Fifty-Fifth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
October 3-6, 2017

Capacity of a Two-way Function Multicast Channel

Seiyun Shin
ETRI, South Korea
Email: seiyun.shin@etri.re.kr

Changho Suh
KAIST, South Korea
Email: chsuh@kaist.ac.kr

***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, we can sometime get all the way to *perfect-feedback computation capacities simultaneously in both directions.* This result draws a parallel with the recent result developed in the context of two-way interference channels [1].**

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 in [2]. However, we are still lacking in our understanding of how to treat two-way information exchanges, 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 [3]. Subsequent work demonstrated that feedback provides a gain for point-to-point channels with memory [4], [5] as well as for many multi-user channels [6]–[8]. For many scenarios, however, capacity improvements due to feedback are rather modest.

On the contrary, one notable result in [9] has changed the traditional viewpoint on the role of feedback. It is shown in [9] that feedback offers more significant capacity gains for the Gaussian interference channel. Subsequent works [1], [10], [11] show more promise on the use of feedback. In particular, [1] demonstrates a very interesting result: Not only feedback can yield a net increase in capacity, we can sometime get *perfect feedback capacities* simultaneously in both directions.

In this work, 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 [12] and cloud computing scenarios [13], [14]. For an idealistic scenario where feedback links are perfect with infinite capacities and are given for free, Suh-Gastpar [15] have shown that feedback provides a significant gain also for computation. However, the result in [15] assumes a dedicated infinite-capacity feedback link as in [9]. As an effort to explore a net gain that reflects feedback cost, [16] investigated a two-way setting of the function multicast channel considered in [15] 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. 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 [1]) has been unanswered.

In this work, we characterize the computation capacity region of the two-way function multicast channel via a new capacity-achieving scheme. In particular, we consider a deterministic model [17] which well captures key properties of the wireless Gaussian channel. 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 [1], interestingly, 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.

Our achievability builds upon the scheme in [1] where feedback allows the exploitation of effectively future information as side information via retrospective decoding (to be detailed later; see Remark 2). A key distinction relative to [1] 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 network decomposition in [18] for ease of achievability proof.

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

978-1-5386-3266-6/17/$31.00 ©2017 IEEE

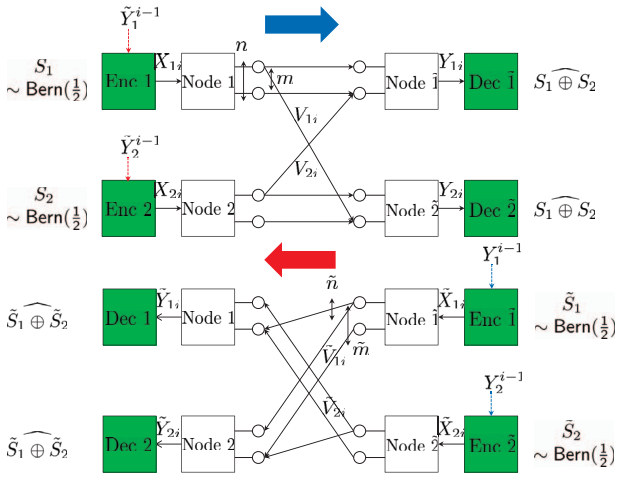

Fig. 1. Four-node ADT deterministic network.

consists of the 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 uses of the network, node k $(k = 1, 2)$ wishes to transmit its own message S_k^K, while node $\tilde{k}$ $(\tilde{k} = \tilde{1}, \tilde{2})$ wishes to transmit its own message $\tilde{S}_k^{\tilde{K}}$. 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 $\mathsf{Bern}(\frac{1}{2})$. Here we use shorthand notation to indicate the sequence up to K (or $\tilde{K}$), e.g., $S_1^K := (S_{11}, \cdots, S_{1K})$. 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 $(\neq k)$. The received signals at node k and $\tilde{k}$ are then given by

