Forty-Ninth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 28 - 30, 2011

Cooperative Video Streaming on Smartphones

Hulya Seferoglu†, Lorenzo Keller‡, Blerim Cici†, Anh Le†, Athina Markopoulou†
†University of California, Irvine, CA, USA
‡EPFL, Lausanne, Switzerland

***Abstract*—Video applications are increasingly popular over smartphones. However, in current cellular systems, the downlink data rate fluctuates and the loss rate can be quite high. We are interested in the scenario where a group of smartphone users, within proximity of each other, are interested in viewing the same video at the same time and are also willing to cooperate with each other. We propose a system that maximizes the video quality by appropriately using all available resources, namely the cellular connections to the phones as well as the device-to-device links that can be established via Bluetooth or WiFi. Key ingredients of our design are: (i) the cooperation among users, (ii) network coding, and (iii) exploiting broadcast in the mobile-to-mobile links. Our approach is grounded on a network utility maximization formulation of the problem. We present numerical results that demonstrate the benefit of our approach, and we implement a prototype on android phones.**

I. Introduction

The significant progress in video compression techniques, wireless data communication, and cross layer design are continuously advancing the state-of-the art in wireless video. On one hand, the data transmission rates of wireless networks are steadily growing, *e.g.,* 100Mbps for mobile users in 4G systems [1]. On the other hand, H.264/MPEG4-AVC [2] achieves more efficient video compression and the Scalable Video Coding (SVC) extension [3] of H.264/MPEG4-AVC obtains both high coding efficiency and high scalability.

However, providing high quality video over wireless networks is still a challenging problem, because the demand for video applications over current networks is exponentially increasing [4]. However, in current 3G/4G systems, the downlink data rate fluctuates and the service loss rate can be as high as 50% [5]. This makes it challenging to provide high quality video to mobile users without interruptions.

In this paper, we are interested in the scenario where a group of smartphone users (or "nodes"[1]), within proximity of each other, are interested in viewing the same video at the same time. We seek to maximize the video quality by cooperatively using all available resources, namely: the cellular connections to all users and mobile-to-mobile links that can be established through Bluetooth or WiFi. Key ingredients of our design are the following:

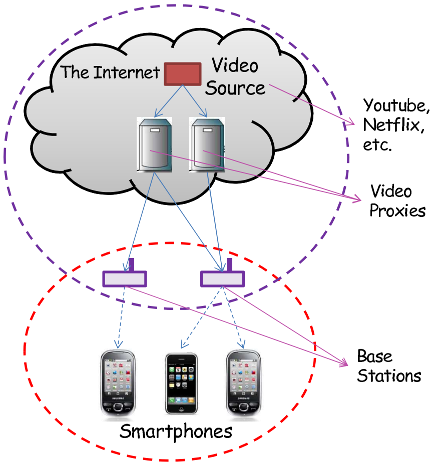


Fig. 1. *Cooperative video streaming system.* A group of smartphone users, within proximity of each other, are interested in viewing the same video at the same time. Each smartphone has Internet connection via 3G or 4G. When a user is interested in viewing a video, it connects to the video source (*e.g.,* YouTube or Netflix) via its base station, which may be the same or different for different users, depending on the provider their use. The proxy, in our system, is responsible for selecting the video rate and performing network coding. Each smartphone receives packets from the source (via the base station and video proxy) as well as from other smartphones in the neighborhood, through device-to-device (Bluetooth or WiFi) links.

1) *Cooperation* among users, in the use of both the downlinks and the local links.[2]
2) *Network coding* used on the downlink (from the proxy to the users) as well as on the local links (between users).
3) Exploiting the *broadcast* nature of the wireless channel on the local links (while the downlinks are unicast 3G).

This scenario is illustrated in Fig. 1.

In designing such a system, a number of questions need to be addressed. (1) What should be the video rate at the source? (2) How to use the downlink connections? They could download the same or different packets, or they could use some level of redundancy. In this paper, we choose to use the downlink connections in parallel and have them download different parts of the video, so as to maximize the aggregate download rate. (3) At which rate should a node transmit packets to its neighbors and which packets should it transmit? (4) What information needs to be exchanged between the nodes and the source/proxy in order to make these decisions?

This work has been supported by an AFOSR MURI (FA9550-09-0643), the NSF CAREER (0747110) and by ArmaSwisse W+T (Project No. 8003413832).

[1] We use the terms "smartphone", "user" and "node" interchangeably.

[2] In this paper, we consider downlink connections (3G) that do not interfere with the local links (WiFi or Bluetooth). We extend our scheme to the case where downlink and local links compete for resources in [6]. This is, for example, the case when smartphones download video from a WiFi access point and also exchange data with each other via WiFi.

978-1-4577-1818-2/11/$26.00 ©2011 IEEE

(5) Should network coding be used at the proxy and/or at the local links? (6) How much benefit do we get from using the local links as broadcast vs. unicast? Is broadcast/overhearing practical in a real implementation?

Our approach is grounded on a network utility maximization formulation of the problem. We consider several variants of the problem, depending on whether the local links are used as unicast or broadcast, and on whether network coding is used. The solution of each problem decomposes into several parts with an intuitive interpretation. We perform numerical calculations for a range of parameters, and we show that the scheme that combines all three ingredients, namely *Cooperation & Broadcast & NC*, outperforms all other schemes. Finally, we have implemented a prototype of these systems on android phones. We briefly mention some challenges we faced in going from theory to practice, but we omit the description of the android testbed.

The structure of the rest of the paper is as follows. Section II presents related work. Section III gives an overview of the system. Section IV presents the NUM formulation and solution. Section V briefly discusses some of the challenges we faced in implementing our scheme on androids. Section VI presents numerical calculations that demonstrate the superior performance of the *Cooperation & Broadcast & NC* scheme in range of scenarios; we also report preliminary results from the android testbed. Section VII concludes the paper.

II. Related Work

Network coding for peer-to-peer systems. Network coding makes distributed scheduling easier and, as a result, it improves the efficiency of content distribution [8], and live peer-to-peer networks in [9] [10]. An excellent review on network coding-based peer-to-peer networks is presented in [11].

