

# Two-Way Function Computation

Seiyun Shin, *Student Member, IEEE*, and Changho Suh<sup>1</sup>, *Member, IEEE*

**Abstract**—We explore the role of interaction for the problem of reliable computation over two-way multicast networks. Specifically we consider a four-node network in which two nodes wish to compute a modulo-sum of two independent Bernoulli sources generated from the other two, and a similar task is done in the other direction. The main contribution of this work lies in the characterization of the computation capacity region for a deterministic model of the network via a novel transmission scheme. One consequence of this result is that, not only we can get an interaction gain over the one-way non-feedback computation capacities, but also we can get all the way to *perfect-feedback*<sup>1</sup> computation capacities simultaneously in both directions for some channel regimes. This result draws a parallel with the recent result developed in the context of two-way interference channels.

**Index Terms**—Computation capacity, interaction, network decomposition, perfect-feedback, two-way function multicast channel.

## I. INTRODUCTION

THE inherent two-way nature of communication links provides an opportunity to enable *interaction* among nodes. It allows the nodes to efficiently exchange their messages by adapting their transmitted signals to the past received signals that can be fed back through backward communication links. This problem was first studied by Shannon who derived the inner and outer bounds on its capacity region [3]. Although there have been results that improve upon Shannon's inner [4]–[6] and outer bounds [7], [8], we are still lacking in our understanding of how to treat two-way information exchange, and the underlying difficulty has impeded progress on this field over the past few decades.

Since interaction is enabled through the use of *feedback*, feedback is a more basic research topic that needs to be understood beforehand. The history of feedback traces back to Shannon who showed that feedback has no bearing on capacity for memoryless point-to-point channels [9]. Subsequent work demonstrated that feedback provides a gain for point-to-point channels with memory [10]–[14] as well as for many

multi-user channels [15]–[17]. For many scenarios, however, capacity improvements due to feedback are rather modest.

On the contrary, [18], [19] have changed the traditional viewpoint on the role of feedback. It is shown in [18], [19] that feedback offers more significant capacity gains for the Gaussian interference channel. Subsequent works [20]–[24] show more promise on the use of feedback. In particular, [24] demonstrates a very interesting result: Not only feedback can yield a net increase in capacity, but also we can sometimes get *perfect-feedback capacities* simultaneously in both directions.

We seek to examine the role of feedback for more general scenarios in which nodes now intend to compute *functions* of the raw messages rather than the messages themselves. These general settings include many realistic scenarios such as sensor networks [25] and cloud computing scenarios [26], [27]. For an idealistic scenario where feedback links are perfect with infinite capacities and are given for free, Suh-Gastpar [28] have shown that feedback provides a significant gain also for computation. However, the result in [28] assumes a dedicated infinite-capacity feedback link as in [19]. As an effort to explore a net gain that reflects feedback cost, [2] investigated a two-way setting of the function multicast channel considered in [28] where two nodes wish to compute a linear function (modulo-sum) of the two Bernoulli sources generated from the other two nodes. The two-way setting includes a backward computation demand as well, thus well capturing feedback cost. For a deterministic model, a scheme is proposed to demonstrate that a net interaction gain can occur also in the computation setting. However, the maximal interaction gain is not fully characterized due to a gap between the lower and upper bounds. In particular, whether or not one can get all the way to perfect-feedback computation capacities in both directions (as in the two-way interference channel [24]) has been unanswered.

To answer the question, we consider the Avestimehr-Diggavi-Tse (ADT) deterministic model [29] which well captures key properties of the wireless Gaussian channel since the model abstracts superposition and broadcast properties of the wireless Gaussian channel. For this model, we characterize the computation capacity region of the two-way function multicast channel via a new capacity-achieving scheme. As a result, we answer the above question positively. Specifically, we demonstrate that for some channel regimes (to be detailed later; see Corollary 1), the new scheme simultaneously achieves the perfect-feedback computation capacities in both directions. As in the two-way interference channel [24], this occurs even when feedback offers gains in both directions and thus feedback w.r.t. one direction must compete with the traffic in the other direction.

Manuscript received October 20, 2018; revised April 29, 2019; accepted July 19, 2019. Date of publication August 16, 2019; date of current version January 20, 2020. This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant 2018R1A1A1A05022889 and by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant FA2386-19-1-4050. This paper was presented in part at the Proceedings of Allerton Conference on Communication, Control, and Computing in 2017 [1] and 2014 [2].

The authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: seiyun.shin@kaist.ac.kr; chsuh@kaist.ac.kr).

Communicated by M. Wigger, Associate Editor for Shannon Theory.

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIT.2019.2935775

<sup>1</sup>This is an idealistic case where feedback links are perfect with infinite capacities and are given for free. We left the exact definition in Section II.

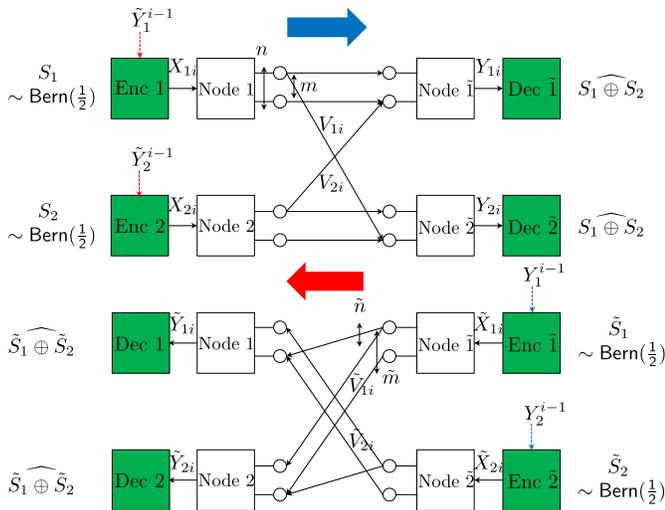


Fig. 1. Four-node ADT deterministic network.

Our achievability builds upon the scheme in [24] where feedback allows the exploitation of effectively future information as side information via retrospective decoding (to be detailed later; see Remark 3). A key distinction relative to [24] is that in our computation setting, the retrospective decoding occurs in a *nested manner* for some channel regimes; this will be detailed when describing our achievability. We also employ the network decomposition method of [30] to simplify the achievability proof.

*Notations:* Before continuing to describe our model, we introduce a few notations that will be used throughout. First, we assume that any vector in the form of  $X \in \mathbb{F}_2^q$  is a row vector. Here  $\mathbb{F}_2$  indicates the finite field with two elements. Also, we use shorthand notation to indicate the sequence, e.g.,  $S_1^K := (S_{11}, \dots, S_{1K})$ . The calligraphic letters  $\mathcal{R}$  and  $\mathcal{C}$  denote the computation rate region and the capacity region, respectively. Finally, we note that tilde notation  $\tilde{\cdot}$  is used to indicate parameters relevant to the backward channel.

## II. MODEL

Consider a four-node Avestimehr-Diggavi-Tse (ADT) deterministic network as illustrated in Fig. 1. This network is a full-duplex bidirectional system in which all nodes are able to transmit and receive signals simultaneously. Our model consists of forward and backward channels which are assumed to be orthogonal. For simplicity, we focus on a setting in which both forward and backward channels are symmetric, but not necessarily the same. In the forward channel,  $n$  and  $m$  indicate the number of signal bit levels (or resource levels) for direct and cross links respectively. The corresponding values for the backward channel are denoted by  $(\tilde{n}, \tilde{m})$ .

With  $N$  channel uses of the network (or in time  $N$ ), node  $k$  ( $k = 1, 2$ ) wishes to transmit the number  $K(N)$  forward messages, while node  $\tilde{k}$  ( $\tilde{k} = \tilde{1}, \tilde{2}$ ) wishes to transmit the number  $\tilde{K}(N)$  backward messages. While  $K(N)$  is a precise expression as it is indeed a function of  $N$ , for notational simplicity, we will denote  $K(N)$  by  $K$ .

Similarly we will denote  $\tilde{K}(N)$  by  $\tilde{K}$ . We define  $S_k^K$  and  $\tilde{S}_k^{\tilde{K}}$  as forward and backward messages respectively. We assume that  $(S_1^K, S_2^K, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}})$  are independent and identically distributed according to  $\text{Bern}(\frac{1}{2})$ . Let  $X_k \in \mathbb{F}_2^q$  be an encoded signal of node  $k$  where  $q = \max(m, n)$  and  $V_k \in \mathbb{F}_2^m$  be part of  $X_k$  visible to node  $\tilde{j}$  ( $\neq \tilde{k}$ ). Similarly let  $\tilde{X}_k \in \mathbb{F}_2^{\tilde{q}}$  be an encoded signal of node  $\tilde{k}$  where  $\tilde{q} = \max(\tilde{m}, \tilde{n})$  and  $\tilde{V}_k$  be part of  $\tilde{X}_k$  visible to node  $j$  ( $\neq k$ ). We note that all the signals are in the form of row vectors. The signals received at node  $k$  and  $\tilde{k}$  are then given by

$$Y_1 = X_1 \mathbf{G}^{q-n} \oplus X_2 \mathbf{G}^{q-m}, \quad Y_2 = X_1 \mathbf{G}^{q-m} \oplus X_2 \mathbf{G}^{q-n}, \quad (1)$$

$$\tilde{Y}_1 = \tilde{X}_1 \tilde{\mathbf{G}}^{\tilde{q}-\tilde{n}} \oplus \tilde{X}_2 \tilde{\mathbf{G}}^{\tilde{q}-\tilde{m}}, \quad \tilde{Y}_2 = \tilde{X}_1 \tilde{\mathbf{G}}^{\tilde{q}-\tilde{m}} \oplus \tilde{X}_2 \tilde{\mathbf{G}}^{\tilde{q}-\tilde{n}}, \quad (2)$$

where  $\mathbf{G}$  and  $\tilde{\mathbf{G}}$  are shift matrices:  $[\mathbf{G}]_{ij} = \mathbf{1}\{j = i + 1\}$  ( $1 \leq i, j \leq q$ ),  $[\tilde{\mathbf{G}}]_{ij} = \mathbf{1}\{j = i + 1\}$  ( $1 \leq i, j \leq \tilde{q}$ ). Here  $\oplus$  indicates bit wise XOR.

The encoded signal  $X_{ki}$  of node  $k$  at time  $i$  is a function of its own message and past received signals:  $X_{ki} = f_{ki}(S_1^K, \tilde{Y}_k^{i-1})$ . We define  $\tilde{Y}_k^{i-1} := \{\tilde{Y}_{kt}\}_{t=1}^{i-1}$  where  $\tilde{Y}_{kt}$  denotes node  $k$ 's received signal at time  $t$ . Similarly the encoded signal  $\tilde{X}_{ki}$  of node  $\tilde{k}$  at time  $i$  is a function of its own message and past received sequences:  $\tilde{X}_{ki} = \tilde{f}_{ki}(\tilde{S}_k^{\tilde{K}}, Y_k^{i-1})$ .

From the received signal  $Y_k^N$ , node  $\tilde{k}$  wishes to compute  $\{S_{1i} \oplus S_{2i}\}_{i=1}^K$ . Similarly node  $k$  wishes to compute  $\{\tilde{S}_{1j} \oplus \tilde{S}_{2j}\}_{j=1}^{\tilde{K}}$  from its received signals  $\tilde{Y}_k^N$ . An error occurs whenever  $\{S_{1i} \oplus S_{2i}\}_{i=1}^K \neq \{S_{1i} \oplus S_{2i}\}_{i=1}^K$  or  $\{\tilde{S}_{1j} \oplus \tilde{S}_{2j}\}_{j=1}^{\tilde{K}} \neq \{\tilde{S}_{1j} \oplus \tilde{S}_{2j}\}_{j=1}^{\tilde{K}}$ . The error probabilities are given by  $\delta_k = \Pr[\{S_{1i} \oplus S_{2i}\}_{i=1}^K \neq \{S_{1i} \oplus S_{2i}\}_{i=1}^K]$  and  $\tilde{\delta}_k = \Pr[\{\tilde{S}_{1j} \oplus \tilde{S}_{2j}\}_{j=1}^{\tilde{K}} \neq \{\tilde{S}_{1j} \oplus \tilde{S}_{2j}\}_{j=1}^{\tilde{K}}]$ . We say that a computation rate pair  $(R, \tilde{R}) = (\frac{K}{N}, \frac{\tilde{K}}{N})$  is achievable if there exists a family of codebooks and encoder/decoder functions such that all the decoding error probabilities go to zero as the number of channel uses  $N$  tends to infinity. The computation capacity region  $\mathcal{C}$  is the closure of the set of achievable computation rate pairs.

In this work, we compare our results (to be stated later) to two other results made under two different scenarios. The first is the non-feedback (non-interaction) scenario [30] in which there is no interaction among the signals arriving from different nodes and hence  $X_{ki} = f_{ki}(S_k^K)$ ,  $\tilde{X}_{ki} = \tilde{f}_{ki}(\tilde{S}_k^{\tilde{K}})$ . The second is the perfect-feedback scenario [28] in which noiseless output feedback is given for free to aid computations in both directions so that  $X_{ki} = f_{ki}(S_k^K, Y_1^{i-1}, Y_2^{i-1})$ ,  $\tilde{X}_{ki} = \tilde{f}_{ki}(\tilde{S}_k^{\tilde{K}}, \tilde{Y}_1^{i-1}, \tilde{Y}_2^{i-1})$ .

## III. MAIN RESULTS

**Theorem 1** (Two-way Computation Capacity). *The computation capacity region  $\mathcal{C}$  is the set of  $(R, \tilde{R})$  such that*

$$R \leq C_{\text{pf}}, \quad (3)$$

$$\tilde{R} \leq \tilde{C}_{\text{pf}}, \quad (4)$$

$$R + \tilde{R} \leq m + \tilde{m}, \quad (5)$$

$$R + \tilde{R} \leq n + \tilde{n}, \quad (6)$$

where  $C_{\text{pf}}$  and  $\tilde{C}_{\text{pf}}$  indicate the perfect-feedback computation capacities in the forward and backward channels respectively (see (9) and (10) in Baseline 2 for detailed formulae).

*Proof:* See Sections IV and V for the achievability and converse proofs respectively. ■

**Baseline 1** (Non-feedback Computation Capacity [30]). Let  $\alpha := \frac{m}{n}$  and  $\tilde{\alpha} := \frac{\tilde{m}}{\tilde{n}}$ . The computation capacity region  $C_{\text{no}}$  for the non-feedback scenario is the set of  $(R, \tilde{R})$  such that  $R \leq C_{\text{no}}$  and  $\tilde{R} \leq \tilde{C}_{\text{no}}$  where

$$C_{\text{no}} = \begin{cases} \min\{m, \frac{2}{3}n\}, & \alpha < 1, \\ \min\{n, \frac{2}{3}m\}, & \alpha > 1, \\ n, & \alpha = 1, \end{cases} \quad (7)$$

$$\tilde{C}_{\text{no}} = \begin{cases} \min\{\tilde{m}, \frac{2}{3}\tilde{n}\}, & \tilde{\alpha} < 1, \\ \min\{\tilde{n}, \frac{2}{3}\tilde{m}\}, & \tilde{\alpha} > 1, \\ \tilde{n}, & \tilde{\alpha} = 1. \end{cases} \quad (8)$$

Here  $C_{\text{no}}$  and  $\tilde{C}_{\text{no}}$  denote the non-feedback computation capacities of forward and backward channels respectively.

**Baseline 2** (Perfect-feedback Computation Capacity [28]). The computation capacity region  $C_{\text{pf}}$  for the perfect-feedback scenario is the set of  $(R, \tilde{R})$  such that  $R \leq C_{\text{pf}}$  and  $\tilde{R} \leq \tilde{C}_{\text{pf}}$  where

$$C_{\text{pf}} = \begin{cases} \frac{2}{3}n, & \alpha < 1, \\ \frac{2}{3}m, & \alpha > 1, \\ n, & \alpha = 1, \end{cases} \quad (9)$$

$$\tilde{C}_{\text{pf}} = \begin{cases} \frac{2}{3}\tilde{n}, & \tilde{\alpha} < 1, \\ \frac{2}{3}\tilde{m}, & \tilde{\alpha} > 1, \\ \tilde{n}, & \tilde{\alpha} = 1. \end{cases} \quad (10)$$

By comparing Theorem 1 and Baseline 1, one can readily see that feedback (that can be provided through interaction) offers a gain (in terms of computation capacity region) as long as  $(\alpha \notin [\frac{2}{3}, \frac{3}{2}], \tilde{\alpha} \notin [\frac{2}{3}, \frac{3}{2}])$ . With Definition 1 below, a further careful inspection reveals that there are channel regimes in which one can enhance  $C_{\text{no}}$  (or  $\tilde{C}_{\text{no}}$ ) without sacrificing the other counterpart. This implies a net interaction gain.

**Definition 1** (Interaction Gain). We say that an interaction gain occurs if one can achieve  $(R, \tilde{R}) = (C_{\text{no}} + \delta, \tilde{C}_{\text{no}} + \tilde{\delta})$  for some  $\delta \geq 0$  and  $\tilde{\delta} \geq 0$  such that  $\max(\delta, \tilde{\delta}) > 0$ . Here a net interaction gain can be quantified as  $(\delta, \tilde{\delta}) := (R, \tilde{R}) - (C_{\text{no}}, \tilde{C}_{\text{no}})$ .

Our earlier work in [2] has demonstrated that an interaction gain occurs in the light blue regime in Fig. 2. We also find the regimes in which feedback does increase computation capacity but interaction cannot provide such increase, meaning that whenever  $\delta > 0$ ,  $\tilde{\delta}$  must be  $-\delta$  and vice versa. The regimes are  $(\alpha < \frac{2}{3}, \tilde{\alpha} < \frac{3}{2})$  and  $(\alpha > \frac{3}{2}, \tilde{\alpha} > \frac{3}{2})$ . One can readily check that this follows from the cut-set bounds (5) and (6).

**Achieving perfect-feedback computation capacities:** It is noteworthy to mention that there exist channel regimes in which both  $\delta$  and  $\tilde{\delta}$  can be strictly positive. This implies that for these regimes, not only feedback does not sacrifice one transmission for the other, but it can actually improve

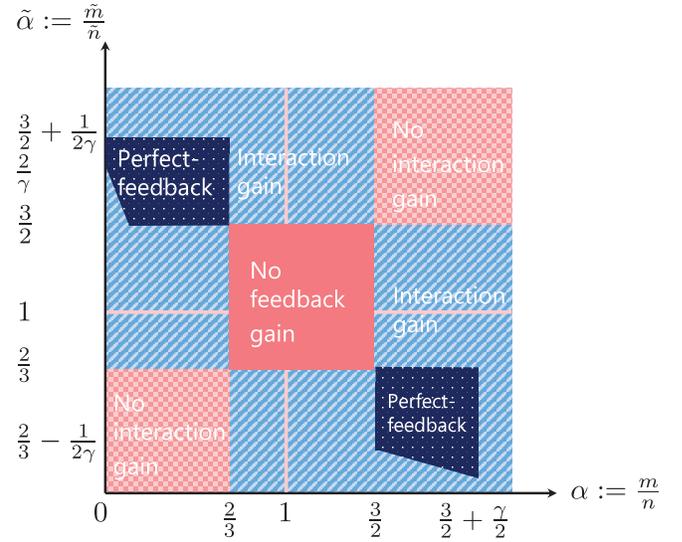


Fig. 2. Gain-vs-nogain picture: The plot is over two key parameters:  $\alpha$  and  $\tilde{\alpha}$ , where  $\alpha$  is the ratio of the interference-to-noise ratio (in dB) to the signal-to-noise ratio (in dB) of the forward channel and  $\tilde{\alpha}$  is the corresponding quantity of the backward channel. The parameter  $\gamma$  is the ratio of the backward signal-to-noise ratio (in dB) to the forward signal-to-noise ratio (in dB) and is fixed to be a value greater than or equal to 1 in the plot. Dark pink/blank region: feedback does not increase computation capacity in either direction and thus interaction is not useful. Light pink/check: feedback does increase computation capacity but interaction cannot provide such increase. Light blue/slash: there is a net interaction gain. Dark blue/dots: interaction is so efficient that one can achieve perfect-feedback computation capacities simultaneously in both directions.

both simultaneously. More interestingly, as in the two-way interference channel [24], the gains  $\delta$  and  $\tilde{\delta}$  can reach up to the maximal feedback gains, reflected in  $C_{\text{pf}} - C_{\text{no}}$  and  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}$  respectively. The dark blue/dots regimes in Fig. 2 indicate such channel regimes when  $1 \leq \gamma := \frac{\tilde{n}}{n}$ . Note that such regimes depend on  $\gamma$ . The amount of feedback that one can send is limited according to available resources, which is affected by the channel asymmetry parameter  $\gamma$ . The following corollary identifies channel regimes in which achieving perfect-feedback computation capacities in both directions is possible.

**Corollary 1.** Consider a case in which feedback helps for both forward and backward channels:  $C_{\text{pf}} > C_{\text{no}}$  and  $\tilde{C}_{\text{pf}} > \tilde{C}_{\text{no}}$ . Under such a case, the channel regimes in which  $C = C_{\text{pf}}$  are as follows:

$$(I) \alpha < \frac{2}{3}, \tilde{\alpha} > \frac{3}{2}, \left\{ \begin{array}{l} C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}, \\ \tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}} \end{array} \right\}, \quad (11)$$

$$(II) \alpha > \frac{3}{2}, \tilde{\alpha} < \frac{2}{3}, \left\{ \begin{array}{l} C_{\text{pf}} - C_{\text{no}} \leq \tilde{n} - \tilde{C}_{\text{pf}}, \\ \tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq m - C_{\text{pf}} \end{array} \right\}. \quad (12)$$

*Proof:* A tedious yet straightforward calculation with Theorem 1 completes the proof. ■

**Remark 1** (Why the Perfect-feedback Regimes?). The rationale behind achieving perfect-feedback computation capacities in both directions bears a resemblance to the one found in the two-way interference channel [24]: Interaction enables full-utilization of available resources, whereas the dearth of

interaction limits that of those. Below we elaborate on this for the considered regime in Corollary 1: ( $\alpha \leq \frac{2}{3}, \tilde{\alpha} \geq \frac{3}{2}$ ).

