

# Video-Aware Opportunistic Network Coding over Wireless Networks

Hulya Seferoglu, *IEEE Student Member*, Athina Markopoulou, *IEEE Member*

**Abstract**—In this paper, we study video streaming over wireless networks with network coding capabilities. We build upon recent work, which demonstrated that network coding can increase throughput over a broadcast medium, by mixing packets from different flows into a single packet, thus increasing the information content per transmission. Our key insight is that, when the transmitted flows are video streams, network codes should be selected so as to maximize not only the network throughput but also the video quality. We propose video-aware opportunistic network coding schemes that take into account both the decodability of network codes by several receivers and the importance and deadlines of video packets. Simulation results show that our schemes significantly improve both video quality and throughput. This work is a first step towards content-aware network coding.

**Index Terms**—Network coding, video streaming, prioritized transmission, wireless networks, cross-layer optimization.

## I. INTRODUCTION

Providing high quality video over wireless networks is a challenging problem, due to both the erratic and time-varying nature of a wireless channel and the stringent delivery requirements of media traffic. Developments in video compression and streaming, wireless networking, and cross-layer design, are continuously advancing the state-of-the art in wireless video [1], [2]. In this paper, we propose a novel technique for video streaming in a wireless environment inspired by the emerging paradigm of network coding [3]–[6].

Our work builds on recent work in [7], [8] that used network coding to improve throughput in a wireless mesh network. In particular, [7], [8] proposed that wireless routers mix packets from different flows, so as to increase the information content of each -broadcast- transmission and therefore the throughput for data applications. In this paper, we build on this idea, and propose a network coding and scheduling scheme for transmitting several video streams over a wireless mesh network.

Our key insight is that the transmission of video streams in a network coding-capable wireless network should be optimized not only for network throughput but also, and more importantly, for video quality. The fact that video packets have unequal importance is well understood and extensively studied in the video streaming community, e.g. for rate-distortion optimized streaming [9]–[12]. The fact that mixing different information flows can increase throughput in multicast networks is well understood in the network coding community [3], [4], [13], [14]. Our work bridges the gap between the

two approaches, and proposes a new video-aware scheme for network coding and packet scheduling that improves both aspects, namely video quality and throughput. More generally, this idea can be extended from video-aware to content-aware network coding to take into account packet-, flow-, or class-based prioritization of packets in the network code selection.

In this paper, we consider a wireless mesh network, in which routers can mix different incoming flows/streams, using simple network coding operations (XOR). The resulting network code is broadcasted to the neighborhood of the router. Nodes in the same neighborhood listen to each other's transmission and store overheard packets; these are used later to decode received coded packets and also to construct new coded packets. The core question in this architecture is how to select the best -according to an appropriate metric- network code for transmission among all possible codes. In [7], [8], a transmitting node chooses a network code that can be decoded by several neighbors at the same time slot; this policy increases the information per packet transmission thus the throughput. However, when the transmitted flows are video streams, this is not necessarily the best choice. Video quality can be improved by intelligently selecting network codes that combine those video packets that are decodable by several neighbors but also contribute the most to video quality. In other words, when video streams are transmitted, it is not only the quantity but also the quality/content of information transferred that should be taken into account in the selection of network codes. In this paper, we develop schemes for network code selection and packet scheduling that take into account both (i) the importance and deadlines of video packets and (ii) the network state and the received/overheard packets in the neighborhood.

The paper is organized as follows. Section II discusses related work. Section III gives an overview of the system model. Section IV presents the algorithms for network coding. Section V presents simulation results that demonstrate the benefits of the proposed algorithms over baseline schemes, in terms of video quality and application-level throughput. Section VI concludes the paper.

## II. RELATED WORK

This work combines ideas and techniques from two bodies of work: video streaming and network coding.

Several network-adaptive techniques have been proposed to support streaming media over unreliable and/or time-varying networks [15]. Supporting video over wireless is particularly challenging due to the limited, time-varying resources of the wireless channel [1], [2]. There is a large body of work on cross-layer design for video over wireless, such as [16]–[20],

Manuscript received 1 August 2008; revised 20 February 2009.

The authors are with

Digital Object Identifier 10.1109/JSAC.2009.0906xx.

exploiting the fact that packets in a video stream have different importance and therefore should be treated differently by network mechanisms. Packet scheduling is an important control at the medium access control layer. The problem of rate-distortion optimized packet scheduling has been studied in the RaDiO family of techniques [9]–[12]: in every transmission opportunity, media units are selected for transmission so as to maximize the expected quality of received video subject to a constraint in the transmission rate, and taking into account transmission errors, delays and decoding dependencies.

Independently, the network coding paradigm has emerged from the pioneering work in [13], [14], which showed that, in multicast networks where intermediate nodes do simple linear operations on incoming packets, one can achieve the min-cut throughput of the network to each receiver. The linearly combined packets can be utilized at the receivers to recover the original packets by solving a set of linear equations over a finite field. This breakthrough idea inspired significant effort in several directions [3]–[6], including practical application of network coding, studying topologies beyond multicast, such as unicast [21]–[23] and broadcast scenarios. The broadcast nature of the wireless medium offers an opportunity for exploiting the throughput benefits of network coding [24], [25]. The recent work in [7], [8] applied these ideas from the network coding community in the context of wireless mesh networks. [7] implemented a pseudo-broadcast mechanism for 802.11 together with opportunistic listening and a coding layer between IP and MAC that is used to detect coding opportunities and pack packets from different flows into a single transmission, thus increasing network throughput.

Our paper introduces a novel technique for video streaming over wireless that combines the above two approaches. On one hand, we build on [7], [8] to exploit the broadcast nature of the wireless medium and use network coding to pack several packets *from different streams* into a single code for transmission, thus increasing throughput. On the other hand, we construct and select network codes taking into account the importance of video packets (in terms of video distortion and playout deadlines) *within the same stream*, as well as their contribution to the throughput and video quality. This combined approach allows us to improve video quality, e.g. compared to [8], while maintaining the same level of throughput.

An earlier workshop version of this work appeared in [26], where we first introduced the idea of media-aware network coding algorithms (NCV and NCVD). This journal version has a significantly extended evaluation part using a new module we implemented in the GloMoSim environment [28], considering a wide range of traffic scenarios and topologies, comparing against a range of baseline algorithms, and discussing complexity issues. In addition, this paper generalizes and formalizes the original idea by introducing NC-RaDiO, an extension of the RaDiO framework to take into account the network coding opportunities, and by showing how our video-aware network coding algorithms fit within this framework and that they achieve near optimal performance.

Looking forward, this idea can be generalized from video-aware to content-aware network coding. In the video-specific application, the importance of the packets reflects the video distortion as determined by content, encoding and playout

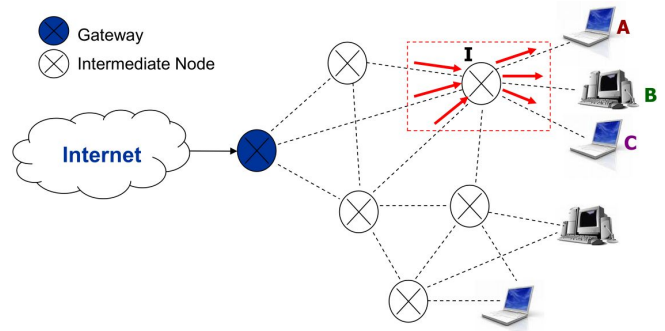


Fig. 1. A wireless mesh network.  $I$  is an intermediate node,  $A, B, C$  are receiving (and/or sending) nodes.

deadlines. More generally, the importance/priority of packets can be determined by price, policy or other considerations and can express differentiation at various levels of granularity: across packets of the same flow, across different flows/users, or across different classes of users.

### III. SYSTEM OVERVIEW

We consider video streaming over wireless mesh networks where intermediate nodes (wireless mesh routers) are able to forward packets to other intermediate nodes and/or clients, as shown in Fig. 1. In this paper, we propose algorithms that can be used at the intermediate node to maximize video quality and throughput. We assume that intermediate nodes can perform simple network coding operations (bit-wise XOR) and combine packets from several incoming streams into a single outgoing packet. This packet is broadcasted to the entire neighborhood, thus reaching several nodes at the same time. We assume that nodes can overhear all transmissions in their neighborhood, whether they are intended for them or not; they can decode a network-coded packet using overheard packets. The idea of combining network coding with broadcast to increase the information content per transmission, is well understood in the network coding community. This idea has been recently applied in 802.11-based multi-hop wireless networks and throughput benefits have been demonstrated for data applications [7], [8].

Our key observation is that, when the transmitted flows are video streams, this is not necessarily the best choice and video quality must also be considered. The importance and deadlines of video packets must be taken into account to select those codes that contribute the most to the quality of video streams. In this paper, we develop schemes for network coding across different flows, and packet selection within each flow, to improve both video quality and throughput.

1) *Code Selection at an Intermediate Node*: Let us consider an intermediate node that receives  $N$  packets from different video streams and forwards them to  $N$  nodes in its neighborhood. The intermediate node maintains a transmission (Tx) queue with incoming video packets. At a given time slot a packet is selected from the Tx queue for transmission. The selected packet is called the primary packet and its destination node is called the target node. The primary packet can be thought as the main packet we try to transmit during a time slot. Depending on the network coding scheme, the primary

TABLE I  
TERMINOLOGY

| Term            | Definition   |
|-----------------|--|
| Primary Packet  | The packet selected from the Tx queue before network coding. It must be included in all network codes. It can be thought as the main packet we try to transmit in a given time-slot. |
| Side Packet     | Packet in the Tx queue, other than the primary, included in the network code.  |
| Network Code    | One primary and all side packets XOR-ed together into a single packet.   |
| Active Packet   | Packet in the Tx queue that can be considered as primary. (Not transmitted within the last RTT.)   |
| Inactive packet | Packet in the Tx queue that cannot be considered as primary. (It has already been transmitted within the last RTT, and the acknowledgement is still pending.)                        |
| Target Node     | The intended recipient of the primary packet.  |
| Tx Queue        | The output queue of the transmitting node.   |
| Rx Buffer       | The receive queue of the receiving node. It stores received packets, destined to this node.  |
| Virtual Buffer  | Also maintained at a receiving node. It stores overheard packets, destined to other nodes.   |

packet may be the first packet from the head of the queue, or any packet in Tx queue that is marked as active (*i.e.*, not transmitted within the last round-trip time, as discussed later). In addition to the primary packet, all packets in the queue are considered as candidate side packets, *i.e.*, candidates for a transmission in the same time slot together with the primary packet; they are useful to nodes other than the target node. The primary and the side packets are all XOR-ed together into a single packet, called the network code. The core question then is: *which network code (i.e. XOR of the primary and side packets) to select and transmit so as to maximize the total video quality and throughput*. The algorithms addressing this question are the main part of this paper, and will be discussed separately in the next section (IV). In the rest of this section, we describe the remaining components and functions of the system. The terminology is summarized in Table I.

2) *Receiving, Overhearing and ACKing a Packet (at Receiving Nodes)*: Once the network code is chosen, it is broadcast to all nodes in the neighborhood. Depending on the channel conditions, some nodes successfully receive it. When the target node receives it, it decodes it (which is guaranteed by the construction of the network code in the next section), stores the primary packet in its receive (Rx) buffer, and sends an acknowledgement (ACK) back to the intermediate node. Nodes, other than the target node, overhear the transmitted packet and try to decode it; if they overhear a new packet destined to them, they store it in their Rx buffer and send an ACK back to the intermediate node; if they obtain a packet destined for another node, they store it in their virtual buffer. An overheard packet stays in the virtual buffer until an ACK from the target is overheard or until its deadline expires. We also assume that the asynchronous ACK mechanism, proposed in [7], is used to combat ACK implosion.

3) *Active/Inactive Packets (at an Intermediate Node)*: The intermediate node waits for a mean round-trip time (RTT) from the time it transmits the network code until it receives an ACK. During that period, all packets that were part of the code stay in the Tx queue but are marked as inactive. Inactive packets are not considered for primary transmission (in order to avoid unnecessary duplicate transmissions) but

are still considered as candidates for side packets (to increase coding opportunities). When the transmitter receives an ACK, it removes the corresponding packet from the Tx queue. If an RTT expires without receiving an ACK, the packet is marked as active again and the process is repeated. A packet stays in the Tx queue until either it is successfully transmitted or its deadline expires; when either of these occur, the packet is removed from the transmission buffer.<sup>1</sup>

4) *Requirements*: Our system relies on the following capabilities. First, broadcast is needed to harvest the benefits of network coding. Although wireless is inherently a broadcast medium, this may be hidden by some communication protocols. We make use of the broadcast capability, implemented as pseudo-broadcast on top of 802.11 unicast in [7], [8]. Second, nodes need to learn the contents of the virtual buffers of all their neighbors, in order to select a code that is decodable in their neighborhood. This can be achieved by explicitly exchanging and/or implicitly learning this information as in [7], [8]. Third, nodes must be capable of coding/decoding in real time, which is a realistic assumption for simple (bit-wise XOR) operations. Network coding is implemented as a thin layer between IP and MAC, exactly as in the original COPE [8]. Nodes are considered fixed (not mobile) and routing is considered given, *i.e.*, decided by a routing module orthogonal to the network coding algorithms considered in this paper.

