

# Capacity Regions for Multiple Unicast Flows using Inter-session Network Coding

(Invited paper)

Abdallah Khreishah  
School of ECE  
Purdue University  
akhreish@purdue.edu

Chih-Chun Wang  
School of ECE  
Purdue University  
chihw@purdue.edu

Ness B. Shroff  
Departments of ECE and CSE  
The Ohio State University  
shroff@ece.osu.edu

## ABSTRACT

In this paper we consider the problem of transmitting packets through multiple unicast sessions when inter-session linear network coding is permitted. We propose and formulate two new capacity regions for this problem. Distributed rate allocation algorithms and coding schemes are also provided for these capacity regions. The two capacity regions are compared with other known capacity regions in the literature to represent their benefits in terms of throughput, fairness, complexity, and ease of implementation.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Distributed Networks, Network Communications*; E.4 [Coding and Information Theory]: Formal Models of Communication

## General Terms

Algorithms, Performance, Theory

## Keywords

Capacity region, inter-session network coding, multiple-unicast-sessions problem, rate control, distributed algorithm, coding scheme, fairness

## 1. INTRODUCTION

Maximizing the network throughput and achieving fairness among network users are some of the most important challenges in designing efficient networks protocols. More recently, a new area of research has emerged termed network coding that has the potential to increase the achievable throughput by mixing different flows at intermediate nodes [1]. Intra-session multicast network coding where coding is restricted to packets of the same session has been studied extensively in the literature (see, for example [1, 16, 13, 8, 22]).

Network coding can also be performed between packets of different sessions referred to as *inter-session network coding*, as in the butterfly network in Figure 1(a). As is the conven-

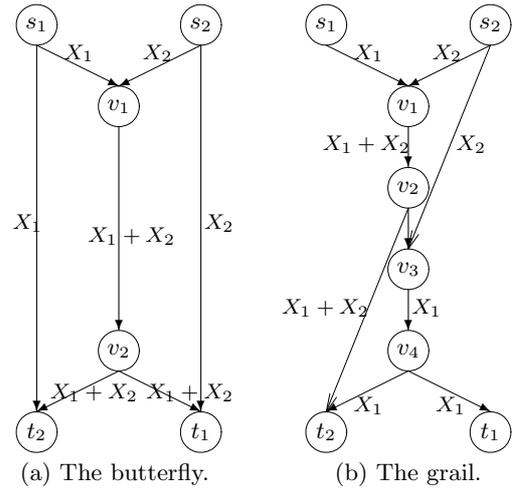


Figure 1: Two examples of graphs benefiting from network coding and their corresponding network coding schemes. Let  $X_1$  and  $X_2$  represent bits being transmitted from sources 1 and 2, respectively. The “+” sign stands for the exclusive or (XOR) coding operation.

tion in the network coding literature, we use the term “routing” to describe the solution for which no network coding is used. In the simple butterfly configuration of Figure 1(a), we assume that each link can sustain a throughput of at most 1 packet per second (in subsequent discussions we drop the units). Senders  $s_1$  and  $s_2$  want to send packets to receivers  $t_1$  and  $t_2$ , respectively. For this topology, if only routing solutions are permitted, we could achieve strict fairness (i.e., transmissions at equal rates from both sources) with the rate for each sender being no higher than 0.5. However, using network coding, it can be easily seen that we can sustain unit rates for both senders simultaneously. The coded messages at each link that achieve the afore mentioned rates using network coding is illustrated in Figure 1(a).

Inter-session network coding for multiple unicast/multicast sessions has been relatively less studied. In [3] it was shown that *linear* network coding is no longer capable of achieving the optimal capacity region in the multiple unicast/multicast case in contrast with the single multicast session. Since then, many studies have targeted suboptimal solutions to the multi-session unicast problem using network coding. For wireless networks, the nature of links further enriches the coding possibilities, as broadcasting can be achieved without power penalty. Therefore, a much smaller number of transmissions is necessary than its wired counterpart. Based on

the broadcasting nature of wireless networks, opportunistic exclusive OR (XOR) coding was introduced in [9], and further improved and analyzed in [20]. The  $TRCLKM$  capacity region was introduced in [23], which is a capacity region for which only XOR coding is permitted.  $TRCLKM$  region captures any possible butterfly structure in the network. Two distributed algorithms for the  $TRCLKM$  region using back-pressure techniques were provided in [4] and [7]. In this paper, we utilize the results in our previous works [24] and [25] and introduce two capacity regions for the multiple unicast problem. One of these capacity regions is based on finding good paths in the network, while the other is based on finding specific structures in the network. Distributed implementations for both capacity regions are also provided in this paper. Our results show that the path-based approach results in an algorithm with reduced complexity and is more suitable for online implementation.

The rest of the paper is organized as follows. In Section 2, we describe the system settings followed by short review of the traditional routing region in Section 3. In Section 4, we review the description, formulation, and distributed implementation of the  $TRCLKM$  region. We introduce the  $WS$  capacity region in Section 5, by describing it, formulating it, and providing a distributed rate control algorithm and distributed coding scheme for it. In Section 6, the  $\mathcal{I} - TRCLKM$  capacity region is described and formulated. We also provide a distributed implementation of the  $\mathcal{I} - TRCLKM$  in Section 6. A comparison of the different types of capacity regions is in Section 7. We conclude the paper in Section 8.

## 2. SYSTEM SETTINGS

We consider a general directed, acyclic, graphs (DAG)  $G = (V, E)$ , where  $V$  and  $E$  are the set of all nodes and links, respectively. We denote the capacity of each link  $n$  by  $C_n$ . The problem is defined by the set of tuples  $(s_i, t_i, R_i)$   $i \in 1, 2, \dots, I$ . Session  $i$  is represented by the tuple with index  $i$ , where  $s_i$  and  $t_i$  are the source and sink nodes, respectively, for session  $i$ . Packets transmitted through session  $i$  are independent of those transmitted through session  $j$ ,  $\forall i \neq j$ .  $R_i$  is the transmission rate supported between  $s_i$  and  $t_i$ . For any link  $n = (u, v)$ , let  $\text{tail}(n)=u$  and  $\text{head}(n)=v$ .  $\Gamma_I(v)$  is used to denote the set of links with head  $v$  and  $\Gamma_O(v)$  is used to denote the set of links with tail  $v$ . For the following sections different achievable capacity regions will be defined, and we begin by describing the capacity region that can be achieved without using network coding.

## 3. THE ROUTING REGION

The term routing region refers to the rate region achieved by forwarding and duplicating without any coding operations. This region has been studied extensively for wireline and wireless networks as in [5, 21]. Let  $x_n(i)$  represents the rate for session  $i$  at link  $n$ . For wireline networks the following constraints describe the problem.

$$\sum_{n \in \Gamma_O(g)} x_n(i) - \sum_{n \in \Gamma_I(g)} x_n(i) = \begin{cases} R_i & g = s_i \\ -R_i & g = t_i \\ 0 & \text{else} \end{cases} \quad (1)$$

$$\sum_{i=1}^I x_n(i) \leq C_n \quad \forall n \in E \quad (2)$$

Similar formulations can be given for wireless networks. Many distributed algorithms have been developed for this region (see, for example, [10, 17, 18, 26, 6]). If every source wants

to send at specific rate, then this problem is known as the multi-commodity flow problem, see [19, 14, 2] for further references.

## 4. THE $TRCLKM$ REGION

In this section, we describe the  $TRCLKM$  capacity region introduced in [23], and the distributed implementation of this capacity region in [4, 7].