We first note that the total number of available resources for the forward and backward channels depend on  $n$  and  $\tilde{m}$  in this regime. In the non-feedback case, observe from Baseline 1 that some resources are under-utilized; specifically one can interpret  $n - C_{\text{pf}}$  and  $\tilde{m} - \tilde{C}_{\text{pf}}$  as the remaining resource levels that can potentially be utilized to aid function computations. It turns out feedback can maximize resource utilization by filling up such resource holes under-utilized in the non-feedback case. Note that  $C_{\text{pf}} - C_{\text{no}}$  represents the amount of feedback that needs to be sent for achieving  $C_{\text{pf}}$ . Hence, the condition  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$  (similarly  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$ ) in Corollary 1 implies that as long as we have enough resource holes, we can get all the way to perfect-feedback computation capacity. We will later provide an intuition as to why feedback can do so while describing our achievability; see Remark 3 in particular.  $\square$

#### IV. PROOF OF ACHIEVABILITY

Our achievability proof consists of four parts. We initially outline what the key ingredients of our achievability are. For the next two subsections, we provide achievable schemes for two toy examples in which the key ingredients of our achievability idea are well presented. Once the description of the two schemes is done, we will then outline the proof for generalization as to how to combine these case-by-case results to obtain the computation capacity of the overall network. We leave the detailed proof for these in Appendices A, B and C.

##### A. Key Techniques

The key ingredients of the proposed coding schemes entail (1) superposition coding; (2) the idea of interference neutralization [31]; and (3) retrospective [24] (and nested retrospective) decodings. Specifically, in our two-way setting, the transmission of superposition coded signals w.r.t. one direction can simultaneously deliver its own message signals and feedback signals that can aid the transmission w.r.t. the other direction. This has given us an opportunity to boost up the computation rate since message signals and feedback signals are sent using the same resource. Despite the potential efficiency w.r.t. the achievable computation rate, a challenge of “being corrupted” arises due to the hardness of extracting the desired modulo-2 sum functions from the superimposed signals. The idea of interference neutralization now comes into play which enables somewhat further recovering the desired computation results. In addition to this second ingredient, we highlight that retrospective decoding plays a crucial role in completely resolving the challenge. Together with these ingredients, it turns out that our scheme asymptotically approaches the perfect-feedback computation capacity as the number of time slots goes to infinity.

*B. Example 1:  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$*

1) *Perfect-Feedback Strategy:* We first review the perfect-feedback scheme [28], which we will use as a baseline for

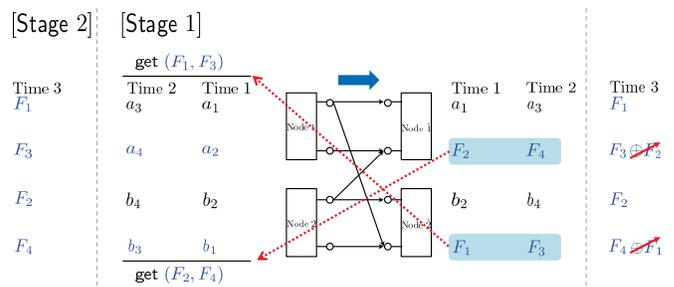


Fig. 3. A perfect-feedback scheme for  $(m, n) = (1, 2)$  model.

comparison to our achievable scheme. It suffices to consider the case of  $(m, n) = (1, 2)$ , as the other case of  $(\tilde{m}, \tilde{n}) = (2, 1)$  follows similarly by symmetry. The perfect-feedback scheme for  $(m, n) = (1, 2)$  consists of two stages; the first stage has two time slots; and the second stage has one time slot. See Fig. 3. Observe that the bottom level at each receiving node naturally forms a modulo-2 sum function, say  $F_\ell$  ( $:= a_\ell \oplus b_\ell$ ). In the first stage, we send forward symbols at nodes 1 and 2. At time 1, node 1 sends  $(a_1, a_2)$ ; and node 2 sends  $(b_2, b_1)$ . Node  $\tilde{1}$  then obtains  $F_2$  ( $:= a_2 \oplus b_2$ ); and node  $\tilde{2}$  obtains  $F_1$ . As in the first time slot, nodes 1 and 2 deliver  $(a_3, a_4)$  and  $(b_4, b_3)$  respectively at time 2. Then nodes  $\tilde{1}$  and  $\tilde{2}$  obtain  $F_4$  and  $F_3$  respectively. Note that until the end of time 2,  $(F_1, F_3)$  are not yet delivered to node  $\tilde{1}$ . Similarly  $(F_2, F_4)$  are missing at node  $\tilde{2}$ .

Feedback can, however, accomplish the computation of these functions of interest. With feedback, each transmitting node can now obtain the desired functions which were obtained only at one receiving node. Exploiting a feedback link from node  $\tilde{2}$  to node 1, node 1 can obtain  $(F_1, F_3)$ . Similarly, node 2 can obtain  $(F_2, F_4)$  from node  $\tilde{1}$ .

The strategy in Stage 2 is to forward all of these feedback functions at time 3. Node  $\tilde{1}$  then receives  $F_1$  cleanly at the top level. At the bottom level, it gets a mixture of the two desired functions:  $F_3 \oplus F_2$ . Note that  $F_2$  in the mixture was already obtained at time 1. Hence, using  $F_2$ , node  $\tilde{1}$  can decode  $F_3$ . Similarly, node  $\tilde{2}$  can obtain  $(F_2, F_4)$ . In summary, nodes  $\tilde{1}$  and  $\tilde{2}$  can compute four modulo-2 sum functions during three time slots, thus achieving  $R = \frac{4}{3}$  ( $= C_{\text{pf}}$ ).

In our model (see Fig. 4), however, feedback is provided in a limited fashion, as feedback signals are delivered only through the backward channel. There are two different types of transmissions for using the backward channel. The channel can be used (1) for backward-message computation, or (2) for sending feedback signals. Usually, unlike the perfect-feedback case, the channel use for one purpose limits that for the other (see Remark 2 for details), and this tension incurs a new challenge. Before getting into our scheme, we will elaborate such tension through the backward channel  $(\tilde{m}, \tilde{n}) = (1, 2)$ , which is different from our case  $(\tilde{m}, \tilde{n}) = (2, 1)$  of interest.

**Remark 2** (One Bit Feedback Through  $(\tilde{m}, \tilde{n}) = (1, 2)$  Costs Exactly One Bit). *From the above strategy, observe that  $F_1$  is delivered from node  $\tilde{2}$  to node 1, while  $F_2$  is delivered from*

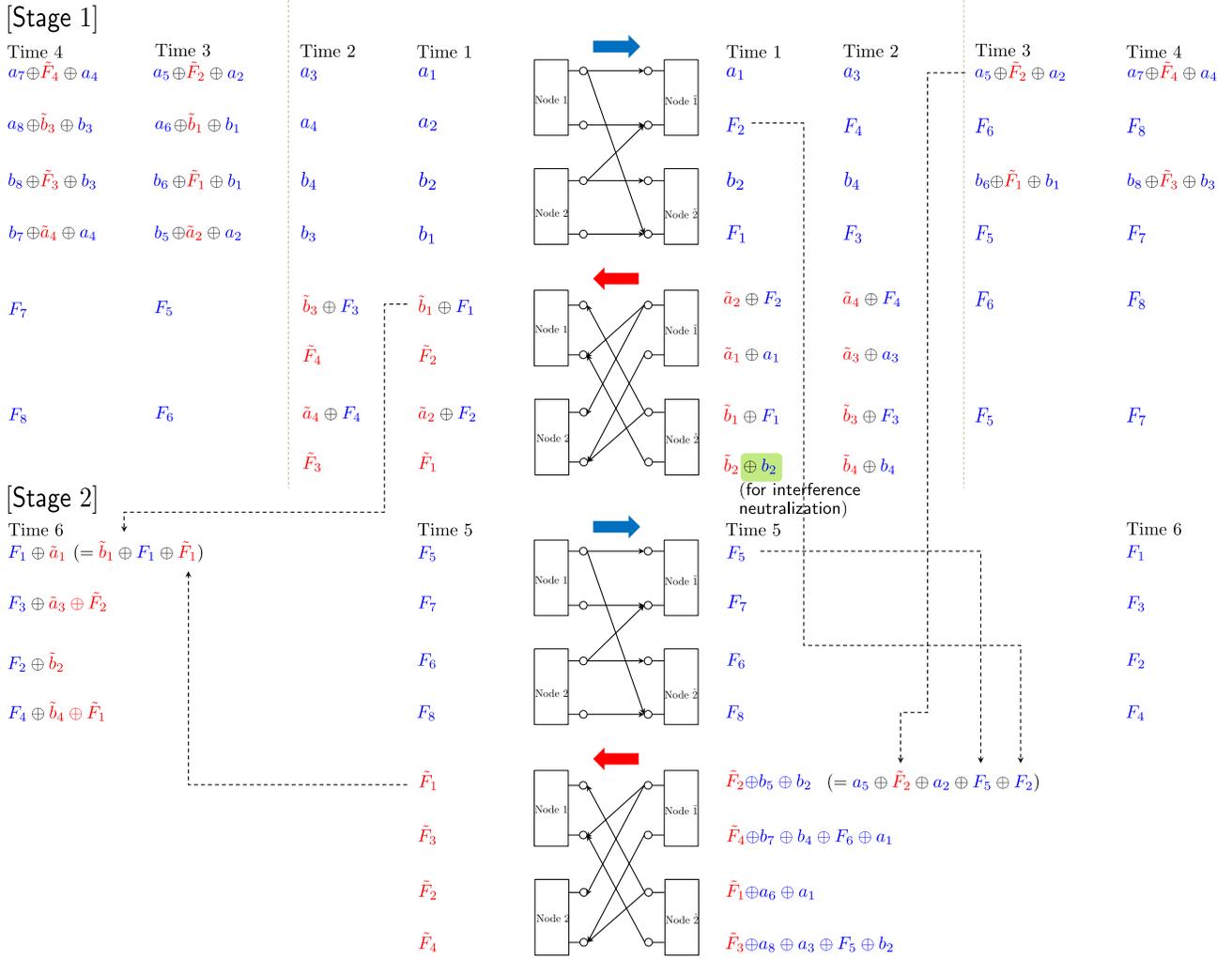


Fig. 4. An achievable scheme for  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$ , and  $L = 2$ .

node  $\tilde{1}$  to node 2. In order to implement this feedback strategy through  $(\tilde{m}, \tilde{n}) = (1, 2)$  (which is a symmetric version of the forward channel described in Fig. 3), one can see that nodes  $\tilde{1}$  and  $\tilde{2}$  must use a cross link. Then, nodes 1 and 2 can receive the desired fed-back functions through the bottom level. However, note that the bottom level at each receiving node naturally forms a modulo-2 sum function. Hence, there is a backward-message computation demand through this bottom level as well. Owing to this tension between feedback transmission and the backward-message computation, we see that there is one-to-one tradeoff between feedback and the backward-message computation. In fact, this is the case where feedback does increase capacity, but interaction cannot provide such gain (i.e., feedback cost equals to feedback gain).

In our achievability for  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$ , however, we develop an achievable scheme that can completely resolve the tension, thus achieving the perfect-feedback performance. See below.

2) *Achievability:* Like the perfect-feedback case, our scheme has two stages. The first stage has  $2L$  time slots; and the second stage has  $L$  time slots. During the first stage, the number  $4L$  forward fresh symbols are transmitted at

nodes 1 and 2; and the number  $4(L - 1)$  of backward fresh symbols are transmitted at nodes  $\tilde{1}$  and  $\tilde{2}$ . No fresh symbols are transmitted at the second stage, but some refinements are performed to recover the desired function computations. In this example, we claim that the following computation rate pair is achievable:  $(R, \tilde{R}) = (\frac{4}{3}, \frac{4L-4}{3L})$ . In other words, during the total  $3L$  time slots, our scheme ensures  $4L$  forward and  $4L - 4$  backward message computations. As  $L \rightarrow \infty$ , we obtain the desired result:  $(R, \tilde{R}) \rightarrow (\frac{4}{3}, \frac{4}{3}) = (C_{\text{pf}}, \tilde{C}_{\text{pf}})$ .

**Stage 1:** The purpose of this stage is to compute  $2L$  and  $2(L - 1)$  modulo-2 sum functions on the bottom level of forward and backward channels, while relaying feedback signals (as in the perfect feedback case) on the top level. To this end, we employ (1) superposition coding (i.e., the first idea that we introduced in Section IV-A): Each node superimposes fresh symbols and feedback symbols. For ease of understanding, we focus on the case where  $L = 2$ . Also see Fig. 4. It turns out that one can readily extend this scheme for the large  $L$ . For an arbitrary  $L$ , we leave detailed scheme in Appendix A.

*Time 1 & 2:* Node 1 sends  $(a_1, a_2)$ ; and node 2 sends  $(b_2, b_1)$ . Nodes  $\tilde{1}$  and  $\tilde{2}$  then receive  $(a_1, F_2)$  and  $(b_2, F_1)$  respectively. Observe that  $F_1$  and  $F_2$  have not yet been

delivered to nodes  $\tilde{1}$  and  $\tilde{2}$  respectively. In the backward channel, a similar transmission strategy is employed in for the backward-message computation. Nodes  $\tilde{1}$  and  $\tilde{2}$  wish to transmit fresh backward symbols:  $(\tilde{a}_2, \tilde{a}_1)$  and  $(\tilde{b}_1, \tilde{b}_2)$  so that nodes 1 and 2 can compute  $(\tilde{b}_1, \tilde{F}_2)$  and  $(\tilde{a}_2, \tilde{F}_1)$ . However, feedback transmission over the backward channel must be accomplished in order to achieve the forward perfect-feedback computation capacity. Recall that in the perfect-feedback strategy, the received signals  $F_2$  and  $F_1$  are desired to be fed back. One way to accomplish both tasks is to superimpose feedback signals onto fresh symbols. Specifically nodes  $\tilde{1}$  and  $\tilde{2}$  encode  $\tilde{a}_2 \oplus F_2$  and  $\tilde{b}_1 \oplus F_1$  on the top level respectively. Then, a challenge arises if these signals are transmitted without additional encoding procedure. Observe that node 1 would receive  $\tilde{F}_2 \oplus F_2$ , while the original goal is to compute the backward functions solely on the bottom level. In other words, the feedback signal  $F_2$  causes interference to node 1, because there is no way to cancel out this signal.

Interestingly, the idea of (2) *interference neutralization* [31] can play a role. On the bottom level, node  $\tilde{2}$  sending the mixture of  $\tilde{b}_2$  (fresh symbol) and  $b_2$  (received on the top level) enables the interference to be neutralized. This allows node 1 to obtain  $\tilde{F}_2 \oplus a_2$ , which in turn leads node 1 to obtain  $\tilde{F}_2$  by canceling  $a_2$  (own symbol). Similarly node  $\tilde{1}$  delivers  $(\tilde{a}_2 \oplus F_2, \tilde{a}_1 \oplus a_1)$ . As a result, nodes 1 and 2 can obtain  $(\tilde{b}_1 \oplus F_1, \tilde{F}_2)$  and  $(\tilde{a}_2 \oplus F_2, \tilde{F}_1)$  respectively.

At time 2, we repeat this w.r.t. new symbols. As a result, nodes  $\tilde{1}$  and  $\tilde{2}$  receive  $(a_3, F_4)$  and  $(b_4, F_3)$  respectively, while nodes 1 and 2 receive  $(\tilde{b}_3 \oplus F_3, \tilde{F}_4 \oplus a_4)$  and  $(\tilde{a}_4 \oplus F_4, \tilde{F}_3 \oplus b_3)$ . Similar to the first time slot, nodes 1 and 2 utilize their own symbols as side information to obtain  $\tilde{F}_4$  and  $\tilde{F}_3$  respectively.

Using the fed-back  $\tilde{F}_2 \oplus a_2$  (received at time 1), node 1 now delivers  $a_5 \oplus \tilde{F}_2 \oplus a_2$  on top at time 3; furthermore, using  $\tilde{b}_1 \oplus F_1$  (also received at time 1) and  $a_1$  (own symbol), node 1 also delivers  $a_6 \oplus \tilde{b}_1 \oplus b_1$  on bottom. With a similar strategy, node 2 delivers  $(b_6 \oplus \tilde{F}_1 \oplus b_1, b_5 \oplus \tilde{a}_2 \oplus a_2)$  at the same time. Then, node  $\tilde{1}$  receives  $(a_5 \oplus \tilde{F}_2 \oplus a_2, F_6 \oplus \tilde{a}_1)$ , while node  $\tilde{2}$  receives  $(b_6 \oplus \tilde{F}_1 \oplus b_1, F_5 \oplus \tilde{b}_2)$ . Note that using their own symbols  $\tilde{a}_1$  and  $\tilde{b}_2$ , nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain  $F_6$  and  $F_5$  respectively. At time 4, we repeat the same process w.r.t. new symbols. As a result, nodes  $\tilde{1}$  and  $\tilde{2}$  obtain  $(a_7 \oplus \tilde{F}_4 \oplus a_4, F_8)$  and  $(b_8 \oplus \tilde{F}_3 \oplus b_3, F_7)$ . During these two time slots, nodes  $\tilde{1}$  and  $\tilde{2}$  do not send any fresh backward symbols. Instead, they mimic the perfect-feedback strategy; that is, through the top level, node  $\tilde{1}$  feeds back  $(F_6, F_8)$ , while node  $\tilde{2}$  feeds back  $(F_5, F_7)$ .

Note that until the end of time 4,  $(F_1, F_3, F_5, F_7)$  and  $(F_2, F_4, F_6, F_8)$  are not yet delivered to nodes  $\tilde{1}$  and  $\tilde{2}$ , while  $(\tilde{F}_1, \tilde{F}_3)$  and  $(\tilde{F}_2, \tilde{F}_4)$  are missing at nodes 1 and 2 respectively.

**Stage 2:** During the next two time slots at the second stage, we accomplish the computation of the desired functions not yet obtained by each node. Recall that the transmission strategy in the perfect-feedback scenario is simply to forward all of the received signals at each node. The received signals are in the form of modulo-2 sum functions of interest (see Fig. 3). In our model, however, the received signals include symbols generated from the other-side nodes. See Fig. 4. For instance,

the received signal at node 1 in time 1 is  $\tilde{b}_1 \oplus F_1$ , which contains the backward symbol  $\tilde{b}_1$ . Hence, unlike the perfect-feedback scheme, forwarding the signal directly from node 1 to node  $\tilde{1}$  is not guaranteed for node  $\tilde{1}$  to decode the desired function  $F_1$ .

To address this, we introduce a recently developed approach [24]: *Retrospective decoding*. The key feature of this approach is that the successive refinement is done in a retrospective manner, allowing us to resolve the aforementioned issue. The outline of the strategy is as follows: Nodes  $\tilde{1}$  and  $\tilde{2}$  start to decode  $(F_5, F_7)$  and  $(F_6, F_8)$  respectively. Here one key point to emphasize is that these decoded functions act as *side information*. Ultimately, this information enables the other-side nodes to obtain the desired functions w.r.t. the past symbols. Specifically the decoding order reads backward:

$$(F_5, F_6, F_7, F_8) \rightarrow (\tilde{F}_1, \tilde{F}_2, \tilde{F}_3, \tilde{F}_4) \rightarrow (F_1, F_2, F_3, F_4).$$

In order for nodes  $\tilde{1}$  and  $\tilde{2}$  to decode the first set of desired functions, the transmission strategy of nodes 1 and 2 at time 5 is to deliver the received functions. In other words, they forward  $(F_5, F_7)$  and  $(F_6, F_8)$  (received at time 3 and 4) respectively. Then nodes  $\tilde{1}$  and  $\tilde{2}$  obtain  $(F_5, F_7 \oplus F_6)$  and  $(F_6, F_8 \oplus F_5)$ . Using  $F_6$  (received at time 3), node  $\tilde{1}$  can decode  $F_7$ . Similarly node  $\tilde{2}$  can decode  $F_8$ . As mentioned above, now the idea in the backward channel is to exploit the newly decoded  $F_5$ . Using  $F_2$  (received at time 1) and  $a_5 \oplus \tilde{F}_2 \oplus a_2$  (received at time 3) further, node  $\tilde{1}$  can construct:

$$\tilde{F}_2 \oplus b_5 \oplus b_2 = F_5 \oplus F_2 \oplus (a_5 \oplus \tilde{F}_2 \oplus a_2).$$

This constructed signal is sent at the top level. Furthermore, with the newly decoded  $F_7$ ,  $(a_1, F_4, F_6)$  (received at time 1, 2 and 3) and  $a_7 \oplus \tilde{F}_4 \oplus a_4$  (received at time 4), node  $\tilde{1}$  can construct:

$$\begin{aligned} \tilde{F}_4 \oplus b_7 \oplus b_4 \oplus F_6 \oplus a_1 \\ = F_7 \oplus a_1 \oplus F_4 \oplus F_6 \oplus (a_7 \oplus \tilde{F}_4 \oplus a_4). \end{aligned}$$

This is sent at the bottom level. In a similar manner, node  $\tilde{2}$  encodes  $(\tilde{F}_1 \oplus a_6 \oplus a_1, \tilde{F}_3 \oplus a_8 \oplus a_3 \oplus F_5 \oplus b_2)$ . Sending all of the encoded signals, nodes 1 and 2 then receive  $(\tilde{F}_1 \oplus a_6 \oplus a_1, \tilde{F}_3 \oplus \tilde{F}_2 \oplus a_8 \oplus a_3 \oplus a_5)$  and  $(\tilde{F}_2 \oplus b_5 \oplus b_2, \tilde{F}_4 \oplus \tilde{F}_1 \oplus b_7 \oplus b_4 \oplus b_6)$  respectively.

Observe that from the top level, node 1 can finally decode  $\tilde{F}_1$  of interest using  $(a_6, a_1)$  (own symbols). From the bottom level, node 1 can also obtain  $\tilde{F}_3$  from  $\tilde{F}_3 \oplus \tilde{F}_2 \oplus a_8 \oplus a_3 \oplus a_5$  by utilizing  $\tilde{F}_2$  (received at time 1) and  $(a_8, a_3, a_5)$  (own symbols). Similarly, node 2 can decode  $(\tilde{F}_2, \tilde{F}_4)$ .