Network utility maximization (NUM) of coded systems. The NUM framework [12], [13] can be applied in networks with network coding, in order to understand how different layers and/or functional modules (such as flow control, congestion control, routing, etc.) should be modified when network coding is used. The problem of establishing minimum-cost multicast connections over coded wired and wireless networks is considered in [14] and was extended for end-to-end rate/congestion control over wired coded networks in [15]. A cross-layer optimization framework including routing and scheduling to maximize throughput over coded wireless mesh networks for multicast flows is studied in [16]. Linear optimization models for computing a high-bandwidth routing strategy for media multicast in coded wireless networks are proposed in [17].

The NUM framework has also been applied to peer-to-peer networks with network coding. In [18], the aggregate application-specific utility is maximized by distributed algorithms on peers, which are constrained by their uplink capacities. [20] extends [18] by considering node-capacities and constraints on both node upload capacity and node download capacity. In [19], the authors derive performance bounds for minimum server load, maximum streaming rate, and minimum tree depth under different peer selection constraints, but without network coding. The optimal bandwidth sharing in multi-swarm multiparty peer-to-peer video-conferencing systems with helpers is considered in [21]. Multi-rate peer-to-peer multi-party conferencing applications, where different receivers in the same group can receive videos at different rates using, for example, scalable layered coding are considered in [22]. The Implicit-Primal-Dual scheme for flow control in live streaming peer-to-peer systems is introduced in [23]. [24], which is the closest to our work, proposes a scalable video broadcast/multicast scheme that efficiently integrates scalable video coding, 3G broadcast, and ad-hoc forwarding so as to balance the system-wide and worst-case video quality of all viewers at 3G cell. Different from this work we consider (i) 3G links as multiple unicast transmissions, while in [24], multicast links are assumed, and (ii) network coding. Note that considering multiple unicast links are crucial in practice because current 3G/4G systems only support unicast transmissions.

Network coding for error correction and local cooperation. Wireless links (3G/4G or WiFi) suffer from packet loss due to noise and interference. One possible solution to tackle this problem is to have several devices in a close proximity help each other with retransmissions of lost packets. In this case, network coding is particularly beneficial because a linear combination is more useful to more than one nodes. Rate-distortion optimized network coding for cooperative video system repair in wireless peer-to-peer networks is considered in [25]. Wireless video broadcasting with peer-to-peer error recovery is proposed in [26]. An efficient scheduling approach to network coding for wireless local repair is introduced in [27]. [28] proposes a cooperative IPTV system with pseudo-broadcast to improve reliability. [29] proposes a system, in which a group of smartphone users (which are connected to the Internet via LTE links) help each other for error correction: base stations broadcast packets as in [24]. Although Multimedia Broadcast Multicast Services (MBMS) are provisioned for LTE [1], they are currently not implemented.

Network coding on smartphones and WiFi testbeds. The practicality of random network coding over iPhones is discussed in [31]. A toolkit to make network coding practical for system devices from servers to smartphones is introduced in [32]. A gesture broadcast protocol is designed for concurrent gesture streams in multiple broadcast sessions in [33] over smartphones using inter-session network coding. Our work is different in (i) the application scenario, *i.e.,* cooperative video streaming on smartphones using network coding, (ii) the fact that we combine application layer network coding with broadcast in the local connections, and (iii) the fact that we combine NUM and implementation. There are also WiFi testbeds that implement network coding, such as the COPE testbed [30]. However some of the functionality needed is much more challenging on androids than on laptops. For example, to support pseudobroadcast, we had to implement a sniffer to pass overheard packers to the application layer.

Our work in perspective. The contributions of this work lie: in (i) the particular combination of the three ingredients of cooperation, network coding and (pseudo)broadcast and in (ii)

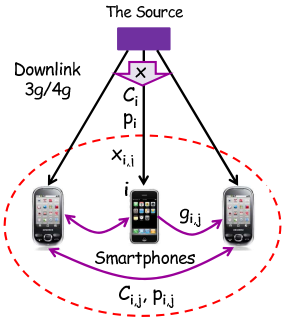


Fig. 2. A simplified model of the video streaming system. Here, the "source" corresponds to the video source and a video proxy. The source transmits a video flow to a set of smartphones; $\mathcal{N}$.

the combination of analysis and implementation. In summary, the main differences from prior work are the following:

- We consider each cellular connection as a separate unicast, which is the case in today's 3G/4G systems. In contrast, [24], [29] considers multicast on the downlink.
- In prior work [25], [26], [29], cooperation is used for error recovery in the local area. In addition, in our scheme, nodes cooperate in the use of their downlink as well: nodes download different parts of the stream through their cellular connection and share it with their neighbors through local links.
- We consider that nodes cooperate for the use of both the 3G/4G links and the local area (WiFi/Bluetooth) links.
- We propose a practical scheme based on the structure of the NUM solution, and we implement a proof-of-concept prototype on androids.
- We employ pseudo-broadcast in local WiFi links (instead of broadcast) to use wireless media efficiently. To the best of our knowledge, this is the first time that this is done specifically on smartphones.

In this paper, we focus on the NUM formulation and solution. Additional materials (including convergence results, packet-based implementation based on the structure of the NUM solution, formulation when WiFi is used on both downlink and local links) is in [6]. The android testbed is ongoing work; information can be found on the project webpage [7].

III. System Overview

In the rest of the paper, we consider a system model presented in Fig. 2: the source transmits a video to a set of smartphones; $\mathcal{N}$. This is a simplified version of Fig. 1 in that the "source" represents the video source, the proxy and the base station. This allows us to focus on the bottlenecks of the system, namely the downlinks from the base station to the smartphones and the local area links. The links between the source and a video proxy and the links from the video proxy to the base stations are high capacity, low delay links, thus not the bottleneck.

A. Notation and Setup

1) Source and Flows: The source transmits a video flow to a set of nodes $\mathcal{N}$ with rate x. The video flow is associated with a utility function $U(x)$, which we assume to be a strictly concave function of x.

