

UC Irvine

UC Irvine Previously Published Works

Title

Intra- and Inter-Session Network Coding in Wireless Networks

Permalink

<https://escholarship.org/uc/item/69c2326j>

Authors

Seferoglu, Hulya
Markopoulou, Athina
Ramakrishnan, KK

Publication Date

2010-08-30

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

Intra- and Inter-Session Network Coding in Wireless Networks

Hulya Seferoglu, *Member, IEEE*, Athina Markopoulou, *Member, IEEE*, K. K. Ramakrishnan, *Fellow, IEEE*

Abstract—In this paper, we are interested in improving the performance of constructive network coding schemes in lossy wireless environments. We propose I²NC - a cross-layer approach that combines inter-session and intra-session network coding and has two strengths. First, the error-correcting capabilities of intra-session network coding make our scheme resilient to loss. Second, redundancy allows intermediate nodes to operate without knowledge of the decoding buffers of their neighbors. Based only on the knowledge of the loss rates on the direct and overhearing links, intermediate nodes can make decisions for both intra-session (*i.e.*, how much redundancy to add in each flow) and inter-session (*i.e.*, what percentage of flows to code together) coding. Our approach is grounded on a network utility maximization (NUM) formulation of the problem. We propose two practical schemes, I²NC-state and I²NC-stateless, which mimic the structure of the NUM optimal solution. We also address the interaction of our approach with the transport layer. We demonstrate the benefits of our schemes through simulations.

Index Terms—Network coding, wireless networks, error correction, cross-layer optimization.

I. INTRODUCTION

Wireless environments lend themselves naturally to network coding (NC), thanks to their inherent broadcast and overhearing capabilities. In this paper, we are interested in wireless mesh networks used for carrying traffic from unicast sessions, which is the dominant traffic today. Network coding has been used as a way to improve throughput over such wireless environments. Given that optimal inter-session NC for unicast is still an open problem, constructive approaches are used in practice [1], [2], [3], [4], [5]. One of the first practical wireless NC systems is COPE [2] - a coding shim between the IP and MAC layers that performs one-hop, opportunistic NC. COPE codes packets from different unicast sessions, and relies on receivers being able to decode these using overheard packets. This way, COPE combines multiple packets by using information on overheard packets which are exchanged through transmission reports and effectively forwards multiple packets in a single transmission to improve throughput. In order for COPE to work in a multihop network, nodes must cooperate to (i) exchange information about what packets they have

overheard and also (ii) code so that all one-hop downstream nodes can decode. This must be done at every hop across the path of a flow and cross-layer optimization approaches can be used [6] to further boost the performance.

One important problem that remains open, and is the focus of this paper, is COPE's performance in the presence of non-negligible loss rates. The reason is that intermediate nodes in COPE require the knowledge of what their neighbors have overheard, in order to perform one-hop inter-session NC. However, in the presence of medium-high loss rate, although each node fully cooperates to report what it has overheard, this information is limited, possibly corrupted, and/or delayed over lossy wireless channels. COPE turns off NC if loss rate exceeds a threshold with default value 20% [2]. However, this does not take full advantage of all the available NC opportunities. To better illustrate this key point, let us discuss the following example.

Example 1: Let us consider Fig. 1, and focus on the neighborhood of node I , *i.e.*, only the packets transmitted via I , from A_1 to A_2 and from B_1 to B_2 . This forms an "X" topology which is a well-known, canonical example of one-hop opportunistic NC [1], [2]. In the absence of loss, throughput is improved by 33.3%, because I delivers two packets in three transmissions (with NC), instead of four (without NC). Let us re-visit this example when there is packet loss. Assume that there is loss only on the overhearing link $A_1 - B_2$, with probability $\rho_{\{A_1, B_2\}} = 0.3$, and all other links have no loss. In this case, 70% of the packets can still be coded together, and throughput can be improved by 26%, which is still a significant improvement. Even at higher loss rate, *e.g.*, $\rho_{\{A_1, B_2\}} = 0.5$, inter-session coding still improves throughput up to 20%. This is under the assumption that I knows the exact state of A_2, B_2 , *i.e.*, what packets were overheard, and thus I is able to decide what packets to code together so as to guarantee decodability at the receivers. However, at high loss rates, cooperation among nodes becomes difficult. This is why COPE turns off the coding functionality when loss rate is higher than a threshold with default value 20%, thus not taking full advantage of all coding opportunities. ■

We propose a solution to this problem with a design which combines intra- and inter-session NC over wireless mesh networks. We use intra-session NC to combine packets within the same flow and introduce parity packets to protect against loss. Then, we use inter-session NC to combine packets from different (already intra-session coded) flows, and thus increase throughput. Our approach for combining intra-session with inter-session NC, which we refer to as I²NC, has two key benefits. First, it can correct packet loss and still perform inter-

H. Seferoglu is with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology. Email: hseferog@mit.edu. Mail: 77 Massachusetts Avenue, Room 32-D671, Cambridge, MA 02139.

A. Markopoulou is with the Electrical Engineering and Computer Science Department, University of California, Irvine. Email: athina@uci.edu. Mail: CalIT2 Bldg, Suite 4100, Irvine, CA 92697.

K. K. Ramakrishnan is with AT&T Labs Research. Email: kkrama@research.att.com. Mail: 180 Park Avenue, Building 103 Florham Park, NJ 07932.

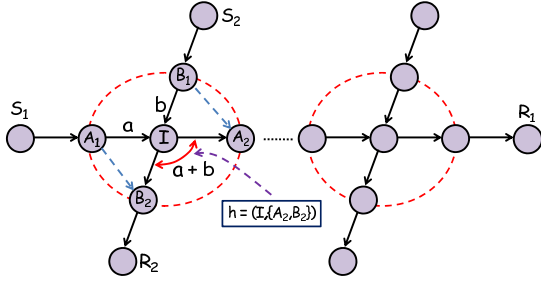


Fig. 1. Example of a unicast flow (from S_1 to R_1) traversing multiple wireless hops. Each node performs (intra- and inter-session) NC. The neighborhood of I is shown here in detail. (Two unicast flows, $S_1 - R_1$ and $S_2 - R_2$, meeting at intermediate node I . I receives packets a, b from nodes A_1, B_1 , respectively. It can choose to broadcast a, b or $a + b$ in a single transmission to both receivers. The next hops A_2, B_2 can decode $a + b$ because they overhear packets b and a transmitted from B_1, A_1 , respectively.)

session NC, even in the presence of medium-high loss rates, thus improving throughput. Second, the use of intra-session NC makes all packets in the session equally beneficial. Thus, I²NC eliminates the need to know the exact packets that have been overheard by the neighbors of intermediate node I . It is sufficient to know the loss probabilities of overheard and transmitted packets. In our scheme, this information is reported by each node to the nodes in its neighborhood which makes NC possible even at higher loss rates.

Adding redundancy in this setting is non-trivial, since a flow is affected not only by loss on its direct links, but also by loss on overhearing links. This affects the decodability of coded packets. Therefore, the amount of redundancy needed to be determined carefully.

Example 1 - continued: Consider again the neighborhood of I in Fig. 1. Flow 2 (originated from S_2) is affected not only by loss on its own path $B_1 - I - B_2$, but also by loss on the overhearing link $A_1 - B_2$, which affects the decodability of coded packet $a + b$ at B_2 . In order to protect flow 2 from high loss rate on the overhearing link $A_1 - B_2$, I may decide either to add redundancy on flow 2, or to not perform coding, or a combination of the two. On the other hand, I may also decide to add redundancy on flow 1 (originated from S_1), to correct loss on the overhearing link $A_1 - B_2$, thus helping B_2 to receive a and decode $a + b$. ■

Therefore, a number of questions need to be addressed in the design of a system that combines both intra- and inter-session NC. In particular:

- Q1: *How to gracefully combine intra- and inter-session NC?* We propose a generation-based design, and specify the order we perform the two types of coding.
- Q2: *How much redundancy to add in each flow?* We show how to adjust the amount of redundancy after taking into account the loss on the direct and overhearing links. We implement the intra-session NC functionality as a thin layer between IP and transport layer.
- Q3: *What percentage of flows should be coded together and what parts should remain uncoded?* We design algorithms that make this decision taking into account the

loss characteristics on the direct and overhearing links. We implement this and other functionality (e.g., queue management) performed with or after inter-session NC as a layer between MAC and IP.

- Q4: *What information should be reported to make these decisions?* We propose two schemes: I²NC-state, which needs to know the state (i.e., overheard packets) of the neighbors; and I²NC-stateless, which only needs to know the loss rate of links in the neighborhood.

Our approach is grounded on a network utility maximization (NUM) framework [7]. We formulate two variants of the problem, depending on available information (as in Q4 above). The solution of each problem decomposes into several parts with an intuitive interpretation, such as rate control, NC rate, redundancy rate, queue management, and scheduling. The structure of the optimal solution provides insight into the design of our two schemes, I²NC-state and I²NC-stateless.

We evaluate our schemes in a multi-hop setting, and we consider their interaction with the transport layer, including TCP and UDP. We propose a thin adaptation layer at the interface between TCP and the underlying coding, to best match the interaction of the two. We perform simulations in GloMoSim [8], and we show that our schemes significantly improve throughput compared to COPE.

The structure of the rest of the paper is as follows. Section II presents related work. Section III gives an overview of the system model. Section IV presents the NUM formulation and solution. Section V presents the design of the I²NC schemes in detail. Section VI presents simulation results. Section VII concludes the paper.

II. RELATED WORK

COPE and follow-up work. This paper builds on COPE, a practical scheme for one-hop NC across unicast sessions in wireless mesh networks [2], which has generated a lot of research interest. Some researchers tried to model and analyze COPE [9], [10], [11]. Some others proposed new coded wireless systems, based on the idea of COPE [12], [5]. In [13], the performance of COPE is improved by looking at its interaction with MAC fairness. Our recent work in [6] improves TCP's performance over COPE with a NC-aware queue management scheme. This paper also improves COPE by adding intra-session redundancy with a cross-layer design and reducing the amount of information that is needed to be exchanged among nodes cooperatively, i.e., nodes no longer need to know the exact packets overheard by their neighbors and can operate only with knowledge of the link loss rates.

NUM in coded systems. The NUM framework can be applied in networks, to understand how different layers and/or modules (such as flow control, congestion control, routing, etc.) should be restructured when NC is used. Although the approach is general, the parts and interpretation of the distributed solution is highly problem-specific. For NUM to be successful, the optimization model must be formulated so as to capture and exploit the NC properties. This is highly non-trivial and problem-specific. A body of work has looked at the joint optimization of NC of unicast flows, formulated in a NUM framework.

Optimal scheduling and routing for COPE are considered in [9] and [11], respectively. A linear optimization framework for packing butterflies is proposed in [4]. A re-transmission scheme for one-hop NC is proposed in [14]. Forward error correction over wireless for pairwise NC is proposed in [15], [16], which are also the most closely related formulations to ours. Our main differences are that we consider: (i) multiple flows coded together instead of pairwise, (ii) local instead of end-to-end redundancy, and (iii) the effect of losses over direct and overhearing links, to generate the right amount of redundancy.

Dealing with wireless loss. Recent studies of IEEE 802.11b based wireless mesh networks [17], [18], have reported packet loss rates as high as 50%. Dealing such level of loss in wireless networks is a hard enough problem on its own, which is further amplified by NC. There is a wide spectrum of well-studied options for dealing with loss, *e.g.*, using redundancy and/or re-transmissions, locally (MAC) and/or end-to-end (transport layer). Local re-transmissions increase end-to-end delay and jitter, which, if excessive, may cause TCP timeouts or hurt real-time multimedia. Furthermore, the best re-transmission scheme for network coded packets varies with the loss probability¹ and it is hard to switch among re-transmission policies when the loss rate varies over time. Re-transmission also requires state synchronization to perform inter-session NC, which is not reliable at all loss rates. We follow an alternative approach of local redundancy because we are interested in keeping delay low and we want to eliminate the need for knowing the state of neighbors.

There is extended work on TCP over wireless. One key problem is the need to distinguish between wireless and congestion loss and have TCP react only to congestion; this is possible *e.g.*, through Explicit Congestion Notification (ECN). When re-transmissions exceed the delay budget, end-to-end redundancy may also be used to combat loss on the path [19]. The error-correcting capabilities of intra-session NC have recently been used in conjunction with the TCP sliding window in [20]. In contrast, we focus on one-hop inter-session coding rather than end-to-end intra-session coding.

III. SYSTEM OVERVIEW

We consider multi-hop wireless networks, where intermediate nodes perform intra- and inter-session NC (I²NC). Next, we provide an overview of the system and highlight some of its key characteristics.

A. Notation and Setup

1) *Sources and Flows:* Let \mathcal{S} be the set of unicast flows between source-destination pairs in the network. Each flow $s \in \mathcal{S}$ is associated with a rate x_s and a utility function $U_s(x_s)$, which we assume to be a strictly concave function of x_s .