With the help of these decoded functions, nodes 1 and 2 can then construct signals that can aid in the decoding of the desired functions at the other-side nodes. Node 1 uses newly decoded  $\tilde{F}_1$  and  $\tilde{b}_1 \oplus F_1$  (received at time 1) to generate  $F_1 \oplus \tilde{a}_1$  on the top level; using  $(\tilde{b}_3 \oplus F_3, \tilde{F}_2, \tilde{F}_3)$ , it also constructs  $F_3 \oplus \tilde{a}_3 \oplus \tilde{F}_2$  on the bottom level. In a similar manner, node 2 encodes  $(F_2 \oplus \tilde{b}_2, F_4 \oplus \tilde{b}_4 \oplus \tilde{F}_1)$ . Forwarding all of these signals at time 6, nodes  $\tilde{1}$  and  $\tilde{2}$  receive  $(F_1 \oplus \tilde{a}_1, F_3 \oplus F_2 \oplus \tilde{a}_3 \oplus \tilde{a}_2)$  and  $(F_2 \oplus \tilde{b}_2, F_4 \oplus F_1 \oplus \tilde{b}_4 \oplus \tilde{b}_1)$  respectively. Here using their past decoded functions and own symbols, nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain  $(F_1, F_3)$  and  $(F_2, F_4)$ .

Consequently, during six time slots, eight modulo-2 sum functions w.r.t. forward symbols are computed, while four backward functions are computed. This gives  $(R, \tilde{R}) = (\frac{4}{3}, \frac{2}{3})$ . By further transmitting fresh symbols and feedback signals at Stage 1 and applying retrospective decoding idea at Stage 2, one can see that  $(R, \tilde{R}) = (\frac{4}{3}, \frac{4L-4}{3L})$  is achievable. Note that as  $L \rightarrow \infty$ , we get the desired computation rate pair:  $(R, \tilde{R}) \rightarrow (\frac{4}{3}, \frac{4}{3}) = (C_{\text{pf}}, \tilde{C}_{\text{pf}})$ . We present the scheme for arbitrary  $L$  in Appendix A.

**Remark 3** (How to achieve the perfect-feedback bound?). As in the two-way interference channel [24], the key point in our achievability lies in exploiting the following three types of information as side information: (1) past received signals; (2) own message symbols; and (3) future decoded functions. Recall that in our achievability in Fig. 4, the encoding strategy is to combine own symbols with past received signals, e.g., at time 1 node  $\tilde{1}$  encodes  $(\tilde{a}_2 \oplus F_2, \tilde{a}_1 \oplus a_1)$ , which is the mixture of its own symbols  $(\tilde{a}_2, \tilde{a}_1)$  and the received signals  $(F_2, a_1)$ . The decoding strategy is to utilize past received signals, e.g., at time 1, node 1 exploits its own symbol  $a_2$  to decode  $\tilde{F}_2$ .

The most interesting part that is also highlighted in the two-way interference channel [24] is the utilization of the last type of information: Future decoded functions. For instance, with  $\tilde{b}_1 \oplus F_1$  (received at time 1) only, node 1 cannot help node  $\tilde{1}$  to decode  $F_1$ . However, note that our strategy is to forward  $F_1 \oplus \tilde{a}_1$  at node 1 at time 6. Here the signal is the summation of  $\tilde{b}_1 \oplus F_1$  and  $\tilde{F}_1$ . Additionally,  $\tilde{F}_1$  is in fact the function that node 1 wishes to decode in the end; it can be viewed as a future function because it is not available at time 1. Thus, the approach is to defer the decoding procedure for  $F_1$  until  $\tilde{F}_1$  becomes available at node 1; note in Fig. 4 that  $\tilde{F}_1$  is computed at time 5 (a deferred time slot) in the second stage. The decoding procedure for  $F_3$  and  $(F_2, F_4)$  at nodes  $\tilde{1}$  and  $\tilde{2}$  proceeds similarly as follows: Deferring the decoding of these functions until  $\tilde{F}_3$  and  $(\tilde{F}_2, \tilde{F}_4)$  become available at nodes 1 and 2 respectively. Note that the decoding of  $(F_5, F_7)$  and  $(F_6, F_8)$  at nodes  $\tilde{1}$  and  $\tilde{2}$  precedes that of  $(\tilde{F}_1, \tilde{F}_3)$  and  $(\tilde{F}_2, \tilde{F}_4)$  at nodes 1 and 2 respectively. The idea of deferring the refinement together with the retrospective decoding plays a key role in achieving the perfect-feedback bound in the limit of  $L$ .

C. Example 2:  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (1, 0)$

1) *Perfect-Feedback Strategy*: Similar to the previous example, we first review the perfect-feedback scheme presented in our earlier work [28], which we will use as a baseline for comparison with our achievable scheme. We focus on the case of  $(\tilde{m}, \tilde{n}) = (1, 0)$ , as that for  $(m, n) = (1, 2)$  was already presented. The perfect-feedback scheme for  $(\tilde{m}, \tilde{n}) = (1, 0)$  consists of two stages; the first stage has one time slot; and the second stage has two time slots. At time 1, we send backward symbols  $\tilde{a}_1$  and  $\tilde{b}_2$  at nodes  $\tilde{1}$  and  $\tilde{2}$  respectively. Then nodes 1 and 2 receive  $\tilde{b}_2$  and  $\tilde{a}_1$  respectively. Node 1 can then deliver the received symbol  $\tilde{b}_2$  to node  $\tilde{1}$  through feedback. Similarly, node  $\tilde{2}$  can obtain  $\tilde{a}_1$  from node 2.

At time 2 (the first time of Stage 2), with the feedback signals, nodes  $\tilde{1}$  and  $\tilde{2}$  can construct  $\tilde{F}_2$  and  $\tilde{F}_1$  respectively

and send them over the backward channel. Then nodes 1 and 2 obtain  $\tilde{F}_1$  and  $\tilde{F}_2$  respectively. Note that until the end of time 2,  $\tilde{F}_2$  is not delivered to node 1. Similarly,  $\tilde{F}_1$  is missing at node 2. Using one more time slot, we can deliver these functions to the intended nodes. With feedback, node  $\tilde{2}$  can obtain  $\tilde{F}_2$  from node 2. Sending this at time 3 allows node 1 to obtain  $\tilde{F}_2$ . Similarly, node 2 can obtain  $\tilde{F}_1$ . As a result, nodes 1 and 2 obtain  $(\tilde{F}_1, \tilde{F}_2)$  during three time slots. This gives a computation rate of  $\frac{2}{3}$  ( $= \tilde{C}_{\text{pf}}$ ). We note that compared to the example  $(\tilde{m}, \tilde{n}) = (2, 1)$  (the prior perfect-feedback case), the current strategy does not finish the decoding procedure at Stage 2 in one shot. Rather, it needs one more time slot for relaying and computing the desired functions.

2) *Achievability*: In the two-way setting, a challenge arises due to the tension between feedback transmission and traffic w.r.t. the other direction. The underlying idea to resolve this challenge is similar to that for  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$ . However, one noticeable distinction relative to Example 1 is that the retrospective decoding occurs in a *nested manner*. It was found that this phenomenon occurs due to the fact that the decoding procedure of backward functions at the second stage is not done in one shot (recall the above perfect-feedback scheme); it needs additional time for relaying and computing the desired functions. Hence the decoding of the functions of interest w.r.t. fresh message symbols generated during one stage may not be completed in the very next stage.

Our achievability now introduces the concept of multiple layers, say  $M$  layers. Each layer consists of two stages as in Example 1. Hence there are  $2M$  stages overall. For each layer, the first stage consists of  $2L$  time slots; and the second stage consists of  $L + 1$  time slots. For the first stage of each layer,  $4L$  and  $2L$  of fresh symbols are transmitted through the forward and backward channels respectively. In the second stage, no fresh forward and backward symbols are transmitted, but some refinements are performed (to be specified later).

Among the total  $4LM$  forward and  $2LM$  backward functions, we claim that our scheme ensures the computation of the  $4L(M - (2^{L+1} - 2L - 2))$  number of forward functions and the  $2L(M - (2^{L+1} - 2L - 2))$  number of backward functions at the end of Layer  $M$ . However, we note that the remaining  $4L(2^{L+1} - 2L - 2)$  forward and  $2L(2^{L+1} - 2L - 2)$  backward functions can be successfully computed as we proceed with our scheme further. At the moment of time  $(3L + 1)M$ , we get the computation rate pair of:

$$\left( \frac{4L(M - (2^{L+1} - 2L - 2))}{(3L + 1)M}, \frac{2L(M - (2^{L+1} - 2L - 2))}{(3L + 1)M} \right). \quad (13)$$

As the scheme is somewhat complicated, we first illustrate the scheme for a simple case  $(L, M) = (2, \infty)$  that well presents the idea of achievability although not achieving the optimal computation rate pair of  $(C_{\text{pt}}, \tilde{C}_{\text{pt}}) = (\frac{4}{3}, \frac{2}{3})$  in this case. The exact achievability for an arbitrary  $(L, M)$  will be presented in Appendix B. One can see from (13) that by setting  $M = (2 + \epsilon)^L$  where  $\epsilon > 0$ , and letting  $L \rightarrow \infty$  with the general scheme, we get the optimal performance:  $(R, \tilde{R}) = (\frac{4}{3}, \frac{2}{3}) = (C_{\text{pt}}, \tilde{C}_{\text{pt}})$ .

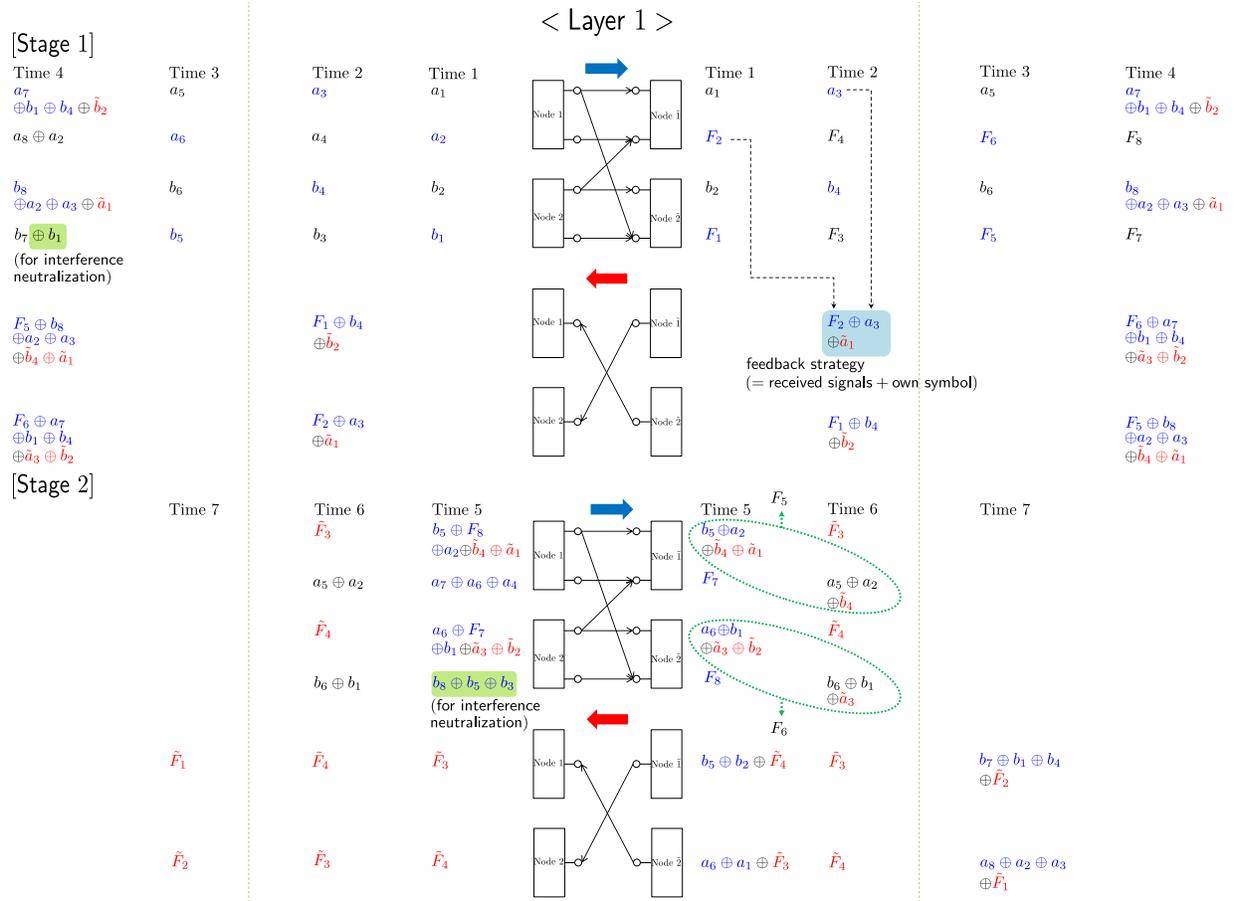


Fig. 5. An achievable scheme for  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (1, 0)$ , and  $(L, M) = (2, \infty)$  in Layer 1.

**Stage 1:** Let us illustrate the scheme for  $(L, M) = (2, \infty)$ . We claim that  $(R, \tilde{R}) = (\frac{8}{7}, \frac{4}{7})$  is achievable, which coincides with (13). The proposed scheme consists of  $7M$  ( $= (3L + 1)M$ ) time slots. And the first stage within the first layer consists of 4 ( $= 2L$ ) time slots. See Fig. 5.

At time 1, node 1 sends  $(a_1, a_2)$ ; node 2 sends  $(b_2, b_1)$ . Then nodes  $\tilde{1}$  and  $\tilde{2}$  receive  $(a_1, F_2)$  and  $(b_2, F_1)$  respectively. Repeating this forward transmission strategy w.r.t. fresh forward symbol at time 2 and 3, nodes  $\tilde{1}$  and  $\tilde{2}$  receive  $(a_3, F_4, a_5, F_6)$  and  $(b_4, F_3, b_6, F_5)$  respectively. Through the backward channel, nodes  $\tilde{1}$  and  $\tilde{2}$  keep silent at time 1 and 3, while they employ a feedback strategy at time 2 in order to send the desired feedback signals and a fresh backward symbol in one shot. Specifically nodes  $\tilde{1}$  and  $\tilde{2}$  deliver  $F_2 \oplus a_3 \oplus \tilde{a}_1$  and  $F_1 \oplus b_4 \oplus \tilde{b}_2$ . Nodes 1 and 2 then get  $F_1 \oplus b_4 \oplus \tilde{b}_2$  and  $F_2 \oplus a_3 \oplus \tilde{a}_1$  respectively.

From the received  $F_1 \oplus b_4 \oplus \tilde{b}_2$ , node 1 cancels out its odd-index symbol  $a_1$  and adds the fresh symbol  $a_7$ , thus encoding  $a_7 \oplus b_1 \oplus b_4 \oplus \tilde{b}_2$ . Similarly, node 2 encodes  $b_8 \oplus a_2 \oplus a_3 \oplus \tilde{a}_1$ . At time 4, nodes 1 and 2 forward the encoded signal on the top level. Furthermore, through the bottom level, each node forwards its own symbols in order to ensure additional function computations at the receiver-side nodes. We note that for each transmitting node, the indices of the transmitted symbols coincide with those of the other

transmitting node's own symbols added and canceled out on the top level during the same period. In particular, node 2 forwards  $b_7 \oplus b_1$  on the bottom level, as node 1 adds  $a_7$  and cancels out  $a_1$  at time 4. Similarly, node 1 forwards  $a_8 \oplus a_2$ . Nodes  $\tilde{1}$  and  $\tilde{2}$  then receive  $(a_7 \oplus b_1 \oplus b_4 \oplus \tilde{b}_2, F_8 \oplus a_3 \oplus \tilde{a}_1)$  and  $(b_8 \oplus a_2 \oplus a_3 \oplus \tilde{a}_1, F_7 \oplus b_4 \oplus \tilde{b}_2)$ . Note that node  $\tilde{1}$  can decode  $F_8$  from  $F_8 \oplus a_3 \oplus \tilde{a}_1$  using  $\tilde{a}_1$  (own symbol) and  $a_3$  (received at time 2). Similarly, node  $\tilde{2}$  can decode  $F_7$ .

Similar to the feedback strategy at time 2, node  $\tilde{1}$  delivers  $F_6 \oplus a_7 \oplus b_1 \oplus b_4 \oplus \tilde{a}_3 \oplus \tilde{b}_2$  which is the mixture of  $F_6$  (received at time 3),  $a_7 \oplus b_1 \oplus b_4 \oplus \tilde{b}_2$  (received at time 4), and  $\tilde{a}_3$  (fresh symbol). Similarly, node  $\tilde{2}$  delivers  $F_5 \oplus b_8 \oplus a_2 \oplus a_3 \oplus \tilde{b}_4 \oplus \tilde{a}_1$ . Nodes 1 and 2 then get  $F_5 \oplus b_8 \oplus a_2 \oplus a_3 \oplus \tilde{b}_4 \oplus \tilde{a}_1$  and  $F_6 \oplus a_7 \oplus b_1 \oplus b_4 \oplus \tilde{a}_3 \oplus \tilde{b}_2$  respectively.

Note that until the end of time 4,  $(F_1, F_3, F_5, F_7)$  and  $(F_2, F_4, F_6, F_8)$  are not yet delivered to nodes  $\tilde{1}$  and  $\tilde{2}$  respectively, while  $(\tilde{F}_1, \tilde{F}_2, \tilde{F}_3, \tilde{F}_4)$  are missing at both nodes 1 and 2.

**Stage 2:** The transmission strategy at the second stage is to accomplish the computation of the desired functions not yet obtained by each node. We employ the retrospective decoding strategy introduced in Example 1. This stage consists of 3 time slots. At time 5, from the signal received at time 4 ( $= 2L$ ), node 1 cancels out all of its odd-index symbols  $(a_3, a_5)$  and adds the even-index symbol  $a_8$  ( $= a_{4L}$ ), thus

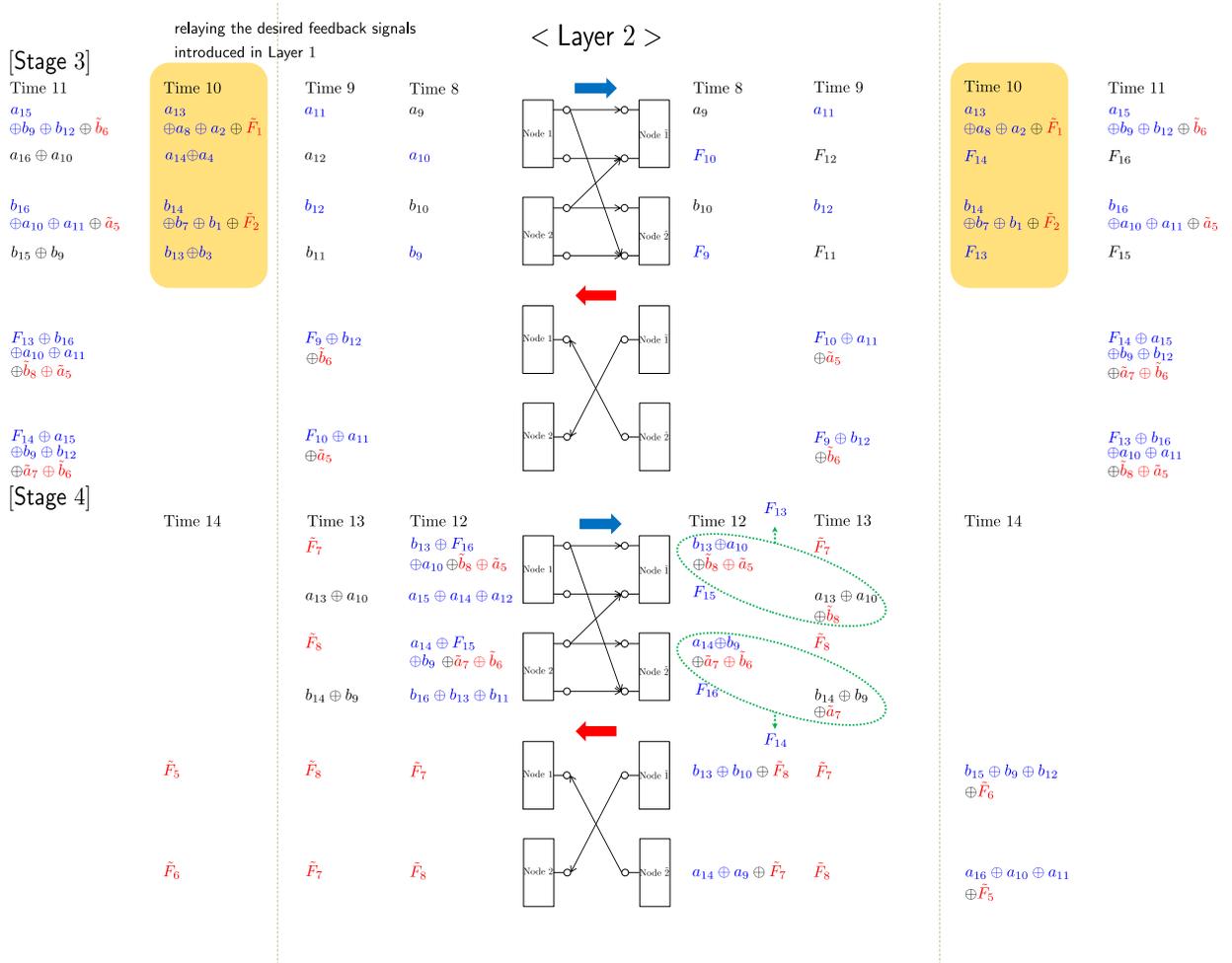


Fig. 6. An achievable scheme for  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (1, 0)$ , and  $(L, M) = (2, \infty)$  in Layer 2.

encoding  $b_5 \oplus F_8 \oplus a_2 \oplus \tilde{b}_4 \oplus \tilde{a}_1$ . In a similar manner, node 2 encodes  $a_6 \oplus F_7 \oplus b_1 \oplus \tilde{a}_3 \oplus \tilde{b}_2$  using even-index symbols  $(b_4, b_6)$  and the odd-index symbol  $b_7$ . The transmission strategy for each node is to forward the encoded signal on the top level.