2) Wireless Transmission & Loss Model:

Cellular Links: Each node $i \in \mathcal{N}$ is connected to the source via cellular (3G/4G) link with capacity C_i and loss probability p_i. Note that since the nodes may connect to different base stations and the interference of cellular links are handled by the base-stations, we consider that the downlinks are interference free. The interference implicitly affects the capacity of the link, and we consider that the link capacity information is available in our analysis. Consequently, we consider $|\mathcal{N}|$ parallel interference-free links, operating simultaneously, from the source to each node.

Local Area Links - WiFi: Each node $n \in \mathcal{N}$ is connected to each other in the local area. The capacity between smartphone i and smartphone j is $C_{i,j}$ and the loss probability is $p_{i,j}$. We consider the interference model in [34]: each node can either transmit or receive at a time and all transmissions in the range of the receiver are considered as interfering. Note that since we consider a group of nodes (*i.e.,* smartphones) within proximity of each other, we do not consider multi-hop packet transmissions in the local area. Therefore, any transmission in the local area interferes with any other transmission, and only one node can transmit at a time.

Local Area Links - Bluetooth: The nodes $n \in \mathcal{N}$ in a cooperating group form a piconet in which one node behaves as a master and the others as slaves. One slave can transmit to another slave via the master. The master coordinates the frequency hopping pattern as well as who should transmit at a time (according to time division multiplexing). Thus, only one node can transmit at a given time. Yet, piconets are limited up to eight nodes (one master and seven slaves). Therefore, if the number of nodes in a cooperating group increases, we consider multiple piconets, as known as scatternets. Scatternets allow the master in one piconet to operate as a slave in another piconet. In scatternets, only one node can transmit at a time slot, because each node would like to receive data from each other in a cooperating group. In summary, each node in a piconet (or multiple piconets) transmit data in a time slot allocated to this node at rate C_b and with loss probability p_b (which is in general very small, *e.g.,* 1%).

Loss Model: In our formulation and analysis, we assume that p_i, $p_{i,j}$, and p_b are i.i.d. according to a uniform distribution. In practice, the channel model may follow a different (and most probably time varying) distribution. However, our system implementation does not need to know the loss probabilities or probability distributions [6].

B. Network Coding & Cooperation

1) Network Coding: In our NUM formulation, we consider that the size of video is infinite, and each packet is a linear combination of all packets in the video. In practice, we consider the commonly used generation-based network coding [35]: packets from a flow are divided into generations (note that we use "generation" and "block" terms interchangeably), with size G. At the source, packets within the same generation are linearly combined (assuming large enough field size) to generate G network coded packets. Network coding makes the process sequence agnostic, which allows each receiving node to be able to decode packets from a generation with high probability as long as it receives G packets.

The source divides video flow into generations. The set of packets in the g-th generation is $\mathcal{G}_g$ which consists of $G_g = |\mathcal{G}_g|$ packets. The source uses random linear network coding [36], in which packets are coded by using coefficients from a large enough finite field. The set of coded packets is $\mathcal{G}_g^{'}$. The source does not generate one set of coded packets $\mathcal{G}_g^{'}$ to transmit a set of $\mathcal{N}$ nodes, but a different set for each node. In other words, all network coded packets from all sets are linearly independent); *i.e.,* $\mathcal{G}_g^{'1}$, $\mathcal{G}_g^{'2}$, ..., $\mathcal{G}_g^{'N}$ where $N = |\mathcal{N}|$. This allows us to utilize the multiple 3G/4G links in parallel.

2) Cooperation: A number of nodes which are interested in the same video content construct a cooperating group. In general, the nodes can join to and leave the group according to some rules [6]. However, in this paper, we consider that all of the nodes cooperate to construct a single group and are all trusted, *i.e.,* do not engage in malicious behavior. We do not consider multi-hop transmission, because the nodes are in the same geographical area with close proximity.

Cooperation Policy: In the NUM formulations, we consider two transmission policies: *broadcast* and *unicast*. Broadacst can be achieved over WiFi, and more efficiently over pseudo-broadcast as discussed in Section V, but not over Bluetooth. Unicast can be achieved over either WiFi or Bluetooth.

Each node $i \in \mathcal{N}$ receives packets from the source or from its neighbors, stores them in its receive queue and decodes them if it is possible. At the same time, the received packets are inserted in an output queue; node i maintains an output queue for each neighboring node (*i.e.,* $j \in \mathcal{N},\ j \neq i$). When a transmission opportunity arises for node i (we consider standard MAC protocols, *e.g.,* 802.11, and Bluetooth 2.1+EDR), the node transmits a packet from an output queue to the corresponding node. We note that a node transmits packets that it receives from the source to all other nodes; this is because nodes downloads different packets from the source. However, a node does not re-transmit packets received from other neighbors; this would be redundant since all nodes can hear all local transmissions.

IV. Optimization Formulation

In the system described in Section III, the source transmits video with rate x. For node $i \in \mathcal{N}$, we consider N different rates; $x_{i,1}, x_{i,2}, ..., x_{i,j}, ..., x_{i,N}$ where $j \in \mathcal{N}$. The rate $x_{i,j}$ is the rate of data transmitted from the source to node i to help user j. Our goal is to maximize the utility $U(x)$, which is a strictly concave function of the video source rate x.