$$Y_1 = \mathbf{G}^{q-n} X_1 \oplus \mathbf{G}^{q-m} X_2, \; \tilde{Y}_1 = \tilde{\mathbf{G}}^{\tilde{q}-\tilde{n}} \tilde{X}_1 \oplus \tilde{\mathbf{G}}^{\tilde{q}-\tilde{m}} \tilde{X}_2,$$
$$Y_2 = \mathbf{G}^{q-m} X_1 \oplus \mathbf{G}^{q-n} X_2, \; \tilde{Y}_2 = \tilde{\mathbf{G}}^{\tilde{q}-\tilde{m}} \tilde{X}_1 \oplus \tilde{\mathbf{G}}^{\tilde{q}-\tilde{n}} \tilde{X}_2,$$

where $\mathbf{G}$ and $\tilde{\mathbf{G}}$ are shift matrices and operations are performed in $\mathbb{F}_2$: $[\mathbf{G}]_{ij} = \mathbf{1}\{i = j+1\}$ $(1 \le i, j \le q)$, $[\tilde{\mathbf{G}}]_{ij} = \mathbf{1}\{i = j+1\}$ $(1 \le i, j \le \tilde{q})$.

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 modulo-2 sums of S_1^K and S_2^K, i.e., $\{S_{1i} \oplus S_{2i}\}_{i=1}^K$. Similarly node k wishes to compute $\{\tilde{S}_{1j} \oplus \tilde{S}_{2j}\}_{j=1}^{\tilde{K}}$ from its received signals $\tilde{Y}_k^N$. We say that a computation rate pair $(R, \tilde{R})$ is achievable if there exists a family of codebooks and encoder/decoder functions such that the decoding error probabilities go to zero as code length N tends to infinity. Here $R := \frac{K}{N}$ and $\tilde{R} := \frac{\tilde{K}}{N}$. The capacity region $\mathcal{C}$ is the closure of the set of achievable rate pairs.

III. Main Results

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

$$R \le C_{\mathsf{pf}}, \tag{1}$$
$$\tilde{R} \le \tilde{C}_{\mathsf{pf}}, \tag{2}$$
$$R + \tilde{R} \le m + \tilde{m}, \tag{3}$$
$$R + \tilde{R} \le n + \tilde{n}, \tag{4}$$

where C_{pf} and $\tilde{C}_{\mathsf{pf}}$ indicate the perfect-feedback computation capacities in the forward and backward channels respectively (see (7) and (8) in Baseline 2 for detailed formulas).

Proof: See Section IV for the achievability proof. Here we provide the converse proof. Note that the bounds of (1) and (2) are the perfect-feedback bounds in [15], the proof of which will be omitted here. The upper bound of (3) is due to a cut-set argument; see below for a detailed proof. By symmetry, the proof of (4) can be derived in a similar manner.

Proof of (3) *(Cut-set Bound):* Starting with Fano's inequality, we get

$$\begin{aligned}
&N\left(R + \tilde{R} - \epsilon_N\right) \\
&\le I\left(S_1^K \oplus S_2^K, \tilde{S}_1^{\tilde{K}} \oplus \tilde{S}_2^{\tilde{K}}; Y_2^N, \tilde{Y}_2^N, S_2^K, \tilde{S}_2^{\tilde{K}}\right) \\
&= I\left(S_1^K \oplus S_2^K, \tilde{S}_1^{\tilde{K}} \oplus \tilde{S}_2^{\tilde{K}}; Y_2^N, \tilde{Y}_2^N | S_2^K, \tilde{S}_2^{\tilde{K}}\right) \\
&= H\left(Y_2^N, \tilde{Y}_2^N | S_2^K, \tilde{S}_2^{\tilde{K}}\right) \\
&= \sum H\left(Y_{2i}, \tilde{Y}_{2i} | S_2^K, \tilde{S}_2^{\tilde{K}}, Y_2^{i-1}, \tilde{Y}_2^{i-1}\right) \\
&\overset{(a)}{=} \sum H\left(Y_{2i}, \tilde{Y}_{2i} | S_2^K, \tilde{S}_2^{\tilde{K}}, Y_2^{i-1}, \tilde{Y}_2^{i-1}, X_{2i}, \tilde{X}_{2i}\right) \\
&\overset{(b)}{\le} \sum H\left(Y_{2i} | X_{2i}\right) + H\left(\tilde{Y}_{2i} | \tilde{X}_{2i}\right) \\
&\overset{(c)}{\le} \sum H\left(V_{1i}\right) + H\left(\tilde{V}_{1i}\right) \le \sum N\left(m + \tilde{m}\right)
\end{aligned}$$