As in the transmission strategy on the bottom level at time 4, each node forwards its own symbols in order to ensure additional function computations at the other-side nodes. Specifically node 2 forwards  $b_8 \oplus b_5 \oplus b_3$  since node 1 cancels out  $(a_3, a_5)$  and adds  $a_8$  at time 5. Similarly, node 1 forwards  $a_7 \oplus a_6 \oplus a_4$  on the bottom level. Nodes  $\tilde{1}$  and  $\tilde{2}$  then receive:

$$\begin{aligned} \text{node } \tilde{1} : & (b_5 \oplus F_8 \oplus a_2 \oplus \tilde{b}_4 \oplus \tilde{a}_1, b_7 \oplus a_4 \oplus b_1 \oplus \tilde{a}_3 \oplus \tilde{b}_2); \\ \text{node } \tilde{2} : & (a_6 \oplus F_7 \oplus b_1 \oplus \tilde{a}_3 \oplus \tilde{b}_2, a_8 \oplus b_3 \oplus a_2 \oplus \tilde{b}_4 \oplus \tilde{a}_1). \end{aligned}$$

From the received signal on the bottom level, node  $\tilde{1}$  can decode  $F_7 (= F_{4L-1})$  by adding  $a_7 \oplus b_1 \oplus b_4 \oplus \tilde{b}_2$  (received at time 4),  $F_4$  (received at time 2), and  $\tilde{a}_3$  (own symbol). Similarly, node  $\tilde{2}$  can decode  $F_8 (= F_{4L})$ . From the received signal on the top level, nodes  $\tilde{1}$  and  $\tilde{2}$  use  $(F_8, F_2, \tilde{a}_4, \tilde{a}_1)$  and  $(F_7, F_1, \tilde{b}_3, \tilde{b}_2)$  to generate  $b_5 \oplus b_2 \oplus \tilde{F}_4$  and  $a_6 \oplus a_1 \oplus \tilde{F}_3$  respectively. Note that sending them back allows nodes 1 and 2 to obtain  $\tilde{F}_3 (= \tilde{F}_{2L-1})$  and  $\tilde{F}_4 (= \tilde{F}_{2L})$  by canceling  $(a_6, a_1)$  and  $(b_5, b_2)$  (own symbols) respectively.

At time 6, nodes 1 and 2 forward what they just decoded on the top level:  $\tilde{F}_3$  and  $\tilde{F}_4$ . Similar to the transmission strategy on the bottom level at time 5, nodes 1 and 2 additionally forward  $a_5 \oplus a_2$  and  $b_6 \oplus b_1$ . Then nodes  $\tilde{1}$  and  $\tilde{2}$  obtain  $(\tilde{F}_3, a_5 \oplus a_2 \oplus \tilde{F}_4)$  and  $(\tilde{F}_4, b_6 \oplus b_1 \oplus \tilde{F}_3)$  respectively. Observe that node  $\tilde{1}$  can now obtain  $F_5$  by adding  $b_5 \oplus a_2 \oplus \tilde{b}_4 \oplus \tilde{a}_1$  (received on the top level at time 5),  $a_5 \oplus a_2 \oplus \tilde{F}_4$  (received on the bottom level at time 6), and  $(\tilde{a}_4, \tilde{a}_1)$  (own symbols). Similarly, node  $\tilde{2}$  can obtain  $F_6$ . Subsequently, transmitting  $\tilde{F}_3$  and  $\tilde{F}_4$  (received on the top level) over the backward channel enables nodes 1 and 2 to obtain  $\tilde{F}_4$  and  $\tilde{F}_3$  respectively.

Note that until the end of time 6,  $(F_1, F_3)$  and  $(F_2, F_4)$  are not yet delivered to nodes  $\tilde{1}$  and  $\tilde{2}$ , while  $(\tilde{F}_1, \tilde{F}_2)$  is missing at nodes 1 and 2. We have one more time in Stage 2 to resolve this, but unlike the prior example, the decoding of all the remaining functions appears to be impossible during this stage. For instance, with  $F_1 \oplus b_4 \oplus \tilde{b}_2$  (received at time 2) solely, node 1 cannot help node  $\tilde{1}$  to decode  $F_1$ . However, if  $\tilde{F}_2$  is somehow obtained at node 1, it can forward  $F_1 \oplus F_4 \oplus \tilde{a}_2$  (which is the summation of  $F_1 \oplus b_4 \oplus \tilde{b}_2$ ,  $\tilde{F}_2$ , and  $a_4$  (own symbol)), and thus can achieve  $F_1$  at node  $\tilde{1}$  (by canceling  $F_4$  (decoded functions at Stage 1) and  $\tilde{a}_1$  (own symbol)). Note that  $\tilde{F}_2$  is in fact the function that node 1 wishes to decode

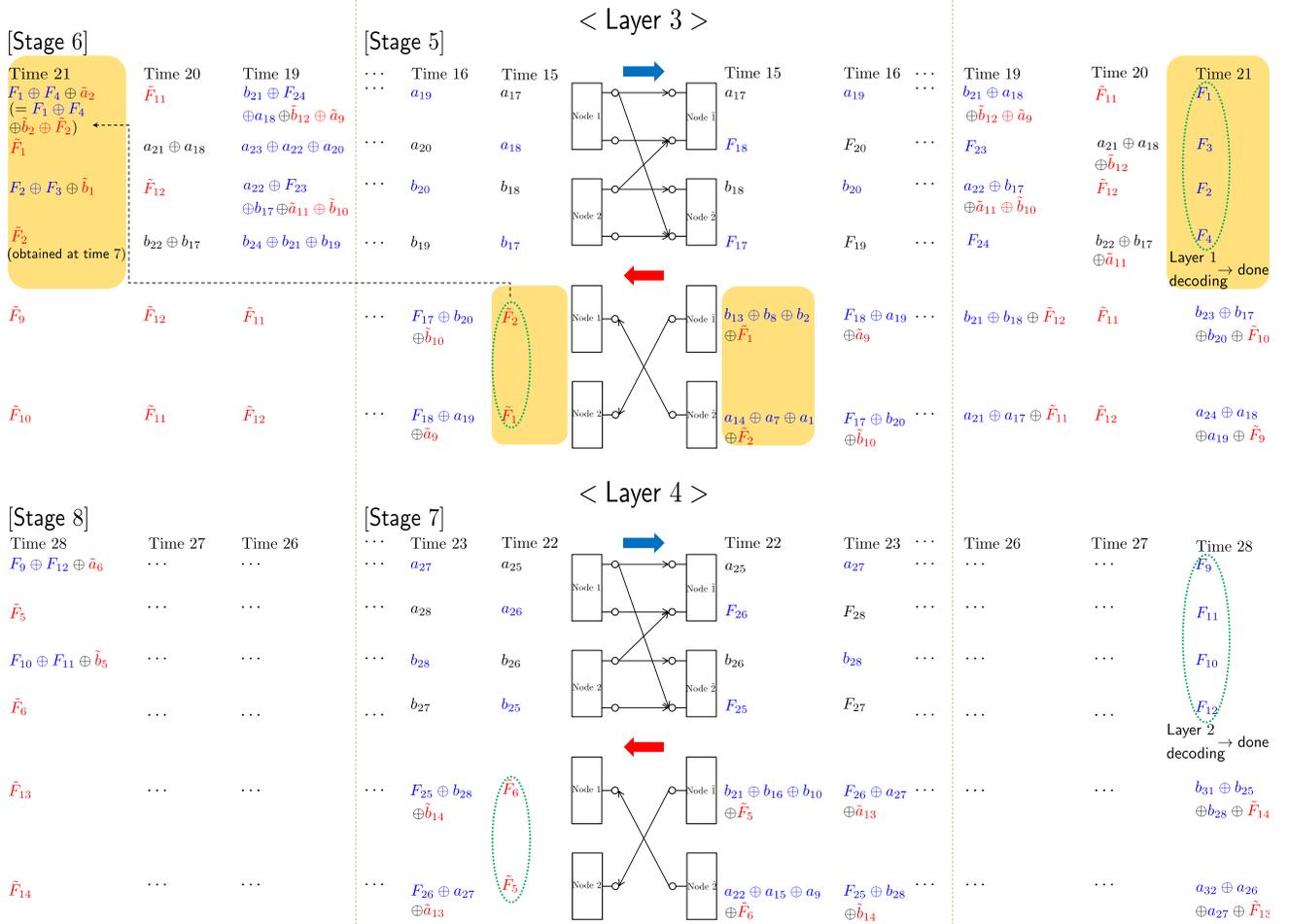


Fig. 7. An achievable scheme for  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (1, 0)$ , and  $(L, M) = (2, \infty)$  in Layer 3 and 4.

in the end; it can be viewed as a future function, as it is not available at the moment. Consequently, the approach is to *additionally* postpone the decoding procedure to another layer. Hence, nodes 1 and 2 remain silent at time 7 and defer the decoding strategy until time 21 (in Layer 3).

Through the backward channel, however, additional backward-message computations are possible via newly-decoded forward functions. With the newly decoded  $F_7$  and  $a_7 \oplus b_1 \oplus b_4 \oplus \tilde{b}_2$  (received at time 4), node  $\tilde{1}$  generates  $b_7 \oplus b_1 \oplus b_4 \oplus \tilde{F}_2$ . Interestingly, sending this through the backward channel allows node 2 to obtain  $\tilde{F}_2$ . Similarly, constructing  $a_8 \oplus a_2 \oplus a_3 \oplus \tilde{F}_1$  and sending this at node  $\tilde{2}$  permits node 1 to obtain  $\tilde{F}_1$ . Nonetheless, one can see that  $\tilde{F}_2$  and  $\tilde{F}_1$  are still missing at nodes 1 and 2 respectively. We will illustrate that these unresolved function computations will be accomplished as we proceed with our scheme further.

**Stage 3 and 4:** The scheme for Layer 2 is essentially identical to that for Layer 1 except for the transmission scheme over the forward channel at time 10. See Fig. 6 (shaded in light yellow).

*Time 10:* The distinction relative to Layer 1 is that nodes 1 and 2 additionally exploit the most recently received signal w.r.t. the previous layer. The purpose of this is to relay signals that can help resolve the unresolved function

computations in Layer 1. Specifically, using  $a_8 \oplus a_2 \oplus a_3 \oplus \tilde{F}_1$  (received at time 7 in Stage 2), node 1 constructs  $a_{13} \oplus a_8 \oplus a_2 \oplus \tilde{F}_1$  and sends it on the top level. The construction idea is to cancel out node 1's odd-index symbol  $a_3$  and to add the fresh symbol  $a_{13}$ . Similarly node 2 constructs  $b_{14} \oplus b_7 \oplus b_1 \oplus \tilde{F}_1$  and sends it on the top level. Then nodes  $\tilde{1}$  and  $\tilde{2}$  receive  $a_{13} \oplus a_8 \oplus a_2 \oplus \tilde{F}_1$  and  $b_{14} \oplus b_7 \oplus b_1 \oplus \tilde{F}_2$ . These relayed signals will be exploited in the next layer to accomplish the computation of  $\tilde{F}_2$  and  $\tilde{F}_1$  (introduced in Layer 1) at nodes 1 and 2 respectively.

Through the bottom level, nodes 1 and 2 transmit additional signals in order to ensure the modulo-2 sum function computation at the other-side nodes. In particular, node 1 transmits  $a_{14} \oplus a_4$ . Then node  $\tilde{1}$  gets  $F_{14} \oplus b_7 \oplus a_4 \oplus b_1 \oplus \tilde{F}_2$ . Using  $b_7 \oplus b_1 \oplus b_4 \oplus \tilde{F}_2$  (the transmitted signal of node  $\tilde{1}$  at time 7) and  $F_4$  (received at time 2), node  $\tilde{1}$  can obtain  $F_{14}$ . Similarly, transmitting  $b_{13} \oplus b_3$  at node 2 ensures node  $\tilde{2}$  to obtain  $F_{13}$ .

Similar to the case of Layer 1, at the end of time 14 in Layer 2, one can see that  $(F_9, F_{11})$  and  $(F_{10}, F_{12})$  are not yet delivered to nodes  $\tilde{1}$  and  $\tilde{2}$ , while  $\tilde{F}_6$  and  $\tilde{F}_5$  are missing at nodes 1 and 2 respectively. We will resolve these computations later.

**Stage 5 and 6:** The scheme for Layer 3 is identical to that for Layer 2 except for two parts: the transmission scheme over

the backward channel at time 15; and that over the forward channel at time 21. See Fig. 7.

*Time 15:* The first distinction relative to Layer 2 is the transmitted signals at node  $\tilde{1}$  and  $\tilde{2}$ :

$$\text{node } \tilde{1} : b_{13} \oplus b_8 \oplus b_2 \oplus \tilde{F}_1;$$

$$\text{node } \tilde{2} : a_{14} \oplus a_7 \oplus a_1 \oplus \tilde{F}_2.$$

The construction idea of these signals is to use the relayed signals, the newly decoded functions in Layer 2, and previously decoded functions. For instance,  $b_{13} \oplus b_8 \oplus b_2 \oplus \tilde{F}_1$  is the summation of  $a_{13} \oplus a_8 \oplus a_2 \oplus \tilde{F}_1$  (received at time 10) and  $(F_{13}, F_8, F_2)$  (decoded at time 12, 4, and 1). One can see that nodes 1 and 2 can now obtain  $\tilde{F}_2$  and  $\tilde{F}_1$  using their own symbols. We find that all of the backward functions introduced in Layer 1 are successfully computed at nodes 1 and 2.

*Time 21:* Here we accomplish the remaining function computation demands introduced in Layer 1. The idea is to exploit  $\tilde{F}_2$  and  $\tilde{F}_1$  decoded at time 15. Using  $\tilde{F}_2, F_1 \oplus b_4 \oplus \tilde{b}_2$  (received at time 2), and  $a_4$  (own symbol), node 1 encodes  $F_1 \oplus F_4 \oplus \tilde{a}_2$  and sends it on the top level. One can see that node  $\tilde{1}$  can obtain  $F_1$  by canceling  $F_4$  (decoded at time 2) and  $\tilde{a}_2$  (own symbol). In a similar manner, constructing  $F_2 \oplus F_3 \oplus \tilde{b}_1$  and delivering it on the top level enables node  $\tilde{2}$  to obtain  $F_2$ . In order to achieve additional modulo-2 sum computations at the same time, nodes 1 and 2 deliver  $\tilde{F}_1$  and  $\tilde{F}_2$  (obtained at time 7) on the bottom level. It is found that applying a similar decoding strategy ensures nodes  $\tilde{1}$  and  $\tilde{2}$  to obtain  $F_3$  and  $F_4$  respectively.

Note that all of the function computations w.r.t. the symbols introduced in Layer 1 are accomplished. In other words, nodes  $\tilde{1}$  and  $\tilde{2}$  obtain  $\{F_\ell\}_{\ell=1}^8$ , while nodes 1 and 2 obtain  $\{\tilde{F}_\ell\}_{\ell=1}^4$ .

**Stage 7 and 8:** We repeat the same procedure as before. Note that the strategy at time 28 in Layer 4 is identical to that at time 21 in Layer 3. In turn, all of the function computation demands introduced in Layer 2 are perfectly accomplished. In other words, nodes  $\tilde{1}$  and  $\tilde{2}$  obtain  $\{F_\ell\}_{\ell=9}^{16}$ , while nodes 1 and 2 obtain  $\{\tilde{F}_\ell\}_{\ell=5}^8$ .

As we proceed with our scheme, one can see that all of the function computation demands introduced in Layer  $i - 2$  can be completely accomplished at the end of Layer  $i$ . At the end of Layer  $M$ , i.e., time  $7M$  ( $= (3L + 1)M$ ), nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain  $\{F_\ell\}_{\ell=1}^{8(M-2)}$ , while nodes 1 and 2 can obtain  $\{\tilde{F}_\ell\}_{\ell=1}^{4(M-2)}$ . This yields  $(R, \tilde{R}) = \left(\frac{8(M-2)}{7M}, \frac{4(M-2)}{7M}\right)$ . As  $M$  tends to infinity, the scheme can achieve  $(\frac{8}{7}, \frac{4}{7})$ . Following the aforementioned strategy, we find that this idea can be extended to arbitrary values of  $(L, M)$ , thus yielding:  $(R, \tilde{R}) = \left(\frac{4L(M-(2^{L+1}-2L-2))}{(3L+1)M}, \frac{2L(M-(2^{L+1}-2L-2))}{(3L+1)M}\right)$ . We present details about the scheme for an arbitrary  $(L, M)$  in Appendix B.

**Remark 4** (Why nested retrospective decoding can achieve the desired performance?). *In Stage 2 of our scheme (see Fig. 5), we see that feedback-aided successive refinement w.r.t. the fresh symbols sent previously enables each node to compute additional functions; however, each node could not compute all of the desired functions within the current layer.*

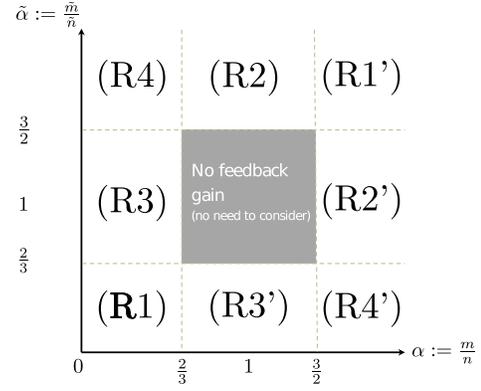


Fig. 8. Regimes to check for achievability proof. By symmetry, it suffices to consider (R1), (R2), (R3), and (R4).

*Our scheme at time 7 in Layer 1 for the forward channel is to remain silent and defer the desired function computations. This vacant time slot causes inefficiency in the performance.*

*The good news is that additional relaying of functions of interest in Layer 2 (see time 10 in Fig. 6) enables an additional forward channel use at the second stage of Layer 3 (see time 21 in Fig. 7). In particular, nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain  $(F_1, F_3)$  and  $(F_2, F_4)$  through this channel use. And from Layer 3, one can see that the second stage of each layer is fully packed. From this observation, we can conclude that the sum of the vacant time slots is finite. Therefore, we can make the inefficiency stemming from the vacant time slots negligible by setting  $M \rightarrow \infty$ . Similar to Example 1, it is found that by setting  $L \rightarrow \infty$ , we can eventually achieve the optimal performance. See details in Appendix B.*

#### D. Proof Outline

We now prove the achievability for arbitrary values of  $(m, n)$ ,  $(\tilde{m}, \tilde{n})$ . Note that  $C = C_{\text{no}}$  when  $((\alpha \in [\frac{2}{3}, 1), \alpha \in (1, \frac{3}{2}]), (\tilde{\alpha} \in [\frac{2}{3}, 1), \tilde{\alpha} \in (1, \frac{3}{2}]))$ . Furthermore, for the case where  $\alpha = 1$ ,  $C = n$  and one can see that sacrificing the forward transmission for aiding the backward-message transmission (by sending feedback signals on forward link levels) incurs one-to-one trade-off unless  $\tilde{\alpha} \leq \frac{2}{3}$  or  $\tilde{\alpha} > \frac{3}{2}$ . The case for  $\tilde{\alpha} = 1$  similarly follows. We rule out these three cases since the proofs for them are somewhat straightforward. In addition to these cases, by symmetry, it suffices to consider the following four regimes. See Fig. 8:

$$(R1) \alpha \leq 2/3, \tilde{\alpha} \leq 2/3; \quad (14)$$

$$(R2) (\alpha \in [2/3, 1), \alpha \in (1, 3/2]), \tilde{\alpha} \geq 3/2; \quad (15)$$

$$(R3) \alpha \leq 2/3, (\tilde{\alpha} \in [2/3, 1), \tilde{\alpha} \in (1, 3/2]); \quad (16)$$

$$(R4) \alpha \leq 2/3, \tilde{\alpha} \geq 3/2. \quad (17)$$

*1) Regimes in Which Interaction Provides No Gain:* Referring to Fig. 2, the channel regimes of this category are (R1), (R1'), and  $(\alpha = 1 \text{ or } \tilde{\alpha} = 1)$ . A simple combination of the non-feedback scheme [30] and the interactive scheme in [2] can yield the desired result for the regimes.

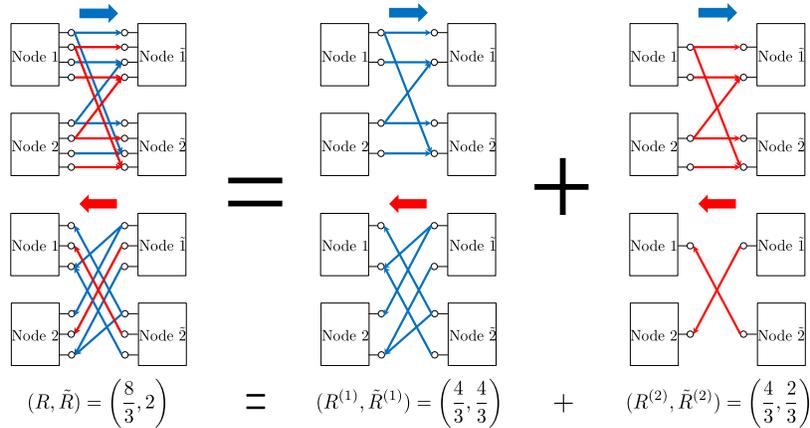


Fig. 9. A network decomposition example of  $(m, n) = (2, 4)$ ,  $(\tilde{m}, \tilde{n}) = (3, 1)$  model. The decomposition is given by  $(2, 4), (3, 1) \rightarrow (1, 2), (2, 1) \times (1, 2), (1, 0)$ .

2) *Regimes in Which Interaction Helps Only Either in Forward or Backward Direction:* It is found that the achievability in this case is also a simple combination of the non-feedback scheme [30] and the interactive scheme in [2]. The channel regimes of this category are: (R2), (R2'), (R3), and (R3').