A. Formulation

1) Cooperation & Broadcast: We consider the case that broadcast is available in the local area and that packets are network coded (at the source/proxy and in the local area).

$$
\begin{aligned}
\text{P1: } \max_{x}\ & U(x) \\
\text{s.t. } & \sum_{i\in\mathcal{N}} x_{i,j} - x \geq 0,\ \forall j \in \mathcal{N} \\
& g_{i,j} - x_{i,j} \geq 0,\ \forall i \in \mathcal{N}, j \in \mathcal{N} - \{i\} \\
& x_{i,j} \leq C_i(1-p_i),\ \forall i \in \mathcal{N}, j \in \mathcal{N} \\
& g_{i,j} \leq \sum_{\mathcal{J}|j\in\mathcal{J}} f_{i,\mathcal{J}},\ \forall i \in \mathcal{N}, j \in \mathcal{N} - \{i\} \\
& f_{i,\mathcal{J}} \leq \min_{j\in\mathcal{J}}\{C_{i,j}(1-p_{i,j})\}\tau_{i,\mathcal{J}},\ \forall i \in \mathcal{N}, J \in \mathcal{H} \\
& \sum_{i\in\mathcal{N}}\sum_{J\in\mathcal{H}} \tau_{i,\mathcal{J}} \leq \gamma
\end{aligned}
\tag{1}
$$

The first constraint requires that the total received data by node j should be larger than the targeted video rate. The second constraint is the flow conservation in the local area: the outgoing rate from node i to j, $g_{i,j}$ should be larger than or equal to incoming rate $x_{i,j}$. The third constraint is the capacity constraint in the downlink. Note that the packets for the data transmissions (*i.e.,* with rates $x_{i,i}$ and $x_{i,j}$) can the same packets. Therefore, $\max_j\{x_{i,j}\} \leq C_i(1-p_{i,j})$, which is equivalent to the third constraint. $f_{i,\mathcal{J}}$ in the fourth constraint is the flow rate transmitted from node i to a set of nodes $\mathcal{J}$.

If network coding is not employed in the fifth constraint, it should be; $f_{i,\mathcal{J}} \leq \min\{C_{i,j}\}\prod_{j\in\mathcal{J}}\{(1-p_{i,j})\}\tau_{i,\mathcal{J}},\ \forall i \in \mathcal{N}, J \in \mathcal{H}$. The reason is that transmissions should be successful over all links from i to $j \in \mathcal{J}$ (it is why the product term is used) for successful broadcast if network coding is not used. However, when network coding is used, each transmission is beneficial to any node that receives correctly, independently of other transmissions. We refer to this scheme as *Cooperation & Broadcast & No-NC*. The fourth constraint requires that $g_{i,j}$ should be less than the total of flow rates $f_{i,\mathcal{J}},\ j \in \mathcal{J}$, over all hyperarcs J that lead from i to j. The fifth constraint is the capacity constraint in the local area, and the last constraint is the time sharing constraint. We would like to note that the fifth constraint assumes that network coding is employed. We call the solution of P1 as *Cooperation & Broadcast & NC*.

2) Cooperation & Unicast: Let us assume that unicast connections are used in the local area (*i.e.,* over WiFi and Bluetooth links). The NUM formulation is as follows;

$$
\begin{aligned}
\text{P2: } \max_{x}\ & U(x) \\
\text{s.t. } & \sum_{i\in\mathcal{N}} x_{i,j} - x \geq 0,\ \forall j \in \mathcal{N} \\
& g_{i,j} - x_{i,j} \geq 0,\ \forall i \in \mathcal{N}, j \in \mathcal{N} - \{i\} \\
& x_{i,j} \leq C_i(1-p_i),\ \forall i \in \mathcal{N}, j \in \mathcal{N} \\
& g_{i,j} \leq C_{i,j}(1-p_{i,j})\tau_{i,j},\ \forall i \in \mathcal{N}, j \in \mathcal{N} - \{i\} \\
& \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}-\{i\}} \tau_{i,j} \leq \gamma
\end{aligned}
\tag{2}
$$

Note that the first three constraints of Eq. (2) are the same as Eq. (1). The fourth constraint is the capacity constraint in the local area. The transmission rate from node i to j, *i.e.,* $g_{i,j}$

should be less than the capacity of the link and the percentage of time that the link is used for that transmission, *i.e.,* $\tau_{i,j}$. Note that the fourth constraint is correct for both WiFi and Bluetooth. In particular for Bluetooth, the fourth constraint can be considered as $C_{i,j} = C_b$ and $p_{i,j} = p_b$ for $\forall i \in \mathcal{N}, j \in \mathcal{N}$. The final constraint is the time sharing constraint, *i.e.,* time sharing parameters $\tau_{i,j}$ should be summed up to a provisioning factor γ. We assume that all the nodes in a cooperating group are interfering with each other. This is expected, because all of the nodes (*i.e.,* smartphones) are in the same geographical area and close to each other.

Note that, without network coding, this problem is equivalent to P2 in Eq. (2). Network coding may still improve throughput since it reduces overhead. However, we do not consider the effect of overhead in our formulations. Therefore, we do not consider *Cooperation & Unicast & NC* and *Cooperation & Unicast & No-NC* separately. Instead, we refer as *Cooperation & Unicast* to the solution of P2.

B. Solution

Let us first consider the solution for P1 in Eq. (1). By relaxing the first and the second constraints in Eq. (1), we have $L(x, \lambda, \eta)$

$$\begin{aligned}
&= U(x) + \sum_{j\in\mathcal{N}} \lambda_j (\sum_{i\in\mathcal{N}} x_{i,j} - x) + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}-\{i\}} \eta_{i,j}(g_{i,j} - x_{i,j}) \\
&= U(x) + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \lambda_j x_{i,j} - x \sum_{j\in\mathcal{N}} \lambda_j + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}-\{i\}} \eta_{i,j} g_{i,j} \\
&- \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}-\{i\}} \eta_{i,j} x_{i,j} \\
&= U(x) - x \sum_{j\in\mathcal{N}} \lambda_j + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} x_{i,j}(\lambda_j - \eta_{i,j}) \\
&+ \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \eta_{i,j} g_{i,j} \qquad (3)
\end{aligned}$$

We assume that $\eta_{i,j} = 0$ if $i = j$ in the last equation. The Lagrangian function can be decomposed into several intuitive problems, each of which solves the optimization problem for one variable. We provide the decomposed solution in the following.