¹ We have observed through simulations that if a network coded packet is lost for one receiver but received correctly for other receiver(s), it is better to re-transmit the same network coded packet for low loss rates. However, it is better to combine the packet which is lost in the previous transmission with new packets for high loss rates.

2) *Wireless Transmission:* Packets from a source (*e.g.*, S_1 in Fig. 1) traverse potentially multiple wireless hops before being received by the receiver (*e.g.*, R_1). We consider a model for interference described in [22]: each node can either transmit or receive at the same time, and all transmissions in the range of the receiver are considered as interfering.

We use the following terminology for wireless. 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 the set of nodes in \mathcal{J} , *i.e.*, $h(i) = i$ and $h(\mathcal{J}) = \mathcal{J}$. We use these notations interchangeably in the rest of the paper. Each hyperarc h is associated with a channel capacity R_h . Since h is a set of links, R_h is the minimum capacity of all the links in the hyperarc, *i.e.*, $R_h = \min_{j \in h(\mathcal{J})} \{R_{i,j}\}$ s.t. $i \in \mathcal{N}$. In the example of node I in Fig. 1, $h = (I, \{B_2, A_2\})$ is one of the hyperarcs, and its capacity is $\min\{R_{\{I, B_2\}}, R_{\{I, A_2\}}\}$.

Note that with both intra- and inter-session NC, it is possible to construct more than one code over a hyperarc h . Let \mathcal{K}_h be the set of inter-session network codes over a hyperarc h . $\mathcal{S}_k \subseteq \mathcal{S}$ be the set of flows coded together using code $k \in \mathcal{K}_h$ and broadcast over h .²

Given \mathcal{H} , we can construct the conflict graph $\mathcal{C} = (\mathcal{A}, \mathcal{I})$, whose vertices are the hyperarcs of \mathcal{H} and edges indicate interference between hyperarcs. A clique $\mathcal{C}_q \subseteq \mathcal{A}$ consists of several hyperarcs, at most one of which can transmit without interference, *i.e.*, a transmission over a hyperarc interferes with transmissions over other hyperarcs in the same clique.

3) *Loss Model:* A flow s may experience loss in two forms: loss ρ_h^s over the direct transmission links; or loss $\rho_{h,k}^{s,s'}$ of antidotes³ on overhearing links. These two types of loss have different impact on network coded flows.

First, let us discuss loss on the direct links. A flow s transmitted over hyperarc h experiences loss with probability ρ_h^s . This probability is different per flow s , even if several flows are coded and transmitted over the same hyperarc h , because different flows are transmitted to different next hops, thus see different channels. For example, in Fig. 1, $\rho_{(I, \{B_2, A_2\})}^{S_1}$ is equal to the loss probability over link $I - A_2$ and $\rho_{(I, \{B_2, A_2\})}^{S_2}$ is equal to the loss probability over link $I - B_2$.

Second, let us discuss the effect of lost antidotes on the overhearing link. Consider that flow s is combined with flow s' s.t. $s \neq s'$, and that some packets of flow s' are lost on the overhearing link to the next hop of s . Then, coded packets cannot be decoded at the next hop and flow s loses packets, with probability $\rho_{h,k}^{s,s'}$. For example, in Fig. 1, packets from flow S_1 cannot be decoded (hence are lost) at node A_2 due to loss of antidotes from flow S_2 on the overhearing link $B_1 - A_2$.

In our formulation and analysis, we assume that ρ_h^s and $\rho_{h,k}^{s,s'}$

²Note that we consider constructive inter-session NC, *i.e.*, network codes $k \in \mathcal{K}_{i,\mathcal{J}}$ as well as $h = (i, \mathcal{J})$ is determined at each node with periodic control packet exchanges or estimated through routing table.

³Following the poison-antidote terminology of [4], we call “antidotes” the packets of flows s' that are coded together with s , and thus are needed for the next hop of s to be able to decode. *E.g.*, in Fig. 1, a is the “antidote” that B_2 needs to overhear over link $A_1 - B_2$, to decode $a + b$ and obtain b .

are i.i.d. according to a uniform distribution. However, in our simulations, we consider a Rayleigh fading channel model. The loss probabilities are calculated at each intermediate node as explained later in this section.

4) *Routing*: Each flow $s \in \mathcal{S}$ follows a single path $\mathcal{P}_s \subseteq \mathcal{N}$ from the source to the destination, which is pre-determined by a routing protocol, *e.g.*, OLSR or AODV, and given as input to our problem. Note that the nature of wireless networks is time varying, *i.e.*, nodes join and leave the system dynamically. In such cases, the routing protocol actively determines new paths which are used as input to our problem. It is not critical that the paths remain fixed, neither from a theoretical nor from a practical point of view, as explained in the following sections. Also, note that several different hyperarcs may connect two consecutive nodes along the path. We define $H_{h,k}^s = 1$ if s is transmitted through hyperarc h using network code $k \in \mathcal{K}_h$; and $H_{h,k}^s = 0$, otherwise.

B. Intra- and Inter-session Network Coding

Next, we give an overview of how an intermediate node performs intra- and inter-session NC. The implementation details are provided in Section V.

1) Intra-session Network Coding (for Error Correction):

Consider the commonly used generation-based NC [23]: packets from flow $s \in \mathcal{S}$ are divided into generations (note that we use “generation” and “block” terms interchangeably), with size G^s . At the source s , packets within the same generation are linearly combined (assuming large enough field size) to generate G^s network coded packets. Each intermediate node along the path of flow s adds P^s parity packets, depending on the loss rates of the links involved in this hop. At the next hop, it is sufficient to receive G^s out of $G^s + P^s$ packets. The same process is repeated at every intermediate node until the receiver receives G^s error-free packets, which can then be decoded and be passed on to the application.

There are many ways to generate parities (P^s) in practice. We use generation based intra-session NC [23] for this purpose. Although one could use various coding techniques, such as Reed-Solomon or Fountain codes, using intra-session NC has several advantages. First, it has lower computational complexity. Second, in systems like COPE that already implement inter-session NC, it is natural to incrementally add intra-session NC functionality. Moreover, in this setting, hop-by-hop intra-session coding (in which redundant packets are generated at each hop) is clearly a better choice than end-to-end coding for dealing with loss. In terms of performance, hop-by-hop coding achieves higher end-to-end throughput (thanks to introducing less redundancy than end-to-end coding), without adding high complexity (and thus delay) to the intermediate nodes. Furthermore, in terms of system implementation, our hop-by-hop scheme requires minimal modifications on top of the inter-session NC, which is already implemented.

2) Inter-session Network Coding (for Throughput):

After an intermediate node has added redundancy (P^s) to flow s , it treats all $(G^s + P^s)$ packets as indistinguishable parts of the same flow. Inter-session NC is applied on top of the already intra-coded flows, as a thin layer between MAC and IP (similar

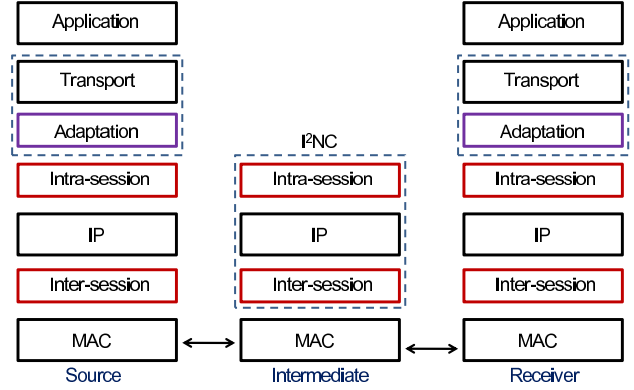


Fig. 2. Operations taking place at end-points and intermediate nodes.

to COPE), shown in Fig. 2. We design two schemes, I^2NC -state and I^2NC -stateless, depending on the type of information that is needed to make network coding decisions. We define as state of a node the information about which exact packets have been overheard at that node.

I^2NC -state: First, we assume that intermediate nodes use COPE [2] for inter-session coding. Each node i listens all transmissions in its neighborhood, stores the overheard packets in its decoding buffer, and periodically advertises the content of this buffer to its neighbors. When a node i wants to transmit a packet, it checks or estimates the contents of the decoding buffer of its neighbors. If there is a coding opportunity, the node combines the relevant packets using simple coding operations (XOR) and broadcasts the combination to \mathcal{J} . The content of the decoding buffers needs to be exchanged, in order to make NC decisions, *i.e.*, state synchronization is required.

I^2NC -stateless: Second, we design an improved version of COPE, which no longer requires state synchronization. The key idea is to exploit the fact that the redundancy already introduced by intra-session coding makes all $G^s + P^s$ packets in a generation equally important.⁴ In this improved scheme, each node i still listens to all transmissions in its neighborhood and stores the overheard packets.⁵ The node periodically advertises the loss rate for each received and overheard flow, which is then provided as input to the intra-session NC module to determine the amount of redundancy needed. In particular, the loss rates are calculated at each intermediate node as one minus the ratio of correctly received packets over all the packets in a generation. Also, the loss rate over overhearing links is calculated as effective loss rate. *E.g.*, in Fig. 1, the loss rate at node A_2 is calculated as follows. If $G^{S_1} + P^{S_1}$ packets are sent by B_1 and at least G^{S_1} packets are received at A_2 , then the loss rate is set to 0. If $G^{S_1} - u$ packets are received by A_2 such that $u \leq G^{S_1}$, then the loss rate is set to u/G^{S_1} . The loss rates calculated for each generation are advertised to other nodes in the neighborhood. Then, each node calculates

⁴It no longer matters which exact packets a node has. As long as a node has any G^s out of $G^s + P^s$, it can decode with high probability. As long as it knows the percentage of received packets it can make coding decisions.

⁵Note that when inter-session network coded packets are overheard, they are not stored in the “decoding buffer”, but discarded.

its loss probabilities (ρ_h^s and $\rho_{h,k}^{s,s'}$) as weighted average of the loss rates it has received.

In summary, there is a synergy between intra- and inter-session NC. Intra-session makes the process sequence agnostic, which allows inter-session coding to operate using only information about the loss rates, not about the identity of the packets. The loss rates can be used as input for tuning the amount of redundancy in intra-session NC. In terms of implementation, the two modules are separable: an intermediate node first performs intra-session, then inter-session NC.

IV. NETWORK UTILITY MAXIMIZATION FORMULATION

A. I^2NC -state Scheme

1) *Formulation*: Our objective is to maximize the total utility function by optimally choosing the flow rates x_s at sources $s \in \mathcal{S}$, as well as the following variables at the intermediate nodes: the fraction $\alpha_{h,k}^s$ (or ‘‘traffic splitting’’ parameters, following the terminology of [24]) of flows inter-session coded using code $k \in \mathcal{K}_h$ over hyperarc h ; and the percentage of time $\tau_{h,k}$ each hyperarc is used.

$$\begin{aligned} & \max_{\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\tau}} \sum_{s \in \mathcal{S}} U_s(x_s) \\ & \text{s.t.} \quad \frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} \leq R_h \tau_{h,k}, \\ & \quad \forall h \in \mathcal{A}, k \in \mathcal{K}_h, s \in \mathcal{S}_k \\ & \quad \sum_{h(J)|h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h | s \in \mathcal{S}_k} \alpha_{h,k}^s = 1, \quad \forall s \in \mathcal{S}, i \in \mathcal{P}_s \\ & \quad \sum_{h \in \mathcal{C}_q} \sum_{k \in \mathcal{K}_h} \tau_{h,k} \leq \gamma, \quad \forall \mathcal{C}_q \subseteq \mathcal{A} \end{aligned} \quad (1)$$

The first constraint is the capacity constraint for each flow $s \in \mathcal{S}_k$. It is well-known, [25], that NC allows flows that are coded together in code $k \in \mathcal{K}_h$, to coexist, *i.e.*, each have rate up to the rate allocated to that code k . The right hand side, $R_h \tau_{h,k}$, is the capacity of hyperarc h ; $\tau_{h,k}$ is the percentage of time hyperarc h can be used for transmitting the k -th network code. $\tau_{h,k}$ is determined by scheduling in the third constraint, taking into account interference: all hyperarcs in a clique interfere and should time-share the medium. Therefore, the sum of the time allocated to all hyperarcs in a clique should be less than an over-provisioning factor, $\gamma \leq 1$. The second constraint is the flow conservation: at every node i on the path \mathcal{P}_s of source s , the sum of $\alpha_{h,k}^s$ over all network codes and hyperarcs should be equal to 1. Indeed, when a flow enters a particular node i , it can be transmitted to its next hop j as part of different network coded and uncoded flows.

The first constraint is key to our work because it determines how to deal with loss on the direct (ρ_h^s) and overhearing ($\rho_{h,k}^{s,s'}$) links and how large a fraction ($\alpha_{h,k}^s$) of flow rate (x_s) to code in the k -th code over hyperarc h . Let us discuss the left hand side in more detail.⁶

⁶Note that our formulation has two novel aspects, compared to prior work, which allow us to better handle loss and parities. First, we allow for flows coded together to have different rates (in the first constraint in Eq. (1)). Second, we allow for loss rates of each link to be specified separately, even for links in the same hyperarc.