3) *Regimes in Which Interaction Helps Both in Forward and Backward Directions:* As mentioned earlier, the key idea is to employ the retrospective decoding. For ease of generalization to arbitrary channel parameters in the regime, here we employ the network decomposition method [30] where an original network is decomposed into elementary orthogonal subnetworks and achievable schemes are applied separately into the subnetworks. See Fig. 9 for an example of such network decomposition. The idea is to use graph coloring. The figure graphically proves the fact that  $(m, n) = (2, 4)$ ,  $(\tilde{m}, \tilde{n}) = (3, 1)$  model can be decomposed into the following two orthogonal subnetworks:  $(m^{(1)}, n^{(1)}) = (1, 2)$ ,  $(\tilde{m}^{(1)}, \tilde{n}^{(1)}) = (2, 1)$  model (blue color); and  $(m^{(2)}, n^{(2)}) = (1, 2)$ ,  $(\tilde{m}^{(2)}, \tilde{n}^{(2)}) = (1, 0)$  model (red color). Note that the original network is simply a concatenation of these two subnetworks. We denote the decomposition by  $(2, 4), (3, 1) \rightarrow (1, 2), (2, 1) \times (1, 2), (1, 0)$ . As mentioned earlier, the idea is simply to apply the developed achievable schemes separately for the two subnetworks. Notice that we developed the schemes for  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$  and  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (1, 0)$  model. For the case of  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$ , our proposed scheme achieves  $(R, \tilde{R}) = (\frac{4}{3}, \frac{4L-4}{3L})$ . And for the case of  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (1, 0)$ , our strategy achieves  $(R, \tilde{R}) = (\frac{4L(M-(2^{L+1}-2L-2))}{(3L+1)M}, \frac{2L(M-(2^{L+1}-2L-2))}{(3L+1)M})$ . Setting  $M = (2 + \epsilon)^L$ ,  $\epsilon > 0$ , and letting  $L \rightarrow \infty$ , the first scheme achieves  $(\frac{4}{3}, \frac{4}{3})$ , while the second achieves  $(\frac{4}{3}, \frac{2}{3})$ . Thus, the separation approach gives:

$$(R, \tilde{R}) = \left(\frac{4}{3}, \frac{4}{3}\right) + \left(\frac{4}{3}, \frac{2}{3}\right) = \left(\frac{8}{3}, 2\right),$$

which coincides with the claimed computation rate region of  $\{(R, \tilde{R}) : R \leq C_{\text{pf}} = \frac{8}{3}, \tilde{R} \leq \tilde{C}_{\text{pf}} = 2\}$ .  $\square$

We find that this idea can be extended to arbitrary values of  $(m, n)$ ,  $(\tilde{m}, \tilde{n})$ . The channel regimes of this category are the

remaining regimes: (R4) and (R4'). See Appendix C for the detailed proof.

## V. PROOF OF CONVERSE

Since the bounds of (5) and (6) are straightforward cut-set bounds, we omit them. Note that the bounds of (3) and (4) are the perfect-feedback bounds in [28]. For completeness, we will provide the proof for (3). Note that the bound of (4) follows by symmetry.

*Proof of (3) (Perfect-feedback Bound):* The proof for the case of  $\alpha = 1$  is straightforward owing to the standard cut-set argument:  $N(R - \epsilon_N) \leq I(S_1^K \oplus S_2^K; Y_1^N, \tilde{S}_1^K) \stackrel{(a)}{=} I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^K) \leq \sum_{i=1}^N H(Y_{1i}) \leq N \max(m, n)$ . Here (a) follows from the independence of  $S_1^K \oplus S_2^K$  and  $\tilde{S}_1^K$ . If  $R$  is achievable, then  $\epsilon_N \rightarrow 0$  as  $N$  tends to infinity, and hence  $R \leq \max(m, n) = n$ .

Now consider the case where  $\alpha \neq 1$ . Starting with Fano's inequality, we get:

$$\begin{aligned} & N(3R - \epsilon_N) \\ & \leq \left[ I(S_1^K \oplus S_2^K; Y_1^N, \tilde{S}_1^K) + I(S_1^K \oplus S_2^K; Y_2^N, \tilde{S}_2^K) \right. \\ & \quad \left. + I(S_1^K \oplus S_2^K; Y_1^N, \tilde{S}_1^K) \right] \\ & = \left[ I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^K) + I(S_1^K \oplus S_2^K; Y_2^N | \tilde{S}_2^K) \right. \\ & \quad \left. + I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^K) \right] \\ & \leq \left[ H(Y_1^N | \tilde{S}_1^K) - H(Y_1^N | S_1^K \oplus S_2^K, \tilde{S}_1^K) + H(Y_2^N | \tilde{S}_2^K) \right. \\ & \quad \left. - H(Y_2^N | S_1^K \oplus S_2^K, \tilde{S}_2^K) + I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^K) \right] \\ & \stackrel{(b)}{\leq} \left[ H(Y_1^N) - H(Y_1^N | S_1^K \oplus S_2^K, \tilde{S}_1^K, \tilde{S}_2^K) + H(Y_2^N) \right. \\ & \quad \left. - H(Y_2^N | S_1^K \oplus S_2^K, \tilde{S}_1^K, \tilde{S}_2^K, Y_1^N) \right. \\ & \quad \left. + I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^K) \right] \end{aligned}$$

$$\begin{aligned}
&= \left[ H(Y_1^N) + H(Y_2^N) - H(Y_1^N, Y_2^N | S_1^K \oplus S_2^K, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right. \\
&\quad \left. + I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^{\tilde{K}}) \right] \\
&\stackrel{(c)}{\leq} \left[ H(Y_1^N) + H(Y_2^N) - H(Y_1^N, Y_2^N | S_1^K \oplus S_2^K, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right. \\
&\quad \left. + I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right] \\
&\stackrel{(d)}{\leq} \left[ H(Y_1^N) + H(Y_2^N) - H(Y_1^N, Y_2^N | S_1^K \oplus S_2^K, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right. \\
&\quad \left. + I(S_1^K; Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right] \\
&\stackrel{(e)}{\leq} \left[ H(Y_1^N) + H(Y_2^N) - H(Y_1^N, Y_2^N | S_1^K \oplus S_2^K, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right. \\
&\quad \left. + I(S_1^K; Y_1^N, Y_2^N | S_1^K \oplus S_2^K, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right] \\
&= H(Y_1^N) + H(Y_2^N) \leq \sum_{i=1}^N [H(Y_{1i}) + H(Y_{2i})] \\
&\leq 2N \max(m, n)
\end{aligned}$$

where (b) follows from the fact that conditioning reduces entropy; (c) follows from the non-negativity of mutual information and the fact that  $S_1^K \oplus S_2^K$  and  $\tilde{S}_2^{\tilde{K}}$  are independent conditioned on  $\tilde{S}_1^{\tilde{K}}$ ; (d) follows from Lemma 1 below; and (e) follows from the non-negativity of mutual information and the fact that  $S_1^K$  and  $S_1^K \oplus S_2^K$  are independent conditioned on  $(\tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}})$ . If  $R$  is achievable, then  $\epsilon_N \rightarrow 0$  as  $N$  tends to infinity, and hence  $R \leq \frac{2}{3} \max(m, n)$ . We therefore acquire the desired bound.

**Lemma 1.**

$$I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \leq I(S_1^K; Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}).$$

*Proof:*

$$\begin{aligned}
&I(S_1^K \oplus S_2^K; Y_1^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \\
&\stackrel{(a)}{=} H(S_1^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) - H(S_1^K \oplus S_2^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, Y_1^N) \\
&\leq H(S_1^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) - H(S_1^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, Y_1^N, Y_2^N, S_2^K) \\
&\stackrel{(b)}{=} H(S_1^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) - H(S_1^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, Y_1^N, Y_2^N) \\
&= I(S_1^K; Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}})
\end{aligned}$$

where (a) follows from the fact that  $H(S_1^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) = H(S_1^K) = H(S_1^K \oplus S_2^K) = H(S_1^K \oplus S_2^K | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}})$ ; and (b) follows from  $S_1^K - (Y_1^N, Y_2^N, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) - S_2^K$  (see Lemma 2 below). ■

**Lemma 2.**  $S_1^K - (Y_1^N, Y_2^N, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) - S_2^K$ .

*Proof:*

$$\begin{aligned}
&I(S_1^K; S_2^K | Y_1^N, Y_2^N, \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \\
&= \left[ I(S_1^K; S_2^K, Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right. \\
&\quad \left. - I(S_1^K; Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right]
\end{aligned}$$

$$\begin{aligned}
&= \left[ I(S_1^K; Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_2^K) \right. \\
&\quad \left. - I(S_1^K; Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) \right] \\
&= \left[ -H(Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) + H(Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_1^K) \right. \\
&\quad \left. + H(Y_1^N, Y_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_2^K) \right] \\
&\stackrel{(a)}{=} \left[ -H(X_1^N, X_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}) + H(X_1^N, X_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_1^K) \right. \\
&\quad \left. + H(X_1^N, X_2^N | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_2^K) \right] \\
&\stackrel{(b)}{=} \left[ -\sum_{i=1}^N H(X_{1i}, X_{2i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, X_1^{i-1}, X_2^{i-1}) \right. \\
&\quad + \sum_{i=1}^N H(X_{1i}, X_{2i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_1^K, X_1^{i-1}, X_2^{i-1}, \tilde{Y}_1^{i-1}, \tilde{Y}_2^{i-1}) \\
&\quad \left. + \sum_{i=1}^N H(X_{1i}, X_{2i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_2^K, X_1^{i-1}, X_2^{i-1}, \tilde{Y}_1^{i-1}, \tilde{Y}_2^{i-1}) \right] \\
&\stackrel{(c)}{=} \left[ -\sum_{i=1}^N H(X_{1i}, X_{2i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, X_1^{i-1}, X_2^{i-1}) \right. \\
&\quad + \sum_{i=1}^N H(X_{2i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_1^K, X_1^{i-1}, X_2^{i-1}) \\
&\quad \left. + \sum_{i=1}^N H(X_{1i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_2^K, X_1^{i-1}, X_2^{i-1}) \right] \\
&\stackrel{(d)}{=} \left[ -\sum_{i=1}^N \left[ H(X_{1i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, X_1^{i-1}, X_2^{i-1}) \right. \right. \\
&\quad \left. - H(X_{1i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_2^K, X_1^{i-1}, X_2^{i-1}) \right] \\
&\quad - \sum_{i=1}^N \left[ H(X_{2i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, X_1^i, X_2^{i-1}) \right. \\
&\quad \left. - H(X_{2i} | \tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, S_1^K, X_1^{i-1}, X_2^{i-1}, X_{1i}) \right] \leq 0
\end{aligned}$$

where (a) follows from the fact that  $(X_1, X_2)$  is a function of  $(Y_1, Y_2)$  (see Claim 1 below); (b) follows from the fact that  $(Y_1^{i-1}, Y_2^{i-1})$  is a function of  $(X_1^{i-1}, X_2^{i-1})$ ,  $(\tilde{X}_1^{i-1}, \tilde{X}_2^{i-1})$  is a function of  $(\tilde{S}_1^{\tilde{K}}, \tilde{S}_2^{\tilde{K}}, Y_1^{i-1}, Y_2^{i-1})$ , and  $(\tilde{Y}_1^{i-1}, \tilde{Y}_2^{i-1})$  is a function of  $(\tilde{X}_1^{i-1}, \tilde{X}_2^{i-1})$ ; (c) and (d) follow from the fact that  $X_{ki}$  is a function of  $(S_k^K, \tilde{Y}_k^{i-1})$ ,  $k = 1, 2$ . This completes the converse proof. ■

**Claim 1.** For  $\alpha \neq 1$  (i.e.,  $m \neq n$ ),  $(X_1, X_2)$  is a function of  $(Y_1, Y_2)$ .

*Proof:* It suffices to consider the case of  $m < n$ , as the other case follows by symmetry. From (1), we get:  $Y_1 \oplus (Y_2 \mathbf{G}^{n-m}) = X_1 (\mathbf{I}_n \oplus \mathbf{G}^{2(n-m)})$ . Note that  $\mathbf{I}_n \oplus \mathbf{G}^{2(n-m)}$  is invertible when  $m \neq n$ . Hence,  $X_1$  is a function of  $(Y_1, Y_2)$ . By symmetry,  $X_2$  is a function of  $(Y_1, Y_2)$ . ■

## VI. DISCUSSION

### A. Role of Interaction in General Networks

Although there have been results which show interaction gain in many traditional communication settings [24], [32]–[34], an explicit comparison between non-interactive vs. interactive scenarios in the context of computation settings was not made yet. Note that as suggested in Remark 3, the nature of the interaction gain comes from exploiting the past received signals, partially decoded symbols and nodes' own information as side information. This nature is not limited to the ADT set-up. Hence, we believe that these insights can carry over a variety of other configurations. With the idea of our achievability, it would be interesting to explore the role of interaction for a variety of channels in computation scenarios such as multi-hop networks or fully-connected multicast channels in which the two nodes in each side can cooperate with each other.

### B. Translation to the Gaussian Channel

Although the capacity of point-to-point memoryless Gaussian two-way networks are established [4], we are still lacking in our understanding of how to characterize capacities of multi-node Gaussian two-way networks. Nevertheless, the deterministic-channel achievability proposed in this work gives insights into an achievable scheme in the noisy case. This is inspired by several observations that can be made in our achievability. First, extracting the desired functions or feedback signals from the whole received signals motivates us to consider quantize-and-binning [29], [35] in the Gaussian channel. Furthermore, interference neutralization via XORing motivates us to consider structured coding [36] and superposition coding with dirty paper coding [37] for nulling. We also believe that Han-Kobayashi message splitting [38] can play a significant role in implementing retrospective decoding for the Gaussian channel since the decoding strategy includes a sophisticated decoding order as well as sets proper symbols to decode for each time slot. One future work of interest is extending our capacity results to the Gaussian channel by employing such techniques carefully.

### C. Extension to Non-orthogonal Two-Way Networks

Our model assumes orthogonal forward and backward channels. In section IV-A, we mentioned that a careful adoption of superposition coding, interference neutralization, and retrospective decoding is the key to achieve the optimal performance.

For non-orthogonal two-way networks, one needs to further mitigate the interference between the two opposite transmissions. Since the interference management in addition to the above key techniques requires an additional non-straightforward step to employ the techniques, we consider the complete extension as a separate piece of work, leaving it as a future work. However, in general, one can expect to achieve a smaller computation capacity region compared to the orthogonal case (i.e.,  $C_{\text{non}} \subseteq C$ ).

## VII. CONCLUSION

We investigated the role of interaction for computation problem settings. Our main contribution lies in the complete characterization of the two-way computation capacity region for the four-node ADT deterministic network. As a consequence of this result, we showed that interaction not only offers a net increase in computation capacity, but also it leads us to get all the way to perfect-feedback computation capacities simultaneously in both directions.

### APPENDIX A

ACHIEVABILITY FOR  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$ ,  
AND ARBITRARY  $L$

The achievability consists of two parts:

1) **Stage 1:** For time  $\ell = 1, \dots, 2L$ , the transmission signals at nodes 1 and 2 are as follows:

node 1: (18)

$$\begin{bmatrix} a_{2\ell-1} \\ a_{2\ell} \end{bmatrix} \oplus \begin{bmatrix} \tilde{F}_{2(\ell-2)} \oplus a_{2(\ell-2)} \\ \tilde{b}_{2(\ell-2)-1} \oplus F_{2(\ell-2)-1} \oplus a_{2(\ell-2)-1} \oplus \tilde{a}_{2(\ell-4)} \end{bmatrix},$$

node 2: (19)

$$\begin{bmatrix} b_{2\ell} \\ b_{2\ell-1} \end{bmatrix} \oplus \begin{bmatrix} \tilde{F}_{2(\ell-2)-1} \oplus b_{2(\ell-2)-1} \\ \tilde{a}_{2(\ell-2)} \oplus F_{2(\ell-2)} \oplus b_{2(\ell-2)} \oplus \tilde{b}_{2(\ell-4)-1} \end{bmatrix}.$$

Similarly, for time  $\ell = 1, \dots, 2L - 2$ , nodes  $\tilde{1}$  and  $\tilde{2}$  deliver:

node  $\tilde{1}$ : (20)

$$\begin{bmatrix} \tilde{a}_{2\ell} \\ \tilde{a}_{2\ell-1} \end{bmatrix} \oplus \begin{bmatrix} F_{2\ell} \oplus \tilde{a}_{2(\ell-2)-1} \\ a_{2\ell-1} \oplus \tilde{F}_{2(\ell-2)} \oplus a_{2(\ell-2)} \oplus \tilde{a}_{2(\ell-2)} \oplus F_{2(\ell-2)} \end{bmatrix},$$

node  $\tilde{2}$ : (21)

$$\begin{bmatrix} \tilde{b}_{2\ell-1} \\ \tilde{b}_{2\ell} \end{bmatrix} \oplus \begin{bmatrix} F_{2\ell-1} \oplus \tilde{b}_{2(\ell-2)} \\ b_{2\ell} \oplus \tilde{F}_{2(\ell-2)-1} \oplus b_{2(\ell-2)-1} \oplus \tilde{b}_{2(\ell-2)-1} \oplus F_{2(\ell-2)-1} \end{bmatrix}.$$

There are a few points to note. First, the transmitted signal of each node includes two parts: Fresh symbols, e.g.,  $(a_{2\ell-1}, a_{2\ell})$  at node 1, and feedback signals, e.g.,  $(\tilde{F}_{2(\ell-2)} \oplus a_{2(\ell-2)}, \tilde{b}_{2(\ell-2)-1} \oplus F_{2(\ell-2)-1} \oplus a_{2(\ell-2)-1} \oplus \tilde{a}_{2(\ell-4)})$ . Moreover, the feedback signals sent through the bottom levels ensure modulo-2 sum function computations at the bottom levels as these null out interference. Finally, we assume that if the index of a symbol is non-positive, we set the symbol as *null*.

For the last two time slots, nodes  $\tilde{1}$  and  $\tilde{2}$  do not send any fresh backward symbols. Instead, they mimic the perfect-feedback scheme; at time  $\ell$  ( $\ell = 2L - 1, 2L$ ), node  $\tilde{1}$  feeds back  $F_{2\ell}$  on the top level, while node  $\tilde{2}$  feeds back  $F_{2\ell-1}$  on the top level.

One can readily check that nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain  $\{F_{2\ell}\}_{\ell=1}^{2L}$  and  $\{F_{2\ell-1}\}_{\ell=1}^{2L}$  respectively. Similarly, nodes 1 and 2 can correspondingly obtain  $\{\tilde{F}_{2\ell}\}_{\ell=1}^{2(L-1)}$  and  $\{\tilde{F}_{2\ell-1}\}_{\ell=1}^{2(L-1)}$ .

2) **Stage 2:** During the next  $L$  time slots at the second stage, we accomplish the computation of the desired functions not yet obtained by each node. The successive refinement is done in a retrospective manner, allowing us to resolve the aforementioned issue. Specifically

the decoding order reads:  $(F_{4L-3}, F_{4L-2}, F_{4L-1}, F_{4L}) \rightarrow (\tilde{F}_{4(L-1)-3}, \tilde{F}_{4(L-1)-2}, \tilde{F}_{4(L-1)-1}, \tilde{F}_{4(L-1)}) \rightarrow \cdots \rightarrow (F_5, F_6, F_7, F_8) \rightarrow (\tilde{F}_1, \tilde{F}_2, \tilde{F}_3, \tilde{F}_4) \rightarrow (F_1, F_2, F_3, F_4)$ . With the refinement at time  $2L + \ell$  ( $\ell = 1, \dots, L$ ) (i.e., the  $\ell$ th time of Stage 2), each node can decode the following:

$$\text{node } \tilde{1}: (F_{4(L-(\ell-1))-3}, F_{4(L-(\ell-1))-1}), \quad (22)$$

$$\text{node } \tilde{2}: (F_{4(L-(\ell-1))-2}, F_{4(L-(\ell-1))}), \quad (23)$$

$$\text{node } 1: (\tilde{F}_{4(L-\ell)-3}, \tilde{F}_{4(L-\ell)-1} \oplus \tilde{F}_{4(L-(\ell+1))-3}), \quad (24)$$

$$\text{node } 2: (\tilde{F}_{4(L-\ell)-2}, \tilde{F}_{4(L-\ell)} \oplus \tilde{F}_{4(L-(\ell+1))-2}). \quad (25)$$

Note that after one more refinement at time  $2L + \ell + 1$ ,  $\tilde{F}_{4(L-(\ell+1))-3}$  and  $\tilde{F}_{4(L-(\ell+1))-2}$  from  $\tilde{F}_{4(L-\ell)-1} \oplus \tilde{F}_{4(L-(\ell+1))-3}$  and  $\tilde{F}_{4(L-\ell)} \oplus \tilde{F}_{4(L-(\ell+1))-2}$  can be canceled out at nodes 1 and 2, and therefore finally decode  $\tilde{F}_{4(L-\ell)-1}$  and  $\tilde{F}_{4(L-\ell)}$  respectively.