Queue Update at the Source:

$$\lambda_j(t+1) = \{\lambda_j(t) + \beta_t[x(t) - \sum_{i\in\mathcal{N}} x_{i,j}(t)]\}^+, \ \forall j \in \mathcal{N} \qquad (4)$$

The Lagrange multiplier λ_j can be considered as the queue size for the packets that should be transmitted to node j, because it is updated as the difference between the incoming traffic x and the outgoing traffic $\sum_{i\in\mathcal{N}} x_{i,j}$. The queue λ_j shall be stored at the source. In our practical implementation, we consider that λ_j is stored on video proxies.

Queue Update at the Nodes:

$$\begin{aligned}
&\eta_{i,j}(t+1) = \\
&\{\eta_{i,j}(t) + \beta_t[x_{i,j}(t) - g_{i,j}(t)]\}^+, \ \forall i \in \mathcal{N}, j \in \mathcal{N} \qquad (5)
\end{aligned}$$

The Lagrange multipliers $\eta_{i,j}$ can also be considered as the queue size. $\eta_{i,j}$ is constructed at node i for packets that should be transmitted from node i to node j.

Source Rate Control:

$$x = (U')^{-1}(\sum_{j\in\mathcal{N}} \lambda_j) \qquad (6)$$

where $(U')^{-1}$ is the inverse of the derivative of U. Since U is strictly concave function of x, x is inversely proportional to the sum of the queues for all nodes in a cooperating group according to Eq. (6). This means that the increase in the average queue backlog is an indicator of undelivered packets either over the downlinks or local areas, so the video rate should be reduced.

Downlink Rate Control:

$$\begin{aligned}
&\max_{\mathbf{x}} \ \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} x_{i,j}(\lambda_j - \eta_{i,j}) \\
&\text{s.t. } x_{i,j} \le C_i(1 - p_i), \ \forall i \in \mathcal{N}, j \in \mathcal{N} \qquad (7)
\end{aligned}$$

According to Eq. (7), the transmission rate $x_{i,j}$ is equal to the downlink capacity $C_i(1 - p_i)$ if the difference of queue backlogs (*i.e.,* $\lambda_j - \eta_{i,j}$) is larger than zero. Otherwise, $x_{i,j}$ is set to zero.

Local Area Rate Control and Scheduling: It can be seen from Eq. (1) that the optimal value of $g_{i,j}$ is $g_{i,j} = \sum_{\mathcal{J}|j\in\mathcal{J}} \min_{j\in\mathcal{J}}\{C_{i,j}(1-p_{i,j})\}\tau_{i,\mathcal{J}}$. Therefore, the local area rate control and scheduling problem can be written as;

$$\begin{aligned}
&\max_{\tau} \ \sum_{i\in\mathcal{N}} \sum_{\mathcal{J}\in\mathcal{H}} \tau_{i,\mathcal{J}}(\sum_{j\in\mathcal{J}} \eta_{i,j} \min_{j\in\mathcal{J}}\{C_{i,j}(1 - p_{i,j})\}) \\
&\text{s.t. } \sum_{i\in\mathcal{N}} \sum_{\mathcal{J}\in\mathcal{H}} \tau_{i,\mathcal{J}} \le \gamma \qquad (8)
\end{aligned}$$

Note that Eq. (8) determines the percentage of time that a hyperarc $(i, \mathcal{J})$ is used for transmitting packets, *i.e.,* $\tau_{i,\mathcal{J}}$.

The decomposed solution of P2 in Eq. (2) exactly follows Eq. (6) for the source rate control, Eq. (7) for the downlink rate control, and Eq. (4) and Eq. (5) for the queue updates at the source and local nodes, respectively. The only different part is the local area rate control which is presented in the following. Noting that the optimal value of $g_{i,j}$ is $g_{i,j} = C_{i,j}(1-p_{i,j})\tau_{i,j}$;

$$\begin{aligned}
&\max_{\tau} \ \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \eta_{i,j} C_{i,j}(1 - p_{i,j})\tau_{i,j} \\
&\text{s.t. } \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \tau_{i,j} \le \gamma \qquad (9)
\end{aligned}$$

Similar to Eq. (8), Eq. (9) determines the percentage of time that a link (i, j) is used for transmitting packets, *i.e.,* $\tau_{i,j}$.

We provide numerical calculations for convergence in [6].

V. Implementation Setup and Challenges

We are currently working on implementing this scheme on android phones; this is work in progress and information can be found on [7]. We use ten Google Nexus-S phones. Each of them has a $1GHz$ Cortex A8 processor, $512MB$ of RAM, and several network interfaces (3G, WiFi, and Bluetooth).

The operating system of these phones is Android 2.3, which is open source. In this section, we briefly mention some of our experience and the major challenges we faced, going from theory to practice and we defer details to [6], [7].

Network coding on the phones turned out not to be a problem in practice. We used network coding utilities (NCUtils), a library we have previously developed [38]. In particular, we used the Java library containing a set of functions that can be used to implement network coding; multiplication and inverse is done with table lookup, addition is done with XOR. Encoding is performed according to random network coding, and decoding is done as early as possible. We used a symbol of 1B, generation size of 10, and payload 1000B. Without any optimization, this implementation of network coding on the phone was able to support encoding and decoding rates of 6Mbps. In addition to network coding on the phones, we had to implement a *proxy*, which codes the source packets before transmitting them on the downlink.

Using *multiple interfaces* turned out to be a challenge, because android phones are optimized for space and battery. The available interfaces in the smarthpones are 3G, WiFi, Bluetooth. Bluetooth and 3G can be simultaneously used, and this is our canonical configuration. However, Bluetooth and WiFi are implemented on the same chip and take turns in using the resources, thus using them simultaneously degrades the performance compared to each of them used alone. Furthermore, the android is optimized to hibernate 3G data transmission while WiFi is used, thus preventing us from using them simultaneously.

Broadcast was also a major challenge. In Bluetooth over androids, broadcast is currently not supported. In WiFi, it is possible to implement broadcast, but it is challenging. WiFi broadcast is not the best option because it operates at low base rate, has reliability problems, and performance problems in congested networks. Therefore, we employ pseudo-broadcast, where all connections are unicast but phones overhear transmissions in their neighborhood. Pseudo-broadcast has been used before in the COPE testbed [30], but to the best of our knowledge not on androids. Pseudo-broadcast is more challenging on androids (compared to PCs) for several reasons: the driver for overhearing is not available on all devices and it currently does not work on ad-hoc mode. More importantly, even when it is available, there is no API to pass overheard packets to the application layer. We had to implement a sniffer to provide that API and make use of the overheard packets.