5) *Importance of Packets*: Nodes make network coding decisions taking into account the importance of video packets. The distortion value ( $\Delta$ ) of every packet can be determined by the source and communicated to the intermediate nodes in order to enable them to take decisions about transmission of video units in a rate-distortion optimized manner [9]. This information can be marked on a special field of the packet header. This field can be at the application level (e.g. RTP extended headers) or part of the network coding header; alternatively, the typically unused TOS/DiffServ byte in the IP header can be overridden. In addition to the individual importance of packets ( $\Delta$ ) within a flow, our formulation also considers the importance of flows ( $\gamma$ ). In general, the overall importance of a packet can be a function of the flow priority and the packet distortion value; in this paper, we use a simple product  $\gamma \cdot \Delta$ .

6) *Handling Video vs. Data Packets*: The main focus of this paper is on network coding for video, and most of the discussion is presented in terms of queues that contain only video packets. This could be implemented in practice on top of 802.11e, using the differentiation mechanisms to separate real-time traffic (in our case video) from data traffic. Our network coding algorithms could then be applied only on the video queue. Independently, our framework could also handle a mixture of video and data packets in the same queue by assigning them different flow priorities.

<sup>1</sup>Note that although a transmitted packet remains inactive for an RTT, it does not block the head of the queue: the next active packets in the queue are coded and transmitted during this period. Also note that, although the Tx queue is basically a FIFO, considering any active packet as primary may lead to reordering in packet delivery. Although this may be a concern for TCP, as it was the case in [7], it is clearly better for video that requires timely delivery and can reorder packets at the playout buffer.



#### IV. CODING ALGORITHMS

The main questions in this system have to do with the construction and selection of network codes. The *code construction* problem is concerned with finding candidate codes that guarantee decodability by the target node. The *code selection* problem is concerned with selecting the best among the candidate codes so as to optimize video quality. The first proposed algorithm, “NCV”, achieves the same throughput gains as in [7] but also intelligently chooses the network codes that maximize video quality. The second algorithm, “NCVD”, uses NCV as a building block but considers more coding options thus further improving video quality and throughput. The third algorithm, “NC-RaDiO”, generalizes these ideas: it extends the rate-distortion optimized (RaDiO) packet scheduling framework [9], so as to find the optimal network coding and transmission policy at every transmission opportunity.

##### A. NCV Algorithm: Network Coding for Video

Assume that there are several video streams coming to an intermediate node. Depending on the content of the virtual buffers at the clients, there may be several combinations of these streams, *i.e.* several network coding opportunities. The main idea behind the Network Coding for Video (NCV) algorithm is to select the best network code to improve video quality. The following example demonstrates this idea.

*Example 1:* Consider the example shown in Fig. 1 and let us focus on a single-hop shown in more detail in Fig. 2. Node  $I$  receives three independent video streams, e.g. from the Internet through the gateway, destined to its neighbors  $A, B, C$ .  $I$  maintains a FIFO Tx queue that stores packets  $\{A_1, A_2, \dots\}$  destined to node  $A$ ,  $\{B_1, B_2, \dots\}$  destined to node  $B$ , and  $\{C_1, C_2, \dots\}$  destined to node  $C$ . Fig. 2 also shows the contents of the virtual buffers at each client: node  $A$  has overheard packets  $\{B_1, C_1\}$  and nodes  $B$  and  $C$  have both overheard packet  $A_1$ , from previous transmissions.  $A_1$  is the first active packet from head of the queue and is selected as the primary packet. Any packet (active or inactive) in the output queue, other than  $A_1$ , can be chosen as a side packet, on the condition that the constructed network code should be decodable at node  $A$ , *i.e.*  $A_1$  can be retrieved. To satisfy this condition, side packets that will be used in the network code should already be available at node  $A$ ; in other words, the decodability of a network code depends on the overheard packets at node  $A$ . Network codes  $c_1 = A_1$ ,  $c_2 = A_1 \oplus B_1$ ,  $c_3 = A_1 \oplus C_1$ , and  $c_4 = A_1 \oplus B_1 \oplus C_1$  can all be decoded by  $A$  and thus are eligible network codes.  $\square$

*The Code Construction Problem.* More generally, consider that there are  $N$  nodes  $\mathbf{N} = \{n_1, n_2, \dots, n_N\}$  in the wireless network. Consider an intermediate node  $n \in \mathbf{N}$ , which transmits to its neighbor nodes. Let  $\phi_n$  be the number of packets in the Tx queue of node  $n$ , and the packets themselves be  $\Phi_n = \{p_1, p_2, \dots, p_{\phi_n}\}$ . Choose the first active packet,  $p_i$ , from the head of the Tx FIFO queue as the primary packet;  $p_i$  has a target node  $t(p_i) \in \mathbf{N}$ . Node  $n$  will construct and broadcast a network code, which consists of the primary packet  $p_i$  XOR-ed together with some side packets, so that the target node  $t(p_i)$  can decode and obtain  $p_i$ . For this to be guaranteed, all side packets must be among the packets that are already overheard

---

#### Algorithm 1 The NCV Algorithm

---

- 1: Initialization:  $I_{max}^i = 0$ ,  $c_{max}^i = \emptyset$
  - 2: Choose the first head-of-queue active packet as primary  $p_i$ .
  - 3: Let  $t(p_i)$  be the target node of packet  $p_i$ . Let  $\{\nu_1, \dots, \nu_{\Psi_{t(p_i)}}\}$  be the overheard packets at  $t(p_i)$ .
  - 4: **for**  $k = 1 \dots 2^{\psi_{t(p_i)}}$  **do**
  - 5:    $c_k^i = \{p_i\} \cup S_k^{t(p_i)}$
  - 6:   Calculate  $I_k^i$  with Eq. (5)
  - 7:   **if**  $I_k^i > I_{max}^i$  **then**
  - 8:      $I_{max}^i = I_k^i$ ,  $c_{max}^i = c_k^i$
  - 9:   **end if**
  - 10: **end for**
  - 11: Choose  $c_{max}^i$  as the network code. XOR all packets and transmit
- 

at the target  $t(p_i)$ . Assume that  $\psi_{t(p_i)}$  packets are overheard at node  $t(p_i)$  and denoted by  $\Psi_{t(p_i)} = \{\nu_1, \nu_2, \dots, \nu_{\Psi_{t(p_i)}}\}$ . Therefore, the candidate network codes at node  $n$  are:

$$c_k^i = \{p_i\} \cup S_k^{t(p_i)}, k = 1, 2, \dots, 2^{\psi_{t(p_i)}} \quad (1)$$

where  $S_k^{t(p_i)}$  is the  $k^{th}$  subset of  $\Psi_{t(p_i)}$ . Note that, since linear operations are limited to bit-wise XOR, a network code  $p_1 \oplus p_2 \oplus \dots \oplus p_k$  is completely specified by the set of packets  $\{p_1, p_2, \dots, p_k\}$  that are XOR-ed together. The next step is to select the best among all candidate codes.

*Example 1 Continued.* Node  $A$  can get packet  $A_1$  from all four possible network codes. Codes  $c_2$  and  $c_3$  improve the video quality at node sets  $\{A, B\}$  and  $\{A, C\}$ , respectively. It is clear that  $c_2$  and  $c_3$  are better codes than  $c_1$  and  $c_4$  both for throughput (they are useful to two instead of one node) and video quality. Comparing  $c_2$  to  $c_3$ , we observe that they are equivalent in terms of throughput but they may contribute differently to video quality depending on the content of video packets  $A_1, B_1, C_1$ . Deciding which candidate code to select between  $c_2 = A_1 \oplus B_1$  and  $c_3 = A_1 \oplus C_1$  should depend on the importance and urgency of the original video packets  $B_1$  and  $C_1$ . NCV exploits this observation.  $\square$

*The Code Selection Problem.* After constructing all candidate codes at a node  $n$ , we need to choose the best code according to an appropriate metric, which we define here so as to capture the contribution of each candidate code to the video quality improvement. Recall that  $p_i$  is the primary packet targeted to node  $t(p_i)$ , and  $\{c_k^i\}_{k=1}^{2^{\psi_{t(p_i)}}}$  are all the candidate codes. Let  $I_k^i(n_\eta)$  be the improvement in video quality at node  $n_\eta$  for  $\eta = 1, 2, \dots, N$ , when  $c_k^i$  is received and decoded:

$$I_k^i(n_\eta) = \sum_{l=1}^{L_k} (1 - P(l)) \Delta(l) \gamma(l) g_l^k(n_\eta) d_l^k(n_\eta) \quad (2)$$

where each factor in this formula is defined as follows:

- $L_k$  is the number of original packets included in network code  $c_k^i$ . Notice that at most one out of these  $L_k$  packets can be useful to a particular node  $n_\eta$ , but different packets are useful to different nodes.
- $d_l^k(n_\eta)$  and  $g_l^k(n_\eta)$  are indicator functions that express whether code  $k$  is useful for node  $n_\eta$ . We define  $d_l^k(n_\eta) = 1$  if  $c_k^i$  is decodable at node  $n_\eta$ , or 0 otherwise. We define  $g_l^k(n_\eta) = 1$  if packet  $l$  is targeted to node  $n_\eta$ , or 0 otherwise.
- $\Delta(l)$  is the improvement in video quality (SNR) if packet  $l$  is received correctly and on time at client  $n_\eta$ . To

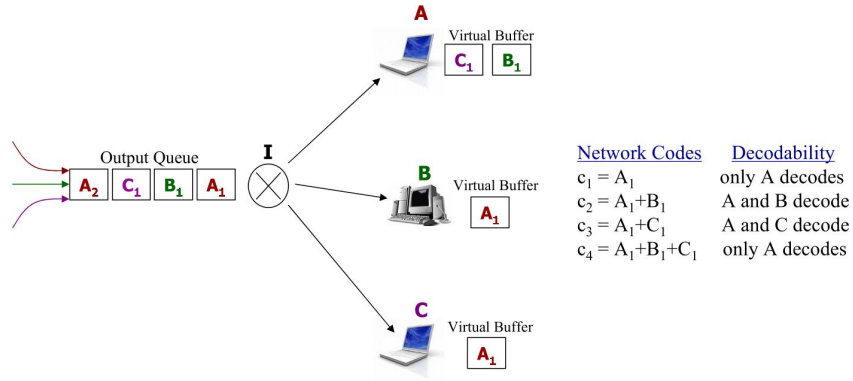


Fig. 2. Example of Network Coding for Video (NCV), for a one-hop downlink scenario with three different streams.

compute  $\Delta(l)$ , we decode the entire video sequence with this packet missing and we compute the resulting distortion.<sup>2</sup> We assume that this computation is performed at the source offline and that the distortion value is marked on each packet.<sup>3</sup>

- $\gamma(l)$  is the importance/priority of the flow that packet  $l$  belongs to. All packets in the same flow have the same (flow) importance, but different flows may have different importance. If some flows are more important, then higher importance should be assigned to them; otherwise they should all be assigned  $\gamma = 1$ . If the average quality differs among encoded video sequences and we still want to treat flows equally, the flow importance can be a normalization factor (the inverse of the average PSNR per sequence); this is what we do in the simulations.
- $P(l)$  is the probability that packet  $l$  is lost due to either channel errors or late arrival for playout:

$$P(l) = P\{FTT' > t_d(l) - t_c\} \quad (3)$$

where  $t_d(l)$  is the deadline of packet  $l$ ,  $t_c$  is the current time and  $\tau = t_d(l) - t_c$  is the remaining time until the playout deadline;  $FTT'$  is the forward trip time in the presence of delay and loss. The complementary cumulative distribution function of  $FTT'$  can be calculated as follows:

$$P\{FTT' > \tau\} = \varepsilon_F + (1 - \varepsilon_F) \int_{\tau}^{\infty} p_F(t) dt \quad (4)$$

The first part in Eq.(4) describes the probability that a packet is lost in the forward channel, due to noise, fading, and interference in the wireless. The second part in Eq.(4) describes the probability that a packet, which is not lost,

<sup>2</sup>This is an approximation as the actual distortion that may also depend on the delivery status of prior and subsequent NALs. The distortion model can be extended to capture these loss correlations [29]–[31]. Furthermore, we assume that distortions caused by loss of multiple packets are additive, which is reasonable for sparse losses. These approximations reduce the computational complexity by separating the total distortion function into a set of individual packet distortion functions and optimizing for each one of them.

