



*Example 1:* Let us consider Fig. 1 which shows the cross-topology, a well-known example of one-hop opportunistic network coding [1]. Unicast flows  $S_1-D_1$ ,  $S_2-D_2$ ,  $S_3-D_3$ ,  $S_4-D_4$  go through the intermediate node  $I$ . In addition, another unicast flow  $S_5-D_5$  is transmitted from node  $I$  to node  $A_3$ . For simplicity, let us assume that the link rate from node  $I$  to each receiver is the same; *i.e.*,  $R_1=R_2=R_3=R_4=R_5=1$  packets per transmission slot (pkt/slot). Each source generates constant bit rate (CBR) traffic with rate 2 pkt/slot, which leads to congestion. Node  $I$  transmits both coded packets ( $a+b+c+d$  targeted to  $D_1, D_2, D_3, D_4$ ) and uncoded packets ( $e$  targeted to node  $D_5$ ).

First, note that UDP flows do not fully exploit the network coding opportunities due to the lack of end-to-end congestion control. When the offered load is high, as it is the case in this example, many packets are successfully transmitted on the uplink (from  $S_1, S_2, S_3, S_4$  to  $I$ ) only to then be dropped at the intermediate node  $I$ , thus wasting system resources. TCP flows would adjust their end-to-end rate to this bottleneck, but UDP flows do not. This problem occurs in both coded and uncoded systems but is more severe in coded systems, because the bandwidth wasted on the uplink could have been used to transmit more information through network coded packets.

Second, coded packets are clearly more valuable than uncoded ones, from a total throughput point-of-view, and thus should be treated preferentially during congestion. However, default queuing and MAC mechanisms at an intermediate node are agnostic to network coding. COPE uses a single queue for all packets, a first-in first-out (FIFO) policy to serve them and a Drop-Tail policy to drop them.

In the example of Fig. 1, packets from flows  $S_1, S_2, S_3, S_4$  arrive at intermediate node  $I$  with rate  $1/5$  pkt/slot. Indeed, the medium access control (MAC) protocol provides fairness among the five nodes, and the arrival rate of  $S_1, S_2, S_3, S_4$  at  $I$  is bounded by  $1/5$  pkt/slot. Packets from the uncoded flow  $S_5$  arrive at  $I$  with rate 2 pkt/slot, since  $S_5$  is located on  $I$ . Because of Drop-Tail, there are twice as many packets from flow  $S_5$  compared to  $S_1, S_2, S_3, S_4$  inserted in the queue at  $I$ . Because of FIFO, uncoded packets from flow  $S_5$  are transmitted twice more often than coded packets. When the offered load increases, less and less coded packets are transmitted thus not exploiting coding opportunities. The summary of input-output queue process for COPE scheme is depicted in Fig. 2(a).  $\square$

In this paper, we address the aforementioned problems by introducing network coding-aware mechanisms at intermediate nodes. These are minimal modifications on top of COPE. We do not introduce any change to end-to-end mechanisms. We combine network coding-aware queue management (NCAQM) with an improved version of network coding-aware MAC level packet prioritization (NCAPP), into a combined scheme, which we refer to as network coding-aware priority queueing (NCAQP). We argue that the NCAQM and NCAPP work synergistically and complement each other. In particular, NCAQM

creates network coding opportunities (which is important in the high load regime) while NCAPP drains network coded packets faster than uncoded ones (which is important in the low load regime). We perform numerical and GloMoSim simulations and show that UDP over NCAQP outperforms UDP over COPE (by a factor of 10) as well as over each individual scheme alone, NCAQM or NCAPP (by a factor of 2). Furthermore, this benefit comes without significantly hurting fairness.

The overview of the rest of the paper is as follows. Section II briefly discusses related work. Section III describes the schemes under consideration. Section IV presents simulation results. Section V concludes the paper.

## II. RELATED WORK

This paper builds on COPE, a practical scheme for one-hop network coding across unicast sessions in wireless mesh networks [1]. Researchers have tried to model and analyze COPE [8], [9], [10]. Some others proposed new coded wireless systems, based on the idea of COPE [11], [7]. In [12], the problem of rate mismatch between flows that are coded together is addressed, by delaying packets. In [3], higher MAC level packet prioritization is given to coded than to uncoded packets. In [2], we improved TCP over COPE via a network coding-aware queue management scheme (NCAQM).

A body of work has looked at the joint optimization of network coding of unicast flows in a network utility maximization (NUM) framework [13]. For example, in [14], minimum cost multicast over network coded wireline and wireless networks was studied. This work was extended for rate control in [15] for wireline networks. The rate region of multicast flows when network coding is used is studied in [16], [17]. Optimal scheduling and routing for COPE are considered in [8] and [10], respectively. Congestion control for unicast flows over wireless is considered in [2]. End-to-end pairwise inter-session network coding is studied in [18]. Energy efficient opportunistic inter-session network coding over wireless are proposed in [19], following a node-based NUM formulation and its solution based on back-pressure. A linear optimization framework for packing butterflies is proposed in [6]. Scheduling over coded wireless networks is considered, and the computational complexity of scheduling and network coding is discussed in [20], [21], [22]

This paper combines the NCAQM and NCAPP mechanisms proposed independently in [2] and [3], respectively. We first show that these mechanisms are complementary: NCAQM creates coding opportunities (which is necessary in high load regimes) while NCAPP drains coded faster than uncoded packets (which is important in low load regimes). We then propose and evaluate a combined scheme.