Comparison to non-NC protocols. In addition to the push-based, network coding system which is presented in this paper, we also implemented pull-based mechanisms for network coding, as a baseline for comparison. In the pull-based schemes nodes advertise the video segments they have, and neighbors request some of them. There is a communication overhead associated with those schemes, not captured by the formulations in this paper. However, we found those schemes to often achieve comparable performance to the network coding one. This comparison strongly depends on the configuration parameters and needs further investigation so as to make a fair comparison considering the best version of both schemes.

NUM solution. Our current android implementation does not implement the distributed control indicated by the NUM solution [6]. This is part of future work [7].

VI. Performance Evaluation

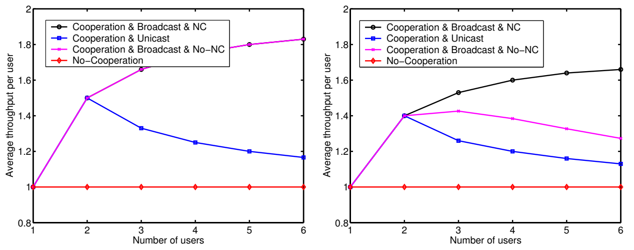
A. Numerical Calculations

We consider the topology shown in Fig. 2 and report from the optimal solution of the NUM problem. The source transmits packets to smartphones via downlinks, and the smartphones construct a cooperating group to improve their video quality. In this section, we consider the achievable throughput as an indicator of video quality improvement, and we present throughput calculations. We compare our schemes; Cooperation & Broadcast & NC, and Cooperation & Unicast with Cooperation & Brodcast & No-NC, and No-Cooperation.

Fig. 3 shows the average throughput vs. the number of users for the following parameters; $C_1 = C_2 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $C_{i,j} = 1$. The average throughput is calculated as the average over all nodes in the system (we use the same definition in the rest of the paper). One can see that the throughput does not change with increasing number of users for No-Cooperation scheme. This is expected, because there is no cooperation in the local area, so the number of users does not affect the throughput. The throughput increases with increasing number of users, then reduces for Cooperation & Unicast. The reason is that cooperation in the local area helps to improve throughput. However, when the number of users increases, the unicast transmissions share the medium, and the throughput of each unicast transmission, hence the overall throughput reduces. Cooperation & Broadcast & NC and Cooperation & Broadcast & No-NC schemes achieve the same throughput levels for Fig. 3(a) (*i.e.,* when $p_{i,j} = 0$). Both schemes improve throughput as the number of users increase. The reason is that since the packets are broadcast, each transmitted packet will be beneficial to more users when the number of users increases. Thus, the average throughput increases. On the other hand, the average throughput of Cooperation & Broadcast & No-NC reduces after a threshold in Fig. 3(b) (*i.e.,* when $p_{i,j} = 0.2$). The reason is that when network coding is not employed, each individual packet should be successfully transmitted to all other nodes in the local area. If it is not successfully delivered, the packet is re-transmitted again. This reduces the average throughput. On the other hand, Cooperation & Broadcast & NC improves throughput with increasing number of users thanks to network coding (which makes all packets equally beneficial).[3]

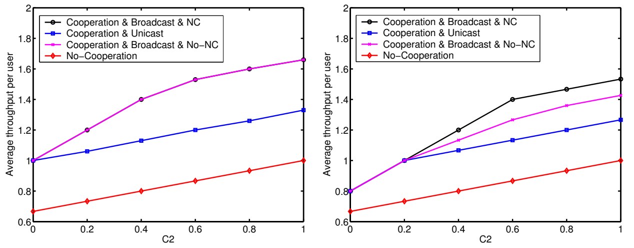
Fig. 4 presents the average throughput versus the second node's downlink capacity; C_2 for the following parameters; $N = 3$, $C_1 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $C_{i,j} = 1$. We see that the average throughput of all schemes improves with increasing C_2. However, the improvement of Cooperation

[3] Note that the number of users in a local area is limited due to geographical and physical constraints as shown in [37]. However, in our system model, we already consider limited number of users, *e.g.,* up to 5-6 users.



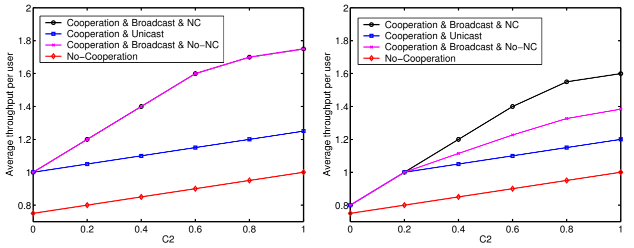
(a) $p_{i,j} = 0$ (b) $p_{i,j} = 0.2$

Fig. 3. Throughput versus number of users. $C_1 = C_2 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $C_{i,j} = 1$.



(a) $p_{i,j} = 0$ (b) $p_{i,j} = 0.2$

Fig. 4. Throughput versus C_2. $N = 3$, $C_1 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $C_{i,j} = 1$.

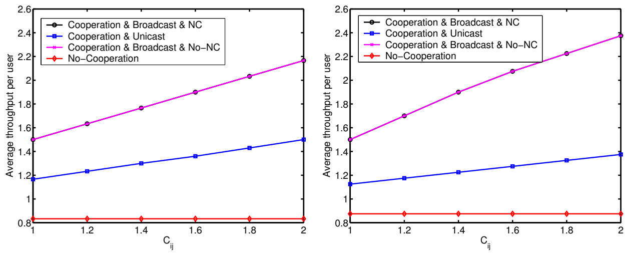


(a) $p_{i,j} = 0$ (b) $p_{i,j} = 0.2$

Fig. 5. Throughput versus C_2. $N = 4$, $C_1 = C_3 = C_4 = 1$, $p_1 = p_2 = p_3 = p_4 = 0$, $C_{i,j} = 1$.