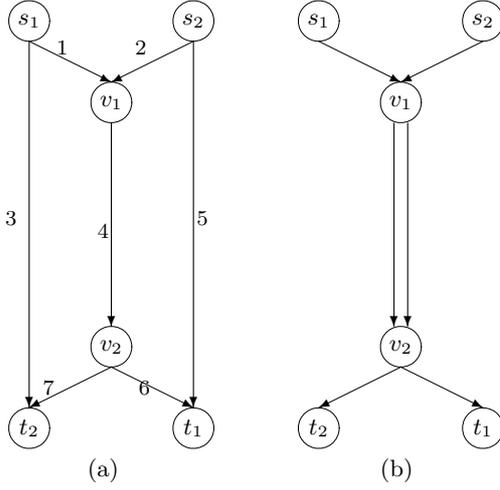
### 4.1 Description and Formulation

The  $TRCLKM$  capacity region uses simple XOR network coding scheme to exploit any butterfly structure in the network. Consider the butterfly network in Figure 2(a), with unit capacity links. Source nodes  $s_1$  and  $s_2$  want to send packets to  $t_1$  and  $t_2$ , respectively. The coding solution in the butterfly topology in Figure 2(a) is, in effect, equivalent to the routing solution in the network in Figure 2(b) where the bottleneck edge is resolved by doubling its capacity (modelled as two parallel edges). Here, links 3 and 5 that carry the remedy packets are used to resolve the bottleneck link 4. The  $TRCLKM$  region resolve any butterfly bottleneck link in the network using the reduction depicted in Figure 2. Let  $x_n(i)$  be the rate for session  $i$  at link  $n$  after resolving the butterfly bottlenecks in the network. For example in Figure 2(a)  $x_n(1)$  equals zero for all  $n$  except  $n = 1, 4, 6$ , and  $x_n(2)$  equals zero for all  $n$  except  $n = 2, 4, 7$ . If no network coding is allowed, edge 4 carries packets of rates  $x_4(1) + x_4(2) = 2$ , which exceeds the capacity  $C_4$ . In the butterfly network in Figure 1(a), when  $X_1$  is XORed with  $X_2$  so that the total rate on edge 4 is 1 instead of two. However, the resulting flow is now *corrupted* as it is a mixture of different packets from  $s_1$  and  $s_2$  respectively. Therefore, it is termed the poisson flow. To be able to decode the poisson flow, a remedy flow should be generated at another node. For the remedy flow to be generated, a remedy request should be sent from  $v_1$  the *encoding* node to  $s_1$  and  $s_2$  the *remedy generating* nodes. The variables  $p_n(i \rightarrow j, u)$ ,  $q_n(\cdot)$ , and  $r_n(\cdot)$  are used to describe the region. These variables represents the poisson, remedy request, and remedy flows at edge  $n$  associated with poisoning session  $j$  data with session  $i$  data at node  $u$ . Figure 3 represents the assignment of these variables on the butterfly topology. The following linear optimization constraints models the  $TRCLKM$  capacity region.

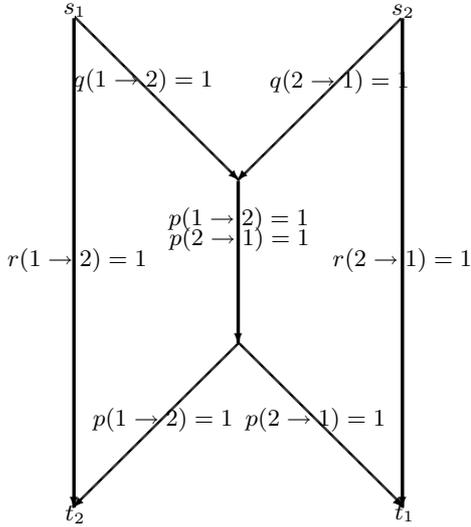
$$\sum_{n \in \Gamma_O(g)} x_n(i) - \sum_{n \in \Gamma_I(g)} x_n(i) = \begin{cases} R_i & g = s_i \\ -R_i & g = t_i \\ 0 & \text{else} \end{cases} \quad \forall i \in \{1, \dots, I\}. \quad (3)$$

$$\begin{aligned} & \sum_{n \in \Gamma_O(g)} r_n(i \rightarrow j, u) - p_n(i \rightarrow j, u) - q_n(i \rightarrow j, u) \\ & = \sum_{n \in \Gamma_I(g)} r_n(i \rightarrow j, u) - p_n(i \rightarrow j, u) - q_n(i \rightarrow j, u) \end{aligned} \quad \forall u, g \in V, \quad i, j \in \{1, \dots, I\}. \quad (4)$$

$$\sum_{n \in \Gamma_I(g)} q_n(i \rightarrow j, u) - \sum_{n \in \Gamma_O(g)} q_n(i \rightarrow j, u) = \begin{cases} \leq 0 & g = u \\ \geq 0 & \text{else} \end{cases} \quad \forall u, g \in V, \quad i, j \in \{1, \dots, I\}. \quad (5)$$



**Figure 2:** Example showing how the  $TRLCM$  region uses remedy packets links to resolve bottleneck in the butterfly structure. Coding solution on (a) is equivalent to routing solution on (b)



**Figure 3:** The assignment of the variables on the  $TRLCM$  formulation on the butterfly topology with unit capacity links.

$$\sum_{n \in \Gamma_I(g)} p_n(i \rightarrow j, u) - \sum_{n \in \Gamma_O(g)} p_n(i \rightarrow j, u) = \begin{cases} \geq 0 & g = u \\ \leq 0 & \text{else} \end{cases} \quad \forall u, g \in V, \quad i, j \in \{1, \dots, I\}. \quad (6)$$

$$p_n(i \rightarrow j, u) = p_n(j \rightarrow i, u) \quad \text{if } n \in \Gamma_O(u) \quad \forall u \in V, \quad i, j \in \{1, \dots, I\}. \quad (7)$$

$$\sum_u \sum_{\substack{i,j \\ i < j}} -\min(p_n(i \rightarrow j, u), p_n(j \rightarrow i, u)) + \sum_{i=1}^I x_n(i) + \sum_u \sum_{\substack{i,j \\ i \neq j}} r_n(i \rightarrow j, u) \leq C_n, \quad \forall n \in E \quad (8)$$

$$x_n(j) - \sum_u \sum_i (p_n(i \rightarrow j, u) + p_n(j \rightarrow i, u)) \geq 0 \quad \forall n \in E, \quad j \in \{1, \dots, I\}. \quad (9)$$

$x_n(\cdot), r_n(\cdot), p_n(\cdot), q_n(\cdot) \geq 0$ .

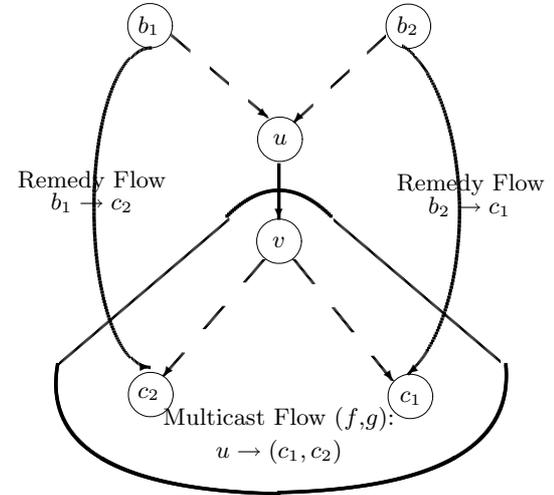
The following theorem, first stated in [23], shows that the constraints in (3)-(9) are sufficient for the existence of inter-session network coding scheme based on finding butterfly structures.

**THEOREM 1.** Consider a network  $G = (V, E)$ , where the capacity of each link  $n$  is  $C_n$ . If the constraints in (3)-(9) are satisfied then there exists an XOR coding scheme based on finding butterfly structures in the network such that the rate supported by each session  $i$  is  $R_i$ .

The proof of this theorem is provided in [23].

## 4.2 The $RSC$ distributed algorithm

Based on back pressure techniques, [4] and [7] develop two distributed algorithms for the  $TRLCM$  region. In this section, we will describe the algorithm in [4]. The  $TRLCM$  region exploits any coding possibility represented by the butterfly structure in the network. In Figure 4<sup>1</sup>, the butterfly structure between sessions  $f$  and  $g$  can be represented by two unicasts flows for remedy packets between  $(b_1 - c_2)$ ,  $(b_2 - c_1)$ , and one multicast flow for poisson packets between node  $u$  and both  $c_1$  and  $c_2$ . The packets are said to be of type  $D$ , where  $D$  is a destination node, if they are part of the multicast flow destined for nodes in  $D$ .  $Q_u^{(d,D)}$  is defined as the queue length at node  $u$  at the beginning of time slot  $t$  for packets of type  $D$  with destination  $d \in D$ . Here, we use  $Q_u^d = Q_u^{(d,\{d\})}$ . The evolution of the queues are described in [4] and omitted here for brevity.