## III. NETWORK CODING-AWARE QUEUEING SCHEMES

### A. Network Coding-Aware Queue Management: NCAQM

NCAQM [2] is designed to better exploit network coding opportunities for unicast flows over coded wireless networks,

using constructive inter-session coding schemes, including single-hop (*e.g.*, COPE [1]) and multi-hop schemes (*e.g.*, BFLY [7]). In this paper, we focus on the benefit of NCAQM for UDP flows over COPE-based networks. In summary, NCAQM introduces the following modifications to the queue-management scheme at the output queue: (i) it stores coded packets instead of uncoded packets; (ii) upon congestion, packets are dropped from the flow that has the largest number of packets in the queue. Next, we explain NCAQM in more detail.

**Maintaining Queues and Network Coding:** A wireless node  $i$  maintains a single physical output queue,  $Q_i$ , which stores all packets (coded and uncoded depending on the opportunities) passing through it. Packets are coded when they are inserted to an output queue. If a network coding opportunity does not exist when the packet arrives at node  $i$ , we just store it in  $Q_i$ . Periodically, all packets in the queue are checked for network coding. Let  $Q_i = \{p_1, p_2, \dots, p_l\}$  where  $p_1$  is the first and  $p_l$  is the last packet in the queue;  $l \leq L$ , where  $L$  is the buffer size, *i.e.*, the maximum number of packets that can be stored in  $Q_i$ . First,  $p_1$  is picked for network coding. Since  $Q_i$  stores network coded packets,  $p_1$  may be already coded. Independently of whether  $p_1$  is network coded or not, it can be further coded with other packets in the queue beginning from  $p_2$ , if the following two conditions are satisfied; (i) the packets constructing  $p_1$  and  $p_2$  should be from different flows, and (ii)  $p_1 \oplus p_2$  should be decodable at the next hop of all packets that construct the network code. If these conditions are satisfied, we say that the network code is an eligible network code, and  $p_1$  is replaced by  $p_1 \oplus p_2$ . Then  $p_1 \oplus p_3$  is checked for network coding, etc. After all packets are checked for network coding, the output queue  $Q_i$  is updated: (i) the final packet  $p_1$  is stored in the first slot of the output queue, and (ii) the memory allocated to other packets (which are used in network coding  $p_1$ ) are freed. Then, the same algorithm is run for packet  $p_2$ , etc. When a transmission opportunity arises, the first packet from the output queue is checked for network coding again and transmitted.

**Virtual Queues and Packet Dropping:** Upon congestion, a node decides which packet to drop. To achieve this, the node compares the number of all (coded and uncoded) packets of each flow, in virtual queues (which is explained next) where the flow has more packets than the other flows in that queue.

We first maintain virtual queues (also called  $h$ -queues) for each hyperarc, which keeps track of packets that are network coded and broadcast over  $h^1$ . The size of an  $h$ -queue is  $Q_h$ .

Second, for each flow  $s$ , we calculate  $\Phi_i^s = \sum_{h(\mathcal{J})|h \in \mathcal{A}} Q_h \tilde{w}_h^s$ , where  $\tilde{w}_h^s$  is the indicator to show whether flow  $s$  has more packets than the other flows in queue  $Q_h$ . Upon congestion, the  $\Phi_i^s$ 's are compared and

<sup>1</sup>A hyperarc  $(i, \mathcal{J})$  is a collection of links from node  $i \in \mathcal{N}$  to a non-empty set of next-hop nodes  $\mathcal{J} \subseteq \mathcal{N}$ . A hypergraph  $\mathcal{H} = (\mathcal{N}, \mathcal{A})$  represents a wireless mesh network, where  $\mathcal{N}$  is the set of nodes and  $\mathcal{A}$  is the set of hyperarcs. For simplicity,  $h = (i, \mathcal{J})$  denotes a hyperarc,  $h(i)$  denotes node  $i$  and  $h(\mathcal{J})$  denotes node  $\mathcal{J}$ , *i.e.*,  $h(i) = i$  and  $h(\mathcal{J}) = \mathcal{J}$ .

