconds with a lower bitrate 1.61 Mbps.

Fig. 4 shows that, compared to the aggressive, equal, and battery-centric collaboration algorithms, our proposed RAS algorithm saves up to 15.0%, 22.4%, and 39.7% of energy, respectively. It is interesting that the battery-centric collaboration algorithm, the most intuitive heuristic approach to conserve battery, is the least energy-efficient solution.

Next, we compare the streaming delay offered by each scheduling algorithm. Fig. 5 shows that the aggressive scheduling offers the least delay as each node downloads as fast as possible from the source. However, the tradeoff is less remaining battery lifetime. The battery-centric scheduling algorithm is the worst among all four algorithms, which also explains why this algorithm leads to higher energy consumption. For example, according to Table III, it costs type II nodes much less power to download segments over 3G links than type IV nodes do. Hence, the longer it takes to download a segment, the more power is consumed. Our optimal RAS algorithm approximates the case best very well, as it schedule nodes to download segments over 3G links according to their 3G download rate $s_i$ and energy efficiency (KJ/GB), $i.e.,$ it prefers type II nodes over type III nodes and type III nodes over type IV nodes. In summary, our algorithm outperforms equal collaboration and battery-centric algorithms by 4.0% and 27.1% less transmission delay, respectively. Although it incurs 13.9% extra average delay compared to aggressive scheduling, due to its tendency to conserve battery power on mobile devices.

C. Impact of the Session Elongation Constraint

At last, we turn our attention to the impact of the shared session elongation constraint on average energy usage, video segment transmission delay, and the uptime of mobile devices. We resort to the same setting as in Sec. IV-B, and reduce the network size to 7 nodes to allow close examination of individual nodes. The type and the default battery level of each node is listed in Table IV. In this experiment, we vary the value of shared session elongation coefficient ($i.e., \psi$) from 0 to 1.0. As discussed in Sec. III, $\psi = 0$ removes the constraint and $\psi = 1$ enforces the mobile devices with higher battery level to use the more expensive 3G downlink to receive the video segments and serve it in the WiFi network.

Fig. 6 shows the decrease in energy level (computed based on the battery level) throughout a streaming session for four different values of the shared session elongation coefficient $\psi$. Fig. 6(a) shows that without this constraint, nodes with low energy efficiency ($e.g.,$ nodes of types III and IV) live longer as the optimal RAS algorithm slaves the high energy-efficient nodes to deliver the streaming content in the WiFi network. Although there are three nodes still alive beyond 16 hours, the other four die as early as 6 hours. Hence, the cooperative session for the entire network ends after 6 hours, although some devices live much longer. The average streaming duration, denoted by $D$, is about 14 hours. According to Fig. 6(b), 6(c) and 6(d), we increase the session elongation coefficient $\psi$, the lifetime of all nodes converges to approximately 16 hours. The longest streaming session can be achieved when $\psi = 1$. The only tradeoff is that nodes 5, 6, and 7 live shorter as they are invited to contribute their battery to assist the streaming session. Nonetheless, the entire group can now have a longer streaming session. Please notice that in Fig. 6(a) and Fig. 6(b) node 7 cannot stream the reference video alone as its downlink capacity is less than the video bit rate.

Fig. 7 compares the average transmission delay of video segments throughout the streaming sessions simulated using different values for shared session elongation coefficient. We observe that higher $\psi$ value leads to longer transmission delays as the optimal RAS algorithm will utilizes slower nodes to avoid fast battery drainage on the powerful nodes. Among all nodes, the nodes with low throughput values will be slow in delivering their segments. However when $\psi = 1$, the delay decreases after a while, as the nodes approach the same battery level and more nodes are involved in the segment download process. Contrarily, when $\psi = 0$, the delay increases after a while due to the decease of powerful nodes. The overall average transmission delay for $\psi = 0, \psi = 0.33, \psi = 0.66$, and $\psi = 1.0$ are measured as 683 ms, 732 ms, 746 ms, and 809 ms, respectively. The increase in average delay when $\psi$ changes from 0 to 1 is less than 126 ms, which is not sufficient enough to degrade the quality of the streaming session.

| Table IV: Specification of heterogeneous nodes |
|-----------------|---|---|---|---|---|
| Node Type       | I | II | III | IV | V |
| Battery Level (%)| 71.4 | 66.3 | 37.2 | 60.5 | 89.4 | 55.1 | 77.9 |

Fig. 2: Impact of cooperation arrangements on energy consumption

Fig. 3: A break down of the average energy consumption

Fig. 4: Effectiveness of the RAS algorithm

Fig. 5: Average transmission delay of video segments offered by different scheduling algorithm

Fig. 6(a) shows that without this constraint, nodes with low energy efficiency ($e.g.,$ nodes of types III and IV) live longer as the optimal RAS algorithm slaves the high energy-efficient nodes to deliver the streaming content in the WiFi network. Although there are three nodes still alive beyond 16 hours, the other four die as early as 6 hours. Hence, the cooperative session for the entire network ends after 6 hours, although some devices live much longer. The average streaming duration, denoted by $D$, is about 14 hours. According to Fig. 6(b), 6(c) and 6(d), we increase the session elongation coefficient $\psi$, the lifetime of all nodes converges to approximately 16 hours. The longest streaming session can be achieved when $\psi = 1$. The only tradeoff is that nodes 5, 6, and 7 live shorter as they are invited to contribute their battery to assist the streaming session. Nonetheless, the entire group can now have a longer streaming session. Please notice that in Fig. 6(a) and Fig. 6(b) node 7 cannot stream the reference video alone as its downlink capacity is less than the video bit rate.
proportional contribution among all nodes effectively prolongs the session elongation constraint shows that enforcing energy saving on mobile devices. Our study on the impact compared to previously proposed heuristic rate allocation and evaluate the system using a simulator driven by energy pro-

V. CONCLUSION

In this paper, we proposed a cooperative streaming system that is designed based on the RAS algorithm, the optimal rate allocation and segment scheduling for minimizing both the streaming traffic in the cellular network and the energy consumed by streaming applications on mobile devices. More specifically, the RAS algorithm determines the number of segments and the actual segments to be transmitted on each link. The actual delivery of segments are achieved through a two-level coding scheme that ensures an energy-efficient recovery of the streaming content on mobile devices. We evaluate the system using a simulator driven by energy profile from real devices. Our experimental results show that, compared to previously proposed heuristic rate allocation and scheduling algorithms, the RAS algorithm leads to significant energy saving on mobile devices. Our study on the impact of the session elongation constraint shows that enforcing proportional contribution among all nodes effectively prolongs the streaming session for the entire cooperative group.

REFERENCES