<sup>3</sup>For real-time traffic, one can still estimate the distortion by performing online analysis with a delay of a few frames. Most distortion occurs in the first few frames after a loss and breaks after the next I frame; the error depends on the video content of subsequent frames and on the coding decisions. Another approach is to assign distortion values based solely on the GOP structure, ignoring the video content and coding decisions, or to use a model for dependencies [31].

arrives late, *i.e.*, after its playout deadline;  $p_F(t)$  is the distribution of the forward-trip time.

After defining the contribution of code  $c_k^i$  to the video quality at a single node  $n_\eta$ ,  $I_k^i(n_\eta)$ , we define the total video quality improvement of code  $c_k^i$  as the sum of the video quality improvements at all clients  $\eta = 1, \dots, N$ , due to code  $c_k^i$ :

$$I_k^i = \sum_{\eta=1}^N I_k^i(n_\eta) \quad (5)$$

The NCV algorithm is summarized in Alg. (1). At each time slot, the NCV algorithm chooses the primary packet  $p_i$  and constructs all candidate network codes  $\{c_k^i\}_{k=1}^{2^{\psi_t(p_i)}}$ . Among all candidate network codes, NCV chooses the code that maximizes the total video quality improvement:

$$\max_k I_k^i \quad (6)$$

Depending on the contents of the virtual buffers, it is possible that no side packets can be used together with a given primary packet  $p_i$ . In that case, the network code is simply  $\{p_i\} \cup \emptyset = \{p_i\}$ .

#### B. NCVD Algorithm: looking into the queue in Depth

As described in the previous section, NCV selects the primary packet from the head of the queue but ignoring packets marked as inactive, and then optimally chooses the side packets. However, the fact that NCV does not optimize the primary packet has two implications: (i) the primary packet itself is important for video quality and (ii) the candidate side codes are limited to those that are decodable for this single primary packet. The second algorithm improves over NCV by also optimizing the selection of the primary packet. NCVD looks into the entire Tx queue “in depth” and considers all, not just the head-of-line, packet as candidates for the primary packet, thus increasing the options for candidate codes. A different set of candidate codes can be constructed for each primary packet. We explain NCVD through the following example.

*Example 2:* Let us look at Fig. 3. The topology is the same as in Fig. 2, but the contents of the Tx queue and of the virtual buffers are different. Assume that all packets are active packets, *i.e.*, they can all be considered as primary. One option is to select the head-of-line packet  $A_1$  as the primary packet.

**Algorithm 2** The NCVD Algorithm

---

```

1: Initialization:  $c_{max} = \emptyset, I_{max} = 0$ 
2: for every packet  $i = 1, \dots, \phi_n$  from the head of Tx queue do
3:   Consider this packet,  $p_i$ , as candidate for primary
4:   Construct all possible codes  $c_k^i$  for  $p_i$ 
5:   Determine the max improvement  $I_{max}^i = \max_k I_k^i$ 
6:   and the corresponding code  $c_k^i: k = \text{argmax}_k I_k^i$  as in NCV
7:   if  $I_{max}^i > I_{max}$  then
8:      $I_{max} = I_{max}^i, c_{max} = c_k^i$ 
9:   end if
10: end for
11: Choose  $c_{max}$  as the network code. XOR all packets and transmit.

```

---

As discussed in Example 1, the best codes for this primary packet are  $c_3 = A_1 \oplus C_1$  or  $c_4 = A_1 \oplus B_1 \oplus C_1$ . A different choice is to select  $B_1$  as the primary packet, which leads to a different set of candidate network codes (listed on the Fig. 3). Code  $c'_4 = B_1 \oplus C_1 \oplus A_2$  achieves the maximum throughput improvement, and potentially the maximum video quality, depending on the importance and urgency of all packets. This example demonstrates that increasing our options of primary packet, increases the set of candidate codes, and thus can improve both throughput and video quality.  $\square$

More generally, NCVD constructs candidate codes  $c_k^i, k = 1, 2, \dots, 2^{\psi_t(p_i)}$  for each candidate primary packet  $p_i$  in the Tx queue. Among all constructed codes, NCVD selects the code that maximizes the total improvement in video quality for all clients:

$$\max_{p_i} \max_k (I_k^i) \quad (7)$$

Algorithm 2 summarizes NCVD.

NCVD can be parameterized by the depth  $d$  of the Tx queue considered in the selection of the primary packet. NCVD( $d = 1$ ) is simply NCV, while NCVD( $d = \infty$ ) considers all packets in the Tx queue. The larger the value of  $d$ , the more coding options, the better the performance of NCVD. Because queue sizes are small for real time applications, we can focus on NCVD( $d = \infty$ ), simply referred to as NCVD.

### C. NC-RaDiO: Rate-Distortion Optimized Network Coding

The NCV and NCVD algorithms, described above, choose the network code for the next transmission opportunity, so as to maximize the video quality. In this section, we formulate this problem within the rate-distortion optimized (RaDiO) packet scheduling framework [9]. Starting from the RaDiO formulation [9], especially for multiple streams sharing the same medium [11], we modify and extend it to account for network coding, instead of just packet, transmission policies (NC-RaDiO). We show how to find the optimal solution and that our previous algorithms (especially NCVD) are efficient heuristic solutions to the general NC-RaDiO optimization problem under some mild assumptions.<sup>4</sup>

1) *Formulation*: Let us consider a single node  $n \in \mathbf{N}$  in the wireless mesh network, with packets  $\Phi_n = \{p_1, p_2, \dots, p_{\phi_n}\}$  in its queue, and let us focus on a single transmission opportunity. Without network coding, in order to do classic RaDiO packet

scheduling, the node would choose a policy  $\pi$  for the next transmission opportunity. The policy would indicate for every packet in the queue,  $p_j \in \Phi_n$ , whether this packet is transmitted  $\pi(j) = 1$  or not  $\pi(j) = 0$ , so as to minimize a weighted function of distortion and rate  $J(\pi) = D(\pi) + \lambda R(\pi)$ .

With network coding, the node  $n \in \mathbf{N}$  chooses some network codes, consisting of packets in the queue XOR-ed together, to transmit. All possible network codes at node  $n$  are  $C_n = \{c_k^i\}_{k=1, \dots, 2^{\psi_t(p_i)}; i=1, \dots, \phi_n}$ . The code transmission policy at node  $n$  consists of a vector  $\Pi_n$  that indicates for every possible code  $c_u \in C_n, u = 1, \dots, |C_n|$ , whether it is transmitted  $\Pi_n(c_u) = 1$  or not  $\Pi_n(c_u) = 0$ , in the next transmission opportunity. To avoid transmitting two network codes  $c_u, c_v \in C_n$  that have common packets from the set  $\Phi_n$ , we restrict our attention to “valid” network code policies  $\Pi_n^{valid}$  that do not allow that to happen; *i.e.*,  $\Pi_n^{valid} \subset \Pi_n$  s.t.  $\Pi_n^{valid}(c_u) = 1 \wedge \Pi_n^{valid}(c_v) = 1$  if and only if  $c_u \cap c_v = \emptyset$ . Our goal is to find the optimal code transmission policy on all nodes  $\Pi^{valid} = \{\Pi_n^{valid}\}_{\forall n \in \mathbf{N}}$ , so as to minimize the total distortion  $D(\Pi^{valid})$ , subject to the rate constraint  $R(\Pi^{valid}) \leq R_{av}$  where  $R_{av}$  is the available bit rate. With Lagrangian relaxation, our problem turns to finding the code transmission policy  $\Pi^{valid}$  so as to minimize  $J(\Pi^{valid}) = D(\Pi^{valid}) + \lambda R(\Pi^{valid})$ .

Instead of finding the optimal code transmission policy, we can map each code to the packets it contains (*i.e.*, XOR-ed together), and find the optimal packet transmission policy. The reason for converting the problem from a code to a packet transmission policy selection is that it is more natural to express distortion values per packet. Let  $\pi$  be the packet transmission policy on all nodes,  $\pi = \{\pi_n(j)\}_{\forall n \in \mathbf{N}, \forall p_j \in \Phi_n}$ .  $\pi$  depends on the code transmission policy  $\Pi^{valid}$  as follows:

$$\pi_n(j) = \begin{cases} 1 & \text{if } \exists c_u \in C_n \text{ s.t. } p_j \in c_u \text{ and } \Pi_n^{valid}(c_u) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

An equivalent problem is to choose a packet policy  $\pi$ , s.t.:

$$\min_{\pi, \lambda} J(\pi) = \min_{\pi, \lambda} \{D(\pi) + \lambda R(\pi)\} \quad (9)$$

In Eq.(9),  $D(\pi)$  is the total distortion over all nodes under policy  $\pi$ :  $D(\pi) = \sum_{n=1}^N D(\pi_n)$ .  $D(\pi_n)$  is the approximate distortion of the flows transmitted from node  $n$  under the policy  $\pi_n$ . Following a similar definition as in [11],  $D(\pi_n) = \sum_{p_j \in \Phi_n} \gamma(j) \Delta(j) P(\pi_n(j))$ , where:  $\gamma(j)$  is the priority/importance of the flow to which packet  $p_j$  belongs;  $\Delta(j)$  is video quality distortion when packet  $p_j$  is lost as defined in section IV-A; and  $P(\pi_n(j))$  is the probability that packet  $p_j$  is lost under policy  $\pi_n(j)$ . In particular,  $P(\pi_n(j)) = P_p(j) P_c(\pi_n(j))$  consists of two parts: the probability  $P_p(j)$  that the packet is lost in previous transmissions; and the probability  $P_c(\pi_n(j))$  that the packet is lost in its current transmission under policy  $\pi_n(j)$ . Let also  $t_d(j)$  be the deadline of packet  $p_j$ ,  $t_c$  the current time, and  $t_m$  the time of  $m^{\text{th}}$  transmission assuming  $M$  transmissions so far.  $FTT'$  and  $RTT'$  are the random variables corresponding to the forward

<sup>4</sup>We note that our NC-RaDiO formulation assumes that the distortion of a flow is approximated by the sum of the distortion incurred at each hop along its path. This allows for the centralized RaDiO framework to be solved in a distributed way, *i.e.* to make decision at each node, and examined in [27].



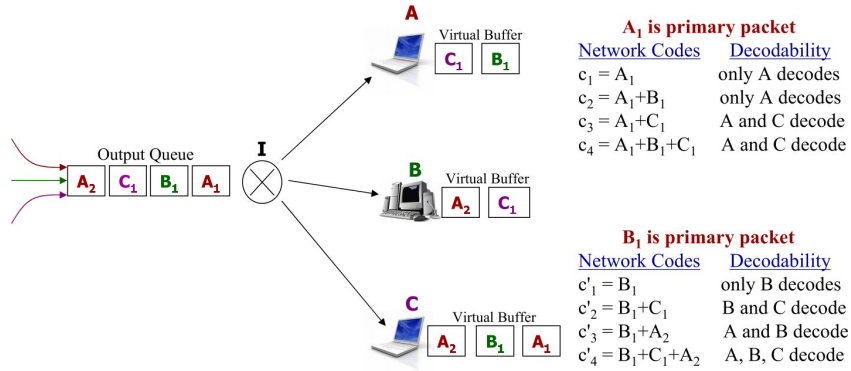


Fig. 3. Example of NCVD in the scenario of one-hop downlink transmission of three different receivers.

and round trip times, respectively<sup>5</sup>. Then the loss probability can be further expressed as follows:

$$P_p(j) = \prod_{m=1}^M P\{FTT' > t_d(j) - t_m | RTT' > t_c - t_m\} \quad (10)$$

$$P_c(\pi_n(j)) = \begin{cases} P\{FTT' > t_d(j) - t_c\} & \text{if } \pi_n(j) = 1, \\ 1 & \text{if } \pi_n(j) = 0 \end{cases} \quad (11)$$

In Eq. (9),  $R(\pi)$  is the total rate function over all node rates under policy  $\pi$ :  $R(\pi) = \sum_{n=1}^N R(\pi_n)$ .  $R(\pi_n)$  is the rate of the flows transmitted from node  $n$  under policy  $\pi_n$ :  $R(\pi_n) = \sum_{\forall c_u \in C_n} \max_{p_j \in c_u} \{B(j)\rho(\pi_n(j))\}$ , where  $B(j)$  is the size of packet  $p_j$  in bytes, and  $\rho(\pi_n(j))$  is the average cost of transmitting packet  $p_j$ . Note that the maximization  $\max_{p_j \in c_u} \{B(j)\rho(\pi_n(j))\}$  term comes from network coding: before getting XOR-ed together, packets may need to be padded up to the length of the longest packet.