where (a) follows from the fact that X_{2i} is a function of $(S_2, \tilde{Y}_2^{i-1})$ and $\tilde{X}_{2i}$ is a function of $(\tilde{S}_2, Y_2^{i-1})$; (b) and (c) follow from the fact that conditioning reduces entropy. If $R + \tilde{R}$ is achievable, then $\epsilon_N \to 0$ as N tends to infinity, and hence $R + \tilde{R} \le m + \tilde{m}$. ∎

For comparison to our result, we state two baselines: (1) The capacity region for the non-interactive scenario in which there is no interaction among the signals arriving from different nodes; (2) the capacity for the perfect-feedback scenario in which feedback is given for free to aid computations in both direction.

Baseline 1 (Non-interaction Computation Capacity [18]): Let $\alpha := \frac{m}{n}$ and $\tilde{\alpha} := \frac{\tilde{m}}{\tilde{n}}$. The computation capacity region

$\mathcal{C}_{\mathsf{no}}$ for the non-interactive scenario is the set of $(R, \tilde{R})$ such that $R \leq C_{\mathsf{no}}$ and $\tilde{R} \leq \tilde{C}_{\mathsf{no}}$ where

$$C_{\mathsf{no}} = \begin{cases} \min\left\{m, \frac{2}{3}n\right\}, & \alpha < 1, \\ \min\left\{n, \frac{2}{3}m\right\}, & \alpha > 1, \\ n, & \alpha = 1, \end{cases} \tag{5}$$

$$\tilde{C}_{\mathsf{no}} = \begin{cases} \min\left\{\tilde{m}, \frac{2}{3}\tilde{n}\right\}, & \tilde{\alpha} < 1, \\ \min\left\{\tilde{n}, \frac{2}{3}\tilde{m}\right\}, & \tilde{\alpha} > 1, \\ \tilde{n}, & \tilde{\alpha} = 1. \end{cases} \tag{6}$$

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

Baseline 2 (Perfect-feedback Computation Capacity [15]): The computation capacity region $\mathcal{C}_{\mathsf{pf}}$ for the perfect-feedback scenario is the set of $(R, \tilde{R})$ such that $R \leq C_{\mathsf{pf}}$ and $\tilde{R} \leq \tilde{C}_{\mathsf{pf}}$ where

$$C_{\mathsf{pf}} = \begin{cases} \frac{2}{3}n, & \alpha < 1, \\ \frac{2}{3}m, & \alpha > 1, \\ n, & \alpha = 1, \end{cases} \tag{7}$$

$$\tilde{C}_{\mathsf{pf}} = \begin{cases} \frac{2}{3}\tilde{n}, & \tilde{\alpha} < 1, \\ \frac{2}{3}\tilde{m}, & \tilde{\alpha} > 1, \\ \tilde{n}, & \tilde{\alpha} = 1. \end{cases} \tag{8}$$

With Theorem 1 and Baseline 1, one can readily see that feedback offers a gain (in terms of capacity region) as long as $(\alpha \notin [\frac{2}{3}, \frac{3}{2}], \tilde{\alpha} \notin [\frac{2}{3}, \frac{3}{2}])$. A careful inspection reveals that there are channel regimes in which one can enhance C_{no} (or $\tilde{C}_{\mathsf{no}}$) 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 $(R, \tilde{R}) = (C_{\mathsf{no}} + \delta, \tilde{C}_{\mathsf{no}} + \tilde{\delta})$ for some $\delta \geq 0$ and $\tilde{\delta} \geq 0$ such that $\max(\delta, \tilde{\delta}) > 0$.