We start Stage 2 by taking the perfect-feedback strategy so that nodes  $\tilde{1}$  and  $\tilde{2}$  can decode  $(F_{4L-3}, F_{4L-1})$  and  $(F_{4L-2}, F_{4L})$  respectively. At time  $2L + \ell$  ( $\ell = 2, \dots, L$ ), a successive refinement is done to achieve reliable function computations both at the top and bottom levels. Exploiting newly decoded functions at time  $2L + \ell - 1$ , the signals transmitted at nodes 1 and 2 at time  $2L + \ell$  ( $\ell = 2, \dots, L$ ) are:

$$\begin{aligned} \text{node } 1: & \begin{bmatrix} \tilde{F}_{4(L-(\ell-1))-3} \\ \tilde{F}_{4(L-(\ell-1))-1} \oplus \tilde{F}_{4(L-\ell)-3} \end{bmatrix} \\ & \oplus \begin{bmatrix} \tilde{b}_{4(L-(\ell-1))-3} \oplus F_{4(L-(\ell-1))-3} \oplus \tilde{b}_{4(L-\ell)-2} \\ \tilde{b}_{4(L-(\ell-1))-1} \oplus F_{4(L-(\ell-1))-1} \oplus \tilde{b}_{4(L-\ell)} \end{bmatrix} \\ & \oplus \begin{bmatrix} \tilde{F}_{4(L-\ell)-2} \\ \tilde{F}_{4(L-\ell)} \oplus \tilde{F}_{4(L-(\ell-1))-2} \end{bmatrix}, \quad (26) \end{aligned}$$

$$\begin{aligned} \text{node } 2: & \begin{bmatrix} \tilde{F}_{4(L-(\ell-1))-2} \\ \tilde{F}_{4(L-(\ell-1))} \oplus \tilde{F}_{4(L-\ell)-2} \end{bmatrix} \\ & \oplus \begin{bmatrix} \tilde{a}_{4(L-(\ell-1))-2} \oplus F_{4(L-(\ell-1))-2} \oplus \tilde{a}_{4(L-\ell)-3} \\ \tilde{a}_{4(L-(\ell-1))} \oplus F_{4(L-(\ell-1))} \oplus \tilde{a}_{4(L-\ell)-1} \end{bmatrix} \\ & \oplus \begin{bmatrix} \tilde{F}_{4(L-\ell)-3} \\ \tilde{F}_{4(L-\ell)-1} \oplus \tilde{F}_{4(L-(\ell-1))-3} \end{bmatrix}. \quad (27) \end{aligned}$$

Notice that the signals in the first bracket are newly decoded functions; the signals in the second bracket are those received at time  $2(L - (\ell - 1)) - 1$ ,  $2(L - (\ell - 1))$  on the top level; and those in the third bracket are modulo-2 sum functions decoded at Stage 1 (e.g., even-index functions for node 1). This transmission allows nodes  $\tilde{1}$  and  $\tilde{2}$  to decode  $(F_{4(L-(\ell-1))-3}, F_{4(L-(\ell-1))-1})$  and  $(F_{4(L-(\ell-1))-2}, F_{4(L-(\ell-1))})$  using their own symbols and previously decoded functions.

Similarly, for time  $2L + \ell$  ( $\ell = 1, \dots, L$ ), nodes  $\tilde{1}$  and  $\tilde{2}$  deliver (28) and (29) respectively, shown at the bottom of the page. Note that the signals in the third bracket are modulo-2 sum functions decoded at Stage 1 and the summation of those and the received signals on the top level. In particular,  $a_{4(L-\ell)-3} \oplus \tilde{F}_{4(L-(\ell+1))-2} \oplus a_{4(L-(\ell+1))-2}$  and  $b_{4(L-\ell)-2} \oplus \tilde{F}_{4(L-(\ell+1))-3} \oplus b_{4(L-(\ell+1))-3}$  (in the third bracket of (28) and (29)) are the received signals at time  $2(L - \ell) - 1$ . As a result, nodes 1 and 2 can compute  $(\tilde{F}_{4(L-\ell)-3}, \tilde{F}_{4(L-\ell)-1} \oplus \tilde{F}_{4(L-(\ell+1))-3})$  and  $(\tilde{F}_{4(L-\ell)-2}, \tilde{F}_{4(L-\ell)} \oplus \tilde{F}_{4(L-(\ell+1))-2})$  using their own symbols and past decoded functions. Consequently, for  $\ell = 1, \dots, L$ , nodes  $\tilde{1}$  and  $\tilde{2}$  can compute the previously missing functions  $\{F_{2\ell-1}\}_{\ell=1}^{2L}$  and  $\{F_{2\ell}\}_{\ell=1}^{2L}$ , while nodes 1 and 2 can compute  $\{\tilde{F}_{2\ell-1}\}_{\ell=1}^{2(L-1)}$  and  $\{\tilde{F}_{2\ell}\}_{\ell=1}^{2(L-1)}$  respectively. Through the entire stages, our scheme can therefore achieve  $(R, \tilde{R}) = (\frac{4L}{3L}, \frac{4L-4}{3L}) = (\frac{4}{3}, \frac{4L-4}{3L})$  is achievable. Note that as  $L \rightarrow \infty$ , we get the desired computation rate pair:  $(R, \tilde{R}) \rightarrow (\frac{4}{3}, \frac{4}{3}) = (C_{\text{pf}}, \tilde{C}_{\text{pf}})$ .

## APPENDIX B

ACHIEVABILITY FOR  $(m, n) = (1, 2)$ ,  $(\tilde{m}, \tilde{n}) = (1, 0)$ ,  
AND ARBITRARY  $(L, M)$

The achievability consists of four parts:

1) *Time  $(3L+1)(i-1)+2\ell$  at Stage  $2i-1$* : For time  $\ell = 1, \dots, L$ , the transmission strategy at nodes 1 and 2 is to send fresh forward symbols along with the past received signals. Note that the signals in the first bracket below refer to fresh forward symbols; and the signals in the remaining

$$\begin{aligned} \text{node } \tilde{1}: & \begin{bmatrix} F_{4(L-(\ell-1))-3} \\ F_{4(L-(\ell-1))-1} \end{bmatrix} \oplus \begin{bmatrix} a_{4(L-(\ell-1))-3} \oplus \tilde{F}_{4(L-\ell)-2} \oplus a_{4(L-\ell)-2} \\ a_{4(L-(\ell-1))-1} \oplus \tilde{F}_{4(L-\ell)} \oplus a_{4(L-\ell)} \end{bmatrix} \\ & \oplus \begin{bmatrix} F_{4(L-\ell)-2} \\ F_{4(L-\ell)} \oplus F_{4(L-(\ell-1))-2} \oplus a_{4(L-\ell)-3} \oplus \tilde{F}_{4(L-(\ell+1))-2} \oplus a_{4(L-(\ell+1))-2} \oplus F_{4(L-(\ell+1))-2} \end{bmatrix}, \quad (28) \end{aligned}$$

$$\begin{aligned} \text{node } \tilde{2}: & \begin{bmatrix} F_{4(L-(\ell-1))-2} \\ F_{4(L-(\ell-1))} \end{bmatrix} \oplus \begin{bmatrix} b_{4(L-(\ell-1))-2} \oplus \tilde{F}_{4(L-\ell)-3} \oplus b_{4(L-\ell)-3} \\ b_{4(L-(\ell-1))} \oplus \tilde{F}_{4(L-\ell)-1} \oplus b_{4(L-\ell)-1} \end{bmatrix} \\ & \oplus \begin{bmatrix} F_{4(L-\ell)-3} \\ F_{4(L-\ell)-1} \oplus F_{4(L-(\ell-1))-3} \oplus b_{4(L-\ell)-2} \oplus \tilde{F}_{4(L-(\ell+1))-3} \oplus b_{4(L-(\ell+1))-3} \oplus F_{4(L-(\ell+1))-3} \end{bmatrix}. \quad (29) \end{aligned}$$

$$\begin{aligned} \text{node } 1: & \begin{bmatrix} a_{4((i-1)L+\ell)-1} \\ a_{4((i-1)L+\ell)} \end{bmatrix} \oplus \begin{bmatrix} b_{4(((i-1)L+\ell)-1)-3} \oplus b_{4(((i-1)L+\ell)-1)} \oplus a_{4(((i-1)L+\ell)-2)-2} \\ a_{4(((i-1)L+\ell)-1)-2} \oplus a_{4(((i-1)L+\ell)-2)} \oplus a_{4(((i-1)L+\ell)-3)-2} \end{bmatrix} \\ & \oplus \begin{bmatrix} \tilde{b}_{2(((i-1)L+\ell)-1)} \oplus \tilde{a}_{2(((i-1)L+\ell)-2)-1} \\ 0 \end{bmatrix}, \quad (30) \end{aligned}$$

$$\begin{aligned} \text{node } 2: & \begin{bmatrix} b_{4((i-1)L+\ell)} \\ b_{4((i-1)L+\ell)-1} \end{bmatrix} \oplus \begin{bmatrix} a_{4(((i-1)L+\ell)-1)-2} \oplus a_{4(((i-1)L+\ell)-1)-1} \oplus b_{4(((i-1)L+\ell)-2)-3} \\ b_{4(((i-1)L+\ell)-1)-3} \oplus b_{4(((i-1)L+\ell)-2)-1} \oplus b_{4(((i-1)L+\ell)-3)-3} \end{bmatrix} \\ & \oplus \begin{bmatrix} \tilde{a}_{2(((i-1)L+\ell)-1)-1} \oplus \tilde{b}_{2(((i-1)L+\ell)-2)} \\ 0 \end{bmatrix}. \quad (31) \end{aligned}$$

$$\text{node } \tilde{1}: \left[ \tilde{a}_{2((i-1)L+\ell)-1} \right] \oplus \left[ F_{4((i-1)L+\ell)-2} \right] \oplus \left[ a_{4((i-1)L+\ell)-1} \oplus b_{4(((i-1)L+\ell)-1)-3} \oplus b_{4(((i-1)L+\ell)-1)} \right] \\ \oplus \left[ b_{4(((i-1)L+\ell)-2)-2} \oplus \tilde{b}_{2(((i-1)L+\ell)-1)} \right], \quad (32)$$

$$\text{node } \tilde{2}: \left[ \tilde{b}_{2((i-1)L+\ell)} \right] \oplus \left[ F_{4((i-1)L+\ell)-3} \right] \oplus \left[ b_{4((i-1)L+\ell)} \oplus a_{4(((i-1)L+\ell)-1)-2} \oplus a_{4(((i-1)L+\ell)-1)-1} \right] \\ \oplus \left[ a_{4(((i-1)L+\ell)-2)-3} \oplus \tilde{a}_{2(((i-1)L+\ell)-1)-1} \right]. \quad (33)$$

$$\text{node } 1: \left[ \begin{array}{c} a_{4((i-1)L+\ell)-3} \\ a_{4((i-1)L+\ell)-2} \end{array} \right] \oplus \left[ \begin{array}{c} a_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell))} \oplus a_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+1))-2} \\ a_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+1))} \oplus a_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+2))-2} \end{array} \right] \\ \oplus \left[ \begin{array}{c} \tilde{F}_{2(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+1))-1} \\ 0 \end{array} \right], \quad (34)$$

$$\text{node } 2: \left[ \begin{array}{c} b_{4((i-1)L+\ell)-2} \\ b_{4((i-1)L+\ell)-3} \end{array} \right] \oplus \left[ \begin{array}{c} b_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell))-1} \oplus b_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+1))-3} \\ b_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+1))-1} \oplus b_{4(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+2))-3} \end{array} \right] \\ \oplus \left[ \begin{array}{c} \tilde{F}_{2(((i-1)-(2^{L-\ell+1}-2))L-(L-\ell+1))} \\ 0 \end{array} \right]. \quad (35)$$

brackets refer to those received previously from (33) and (32) in the current layer. We note that the idea of interference neutralization is also employed by adapting each node's transmitted signal to own symbols. This ensures modulo-2 sum function computations on the bottom level of nodes  $\tilde{1}$  and  $\tilde{2}$  for each time. Here we assume that if the index of a symbol is non-positive, we set the symbol as *null*. Specifically nodes 1 and 2 send (30) and (31) respectively, shown at the bottom of the previous page. With fresh backward symbols, past computed functions, and the received signals from the above, nodes  $\tilde{1}$  and  $\tilde{2}$  deliver (32) and (33) respectively, shown at the top of this page.

2) *Time*  $(3L+1)(i-1)+2\ell-1$  at *Stage*  $2i-1$ : For time  $\ell = 1, \dots, L$ , the transmission strategy at nodes 1 and 2 is as follows. The idea is similar to that in part 1), but here the formulae in the second bracket below refer to the signals received from (57) and (56) at part 4) of Layer  $i-1$ . Again, modulo-2 sum function computations on the bottom level of nodes  $\tilde{1}$  and  $\tilde{2}$  are possible for each time. Specifically, nodes 1 and 2 transmit (34) and (35) respectively, shown at the top of this page. In addition, using the newly decoded  $F_{4(((i-1)-(2^{\ell-2}))L-(\ell-1))-3}$  (60) and  $F_{4(((i-1)-(2^{\ell-2}))L-(\ell-1))-2}$  (61) at part 4) of Layer  $i-1$  and some of the previously received signals, nodes  $\tilde{1}$  and  $\tilde{2}$  transmit:

$$\text{node } \tilde{1}: \quad (36)$$

$$\left[ b_{4(((i-1)-(2^{\ell-2}))L-(\ell-1))-3} \right] \\ \oplus \left[ b_{4((i-(2^{\ell+1}-2))L-(\ell-1))} \oplus b_{4((i-(2^{\ell+1}-2))L-\ell)-2} \right] \\ \oplus \left[ \tilde{F}_{2((i-(2^{\ell+1}-2))L-\ell)-1} \right],$$

$$\text{node } \tilde{2}: \quad (37)$$

$$\left[ a_{4(((i-1)-(2^{\ell-2}))L-(\ell-1))-2} \right] \\ \oplus \left[ a_{4((i-(2^{\ell+1}-2))L-(\ell-1))-1} \oplus a_{4((i-(2^{\ell+1}-2))L-\ell)-3} \right] \\ \oplus \left[ \tilde{F}_{2((i-(2^{\ell+1}-2))L-\ell)} \right].$$

Here, one can see that unless the indices of signals (36) and (37) are positive, the newly decoded functions enable nodes 1 and 2 to obtain additional  $\tilde{F}_{2((i-(2^{\ell+1}-2))L-\ell)}$  and  $\tilde{F}_{2((i-(2^{\ell+1}-2))L-\ell)-1}$  using their own symbols.

Throughout part 1) and 2), the available function computations are as follows:

$$\text{node } 1: \{\tilde{F}_{2((i-(2^{\ell+1}-2))L-\ell)}\}_{\ell=1}^L, \quad (38)$$

$$\text{node } 2: \{\tilde{F}_{2((i-(2^{\ell+1}-2))L-\ell)-1}\}_{\ell=1}^L, \quad (39)$$

$$\text{node } \tilde{1}: \{(F_{4((i-1)L+\ell)-2}, F_{4((i-1)L+\ell)})\}_{\ell=1}^L, \quad (40)$$

$$\text{node } \tilde{2}: \{(F_{4((i-1)L+\ell)-3}, F_{4((i-1)L+\ell)-1})\}_{\ell=1}^L. \quad (41)$$

3-1) *Time*  $(3L+1)(i-1)+2L+1$  at *Stage*  $2i$ : With the received signals at time  $(3L+1)(i-1)+2L$ , the transmission scheme is as follows.

$$\text{node } 1: \quad (42)$$

$$\left[ \begin{array}{c} F_{4iL} \oplus b_{4iL-3} \oplus a_{4(iL-1)-2} \oplus [\tilde{b}_{2iL} \oplus \tilde{a}_{2(iL-1)-1}] \\ a_{4iL-1} \oplus a_{4iL-2} \oplus a_{4(iL-1)} \oplus a_{4(iL-2)-2} \end{array} \right],$$

$$\text{node } 2: \quad (43)$$

$$\left[ \begin{array}{c} F_{4iL-1} \oplus a_{4iL-2} \oplus b_{4(iL-1)-3} \oplus [\tilde{a}_{2iL-1} \oplus \tilde{b}_{2(iL-1)}] \\ b_{4iL} \oplus b_{4iL-3} \oplus b_{4(iL-1)-1} \oplus b_{4(iL-2)-3} \end{array} \right],$$

$$\text{node } \tilde{1}: [b_{4iL-3} \oplus b_{4(iL-1)-2} \oplus \tilde{F}_{2iL}], \quad (44)$$

$$\text{node } \tilde{2}: [a_{4iL-2} \oplus b_{4(iL-1)-3} \oplus \tilde{F}_{2iL-1}]. \quad (45)$$

Together with the past received signals at time  $(3L+1)(i-1)+2L$ , nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain  $F_{4iL-1}$  and  $F_{4iL}$  from the above strategy.

3-2) *Time*  $(3L+1)(i-1)+2L+2$  at *Stage*  $2i$ : With the received signals at time  $(3L+1)(i-1)+2L+1$ , the transmission scheme is as follows.

$$\text{node } 1: \left[ \begin{array}{c} \tilde{F}_{2iL-1} \\ a_{4iL-3} \oplus a_{4(iL-1)-2} \end{array} \right], \quad (46)$$

$$\text{node } 2: \left[ \begin{array}{c} \tilde{F}_{2iL} \\ b_{4iL-2} \oplus b_{4(iL-1)-3} \end{array} \right], \quad (47)$$

$$\text{node } \tilde{1}: [\tilde{F}_{2iL-1}], \quad (48)$$

$$\text{node } \tilde{2}: [\tilde{F}_{2iL}]. \quad (49)$$

Exploiting the signals on the top level at time  $(3L+1)(i-1)+2L+1$ , nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain  $F_{4iL-3}$  and  $F_{4iL-2}$ . In turn, the available function computations from parts 3-1)

and 3-2) are as follows:

$$\text{node 1: } (\tilde{F}_{2iL-1}, \tilde{F}_{2iL}), \quad (50)$$

$$\text{node 2: } (\tilde{F}_{2iL-1}, \tilde{F}_{2iL}), \quad (51)$$

$$\text{node } \tilde{1}: (F_{4iL-3}, F_{4iL-1}), \quad (52)$$

$$\text{node } \tilde{2}: (F_{4iL-2}, F_{4iL}). \quad (53)$$

4) *Time*  $(3L+1)(i-1)+2L+\ell$  at *Stage 2i*: For time  $\ell = 3, \dots, L+1$ , the transmission strategy at nodes 1 and 2 is described as below. The idea is to exploit newly decoded  $\tilde{F}_{2((i-(2^{\ell-1}-2))L-(\ell-2))}$  (38) and  $\tilde{F}_{2((i-(2^{\ell-1}-2))L-(\ell-2))-1}$  (39) from part 2) of the current layer. In turn, nodes  $\tilde{1}$  and  $\tilde{2}$  can obtain two additional functions of interest for each time.

$$\text{node 1:} \quad (54)$$

$$\left[ \begin{array}{c} F_{4((i-(2^{\ell-1}-2))L-(\ell-2))-3} \oplus F_{4((i-(2^{\ell-1}-2))L-(\ell-2))} \\ 0 \end{array} \right] \oplus \left[ \begin{array}{c} \tilde{a}_{2((i-(2^{\ell-1}-2))L-(\ell-2))} \oplus \tilde{a}_{2((i-(2^{\ell-1}-2))L-(\ell-1))-1} \\ F_{2((i-(2^{\ell-1}-2))L-(\ell-2))-1} \end{array} \right],$$

$$\text{node 2:} \quad (55)$$

$$\left[ \begin{array}{c} F_{4((i-(2^{\ell-1}-2))L-(\ell-2))-2} \oplus F_{4((i-(2^{\ell-1}-2))L-(\ell-2))-1} \\ 0 \end{array} \right] \oplus \left[ \begin{array}{c} \tilde{b}_{2((i-(2^{\ell-1}-2))L-(\ell-2))-1} \oplus \tilde{b}_{2((i-(2^{\ell-1}-2))L-(\ell-1))} \\ F_{2((i-(2^{\ell-1}-2))L-(\ell-2))} \end{array} \right].$$

With the newly decoded  $F_{4((i-(2^{\ell-2}-2))L-(\ell-3))-1}$  and  $F_{4((i-(2^{\ell-2}-2))L-(\ell-3))}$ , nodes  $\tilde{1}$  and  $\tilde{2}$  deliver:

$$\text{node } \tilde{1}: \quad (56)$$

$$\left[ \begin{array}{c} b_{4((i-(2^{\ell-2}-2))L-(\ell-3))-1} \\ b_{4((i-(2^{\ell-2}-2))L-(\ell-2))-3} \oplus b_{4((i-(2^{\ell-2}-2))L-(\ell-2))} \\ b_{4((i-(2^{\ell-2}-2))L-(\ell-1))-2} \oplus \tilde{F}_{2((i-(2^{\ell-2}-2))L-(\ell-2))} \end{array} \right],$$

$$\text{node } \tilde{2}: \quad (57)$$

$$\left[ \begin{array}{c} a_{4((i-(2^{\ell-2}-2))L-(\ell-3))} \\ a_{4((i-(2^{\ell-2}-2))L-(\ell-2))-2} \oplus a_{4((i-(2^{\ell-2}-2))L-(\ell-2))-1} \\ a_{4((i-(2^{\ell-2}-2))L-(\ell-1))-3} \oplus \tilde{F}_{2((i-(2^{\ell-2}-2))L-(\ell-2))-1} \end{array} \right].$$

One can readily see that for each time, nodes 1 and 2 can obtain an additional interested function using their own symbols. Consequently, the available function computations in part 4) are as follows:

$$\text{node 1: } \{\tilde{F}_{2((i-(2^{\ell-2}-2))L-(\ell-2))-1}\}_{\ell=3}^{L+1}, \quad (58)$$

$$\text{node 2: } \{\tilde{F}_{2((i-(2^{\ell-2}-2))L-(\ell-2))}\}_{\ell=3}^{L+1}, \quad (59)$$

$$\text{node } \tilde{1}: \quad (60)$$

$$\{(F_{4((i-(2^{\ell-1}-2))L-(\ell-2))-3}, F_{4((i-(2^{\ell-1}-2))L-(\ell-2))-1})\}_{\ell=3}^{L+1},$$

$$\text{node } \tilde{2}: \quad (61)$$

$$\{(F_{4((i-(2^{\ell-1}-2))L-(\ell-2))-2}, F_{4((i-(2^{\ell-1}-2))L-(\ell-2))})\}_{\ell=3}^{L+1}.$$

Recall Remark 4 that unoccupied time slots (where each node keeps silent as the indices of signals from (54) to (57) above are less than or equal to zero) at the second stage of a layer cause inefficiency in the performance. However, one can see at certain moments, the second stage of a layer will eventually be fully packed. From (60) and (61), we can verify

this by putting  $\ell = L+1$  into the indices of (60) and (61), e.g.,  $4((i-(2^{\ell-1}-2))L-(\ell-2))-3$ , and check what condition of  $i$  provides the indices greater than zero. As long as  $i \geq 2^L - 1$ , each layer's second stage remains to be fully packed.