Note that  $\sum_{\forall c_u \in C_n} \sum_{p_j \in c_u}$  and  $\sum_{p_j \in \Phi_n}$  are equivalent. Also  $\rho(\pi_n(j)) = \Pi_n^{valid}(c_u) = 0$  or  $1$  (s.t.  $p_j \in c_u$ ) depending on whether code  $c_u$  is transmitted or not. With these observations and by replacing the distortion  $D(\pi)$  and rate  $R(\pi)$  terms with their detailed expressions discussed above, the NC-RaDiO problem of Eq.(9) can be re-written as follows:

$$\min_{\pi, \lambda} \sum_{n=1}^N \sum_{\forall c_u \in C_n} \left( \sum_{p_j \in c_u} \gamma(j)\Delta(j)P(\pi_n(j)) + \lambda \Pi_n^{valid}(c_u) \max_{p_j \in c_u} \{B(j)\} \right) \quad (12)$$

2) *Optimal Solution*: Since current systems typically transmit one packet (network code in our case) at each transmission opportunity, we will focus on this case from now on; *i.e.*, we will find the optimal network code (instead of finding several network codes) so as to minimize the above rate-distortion function. This was also the case in COPE [7], [8] as well as in our NCV and NCVD algorithms. An approach, introduced in [9], was to increase the Lagrange multiplier  $\lambda$  so that exactly one network code is selected as the optimal code for the total rate-distortion function. However, this approach requires centralized knowledge.

<sup>5</sup> $FTT'$  is distributed as in Eq. (4) and  $RTT'$  has CCDF  $P\{RTT' > \tau\} = \varepsilon_R + (1 - \varepsilon_R) \int_{\tau}^{\infty} p_R(t)dt$ , where  $p_R(t)$  is the distribution of round trip time, and  $\varepsilon_R$  is the loss probability, considering the forward and backward channels together.

A distributed approach is to solve the problem in Eq. (12) for every network code  $c_u \in C_n, n = 1, \dots, N$ , find a threshold value,  $\lambda_n(c_u)$ , used to make the decision whether to transmit  $c_u \in C_n$  and select the network code that maximizes that threshold  $\max_{n, c_u} \{\lambda_n(c_u)\}$ . In particular, let us define the per network code cost function as:

$$J_n(c_u) = \sum_{p_j \in c_u} \gamma(j)\Delta(j)P_p(j)P_c(\pi_n(j)) + \lambda \Pi_n^{valid}(c_u) \max_{p_j \in c_u} \{B(j)\} \quad (13)$$

for every code  $c_u \in C_n$  and node  $n = 1, \dots, N$ . When we decide to not transmit this network code, *i.e.*,  $\Pi_n^{valid}(c_u) = 0$ , the cost becomes  $J_n^0(c_u) = \sum_{p_j \in c_u} \gamma(j)\Delta(j)P_p(j)$ . When we decide to transmit the network code, *i.e.*,  $\Pi_n^{valid}(c_u) = 1$ , the cost is  $J_n^1(c_u) = \sum_{p_j \in c_u} \gamma(j)\Delta(j)P_p(j)P_c(\pi_n(j)) + \lambda \max_{p_j \in c_u} \{B(j)\}$ . Depending on which of the two costs is smaller, we decide whether to transmit  $c_u$  or not. The maximum  $\lambda$  that satisfies the inequality  $J_n^1(c_u) \leq J_n^0(c_u)$  is

$$\lambda_n(c_u) = \frac{\sum_{p_j \in c_u} \gamma(j)\Delta(j)P_p(j)(1 - P_c(\pi_n(j)))}{\max_{p_j \in c_u} \{B(j)\}} \quad (14)$$

The optimal policy decides, in a rate-distortion optimized manner, which node  $n$  should transmit and what code  $c_u$  should be transmitted, by choosing the maximum Lagrange multiplier:  $\max_{\{n, c_u\}} \{\lambda_n(c_u)\}$ . This can be achieved in practice in two rounds: first, we compare  $\lambda_n(c_u)$  in the same node  $n$  and we can find  $\lambda_n = \max_{\{c_u\}} \{\lambda_n(c_u)\}$  for this node; then all nodes  $n \in \mathbf{N}$  need to exchange their  $\lambda_n$  values with all the neighbors; finally, the node with  $\lambda = \max_{\{n\}} \{\lambda_n\}$  is the one transmitting. This is repeated at each transmission opportunity.

3) *Relation to NCV and NCVD*: The NC-RaDiO framework includes NCV and NCVD as special cases for a system that makes the following implementation choices (consistently with COPE [7], [8] and the system discussed here):

- All packets have the same size (possibly using padding):  $B(j) = B, \forall p_j \in \Phi_n, n = 1, \dots, N$ .
- A deterministic rule is used to decide whether a previously transmitted packet is lost or not: if average RTT time ( $RTT_{avg}$ ) has passed since its last transmission and no ACK has been received, the packet is considered lost; otherwise, it is considered successfully received. Let  $\tau(j)$  be the time duration since the most recent transmission of packet  $p_j$ , if  $p_j$  has been transmitted before; or  $\tau(j) = \infty$

if  $p_j$  has not been transmitted before. Then, we can rewrite the probability of loss in previous transmissions  $P_p(j)$ , in the NC-RaDiO formulation, as follows:

$$P_p(j) = \begin{cases} 1 & \text{if } \tau(j) \geq RTT_{avg} \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, we note the correspondence between the NCV/NCVD algorithms and the NC-RaDiO formulations.

- Let  $d(\pi_n(j))$  be the indicator function, which equals 1 if packet  $p_j$  is decodable at its next hop node when transmitted with network code  $c_u \in C_n$ ; or 0 otherwise. This corresponds to the decodability indicator function  $d_l^k(n_\eta)$  in NCV/NCVD.
- The probability of loss in the current transmission,  $P_c(\pi_n(j))$ , in the NC-RaDiO formulation, can be re-written as follows:

$$P_c(\pi_n(j)) = \begin{cases} P\{FTT' > t_d(j) - t_c\} & \text{if } d(\pi_n(j)) = 1 \\ 1 & \text{if } d(\pi_n(j)) = 0 \end{cases}$$

Then  $1 - P_c(\pi_n(j)) = (1 - P\{FTT' > t_d(j) - t_c\})d(\pi_n(j))$ . Further considering Eq. (3), it turns out that  $1 - P_c(\pi_n(j)) = (1 - P(j))d(\pi_n(j))$ .

Under the above assumptions and notations, the Lagrange multipliers in the NC-RaDiO formulation can be re-written:

$$\lambda'_n(c_u) = \sum_{p_j \in c_u \text{ s.t. } \tau(j) \geq RTT_{avg}} \gamma(j)\Delta(j)(1 - P(j))d(\pi_n(j)) \quad (15)$$

Note that the Lagrange multiplier  $\lambda'_n(c_u)$  in Eq. (15) is equivalent to the improvement value  $I_k^i$  in Eq. (5).<sup>6</sup> NCV and NCVD are suboptimal solutions to the NC-RaDiO optimization problem due to: (i) the aforementioned assumptions, *i.e.* not taking into account the exact packet size or the effect of previous transmissions and (ii) the fact that nodes in a practical low-complexity system (such as COPE or NCV/NCVD) take local decisions and do not exchange the improvement values (Lagrange multipliers) to decide which node should transmit. However, the equivalence of Eq. (15) and Eq. (5) is the intuition why NCV and especially NCVD are efficient heuristics to the NC-RaDiO problem. In the next section, we confirm via simulation that the performance of our algorithms is near the optimal NC-RaDiO in a wide range of scenarios.

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed schemes (NCV, NCVD, and NC-RaDiO) in terms of video quality and network throughput in a wide range of scenarios. We compare them to four baseline schemes: no network coding (*noNC*), multimedia streaming (*MM*), network coding optimized for throughput (*NCT*) as in [7], and an improved version of it (*NCTD*). Simulation results show that (i) NCV, NCVD, and NC-RaDiO can significantly improve video quality and application-level throughput, without compromising

MAC-level throughput and (ii) NCVD is an efficient heuristic solution to NC-RaDiO. In V-A we describe the simulation setup, in V-B we present the simulation results, and in V-C we discuss complexity issues.

### A. Simulation Setup

In this paper, we used the GloMoSim simulation environment [28] to implement the proposed algorithms and the baseline schemes. Below, we describe the simulation setup, which includes: the topologies and traffic scenarios considered, the MAC model, the wireless channel model, the video sequences, and the baseline algorithms used for comparison.

1) *Simulation Topologies*: The topology and traffic scenario can strongly affect the gain from using network coding. We considered three practical scenarios shown in Fig. 4.

**Single-Hop Downlink Topology**: In this topology, we consider the single-hop downlink scenario shown in Fig. 4(a). The intermediate node  $I$  receives different video streams, which it forwards downstream towards their destinations.  $I$  can apply different schemes for network coding and packet scheduling. We assume that receivers are placed on a circle with radius 90m and the intermediate node  $I$  which is placed in the center of the circle. Receivers are the only ones using the downlink, hence there is no congestion. However, packets may still be lost due to errors on the wireless channel, and can also experience a random MAC propagation delay, 2ms on average. The one-way delay budget for this single-hop is set to 100ms. We also performed simulations for different delay budgets (50-200ms), for different number of nodes ( $N : 3-11$ ) including the intermediate node and the receivers, and for different channel conditions.

**Cross Topology**: In this topology, we consider multiple crossing flows at an intermediate node as shown in Fig. 4(b): pairs of nodes  $A, C$  and  $B, D$  communicate over an intermediate node  $I$ , *e.g.*,  $A$  transmits to  $C$  and  $C$  transmits to  $A$  via  $I$ . A single channel is used for both uplink and downlink transmissions. MAC scheme will be explained later in the section. In this scenario, each node buffers a packet it has just transmitted as well as all overheard packets; an illustrative example is shown in Fig. 4(b). The intermediate node  $I$  makes decisions on network coding and scheduling. We assume again that nodes are placed on a circle with center  $I$  and radius 90m. The remaining settings are similar to the single-hop downlink.

**Grid Topology**: In this topology, we consider the wireless mesh network (WMN) shown in Fig. 4(c). Nodes are distributed over a 300m×300m terrain according to a grid topology: the area is divided into 9 cells of equal size, 20 nodes are divided into 2 or 3 node sets randomly, and each set is assigned to a different cell. Nodes in a set are randomly placed within their assigned grid. The WMN is connected to the Internet via a high speed lossless link through a gateway  $I$ , placed in the upper leftmost grid shown in Fig. 4(c). Each node receives a video stream from the Internet going through the gateway. Depending on the location of the receiving node  $R$ , the stream is either transmitted directly (one-hop) or routed (two hops). If both  $I$  and  $R$  are either in the same cell or in neighboring cells, there will be a one-hop transmission; otherwise a node in the cell between  $I$  and  $R$  is selected as

<sup>6</sup>In Eq. (15), the  $g_l^k(n_\eta)$  term does not exist, because its value is naturally 1 since we only consider the improvements of packets at their next hop nodes, instead of considering the possible improvement at all nodes in the neighborhood as in Eq. (5). However, this difference is only a matter of notation and it clear that Eq. (15) and Eq. (5) are equivalent.



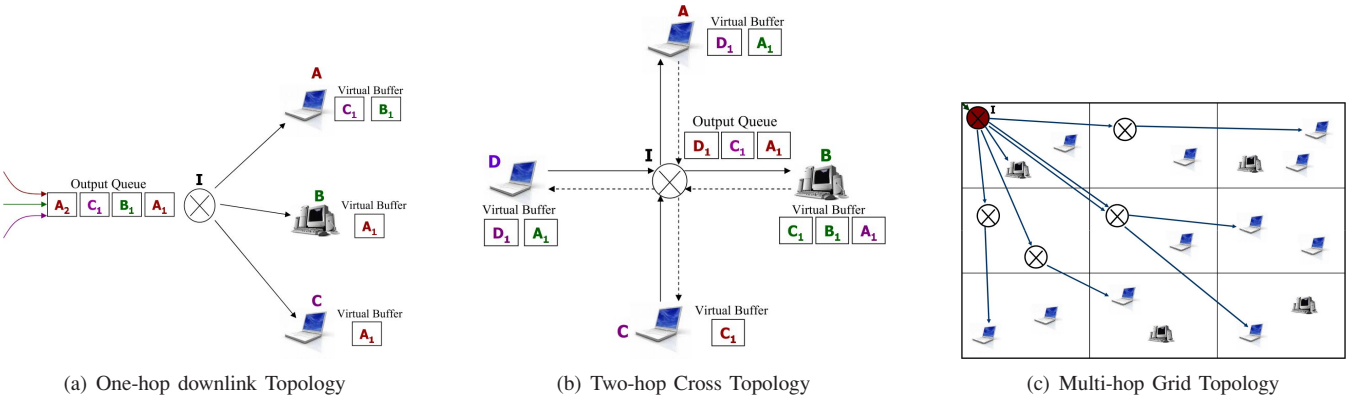


Fig. 4. Topologies and traffic scenarios used in simulations. (We also vary the number of nodes in each topology.)

an intermediate hop, indicated by “ $\otimes$ ” on Fig. 4(c); if there are more than one neighboring cells, one is selected randomly. The remaining settings are similar to the single-hop downlink.