**Figure 4:** Session  $f$  goes from  $b_1$  to  $c_1$  and session  $g$  goes from  $b_2$  to  $c_2$ , both traversing  $u$ . The dashed lines indicate paths composed of multiple links. A decision to perform inter-session coding across flows  $f$  and  $g$  at node  $u$  results in forming two unicasts flows for remedy packets, and one multicast flow.

The Routing-Scheduling-Coding ( $RSC$ ) algorithm as described in [4] works as follows.

*RSC Algorithm:*

For each link  $(u, v)$  the following two sets of weights are

<sup>1</sup>Similar Figure can be found in [4]

computed:

$$\rho_{(u,v)D}[t] \triangleq \sum_{d \in D} (Q_u^{(d,D)}[t] - Q_v^{(d,D)}[t])^+, \quad (10)$$

$$\begin{aligned} & \sigma_{(u,v)}^{(D_1, b_1, c_1), (D_2, b_2, c_2)}[t] \\ & \triangleq \sum_{d \in D_1} (Q_u^{(d, D_1)}[t] - Q_{c_1}^{(d, D_1)}[t])^+ - Q_v^{(c_1, c_1, c_2)}[t] \\ & \quad \sum_{d \in D_2} (Q_u^{(d, D_2)}[t] - Q_{c_2}^{(d, D_2)}[t])^+ - Q_v^{(c_2, c_1, c_2)}[t] \\ & \quad - Q_{b_1}^{c_1}[t] - Q_{b_2}^{c_2}[t], \quad (11) \end{aligned}$$

where  $(\cdot)^+$  is a projection on  $[0, \infty)$ .

Here,  $\rho_{(u,v)D}[t]$  is a differential backlog representing the weight of routing. Also,  $\sigma_{(u,v)}^{(D_1, b_1, c_1), (D_2, b_2, c_2)}[t]$  is a combination of two differential backlogs representing the weight of inter-session network coding, because coding increases the decodability of the poisson packets and increases the traffic for remedy packets at the same time. Let

$$D_{(u,v)}^*[t] \triangleq \arg \max_D \rho_{(u,v)}^D[t],$$

and

$$\begin{aligned} & ((D_1^*, b_1^*, c_1^*), (D_2^*, b_2^*, c_2^*))_{(u,v)}[t] \triangleq \\ & \arg \max_{((D_1, b_1, c_1), (D_2, b_2, c_2))} \sigma_{(u,v)}^{(D_1, b_1, c_1), (D_2, b_2, c_2)}[t]. \end{aligned}$$

Also, let

$$\rho_{(u,v)}^* \triangleq \max_D \rho_{(u,v)}^D[t]$$

and

$$\sigma_{(u,v)}^*[t] \triangleq \max_{((D_1, b_1, c_1), (D_2, b_2, c_2))} \sigma_{(u,v)}^{(D_1, b_1, c_1), (D_2, b_2, c_2)}[t].$$

The final decision at node  $u$  is described as follows.

- *A* : if  $\rho_{(u,v)}^* > \sigma_{(u,v)}^*[t]$ , then no inter-session coding is performed at node  $u$ , and random intra-session coding is performed only within packets of flow  $D_{(u,v)}^*[t]$  as follows: the head of the line packets for those  $d \in D_{(u,v)}^*$  satisfying  $(Q_n^{(d, D_{(u,v)}^*)} - Q_m^{(d, D_{(u,v)}^*)}) > 0$  are multiplied with randomly picked coefficients and added together. Then, the resulting packet is transmitted over link  $(u, v)$  along with the random coefficients.
- *B* : if  $\rho_{(u,v)}^* < \sigma_{(u,v)}^*[t]$ , then inter-session coding is to be performed at node  $u$  across packets of type  $D_1^*$  and  $D_2^*$  through the following steps sequentially:
  - *B1* : *Intra-session Coding*: For each  $i = 1, 2$ , the head of the line packets for those  $d \in D_i^*$  satisfying  $(Q_u^{(d, D_i^*)} - Q_{c_i}^{(d, D_i^*)}) > 0$  are multiplied with randomly picked coefficients and added together. Notice that this operation constitutes the intra-session coding part of the strategy with a different criterion for the choice of participating packets than in the previous case.
  - *B2* : *Inter-session Coding*: The inter-session coding is performed by adding (XORing) the packets generated by the previous intra-session coding operation. Then, the final packet is transmitted over link  $(u, v)$  along with the random coefficients.
  - *B3* : *Remedy Packet Generation*: At nodes,  $b_1^*$  and  $b_2^*$ , generate packets of type  $D_2^*$  and  $D_1^*$ , respectively, and then enqueue them for transmission to  $c_1^*$  and  $c_2^*$ , respectively.

- *C* : if  $\rho_{(u,v)}^* = \sigma_{(u,v)}^*[t]$ , then choose randomly, with equal probabilities, one of the above two modes of operations and implement it.

### 4.3 Another distributed algorithm

In [7] another distributed algorithm for the *TRLCM* region was provided. This algorithm is a non-trivial modification of that for the multi-commodity flow problem in [2] based on back-pressure techniques. This algorithm differs from the *RSC* in the way flows are defined. For example, in the *RSC*, the poisson packets are part of multicast flows which is not the case here. Another difference is that in *RSC* the decoding node is not determined locally as in this algorithm. We remove the details of this algorithm here, because in terms of the comparison in Section 7, it has similar features and performance to the *RSC* algorithm.

## 5. THE *WS* REGION

### 5.1 Description and Formulation

To describe the *WS* region, we first review the results in [24]. In [24], we have studied the problem of network coding with two simple unicast sessions for a finite DAG. The following notations are helpful for the following discussions:  $Q_{v,w}, P_{v,w}$  denote paths from node  $v$  to node  $w$ .  $\mathcal{P}$  is a set of paths, and  $\text{es}_{\mathcal{P}}(n)$  is the number of paths in the set  $\mathcal{P}$  that use link  $n$ . The main results in [24] are:

- *Prelim 1*: For a DAG with unit capacity links and two coexisting unicast sessions between the source-sink pairs  $(s_1, t_1), (s_2, t_2)$ , a network coding scheme that can support unit rates for both sessions exists if and only if one of the following two conditions holds.
  - [Condition 1] There exists a collection  $\mathcal{P}$  of two paths  $P_{s_1, t_1}$  and  $P_{s_2, t_2}$ , such that  $\max_{n \in E} \text{es}_{\mathcal{P}}(n) \leq 1$ .
  - [Condition 2] There exist a collection  $\mathcal{P}$  of three paths  $\{P_{s_1, t_1}, P_{s_2, t_2}, P_{s_2, t_1}\}$ , and a collection  $\mathcal{Q}$  of three paths  $\{Q_{s_1, t_1}, Q_{s_2, t_2}, Q_{s_1, t_2}\}$ , such that  $\max_{n \in E} \text{es}_{\mathcal{P}}(n) \leq 2$  and  $\max_{n \in E} \text{es}_{\mathcal{Q}}(n) \leq 2$ .
- *Prelim 2*: This result is provided in *Corollary 4* in [24]. The result is as follows: Consider  $I$  source-&-sink pairs and each source  $s_i$  would like to transmit  $R_i$  symbols to the corresponding sink  $t_i$  within one ‘‘time-frame’’ over a finite DAG. Each link  $n$  is capable of transmitting  $C_n$  symbols per time-frame with no transmission delay. The rate vector  $(R_1, \dots, R_I)$  is feasible if the original graph  $G$  can be viewed as the superposition of one graph  $G_r$  and many graphs  $G_p$ 's such that (i) routing is performed for every  $(s_i, t_i)$  pair in  $G_r$ , (ii) pairwise linear network coding across  $(s_i, t_i)$  and  $(s_j, t_j)$  is performed in each  $G_p$  individually with a rate  $g_p$ , and (iii) the transmission rates  $(R_1, \dots, R_I)$  can be supported. Here,  $R_i = x_i + \sum_{G_p \in L(i)} g_p$ , where  $x_i$  is the rate supported for session  $i$  in  $G_r$ , and  $L(i)$  is the set of all  $G_p$  where pairwise network coding is performed between session  $(s_i, t_i)$  and any other session. The necessary and sufficient condition of each  $G_p$  is as stated in *Prelim 1* with the modification that the capacity of all links is  $g_p$  instead of the unit rate. This result illustrates how to use the result in *Prelim 1* in a network with more than two sessions.