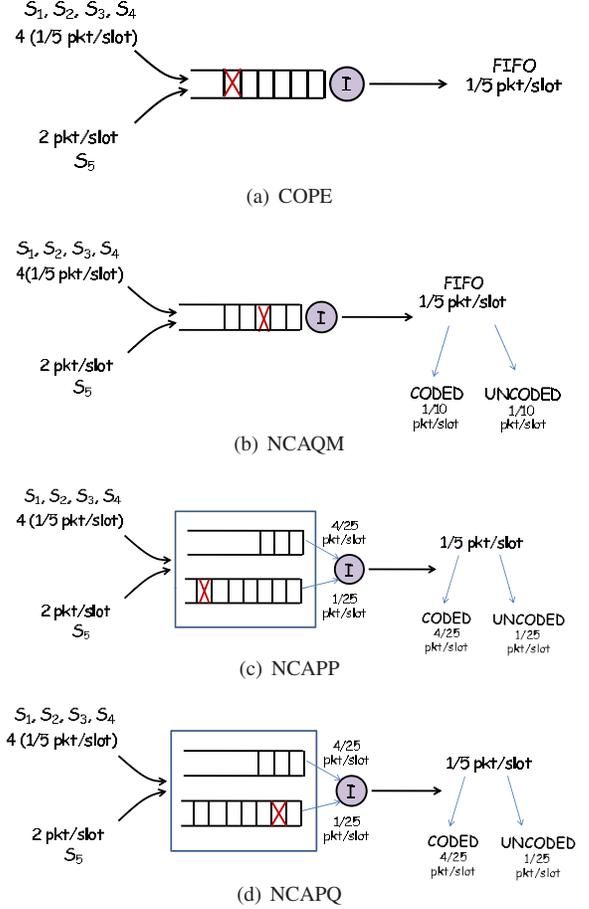


Fig. 2. Operations at the head/tail of the queue(s) in the example topology of Fig. 1 where “X” shows a packet that may be dropped upon congestion for each scheme. (a) COPE. Uses a FIFO policy to serve packets and a Drop-Tail policy to drop them. As a result,  $I$  transmits packets from  $S_5$  most of the time. (b) NCAQM. Upon congestion, a packet from the largest queue is dropped (mostly from  $S_5$ ). This equalizes the number of coded and uncoded packets in the queue. Thus, the number of network coding opportunities increases. Packets are served according to FIFO which limits the achievable throughput. (c) NCAPP. Two queues for “coded” and “uncoded” packets are constructed. Coded and uncoded packets are served with rates  $4/25$  and  $1/25$  pkt/slot, respectively. A Drop-Tail policy is used to drop packets, so  $S_5$  floods the queue. Thus, network coding opportunities are limited. (d) NCAQP. Upon congestion, a packet from the largest queue is dropped (mostly from  $S_5$ ). This equalizes the number of coded and uncoded packets in the queue. Thus, the number of network coding opportunities increases as in NCAQM. Network coded packets are prioritized as in NCAPP. Different from NCAPP, since NCAQM employed, the number of packets in the queue is equalized and there are more network coding opportunities.

a packet from the flow with the largest  $\Phi_i^s$  is dropped, preferably the last uncoded packet. If all packets from the selected flow are coded, a new coming packet(s) is dropped instead. Next, we demonstrate the benefit of NCAQM through an example.

*Example 2:* Let us consider again the cross topology in Fig. 1 and Example 1. Packets from flows  $S_1, S_2, S_3, S_4$  arrive to  $I$  with rate  $1/5$  pkt/slot and packets from flow  $S_5$  arrives with rate  $2$  pkt/slot. In NCAQM, congestion is handled according to the rule described above; *i.e.*, upon congestion,

a packet from the largest queue is dropped, and packets are transmitted according to FIFO. Since packets from  $S_5$  arrives to the queue at a higher rate, they form the largest queues most of the time, thus the probability of dropping a packet from flow  $S_5$  is higher when there is congestion. As a result, there exists an equal number of coded and uncoded packets in the queue with high probability, and  $I$  transmits as many coded packet as uncoded ones (*i.e.*, the transmission rate for coded and uncoded packets are 1/10 pkt/slot). This eliminates the disadvantage of COPE in which uncoded packets are transmitted twice more often than uncoded packets and NCAQM improves over COPE. The input-output queue process for NCAQM is summarized in Fig. 2(b).  $\square$

### B. Network Coding-Aware Packet Prioritization: NCAPP

In [3], the key intuition is that the fairness imposed by 802.11 MAC among competing flows and nodes affects network coding. To increase the total throughput, one should give higher priority to network coded flows than to uncoded flows. Transmission of network coded packets drain queues at bottleneck nodes at a faster rate, thus bottleneck nodes can handle traffic at higher rates, leading to higher throughput. Next, we explain NCAPP in more detail.

**Packet Prioritization:** Let  $K$  be the number of flows (coded or uncoded) transmitted from a node  $i$ . Each packet transmission from flow  $k$  carries  $\beta_k$  packets, *i.e.*, if flow  $k$  is not network coded, then  $\beta_k = 1$ , if flow  $k$  is a network coded flow which combines two flows, then  $\beta_k = 2$ . When a node gets a transmission opportunity, it gives  $\beta_k$  more transmission opportunity to the packets from flow  $k$ . Next, we explain the benefit of NCAPP through an example.

*Example 3:* Let us consider again the cross topology in Fig. 1. Packets from flows  $S_1, S_2, S_3, S_4$  arrive to  $I$  with rate 1/5 pkt/slot and packets from flow  $S_5$  arrives with rate 2 pkt/slot. In NCAPP, congestion is handled according to Drop-Tail. However, packet prioritization, which is used in conjunction with the MAC layer, instead of FIFO gives more transmission opportunities to coded packets. In this example, the intermediate node has two flows: “coded” ( $S_1, S_2, S_3, S_4$ ) and “uncoded” ( $S_5$ ).  $\beta_{coded} = 4$  and  $\beta_{uncoded} = 1$ . Thus, coded packets are transmitted 4 times faster than uncoded flows. As a result, coded flows are served with rate 4/25 pkt/slot and uncoded packets are served with rate 1/25 pkt/slot. This drains the bottleneck node faster and increases the total throughput. The input-output queue process for NCAPP is summarized in Fig. 2(c).  $\square$

### C. Network Coding-Aware Priority Queueing: NCAPO

In this paper, we combine NCAQM and an extended version of NCAPP, into a new scheme, which we refer to as Network Coding-Aware Priority Queueing or NCAPO. First, let us describe how to extend NCAPP to assign priorities by taking into account information not only about network coding, but also about link rates as explained next.