A tedious yet straightforward calculation with this definition leads us to identify channel regimes which exhibit an interaction gain. See the regimes marked in light blue in Fig. 2.

We also find the regimes in which feedback does increase capacity but interaction cannot provide such increase, meaning that whenever $\delta > 0$, $\tilde{\delta}$ mush be $-\delta$ and vice versa. The regimes are $(\alpha < \frac{2}{3}, \tilde{\alpha} < \frac{2}{3})$, $(\alpha > \frac{3}{2}, \tilde{\alpha} > \frac{3}{2})$. One can readily check that this follows from the cut-set bounds (3) and (4).

Achieving perfect feedback capacities: It is noteworthy to mention that there exist channel regimes in which both δ and $\tilde{\delta}$ can be strictly positive. This implies that for the regimes not only feedback does not sacrifice one transmission for the other, it can actually improve both simultaneously. As in the two-way interference channel [1], this occurs for the computation setting as well: more interestingly, the gains δ and $\tilde{\delta}$ can reach up to the maximal feedback gains, reflected in $C_{\mathsf{pf}} - C_{\mathsf{no}}$ and $\tilde{C}_{\mathsf{pf}} - \tilde{C}_{\mathsf{no}}$ respectively. The dark blue regimes in Fig. 2 indicate such channel regimes when γ $(:= \frac{\tilde{n}}{n}) \geq 1$. Note that such regimes depend on γ. The amount of feedback that one can send is limited by available resources, which is affected by the channel asymmetry parameter γ.

The following corollary identifies channel regimes in which achieving perfect-feedback capacities in both directions is possible.

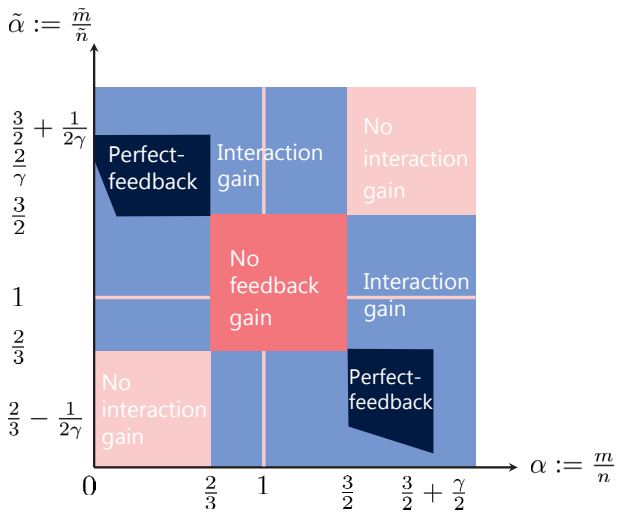

Fig. 2. Gain-vs-nogain picture: The plot is over two key parameters: α and $\tilde{\alpha}$, where α is the ratio of the interference-to-noise ratio (in dB) to the signal-to-noise ratio (in dB) of the forward channel and $\tilde{\alpha}$ is the corresponding quantity of the backward channel. The parameter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