*Prelim 2* describes a capacity region that we will call by the initials of the names of the authors the (*WS*) region.

Since in the  $\mathcal{WS}$  region, the rate  $R_i$  is expressed as the sum of the rates with/without inter-session network coding, two sets of parameters and variables will be used in our formulation. Some parameters and variables are for the routing-only graph  $G_r$  and the others capture the inter-session network coding performed on graphs  $G_p$  in the optimization problem. For  $G_r$ , we define the parameters  $J(i)$  and  $E_i^{nk}$ , and the variable  $x_i^k$ . Let  $J(i)$  represents the number of paths between  $s_i$  and  $t_i$ . If link  $n$  is used by the  $k$ -th path between  $s_i$  and  $t_i$ , where  $k$  ranges from 1 to  $J(i)$ , then  $E_i^{nk} = 1$ . Otherwise  $x_i^k$  is set to zero. We define  $x_i^k$  to represent the uncoded, routing rate supported through the  $k$ -th path between  $s_i$  and  $t_i$  in  $G_r$ .  $\vec{x}$  is a column vector containing  $x_i^k \forall i, k$ . For  $G_p$ , we define the parameters  $\mathcal{R}(i, j)$ ,  $J(i, j)$ , and  $H_{ij}^{nl}$ , and the variable  $g_{ij}^{lm}$ .  $\mathcal{R}(i, j)$  is the set of all tuples containing all possible choices of paths  $\{P_{s_i, t_i}, P_{s_j, t_j}, P_{s_j, t_i}\}$  and  $J(i, j) = |\mathcal{R}(i, j)|$ . Based on  $\mathcal{R}(i, j)$ ,  $H_{ij}^{nl}$  is defined in the following manner:

$$H_{ij}^{nl} = \begin{cases} 0 & \text{if no paths in the } l\text{-th tuple in } \mathcal{R}(i, j) \\ & \text{use link } n \\ 1 & \text{if 1 or 2 paths in the } l\text{-th tuple in} \\ & \mathcal{R}(i, j) \text{ use link } n \\ 2 & \text{if 3 paths in the } l\text{-th tuple in } \mathcal{R}(i, j) \\ & \text{use link } n \end{cases}$$

In this paper we denote any possible choice of paths that can form  $\mathcal{P}$  and  $\mathcal{Q}$  in *Prelim 1* as a *coding session*. Any pairwise coding session between sessions  $i$  and  $j$  can be indexed by  $l$  and  $m$ , namely, when the  $l$ -th tuple in  $\mathcal{R}(i, j)$  and the  $m$ -th tuple in  $\mathcal{R}(j, i)$  are used as the corresponding  $\mathcal{P}$  and  $\mathcal{Q}$ . We denote the coding graph formed by the paths in the  $l$ -th tuple in  $\mathcal{R}(i, j)$  and the paths in the  $m$ -th tuple in  $\mathcal{R}(j, i)$  by  $G_{ij}^{lm}$ , and the rate supported by pairwise network coding for  $i$  and  $j$  is denoted by the  $g_{ij}^{lm}$ . We also define  $\vec{g}$  as a column vector containing  $g_{ij}^{lm}, \forall i, j, l, m$ . Therefore, the total supported rate becomes  $R_i = \sum_{k=1}^{J(i)} x_i^k + \sum_{j \neq i} \sum_{l=1}^{J(i, j)} \sum_{m=1}^{J(j, i)} g_{ij}^{lm}$ .

Consider a specific link  $n$ . The capacity consumed by pure routing traffic is:  $\sum_{i=1}^I \sum_{k=1}^{J(i)} E_i^{nk} x_i^k$ . For any coding session represented by the graph  $G_{ij}^{lm}$  indexed by  $l$  and  $m$ , the capacity consumed by  $\mathcal{P}$  is  $H_{ij}^{nl} g_{ij}^{lm}$ . This is because by *Prelim 1*, the successful pairwise network coding requires that  $\text{es}_{\mathcal{P}}(n) \leq 2$ . If all three paths in  $\mathcal{P}$  use link  $n$ , then the traffic along these three paths must use two parallel links instead of a single one. Otherwise, the link share number will be three, which violates the necessary and sufficient condition for pairwise inter-session network coding. The same argument holds for the traffic along the paths in  $\mathcal{Q}$ , the  $m$ -th tuple in  $\mathcal{R}(j, i)$ , for which paths consume  $H_{ji}^{nm} g_{ij}^{lm}$ . From the above reasoning, the total capacity consumed by inter-session network coding for  $G_{ij}^{lm}$  is the maximum of the two which is formally expressed as  $\max(H_{ij}^{nl}, H_{ji}^{nm}) g_{ij}^{lm}$ . Summing over all pairs of sessions  $i \neq j$ , and all  $l$ -th and  $m$ -th tuples of  $\mathcal{R}(i, j)$  and  $\mathcal{R}(j, i)$ , the total capacity consumed by inter-session network coding becomes  $\sum_{i=1}^I \sum_{i < j} \sum_{l=1}^{J(i, j)} \sum_{m=1}^{J(j, i)} \max(H_{ij}^{nl}, H_{ji}^{nm}) g_{ij}^{lm}$ .

From the above discussion, the following linear constraints represent  $\{\vec{x}, \vec{g}\} \in \mathcal{WS}$ .

$$\begin{aligned} \sum_{i=1}^I \sum_{k=1}^{J(i)} E_i^{nk} x_i^k + \sum_{i=1}^I \sum_{i < j} \sum_{l=1}^{J(i, j)} \sum_{m=1}^{J(j, i)} \max(H_{ij}^{nl}, H_{ji}^{nm}) g_{ij}^{lm} \\ \leq C_n \quad \forall n \in \{1, \dots, |E|\} \\ g_{ij}^{lm} = g_{ji}^{ml} \quad \forall i < j, l, m. \end{aligned} \quad (12)$$

## 5.2 Distributed Rate Control Algorithms

In [11] we have developed a distributed rate control algorithm and coding scheme for the  $\mathcal{WS}$  region. The rate control algorithm is derived using the duality approach and the proximal method. It converges to the optimal solution of:

$$\max \sum_i U_i(R_i)$$

subject to:

$$\{\vec{x}, \vec{g}\} \in \mathcal{WS},$$

where  $U_i$  is a concave utility function associated with each source  $i$ . A dual variable  $\lambda_n$  is defined for each link  $n$ . Also, for each coding session  $G_{ij}^{lm}$ , we define the dual variable  $\mu_{ij}^{lm}$ . The distributed rate control algorithm, Algorithm  $\mathcal{A}$  is described as follows.

*Algorithm A:*

- Initialization phase:

Find all paths between all sources and sinks. This can be done using any routing protocol that finds multiple paths as [15, 27]. After this, sources send control messages to every link  $n$  to set the values of  $H_{ij}^{nl}$  and  $E_i^{nk}$ . Each  $s_i$  chooses the values of  $y_i^k(0)$  and  $h_{ij}^{lm}(0)$  randomly and each link sets the elements of  $\vec{\mu}(0)$  and  $\vec{\lambda}(0)$  to zero.

- Iteration phase: At the  $t$ -th iteration:

1. Fix  $\vec{y} = \vec{y}(t)$ , and  $\vec{h} = \vec{h}(t)$ ,  $\vec{\lambda}(t, 0) = \vec{\lambda}(t)$ ,  $\vec{\mu}(t, 0) = \vec{\mu}(t)$ , and perform the following steps sequentially for  $r = 0, \dots, K - 1$ .