**Packet Prioritization:** Similarly to NCAPP, let  $K$  be the number of different flows (coded or uncoded) transmitted from a node. Each flow is associated with a hyperarc  $h$ . (Note that more than one flow can be associated to the same hyperarc.) The transmission capacity of a hyperarc  $h$  is  $R_h$ . Since  $h$  is a set of links,  $R_h$  is the minimum:  $R_h = \min_{j \in h(\mathcal{J})} \{R_{i,j} \xi_{i,j}\}$  where  $R_{i,j}$  is the capacity of link  $(i,j)$ , and  $\xi_{i,j}$  is the probability of successful transmission over link  $(i,j)$ .  $\xi_{i,j}$  is calculated as the ratio of successfully delivered packets to node  $j$  over all the packets that are transmitted from node  $i$  in a time window. The drain rate of the  $k$ -th flow ( $R_k$ ) is equal to the capacity of hyperarc ( $h$ ) which is associated with this flow, *i.e.*,  $R_k = R_h$ . Each packet transmission from flow  $k$  carries  $\beta_k$  packets.

When a node gets a transmission opportunity, it transmits a packet from flow  $k$  with probability  $p_k = \frac{R_k \beta_k}{\sum_{k=1}^K R_k \beta_k}$ . Let us assume that all transmission rates are equal,  $R_{coded} = 1$  and  $R_{uncoded} = 1$ . Since four packets can be combined together<sup>2</sup>, it is  $\beta_{coded} = 4$  and  $\beta_{uncoded} = 1$ . In this static scenario,  $p_{coded} = 4/5$  and  $p_{uncoded} = 1/5$ , and coded packets are transmitted four times more than uncoded packets, similarly to NCAPP. More generally, when link rates change, a node can adjust these probabilities accordingly. The benefit of incorporating link rates will be demonstrated through simulations in Section IV.

**The synergy between NCAQM and NCAPP:** Let us further discuss how NCAQM and NCAPP complement each other.

On one hand, NCAQM improves COPE with network coding-aware queue management. Upon congestion, NCAQM drops a packet in a longest-queue-first manner so as to equalize the number of packets from flows that are coded together. This increases the network coding opportunities significantly. However, packets are served according to FIFO, which gives an equal share to coded and uncoded flows, although network coded packets should get higher transmission priority in order to increase throughput. NCAPP provides this queueing mechanism which is complementary to NCAQM.

On the other hand, NCAPP provides higher MAC priority and thus more transmission opportunities to coded packets. However, if some flows (*e.g.*, uncoded flow  $S_5$ ) transmits at higher rate and if congestion is handled according to Drop-Tail, their packets flood the queue. Therefore, there may not always exist packets from coded flows (*e.g.*,  $S_1, S_2, S_3, S_4$ ), which reduces the network coding opportunities and eventually throughput. However, NCAQM equalizes the number of packets from all flows in the queue. That is, when NCAQM is employed, there always exist packets (*e.g.*, from  $S_1, S_2, S_3, S_4$ ) to do network coding with high probability. In that sense, NCAQM complements NCAPP.

Since NCAQM and NCAPP complement each other, we

<sup>2</sup>Note that it is also possible to combine two or three of ( $S_1, S_2, S_3, S_4$ ). We consider this general case in our simulations, but we discuss the simple case of combining all four flows into one, for the sake of example.

propose a novel scheme NCA PQ which combines both NCAQM and NCA PP (in fact, its extended version that assigns priorities based on both network coding and link rates). The following example illustrates the benefit of the combined scheme.

*Example 4:* In the combined scheme, packets from flows  $S_1, S_2, S_3, S_4$  arrive to  $I$  with rate  $1/5$  pkt/slot and packets from flow  $S_5$  arrives with rate  $2$  pkt/slot. Congestion is handled according to NCAQM: a packet from the largest flow is dropped. Packets are served according to NCA PP: coded packets are given more transmission opportunities. In this setting, coded flows are served with rate  $4/25$  pkt/slot and uncoded packets are served with rate  $1/25$  pkt/slot. Since queue sizes are equalized according to NCAQM, it is more likely to have packets from flows  $S_1, S_2, S_3, S_4$  in the queue at  $I$  compared to NCA PP. The input-output queue process for NCA PQ is summarized in Fig. 2(d).  $\square$

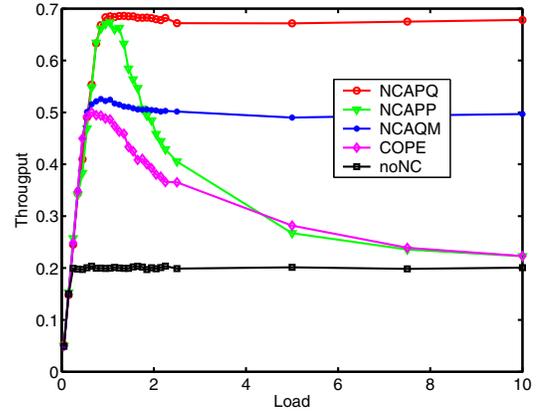
#### IV. PERFORMANCE EVALUATION

In this section, we evaluate the throughput and fairness of UDP over our proposed scheme NCA PQ as well as NCAQM [2] and NCA PP [3] in various topologies and traffic scenarios. We compare it to UDP over the following baseline schemes: no network coding (noNC), which uses FIFO without network coding; COPE [1], which stores packets in a FIFO queue, and decides which packets to code at each transmission.