Essentially, we can calculate the total number of vacant time slots. First, we examine the condition for which the number of unoccupied time slots is less than or equal to 1. Similar to the above, putting  $\ell = L$  into the indices of (60) and (61) allows us to see that as long as  $i \geq 2^{L-1} - 1$ , the number of unoccupied time slots is less than or equal to 1. Hence the number of layers in which the vacant time slot of the layer is 1, is:  $(2^L - 1) - (2^{L-1} - 1) = 2^{L-1}$ . Applying a similar method, one can check that there are  $2^{L-\ell}$  layers whose unoccupied time slots are  $\ell$  ( $\ell = 2, \dots, L-1$ ). Note that the maximum number of unoccupied time slots at the second stage of each layer is  $L-1$ , as the first two time slots of the second stage are allocated for computing functions; see parts 3-1) and 3-2). Now using the formula of  $\sum_{\ell=1}^{L-1} (L-\ell)2^\ell = 2^{L+1} - 2L - 2$ , we see that the total number of unoccupied time slots is  $2^{L+1} - 2L - 2$ . At the end of Layer  $M$ , we observe that our scheme ensures  $4L(M - (2^{L+1} - 2L - 2))$  forward and  $2L(M - (2^{L+1} - 2L - 2))$  backward message computations during  $(3L+1)M$  time slots, and thus can achieve  $(R, \tilde{R}) = \left( \frac{4L(M - (2^{L+1} - 2L - 2))}{(3L+1)M}, \frac{2L(M - (2^{L+1} - 2L - 2))}{(3L+1)M} \right)$ . By setting  $M = (2 + \epsilon)^L$  where  $\epsilon > 0$ , and letting  $L \rightarrow \infty$ , the computation rate pair becomes  $(R, \tilde{R}) = (\frac{4}{3}, \frac{2}{3}) = (C_{\text{pt}}, \tilde{C}_{\text{pt}})$ . This completes the proof.

#### APPENDIX C PROOF OF GENERALIZATION TO ARBITRARY $(m, n)$ , $(\tilde{m}, \tilde{n})$

We now prove the achievability for arbitrary  $(m, n)$  and  $(\tilde{m}, \tilde{n})$ . The idea is to use the network decomposition in [30] (also illustrated in Fig. 9). This idea provides a conceptually simpler proof by decomposing a general  $(m, n)$ ,  $(\tilde{m}, \tilde{n})$  channel into multiple elementary subchannels and taking a proper matching across forward and backward subchannels. See Theorem 2 (stated below) for the identified elementary subchannels. As this approach is well established in our earlier work, while requiring tedious calculations, we will provide the proofs for some representative cases.

**Theorem 2** (Network Decomposition). *For an arbitrary  $(m, n)$  channel, the following network decomposition holds:*

$$(m, n) \longrightarrow (0, 1)^{n-2m} \times (1, 2)^m, \quad \alpha \in [0, 1/2]; \quad (62)$$

$$(m, n) \longrightarrow (1, 2)^{2n-3m} \times (2, 3)^{2m-n}, \quad \alpha \in [1/2, 2/3]; \quad (63)$$

$$(m, n) \longrightarrow (2, 1)^{2m-3n} \times (3, 2)^{2n-m}, \quad \alpha \in [3/2, 2]; \quad (64)$$

$$(m, n) \longrightarrow (1, 0)^{m-2n} \times (2, 1)^n, \quad \alpha \geq 2. \quad (65)$$

Here we use the symbol  $\times$  for the concatenation of orthogonal channels, with  $(i, j)^\ell$  denoting the  $\ell$ -fold concatenation of the  $(i, j)$  channel.

A. *Proof of (R1)*  $\alpha \leq \frac{2}{3}$ ,  $\tilde{\alpha} \leq \frac{2}{3}$

The claimed achievable computation rate region is  $\left\{ R \leq \frac{2}{3}n, \tilde{R} \leq \frac{2}{3}\tilde{n}, R + \tilde{R} \leq m + \tilde{m} \right\}$ . The below achievability

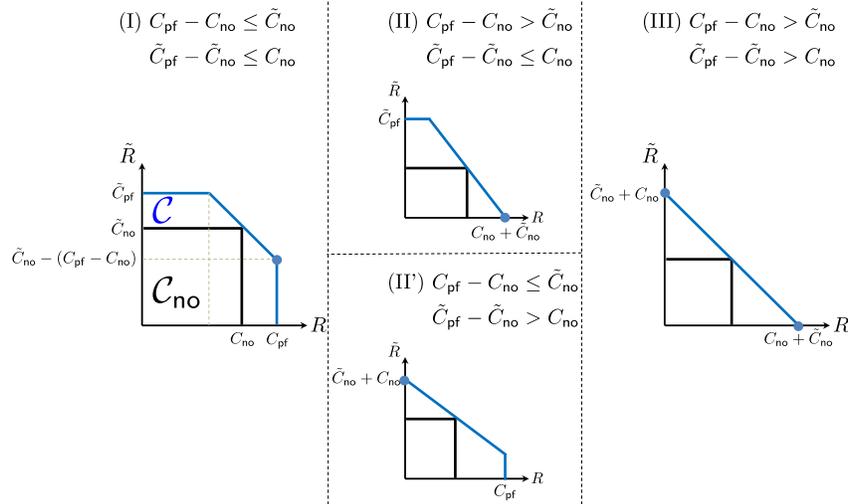


Fig. 10. Three types of shapes of an achievable computation rate region for the regime (R1)  $\alpha \leq \frac{2}{3}$ ,  $\tilde{\alpha} \leq \frac{2}{3}$ .

w.r.t. the elementary subchannels identified in Theorem 2 forms the basis of the proof.

**Lemma 3.** *The following computation rates are achievable:*

- (i) For the pair of  $(m, n) = (1, 2)^i$  and  $(\tilde{m}, \tilde{n}) = (1, 2)^j$ :  $(R, \tilde{R}) = (\frac{4}{3}i, j - \frac{1}{3}i)$ . Here  $\frac{1}{3}i \leq j$ .
- (ii) For the pair of  $(m, n) = (1, 2)^i$  and  $(\tilde{m}, \tilde{n}) = (2, 3)^j$ :  $(R, \tilde{R}) = (\frac{4}{3}i, 2j - \frac{1}{3}i)$ . Here  $\frac{1}{3}i \leq 2j$ .

*Proof:* The proof builds upon a simple combination of the non-feedback scheme [30] and the interactive scheme in our earlier work [2]. While it requires detailed calculations, it contains no new ingredients, hence, we do not provide a detailed proof here. ■

We see that there is no interaction gain in sum computation capacity. This means that one bit of a computation capacity increase due to feedback costs exactly one bit. Depending on whether or not  $C_{\text{pf}}$  (or  $\tilde{C}_{\text{pf}}$ ) exceeds  $C_{\text{no}} + \tilde{C}_{\text{no}}$ , we have four subcases, each of which forms a different shape of the region. See Fig. 10.

**(I)**  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{C}_{\text{no}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq C_{\text{no}}$ : The first case is one in which the amount of feedback for maximal improvement, reflected in  $C_{\text{pf}} - C_{\text{no}}$  (or  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}$ ), is smaller than the available resources offered by the backward channel (or forward channel respectively). In other words, in this case, we have a sufficient amount of resources such that one can achieve the perfect-feedback bound in one direction. By symmetry, it suffices to focus on one corner point that favors the computation rate of forward transmission:  $(R, \tilde{R}) = (C_{\text{pf}}, \tilde{C}_{\text{no}} - (C_{\text{pf}} - C_{\text{no}}))$ .

For efficient use of (62) and (63) in Theorem 2 and Lemma 3, we divide the regime (R1)  $\alpha \leq \frac{2}{3}$ ,  $\tilde{\alpha} \leq \frac{2}{3}$  into the following four sub-regimes: (R1-1)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \in [\frac{1}{2}, \frac{2}{3}]$ ; (R1-2)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \in [0, \frac{1}{2}]$ ; (R1-3)  $\alpha \in [0, \frac{1}{2}]$ ,  $\tilde{\alpha} \in [\frac{1}{2}, \frac{2}{3}]$ ; and (R1-4)  $\alpha \in [0, \frac{1}{2}]$ ,  $\tilde{\alpha} \in [0, \frac{1}{2}]$ .

(R1-1)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \in [\frac{1}{2}, \frac{2}{3}]$ : In this sub-regime, we note that either  $\frac{1}{3}(2n - 3m) = C_{\text{pf}} - C_{\text{no}} \leq 2\tilde{n} - 3\tilde{m}$  or  $C_{\text{pf}} - C_{\text{no}} \leq 2(2\tilde{m} - \tilde{n})$ ; otherwise, we encounter the contradiction of  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{C}_{\text{no}}$  ( $= \tilde{m}$ ).

Consider the case where  $\frac{1}{3}(2n - 3m) \leq 2\tilde{n} - 3\tilde{m}$ . In such a case, we apply Lemma 3 (i) for the pair of  $(1, 2)^{2n-3m}$  and  $(1, 2)^{2\tilde{n}-3\tilde{m}}$ . Note that the condition of (i) holds. Applying the non-feedback schemes for the remaining subchannels gives:

$$R = \frac{4}{3} \times (2n - 3m) + 2 \times (2m - n) = C_{\text{pf}},$$

$$\tilde{R} = \left( 1 \times (2\tilde{n} - 3\tilde{m}) - \frac{1}{3}(2n - 3m) \right) + 2 \times (2\tilde{m} - \tilde{n}) = \tilde{C}_{\text{no}} - (C_{\text{pf}} - C_{\text{no}}).$$

Now consider the case where  $C_{\text{pf}} - C_{\text{no}} = \frac{1}{3}(2n - 3m) \leq 2(2\tilde{m} - \tilde{n})$ . In this case, we apply Lemma 3 (ii) for the pair of  $(1, 2)^{2n-3m}$  and  $(2, 3)^{2\tilde{m}-\tilde{n}}$ . Note that the condition of (ii) holds. One can see that applying the non-feedback schemes for the remaining subchannels gives  $(R, \tilde{R}) = (C_{\text{pf}}, \tilde{C}_{\text{no}} - (C_{\text{pf}} - C_{\text{no}}))$ .

For the proofs of the remaining regimes (R1-2), (R1-3) and (R1-4), we omit details as the proofs follow similarly. As seen from all the cases above, one key observation is that the computation capacity increase due to feedback  $C_{\text{pf}} - C_{\text{no}}$  plus the backward computation rate is always  $\tilde{C}_{\text{no}}$ , meaning that there is *one-to-one tradeoff* between feedback and independent message computation.

**(II)**  $C_{\text{pf}} - C_{\text{no}} > \tilde{C}_{\text{no}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq C_{\text{no}}$ : Similar to the first case, one can readily prove that the same one-to-one tradeoff relationship exists when achieving one corner point  $(R, \tilde{R}) = (C_{\text{no}} - (\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}), \tilde{C}_{\text{pf}})$ . Hence, we omit the detailed proof. On the other hand, we note that there is a limitation in achieving the other counterpart. Note that the maximal feedback gain  $C_{\text{pf}} - C_{\text{no}}$  for forward computation does exceed the resource limit  $\tilde{C}_{\text{no}}$  offered by the backward channel. This limits the maximal achievable computation rate for forward computation to be saturated by  $R \leq C_{\text{no}} + \tilde{C}_{\text{no}}$ . Hence the other corner point reads  $(C_{\text{no}} + \tilde{C}_{\text{no}}, 0)$  instead. We will show this is indeed the case as below. Similar to the previous case, we provide the proof only for (R1-1). We omit the proofs for the regimes (R1-2), (R1-3) and (R1-4). Also, by symmetry, we omit the case of (II').

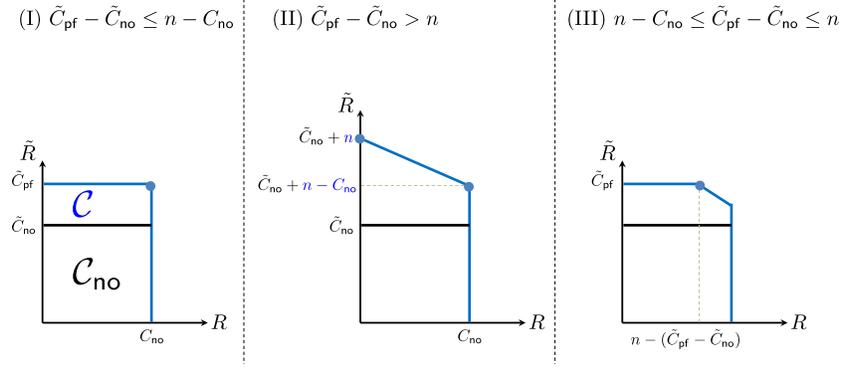


Fig. 11. Three types of shapes of an achievable computation rate region for the regime (R2) ( $\alpha \in [\frac{2}{3}, 1)$ ,  $\alpha \in (1, \frac{3}{2}]$ ),  $\tilde{\alpha} \geq \frac{3}{2}$ .

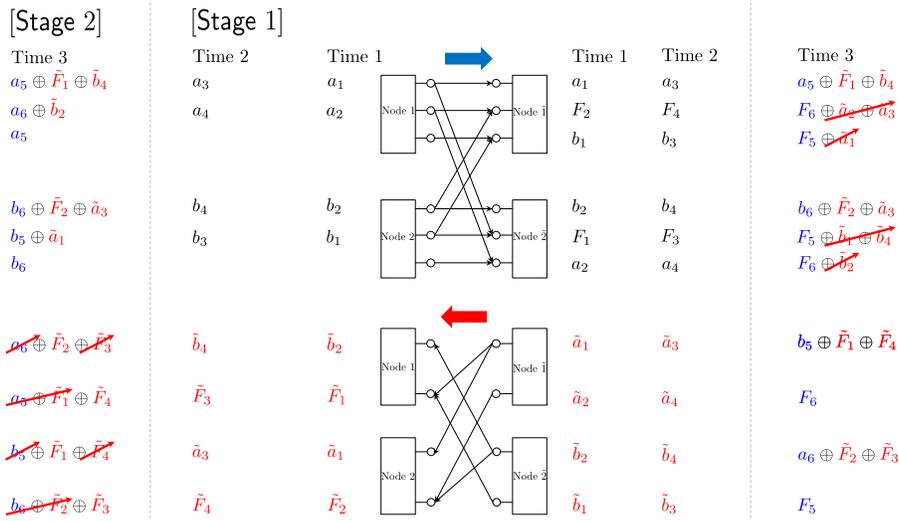


Fig. 12. Illustration of achievability for the regime (R2-1) via an example of  $(m, n) = (2, 3)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$ . This is an instance in which we have a sufficient amount of resources that enables achieving the perfect-feedback bound in the backward channel:  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} = \frac{1}{3} \leq 1 = n - C_{\text{no}}$ . Hence we achieve  $(R, \tilde{R}) = (C_{\text{no}}, \tilde{C}_{\text{pf}}) = (2, \frac{4}{3})$ .

(R1-1)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \in [\frac{1}{2}, \frac{2}{3}]$ : We apply Lemma 3 (i) for the pair of  $(1, 2)^{3(2\tilde{n}-3\tilde{m})}$  and  $(1, 2)^{2\tilde{n}-3\tilde{m}}$ . Also, we apply Lemma 3 (ii) for the pair of  $(1, 2)^{6(2\tilde{m}-\tilde{n})}$  and  $(2, 3)^{2\tilde{m}-\tilde{n}}$ . Applying the non-feedback schemes for the remaining subchannels  $(1, 2)^{(2n-3m)-3\tilde{m}}$  and  $(2, 3)^{2m-n}$  gives:  $(C_{\text{no}} + \tilde{C}_{\text{no}}, 0)$ .

(III)  $C_{\text{pf}} - C_{\text{no}} > \tilde{C}_{\text{no}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > C_{\text{no}}$ : This is the case in which there are limitations now in achieving both  $R = C_{\text{pf}}$  and  $\tilde{R} = \tilde{C}_{\text{pf}}$ . With the same argument as above, what we can maximally achieve for  $R$  (or  $\tilde{R}$ ) in exchange of the other channel is  $C_{\text{no}} + \tilde{C}_{\text{no}}$  which implies that  $(R, \tilde{R}) = (C_{\text{no}} + \tilde{C}_{\text{no}}, 0)$  or  $(0, C_{\text{no}} + \tilde{C}_{\text{no}})$  is achievable. The proof follows exactly the same as above, so we omit details.

*B. Proof of (R2) ( $\alpha \in [\frac{2}{3}, 1)$ ,  $\alpha \in (1, \frac{3}{2}]$ ),  $\tilde{\alpha} \geq \frac{3}{2}$ .*

For the regime of (R2), we note that  $C_{\text{pf}} = C_{\text{no}}$  and  $C_{\text{no}} + \tilde{C}_{\text{pf}} = \frac{2}{3} \max(m, n) + \frac{2}{3} \tilde{m} \leq m + \tilde{m}$ , so the claimed achievable computation rate region is:  $\left\{ R \leq \frac{2}{3}n, \tilde{R} \leq \frac{2}{3}\tilde{m}, R + \tilde{R} \leq n + \tilde{n} \right\}$ . Unlike the previous regime, there is an interaction gain for this regime. Note that

the sum computation rate bound exceeds  $C_{\text{no}} + \tilde{C}_{\text{no}}$ ; however, there is no feedback gain in the forward channel. The network decompositions (64) and (65) together with  $3(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}) = 2\tilde{m} - 3\tilde{n}$  give:

$$(\tilde{m}, \tilde{n}) \longrightarrow \begin{cases} (2, 1)^{3(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}})} \times (3, 2)^{2\tilde{n} - \tilde{m}}, & \tilde{\alpha} \in [3/2, 2]; \\ (1, 0)^{\tilde{m} - 2\tilde{n}} \times (2, 1)^{\tilde{n}}, & \tilde{\alpha} \geq 2. \end{cases}$$

We find that the shape of the region depends on where  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}$  lies in between  $n - C_{\text{no}}$  and  $n$ . See Fig. 11.

(I)  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{no}}$ : The first case is one in which the amount of feedback for maximal improvement, reflected in  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}$ , is small enough to achieve the maximal feedback gain without sacrificing the performance of the forward computation. Now let us show how to achieve  $(R, \tilde{R}) = (C_{\text{no}}, \tilde{C}_{\text{pf}})$ . To do this, we divide the backward channel regime into the two sub-regimes: (R2-1)  $\tilde{\alpha} \in [\frac{3}{2}, 2]$ ; and (R2-2)  $\tilde{\alpha} \geq 2$ . Here we provide details for (R2-1).

(R2-1)  $\tilde{\alpha} \in [\frac{3}{2}, 2]$ : For the first sub-regime, the idea is to pair up  $(m, n)$  and  $(2, 1)^{3(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}})}$ , while applying the non-feedback schemes for the remaining backward subchannels  $(3, 2)^{2\tilde{n} - \tilde{m}}$ . To give an achievability idea for the first pair,

we exploit a simple scheme described in Fig. 12 where  $(m, n) = (2, 3)$  and  $(\tilde{m}, \tilde{n}) = (2, 1)$ . As the idea of the scheme is the same as that of our aforementioned toy examples, we omit the detailed explanation. Eventually, one can see that nodes  $\tilde{1}$  and  $\tilde{2}$  obtain  $F_\ell$  ( $\ell = 1, \dots, 6$ ) during three time slots, thus achieving  $R = 2$  ( $= C_{\text{no}}$ ). Furthermore, nodes 1 and 2 obtain  $F_\ell$  ( $\ell = 1, \dots, 4$ ), thus achieving  $\tilde{R} = \frac{4}{3}$  ( $= \tilde{C}_{\text{pf}}$ ). Here one can make two observations. First, at time 3,  $C_{\text{no}} = m$  ( $= 2$ , which is the second and third) levels are utilized to perform forward-message computation in each time. Through the remaining first direct-link level, feedback transmissions are performed. Observe that feedback signals are interfered by fresh forward symbols through the  $m$  levels, but it turns out that the interference does not cause any problem in computing functions. In general, one can maximally utilize available resource levels: The total number of direct-link levels for forward channel is  $n$ , accordingly,  $n - C_{\text{no}}$  levels can be exploited for feedback. In the general case of  $(2, 1)^{3(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}})}$ , the maximal feedback gain is  $(\tilde{C}_{\text{pf}}^{(2,1)} - \tilde{C}_{\text{no}}^{(2,1)}) \times 3(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}) = \tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}$ , which does not exceed the limit on the exploitable levels  $n - C_{\text{no}}$  under the considered regime. Here  $\tilde{C}_{\text{no}}^{(2,1)}$  denotes the non-feedback computation capacity of  $(2, 1)$  model. Hence, we achieve:

$$\tilde{R}^{(1)} = \tilde{C}_{\text{pf}}^{(2,1)} \times 3(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}) = 4(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}). \quad (66)$$

Now the second observation is that the feedback transmission does not cause any interference to nodes  $\tilde{1}$  and  $\tilde{2}$ . This ensures that  $R^{(1)} = C_{\text{no}}$ . On the other hand, for the remaining subchannels  $(3, 2)^{2\tilde{n} - \tilde{m}}$ , we apply the non-feedback schemes to achieve:

$$\tilde{R}^{(2)} = \tilde{C}_{\text{no}}^{(3,2)} \times (2\tilde{n} - \tilde{m}) = 2(2\tilde{n} - \tilde{m}). \quad (67)$$

Combining all of the above, we get:  $(R, \tilde{R}) = (C_{\text{no}}, \tilde{C}_{\text{pf}})$ .

**(II)  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > n$ :** In this case, we do not have a sufficient amount of resources for achieving  $\tilde{R} = \tilde{C}_{\text{pf}}$ . The maximally achievable backward computation rate is saturated by  $\tilde{C}_{\text{no}} + n$  and this occurs when  $R = 0$ . On the other hand, under the constraint of  $R = C_{\text{no}}$ , what one can achieve for  $\tilde{R}$  is  $\tilde{C}_{\text{no}} + n - C_{\text{no}}$ .

**(III)  $n - C_{\text{no}} < \tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n$ :** This is the case in which we have a sufficient amount of resources for achieving  $\tilde{R} = \tilde{C}_{\text{pf}}$ , but not enough to achieve  $R = C_{\text{no}}$  at the same time. Hence aiming at  $\tilde{R} = \tilde{C}_{\text{pf}}$ ,  $R$  is saturated by  $n - (\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}})$ .

*C. Proof of (R3)  $\alpha \leq \frac{2}{3}$ , ( $\tilde{\alpha} \in [\frac{2}{3}, 1)$ ,  $\tilde{\alpha} \in (1, \frac{3}{2})$ )*

This computation rate region is almost the same as that of (R2). The only difference is that the sum computation rate bound now reads  $C_{\text{no}} + \tilde{m}$  instead of  $\tilde{C}_{\text{no}} + n$ . Hence, the shape of the region depends now on where  $C_{\text{pf}} - C_{\text{no}}$  lies in between  $\tilde{m} - \tilde{C}_{\text{no}}$  and  $\tilde{m}$ . As one can make similar arguments as those in the regime (R2), we omit the proof.