The first term refers to the direct link of flow s . $H_{h,k}^s \alpha_{h,k}^s x_s$ is the fraction of flow rate x_s allocated to code k and hyperarc h . It is scaled by $1 - \rho_h^s$ to indicate that we use redundancy to protect against loss that flow s experiences with probability ρ_h^s . ($H_{h,k}^s \alpha_{h,k}^s x_s / (1 - \rho_h^s)$) is the total rate of flow s , including data and redundancy.

The second term refers to loss on the overhearing links. $\sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'}$ is the amount of redundancy (via intra-session coding) added by the intermediate node on flow(s) s' to protect flow s against loss of antidote packets. These antidotes come from other flows ($s' \in \mathcal{K}_h$) that are coded together with flow s , reach the next hop for flow s through the overhearing links, and are needed to decode inter-session coded packets.

Example 1- continued. In Fig. 1, let us consider flow 2 from B_1 to B_2 , as the flow of interest. The intermediate node I adds redundancy to S_2 to protect against loss rate $\rho_{(I, \{B_2, A_2\})}^{S_2}$ on the direct link $I - B_2$. It also adds redundancy to flow 1 to protect against loss rate $\rho_{(I, \{B_2, A_2\})}^{S_2, S_1}$ of antidotes coming to B_2 from flow 1 over the overhearing link $A_1 - B_2$.

2) *Optimal Solution*: To solve Eq. (1) we follow a similar approach proposed in [36]. First, we relax the capacity constraint in Eq. (1), and we have the Lagrangian function:

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\tau}, \mathbf{q}) = & \sum_{s \in \mathcal{S}} U_s(x_s) - \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} q_{h,k}^s \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right. \\ & \left. + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right), \end{aligned} \quad (2)$$

where $q_{h,k}^s$ is the Lagrange multiplier, which can be interpreted as the queue size for k -th network code at hyperarc h for flow s . We define $\rho_{h,k}^{s,s'} = 0$ if $s = s', \forall s, s' \in \mathcal{S}$ and we rewrite $\sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k}$ as $\sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}_h | s \in \mathcal{S}_k}$. The Lagrange function is $L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\tau}, \mathbf{q}) = \sum_{s \in \mathcal{S}} (U_s(x_s) - x_s \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h | s \in \mathcal{S}_k} H_{h,k}^s \alpha_{h,k}^s ((q_{h,k}^s) / (1 - \rho_h^s) + \sum_{s' \in \mathcal{S}_k} q_{h,k}^{s'} \rho_{h,k}^{s',s})) + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} q_{h,k}^s R_h \tau_{h,k}$. It can be decomposed into several intuitive parts (rate control, traffic splitting, scheduling, and queue update), each of which solves the optimization problem for one variable.

Rate Control. First, we solve the Lagrangian w.r.t x_s :

$$x_s = (U'_s)^{-1} \left(\sum_{i \in \mathcal{P}_s} Q_i^s \right), \quad (3)$$

where $(U'_s)^{-1}$ is the inverse function of the derivative of U_s , and Q_i^s is the occupancy of flow s at node i and expressed as

$$Q_i^s = \sum_{h(J)|h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h | s \in \mathcal{S}_k} H_{h,k}^s \alpha_{h,k}^s Q_{h,k}^s, \quad (4)$$

where $Q_{h,k}^s$ is the queue size of flow s associated with hyperarc and network code pair $\{h, k\}$:

$$Q_{h,k}^s = \frac{q_{h,k}^s}{1 - \rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} q_{h,k}^{s'} \rho_{h,k}^{s',s} \quad (5)$$

Traffic Splitting. Second, we solve the Lagrangian for $\alpha_{h,k}^s$. At each node i along the path (*i.e.*, $i \in \mathcal{P}_s$), the traffic splitting problem can be expressed as follows:

$$\begin{aligned} & \min_{\boldsymbol{\alpha}} \sum_{h(J)|h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h | s \in \mathcal{S}_k} \alpha_{h,k}^s H_{h,k}^s Q_{h,k}^s \\ & \text{s.t.} \quad \sum_{h(J)|h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h | s \in \mathcal{S}_k} \alpha_{h,k}^s = 1. \end{aligned} \quad (6)$$

Let us assume that $E_i[Q(t)]$ is the maximal $\bar{Q}_i(t)$ at time t such that $\bar{Q}_i(t) = \frac{1}{|\mathcal{A}'_i(t)|} \sum_{\varphi \in \mathcal{A}'_i(t)} H_{h,k}^s Q_{h,k}^s(t)$ with $\mathcal{A}'_i(t) := \{\varphi = \{h(\mathcal{J}), k\} | \alpha_{h,k}^s > 0 \text{ or } H_{h,k}^s Q_{h,k}^s \leq \bar{Q}_i(t), h(\mathcal{J}) \in \mathcal{N} \text{ s.t. } h \in \mathcal{A}, k \in \mathcal{K}_h\}$. At each node i , the amount of traffic splitting factor $\alpha_{h,k}^s$ for flow s over hyperarc h and code k follows; $\dot{\alpha}_{h,k}^s = \kappa_i [E_i[Q] - H_{h,k}^s Q_{h,k}^s]_{\alpha_{h,k}^s}^+$, where κ_i is a positive constant, and $[b]_z^+ = b$ if $z \geq 0$ and $[b]_z^+ = 0$ if $b \leq 0$ and $z = 0$. It can be seen that $\sum_{h(\mathcal{J})|h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} \dot{\alpha}_{h,k}^s = 0$ and $\sum_{h(\mathcal{J})|h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} \dot{\alpha}_{h,k}^s H_{h,k}^s Q_{h,k}^s \leq 0$. Also, $\sum_{h(\mathcal{J})|h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} \dot{\alpha}_{h,k}^s H_{h,k}^s Q_{h,k}^s = 0$ only if $\dot{\alpha}_{h,k}^s = 0$ which is possible only if $H_{h,k}^s Q_{h,k}^s \geq \bar{Q}_i$, and $\alpha_{h,k}^s (H_{h,k}^s Q_{h,k}^s - \bar{Q}_i) = 0$.

The structure of the optimal solution of Eq. (6) (*i.e.*, $\dot{\alpha}_{h,k}^s = \kappa_i [E_i[Q] - H_{h,k}^s Q_{h,k}^s]_{\alpha_{h,k}^s}^+$) has the following interpretation: the higher the loss rate of antidotes on overhearing links $\rho_{h,k}^{s',s}$, the higher $Q_{h,k}^s$, and the smaller $\alpha_{h,k}^s$. This means that flow s should code fewer packets with packets from flow(s) s' in code k , when antidotes from s' are likely to be lost.

Example 1 - continued: In Fig. 1, this means that I should combine fewer packets from the two flows if there is loss on the overhearing link $A_1 - B_2$. In the extreme case where loss rate is 1 over the link $A_1 - B_2$, inter-session coding should be turned off. At the other extreme, where there is no loss, the two flows should always be combined. ■

Scheduling. Third, we solve the Lagrangian for $\tau_{h,k}$. This problem is solved for every hyperarc and every clique for the conflict graphs in the hypergraph.

$$\begin{aligned} \max_{\tau} \quad & \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} q_{h,k}^s R_h \tau_{h,k} \\ \text{s.t.} \quad & \sum_{h \in \mathcal{C}_q} \sum_{k \in \mathcal{K}_h} \tau_{h,k} \leq \tau, \quad \forall \mathcal{C}_q \subseteq \mathcal{A} \end{aligned} \quad (7)$$

Let us assume that $Q_{h,k} = R_h \sum_{s \in \mathcal{S}_k} q_{h,k}^s$, and $E_{\mathcal{C}_q}[Q(t)]$ is the minimal $\bar{Q}_{\mathcal{C}_q}(t)$ at time t such that; $\bar{Q}_{\mathcal{C}_q}(t) = \frac{1}{|\mathcal{A}'_{\mathcal{C}_q}(t)|} \sum_{\phi \in \mathcal{A}'_{\mathcal{C}_q}(t)} Q_{h,k}(t)$ with $\mathcal{A}'_{\mathcal{C}_q}(t) := \{\phi = \{h, k\} | \tau_{h,k} > 0 \text{ or } Q_{h,k}(t) \geq \bar{Q}_{\mathcal{C}_q}(t), h \in \mathcal{A}, k \in \mathcal{K}_h\}$. At each clique \mathcal{C}_q , the fraction of the time $\tau_{h,k}$ that is allocated to hyperarc h , and code k is as follows; $\dot{\tau}_{h,k} = \varepsilon_{\mathcal{C}_q} [Q_{h,k} - E_{\mathcal{C}_q}[Q]]_{\tau_{h,k}}^+$, where $\varepsilon_{\mathcal{C}_q}$ is a positive constant and $[b]_z^+ = b$ if $z \geq 0$ and $[b]_z^+ = 0$ if $b \leq 0$ and $z = 0$. It can be seen that $\sum_{h \in \mathcal{C}_q} \sum_{k \in \mathcal{K}_h} \dot{\tau}_{h,k} = 0$ and $\sum_{h \in \mathcal{C}_q} \sum_{k \in \mathcal{K}_h} \dot{\tau}_{h,k} Q_{h,k} \geq 0$. Also, $\sum_{h \in \mathcal{C}_q} \sum_{k \in \mathcal{K}_h} \dot{\tau}_{h,k} Q_{h,k} = 0$ only if $\dot{\tau}_{h,k} = 0$ which requires that $Q_{h,k} = \bar{Q}_{\mathcal{C}_q}$ or $\tau_{h,k} = 0$ and $Q_{h,k} < \bar{Q}_{\mathcal{C}_q}$.

Queue Update. We find the Lagrange multipliers (queue sizes) $q_{h,k}^s$, using the gradient descent:

$$\begin{aligned} q_{h,k}^s(t+1) = & \{q_{h,k}^s(t) + c_t \left\{ \frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} + \right. \\ & \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right\}^+ \end{aligned} \quad (8)$$

where t is the iteration number, c_t is a small constant, and the $^+$ operator makes the Lagrange multipliers positive. $q_{h,k}^s$ is interpreted as the queue for flow s allocated for the k -

th network code over hyperarc $\forall h \in \mathcal{A}$. Indeed, in Eq. (8), $q_{h,k}^s$ is updated with the difference between the incoming $(H_{h,k}^s \alpha_{h,k}^s x_s) / (1 - \rho_h^s) + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'}$ and outgoing $R_h \tau_{h,k}$ traffic rates at h .⁷

B. P²NC-stateless Scheme

The second term in Eq. (1) describes the redundancy added by node i to protect flow s from loss of antidotes on the overhearing link. An implicit assumption was that node i knows what antidotes are available at the next hop and uses only those packets for inter-session coding. However, this knowledge can be imperfect, especially in the presence of loss. Here, we formulate a variation of the problem, where such knowledge is not necessary. Instead, node i needs to know only the loss rate on all the links to the next hop for flow s (*e.g.*, in Fig. 1 for flow 2 (S_2), these are links $I - B_2$ and $A_1 - B_2$).

We replace the capacity constraint in Eq. (1) with:

$$\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} \frac{H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'}}{1 - \rho_h^s} \leq R_h \tau_{h,k} \quad (9)$$

and this is $\forall h \in \mathcal{A}, k \in \mathcal{K}_h, s \in \mathcal{S}_k$. The other constraints remain the same as in Eq. (1). The difference from Eq. (1) is in the second term, related to the overheard packets at the next hop. Any fraction of flow s' added as redundancy to flow s , as well as overheard packets from s' in the next hop, help to decode inter-session coded packets of s with flow s' . To protect transmissions of these ‘‘helping’’ fractions $(H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'})$ from being lost on the direct link to the next hop of flow s (*e.g.*, from I to B_2), we add redundancy to match the loss rate of that direct link (ρ_h^s in general, $\rho_{\{I, B_2\}}^{S_2}$ in the example).

This is why the term $H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'}$ is divided by $1 - \rho_h^s$. The solution of this optimization problem also decomposes into rate control, traffic splitting, and scheduling problems, which correspond to Eq. (3), (6), and (7), respectively. $Q_{h,k}^s$ needs to be updated:

$$Q_{h,k}^s = \frac{q_{h,k}^s}{1 - \rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} \frac{q_{h,k}^{s'} \rho_{h,k}^{s,s'}}{1 - \rho_h^s}. \quad (10)$$

The Lagrange multiplier is updated as follows;

$$\begin{aligned} q_{h,k}^s(t+1) = & \{q_{h,k}^s(t) + c_t \left\{ \frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} + \right. \\ & \left. \sum_{s' \in \mathcal{S}_k - \{s\}} \frac{H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'}}{1 - \rho_h^s} - R_h \tau_{h,k} \right\}^+ \end{aligned} \quad (11)$$

We provide the convergence analysis of our solution in Appendix A.⁸ We first give the proof of convergence, then we verify the convergence through numerical calculations.