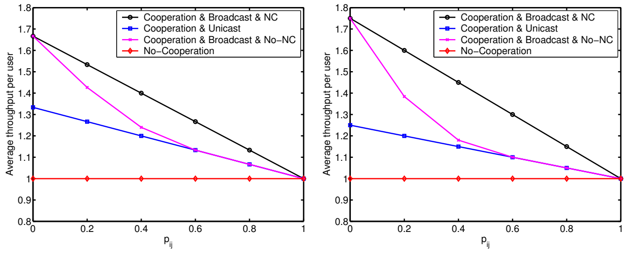
& Broadcast & NC is higher as compared to other schemes especially when there is loss in the local links. We repeat the same calculations for $N = 4$ in Fig. 5. In this case, Cooperation & Broadcast & NC improves more as compared to $N = 3$, because the improvement of Cooperation & Broadcast & NC improves with increasing number of users.

Fig. 6 shows the results for average throughput versus $C_{i,j}$ for the following parameters; (a) $C_2 = 0.5$, $C_1 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $p_{i,j} = 0$. (b) $C_2 = 0.5$, $C_1 = C_3 = C_4 = 1$, $p_1 = p_2 = p_3 = p_4 = 0$, $p_{i,j} = 0$. The average throughput of No-Cooperation scheme does not change with $C_{i,j}$, because it does not employ cooperation ($C_{i,j}$ only affects the cooperation). The average throughput of the other schemes improves with increasing $C_{i,j}$. This is expected, because at higher $C_{i,j}$, more packets can be transmitted among nodes, which improves the overall throughput. Cooperation & Broadcast & NC and Cooperation & Broadcast & No-NC improve more than Cooperation & Unicast, because the schemes using broadcast use the wireless medium more efficiently. Cooperation & Broadcast & NC and Cooperation & Broadcast & No-NC show the same performance, because there is no loss in the local area (*i.e.,* $p_{i,j} = 0$).



(a) $N = 3$ (b) $N = 4$

Fig. 6. Throughput versus $C_{i,j}$. (a) $C_2 = 0.5$, $C_1 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $p_{i,j} = 0$. (b) $C_2 = 0.5$, $C_1 = C_3 = C_4 = 1$, $p_1 = p_2 = p_3 = p_4 = 0$, $p_{i,j} = 0$.



(a) $N = 3$ (b) $N = 4$

Fig. 7. Throughput versus $p_{i,j}$. (a) $C_1 = C_2 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $C_{i,j} = 1$. (b) $C_1 = C_2 = C_3 = C_4 = 1$, $p_1 = p_2 = p_3 = p_4 = 0$, $C_{i,j} = 1$.

Fig. 7 shows the average throughput vs. the local area loss probability (*i.e.,* $p_{i,j}$) for the following parameters; (a) $C_1 = C_2 = C_3 = 1$, $p_1 = p_2 = p_3 = 0$, $C_{i,j} = 1$. (b) $C_1 = C_2 = C_3 = C_4 = 1$, $p_1 = p_2 = p_3 = p_4 = 0$, $C_{i,j} = 1$. Cooperation & Broadcast & NC is especially beneficial for the range of local area loss probabilities (*i.e.,* $p_{i,j}$) from small to high values. The gap between Cooperation & Broadcast & NC and Cooperation & Broadcast & No-NC increases when the number of users increases (as in Fig. 7(b) as compared Fig. 7(a)). Also, we can see that the gap between Cooperation & Broadcast & NC and Cooperation & Unicast increases with increasing number of users. These show the benefit of using network coding and broadcast.

B. Implementation Results

We present preliminary results from our implementation on androids to demonstrate the performance of using *Cooperation & Broadcast & NC* compared to all alternatives.

Experiment 1. Consider two smartphones connected to a video source through 3G, and potentially to each other through 3G. The goal is to download a video file of 1.57MB. We compared the following policies: (I) independent download through 3G alone; (II) 3G and pull-based local, without network coding; (III) 3G and push-based local cooperation with network coding (using a proxy). The download time (averaged over 5 experiments) for policies I,II, and III was 55sec, 40 sec and 32 sec, respectively. This experiment demonstrates the value of using cooperation and network coding.

Experiment 2. Consider three smartphones. One of them, A, has downloaded a 4.92MB mp3 file and wants to share it with the other two phones, B and C. A establishes two UDP unicast connections A-B, and A-C, over WiFi in managed mode. A always sends network coded packets. There is no communication between phones B and C. We compare two policies: (I) no overhearing; (II) B and C can overhear packets transmitted from A to C and B, respectively. The download times (averaged over 5 experiments) are 56sec and 40sec, respectively. Also notice that, because UDP connections are used, there can be lost packets in this scenario. In case (I), phones B and C lose 12.2 and 8.48 segments on average. In case (I), phones B and C lose 2.5 and 0.5 segments on average. This experiment demonstrates the feasibility value of using (pseudo)broadcast with network coding.

These preliminary results serve as a proof-of-concept and motivate the use of *Cooperation & Broadacst & NC*. Videos demonstrating experiments 1 and 2 above can be found on the project website [7].

VII. Conclusion

In this paper, we proposed a cooperative system where a group of smartphone users, within proximity of each other, are all interested in viewing the same video at the same time. Our proposed scheme is grounded on a NUM formulation and its solution, and better utilizes the cellular connections of smartphones as well as their local (either Bluetooth or WiFi) links by exploiting the broadcast nature of the wireless medium and network coding to increase diversity. Numerical calculations demonstrate the effectiveness of *Cooperation & Broadacst & NC* in a range of scenarios. Preliminary results on an android testbed confirm that and serve as proof-of-concept. Additional materials and ongoing work on implementation can be found at [6] and [7], respectively.

References

[1] http://www.3gpp.org/article/lte.