- Let  $\{\vec{x}(t, r), \vec{g}(t, r)\} = \arg \max_{\{\vec{x}, \vec{g}\} \geq 0} L(\vec{x}, \vec{g}, \vec{\lambda}(t, r), \vec{\mu}(t, r), \vec{y}(t), \vec{h}(t))$ .

This can be computed in a distributed way at each source, because the  $L$  function is separable.

- Update the dual variables at each link  $n$  by:

$$\begin{aligned} \lambda^n(t, r + 1) \\ = [\lambda^n(t, r) + \alpha_n (\sum_{i=1}^I \sum_{k=1}^{J(i)} E_i^{nk} x_i^k(t, r) \\ + \sum_{i=1}^I \sum_{i \neq j} \sum_{l=1}^{J(i, j)} \sum_{m=1}^{J(j, i)} F_{ij}^{nlm} g_{ij}^{lm}(t, r) - C_n)]^+. \end{aligned} \quad (13)$$

Here,  $F_{ij}^{nlm} = \frac{1}{2} \max(H_{ij}^{nl}, H_{ji}^{nm})$ ,  $[\cdot]^+$  is a projection on  $[0, \infty)$ , and  $\alpha_n$ , is a positive step size. Also,  $(\sum_{i=1}^I \sum_{k=1}^{J(i)} E_i^{nk} x_i^k(t, r) + \sum_{i=1}^I \sum_{i \neq j} \sum_{l=1}^{J(i, j)} \sum_{m=1}^{J(j, i)} F_{ij}^{nlm} g_{ij}^{lm}(t, r) - C_n)$  is the average queue length at link  $n$ .

- Set

$$\begin{aligned} \mu_{ij}^{lm}(t, r + 1) &= \mu_{ij}^{lm}(t, r) + \\ \beta_{ij}^{lm} (g_{ij}^{lm}(t, r) - g_{ji}^{ml}(t, r)) \quad \forall i < j. \end{aligned} \quad (14)$$

This can be implemented at the destination, where  $\beta_{ij}^{lm}$  is a positive step size.

It is worth noting that computing  $\vec{x}(t, r)$  and  $\vec{g}(t, r)$  needs the values of (i)  $\sum_n F_{ij}^{nlm} \lambda_n(t, r) \forall i < j, l, m$ , which can be computed along the

paths, (ii)  $\mu_{ij}^{lm}(t, r) \forall i < j, l, m$ , and (iii)  $\mu_{ji}^{lm}(t, r) \forall i > j, l, m$ . All this information can be sent back to the source using the acknowledgment message.

2. Let  $\vec{\lambda}(t+1) = \vec{\lambda}(t, K)$  and  $\vec{\mu}(t+1) = \vec{\mu}(t, K)$ . Set:

$$y_i^k(t+1) = f_i^k(t) \quad \forall i, k$$

and

$$h_{ij}^{lm}(t+1) = z_{ij}^{lm}(t) \quad \forall i < j, l, m.$$

Here,  $\vec{f}(t)$  and  $\vec{z}(t)$  are the primal variables that maximize the Lagrangian given  $\vec{\lambda}(t+1)$ ,  $\vec{\mu}(t+1)$ ,  $\vec{y}(t)$ , and  $\vec{h}(t)$ .

In our online technical report [12], we prove the convergence of the algorithm as  $K \rightarrow \infty$ . For the case when  $K$  is bounded away from infinity, the convergence of Algorithm  $\mathcal{A}$  is verified by simulations in [11].

### 5.3 Pairwise Random Coding Scheme

The optimization problem and the solution described so far allocate rates at each link so that the utility function can be optimized subject to  $\{\vec{x}, \vec{y}\} \in \mathcal{WS}$ . The next question is how to design a network coding scheme that can achieve the rate assignment? In this section, we propose the use of the Pairwise Random coding (PRC) scheme. From the construction of  $G_{ij}^{lm}$  in Section 2,  $G_{ij}^{lm}$  is a graph satisfying the necessary and sufficient condition for pairwise network coding specified in *Prelim 1*. Since our discussion is based on fixed values of  $i, j, l$ , and  $m$ , for simplicity, we will use here  $G_p = G_{ij}^{lm}$  as a shorthand. By normalizing over the rate  $g_{ij}^{lm}$  associated for  $G_{ij}^{lm}$ , we can assume that  $G_p$  has unit rate link capacity. We can further assume that the messages for the unicast sessions  $(s_1, t_1)$ ,  $(s_2, t_2)$  are  $X_1, X_2$ , respectively. Based on this, the PRC scheme is described in Figure 5.3 as follows.

*PRC Coding Scheme:*

```

foreach node  $v$  in the network do
  foreach outgoing link  $n$  of  $v$  do
    if  $\{P_{s_1, t_1}, P_{s_2, t_1}, Q_{s_1, t_1}\}$  share link  $n$  then
      decode  $X_1$  and send it through  $n$ 
    end
    else
      if  $\{P_{s_2, t_2}, Q_{s_1, t_2}, Q_{s_2, t_2}\}$  share link  $n$  then
        decode  $X_2$  and send it through  $n$ 
      end
      else
        perform random coding between input
        messages and form the output messages
        in  $n$ .
      end
    end
  end
end

```

Figure 5: The PRC scheme

The following theorem provides a lower bound on the probability of success of the PRC scheme.

**THEOREM 2.** *Given that pairwise inter-session network coding is feasible on  $G_p$ , the probability that the PRC scheme is able to transmit  $X_1$  and  $X_2$  successfully through sessions  $(s_1, t_1)$  and  $(s_2, t_2)$ , respectively, at unit rates is lower bounded by  $P_r(\text{success}) \geq (1 - \frac{4}{q})^{6|E|} \forall q > 4$ .*

The proof of this theorem is quite technical and long. It is provided in our online technical report [12].

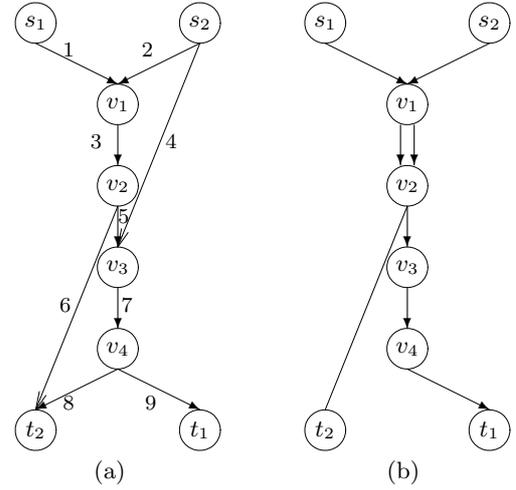


Figure 6: Example showing how the  $\mathcal{I} - \mathcal{TRLMK}$  region uses remedy packets links to resolve bottleneck in the grail structure. Coding solution on (a) is equivalent to routing solution on (b).

The PRC scheme can be implemented in a distributed way. Each node  $v$  needs only to know whether the paths in  $\mathcal{P}_1$  (or  $\mathcal{P}_2$ ) share the same outgoing link adjacent to  $v$ , which can be obtained during the initialization phase of Algorithm  $\mathcal{A}$ . Based on that piece of information, node  $v$  decides whether to perform decoding or random mixing. Furthermore, from *Theorem 2*, we can see that the success probability of PRC scheme approaches one when the size of the finite field is sufficiently large.

## 6. IMPROVED $\mathcal{TRLMK}$ REGION $\mathcal{I} - \mathcal{TRLMK}$

The  $\mathcal{I} - \mathcal{TRLMK}$  region is structure-based capacity region. It captures any butterfly or grail topology shown in Figure 1 to increase network performance.