*D. Proof of (R4)  $\alpha \leq \frac{2}{3}$ ,  $\tilde{\alpha} \geq \frac{3}{2}$*

Recall in Remark 1 that  $C_{\text{pf}} - C_{\text{no}}$  indicates the amount of feedback that needs to be sent for achieving  $C_{\text{pf}}$  and we interpret  $\tilde{m} - \tilde{C}_{\text{pf}}$  as the remaining resource levels that can

potentially be utilized to aid forward computation. Whether or not  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$  (i.e., we have enough resource levels to achieve  $R = C_{\text{pf}}$ ), the shape of the above claimed region is changed. Note that the third inequality in the computation rate region becomes inactive when  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$ . Similarly, the last inequality is inactive when  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$ . Depending on these two conditions, we consider the following four subcases: (I)  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$ ; (II)  $C_{\text{pf}} - C_{\text{no}} > \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$ ; (III)  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > n - C_{\text{pf}}$ ; and (IV)  $C_{\text{pf}} - C_{\text{no}} > \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > n - C_{\text{pf}}$ .

The following achievability w.r.t. the elementary subchannels identified in Theorem 2 forms the basis.

**Lemma 4.** *The following computation rates are achievable:*

- (i) *For the pair of  $(m, n) = (1, 2)^i$  and  $(\tilde{m}, \tilde{n}) = (2, 1)^j$ :  $(R, \tilde{R}) = (\frac{4}{3}i, \frac{4}{3}j) = (C_{\text{pf}} \cdot i, \tilde{C}_{\text{pf}} \cdot j)$ . Here  $2i \geq j$  and  $2j \geq i$ .*  
(ii) *For the pair of  $(m, n) = (2, 3)^i$  and  $(\tilde{m}, \tilde{n}) = (2, 1)^j$ :  $(R, \tilde{R}) = (2i, \frac{4}{3}j) = (C_{\text{pf}} \cdot i, \tilde{C}_{\text{pf}} \cdot j)$ . Here  $3i \geq j$ .*

*Proof:* See Appendix D. ■

**(I)  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$ :** The first case is one in which there are enough resources available for enhancing the computation capacity up to perfect-feedback computation capacities in both directions. Hence we claim that the following computation rate region is achievable:  $(R, \tilde{R}) = (C_{\text{pf}}, \tilde{C}_{\text{pf}})$ . For efficient use of Theorem 2 and Lemma 4, we divide the regime (R4) into the following four sub-regimes: (R4-1)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \in [\frac{3}{2}, 2]$ ; (R4-2)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \geq 2$ ; (R4-3)  $\alpha \in [0, \frac{1}{2}]$ ,  $\tilde{\alpha} \in [\frac{3}{2}, 2]$ ; and (R4-4)  $\alpha \in [0, \frac{1}{2}]$ ,  $\tilde{\alpha} \geq 2$ . Here we provide details for (R4-1) and for the remaining sub-regimes, one can make a similar argument to show  $(R, \tilde{R}) = (C_{\text{pf}}, \tilde{C}_{\text{pf}})$ .

**(R4-1)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \in [\frac{3}{2}, 2]$ :** Here we use the fact that  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$  is equivalent to  $2n - 3m \leq \tilde{m}$  and that  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$  is equivalent to  $2\tilde{m} - 3\tilde{n} \leq n$ . Without loss of generality, let us assume  $2n - 3m \leq 2\tilde{m} - 3\tilde{n}$ . We now apply Lemma 4 (i) for the pair of  $(1, 2)^{2n-3m}$  and  $(2, 1)^{\min\{2\tilde{m}-3\tilde{n}, 2(2n-3m)\}}$ . Also we apply Lemma 4 (ii) for the pair of  $(2, 3)^{2m-n}$  and  $(2, 1)^{2\tilde{m}-3\tilde{n}-\min\{2\tilde{m}-3\tilde{n}, 2(2n-3m)\}}$ . Note that a tedious calculation guarantees the condition of (v):  $3(2m - n) \geq 2\tilde{m} - 3\tilde{n} - \min\{2\tilde{m} - 3\tilde{n}, 2(2n - 3m)\}$ . Lastly we apply the non-feedback schemes for the remaining subchannels  $(3, 2)^{2\tilde{n} - \tilde{m}}$ . A straightforward calculation yields:  $(R, \tilde{R}) = (C_{\text{pf}}, \tilde{C}_{\text{pf}})$ .

**(II)  $C_{\text{pf}} - C_{\text{no}} > \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$ :** In this case, there are two corner points to achieve. The first corner point is  $(R, \tilde{R}) = (C_{\text{no}} + \tilde{m} - \tilde{C}_{\text{pf}}, \tilde{C}_{\text{pf}})$ . The second corner point depends on where  $C_{\text{pf}} - C_{\text{no}}$  lies in between  $\tilde{m} - \tilde{C}_{\text{no}}$ ,  $\tilde{m}$  and beyond. See Fig. 13. For the cases of (II-1) and (II-2), the corner point reads  $(R, \tilde{R}) = (R, \tilde{R}) = (C_{\text{pf}}, \tilde{m} - (C_{\text{pf}} - C_{\text{no}}))$ , while for the case of (II-3),  $(R, \tilde{R}) = (C_{\text{no}} + \tilde{m}, 0)$ . Let us first focus on the first corner point where  $(R, \tilde{R}) = (C_{\text{no}} + \tilde{m} - \tilde{C}_{\text{pf}}, \tilde{C}_{\text{pf}})$ . Similar to (I), we provide details only for (R4-1).

**(R4-1)  $\alpha \in [\frac{1}{2}, \frac{2}{3}]$ ,  $\tilde{\alpha} \in [\frac{3}{2}, 2]$ :** Note that it suffices to consider the case where  $2(2\tilde{m} - 3\tilde{n}) \leq 2n - 3m$  since the other case implies that  $2(2\tilde{m} - 3\tilde{n}) > 2n - 3m = 3(C_{\text{pf}} - C_{\text{no}}) >$

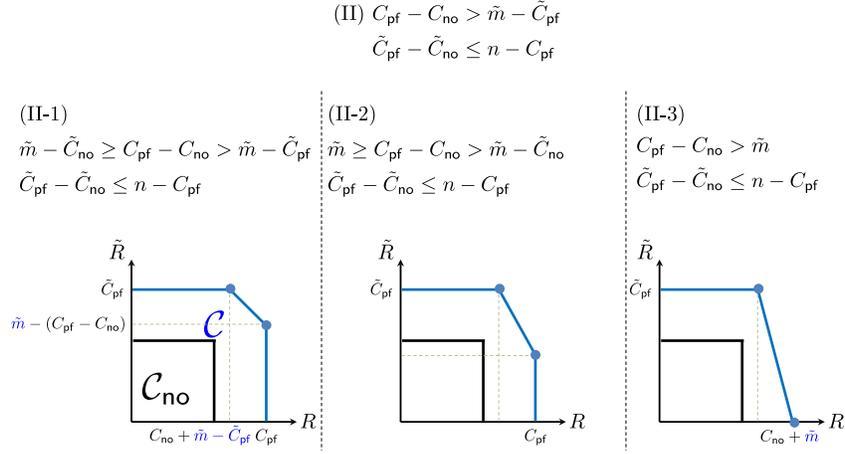


Fig. 13. Three types of shapes of an achievable computation rate region for the regime (R4)  $\alpha \leq \frac{2}{3}$ ,  $\tilde{\alpha} \geq \frac{3}{2}$  and the case (II)  $C_{\text{pf}} - C_{\text{no}} > \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} \leq n - C_{\text{pf}}$ .

$3(\tilde{m} - \tilde{C}_{\text{pf}}) = \tilde{m}$ . This condition holds when  $\tilde{\alpha} > 2$ , and therefore contradicts the condition  $\tilde{\alpha} \in [\frac{3}{2}, 2]$ .

We now apply Lemma 4 (i) for the pair of  $(1, 2)^{2(2\tilde{m}-3\tilde{n})}$  and  $(2, 1)^{2\tilde{m}-3\tilde{n}}$ . Also, we apply Lemma 4 (ii) for the pair of  $(1, 2)^{\tilde{m}-2(2\tilde{m}-3\tilde{n})}$  and  $(3, 2)^{2\tilde{n}-\tilde{m}}$ . Lastly we apply the non-feedback schemes for the remaining subchannels  $(1, 2)^{2n-3m-\tilde{m}}$  and  $(2, 3)^{2m-n}$ . Then we get:

$$\begin{aligned} R &= \frac{4}{3} \times 2(2\tilde{m} - 3\tilde{n}) + \frac{4}{3} \times (\tilde{m} - 2(2\tilde{m} - 3\tilde{n})) \\ &\quad + 1 \times (2n - 3m - \tilde{m}) + 2 \times (2m - n) \\ &= m + \frac{1}{3}\tilde{m} = C_{\text{no}} + \tilde{m} - \tilde{C}_{\text{pf}}, \\ \tilde{R} &= \frac{4}{3} \times (2\tilde{m} - 3\tilde{n}) + 2 \times (2\tilde{n} - \tilde{m}) = \frac{2}{3}\tilde{m} = \tilde{C}_{\text{pf}}. \end{aligned}$$

We are now ready to prove the second corner point which favors  $\tilde{R}$ . Depending on the quantity of  $C_{\text{pf}} - C_{\text{no}}$ , we have three subcases.

**(II-1)**  $\tilde{m} - \tilde{C}_{\text{pf}} < C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{no}}$ : For the regime of (R4-1), we showed that the following computation rate pair is achievable:  $(C_{\text{no}} + \tilde{m} - \tilde{C}_{\text{pf}}, \tilde{C}_{\text{pf}})$ . Now the idea is to tune the scheme which yields the above computation rate to prove the achievability of the second corner point. We use part of the backward channel for aiding forward computation instead of its own backward traffic. Specifically we utilize  $2n - 3m - \tilde{m}$  number of top levels in the backward channel once in three time slots in an effort to relay forward-message signal feedback. This naive change incurs one-to-one tradeoff, thus yielding:

$$\begin{aligned} R &= C_{\text{no}} + \tilde{m} - \tilde{C}_{\text{pf}} + \frac{1}{3}(2n - 3m - \tilde{m}) = C_{\text{pf}}, \\ \tilde{R} &= \tilde{C}_{\text{pf}} - \frac{1}{3}(2n - 3m - \tilde{m}) = \tilde{m} - (C_{\text{pf}} - C_{\text{no}}). \end{aligned}$$

**(II-2)**  $\tilde{m} - \tilde{C}_{\text{no}} < C_{\text{pf}} - C_{\text{no}} \leq \tilde{m}$ : Applying the same strategy as in (II-1), we get  $(C_{\text{pf}}, \tilde{m} - (C_{\text{pf}} - C_{\text{no}}))$ .

**(II-3)**  $C_{\text{pf}} - C_{\text{no}} > \tilde{m}$ : If we sacrifice all of the  $\tilde{m}$  direct links in the backward channel only for the purpose of assisting the forward computation, we can see that  $(R, \tilde{R}) = (C_{\text{no}} + \tilde{m}, 0)$  is achievable.

**(III)**  $C_{\text{pf}} - C_{\text{no}} \leq \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > n - C_{\text{pf}}$ : Similar to the previous case, this case requires the proof of two corner points. The first corner point is  $(R, \tilde{R}) = (C_{\text{pf}}, \tilde{C}_{\text{no}} + n - C_{\text{pf}})$ . The second corner point is depends on where  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}$  lies in between  $n - C_{\text{no}}$ ,  $n$  and beyond. We omit details.

**(IV)**  $C_{\text{pf}} - C_{\text{no}} > \tilde{m} - \tilde{C}_{\text{pf}}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > n - C_{\text{pf}}$ : For the following case, it suffices to consider only (R4-4)  $\alpha \in [0, \frac{1}{2}]$ ,  $\tilde{\alpha} \geq 2$  given that  $2n - 3m = 3(C_{\text{pf}} - C_{\text{no}}) > 3(\tilde{m} - \tilde{C}_{\text{pf}}) = \tilde{m} \geq \tilde{m} - \frac{3}{2}\tilde{n} \stackrel{(a)}{>} \frac{1}{2}n$ , where (a) follows because we consider  $2\tilde{m} - 3\tilde{n} > n$  (or equivalently,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > n - C_{\text{pf}}$ ). With the first and the last formulae, this clearly implies that  $\alpha < \frac{1}{2}$ . Similarly,  $2\tilde{m} - 3\tilde{n} = 3(\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}) > 3(n - C_{\text{pf}}) = n \geq n - \frac{3}{2}m \stackrel{(b)}{>} \frac{1}{2}\tilde{m}$ , where (b) follows as we consider  $2n - 3m > \tilde{m}$ . This implies that  $\tilde{\alpha} > 2$ . For the regime of (R4-4), one can make arguments similar to those in (II) and (III). Specifically, the first corner point (as well as the second corner point) depends on where  $C_{\text{pf}} - C_{\text{no}}$  (and  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}}$ ) lies in between  $\tilde{m} - \tilde{C}_{\text{no}}$  (and  $n - C_{\text{no}}$ );  $\tilde{m}$  (and  $n$  respectively) and beyond. As each condition takes three types, there can be nine cases in total. However, of the nine cases, the case in which  $C_{\text{pf}} - C_{\text{no}} > \tilde{m}$ ,  $\tilde{C}_{\text{pf}} - \tilde{C}_{\text{no}} > n$  implies that  $(2n - 3m) + (2\tilde{m} - 3\tilde{n}) > 3\tilde{m} + 3n$ . This is equivalent to  $0 > -n - 3m > \tilde{m} + 3\tilde{n} > 0$ , which results in a contradiction. Therefore, we can conclude that there are eight cases in total. Of the eight cases, it is found that this case takes two types of corner point: Either  $(R, \tilde{R}) = (C_{\text{no}} + \tilde{m} - \tilde{C}_{\text{pf}}, \tilde{C}_{\text{pf}})$  or  $(R, \tilde{R}) = (C_{\text{pf}}, \tilde{C}_{\text{no}} + n - C_{\text{pf}})$ . If the first corner point is  $(C_{\text{no}} + \tilde{m} - \tilde{C}_{\text{pf}}, \tilde{C}_{\text{pf}})$ , the second corner point corresponds to that in (II); otherwise the corner point corresponds to that in (III). As we already described the idea of showing first and second corner point explicitly, we omit details.

#### APPENDIX D PROOF OF LEMMA 4

We now provide the proof of Lemma 4. Note that we demonstrated the case of (i) IV-B, and a slight modification of the scheme in Section allows us to achieve the desired computation rate pair. Hence we will provide the achievability for (ii).

(ii)  $(m, n) = (2, 3)^i$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)^j$ : We see in Fig. 12 that  $(R, \tilde{R}) = (2, \frac{4}{3})$  is achievable for the case of  $(m, n) = (2, 3)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)$ . Consider the case of  $(m, n) = (2, 3)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)^3$ . For the remaining two  $(2, 1)$  backward channels, we repeat the above procedure w.r.t. new backward symbols. Note that feedback transmissions can be performed at time 1 and 2. This gives  $(R, \tilde{R}) = (2, \frac{4}{3} \times 3) = (2, 3)$ . In this case, it suffices to show the scheme for  $(m, n) = (2, 3)$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)^3$ . Note that  $(m, n) = (2, 3)^i$ ,  $(\tilde{m}, \tilde{n}) = (2, 1)^{3i}$  is a simple multiplication with  $i$ . Note that as long as  $3i \geq j$ , the claimed computation rate pair is still achievable. This completes the proof.

## REFERENCES

- [1] S. Shin and C. Suh, "Capacity of a two-way function multicast channel," in *Proc. Allerton Conf. Commun., Control, Comput.*, Oct. 2017, pp. 125–133.
- [2] S. Shin and C. Suh, "Two-way function computation," in *Proc. Allerton Conf. Commun., Control, Comput.*, Oct. 2014, pp. 1309–1316.
- [3] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley Symp. Math. Stat. Prob.*, Jun. 1961, pp. 611–644.
- [4] T. S. Han, "A general coding scheme for the two-way channel," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 1, pp. 35–44, Jan. 1984.
- [5] G. Dueck, "The capacity region of the two-way channel can exceed the inner bound," *Inf. Control*, vol. 40, no. 3, pp. 258–266, Mar. 1979.
- [6] J. Schalkwijk, "On an extension of an achievable rate region for the binary multiplying channel," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 445–448, May 1983.
- [7] Z. Zhang, T. Berger, and J. Schalkwijk, "New outer bounds to capacity regions of two-way channels," *IEEE Trans. Inf. Theory*, vol. 32, no. 3, pp. 383–386, May 1986.
- [8] A. P. Hekstra and F. M. J. Willems, "Dependence balance bounds for single-output two-way channels," *IEEE Trans. Inf. Theory*, vol. 35, no. 1, pp. 44–53, Jan. 1989.
- [9] C. E. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 8–19, Sep. 1956.
- [10] S. Ihara, "Capacity of discrete time Gaussian channel with and without feedback, I," *Mem. Fac. Sci. Kochi Univ. (Math.)*, vol. 9, pp. 21–36, Sep. 1988.
- [11] S. Ihara and K. Yanagi, "Capacity of discrete time Gaussian channel with and without feedback, II," *Jpn. J. Appl. Math.*, vol. 6, pp. 245–258, Jun. 1989.
- [12] T. M. Cover and S. Pombra, "Gaussian feedback capacity," *IEEE Trans. Inf. Theory*, vol. 35, no. 1, pp. 37–43, Jan. 1989.
- [13] Y.-H. Kim, "Feedback capacity of the first-order moving average Gaussian channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3063–3079, Jul. 2006.
- [14] H. H. Permuter, H. Asnani, and T. Weissman, "Capacity of a post channel with and without feedback," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6041–6057, Oct. 2014.
- [15] N. Gaarder and J. Wolf, "The capacity region of a multiple-access discrete memoryless channel can increase with feedback," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 100–102, Jan. 1975.
- [16] L. H. Ozarow, "The capacity of the white Gaussian multiple access channel with feedback," *IEEE Trans. Inf. Theory*, vol. 30, no. 4, pp. 623–629, Jul. 1984.
- [17] L. H. Ozarow and S. K. Leung-Yan-Cheong, "An achievable region and outer bound for the Gaussian broadcast channel with feedback," *IEEE Trans. Inf. Theory*, vol. 30, no. 4, pp. 667–671, Jul. 1984.
- [18] G. Kramer, "Feedback strategies for white Gaussian interference networks," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1423–1438, Jun. 2002.
- [19] C. Suh and D. N. C. Tse, "Feedback capacity of the Gaussian interference channel to within 2 bits," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 2667–2685, May 2011.
- [20] S. Yang and D. Tuninetti, "Interference channel with generalized feedback (a.k.a. With source cooperation): Part I: Achievable region," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 2686–2710, May 2011.
- [21] C. Suh, I. Wang, and D. Tse, "Two-way interference channels," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2801–2805.
- [22] Z. Cheng and N. Devroye, "Two-way networks: When adaptation is useless," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1793–1813, Mar. 2014.
- [23] S. M. Perlaza, R. Tandon, H. V. Poor, and Z. Han, "Perfect output feedback in the two-user decentralized interference channel," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5441–5462, Oct. 2015.
- [24] C. Suh, J. Cho, and D. Tse, "Two-way interference channel capacity: How to have the cake and eat it too," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4259–4281, Mar. 2018.
- [25] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 23, pp. 755–764, Apr. 2005.
- [26] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.
- [27] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.
- [28] C. Suh and M. Gastpar, "Interactive function computation," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 896–900.
- [29] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "Wireless network information flow: A deterministic approach," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 1872–1905, Apr. 2011.
- [30] C. Suh, N. Goela, and M. Gastpar, "Computation in multicast networks: Function alignment and converse theorems," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 1866–1877, Apr. 2016.
- [31] S. Mohajer, S. N. Diggavi, C. Fragouli, and D. N. C. Tse, "Approximate capacity of a class of Gaussian interference-relay networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 2837–2864, May 2011.
- [32] S. Rini, D. Tuninetti, and N. Devroye, "New inner and outer bounds for the memoryless cognitive interference channel and some new capacity results," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4087–4109, Jul. 2011.
- [33] V. Prabhakaran and P. Viswanath, "Interference channels with source cooperation," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 156–186, Jan. 2011.
- [34] J. Chen, A. Ozgur, and S. Diggavi, "Feedback through overhearing," in *Proc. Allerton Conf. Commun., Control, Comput.*, Oct. 2014, pp. 358–365.
- [35] S. H. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung, "Noisy network coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 3132–3152, May 2011.
- [36] B. Nazer and M. Gastpar, "Compute-and forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.
- [37] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 3, pp. 439–441, May 1983.
- [38] T. S. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 1, pp. 49–60, Jan. 1981.

**Seiyun Shin** is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at University of Illinois, Urbana–Champaign (UIUC). He received the B.S. and M.S. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2012 and 2015, respectively. He is a recipient of Kwanjeong Educational Foundation Fellowship from 2019. His research interests lie in information theory and machine learning.

**Changho Suh** (S'10–M'12) is an Associate Professor in the School of Electrical Engineering at Korea Advanced Institute of Science and Technology (KAIST). He received the B.S. and M.S. degrees in Electrical Engineering from KAIST in 2000 and 2002 respectively, and the Ph.D. degree in Electrical Engineering and Computer Sciences from University of California, Berkeley in 2011. From 2011 to 2012, he was a postdoctoral associate at the Research Laboratory of Electronics in Massachusetts Institute of Technology. From 2002 to 2006, he had been with the Telecommunication R&D Center, Samsung Electronics.

Dr. Suh received the 2018 IEIE/IEEE Joint Award, the 2015 IEIE Haedong Young Engineer Award, a 2015 Bell Labs Prize finalist, the 2013 IEEE Communications Society Stephen O. Rice Prize, the 2011 David J. Sakrison Memorial Prize, and the 2009 IEEE International Symposium on Information Theory Best Student Paper Award.