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

$$\textbf{(I)}\ 0 \leq \alpha < \frac{2}{3}, \tilde{\alpha} > \frac{3}{2}, \left\{ \begin{matrix} C_{\mathsf{pf}} - C_{\mathsf{no}} \leq \tilde{m} - \tilde{C}_{\mathsf{pf}}, \\ \tilde{C}_{\mathsf{pf}} - \tilde{C}_{\mathsf{no}} \leq n - C_{\mathsf{pf}} \end{matrix} \right\},$$

$$\textbf{(II)}\ \alpha > \frac{3}{2}, 0 \leq \tilde{\alpha} < \frac{2}{3}, \left\{ \begin{matrix} \tilde{C}_{\mathsf{pf}} - \tilde{C}_{\mathsf{no}} \leq m - C_{\mathsf{pf}}, \\ C_{\mathsf{pf}} - C_{\mathsf{no}} \leq \tilde{n} - \tilde{C}_{\mathsf{pf}} \end{matrix} \right\}.$$

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 capacities in both directions bears a resemblance to the one found in the two-way interference channel [1]: Interaction enables full-utilization of available resources, whereas the dearth of interaction limits that of those. We will elaborate this for the considered regime in Corollary 1 : $0 \leq \alpha < \frac{2}{3}, \tilde{\alpha} > \frac{3}{2}$.

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-interaction case, observe from Baseline 1 that some resources are under-utilized; specifically one can interpret $n - C_{\mathsf{no}}$ and $\tilde{m} - \tilde{C}_{\mathsf{no}}$ as the remaining resource levels (resource holes) 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-interactive case. Note that $C_{\mathsf{pf}} - C_{\mathsf{no}}$ represents the amount

of feedback that needs to be sent for achieving C_{pf}. Hence, the condition $C_{\mathsf{pf}} - C_{\mathsf{no}} \le \tilde{m} - \tilde{C}_{\mathsf{pf}}$ (similarly $\tilde{C}_{\mathsf{pf}} - \tilde{C}_{\mathsf{no}} \le n - C_{\mathsf{pf}}$) in Corollary 1 implies that as long as we have enough resource holes, we can get all the way to perfect feedback capacity. We will later provide an intuition as to why feedback can do so while describing our achievability; see Remark 2 in particular. □

IV. Proof of Achievability

Our achievability proof consists of three parts. We will first provide two 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 while leaving the detailed proof in the full version [19].

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

We first review the perfect-feedback scheme [15], which we will use as a baseline for 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.

1) Perfect-feedback strategy: 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 F_ℓ, say $F_\ell := a_\ell \oplus b_\ell$ where a_ℓ (or b_ℓ) denotes a source symbol of node 1 (or 2). For ease of presentation, we use a simple notation F_i. In the first stage, we send forward symbols at node 1 and 2. In time 1, node 1 sends (a_1, a_2); similarly, node 2 sends (b_2, b_1). Then node $\tilde{1}$ gets F_2 $(:= a_2 \oplus b_2)$; and node $\tilde{2}$ gets F_1 $(:= a_1 \oplus b_1)$. As in the first time slot, node 1 and 2 deliver (a_3, a_4) and (b_4, b_3) respectively at time 2. Then node $\tilde{1}$ and $\tilde{2}$ obtain F_4 and F_3 respectively. Note that until the end of time 2, (F_1, F_3) are not delivered yet to node $\tilde{1}$. Similarly (F_2, F_4) are missing at node $\tilde{2}$.

Feedback can however accomplish the computation of these interested functions. 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 get (F_1, F_3). Similarly, node 2 can get (F_2, F_4) from node $\tilde{1}$.

Now the strategy in Stage 2 is to forward all of these feedback functions at time 3. Node $\tilde{1}$ then gets F_1 cleanly on the top level. On 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, node $\tilde{1}$ and $\tilde{2}$ can compute four modulo-2 sum functions during three time slots, thus achieving $R = \frac{4}{3}$ $(= C_{\mathsf{pf}})$.

In our model, however, feedback is provided in the limited fashion since 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, and this tension incurs a new challenge. We develop an achievable scheme that can completely resolve the tension, thus achieving the perfect-feedback performance.