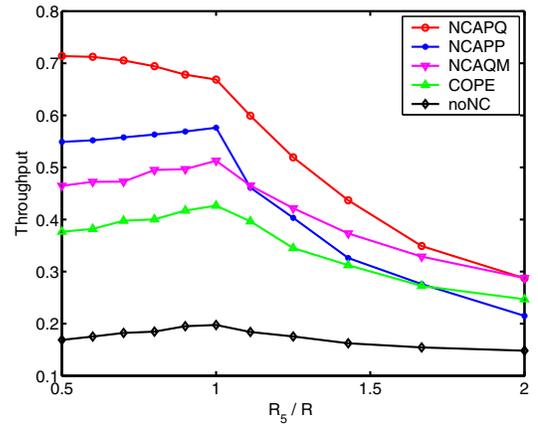
##### A. Numerical Simulations

1) *Setup:* Let us consider the scenario shown in Fig. 1. Channel capacities are 1 packet/transmission, and buffer sizes are 100 packets. We consider a slotted system where each node accesses medium at each slot (transmission) randomly according to equal probability. (We also simulate 802.11 MAC using GloMoSim later in this section.) As a wireless channel model, we consider iid packet erasure channel. We consider that each source injects packets to the network according to Poisson distribution. We report the average throughput as well as fairness. The throughput results are averaged over 10000 packet transmission times first, then 10 results with different seeds.

2) *Throughput Results:* Fig. 3 presents results for the cross topology shown in Fig. 1. Fig. 3(a) shows throughput versus offered load (total load offered by all flows  $S_1, S_2, S_3, S_4$ ) graph. It is seen that all coded schemes perform better than noNC. COPE performs better than noNC thanks to network coding. However, when the offered load increases, the throughput of COPE decreases. The reason is that at high loads,  $S_5$  floods the queue with its packets and most of the time there are no packets from  $S_1, S_2, S_3, S_4$  in  $I$ 's queue to perform network coding. NCA PP improves over COPE, because it gives more transmission opportunities to network coded packets than uncoded packets. Thus, NCA PP better exploits network coding opportunities. However, when the offered load increases, its improvement reduces for the same reason as COPE. NCAQM performs worse than NCA PP in



(a) Throughput vs. Load



(b) Throughput vs. Rate Ratio

Fig. 3. Average throughput (averaged in 10000 packet transmissions first, then over 10 seeds) vs. (a) offered load, and (b) rate ratio for the cross-topology shown in Fig. 1. In (a), link rates are set to 1 packet/transmission, *i.e.*,  $R_1=R_2=R_3=R_4=R_5=1$ . In (b), the rate ratio is defined as the ratio of  $R_5$  over  $R$ , where  $R_1=R_2=R_3=R_4=R$ . The total offered load is set to 1.5 packet/transmission.

the low load regime and better in the high load regime. In the low load regime, NCAQM transmits coded and uncoded packets at similar rates (*i.e.*, it does not give priority to network coded packets), therefore, its performance is similar to COPE in this regime. When offered load increases, it performs better than NCA PP, because it can still do network coding (since it avoids flooding introduced by  $S_5$ ). NCA PQ achieves higher throughput than all other schemes thanks to using the advantages of both NCAQM and NCA PP.

Fig. 3(b) shows throughput versus rate ratio graph. We consider that  $R_1=R_2=R_3=R_4=R$ , and we define the rate ratio as the ratio of  $R_5$  over  $R$ . As expected, noNC performs worse than the other schemes. Network coding schemes improve over noNC in all rate levels. However, if the rate ratio increases beyond 1, the performance of all network coding schemes reduces. The reason is that when  $R_5$  is larger than  $R$ , it is more beneficial to transmit more packets from  $S_5$ , because packets from  $S_5$  can be transmitted faster. Thus, transmitting

coded packets becomes less and less beneficial in terms of throughput, and the improvement of all network coding schemes reduces. The decrease in improvement is more drastic on NCAPP's performance which even becomes worse than COPE when  $R_5$  is larger than  $R$ . The reason is that NCAPP gives more priority to coded flows whose transmission is not much beneficial in this scenario. On the other hand, NCAPQ is better than all other schemes at all rate ratios, because it arranges the priority level according to link rates as well as the number of packets that are coded together as presented in Section III-C.

3) *Fairness Results:* Now, we consider how packet prioritization employed by NCAPP and NCAPQ affects throughput and fairness for the cross topology shown in Fig. 1. In these simulations, packet prioritization value ( $p_{\text{coded}}/p_{\text{uncoded}}$  for NCAPQ) is hard coded and ranges from 1 to 10. The link rates are  $R_1=R_2=R_3=R_4=R_5=1$  and the total offered load is 1.5 packet/transmission.

Fig. 4(a) shows the throughput vs. packet prioritization. When coded packets are given more priority, the throughput increases. This is expected, because when link rates are equal, it is better to transmit more coded packets than uncoded ones for throughput improvement. Next, we report the effect of priority to fairness. Fig. 4(b) shows the throughput vs. fairness for the exact same scenarios as in Fig. 4(a). Fairness changes according to priority level ( $p_{\text{coded}}/p_{\text{uncoded}}$  for NCAPQ). We use Jain's fairness index [23]:

$$J_f = \frac{(\sum_{s \in \mathcal{S}} x_s)^2}{|\mathcal{S}|(\sum_{s \in \mathcal{S}} x_s^2)},$$