2) *MAC Model*: IEEE 802.11 is used in the MAC layer, with the following modifications needed for network coding. First, to obtain the network coding benefit we need a broadcast medium, which is hidden by the 802.11 protocol. Similarly to [7] we used the pseudo-broadcasting mechanism: packets are XORed in a single unicast packet, an XOR header is added for all nodes that should receive that packet, and the MAC address is set to the address of one of the receivers. A receiver knows whether a packet is destined to it from the MAC address or the XOR header. Second, 802.11 waits for an ACK after a packet is transmitted, which reduces the network coding opportunities and increases the overhead. Instead, we consider that packets are transmitted one after the other without waiting for an ACK and the “active-inactive” mechanism is used to reduce unnecessary re-transmissions. For the NC-RaDiO scheme in particular, we assume that the Lagrange multipliers are exchanged through a separate channel.

3) *Wireless Channel Model*: We consider the two-ray path loss model and Rayleigh fading channel model implemented in GloMoSim. The two-ray path loss model is a propagation path loss model using free space path loss for near sight and plane earth path loss for far sight. For the Rayleigh fading model, we consider average channel SNR  $\{3, 5, 7, 9\}$  dB.

4) *Video Sequences*: As our test sequences, we used standard sequences: *Carphone*, *Foreman*, *Mother & Daughter*, *Claire*, *Coastguard*, *News*, *Grandma*, and *Salesman*. These were QCIF sequences encoded using the JM 8.6 version of the H.264/AVC codec [32], [33]. The group of pictures consisted of one I and nine P frames. All encoded sequences had data rate 70 kbps and frame rate 30 fps. Each frame consists of at least one slice. Each slice was packetized into an independent NAL (network abstraction layer) unit of size 250B. NAL units are encapsulated using the Real-time Transport Protocol (RTP) and User Datagram Protocol (UDP).

As metric for the video quality of an encoded sequence, we use the average PSNR, *i.e.*, the peak-signal-to-noise ratio based on the luminance (Y) component of video sequences, measured in dB, and averaged over the entire duration of the video sequence. The PSNR of the encoded sequences *Carphone*, *Foreman* and *Mother & Daughter*, before any transmission, was 29.95dB, 28.70dB and 40.74dB respectively; these PSNR values are denoted as “No Error” in Table II. We

repeated and concatenated the standard sequences to create longer test sequences of duration 30sec each. At the receiver side, basic copy-concealment scheme is used when an entire frame is lost.

5) *Baseline Algorithms for Comparison*: We compare our algorithms, NCV, NCVD, and NC-RaDiO, against four baselines for packet scheduling: no Network Coding (noNC), Multimedia Streaming Algorithm (MM), Network Coding for Throughput (NCT), and its improved version NCTD.

Fig. 5 summarizes all algorithms and classifies them, in increased sophistication, across two dimensions: packet scheduling and network coding. noNC takes no action in either dimension - it is a simple FIFO. NCT and NCTD do network coding and combine several packets in one transmission so as to maximize throughput; NCT optimizes only the side packet selection while NCTD optimizes both primary and side packet selection. Both NCT and NCTD are agnostic to the content of the packets. In contrast, MM does not use any network coding but prioritizes packet transmission, considering packet distortion and deadlines. The proposed algorithms, NCV and NCVD, combine both ideas and NC-RaDiO further extends these ideas by prioritizing nodes. NCVD can be thought as a combination of network coding (NCTD) and content awareness (MM). NCV can be thought as a combination of NCT and MM, but unlike MM, it is restricted in its choice of primary packet. For a fair comparison, the active-inactive mechanism described in section III, is employed in all algorithms except for NC-RaDiO (which relies on the success probabilities of previous transmissions).

**No Network Coding (noNC)**: This is a FIFO Tx queue without network coding. Consider again Example 1 and Fig. 2: node *I* stores packets for all three streams destined to nodes *A*, *B*, *C*. In every time slot, *I* transmits the first packet from the head of the queue.

**Multimedia Streaming Algorithm (MM)**: This is a scheduling scheme that optimally chooses the packet to be transmitted without network coding. We consider it in order to see how much benefit comes from prioritized packet transmission alone, apart from network coding. MM is essentially a reduced version of NCVD with network codes having one packet at most (no network coding), *i.e.*, for primary packet  $p_i$ , and the only eligible network code being  $c_0^i = p_i$ . The improvement at node  $n_\eta$  is  $I_0^i(n_\eta) = (1 - P(i))\Delta(i)g_0^i(n_\eta)$ , where  $P(i)$  is the loss probability of packet  $p_i$  given in Eq. (3),  $\Delta(i)$  is the improvement of video quality if packet  $p_i$  is

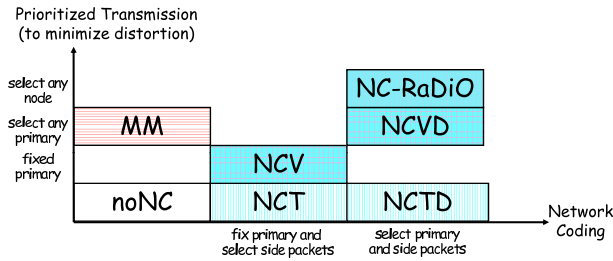


Fig. 5. Summary of Algorithms under Comparison. NCV, NCVD, and NC-RaDiO are the proposed algorithms that combine packet scheduling and network coding. The rest are baselines that use at most one of the mechanisms: noNC is simply FIFO, MM uses optimal packet selection to minimize distortion, NCT and NCTD use network coding to maximize throughput.

received correctly and on time at client  $n_\eta$ , and  $g_i^0(n_\eta)$  is the indicator function that shows whether packet  $p_i$  is destined to node  $n_\eta$ . Among all packets in the Tx queue, MM selects the  $p_i$  that maximizes the total video quality, considering the improvement to all nodes in its neighborhood:

$$p_i = \underset{i}{\operatorname{argmax}} \sum_{\eta=1}^N I_0^i(n_\eta) \quad (16)$$

In other words, MM transmits the most important packet in the Tx queue, considering per packet distortion and loss probability across all streams.

**Network Coding for Throughput (NCT):** This is an improved version of the algorithm proposed in [7]. The packet transmission mechanism is the same as in the noNC scheme, but network coding is used to maximize throughput, as follows. The first active packet in the Tx queue is selected as primary; side packets are chosen to be XOR-ed together with the primary packet so as to construct a network code that is decodable by the maximum number of receivers possible.

There are two improvements in NCT compared to the coding algorithm in [7] that allow NCT to achieve even higher throughput than [7]. First, NCT follows the same ACK and retransmission mechanism described in section III: packets with pending acknowledgments are marked as inactive for one RTT, while the channel is used to transmit other packets as primary. In [7] and in general MAC retransmissions, a packet stays at the head of the queue blocking other packets, until it goes through successfully or it exceeds the maximum number of retransmissions. Another difference is that NCT uses an improved version of the coding procedure in [7]: NCT considers all possible subsets of the candidate side packets thus maximizing the number of receivers that can decode; while [7] considers only the earliest packet from each stream as candidate side packets, thus sacrificing some throughput for reduced complexity and for maintaining the packet ordering. Therefore, we use NCT as our baseline for the maximum achievable throughput per transmission using network coding.

**NCT & looking into the queue in Depth (NCTD):** NCT selects as primary packet the first active packet in the Tx queue. Similarly to NCV, this limits the candidate codes to those that are decodable only for this single primary packet. Similarly to NCVD, we extend NCT to NCTD, which looks

into the entire Tx queue and considers all packets as candidate for the primary packet. Thus, NCTD optimizes throughput by primary packet selection and network code construction.

## B. Simulation Results

In this section, we present simulation results that compare the proposed to baseline algorithms and demonstrate that the former can improve video quality and application-level throughput, without compromising MAC-level throughput.

1) *Summary of Results:* Some general observations across all simulation scenarios are summarized below:

*The best and worst algorithms:* The optimal solution to NC-RaDiO is clearly the best in terms of PSNR and close to the best in terms of throughput. NCVD closely approximates the optimal solution of NC-RaDiO in terms of both PSNR and throughput. As expected, noNC is consistently the worst algorithm in all aspects.

*Media awareness added on network coding:* The proposed media-aware network coding algorithms, NC-RaDiO, NCVD, and NCV consistently outperform the corresponding network coding-only algorithms, NCTD and NCT, in terms of PSNR, while achieving similar throughput.

*Media awareness vs. network coding:* The MM algorithm achieves higher PSNR than the weaker (i.e. other than NCVD) network coding algorithms (NCT, NCTD, NCV) in harsh channel conditions while the network coding algorithms (NCT, NCTD, NCV) perform better in mild channel conditions. In other words, in the former case the quality/importance of the transmitted packets matters, while in the latter case the quantity has a greater effect. Note that NC-RaDiO and NCVD achieves the highest PSNR in all conditions.

*Throughput:* In addition to achieving the highest PSNR, NCV, NCVD, and NC-RaDiO achieve higher application throughput, since they consider playout deadlines. Furthermore, all network coding schemes (NCT/NCTD, NCV/NCVD/NC-RaDiO) achieve similar MAC throughput, much higher than the non-network coding schemes (noNC, MM).

*Node selection:* NC-RaDiO performs slightly better than NCVD, in terms of PSNR, thanks to (i) explicit consideration of previous transmission probabilities and packet sizes and (ii) exchanging Lagrange multipliers among nodes to select a node to transmit. However, the performance improvement of NC-RaDiO over NCVD is negligible in a wide range of scenarios, indicating that NCVD is an efficient heuristic for the NC-RaDiO optimization problem.

*Primary packet selection:* NC-RaDiO and NCVD achieve higher PSNR than NCV, thanks to their node selection (NC-RaDiO) and primary packet optimization (NC-RaDiO and NCVD). MM outperforms NCV (but never NC-RaDiO and NCVD) in some scenarios, for the same reason, i.e. because NCV can only choose the side but not the primary packet. The optimization of primary packet selection is more important for the media-aware than for the network coding schemes: the similar performance of NCT and NCTD indicates that the primary packet optimization does not significantly increase the number of packets in a network code and thus the throughput.

*Comparison of topologies:* The one-hop downlink is a building block for other topologies and can be used as a baseline

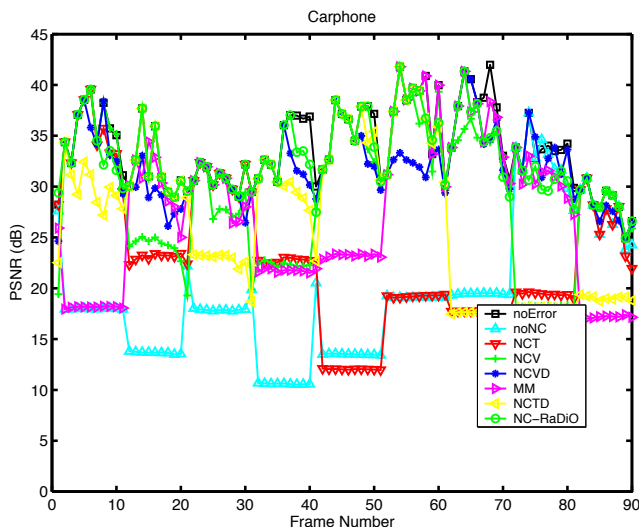


Fig. 6. PSNR per frame for part of the Carphone sequence and for an example channel realization. *One-hop* downlink scenario with  $N = 4$  ( $I$  and three receivers  $A, B, C$ ), channel SNR = 5dB, delay budget 100ms, and data rate 500kbps. Seven schemes (noError, noNC, MM, NCT, NCTD, NCV, NCVD, NC-RaDiO) are compared. (The average PSNR values over the entire sequence are summarized in Table II.)

for comparison; it does not incur any delay in accessing the channel and all flows are network-coded. Furthermore, this is the only topology in which there is no need to exchange Lagrange multipliers among nodes. In the cross topology, there is more delay due to the two-hop transmission and in accessing the uplink channel; therefore there are less network coding opportunities and the network coding schemes perform worse. Similar arguments apply to the grid topology.

We now discuss each simulation scenario in detail.