### 6.1 Description and Formulation

The grail topology in Figure 1(b) is another example where pairwise network coding results can resolve the bottleneck in the network. Consider the grail topology in Figure 6 with unit capacity links. Source nodes  $s_1$  and  $s_2$  want to send packets to  $t_1$  and  $t_2$ , respectively. The coding solution in Figure 6(a) is, in effect, equal to the routing solution in Figure 6(b). Here, the bottleneck at link 3 is resolved by remedy packets at links 4 and 8. The only difference between resolving bottlenecks in the butterfly and grail topologies is that in the butterfly topology the remedy packets have to be generated by nodes previously traversed by  $X_1$  and  $X_2$ , respectively, while in the grail topology,  $v_4$ , the remedy generating node for  $X_2$  is located after  $v_3$ , the decoding node for  $X_1$ . To describe the  $\mathcal{I} - \mathcal{TRLMK}$  region, we perform two changes to the variables used to describe the  $\mathcal{TRLMK}$  region. Firstly, we associate the variables  $p(\cdot)$ ,  $q(\cdot)$ , and  $r(\cdot)$  with three nodes representing the encoding node and the two decoding nodes. For example, for any session  $i$ ,  $p_n(i \rightarrow j, u, v, w)$  represents the poison flow resulted from the poisoning session  $j$  data with that of session  $i$  at node  $u$ , for which the decoding will happen at nodes  $v$  and  $w$ . The second change is that we introduce the variable  $l_n(i \rightarrow j, u, v, w)$  that represents the request to node  $v$  to generate a remedy flow to node  $w$ . Figure 7 represents the assignment of the variables on the grail topology. The following set of constraints represent the  $\mathcal{I} - \mathcal{TRLMK}$  region.

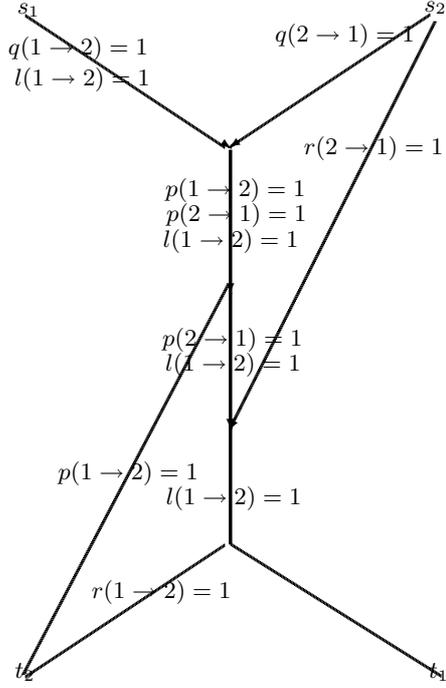


Figure 7: The assignment of the variables on the  $\mathcal{I} - \mathcal{TRLKM}$  formulation on the grail topology with unit capacity links.

$$\sum_{n \in \Gamma_O(g)} x_n(i) - \sum_{n \in \Gamma_I(g)} x_n(i) = \begin{cases} R_i & g = s_i \\ -R_i & g = t_i \\ 0 & \text{else} \end{cases} \quad \forall i \in \{1, \dots, I\}. \quad (15)$$

$$\begin{aligned} & \sum_{n \in \Gamma_O(g)} r_n(i \rightarrow j, u, v, w) + l_n(i \rightarrow j, u, v, w) \\ & - p_n(i \rightarrow j, u, v, w) - q_n(i \rightarrow j, u, v, w) = \\ & \sum_{n \in \Gamma_I(g)} r_n(i \rightarrow j, u, v, w) + l_n(i \rightarrow j, u, v, w) \\ & - p_n(i \rightarrow j, u, v, w) - q_n(i \rightarrow j, u, v, w) \\ & \quad \forall u, v, w, g \in V, i, j \in \{1, \dots, I\}. \quad (16) \end{aligned}$$

$$\begin{aligned} & \sum_{n \in \Gamma_I(g)} q_n(i \rightarrow j, u, v, w) - \sum_{n \in \Gamma_O(g)} q_n(i \rightarrow j, u, v, w) = \\ & \begin{cases} \leq 0 & g = u \\ \geq 0 & \text{else} \end{cases} \\ & \quad \forall u, v, w, g \in V, i, j \in \{1, \dots, I\}. \quad (17) \end{aligned}$$

$$\begin{aligned} & \sum_{n \in \Gamma_I(g)} p_n(i \rightarrow j, u, v, w) - \sum_{n \in \Gamma_O(g)} p_n(i \rightarrow j, u, v, w) = \\ & \begin{cases} \geq 0 & g = u \\ \leq 0 & \text{else} \end{cases} \\ & \quad \forall u, v, w, g \in V, i, j \in \{1, \dots, I\}. \quad (18) \end{aligned}$$

$$p_n(i \rightarrow j, u, v, w) = p_n(j \rightarrow i, u, v, w) \quad \text{if } n \in \Gamma_O(u) \\ \forall u, v, w \in V, i, j \in \{1, \dots, I\}. \quad (19)$$

$$l_n(i \rightarrow j, u, v, w) = 0 \quad \text{if } n \in \Gamma_I(w) \\ \forall u, v, w \in V, i, j \in \{1, \dots, I\}. \quad (20)$$

$$l_n(i \rightarrow j, u, v, w) = x_n(i) - \sum_{k \neq i, m \neq j} p_n(k \rightarrow m, u, v, w) \\ \text{if } l_n(i \rightarrow j, u, v, w) \neq 0 \\ \forall n \in E, u, v, w \in V, i, j \in \{1, \dots, I\} \quad (21)$$

$$\begin{aligned} & \sum_{n \in \Gamma_O(g)} l_n(i \rightarrow j, u, v, w) - \sum_{n \in \Gamma_I(g)} l_n(i \rightarrow j, u, v, w) \geq 0 \\ & \text{if } \sum_{n \in \Gamma_O(g)} p_n(i \rightarrow j, u, v, w) > 0 \\ & \quad \forall n \in E, u, v, w \in V, i, j \in \{1, \dots, I\} \quad (22) \end{aligned}$$

$$\begin{aligned} & \sum_{u, v, w} \sum_{\substack{i, j \\ i < j}} -\min(p_n(i \rightarrow j, u, v, w), p_n(j \rightarrow i, u, w, v)) + \\ & \sum_{i=1}^I x_n(i) + \sum_{u, v, w} \sum_{\substack{i, j \\ i \neq j}} r_n(i \rightarrow j, u, v, w) \leq C_n \quad \forall n \in E \end{aligned} \quad (23)$$

$$x_n(j) - \sum_{u, v, w} \sum_i (p_n(i \rightarrow j, u, v, w) + (p_n(j \rightarrow i, u, w, v))) \geq 0 \\ \forall n \in E, j \in \{1, \dots, I\} \quad (24)$$

$x_n(\cdot), r_n(\cdot), p_n(\cdot), q_n(\cdot), r_n(\cdot) \geq 0$ .

The following theorem shows that the constraints in (15)-(24) are sufficient for the existence of inter-session network coding scheme based on finding butterfly or grail structures.

**THEOREM 3.** Consider a network  $G = (V, E)$ , where the capacity of each link  $n$  is  $C_n$ . If the constraints in (15)-(24) are satisfied, then there exists an XOR coding scheme based on finding butterfly or grail structures in the network such that the rate supported by each session  $i$  is  $R_i$ .

The proof is similar to that of Theorem 1 and is omitted due to space constraints. Moreover, with the help of Theorem 3 in [25], it can be shown that in terms of the sustainable rates,  $\mathcal{I} - \mathcal{TRLKM}$  provides the strictly largest regions when compared to  $\mathcal{TRLKM}$  and  $\mathcal{WS}$  regions:

**COROLLARY 1.** For any network with  $I$  unicast sessions, any rate vector  $(R_1, \dots, R_I)$  that is achievable with the  $\mathcal{TRLKM}$  or the  $\mathcal{WS}$  region is also achievable with the  $\mathcal{I} - \mathcal{TRLKM}$  region.

## 6.2 Distributed Algorithm

The formulation of the  $\mathcal{I} - \mathcal{TRLKM}$  contains some non-linear constraints as in (21) and (22). However, we can obtain a distributed algorithm for the  $\mathcal{I} - \mathcal{TRLKM}$  region by simple modification to the  $\mathcal{RSC}$  algorithm. The only difference between the  $\mathcal{TRLKM}$  and the  $\mathcal{I} - \mathcal{TRLKM}$  is that  $b_1$  or  $b_2$  in Figure 4, but not both, can be a descendant node of  $c_2$  or  $c_1$ . We define  $\sigma_{1(u,v)}^*[t]$  to be the same as  $\sigma_{(u,v)}^*[t]$ , with the change that  $b_1$  is a node that session  $D_1$  traverse after  $c_2$ . Also,  $\sigma_{2(u,v)}^*[t]$  is defined the same as  $\sigma_{(u,v)}^*[t]$ , with the change that  $b_2$  is a node that session  $D_2$  traverse after  $c_1$ . At time  $t$ , the following changes to the  $\mathcal{RSC}$  Algorithm enable it to implement the  $\mathcal{I} - \mathcal{TRLKM}$  capacity region. We will refer to the modified algorithm as  $\mathcal{I} - \mathcal{RSC}$  Algorithm.