⁷Note that the queue update in Eq. (8) can be re-written as; $\dot{q}_{h,k}^s = \gamma_h \left[\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right]_{q_{h,k}^s}^+$, where γ_h is a positive constant.

⁸We do not claim that the solution of our network utility maximization problem is the optimal solution to the general intra- and inter-session NC problem over wireless networks. This is well-known, open problem [26], [27], [28]. Even without an optimal, closed form solution, there is still value in using the structure of the solution to design mechanisms that perform well in practice, as we show through the numerical and simulations results in the next sections.

V. SYSTEM IMPLEMENTATION

We propose practical implementations of the I²NC-state and I²NC-stateless schemes (Fig. 2), following the NUM formulation structure.

A. Operation of End-Nodes

At the end nodes, there is an adaptation layer between transport and intra-session NC which has two tasks: (i) to interface between application and intra-session NC; and (ii) to optimize the reliability mechanism at the transport layer.

Task (i): At the source, the adaptation layer sets the generation (block) size G^s . G^s is set according to application; e.g., media transmission requirements for UDP, or set equal to TCP congestion window for TCP applications and changes over time. The adaptation layer receives G^s original packets p_1, p_2, \dots, p_{G^s} from the transport layer of flow s and generates G^s intra-session coded packets; $a_1 = p_1$, $a_2 = p_1 + p_2$, ..., $a_{G^s} = p_1 + \dots + p_{G^s}$. We call this coding “incremental additive coding”. We chose the incremental additive coding to avoid introducing coding delays (i.e., our algorithm does not need to wait G^s packets to encode packets) as proposed in [20]. The intra-session header includes the block id, packet id, block size, and coding coefficients. At the receiver side, the reverse operations are performed.

Task (ii): To further optimize the interaction between I²NC and transport, particularly TCP, we keep track of and acknowledge the number of received packets in a generation, rather than their sequence numbers (note that this part is not needed for UDP protocol). This idea is similar to the use of end-to-end FEC and intra-session NC that make TCP sequence agnostic [19], [29], [20]. E.g., if a receiver receives the first packet labeled with block id $g^s = 1$, then it generates an ACK with block id $g^s = 1$ and packet id $\eta^s = 1$. The uncoded packets, p_1, p_2, \dots, p_{G^s} , are stored in a buffer at the source for TCP ACK adaptation. E.g., if an ACK for block id $g^s = 1$ and packet id $\eta^s = 1$ is received by the source, then the TCP adapter matches this ACK to packet p_1 and informs TCP that packet p_1 is ACK-ed. As long as the TCP receiver transmits ACKs, the TCP clock moves, thus improving TCP goodput. After the ACK with the block and packet ids is transmitted by the TCP receiver, the packet is stored at the receiving buffer. When the last packet from a generation is received, then packets are decoded and passed to the application.

B. Operation of Intermediate Nodes

An intermediate node needs to take a number of actions when it receives (Alg. 1) or transmits (Alg. 2) a packet.

1) Receiving a packet and intra-session network coding:

Buffer packets. A node i may receive a packet from higher layers or from previous hops. In the latter case, if the received packet is inter-coded, it is decoded and the packet with destination to this node is stored (or is passed to transport if it is the last hop). If it is not the last hop, a packet $a_l \in \{a_1, a_2, \dots, a_{G^s}\}$ is stored in the output queue \mathcal{Q}_i . In addition to the physical output queue \mathcal{Q}_i , the node i keeps track of several virtual queues; $Q_{h,k}^s$ per (flow,

Algorithm 1 Node i processes packet a_l from flow s .

- 1: Read the information: packet a_l , from flow s (generation size G^s)
 - 2: Insert a_l into the physical output queue \mathcal{Q}_i .
 - 3: Determine $\{h^*, k^*\}$ and label a_l with $\{h^*, k^*\}$ pair and s
 - 4: Update q_{h^*,k^*}^s (using Eqs. (8) and (11)) and $q_{h^*,k^*}^{s'}$
 - 5: Calculate Q_{h^*,k^*}^s (using Eqs. (5) and (10)) and Q_i^s (using Eq. (4))
 - 6: $G_{h^*,k^*}^s = G_{h^*,k^*}^s + 1$
 - 7: **if** G^s packets from flow s are received at node i **then**
 - 8: Calculate the number of parities $P_{h,k}^{s,s}$, $P_{h,k}^{s',s}$
 - 9: Create $P_{h,k}^{s,s}$ parities from s and $P_{h,k}^{s',s}$ parities from s'
 - 10: Label all generated parities with $\{h, k\}$ pair and s
-

hyperarc, code). The packet a_l is labeled with (h^*, k^*, s) , which essentially indicates whether and how to code this packet according to the traffic splitting in Eq. (6): we pick $\{h^*, k^*\} = \arg \min_{h,k} \{H_{h,k}^s Q_{h,k}^s\}$, randomly breaking ties. Note that this labeling is local at the node, and does not introduce any transmission overhead.

Note that $H_{h,k}^s$ is the indicator whether flow s is transmitted over hyperarc h with code k . This indicator is determined at each node using a routing table which has a data structure to determine the next hops (note that paths do not need to be known by the sources or any node in the system). Basically, if a packet from flow s is able to reach to the next hop determined by the routing table when it transmitted over hyperarc h and with code k , then the indicator is set to 1, otherwise 0. We also note that in this system, as long as paths remain fixed for longer (at least longer than a time required to transmit a packet) time periods, we can see more benefit from NC, because each node will learn which flows can be network coded and estimate the loss rates better as time gets longer. However, even in the extreme case in which paths change very fast (say for example at every packet transmission), our system works well, but it does not fully exploit NC opportunities, since it cannot estimate whether NC is possible or not. However, it works not worse than a system without NC. Therefore, I²NC is designed to adapt to path changes and to exploit NC benefit if possible.

Update Virtual Queue Sizes. When packet a_l is selected to be transmitted with the k^* -th network code over hyperarc h^* , the virtual queues; Q_{h^*,k^*}^s and q_{h^*,k^*}^s should be updated. q_{h^*,k^*}^s is updated according to Eqs. (8) and (11). Q_{h^*,k^*}^s is calculated according to Eq. (5) for I²NC-state and Eq. (10) for I²NC-stateless. Q_i^s is calculated according to Eq. (4). Then, the number of packets G_{h^*,k^*}^s from the same generation that are allocated to h^*, k^* pair is incremented: $G_{h^*,k^*}^s = G_{h^*,k^*}^s + 1$. $G_{h,k}^s$ is set to 0 for each new generation.

Generate Parities. After G^s packets from a generation of flow s are received at node i , P^s parity packets are generated via intra-session NC (which is performed according to random linear NC [30]) and labeled with information (s, h, k) . There are two types of parities.

- $P_{h,k}^{s,s} = \lceil G_{h,k}^s \rho_h^s / (1 - \rho_h^s) \rceil$ parities are added on flow s 's virtual queue to correct for loss during direct transmission to the next hop over hyperarc h .
- $P_{h,k}^{s',s} = \lceil G_{h,k}^s \rho_{h,k}^{s',s} \rceil$, $\forall s' \in \mathcal{S}_k$ parities are added on the virtual queues of other flows s' that are inter-session coded together with s . This is to help the next hop for s' to decode despite losses on the overhearing link.

Algorithm 2 Node i transmits a packet.

- 1: Select $\{h^\dagger, k^\dagger\}$ pair that maximizes $Q_{h,k} = R_h(\sum_{s \in \mathcal{S}_k} q_{h,k}^s)$
 - 2: Initialize: $\xi = \emptyset$
 - 3: **for** $a_l \in \mathcal{Q}_i$ **do**
 - 4: **if** a_l is labeled with $\{h^\dagger, k^\dagger\}$ AND flow id label of a_l is different from $\forall a_{l'} \in \xi$ **then**
 - 5: **if** I²NC-state AND $\xi \cup a_l$ is decodable OR I²NC-stateless **then**
 - 6: Insert packet to ξ
 - 7: Network code (XOR) all packets in ξ
 - 8: Broadcast the network coded packet over hyperarc h^\dagger
 - 9: Update $q_{h^\dagger, k^\dagger}^s, \forall s \in \mathcal{S}_k$
 - 10: Re-calculate $Q_{h,k} = R_h(\sum_{s \in \mathcal{S}_k} q_{h,k}^s)$ and Q_i^s (using Eq. (4))
-

These parity packets are for I²NC-state. For I²NC-stateless $P_{h,k}^{s,s}$ is the same, but $P_{h,k}^{s',s} = \lceil G_{h,k}^s \rho_{h,k}^{s',s} / (1 - \rho_h^{s'}) \rceil$, i.e., additional redundancy is used to protect parity packets from loss on the direct link.

2) *Transmitting a packet and inter-session network coding:* We consider the 802.11 MAC. When a node i accesses a channel, $\{h^\dagger, k^\dagger\}$ is chosen to maximize $Q_{h,k} = R_h(\sum_{s \in \mathcal{S}_k} q_{h,k}^s)$ according to Eq. (7), randomly breaking ties. Although the pair $\{h^\dagger, k^\dagger\}$ determines the hyperarc, code and flows to be coded together in the next transmission, the specific packets from those flows still need to be selected and coded. We call these packets the set ξ , and select them using the procedure specified in Alg. 2.⁹

To achieve this, we first initialize the set of network coded packets $\xi = \emptyset$. For each packet $a_l \in \mathcal{Q}_i$, check whether a_l is labeled with $\{h^\dagger, k^\dagger\}$. If it is, then we check whether its flow id label already exists in one of the packets in ξ , i.e., another packet from the same flow has already been put in ξ . If not, there is one more check for I²NC-state for decodability at the next hops of all packets in the network code, based on reports or estimates of overheard packets in the next hops, similarly to [2]. If the packet is decodable with some probability larger than a threshold (default value is 0.20) then, a_l is inserted to ξ . In I²NC-stateless, the packet a_l is inserted to ξ without checking the decodability, which is ensured through the additional redundancy packets. This is the strength of I²NC-stateless: it eliminates the need to exchange detailed state, which is costly and unreliable at high loss rates. After all packets in \mathcal{Q}_i are checked, the labels (h, k, s) of the packets in ξ , inter-session NC header is added, and coded (XORed) and broadcast over h .

After a coded packet is transmitted, the virtual queues are updated according to Eqs. (8), (11). The queues Q_{h^\dagger, k^\dagger} and Q_i^s are calculated according to Eqs. (5), (10), (4).¹⁰

We note that in both I²NC-state and stateless, packets are network coded if some conditions are satisfied. However, if these conditions are not met, a packet without NC is still transmitted, because at least one packet is inserted in ξ (Alg. 2). Thus, we do not delay any packets in our schemes. Yet, delaying packets may create more NC opportunities and there is a tradeoff between delay and throughput. These issues have

⁹The inter-session NC header includes the number of coded packets together, next hop address, and the packet id's. Note that this header as well as the IP header of each packet are not network coded.

¹⁰Note that I²NC may cause re-ordering at the receiver, but since we already implemented intra-session NC, and made TCP receiver sequence agnostic in this term, out of packet delivery is not a problem for TCP.

been considered in some previous work [31], [32]. However, this is an aspect orthogonal to the focus of I²NC (which is the synergy between inter- and intra-session NC) and can potentially be combined with it.

3) *Keeping Track of and Exchanging State Information:* For I²NC-state, intermediate nodes need also to keep track of and exchange information with each other, so as to enable the intra- and inter-session NC modules to make their redundancy and coding decisions and to provide reliability. An approach similar to COPE is used: ACKs are sent after the reception and successful decoding of a packet. Information about overheard packets is piggy-backed on the ACKs. With I²NC-stateless, we only need neighbors to exchange information about the loss rates at the neighboring nodes. Information about the loss rates as well as the number of received packets at a generation is reported through control packets for every generation.¹¹ In order to provide reliability, we consider re-transmissions. In I²NC-state, a packet is removed from the output queue only after an ACK related to the packets is received. Otherwise, the packet is re-transmitted after a round trip time. In I²NC-stateless, packets are removed from the output queue when a control packets is received and confirms the successful transmission of all packets of the corresponding generation. Otherwise, a number of intra-session coded packets from the generation which are missing at the receiver are generated from the packets kept in the queue and transmitted.

4) *Congestion Control and Queue Management:* End-to-end congestion control (i.e., rate control) is given by Eq. (3) in which if $U_s(x_s) = \log(x_s)$, then $x_s = 1 / (\sum_{i \in \mathcal{P}_s} Q_i^s)$. This means that flow rate x_s is inversely proportional with increasing queue size over the path of flow s . This behavior is similar to TCP's end-to-end congestion control algorithm, where congestion at a node may result in one or more packets may be dropped from the buffer at this node. TCP reacts to packet drops by reducing its rate. Thus, TCP reduces its flow rate when queue size increases. This gives us intuition that TCP mimics the rate control part of the decomposed solution. This intuition has been validated in [7], [33], [34], [35].