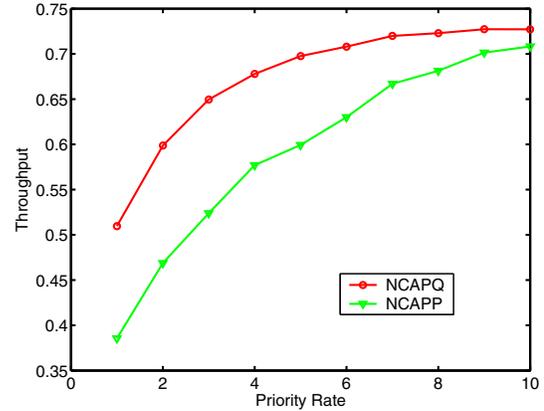
where  $\mathcal{S}$  is the set of flows in the network such that  $s \in \mathcal{S}$  and  $x_s$  is the flow rate associated to flow  $s$ . One can see that when throughput increases, fairness decreases, but not significantly. When the priority rate is set to 10 the fairness index is at its lowest value (0.84); *i.e.*, a reasonably fair rate allocation even at such a low level of fairness. Actually, NCAPP and our scheme NCAPQ sets priority ratio to 4 in this scenario which corresponds to 0.9 fairness index which is significantly high.

### B. GloMoSim Simulations

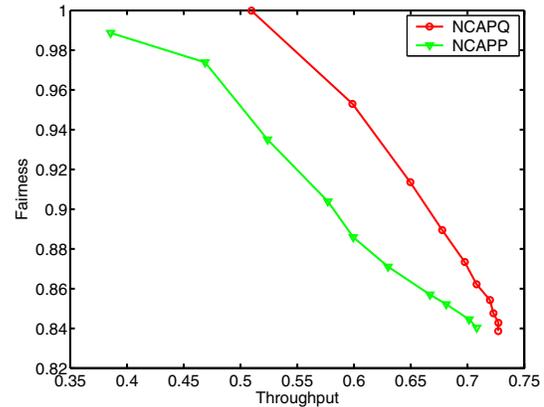
1) *Setup:* Finally, we evaluate the throughput of UDP flows over the schemes under consideration (NCAQM, NCAPP, NCAPQ) as well as no network coding scheme (noNC), and COPE using the GloMoSim simulator [24], which is well suited for wireless environments.

We simulated three illustrative topologies shown in Fig. 1, and Fig. 5. In the cross and Alice&Bob topologies, an intermediate node  $I$  is placed in the center of a circle with 90m radius over  $200m \times 200m$  terrain and all other nodes are placed around the circle. In the grid topology, nodes are distributed over a  $300m \times 300m$  terrain, divided into 9 cells of equal size. 15 nodes are divided into sets consisting of 1 or 2 nodes and each set is assigned to a different cell.

Channel capacities are set to 1Mbps, buffer size to 50 packets, and packet size to 500B. In the MAC, we simulated IEEE



(a) Throughput vs. Priority Rate

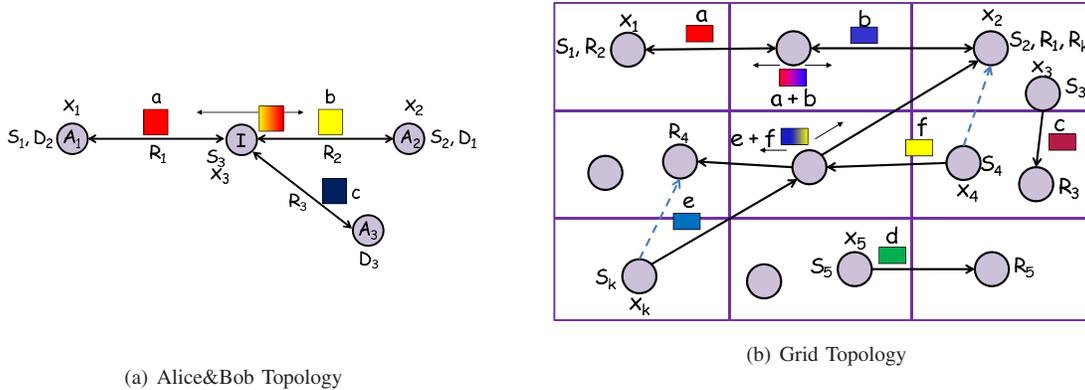


(b) Throughput vs. Fairness

Fig. 4. Average throughput (averaged in 10000 packet transmissions first, then over 10 seeds) vs. (a) priority rate ( $p_{\text{coded}}/p_{\text{uncoded}}$  for NCAPQ), and (b) fairness. The topology under consideration is the cross-topology shown in Fig. 1. The link rates are set to  $R_1=R_2=R_3=R_4=R_5=1$  packet/transmission, and the offered load is 1.5 packet/transmission.

802.11 with RTS/CTS enabled and with the modifications (pseudo broadcasting and overhearing) for network coding. As a wireless channel model, we used the two-ray path loss model and Rayleigh fading in GloMoSim. The average loss rate is 5%. We consider CBR/UDP traffic. Each flow starts a random time within the first 5sec and live until the end of the simulation. In the grid topology, UDP flows arrive according to a Poisson distribution with average 6 flows per 30sec. We report the average (transport level) throughput of each scheme in *kbps*. The throughput results are averaged over 1min simulation duration first, then over 10 simulations with different seeds.