[2] Advanced video coding for generic audiovisual services. ITU-T Recommendation H.264-ISO/IEC 14496-10(AVC), ITU-T and ISO/IEC JTC 1, 2003.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *in IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17(9), Sep. 2007.

[4] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010 - 2015.

[5] http://www.pcworld.com/article/167391/a_day_in_the_life_of_3g.html.

[6] H. Seferoglu, L. Keller, B. Cici, A. Le, A. Markopoulou, "Cooperative Video Streaming on Smartphones," Tech. Report, available at *http://odysseas.calit2.uci.edu/doku.php/public:publication*.

[7] "Wireless Network Coding: From Theory to Practice," project wiki-page, *http://odysseas.calit2.uci.edu/doku.php/public:muri09*.

[8] C. Gkantsidis, P. R. Rodriguez, "Network coding for large scale content distribution," *in Proc. of Infocom*, Miami, FL, March 2005.

[9] M. Wang, B. Li, "R2: random push with random network coding in live peer-to-peer streaming," *in IEEE JSAC*, vol. 25(9), pp. 1655–1666, Dec. 2007.

[10] Z. Liu, C. Wu, B. Li, S. Zhao, "UUSee: large-scale operational on-demand streaming with random network coding," *in Proc. of Infocom*, San Diego, CA, March 2010.

[11] B. Li, D. Niu, "Random network coding in peer-to-peer networks: from theory to practice," *in Proceedings of the IEEE*, vol. 99(3), pp. 513-523, March 2011.

[12] M. Chiang, S. T. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: a mathematical theory of network architectures," *in Proceedings of the IEEE*, vol. 95(1), Jan. 2007.

[13] R. Srikant, "The mathematics of Internet congestion control", Birkhauser, 2003.

[14] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *in IEEE Trans. on Information Theory*, vol. 52(6), June 2006.

[15] L. Chen, T. Ho, S. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," *in Proc. of Infocom*, Anchorage, AK, May 2007.

[16] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multi-hop multicast in wireless mesh networks," *in IEEE JSAC*, vol. 24(11), Nov. 2006.

[17] Z. Li, B. Li, and M. Wang, "Optimization models for streaming in multihop wireless networks," *in Proc. of ICCCN*, Honolulu, HI, Aug. 2007.

[18] M. Chen, M. Ponec, S. Sengupta, J. Li, P. A. Chou, "Utility maximization in peer-to-peer systems," *in Proc. of ACM Sigmetrics*, Annapolis, MD, June 2008.

[19] S. Liu, R. Z.-Shen, W. Jiang, J. Rexford, M. Chiang, "Performance bounds for peer-assisted live streaming," *in Proc. of ACM Sigmetrics*, Annapolis, MD, June 2008.

[20] Z. Shao and S.-Y. R. Li, "To code or not to code: rate optimality of network coding versus routing in peer-to-peer networks," in IEEE Trans. on Communications, vol. 59(4), pp. 948-954, April 2011.

[21] C. Liang, M. Zhao, and Y. Liu, "Optimal bandwidth sharing in multi-swarm multiparty p2p video-conferencing systems," *in IEEE/ACM Trans. on Networking*, March 2011.

[22] M. Ponec, S. Sengupta, M. Chen, J. Li, P. A. Chou, "Multi-rate peer-to-peer video conferencing: a distributed approach using scalable coding," *in Proc. of ICME*, New York, NY, July 2009.

[23] D.-C. Tomozei and L. Massoulie, "Flow Control for Cost-Efficient Peer-to-Peer Streaming," *in Proc. of Infocom*, San Diego, CA, March 2010.

[24] S. Hua, Y. Guo, Y. Liu, H. Liu, and S. S. Panwar, "Scalable video multicast in hybrid 3g/ad-hoc networks," *in IEEE Trans. on Multimedia*, vol. 13(2), pp. 402-413, April 2011.

[25] X. Liu, G. Cheung, C. N. Chuah, "Rate-distortion optimized network coding for cooperative video system repair in wireless peer-to-peer networks," *in Proc. of IEEE MoViD*, June 2008.

[26] S. Li, S. H. G. Chan, "Bopper: Wireless video broadcasting with peer-to-peer error recovery," *in Proc. of IEEE ICME*, August 2007.

[27] J. B. Saleh, D. Qiu, A. K. Elhakeem, "Performance of an efficient scheduling approach to network coding for wireless local repair," *in JSAT*, Jan. 2011.

[28] Y. Park, C. Jo, S. Yun, H. Kim, "Multi-room IPTV delivery through pseudo-broadcast over IEEE 802.11 links," *in Proc. of VTC*, Taipei, Taiwan, May 2010.

[29] P. Vingelmann, M. V. Pedersen, F. H. P. Fitzek, and J. Heide, "On-the-fly packet error recovery in a cooperative cluster of mobile devices," *in Proc. of Globecom*, Houston, TX, Dec. 2011.

[30] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *in IEEE Trans. on Networking*, vol. 16(3), June 2008.

[31] H. Shojania, B. Li, "Random network coding on the iPhone: fact or fiction?," *in Proc. of ACM NOSSDAV*, Williamsburg, VI, June 2009.

[32] H. Shojania, B. Li, "Tenor: making coding practical from servers to smartphones," *in Proc. of ACM Multimedia*, Firenze, Italy, Oct. 2010.

[33] Y. Feng, Z. Liu, B. Li, "GestureFlow: streaming gestures to an audience," in Proc. of Infocom, Shanghai, China, April 2011.

[34] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *in IEEE Trans. on Information Theory*, vol. 34(5), 2000.

[35] P. A. Chou and Y. Wu,"Network coding for the Internet and wireless networks," *in IEEE Signal Proc. Magazine*, vol. 24(5), pp. 77-85, Sept. 2007.

[36] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *in IEEE Trans. on Information Theory*, vol. 52(10), Oct. 2006.

[37] J. Le, J. Lui, and D. M. Chiu, "How many packets can we encode? - an analysis of practical wireless network coding," *in Proc. of Infocom*, Phoenix, AZ, April 2008.

[38] http://arni.epfl.ch/software/