Similarly, we consider that TCP already mimics the structure of the rate control part in Eq. (3). Therefore, upon congestion at node i , the per-flow queue sizes Q_i^s are compared and the last packet from flow s having the largest Q_i^s is dropped from the queue; in case of a tie, an incoming packet is dropped. We do not make any additional updates to TCP's end-to-end congestion control algorithm. Also, we do not implement any end-to-end congestion control mechanism for UDP. Our goal is to keep UDP as it is (without any end-to-end control) and show the effectiveness of I²NC-state and I²NC-stateless when there is no end-to-end control.

Example 2: Let us re-visit the X-topology from Fig. 1, shown again for convenience in Fig. 3, and illustrate how we perform intra- and inter-session NC under scheme I²NC-stateless. The loss probabilities over the direct $(I - B_2)$ and

¹¹In our implementation, the loss probabilities are calculated as weighted average of the loss rates. The weighted average is calculated over a window of 10 samples. The last 10 samples are ordered such that the newest sample is the first sample, and the oldest sample is the 10th sample. Each sample is given a weight inversely proportional to its sample number.

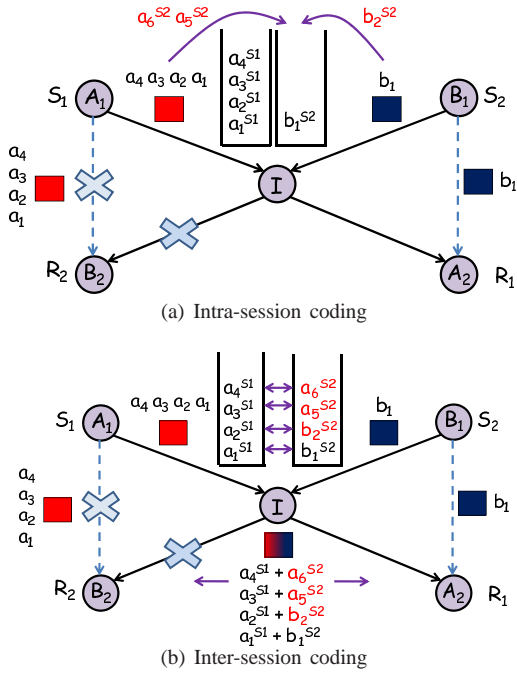


Fig. 3. Example of coding (under scheme I^2NC -stateless) at node I in the X-topology. There is loss only on two links: the direct link $I - B_2$ (with probability 0.5) and the overhearing link $A_1 - B_2$ (with probability 0.25).

overhearing ($A_1 - B_2$) links are assumed 0.5 and 0.25.

In Fig. 3(a), we describe intra-session NC. Let us assume the generation size of S_1 is $G^{S_1} = 4$ and S_2 is $G^{S_2} = 1$. The packets transmitted by A_1 , B_1 are a_1, a_2, a_3, a_4 and b_1 , respectively. Note that there is only one option for inter-session NC, *i.e.*, to XOR packets from the two flows, thus there exists only one possible network code $k = 1$ over hyperarc $h = (I, \{B_2, A_2\})$. All packets are labeled with this information and their flow ids. The labeled packets are $a_1^{S_1}, a_2^{S_1}, a_3^{S_1}, a_4^{S_1}$ and $b_1^{S_2}$. Parities are generated as follows. Since $G_{I, \{B_2, A_2\}}^{S_1} = 4$ and $G_{I, \{B_2, A_2\}}^{S_2} = 1$, the number of parities is $P_{I, \{B_2, A_2\}}^{S_1, S_1} = 0$, $P_{I, \{B_2, A_2\}}^{S_1, S_2} = 0$, $P_{I, \{B_2, A_2\}}^{S_2, S_2} = 1$ (thus generating one parity from flow S_2 and labeling it with S_2 , *i.e.*, $b_2^{S_2}$), and $P_{I, \{B_2, A_2\}}^{S_2, S_1} = 2$ (thus generating two parities from flow S_1 and labeling them with S_2 , *i.e.*, $a_5^{S_2}, a_6^{S_2}$).

In Fig. 3(b), we describe inter-session NC. Node I performs inter-session NC and transmits packets according to Alg. 2: it XORs packets from the two queues, for S_1, S_2 , and broadcasts over the hyperarc $(I, \{B_2, A_2\})$. In particular, it transmits the following packets: $a_1^{S_1} \oplus b_1^{S_2}$, $a_2^{S_1} \oplus b_2^{S_2}$, $a_3^{S_1} \oplus a_5^{S_2}$, and $a_4^{S_1} \oplus a_6^{S_2}$. A_2 receives and decodes all the packets. B_2 receives 3 packets on the average over overhearing link $A_1 - B_2$ and receives 2 packets over transmission link $I - B_2$. Five received packets allows B_2 to decode all five packets a_1, a_2, a_3, a_4, b_1 , so b_1 is successfully decoded. ■

VI. PERFORMANCE EVALUATION

A. Simulation Setup

We used the `GLoMoSim` simulator [8], which is well suited for simulating wireless environments. We considered various topologies: X topology, shown in part of Fig. 1 and repeated in Fig. 4(a); the cross-topology with four end-nodes generating

bi-directional traffic, with one relay shown in Fig. 4(b); the wheel topology shown in Fig. 4(c); and the multi-hop topology shown in Fig. 1. In X, cross, and wheel topologies, the intermediate node I is placed in a center of circle with radius $90m$ over $200m \times 200m$ terrain and all other nodes A_1, B_1 and etc. are placed around the circle. In the multi-hop topology of Fig. 1, two X topologies are cascaded and the distance between consecutive nodes is set to $90m$. The topology is over a $800m \times 300m$ terrain.

We also considered various *traffic scenarios*: FTP/TCP and CBR/UDP. TCP and CBR flows start at random times within the first $5sec$ and are on until the end of the simulation which is $60sec$. The CBR flow generates data packets at every $0.1ms$. IEEE 802.11b is used in the *MAC layer*, with the addition of the pseudo-broadcasting mechanism, as in COPE [2]. In terms of *wireless channel*, we simulated the two-ray path loss model and a Rayleigh fading channel with average channel loss rates 0, 20, 30, 40, 50 %. We have repeated each $60sec$ simulation for 10 seeds. Channel capacity is $1Mbps$, the buffer size at each node is set to 100 packets, packet sizes are set to $500B$, the generation size is set to 15 packets for UDP flows and to the TCP window size for TCP flows.

We compare our schemes (I^2NC -state and I^2NC -stateless) to no network coding (*noNC*), and COPE [2], in terms of total transport-level throughput (added over all flows).

B. Simulation Results

TCP Traffic. In Fig. 5, we present simulation results for two TCP flows in X topology shown in Fig. 4(a) to illustrate the key intuition of our approach. Consider, for the moment, that loss occurs only on one link, either (a) the overhearing link $A_1 - B_2$ or (b) the direct link $I - B_2$.

The first case is depicted in Fig. 5(a). Loss on the overhearing link does not affect the uncoded streams, thus the throughput of TCP+noNC does not change with loss rate. When NC is employed, reports carrying information about overheard packets may be delivered late to intermediate node I . Thus, there are some instances that intermediate node should make a decision even if it does not have the exact knowledge. In this case, I makes a decision probabilistically. Specifically, if decoding probability exceeds some threshold (20% in our simulations), I codes packets. However, some of these packets may not be decodable at the receiver. It is why the performance of TCP+COPE and TCP+ I^2NC -state reduce with increasing loss rate and equals to the throughput of TCP+noNC after 20% loss rate (NC is turned off after 20% loss rate). However, TCP+ I^2NC -state is still better than TCP+COPE, because when it makes probabilistic NC decision (when loss rate is less than 20%), it adds redundancy considering the loss rate over the overhearing link. This improves throughput, because adding redundancy using intra-session NC makes all packets equally beneficial to the receiver and the probability of decoding inter-session network coded packets increases. TCP+ I^2NC -stateless outperforms other schemes over the entire loss range. For example, if there is no loss, I^2NC -stateless still brings the benefit due to eliminating ACK packets and using less overhead to communicate information (*i.e.*, COPE and I^2NC -state exchanges the information about the overheard packets,

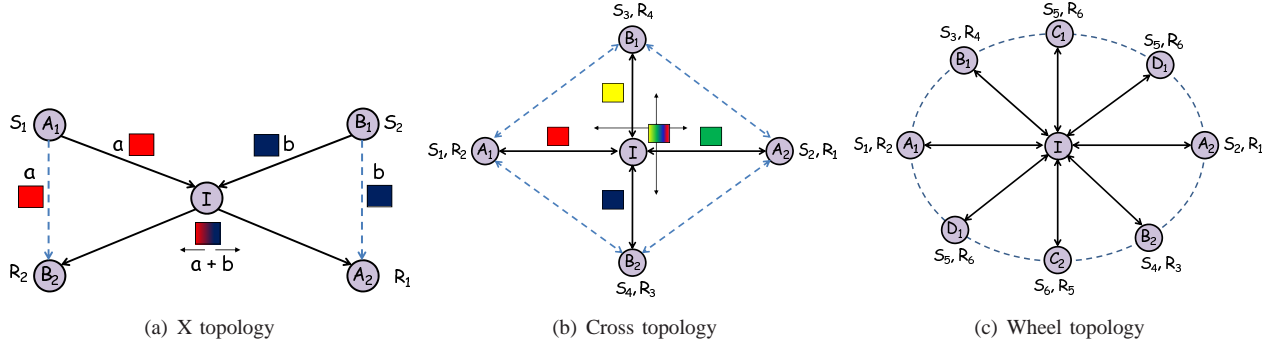


Fig. 4. Topologies under consideration. (a) X topology. Two unicast flows, S_1, R_1 , and S_2, R_2 , meeting at intermediate node I . (b) Cross topology. Four unicast flows, $S_1, R_1, S_2, R_2, S_3, R_3$, and S_4, R_4 , meeting at intermediate node I . (c) Wheel topology. Multiple unicast flows S_1, R_1, S_2, R_2 , etc., meeting at intermediate node I . In all three topologies, I opportunistically combine the packets and broadcast.

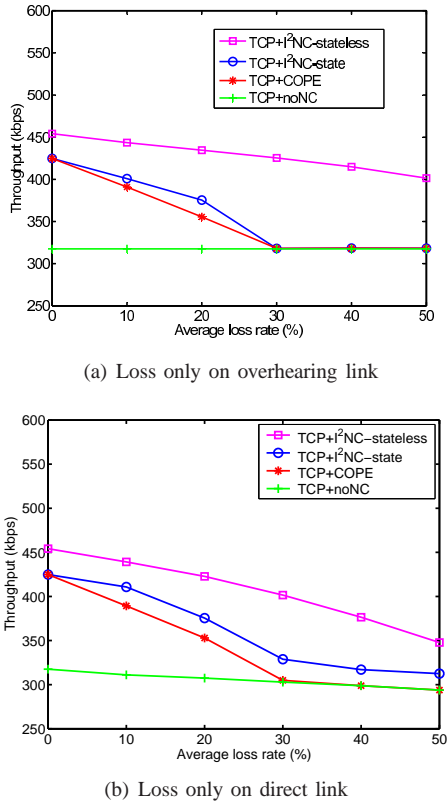


Fig. 5. X topology in Fig. 4(a). We show the total TCP throughput (added over two flows) vs. link loss rate, for two specific loss patterns. Loss happens only on one link, either: (a) the overhearing link $A_1 - B_2$ or (b) the direct link $I - B_2$. All other links are lossless.

while I²NC-stateless exchanges the information about the loss rates), thus using the medium more efficiently. When the loss rate increases, the improvement of I²NC-stateless becomes significant, reaching up to 30%. The reason is that at high loss rates, I²NC-state and COPE do not have reliable knowledge of the decoding buffers of their neighbors and cannot do NC efficiently. In contrast, I²NC-stateless does not rely on this information, but on the loss rate of the overhearing link to make NC decision. In the discussion of Example 1, we mentioned that at 50% loss rate, 16.6% improvement can be achieved via NC. Here, we see this improvement (13%) as well as the additional benefit of eliminating ACK packets (12%). Note that the total improvement is 25%.

The second case is depicted in Fig. 5(b). The throughput of

TCP+noNC decreases with increasing loss rate because, the loss is over the direct link and some packets whether they are coded or not are lost on the direct link ($I - B_2$). This leads to decrease in throughput level. TCP+I²NC-state outperforms TCP+COPE in this scenario, because I²NC-state corrects errors on the direct link thanks to the added redundancy which reduces the number of re-transmissions. Thus, I²NC-state uses the channel more efficiently than COPE and improves the throughput. Note that TCP+I²NC-state outperforms TCP+COPE even after 20% loss rate, although inter-session NC is turned off after this level. The reason is that although I²NC-state does not do inter-session NC after 20% loss rate, it keeps doing intra-session NC which adds redundancy to correct errors. Due to this property, TCP+I²NC-state outperforms TCP+COPE even at high loss rates. TCP+I²NC-stateless significantly outperforms all alternatives again due to performing NC at all loss rates and eliminating ACK packets.