2) *Single-Hop Downlink Topology*: We consider the single-hop downlink topology of Fig. 4(a), when node  $I$  streams sequences *Foreman*, *Mother & Daughter*, and *Carphone* to clients  $A, B, C$ , respectively. When the number of nodes ( $N$ ) is greater than four the video sequences *Coastguard*, *Salesman*, *News*, *Grandma*, and *Claire* are used sequentially. First we focus on the scenario with three receivers in the system and evaluate the performance of the algorithms for different delay budgets and channel SNR levels.

**Video Quality Improvements**: Fig. 6 shows the video quality experienced by one client (PSNR over frame number for parts of the Carphone sequence) for the seven algorithms under comparison, namely noNC, MM, NCT, NCTD, NCV, NCVD, and NC-RaDiO as well as for the encoded sequences before transmission (noError). The simulation is performed for channel SNR 5dB with 100 ms delay budget and 500kbps channel data rate; for comparison, the same wireless channel trace is used as input to all six algorithms. As expected, there are time periods, during which the channel is bad, the quality degrades for all algorithms. However, the degradation for NC-RaDiO, NCVD, and NCV is much less than for NCTD, NCT, and noNC, because NC-RaDiO, NCVD, and NCV select network codes to protect and deliver the most important packets on time, thus improving the video quality; in contrast, NCTD, NCT, and noNC treat all packets similarly. The degradation of MM is less than NCT, NCTD, and noNC

TABLE II  
AVERAGE PSNR FOR THE SCENARIO OF FIG.6 (VIDEO: AT 70KBPS AND 100MS PLAYOUT DEADLINE; CHANNEL WITH SNR=5DB AND 500KBPS DATA RATE)

| avg PSNR (dB) | Carphone | Foreman | Mother&Daughter |
|---------------|----------|---------|-----------------|
| No Error      | 29.95    | 28.70   | 40.74           |
| NC-RaDiO      | 28.46    | 27.51   | 35.08           |
| NCVD          | 27.98    | 26.87   | 35.36           |
| NCV           | 25.40    | 25.14   | 28.66           |
| NCTD          | 24.91    | 24.60   | 28.61           |
| NCT           | 23.95    | 24.38   | 27.19           |
| MM            | 25.17    | 24.61   | 32.12           |
| noNC          | 22.32    | 22.64   | 23.84           |

for most of the region even when the channel goes bad, because it transmits more important packets and is comparable to or worse than NCV, and worse than NCVD and NC-RaDiO, because NCV and NCVD transmit more packets using network coding as well as considering the importance of some (NCV) or all (NCVD, NC-RaDiO) packets.

The average PSNR for each sequence and algorithm is summarized in Table II. We see that, as expected, the noNC scheme performs poorly. NCT improves over noNC because it delivers more packets per time slot. NCV improves over NCT because it chooses important video packets as side packets, NCTD improves over NCT, because it also optimizes the primary packet selection. MM outperforms noNC, NCT, and NCTD. This result is quite interesting, because NCT and NCTD transmit more information than MM; however, not only the amount but also the content of information transmitted is important. MM exhibits similar or better performance than NCV. This is intuitive, because NCV optimizes side packets for video quality improvement but not primary packets; since primary packets are the main packets that are transmitted to all receivers, optimized packet scheduling is performed only in a few of the transmitted packets. NCVD achieves higher video quality compared to NCV, NCTD, NCT, MM, and noNC, since both primary and side packet selection is optimized. NC-RaDiO is slightly better than NCVD thanks to considering different packet sizes and to the exact calculation of previous packet transmission probability. However, since packet sizes are almost the same in different packets, and the deterministic decision is a good estimator of probabilistic decision, the improvement of NC-RaDiO over NCVD is very small. Actually, both NC-RaDiO and NCVD achieve a PSNR close to that of the original encoded sequence (noError), even for harsh channel conditions (e.g., 5dB channel SNR).

The same scenario as in Fig. 6 is considered, but with channel SNR varying in a range from 3dB to 11dB. Fig. 7(a) shows the average PSNR achieved by each algorithm. Clearly, NC-RaDiO, NCVD and NCV outperform NCT (by 1 – 4dB) and noNC (by 3–5 dB) for all channel SNRs. NCT and NCTD exhibit similar performances. NC-RaDiO and NCVD always outperforms to MM (by 3dB). MM outperforms NCT, NCV and NCTD for low channel SNR levels, but does not for higher channel SNR levels. When the channel is good (11dB channel SNR), all algorithms transmits almost all of their packets. For a worse channel (SNR 9dB), the network coding algorithms NC-RaDiO, NCVD, NCV, NCTD, NCT are better than the no network coding schemes (noNC and MM), because they



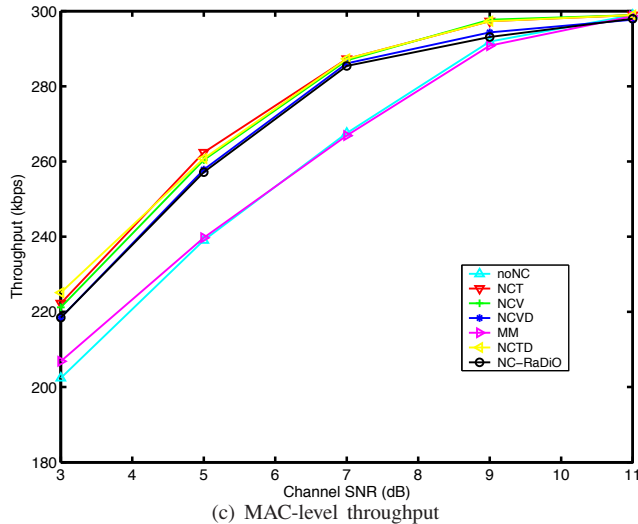
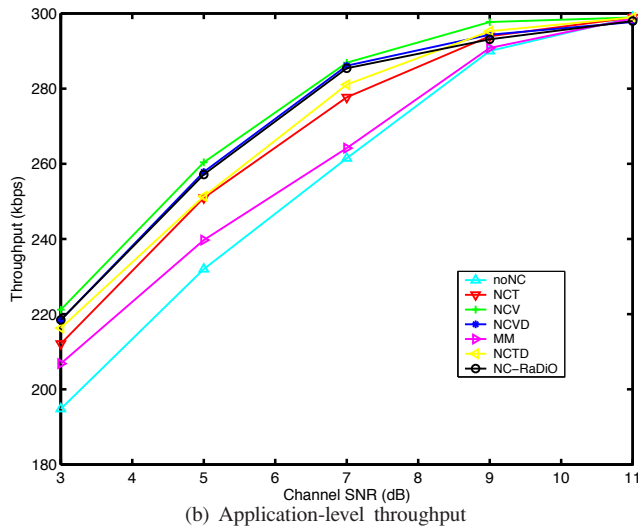
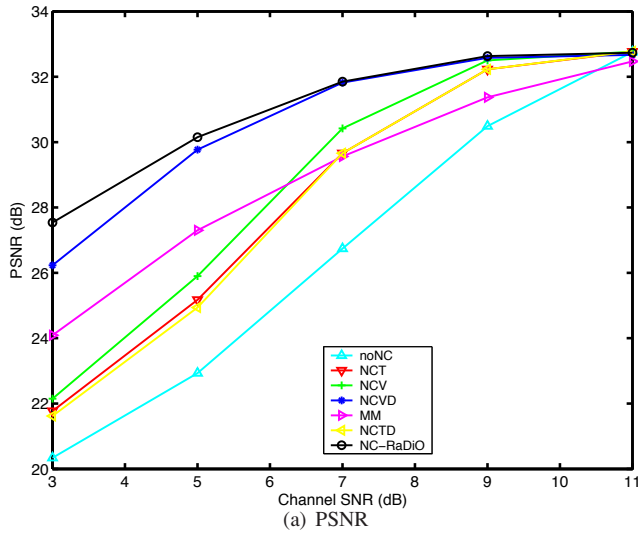


Fig. 7. *One-hop downlink topology* with three receivers (and one intermediate node  $I$ , i.e.  $N = 4$ ). Performance for different channel SNR levels in terms of: (a) PSNR (averaged across each sequence and across all three sequences) (b) total application-level throughput (added over all three streams) (c) total MAC-level throughput. (The delay budget is 100ms and the data rate is 500kbps.)

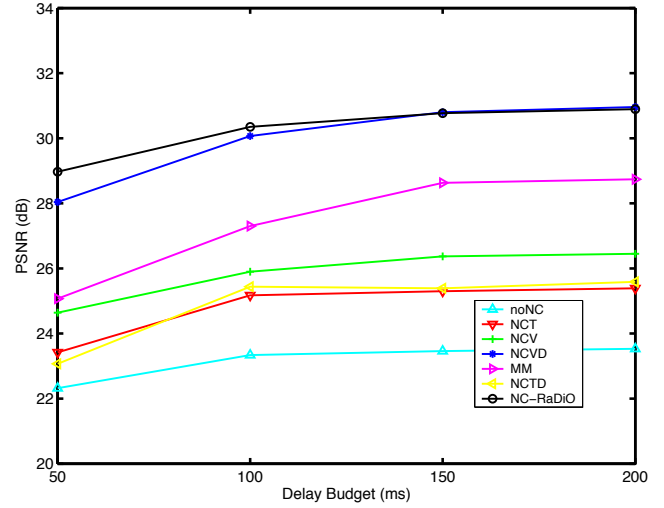


Fig. 8. PSNR values (averaged over each sequence and across sequences) for different delay budgets in *downlink topology*,  $N = 4$ . Channel SNR is 5dB and data rate is 500kbps.

transmit more packets. When the channel is really bad (5dB channel) NC-RaDiO, NCVD and MM outperform NCTD, NCT, NCV because some packets have to be dropped under these harsh conditions, even if network coding is used, and the quality of selected packets has a dominant effect. Actually, as we will show later, MM transmits less packet than NCTD, NCT and NCV but since it optimizes packet scheduling, the video quality remains high even for bad channels. NC-RaDiO and NCVD outperforms MM for all channel SNR levels, because they have the advantages of both network coding and multimedia streaming: they transmit more packets using network coding and do packet scheduling considering packet importance and deadlines. NC-RaDiO improves slightly over NCVD, especially for harsh channel conditions.

In the scenarios discussed so far, we have considered a delay budget of 100ms. In Fig. 8, we show the average PSNR for a delay constraint ranging from 50 to 200ms. NC-RaDiO, NCVD, and NCV improve video quality for the entire range of delay values as compared to NCTD, NCT and noNC. MM is better than NCV since NCV lacks primary packet scheduling. The network coding algorithms bring less improvement for a tight delay budget, which limits the number of retransmissions and the lifetime of packets both at the Tx queue and in the virtual buffers, thus decreasing network coding and selection opportunities. However, even with tight delay constraints, there is significant video quality improvement from NC-RaDiO and NCVD compared to all other algorithms.

**Throughput Improvements:** The video-aware schemes improve video quality because they explicitly take it into account in the code selection. In this section, we show that, our schemes also significantly improve application-level throughput while maintain the same levels of MAC-level throughput. In other words, our algorithms deliver the same amount of packets but choose to deliver more useful video packets.

*Application Throughput.* Fig. 7(b) shows the total throughput as seen by the application-layer (i.e., NAL units per sec) added over all clients. The figure clearly shows that NC-RaDiO, NCVD, and NCV achieve higher throughput as com-

pared to NCT, NCTD, noNC and MM. The main reason is that NC-RaDiO, NCVD, and NCV do not select codes consisting of packets whose deadlines are within one transmission time, while NCT and NCTD transmit all packets. Late packets do not contribute to application-level throughput, because those packets are discarded at the client even if they are received successfully. All the network coding schemes (NC-RaDiO, NCVD, NCV, NCTD and NCT) are better than non-network coding schemes (MM and noNC) because they transmit more packets. As expected, MM is better than noNC, because it considers packet deadlines for packet scheduling.

*MAC Throughput.* For completeness, we also show the MAC-layer throughput in Fig. 7(c). As expected, all network coding schemes (NC-RaDiO, NCVD, NCV, NCTD, NCT) achieve higher MAC-level throughput than noNC and MM, because they convey more information content per transmission. Interestingly, all network coding schemes achieve similar throughput, although NCT/NCTD are the ones explicitly designed to maximize throughput.

Next, we consider the performance of the proposed algorithms, when varying the number of nodes ( $N$ ), for fixed delay budget 100ms, data rate 1Mbps, and channel SNR 5dB. Fig. 9(a) shows the average PSNR (averaged over three video sequences; *Foreman*, *Carphone*, *Mother & Daughter* when  $N = 4$  or higher; or averaged over two video sequences, namely *Carphone* and *Mother & Daughter*, when  $N = 3$  in the system. Fig. 9(b) and (c) are the application and MAC level throughput seen at all receivers, respectively.