- If  $\rho_{(u,v)}^*[t] > \max(\sigma_{(u,v)}^*[t], \sigma_{1(u,v)}^*[t], \sigma_{2(u,v)}^*[t])$ , step A is performed.

- If  $\sigma_{(u,v)}^*[t] > \max(\rho_{(u,v)}^*[t], \sigma_{1(u,v)}^*[t], \sigma_{2(u,v)}^*[t])$ , step  $B$  is performed.
- If  $\sigma_{1(u,v)}^*[t] > \max(\rho_{(u,v)}^*[t], \sigma_{(u,v)}^*[t], \sigma_{2(u,v)}^*[t])$ , step  $B$  is performed with the change that in step  $B3$  the address of  $b_1^*$  is added to the header of the XORed packet so that a remedy packet can be generated at  $b_1^*$ .
- If  $\sigma_{2(u,v)}^*[t] > \max(\rho_{(u,v)}^*[t], \sigma_{(u,v)}^*[t], \sigma_{1(u,v)}^*[t])$ , step  $B$  is performed with the change that in step  $B3$  the address of  $b_2^*$  is added to the header of the XORed packet so that a remedy packet can be generated at  $b_2^*$ .

## 7. COMPARISON

### 7.1 Capacity and Fairness

It is worth noting that all the capacity regions discussed previously focus on some special opportunities of intersession network coding, including taking advantages of the butterfly structure ( $TRCLKM$ ), butterfly + grail ( $I - TRCLKM$ ), or the pairwise intersession network coding. For network coding with its most general form, it is still an open problem to come up with analytical expression to represent the benefits of inter-session network coding over routing. In this section we will compare the benefits of aforementioned capacity regions based on some experimental examples. backlog algorithms of the former, such as [4, 7] for the  $TRCLKM$  region and the one in Section 6.2 for  $I - TRCLKM$  region, focus on sustaining a given rate within the capacity region instead of maximizing any general concave utility function. On the other hand, the distributed implementation for the path-based capacity region as in Section 5.2 admits decentralized rate control to optimize any given concave utility function. The first example to consider in this section is the topology represented in Figure 8. For this topology, the  $TRCLKM$  work [23] and its distributed implementation in [4] and [7] cannot realize any benefits of network coding since there is no butterfly structure and the performance of these algorithms is the same as that of routing. We define the utility gain of inter-session network coding  $UG(\mathcal{B})$  as

$$UG(\mathcal{B}) = \frac{Utility(\mathcal{B}) - Utility(TRCLKM)}{Utility(TRCLKM)}.$$

We denote the total throughput of the network when the optimal utility is achieved under the  $\mathcal{B}$  region by  $\sum_i R_i(\mathcal{B})$ . The throughput gain,  $TG$  is defined as

$$TG(\mathcal{B}) = \frac{\sum_i R_i(\mathcal{B}) - \sum_i R_i(TRCLKM)}{\sum_i R_i(TRCLKM)}.$$

For this topology, the throughput and utility gains of both  $WS$  and  $I - TRCLKM$  regions are the same:  $TG(WS) = TG(I - TRCLKM)$  and  $UG(WS) = UG(I - TRCLKM)$ , since both of them will take full advantages of the grail structures in Figure 8. We evaluate the gains using different utility functions. The first type of the utility function is  $\log_2(\gamma + R_i)$ , where  $\gamma$  is a constant in the range  $[0, 1]$ . The second type of utility function is of the form  $\frac{R_i^{1-\alpha}}{1-\alpha}$ , where  $\alpha$  is a constant in the range  $(0, 1)$ . The simulation results are shown in Figures 9 and 10. The  $WS$  and  $I - TRCLKM$  regions outperform both routing and the  $TRCLKM$  region on this topology. Moreover, the largest throughput gain happens when fairness is the design criteria for the network, i.e., when  $\gamma$  is small and when  $\alpha$  is large. Using the  $WS$  and  $I - TRCLKM$  regions indeed enhance fairness.

The second example is the network shown in Figure 11. Every link in this network has a unit capacity links. Every source node  $s_i$  want to send its packets to  $t_i$ . The rate vector

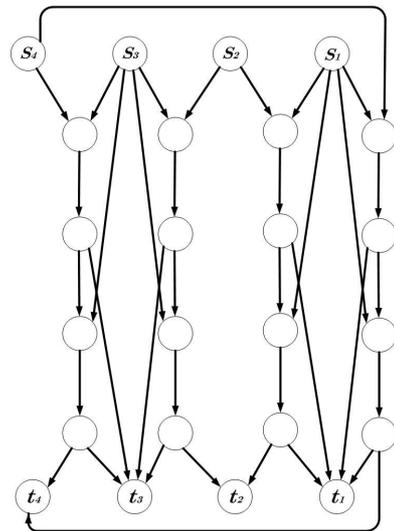


Figure 8: Topology contains four source-sink pairs

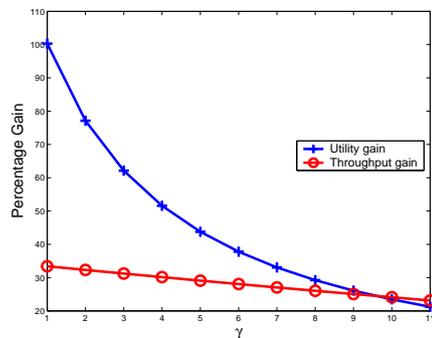


Figure 9: Gain for objective function  $\sum_i \log_2(\gamma + R_i)$  with different values of  $\gamma$

$\vec{R} = [1, 1, 1, 1]^T$  is achievable with both the  $TRCLKM$  and  $I - TRCLKM$  regions, but not with the  $WS$  region. In the network in Figure 12, with the same link capacities and requirements in Figure 11, The rate vector  $\vec{R} = [1, 1, 1, 1]^T$  is achievable only with the  $I - TRCLKM$  region. These examples shows that the gain of using different capacity regions depends on the network topology and the locations of the sources and sinks.

### 7.2 Complexity

The analytical comparison of the complexities of the optimization problems associated with the three capacity regions is in Table 1. The complexity includes number of variables and constraints and the nature of constraints for each optimization problem. The optimization problem devised for the  $WS$  region reduces the number of constraints tremendously. The number of variables in the optimization problem for the  $WS$  region depends on the number of paths between different sender receiver pairs. The number of variables for both the  $TRCLKM$  and  $I - TRCLKM$  is polynomial with relatively large order. All three capacity regions can be used for utility optimization if the optimization is performed centrally and offline. Based on the backlog method, the distributed implementations of the  $TRCLKM$  and the  $I - TRCLKM$  regions focus on sustain the rates while Algorithm  $\mathcal{A}$  for the  $WS$  region is a fully distributed version of utility optimization. The difference will be further discussed in the next subsection.