Fig. 6 presents simulation results for TCP traffic over X, cross, and the multi-hop topologies assuming loss on all links. For ease of presentation, here, we report only the results when all links have the same loss probability.

Fig. 6(a) shows the results for the X topology. At low-medium loss rates (10% - 30%), I²NC-state and COPE are still able to do NC, so TCP+I²NC-state and TCP+COPE improve throughput significantly as compared to TCP+noNC. At higher loss rates, I²NC-state and COPE do not have reliable knowledge of the decoding buffers of their neighbors and cannot do NC efficiently. As a result, the improvement of TCP+I²NC-state and TCP+COPE as compared to TCP+noNC reduce with increasing loss rate. TCP+I²NC-state is better than TCP+COPE at higher loss rates thanks to its error correction mechanism. TCP+I²NC-stateless outperforms other schemes over the entire loss range thanks to combining NC and error correction as well as eliminating ACKs. For example, if there is no loss, TCP+I²NC-stateless still brings the benefit by eliminating ACK packets, thus using the medium more efficiently. When the loss rate increases, the improvement of I²NC-stateless becomes significant, because I²NC-stateless does not rely on the knowledge of the decoding buffers of their neighbors, but only on the link loss rates for inter-session NC.

Fig. 6(b) shows the results for the cross topology. The improvement of TCP+I²NC-stateless is higher as compared to the X topology, because there are more NC opportunities here

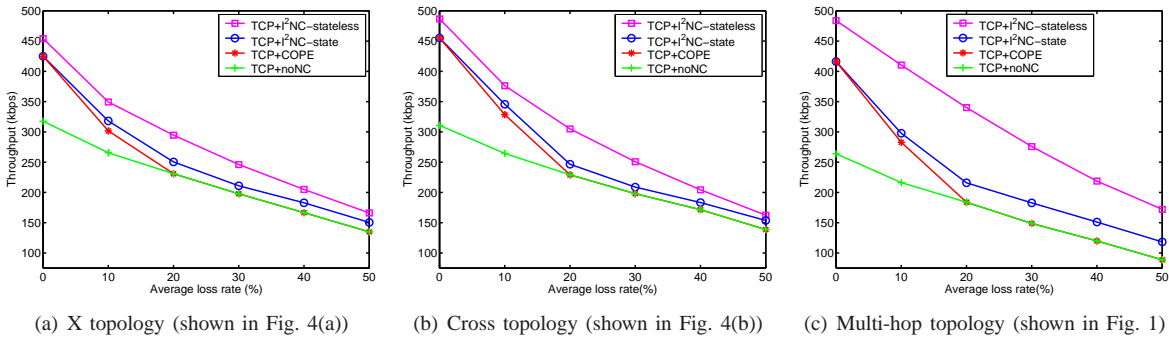


Fig. 6. Total TCP throughput vs. average loss rate (for ease of presentation, the same loss rate is assumed on all links) in three different topologies.

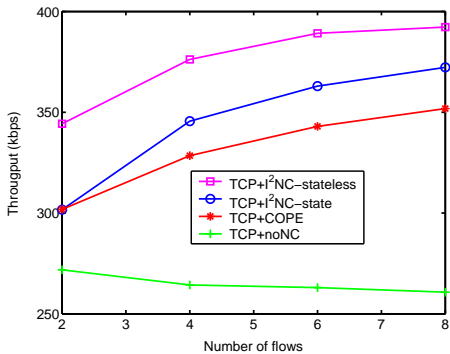


Fig. 7. Wheel topology shown in Fig. 4(c) with increasing number of flows. Loss rate on all links is set to 10%.

for I^2NC -stateless to exploit. We also performed simulations with increasing number of flows, (*i.e.*, nodes in this topology); the details are provided later in this section.

Fig. 6(c) presents the results for the multi-hop topology in Fig. 1. The improvement of $TCP+I^2NC$ -state is higher than in the X and cross topologies, especially at higher loss rates. This is because intra-session coding, employed by I^2NC -state, reduces the dependency on link level ARQ. More specifically, in this multi-hop topology, the end-to-end residual loss rate increases with the number of hops. Intra-session NC overcomes this, thus increasing TCP throughput. The improvement of I^2NC -stateless is even more significant for this topology, because the benefit of eliminating ACKs is more pronounced with larger number of hops.

We also performed simulations with increasing number of flows, *i.e.*, nodes in wheel topology in Fig. 4(c). It is seen in Fig. 7 that the total throughput achieved by NC schemes increases with the increasing number of flows. When the number of flows increases, the probability of NC at the intermediate node I increases. More NC opportunities leads to higher throughput.

UDP traffic. We repeated the simulations for the three topologies for the case that there is loss over all links. The results are presented in Fig. 8.

Fig. 8(a) presents the results for the X topology. The improvement of $UDP+I^2NC$ -stateless is up to 60% as compared to $UDP+noNC$. This is significantly higher than the improvement of $TCP+I^2NC$ -stateless and the optimal scheme (in which the improvement is 33.3%). The reason is the MAC

gain as explained in [2].¹² We present the results for the load at which the system saturates. At this load, $UDP+noNC$ is already saturated, several packets are dropped from the buffers, and they do not arrive to their receivers. This reduces the throughput of $noNC$, while NC schemes still handle the traffic created by the load. Notice that even at 50% loss rate, $UDP+I^2NC$ -stateless improves over $UDP+noNC$ by 40%, which is significant.

Fig. 8(b) presents the results for the cross topology. In this topology, the improvement of NC is very large. When there is no loss, the improvement is around 250%. The effectiveness of $UDP+I^2NC$ -stateless is also significant in this topology: at 50% loss rate the improvement of $UDP+I^2NC$ -stateless over $UDP+noNC$ is 70%.

Fig. 8(c) presents the results for multi-hop topology. We see similar behavior as observed by Figs. 8(b) and (c). However, the improvement of $UDP+I^2NC$ -stateless is larger in this topology, because the benefit of eliminating ACKs is more pronounced with larger number of hops.

C. Numerical Results

We consider the X and cross topologies shown in Figs. 4(a) and 4(b). In the X topology, A_1 transmits packets to A_2 via I with rate x_1 , and B_1 transmits packets to B_2 via I with rate x_2 . In the cross topology, A_1 transmits packets to A_2 with rate x_1 , A_2 transmits packets to A_1 with rate x_2 , B_1 transmits packets to B_2 with rate x_3 , and B_2 transmits packets to B_1 with rate x_4 . All transmissions are via I . In both topologies, the data rate of each link is set to 1 packet/transmission. We compare our schemes I^2NC -state and I^2NC -stateless with $noNC$ which is also formulated in a network utility maximization framework without any NC constraints.

Fig. 9 shows the total throughput; $x_1 + x_2$ for X topology. Fig. 9(a) shows the results when there is loss on $A_1 - B_2$. It is seen that the throughput of $noNC$ is flat with increasing

¹²The MAC gain observed with UDP flows when NC is used can be summarized as follows. When NC is employed, the coded wireless network can handle larger amount of load as compared to its uncoded counterpart. Therefore, when coded system saturates at some load level, uncoded system can not handle this level of load. Thus, several packets are dropped from output queues at each node in the system. Some of these packets may be dropped from intermediate packets. In this case, resources (bandwidth in our case) to transmit these packets (which will be eventually dropped) is wasted. Therefore, the gap between the achieved throughput level of coded and uncoded systems becomes significant.

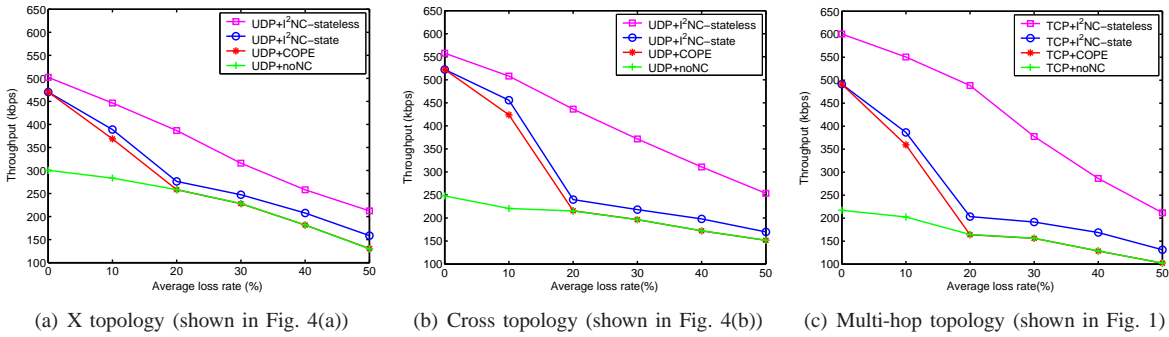


Fig. 8. Total UDP throughput vs. average loss rate (the same loss rate is assumed on all links) in three different topologies.

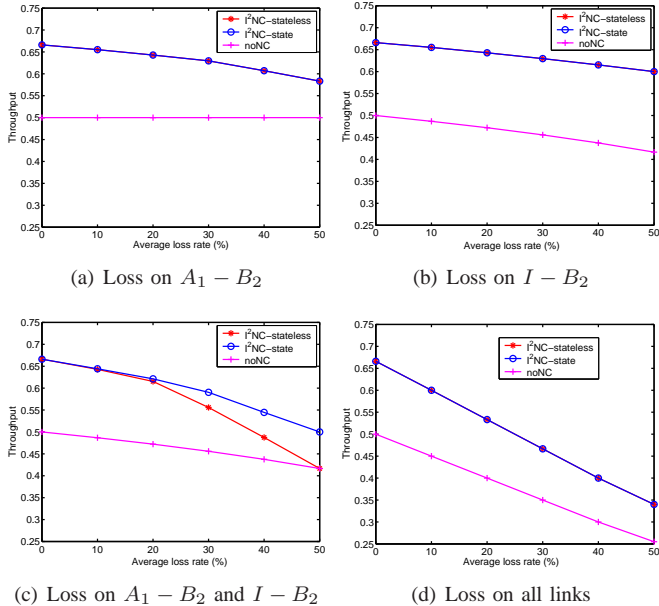


Fig. 9. X topology. Throughput vs. loss rate (the same loss rate is assumed on $A_1 - B_2$ and $I - B_2$ in (c), and the same loss rate is assumed on all links in (d)).

loss rate, because it is not affected by the loss rate on the overhearing link. I²NC-state and I²NC-stateless improve over noNC, because they exploit NC benefit. When the loss rate increases, the improvement reduces, because B_2 overhears only part of the data transmitted by A_1 . Although the improvement decreases with increasing loss rate, it is still significant, *e.g.*, 16.6% at 50% loss rate. Note that Fig. 9(a) is the counterpart of the simulation results presented in Fig. 5(a). It is seen that TCP+I²NC-stateless in Fig. 5(a) shows similar performance as I²NC-stateless in Fig. 9(a). This shows the effectiveness of I²NC-stateless in a realistic simulation environment.

Fig. 9(b) shows the results when there is loss on $I - B_2$. It is seen that I²NC-state and I²NC-stateless improve over noNC significantly at all loss rates. It is also interesting to note that at 50% loss rate, I²NC-state and I²NC-stateless improve over noNC by 44% which is even higher than in the no loss case (33%). The reason is in the following. In the optimal solution, the throughput values are $x_1 = 0.4$ and $x_2 = 0.2$. In this case, in the downlink $I - B_2$, data part of x_2 with rate 0.2 and the parity part with rate 0.2 (considering loss rate 50%) are combined with x_1 . This means that our schemes

combine both parity and data parts of a flow with other flows and this improves the throughput significantly. This is one of the important contributions of I²NC.

Fig. 9(c) shows the results when there is loss on links $A_1 - B_2$ and $I - B_2$. It is seen that I²NC-state improves the throughput significantly while the improvement of I²NC-stateless reduces to 0 with increasing loss rate. The reason is that, I²NC-stateless is a more conservative scheme as compared to I²NC-state in the sense that it eliminates the perfect knowledge on antidotes. Yet, it still improves the throughput significantly, *e.g.*, it improves over noNC by 22% at 30% loss rate.

Fig. 9(d) shows the results when there is loss on all links. It is seen that I²NC-state and I²NC-stateless improve over noNC significantly at all loss rates. Note that throughput of I²NC-stateless reduces to that of noNC at 50% loss rate in Fig. 9(c). The reader might wonder why we do not see such behavior in Fig. 9(d). The reason is that since there is loss over link $A_1 - I$ as well as $A_1 - B_2$, the number of parities added by A_1 to correct losses over link $A_1 - I$ also increases the number of overheard packets at B_2 . Therefore, I²NC-stateless does not add redundancy at node I for both $A_1 - B_2$ and $I - B_2$ as in Fig. 9(c), but adds redundancy only for loss on link $I - B_2$. This improves the performance of I²NC-stateless. Note that the counterpart of these results are presented in Fig. 6(a). It is seen that the throughput improvement of I²NC-stateless over noNC at 50% loss rate is around 30% in Fig. 9(d). As compared to this, the improvement of TCP+I²NC-stateless over noNC is limited in Fig. 6(a), because, in simulations, the block size is limited and fixed, and the scheduling is not perfect (we consider IEEE 802.11). Yet, the throughput improvement of TCP+I²NC-stateless over noNC is around 20% in Fig. 6(a), which is significant.