Fig. 9(a) shows that PSNR values of all algorithms are almost the same when  $N = 3$ . The reason is that the data rate (1Mbps) is sufficiently large to transmit and re-transmit almost all packets. When  $N$  increases, the PSNR of all algorithms decreases. When  $N = 5$ , NC-RaDiO, NCVD, NCV, NCTD, and NCT have almost the same PSNR while noNC and MM start deteriorating. The reason is that the network coding algorithms transmit effectively more packets than the non-network coding algorithms (noNC and MM). For  $N = 6$ , the network coding algorithms are still better than noNC and MM; we also note that NC-RaDiO, NCVD, NCV, and NCTD are better than NCT because NC-RaDiO, NCVD and NCTD transmit more packets due to primary packet optimization, and NCV transmits more important packets. When  $N$  increases further, the PSNR performance becomes more interesting: noNC is the clearly the worst; NCT and NCTD are similar to each other and better than noNC; NCV is better than NCT and NCTD, because it optimizes side packet selection, hence transmits more important packets. The most interesting part is that while the PSNR of noNC, NCT, NCTD, and NCV decreases sharply, the decrease in MM's PSNR is roughly linear. The reason is that MM utilizes limited resources to transmit important packets. On the other hand, NCT and NCTD combines packets to transmit effectively more packets; however, since they do not consider the deadlines of the packets, they transmit obsolete packets. NCV is also worse than MM, because it does not optimize the primary packets for video quality. However, when the resources get scarce (larger  $N$ ), the optimal selection of each packet becomes more important than the amount of data transmitted. NC-RaDiO and NCVD outperform all algorithms by 2dB-5dB.

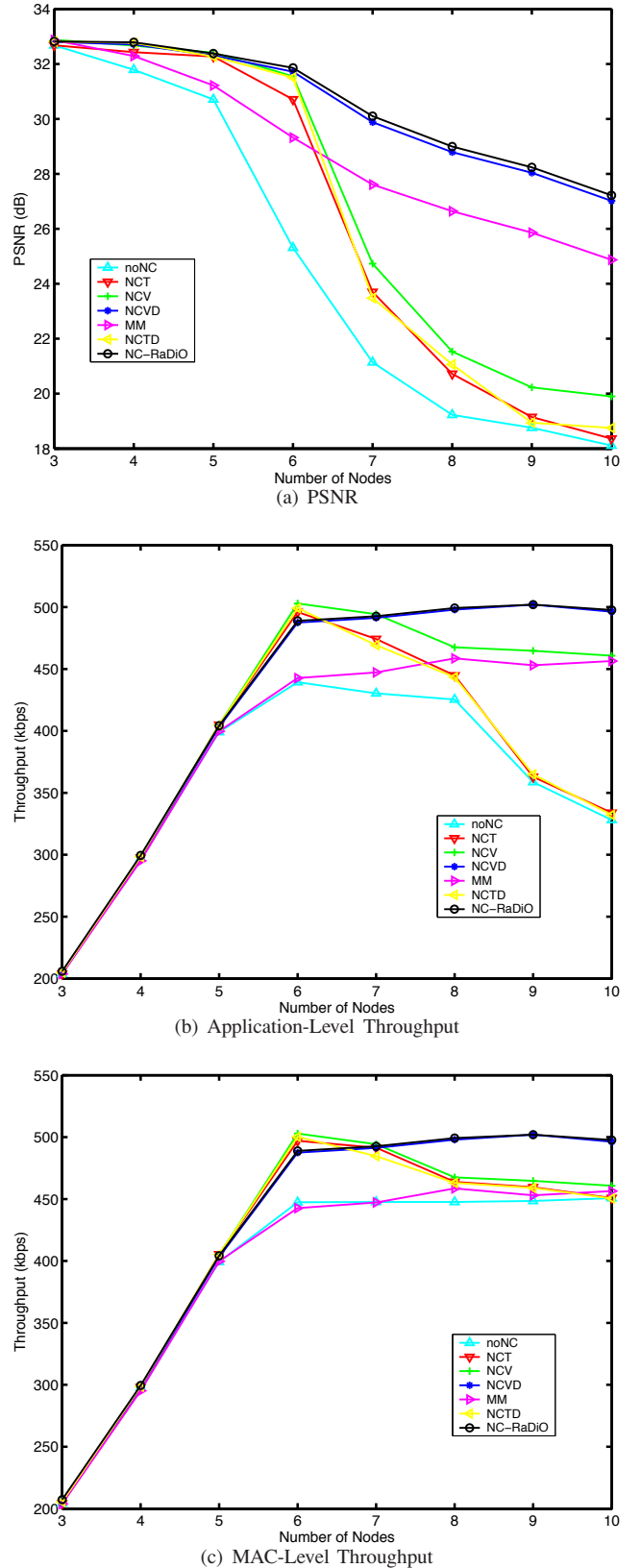


Fig. 9. *One-hop downlink topology* for different number of nodes in the system. Performance of all algorithms in terms of (a) video quality (PSNR) (b) application-level and (c) MAC-level throughput. (Channel SNR is 5dB, delay budget is 100ms, and channel data rate is 1Mbps).

The application- and MAC-level throughput are shown in

Fig. 9(b) and Fig. 9(c). For up to  $N = 5$ , all algorithms deliver the same amount of packets. For  $N = 5, \dots, 8$ , the network coding algorithms deliver more data (both application and MAC level) as compared to noNC and MM. For  $N > 8$ , the MAC throughput of NCV, NCT, and NCTD decreases, for two reasons: (i) having more streams sharing the same Tx queue decreases the lifetime of packets, hence the network coding opportunities (ii) NCT and NCTD transmit obsolete packets. NCV and MM have similar application throughput, since there are less network coding opportunities for NCV with increasing  $N$ . NC-RaDiO and NCVD achieve the highest application and MAC level throughput, because they create more network coding opportunities.

3) *Cross Topology*: We consider the cross topology shown in Fig. 4(b) when  $A, C$  transmit *Foreman* and *Mother & Daughter* to each other and  $B, C$  transmit *Carphone* and *Coastguard* to each other over the intermediate node  $I$ . When the number of nodes ( $N$ ) gets larger *Salesman*, *News*, and *Grandma*, *Claire* pairs are used sequentially. First, we focus on the scenario with four nodes actively transmitting and receiving video ( $N = 5$  including  $I$ ) and evaluate the performance of the algorithms for different delay budgets and channel SNR. We fix the delay budget to 100ms, the data rate to 1.3Mbps and vary the channel SNR from 3 to 5dB.

Fig. 10 shows the performance of all algorithms in this topology and for channel SNR in the range 3-11dB. Fig. 10(a) shows the PSNR: The ranking of the algorithms in decreasing PSNR is similar to the downlink scenario shown in Fig. 7(a). However, there are some differences. First, the performance gap between NC-RaDiO and NCVD is larger in the cross topology, because the exchange of Lagrange multipliers among nodes, to decide which node should transmit, becomes more important since all nodes transmit. NCVD is again very close to the optimal NC-RaDiO, which confirms that it is a good heuristic. Second, in the downlink scenario, all algorithms have the same PSNR at channel SNR 11dB, while this is not the case in the cross topology. Even with a good channel (11dB), there are still packet lost in the channel. In the cross topology, more packets are lost, on the uplink and the downlink channels. Third, in the downlink topology, the network coding algorithms improve PSNR more than MM, for channel SNR greater than 7dB. However, in the cross topology MM's improvement is higher than that of NCV, NCT, and NCTD for all channel SNR levels since MM optimizes packet transmission in both uplink and downlink.

Fig. 10(b) and Fig. 10(c) depict the application and MAC throughput, respectively, for the same setting as in Fig. 10(a). noNC and MM have the same MAC throughput while all network coding algorithms have similar MAC throughput. MM achieves slightly higher application-level throughput than noNC. NC-RaDiO, NCVD, NCV, NCTD, and NCT achieve decreasing order of application level throughput. As compared to the downlink scenario and the throughput values shown in Fig. 7, the throughput difference between network coding and no network coding algorithms is less in the cross topology compared to other topologies; the reason is that there are more independent flows and thus less network coded packets.

Fig. 11 shows the PSNR achieved by all algorithms when we vary the number of nodes in the cross topology. We fix the

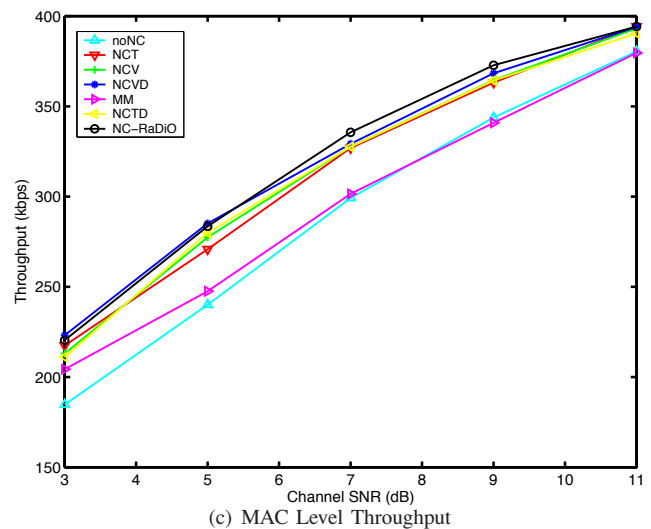
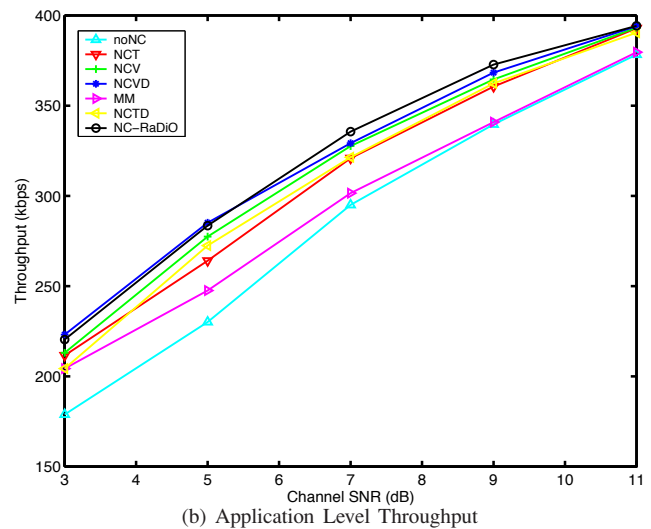
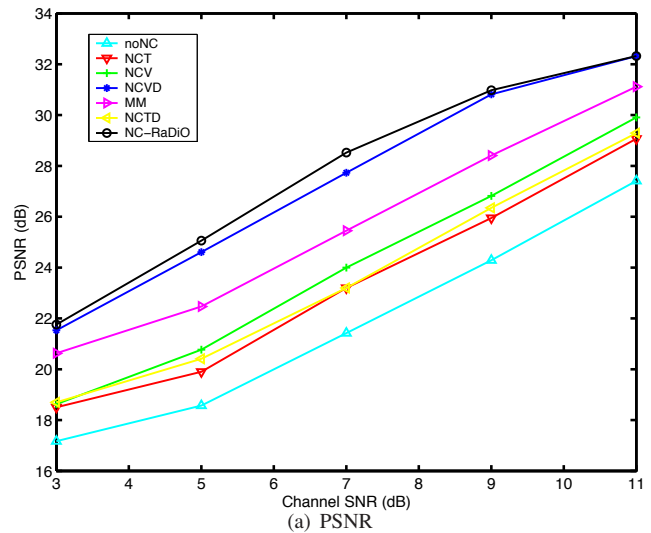


Fig. 10. *Cross topology*, with  $N = 5$  nodes including  $I$ , for different channel SNR levels. Performance in terms of (a) PSNR (b) application throughput and (c) MAC throughput. (Delay budget is 100ms, channel SNR is 5dB, and data rate is 1.3Mbps.)

channel SNR to 5dB, the delay budget to 100ms, and the data



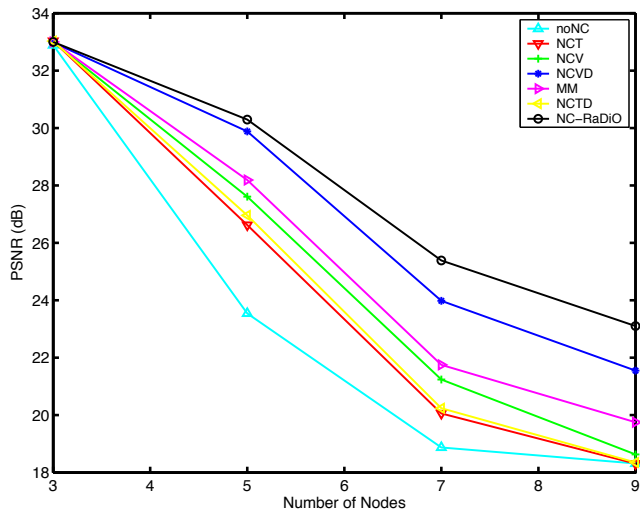


Fig. 11. *Cross Topology*. PSNR performance for a different number of nodes ( $N$ ). (Channel SNR is  $5\text{dB}$ , delay budget is  $100\text{ms}$ , and data rate is  $2\text{Mbps}$ .)

rate to  $2\text{Mbps}$ . NC-RaDiO and NCVD perform best, MM is second for the interesting part of the  $N$  range; NCV is better than NCT and NCTD which exhibits similar performance; and noNC is the worst as expected. If we compare this graph with the corresponding graph shown in Fig. 9(a), we see again that the benefit of network coding is less in the cross topology. One reason is that there are more independent than network coded flows in the system. Another reason is the increased delay in the two-hop transmission, while the deadline remains the same. However, NC-RaDiO and NCVD still improve over all other algorithms by  $2.5 - 5\text{dB}$ .