2) *Results:* Fig. 6(a) shows the total achieved throughput vs. the total offered load for the Alice&Bob topology shown in Fig. 5(a). The performance of noNC is linearly increasing with increasing load (as expected) and decreases after some threshold load level. We see such behavior (*i.e.*, decreasing throughput with increasing load level) for all other schemes. This is different from what we observe through numerical simulations (in Fig. 3(a)) especially for noNC, NCAQM,



(a) Alice&Bob Topology

(b) Grid Topology

Fig. 5. (a) Alice&Bob topology. Unicast flows  $S_1-D_1$  and  $S_2-D_2$  meet at  $I$ . Another unicast flow  $S_3-D_3$  is transmitted from node  $I$  to  $A_3$ . Intermediate node  $I$  can broadcast network coded packet  $a+b$  in a single transmission to receivers  $D_1, D_2$  which can decode  $a+b$ , because they already know the packet they have transmitted. Packet  $c$  cannot be coded, because it is generated from  $I$ , and it cannot be known (overheard) by other receivers. (b) Grid topology. UDP flows arrive according to a Poisson distribution with average 6 flows per 30sec. The sender and the receiver of a UDP flow are chosen randomly. If the same node is chosen, the random selection is repeated. If both sender and receiver are in the same cell or in neighboring cells, there is a direct transmission; otherwise, a node in a neighboring cell acts as a relay. If there are more than one neighboring cells, one is chosen at random. If a network coding opportunity arises, intermediate nodes combine packets and transmit.

and NCAPQ. This behavior in the GloMoSim simulations is due to the underlying medium access control (MAC) scheme (CSMA/CA of 802.11), which was abstracted away in the numerical simulations. Sources  $S_1$  and  $S_2$  transmit their packets to  $I$ . When the load is high, some of these packets are dropped at  $I$ , MAC level ACKs are not sent back to  $S_1$  and  $S_2$ , and the sources ( $S_1$  and  $S_2$ ) increase their contention windows. This makes them to access medium less frequently than the intermediate node, and eventually the number of packets transmitted by  $S_1$  and  $S_2$  to  $I$  reduces, which reduces overall throughput.

Other than this behavior, we see similar trends in this graph as in Fig. 3(a). Specifically, COPE performs better than noNC due to network coding. However, when the offered load increases, the throughput of COPE decreases due to flooding by  $S_5$ . NCAPP improves over COPE, because it gives more transmission opportunities to network coded packets. NCAQM performs worse than NCAPP in the low load regime and better in the high load regime for the same reasons explained in the numerical simulations. NCAPQ achieves higher throughput than all other schemes thanks to using the advantages of both NCAQM and NCAPP. Specifically, at 1.2Mbps offered load, NCAPQ improves the total throughput by 40% over noNC. At the same load, NCAPP, NCAQM, and COPE improve up to 20%, 20%, and 10% over noNC, respectively. In summary, the improvement from each of NCAPP and NCAQM is twice the improvement of COPE. NCAPQ further doubles the improvement, compared to NCAPP and NCAQM alone.

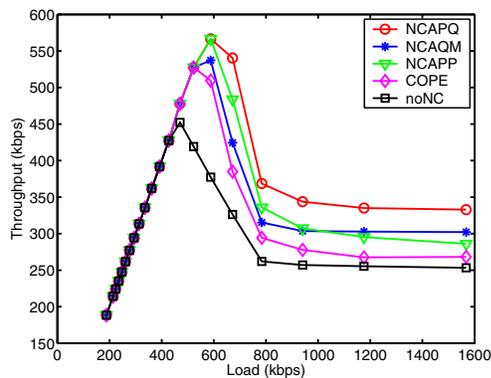
Fig. 6(b) shows the throughput vs. the total offered load graph for the cross topology shown in Fig. 1. Network coding schemes improve throughput more in the cross topology as compared their counterparts in Alice&Bob topology, because network coding in cross topology naturally brings more ben-

efit. In particular, four packets can be network coded in cross topology, while two packets can be network coded in Alice&Bob topology. NCAPQ improves throughput up to 200% as compared to noNC when the total offered load is 800kbps. At the same load, NCAPP and NCAQM improve throughput up to 100% as compared to noNC, while the improvement from COPE is around 20%.

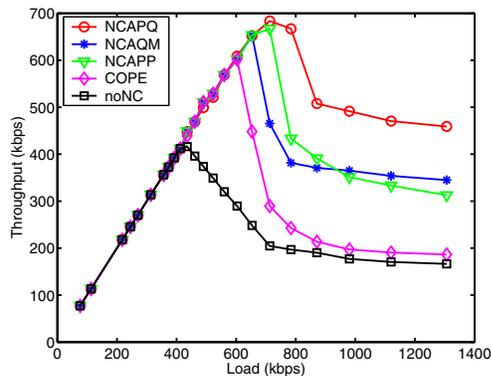
Finally, Fig. 6(c) shows the total throughput vs. the total offered load for the grid topology shown in Fig. 5(b). We observe that the improvement of network coding in the grid topology is higher than in the Alice&Bob topology but less than in the cross topology. The reason is that in the grid topology, there might be different network coding structures that combine two, three, four, or more packets. In the total throughput, we see the average throughput of these network coding structures. In particular, when the total offered load is 940kbps, the improvement from NCAPQ is around 50%, the improvement from NCAQM or NCAPP alone is around 30%, and the improvement from COPE is around 12%.