Fig. 10 shows the total throughput; $x_1 + x_2 + x_3 + x_4$ for I²NC-state, I²NC-stateless and noNC for the cross topology shown in Fig. 4(b) for different loss patterns. It is seen that the results are similar to the ones in Fig. 9. One difference is that the throughput improvement of NC schemes is higher, *i.e.*, up to 80%, because there are more NC opportunities in the cross topology.

VII. CONCLUSION

In this paper, we proposed I²NC: a one-hop intra- and inter-session network coding approach for wireless networks. I²NC

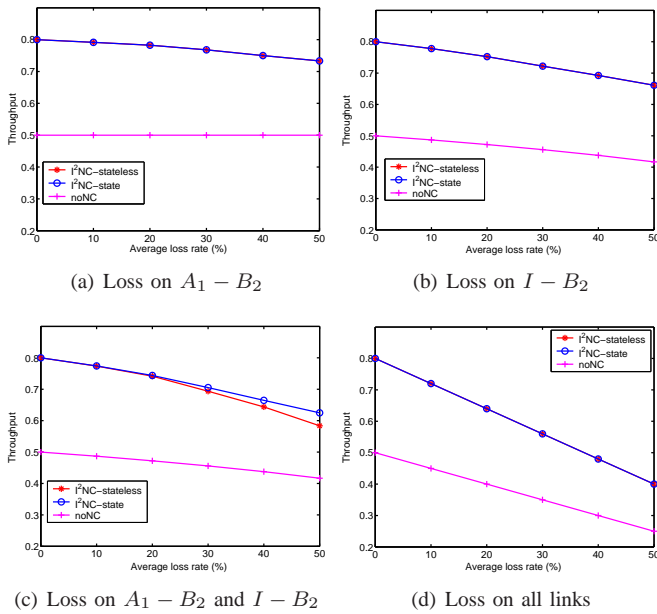


Fig. 10. Cross topology. Throughput vs. loss rate (the same loss rate is assumed on $A_1 - B_2$ and $I - B_2$ in (c), and the same loss rate is assumed on all links in (d)).

builds on and improves COPE in two aspects: it is resilient to loss and it does not need to rely on the exact knowledge of the state of the neighbors. Our design is grounded on a NUM formulation and its solution. Simulations in GloMoSim demonstrate significant throughput gain of our approach compared to no network coding and COPE.

REFERENCES

- [1] 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.
- [2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *IEEE Trans. on Networking*, vol. 16(3), June 2008.
- [3] 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.
- [4] 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.
- [5] S. Omiwade, R. Zheng, and C. Hua, "Butterflies in the mesh: lightweight localized wireless network coding," in *Proc. of NetCod*, Lausanne, Switzerland, Jan. 2008.
- [6] 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.
- [7] M. Chiang, S. T. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: a mathematical theory of network architectures," in *Proceedings of the IEEE*, vol. 95(1), Jan. 2007.
- [8] GloMoSim Version 2.0, "Global Mobile Information Systems Simulation Library," available at pcl.cs.ucla.edu/projects/gloimosim/.
- [9] 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.
- [10] 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.
- [11] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: the case for coding-aware routing," in *Proc. of Infocom*, Anchorage, AK, May 2007.
- [12] Q. Dong, J. Wu, W. Hu, and J. Crowcroft, "Practical network coding in wireless networks," in *Proc. of MobiCom*, Montreal, Canada, Sept. 2007.

- [13] F. Zhao and M. Medard, "On analyzing and improving COPE performance," in *Proc. of ITA*, San Diego, CA, Feb. 2010.
- [14] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation," in *Proc. of Sigmetrics*, Annapolis, MD, June 2008.
- [15] J. Y. Lee, W. J. Kim, J. Y. Baek, and Y. J. Suh, "A wireless network coding scheme with forward error correction code in wireless mesh networks," in *Proc. of Globecom*, Honolulu, HI, Dec. 2009.
- [16] K. Ronasi, A. H. Mohsenian-Rad, V. W. S. Wong, S. Gopalakrishnan, and R. Schober, "Reliability-based rate allocation in wireless inter-session network coding systems," in *Proc. of Globecom*, Honolulu, HI, Dec. 2009.
- [17] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proc. of ACM SIGCOMM*, Portland, OR, Sept. 2004.
- [18] C. Steger, P. Radosavljevic, and J. P. Frantz, "Performance of IEEE 802.11b wireless LAN in an emulated mobile channel," in *Proc. of VTC*, Orlando, FL, Oct. 2003.
- [19] O. Tickoo, V. Subramanian, S. Kalyanaraman, and K. K. Ramakrishnan, "LT-TCP: End-to-end framework to improve TCP performance over networks with lossy channels," in *Proc. of IWQoS*, Passau, Germany, June 2005.
- [20] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, and J. Barros, "Network coding meets TCP," in *Proc. of Infocom*, Rio de Janeiro, Brazil, April 2009.
- [21] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I²NC: Intra- and Inter-Session Network Coding for Unicast Flows in Wireless Networks," in *Proc. of Infocom*, Shanghai, China, April 2011.
- [22] P. Gupta and P. R. Kumar, "The capacity of wireless networks," in *IEEE Trans. on Information Theory*, vol. 46(2), March 2000.
- [23] P. A. Chou and Y. Wu, "Network coding for the Internet and wireless networks," in *IEEE Signal Proc. Magazine*, vol. 24(5), Sept. 2007.
- [24] L. Chen, T. Ho, S. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proc. of Infocom*, Anchorage, AK, May 2007.
- [25] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," in *IEEE Trans. on Information Theory*, vol. 52(6), June 2006.
- [26] R. Ahlswede, N. Cai, S. Y. R. Li, R. W. Yeung, "Network information flow," in *IEEE Trans. on Information Theory*, vol. 46(4), July 2000.
- [27] R. Koetter, M. Médard, "An algebraic approach to network coding," in *IEEE/ACM Trans. on Networking*, vol. 11(5), Oct. 2003.
- [28] T. Ho and D. S. Lun, "Network coding: an introduction," Cambridge University Press, Cambridge, U.K., April 2008.
- [29] V. Sharma, K. K. Ramakrishnan, K. Kar, and S. Kalyanaraman, "Complementing TCP congestion control with forward error correction," in *Proc. of Networking*, Aachen, Germany, May 2009.
- [30] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," in *IEEE Trans. on Information Theory*, vol. 52(10), Oct. 2006.
- [31] Y. Huang, M. Ghaderi, D. Towsley, and W. Gong, "TCP performance in coded wireless mesh networks," in *Proc. of IEEE SECON*, San Francisco, CA, June 2008.
- [32] H. Seferoglu and A. Markopoulou, "Delay-optimized network coding for video streaming over wireless networks," in *Proc. of ICC*, South Africa, May 2010.
- [33] R. Srikant, "The mathematics of internet congestion control," Birkhauser, 2004.
- [34] S. H. Low, "A duality model of TCP and queue management algorithms," in *IEEE/ACM Transactions on Networking*, vol. 11(4), Aug. 2003.
- [35] S. H. Low, J. Doyle, and F. Paganini, "Internet congestion control," in *IEEE Control Syst. Mag.*, vol. 21(1), Feb. 2002.
- [36] L. Chen, T. Ho, M. Chiang, S. H. Low, and J. C. Doyle, "Congestion control for multicast network coding," <http://www.princeton.edu/~chiangm/netcod.pdf>.
- [37] H. K. Khalil, "Nonlinear Systems," Prentice-Hall, 1996.

APPENDIX A: CONVERGENCE ANALYSIS

In this section, we analyze the convergence of the distributed solution of the NUM problem, given in Section IV. First, we provide a proof of convergence, and then present some numerical calculations to verify the convergence.

A. Proof of Convergence

Let us first consider the optimality conditions below. Note that x_s^* , $\alpha_{h,k}^*$, $\tau_{h,k}^*$, and $q_{h,k}^*$ are the optimal values.

$$x_s^* = (U_s^*)^{-1} \left(\sum_{i \in \mathcal{P}_s} \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} H_{h,k}^{s^*} \alpha_{h,k}^{s^*} \left(\frac{q_{h,k}^*}{1-\rho_h^*} + \sum_{s' \in \mathcal{S}_k - \{s\}} q_{h,k}^{s',s} \right) \right), \quad (12)$$

$$\tau_{h,k}^* = \arg \max_{\sum_{h \in \mathcal{C}_q} \sum_{k \in \mathcal{K}_h} \tau_{h,k} \leq \gamma} \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} Q_{h,k}^* \tau_{h,k}, \quad (13)$$

$$q_{h,k}^* \left(\frac{H_{h,k}^{s^*} \alpha_{h,k}^{s^*} x_s^*}{1-\rho_h^*} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - R_h \tau_{h,k}^* \right) = 0, \quad q_{h,k}^* \geq 0, \quad (14)$$

$$\frac{H_{h,k}^{s^*} \alpha_{h,k}^{s^*} x_s^*}{1-\rho_h^*} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} \leq R_h \tau_{h,k}^*, \quad \sum_{h \in \mathcal{C}_q} \sum_{k \in \mathcal{K}_h} \tau_{h,k}^* \leq \gamma, \quad (15)$$

$$\alpha_{h,k}^* = \arg \min_{\sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \alpha_{h,k}^* = 1} \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \alpha_{h,k}^* H_{h,k}^{s^*} Q_{h,k}^*, \quad (16)$$

We consider a similar Lyapunov function considered in [36]; $V(q, \tau, \alpha) = \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} \frac{(q_{h,k}^s - q_{h,k}^*)^2}{2\gamma_h} + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \frac{(\tau_{h,k} - \tau_{h,k}^*)^2}{2\varepsilon_{\mathcal{C}_q}} + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s^* \frac{(\alpha_{h,k}^s - \alpha_{h,k}^*)^2}{2\kappa_i}$.

The derivative of the Lyapunov function with respect to the Lagrange multipliers is expressed as; $\dot{V}(q, \tau, \alpha) = \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left[\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right] + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} (\tau_{h,k} - \tau_{h,k}^*) [Q_{h,k} - E_{\mathcal{C}_q}[Q]] + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s^* (\alpha_{h,k}^s - \alpha_{h,k}^*) [E_i[Q] - H_{h,k}^s Q_{h,k}^*]$.

By using the definition of the function $[b]_z^\pm$, and considering that Since $E_{\mathcal{C}_q}[Q]$ is the minimal $\bar{Q}_{\mathcal{C}_q}(t)$ and $E_i[Q]$ is the maximal $\bar{Q}_i(t)$, the following holds; $\dot{V}(q, \tau, \alpha) \leq \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right) + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} (\tau_{h,k} - \tau_{h,k}^*) (Q_{h,k} - E_{\mathcal{C}_q}[Q]) + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s^* (\alpha_{h,k}^s - \alpha_{h,k}^*) (E_i[Q] - H_{h,k}^s Q_{h,k}^*)$.

Since $Q_{h,k} \geq Q_{h,k} - E_{\mathcal{C}_q}[Q]$ and $H_{h,k}^s Q_{h,k}^* \geq H_{h,k}^s Q_{h,k}^* - E_i[Q]$, the following holds; $\dot{V}(q, \tau, \alpha) \leq \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right) + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} (\tau_{h,k} - \tau_{h,k}^*) Q_{h,k} + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s^* (\alpha_{h,k}^s - \alpha_{h,k}^*) H_{h,k}^s Q_{h,k}^*$.

Substituting $Q_{h,k} = R_h \sum_{s \in \mathcal{S}_k} q_{h,k}^s$ and

$$Q_{h,k}^s = \frac{q_{h,k}^s}{1-\rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} q_{h,k}^{s',s} \rho_{h,k}^{s',s}$$

to the above inequality; $\dot{V}(q, \tau, \alpha) \leq \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right) + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} (\tau_{h,k} - \tau_{h,k}^*) R_h \sum_{s \in \mathcal{S}_k} q_{h,k}^s + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s^* (\alpha_{h,k}^s - \alpha_{h,k}^*) H_{h,k}^s \left(\frac{q_{h,k}^s}{1-\rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} q_{h,k}^{s',s} \rho_{h,k}^{s',s} \right)$.