4) *Grid Topology*: We consider the grid topology shown in Fig. 4(c).  $I$  receives sequences over a high-speed error-free link and transmits a different sequence to each receiver over the grid topology, using one-hop or two-hops. In particular, node  $I$  goes sequentially through the list of the 8 available videos (*Foreman, Mother & Daughter, Carphone, Coastguard, Salesman, News, Grandma, and Claire*) and sends one to each receiver. When there are more than 8 nodes, the  $8^{\text{th}}$ ,  $9^{\text{th}}$ , etc. sequence is chosen from the beginning of the list (*Foreman, Mother & Daughter, etc.*).

In Fig. 12, we show the PSNR achieved by all algorithms for a varying number of streams in this topology. *E.g.*,  $N = 5$  means there is one transmitter  $I$  and 4 receivers over either one- or two-hops. The delay budget is  $100\text{ms}$ , the channel SNR  $5\text{dB}$  and the channel data rate  $1\text{Mbps}$ . The figure shows a similar trend with the corresponding graph for the downlink topology in Fig. 9(a). This is because the traffic scenarios in the downlink and the grid topologies are similar: the grid scenario consists of one and two hop downlink transmissions. However, there are two differences. First, the decrease in PSNR is sharper in the grid topology: when the number of nodes increases, more nodes are involved in one- and two-hop transmissions, as compared to the downlink topology. Second, the difference between network coding and non network coding schemes is smaller, because after the first hop there are not many network coding opportunities.

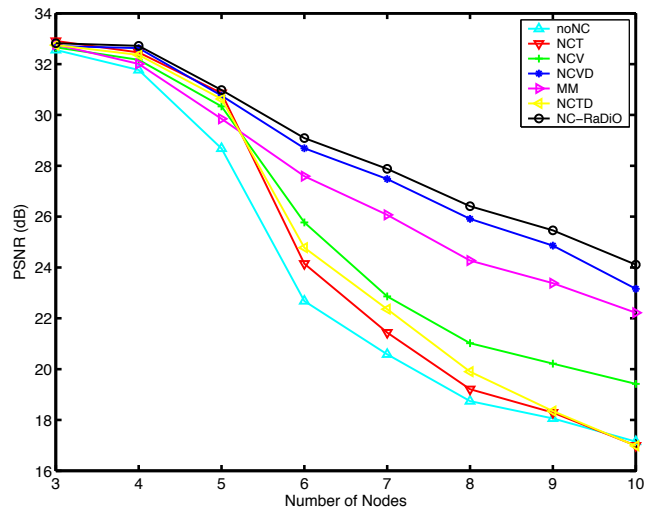


Fig. 12. *Grid topology*. PSNR values achieved by all algorithms, for a different number of streams in the system. (Channel SNR is  $5\text{dB}$ , delay budget is  $100\text{ms}$ , and data rate is  $1\text{Mbps}$ .)

### C. Complexity

The main complexity of NCV comes from considering all possible candidate codes. However, this is no worse than the complexity of NCT: they both consider all possible codes but they evaluate them using a different metric. An important observation is that real-time delay requirements significantly reduce the number of packets in the virtual buffers and therefore the complexity, making the brute-force approach feasible. For a larger delay budget, approximation algorithms for NCV and NCT can be developed by formulating them as a maximum weight independent set problem. Although this problem is NP-complete, it is also well-studied and approximation algorithms can be found in the literature [34].

NCVD runs NCV for each packet (considered as primary) in the Tx queue and selects the best overall code. The NCVD complexity is linear in the number of packets in the Tx queue, which is also small for real-time applications. The main complexity is still due to the NCV part.

Finally, the optimal solution to NC-RaDiO can be thought of as employing NCVD at all nodes with the additional ability to compare improvement values (Lagrange multipliers) among nodes in a neighborhood, in order to decide which node should transmit. This brings a small increase in complexity (on the order of  $\log(N)$  where  $N$  is the number of nodes) but can be costly in terms of network resources. In this paper, we consider NC-RaDiO mainly as a baseline for comparison with the simpler and near optimal NVC/NCVD algorithms.

## VI. CONCLUSION

In this paper, we proposed a novel approach to opportunistic video coding for video streaming over wireless networks that take into account the importance of video packets in network code selection. Essentially, our approach combines for the first time together ideas from (i) network coding for increasing throughput and (ii) prioritized transmission for improving video quality, taking into account distortion and deadlines. Simulation results show that the proposed schemes improve

video quality up to 5dB compared to baseline schemes. Furthermore, they significantly improve the application-level throughput while achieving the same or similar levels of MAC throughput. The main idea of the paper can be extended from video-aware to content-aware network coding, to combine the best of both network coding and content-based prioritization.

## REFERENCES

- [1] N. Färber and B. Girod, "Wireless Video," in A. Reibman, M.-T. Sun (eds.), *Compressed Video over Networks*, M. Dekker, 2000.
- [2] Special issue on "Advances in Wireless Video," in *IEEE Wireless Commun. Mag.*, Aug. 2005.
- [3] "The network coding webpage," <http://www.netcod.org>.
- [4] C. Fragouli, J. Widmer and J.Y. LeBoudec, "Network coding: an instant primer," in *ACM SIGCOMM CCR*, vol. 36(1), pp. 63-68, Jan. 2006.
- [5] P. A. Chou, Y. Wu, K. Jain, "Practical network coding," in *Proc. Allerton Conference*, 2003.
- [6] P. A. Chou, Y. Wu, "Network coding for the Internet and wireless networks," in *IEEE Signal Processing Mag.*, vol.24(5), pp.77-85, Sept.2007.
- [7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical network coding," in *Proc. ACM SIGCOMM*, vol. 36(4), pp. 243-254, Pisa, Italy, Sept. 2006.
- [8] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, "The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments," in *Proc. Allerton Conference*, 2005.
- [9] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," in *IEEE Trans. Multimedia*, vol. 8(2), pp. 390-404, Apr. 2006.
- [10] J. Chakareski and P. Chou, "RaDiO Edge: Rate-distortion optimized proxy-driven streaming from the network edge," in *IEEE/ACM Trans. Networking*, vol. 14, no. 6, pp. 1302-1312, Dec. 2006.
- [11] J. Chakareski and P. Frossard, "Rate-Distortion Optimized Distributed Packet Scheduling of Multiple Video streams over Shared Communication Resources," in *IEEE Trans. Multimedia*, vol. 8(2), pp. 207-218, 2006.
- [12] M. Kalman and B. Girod, "Rate-Distortion Optimized Video Streaming with Multiple Deadlines for Low Latency Applications," in *Proc. IEEE Packet Video Workshop*, Irvine, CA, Dec. 2004.
- [13] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," in *IEEE Trans. on Inf. Theory*, vol. 46(4), pp. 1204-1216, July 2000.
- [14] S.R. Li, R.W. Yeung, and N. Cai, "Linear network coding," in *IEEE Trans. Inform. Theory*, vol. 49(2), pp. 371-381, Feb. 2003.
- [15] B. Girod, J. Chakareski, M. Kalman, Y.J. Liang, E. Setton and R. Zhang, "Advances in Network-adaptive Video Streaming," in *Proc. IWDC 2002*, Capri, Italy, Sept. 2002.
- [16] S. Shankar, Z. Hu, and M. Van der Schaar, "Cross Layer Optimized Transmission of Wavelet Video over IEEE 802.11a/e WLANs," in *Proc. IEEE Packet Video 2004*, Irvine, CA, Dec. 2004.
- [17] Special issue on "Multimedia over Broadband Wireless Networks," in *IEEE Network*, vol. 19(1), pp. 25-25, Jan.-Feb 2005.
- [18] M. Van der Schaar, S. Shankar, "Cross-Layer Wireless Multimedia Transmission: Challenges, principles, and new paradigms," in *IEEE Wireless Commun. Mag.*, vol. 12(4), pp. 50-58, Aug. 2005.
- [19] E. Setton, T. Yoo, X. Zhu, A. Goldsmith and B. Girod, "Cross-layer design of ad hoc networks for real-time video streaming," in *IEEE Wireless Commun. Mag.*, vol. 12(4), pp. 59-65, Aug. 2005.
- [20] G.M. Su, Z. Han, M. Wu, and R. Liu, "Multi-user Cross-Layer Resource Allocation for Video Transmission over Wireless Networks," in *IEEE Network*, vol.20(2), pp.21-27, Mar-Apr 2006.
- [21] Z. Li, B. Li, "Network coding: The case for multiple unicast sessions," in *Proc. of Allerton Conference*, Sept. 2004.
- [22] Y. Wu, P.A. Chou, and S. Y. Kung, "Information exchange in wireless network coding and physical layer broadcast," in *Proc. IEEE CISS*, Baltimore, MD, March 2005.
- [23] T. Ho and R. Koetter, "Online incremental network coding for multiple unicasts," in *DIMACS WG on Network Coding*, Jan. 2005.
- [24] S. Deb, M. Effros, T. Ho, D.R. Karger, R. Koetter, D.S. Lun, M. Medard, and N. Ratnakar, "Network coding for wireless applications: A brief tutorial," in *Proc. IWWAN '05*, London, UK.
- [25] A. Ramamoorthy, J. Shi, and R. Wesel, "On the capacity of network coding for wireless networks," in *Proc. Allerton Conference*, 2003.
- [26] H. Seferoglu, A. Markopoulou, "Opportunistic Network Coding for Video Streaming over Wireless," in *Proc. of the 16<sup>th</sup> Packet Video Workshop*, Lausanne, Switzerland, Nov. 2007.
- [27] H. Shiang and M. van der Schaar, "Multi-user video streaming over multi-hop wireless networks: A distributed, cross-layer approach based on priority queuing," in *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 770-785, May 2007.
- [28] GloMoSim Version 2.0, "Global Mobile Information Systems Simulation Library," available at <http://pcl.cs.ucla.edu/projects/glomosim/>.
- [29] Y. Liang, J. Apostolopoulos and B. Girod, "Analysis of Packet Loss for Compressed Video: Does Burst-Length Matter?," in *Proc. of IEEE ICASSP*, vol. 5, pp. 684-687, Hong Kong, Apr. 2003.
- [30] M. Wang, M. van der Schaar, "Rate Distortion Modeling for Wavelet Video Coders," in *Proc. of IEEE ICASSP*, vol. 2, pp. 53-56, March, 2005.
- [31] A. Eichhorn, "Modeling Dependency in Multimedia Streams," in *Proc. of the 14th annual ACM international conference on multimedia*, pp. 941-950, Santa Barbara, Ca, 2006.
- [32] ITU-T Rec. "H.264 : Advanced video coding for generic audiovisual services", 2005.
- [33] H.264/AVC Reference Software Version JM 8.6 <http://liphome.hhi.de/suehring/ttml/download/>, 2006.
- [34] V. V. Vazirani, "Approximation Algorithms," Springer-Verlag Berlin Heidelberg, 2003.



**Hulya Seferoglu** (SM '04) received the B.S. degree in Electrical Engineering from Istanbul University, Turkey, in 2003 and M.S degree in Electrical Engineering and Computer Science from Sabanci University, Turkey in 2005. She is currently a Ph.D. student in the EECS Dept. at UC Irvine. Her research interests are in the areas of multimedia streaming and network coding.



**Athina Markopoulou** (SM '98, M'02) is an assistant professor in the EECS Dept. at the University of California, Irvine. She received the Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 1996, and the M.S. and Ph.D. degrees, both in Electrical Engineering, from Stanford University in 1998 and 2002, respectively. She has been a postdoctoral fellow at Sprint Labs (2003) and at Stanford University (2004-2005), and a member of the technical staff at Arastra Inc. (2005). Her

research interests include media transmission over IP networks, network measurement, network security, and applications of network coding. She received the NSF CAREER award in 2008.