## V. CONCLUSION

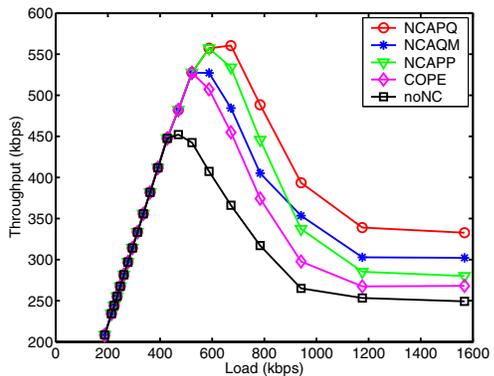
In this paper, we compared and combined network coding-aware queue management (NCAQM) and priority (NCAPP) schemes to improve the performance of UDP flows over coded wireless networks with inter-session network coding such as COPE. Simulation results show that the combined scheme (NCAPQ) improves throughput significantly, not only compared to COPE, but also compared to each individual scheme alone. Furthermore, the increase in throughput comes without significant loss in fairness. Finally, the proposed network coding-aware mechanisms are minimal and natural modifications on top of COPE, while end-to-end transport



(a) Alice&Bob Topology



(b) Cross Topology



(c) Grid Topology

Fig. 6. Average throughput (averaged in 1min simulation first, then over 10 seeds) versus offered load for (a) Alice&Bob topology (shown in Fig. 5(a)), (b) cross topology (shown in Fig. 1), and (c) grid topology (shown in Fig. 5(b)). Packet size is 500B, channel capacity is 1Mbps, and buffer size is 50 packets.

and multiple access protocols remain intact, which makes our scheme well suited for practical deployment.

#### ACKNOWLEDGEMENT

This work has been supported by an AFOSR MURI (prime award FA9550-09-0643) and by the NSF CAREER award 0747110.

#### REFERENCES

- [1] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *IEEE/ACM Trans. on Networking*, vol. 16(3), pp. 497-510, June 2008.
- [2] H. Seferoglu and A. Markopoulou, "Network coding-aware queue management for unicast flows over coded wireless networks," in *Proc. of NetCod*, Toronto, Canada, June 2010.
- [3] F. Zhao and M. Médard, "On analyzing and improving COPE performance," in *Proc. of ITA*, San Diego, CA, Feb. 2010.
- [4] Y. Wu, P. A. Chou, and S. Y. Kung, "Information exchange in wireless network coding and physical layer broadcast," in *Proc. of CISS*, Baltimore, MD, March 2005.
- [5] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," in *Proc. of ITW*, Punta del Este, Uruguay, March 2006.
- [6] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Médard, "Network coding for multiple unicasts: an approach based on linear optimization," in *Proc. of ISIT*, Seattle, WA, July 2006.
- [7] S. Omiwade, R. Zheng, and C. Hua, "Butterflies in the mesh: lightweight localized wireless network coding," in *Proc. of NetCod*, Hong Kong, Jan. 2008.
- [8] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. of ACM Mobicom*, Montreal, Canada, Sep. 2007.
- [9] J. Le, J. Lui, and D. M. Chiu, "How many packets can we encode? - an analysis of practical wireless network coding," in *Proc. of Infocom*, Phoenix, AZ, April 2008.
- [10] S. Sengupta, S. Rayanchu, and S. Banarjee, "An analysis of wireless network coding for unicast sessions: the case for coding-aware routing," in *Proc. of Infocom*, Anchorage, AK, May 2007.
- [11] Q. Dong, J. Wu, W. Hu, and J. Crowcroft, "Practical network coding in wireless networks," in *Proc. of Mobicom*, Montreal, Canada, Sept. 2007.
- [12] Y. Huang, M. Ghaderi, D. Towsley, and W. Gong, "TCP performance in coded wireless mesh networks," in *Proc. of SECON*, San Francisco, CA, June 2008.
- [13] M. Chiang, S. T. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: a mathematical theory of network architectures," in *Proceedings of the IEEE*, vol. 95(1), pp. 255-312, Jan. 2007.
- [14] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," in *IEEE Trans. on Internet Technology*, vol. 52(6), June 2006.
- [15] L. Chen, T. Ho, S. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proc. of Infocom*, Anchorage, AK, 2007.
- [16] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multi-hop multicast in wireless mesh networks," in *IEEE JSAC*, vol. 24(11), Nov. 2006.
- [17] Z. Li, B. Li, and M. Wang, "Optimization models for streaming in multihop wireless networks," in *Proc. of ICCCN*, Honolulu, HI, Aug. 2007.
- [18] A. Khreishah, C. C. Wang, and N. B. Shroff, "Cross-layer optimization for wireless multihop networks with pairwise intersession network coding," in *IEEE JSAC*, vol. 27(5), June 2009.
- [19] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *Proc. of Infocom*, Phoenix, AZ, April 2008.
- [20] D. Traskov, D. S. Lun, R. Koetter, and M. Médard, "Network coding in wireless networks with random access," in *Proc. of ISIT*, Nice, France, June 2007.
- [21] D. Traskov, M. Médard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. of Globecom*, Honolulu, HI, Dec. 2009.
- [22] D. Traskov, M. Heindlmaier, M. Médard, R. Koetter, and D. S. Lun, "Scheduling for network coded multicast: a conflict graph formulation," in *Proc. of Globecom*, New Orleans, LO, Dec. 2008.
- [23] R. K. Jain, "The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling," John Wiley & Sons, April 1991.
- [24] GloMoSim Version 2.0, "Global mobile information systems simulation library," available at <http://pcl.cs.ucla.edu/projects/glomosim/>.