	Constraints	# Variables	# Constraints	Coding scheme
$TRCKM$	Linear	$> 3 E  V (I^2 - I)$	$> (3 V  +  E ) V (I^2 - I)$	Limited to XOR
$WS$	Linear	$\sum_{i=1}^I \sum_{i < j} J(i, j)J(j, i) + \sum_i J(i)$	$ E $	Random
$I - TRCKM$	Non Linear	$> 3 E ( V )^3(I^2 - I)$	$> (3 V  +  E ) V (I^2 - I)$	Limited to XOR

Table 1: The complexity of the optimization problems for different capacity regions

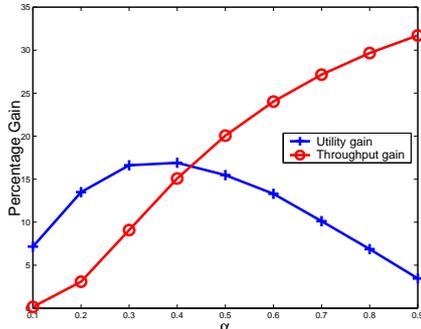


Figure 10: Gain for objective function  $\sum_i \frac{R_i^{1-\alpha}}{1-\alpha}$  with different values of  $\alpha$

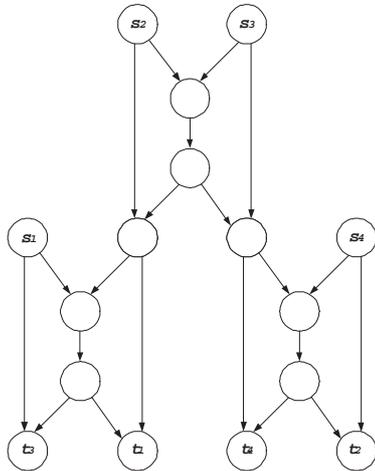


Figure 11: Topology used for comparing the capacity regions

### 7.3 Distributed Implementation Aspects

In this section we brief some of the advantages of the path-based over the structure-based algorithms.

- The path-based approach enables the algorithm to have rate control capability. For example, the  $RSC$  algorithm and the distributed algorithm in [7] assume that the rates of the sources lie strictly inside the capacity region. If the rates are outside the capacity region, the system will be unstable. Using the path-based approach, the system can perform rate control to maximize any concave objective function. This means that different types of fairness can be achieved.
- In structure-based approaches, coding and encoding have to be done at specific places in the network corresponding to the specific substructure captured by the algorithm is designed to capture is found. This makes the rate allocation and coding mechanism dependent, which requires exchanging queue lengths between specific intermediate nodes in the network. In the path-based approach random coding is performed

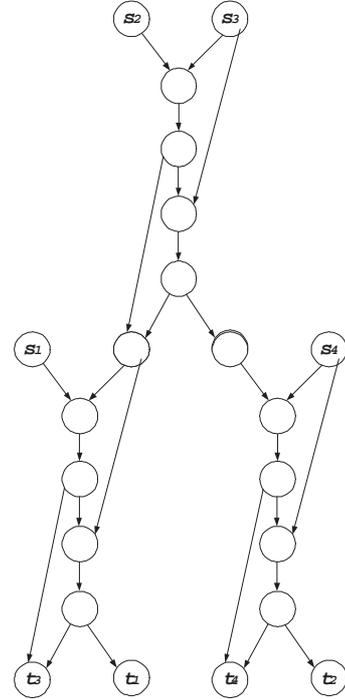


Figure 12: Topology used for comparing the capacity regions

and is independent from rate allocation, which makes it suitable for online implementation.

- With some realistic assumptions, using the path-based approach, an adaptive path selection module can be used to reduce the complexity of the algorithm, which is mainly due to the large number of paths being considered at once. A further detailed description of the interaction between the adaptive path-selection module and Algorithm A can be found in [11]. For the structure-based algorithms, adaptively reducing the complexity generally results in a smaller supportable rate regions since it means that the algorithms only take advantages of some (not all) the beneficial substructures. See the opportunistic network coding scheme described in [9] as an example.

Table 2 summarizes the comparison between different algorithms.

## 8. CONCLUSION

In this paper, we have introduced different capacity regions and distributed implementations for multiple unicasts using inter-session network coding. These capacity regions can be classified into path-based and structure-based capacity regions. We have shown the benefits of path-based region in terms of complexity and suitability of online implementations. For capacity and fairness, we used different examples to show that the performance of the capacity region depends

	Approach	Rate control	Rate alloc and coding	Queues exchange	Adaptive complexity reduction	Coding scheme
$\mathcal{RSC}$	Structure based	NO	Dependent	YES	hard	Limited to XOR
Algorithm A	Path based	YES	Independent	NO	easy	Random
$\mathcal{I} - \mathcal{RSC}$	Structure based	NO	Dependent	YES	hard	Limited to XOR

**Table 2: Comparison between different distributed algorithms**

on the topology used and the locations of sender-receiver pairs.

All the capacity regions discussed in this paper are based on *pairwise* inter-session network coding. We are now investigating the characterization of the necessary and sufficient conditions for inter-session network coding with more than two unicast sessions to enable us to further extend the described capacity regions.

## 9. REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. on Information Theory*, 46:1204–1216, 2000.
- [2] B. Awerbuch and T. Leighton. Improved approximation algorithms for the multicommodity flow problem and local competitive routing in dynamic networks. In *Proceedings of ACM Symposium on Theory of Computing*, 1994.
- [3] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *IEEE Trans. on Information Theory*, 51:2745–2759, 2005.
- [4] A. Eryilmaz and D. S. Lun. Control for inter-session network coding. In *proceedings of Netcod, San Diego*, Jan 2007.
- [5] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE/ACM Trans. on Information Theory* *Theory*, 46 (2):388–404, march 2000.
- [6] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley. Overlay TCP for multi-path routing and congestion control. In *IMA Workshop on Measurements and Modeling of the Internet*, Jan 2004.
- [7] T. Ho, Y. Chang, and K. J. Han. On constructive network coding for multiple unicasts. In *44th Allerton Conference on Communication, Control and Computing, Monticello, IL*, Sept 2006.
- [8] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *proceedings of IEEE International Symposium on information theory, Yokohama, Japan*, June 2003.
- [9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the air: Practical wireless network coding. In *proceedings of ACM SIGCOM, Pisa, Italy*, Sept 2006.
- [10] F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
- [11] A. Khreishah, C.-C. Wang, and N. B. Shroff. Optimization based rate control for communication networks with inter-session network coding. *Submitted to INFOCOM, preprint*, 2007.
- [12] A. Khreishah, C.-C. Wang, and N. B. Shroff. Optimization based rate control for communication networks with inter-session network coding. Technical report, Purdue University, <http://web.ics.purdue.edu/~akhreish/RCTR07.pdf>, July 2007.
- [13] R. Koetter and M. Medard. Beyond routing: An algebraic approach to network coding. In *proceedings of INFOCOM, New York*, June 2002.
- [14] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, Nov 1999.
- [15] R. Leung, J. Liu, E. Poon, C. Chan, and B. Li. MP-DSR: A QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks. In *proceedings of IEEE LCN, Tampa, FL*, Nov 2001.
- [16] R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Trans. on Information Theory*, 49(2):371–381, 2003.
- [17] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks. In *Proceedings of IEEE INFOCOM, Miami, FL*, March 2005.
- [18] X. Lin and N. B. Shroff. Utility maximization for communication networks with multi-path routing. *IEEE Trans on Automatic Control*, May 2006.
- [19] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, 2003.
- [20] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *Proceedings of IEEE INFOCOM, Anochorage, AK*, May 2007.
- [21] G. Sharma, R. R. Mazumdar, and N. B. Shroff. Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. In *Proceedings of IEEE INFOCOM, Barcelona, Spain*, April 2006.
- [22] N. Sundaram, P. Ramanathan, and S. Banerjee. Multirate media streaming using networking coding. In *43th Allerton Conference on Communication, Control, and Computing, Monticello, IL*, Sept 2005.
- [23] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Medard. Network coding for multiple unicasts: An approach based on linear optimization. In *IEEE International Symposium on Information Theory, Seattle*, 2006.
- [24] C.-C. Wang and N. B. Shroff. Beyond the butterfly—a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions. In *proceedings IEEE International Symposium on Information Theory, Nice, France*, June 2007.
- [25] C.-C. Wang and N. B. Shroff. Intersession network coding for two simple multicast sessions. *Submitted to 45th Allerton Conference on Communication, Control and Computing*, 2007.
- [26] W. Wang, M. Palaniswami, and S. H. Low. Optimal flow control and routing in multi-path networks. *Performance Evaluation, vol. 52, no. 2-3*, Apr. 2003.
- [27] W. T. Zaumen and J. Garcia-Luna-Aceves. Loop-free multipath routing using generalized diffusing computations. In *proceedings of IEEE INFOCOM, San Francisco, CA*, March 1998.