When we arrange the terms in the above inequality by adding and removing terms, we have;

$$\begin{aligned} \dot{V}(q, \tau, \alpha) &\leq \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} \right. \\ &+ \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - \frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} \\ &- \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} \left. \right) + \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \\ &\left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} + \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \right) \\ &+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} R_h (\tau_{h,k} - \tau_{h,k}^*) q_{h,k}^s \\ &+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s^* (\alpha_{h,k}^s - \alpha_{h,k}^*) \left(\frac{H_{h,k}^s q_{h,k}^s}{1-\rho_h^s} \right. \\ &+ \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} q_{h,k}^{s',s} \rho_{h,k}^{s',s} \left. \right) \\ &= \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left(\frac{H_{h,k}^s \alpha_{h,k}^s (x_s - x_s^*)}{1-\rho_h^s} \right. \\ &+ \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} \rho_{h,k}^{s,s'} (x_{s'} - x_{s'}^*) \left. \right) \\ &+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} \right. \\ &+ \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - \frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} \\ &- \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} \left. \right) \\ &+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^*) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s^*}{1-\rho_h^s} \right. \\ &+ \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'}^* \rho_{h,k}^{s,s'} - R_h \tau_{h,k} \left. \right) \\ &+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s^* (\alpha_{h,k}^s - \alpha_{h,k}^*) \left(\frac{H_{h,k}^s q_{h,k}^s}{1-\rho_h^s} \right. \\ &+ \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} q_{h,k}^{s',s} \rho_{h,k}^{s',s} \left. \right) \end{aligned}$$

$$\begin{aligned}
&= \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s (x_s - x_s^*)}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} \rho_{h,k}^{s,s'} (x_{s'} - x_{s'}^*) \right) \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - \frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right) \\
&- \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} \rho_{h,k}^{s,s'} \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - R_h \tau_{h,k}^* \right) \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) R_h (\tau_{h,k}^* - \tau_{h,k}) \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} R_h (\tau_{h,k} - \tau_{h,k}^*) q_{h,k}^s \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s (\alpha_{h,k}^{*s} - \alpha_{h,k}^s) \left(\frac{H_{h,k}^s q_{h,k}^s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} q_{h,k}^{s'} \rho_{h,k}^{s',s} \right) \\
&= \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s (x_s - x_s^*)}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} \rho_{h,k}^{s,s'} (x_{s'} - x_{s'}^*) \right) \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - R_h \tau_{h,k}^* \right) \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} q_{h,k}^s R_h (\tau_{h,k} - \tau_{h,k}^*) \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - \frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right) \\
&- \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} \rho_{h,k}^{s,s'} \\
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s (\alpha_{h,k}^{*s} - \alpha_{h,k}^s) \left(\frac{H_{h,k}^s q_{h,k}^s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} q_{h,k}^{s'} \rho_{h,k}^{s',s} \right)
\end{aligned}$$

$$\begin{aligned}
\dot{V}(q, \tau, \alpha) &\leq \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s (x_s - x_s^*)}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} \rho_{h,k}^{s,s'} (x_{s'} - x_{s'}^*) \right) \tag{17}
\end{aligned}$$

$$\begin{aligned}
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - R_h \tau_{h,k}^* \right) \tag{18}
\end{aligned}$$

$$\begin{aligned}
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} q_{h,k}^s R_h (\tau_{h,k} - \tau_{h,k}^*) \tag{19}
\end{aligned}$$

$$\begin{aligned}
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s H_{h,k}^s \left(\frac{q_{h,k}^s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} \rho_{h,k}^{s',s} \right) (\alpha_{h,k}^{*s} - \alpha_{h,k}^s) \tag{20}
\end{aligned}$$

$$\begin{aligned}
&+ \sum_{s' \in \mathcal{S}_k - \{s\}} \rho_{h,k}^{s',s} (\alpha_{h,k}^{*s} - \alpha_{h,k}^s) \tag{21}
\end{aligned}$$

$$\begin{aligned}
\dot{V}(q, \tau, \alpha) &\leq \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s (x_s - x_s^*)}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} \rho_{h,k}^{s,s'} (x_{s'} - x_{s'}^*) \right) \tag{22}
\end{aligned}$$

$$\begin{aligned}
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} (q_{h,k}^s - q_{h,k}^{*s}) \left(\frac{H_{h,k}^s \alpha_{h,k}^s x_s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} H_{h,k}^{s'} \alpha_{h,k}^{s'} x_{s'} \rho_{h,k}^{s,s'} - R_h \tau_{h,k}^* \right) \tag{23}
\end{aligned}$$

$$\begin{aligned}
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} q_{h,k}^s R_h (\tau_{h,k} - \tau_{h,k}^*) \tag{24}
\end{aligned}$$

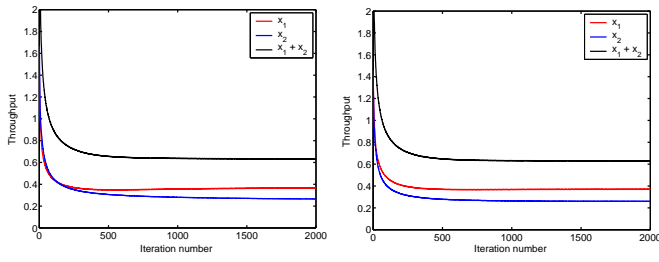
$$\begin{aligned}
&+ \sum_{h \in \mathcal{A}} \sum_{k \in \mathcal{K}_h} \sum_{s \in \mathcal{S}_k} x_s H_{h,k}^s \left(\frac{q_{h,k}^s}{1 - \rho_h^s} \right. \\
&+ \left. \sum_{s' \in \mathcal{S}_k - \{s\}} \rho_{h,k}^{s',s} \right) (\alpha_{h,k}^{*s} - \alpha_{h,k}^s) \tag{25}
\end{aligned}$$

Since the marginal utility $U_s'(\cdot)$ is a decreasing function, its inverse, *i.e.*, the Eq. (22) is less than 0. Due to the optimality condition in Eq. (13) and Eq. (15), Eq. (23) is less than 0. Due to the optimality condition in Eq. (14), Eq. (24) is less than 0. Due to the optimality condition in Eq. (16), Eq. (25) is less than 0. Thus, $\dot{V}(q, \tau, \alpha) \leq 0$. This implies the convergence of our solutions, [36], [37].

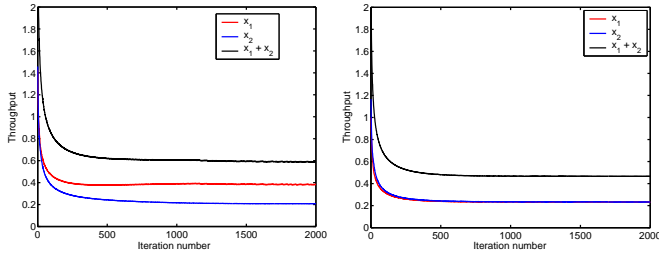
B. Numerical Results

We consider again the X and cross topologies shown in Figs. 4(a) and 4(b). In the X topology, A_1 transmits packets to A_2 via I with rate x_1 , and B_1 transmits packets to B_2 via I with rate x_2 . In the cross topology, A_1 transmits packets to A_2 with rate x_1 , A_2 transmits packets to A_1 with rate x_2 , B_1 transmits packets to B_2 with rate x_3 , and B_2 transmits packets to B_1 with rate x_4 . All transmissions are via I . In both topologies, the data rate of each link is set to 1 packet/transmission and the loss rate is set to 30%.

In Figs. 11 and 12, we present the throughput vs. the iteration number for the X topology at different loss patterns for I²NC-state and I²NC-stateless, respectively. Each figure

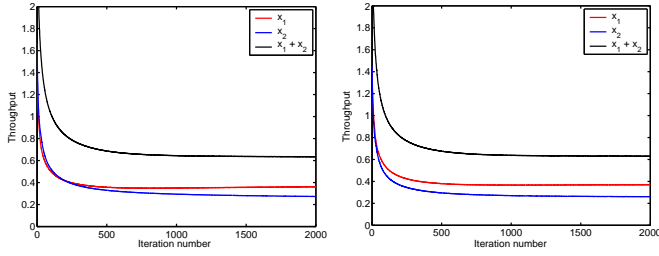


(a) Loss only on overheard link $A_1 - B_2$ (b) Loss only on direct link $I - B_2$

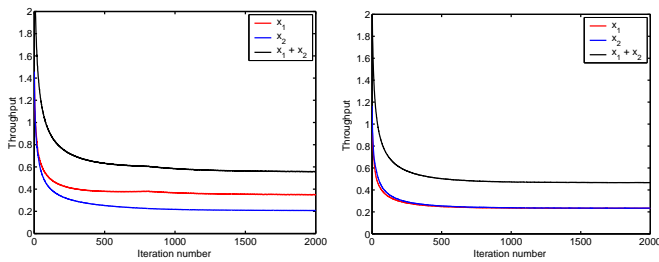


(c) Loss on links $A_1 - B_2$ and $I - B_2$ (d) Loss on all links

Fig. 11. X topology. Convergence of x_1 , x_2 , and $x_1 + x_2$ for I^2NC -state. Loss rate is 30%.



(a) Loss only on overheard link $A_1 - B_2$ (b) Loss only on direct link $I - B_2$

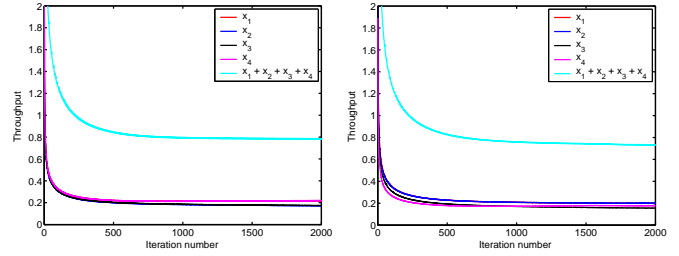


(c) Loss on links $A_1 - B_2$ and $I - B_2$ (d) Loss on all links

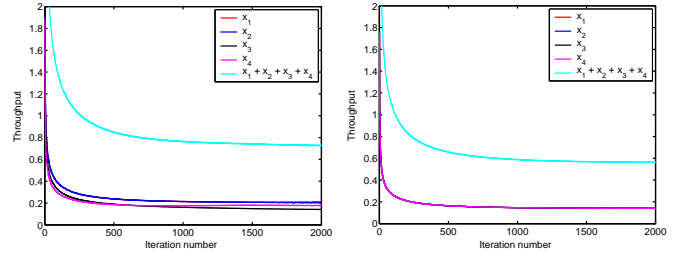
Fig. 12. X topology. Convergence of x_1 , x_2 , and $x_1 + x_2$ for I^2NC -stateless. Loss rate is 30%.

shows the convergence of x_1 , x_2 , and $x_1 + x_2$ to their optimum values. *E.g.*, $x_1 + x_2$ converges to its optimum value 0.59 in Fig. 11(c) and $x_1 + x_2$ converges to its optimum value 0.55 in Fig. 12(c).

Fig. 13 and 14 present the throughput vs. the iteration number for the cross topology at different loss patterns for I^2NC -state and I^2NC -stateless, respectively. We see similar convergence results. Specifically, each flow rate, x_1 , x_2 , x_3 , x_4 , and the total rate converge to their optimum values.

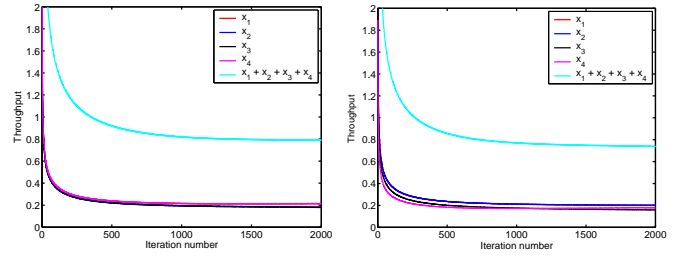


(a) Loss only on overheard link $A_1 - B_2$ (b) Loss only on direct link $I - B_2$

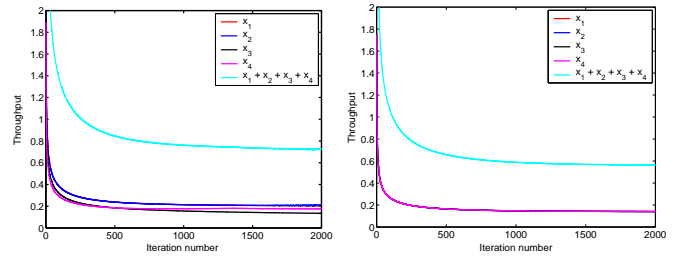


(c) Loss on links $A_1 - B_2$ and $I - B_2$ (d) Loss on all links

Fig. 13. Cross topology. Convergence of x_1 , x_2 , x_3 , x_4 , and $x_1 + x_2 + x_3 + x_4$ for I^2NC -state. Loss rate is 30%.



(a) Loss only on overheard link $A_1 - B_2$ (b) Loss only on direct link $I - B_2$



(c) Loss on links $A_1 - B_2$ and $I - B_2$ (d) Loss on all links

Fig. 14. Cross topology. Convergence of x_1 , x_2 , x_3 , x_4 , and $x_1 + x_2 + x_3 + x_4$ for I^2NC -stateless. Loss rate is 30%.