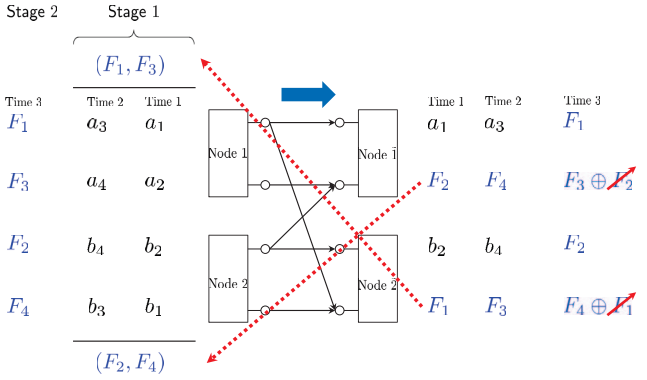

Fig. 3. A perfect-feedback scheme for $(m, n) = (1, 2)$ model.

2) Achievability: Like the perfect-feedback case, our scheme consists of two stages. The first stage consists of $2L$ time slots; and the second stage consists of L time slots. During the first stage, the number $4L$ and $4(L-1)$ of fresh symbols are transmitted through the forward and backward channels respectively. And no fresh symbols are transmitted in the second stage. In this example, we claim that the following rate pair is achievable: $(R, \tilde{R}) = (\frac{4L}{3L}, \frac{4(L-1)}{3L}) = (\frac{4}{3}, \frac{4L-4}{3L})$. In other words, during the total $3L$ time slots, our scheme ensures $4L$ and $4L-4$ forward and backward-message computations. As $L \to \infty$, we get the desired result: $(R, \tilde{R}) \to (\frac{4}{3}, \frac{4}{3}) = (C_{\mathsf{pf}}, \tilde{C}_{\mathsf{pf}})$.

Stage 1: In this stage, each node superimposes fresh symbols and feedback symbols. Details are as follows.

Time 1 & 2: Node 1 sends (a_1, a_2); and node 2 sends (b_2, b_1). Node $\tilde{1}$ and $\tilde{2}$ then get (a_1, F_2) and (b_2, F_1) respectively. Now observe that F_1 and F_2 are not delivered yet to node $\tilde{1}$ and $\tilde{2}$ respectively. In an attempt to satisfy these demands, note that the perfect-feedback strategy is to feed back F_2 from node $\tilde{1}$ to node 2, and to feed back F_1 from node $\tilde{2}$ to node 1.

A similar feedback strategy is employed in our backward channel. A distinction is that fresh backward symbols are superimposed. Specifically node $\tilde{1}$ delivers $(\tilde{a}_2 \oplus F_2, \tilde{a}_1 \oplus a_1)$, the mixtures of the fresh symbols $(\tilde{a}_2, \tilde{a}_1)$ and the feedback signals (F_2, a_1). Similarly node $\tilde{2}$ delivers $(\tilde{b}_1 \oplus F_1, \tilde{b}_2 \oplus b_2)$. Note that unlike the perfect-feedback strategy, not only F_2 on top is transmitted, but also a_1 on bottom is transmitted. And these two signals (F_2, a_1) are exactly the ones received at time 1. Node 1 and 2 then receive $(\tilde{b}_1 \oplus F_1, \tilde{F}_2 \oplus a_2)$ and $(\tilde{a}_2 \oplus F_2, \tilde{F}_1 \oplus b_1)$ respectively. Notice here that exploiting its own symbols, each node can obtain the interested function. Node 1 can obtain $\tilde{F}_2$ from $\tilde{F}_2 \oplus a_2$ by exploiting its own symbol a_2. Similarly, node 2 can obtain $\tilde{F}_1$.

In time 2, we repeat this w.r.t. new symbols. As a result, node $\tilde{1}$ and $\tilde{2}$ get (a_3, F_4) and (b_4, F_3). respectively, while node 1 and 2 get $(\tilde{b}_3 \oplus F_3, \tilde{F}_4 \oplus a_4)$ and $(\tilde{a}_4 \oplus F_4, \tilde{F}_3 \oplus b_3)$.

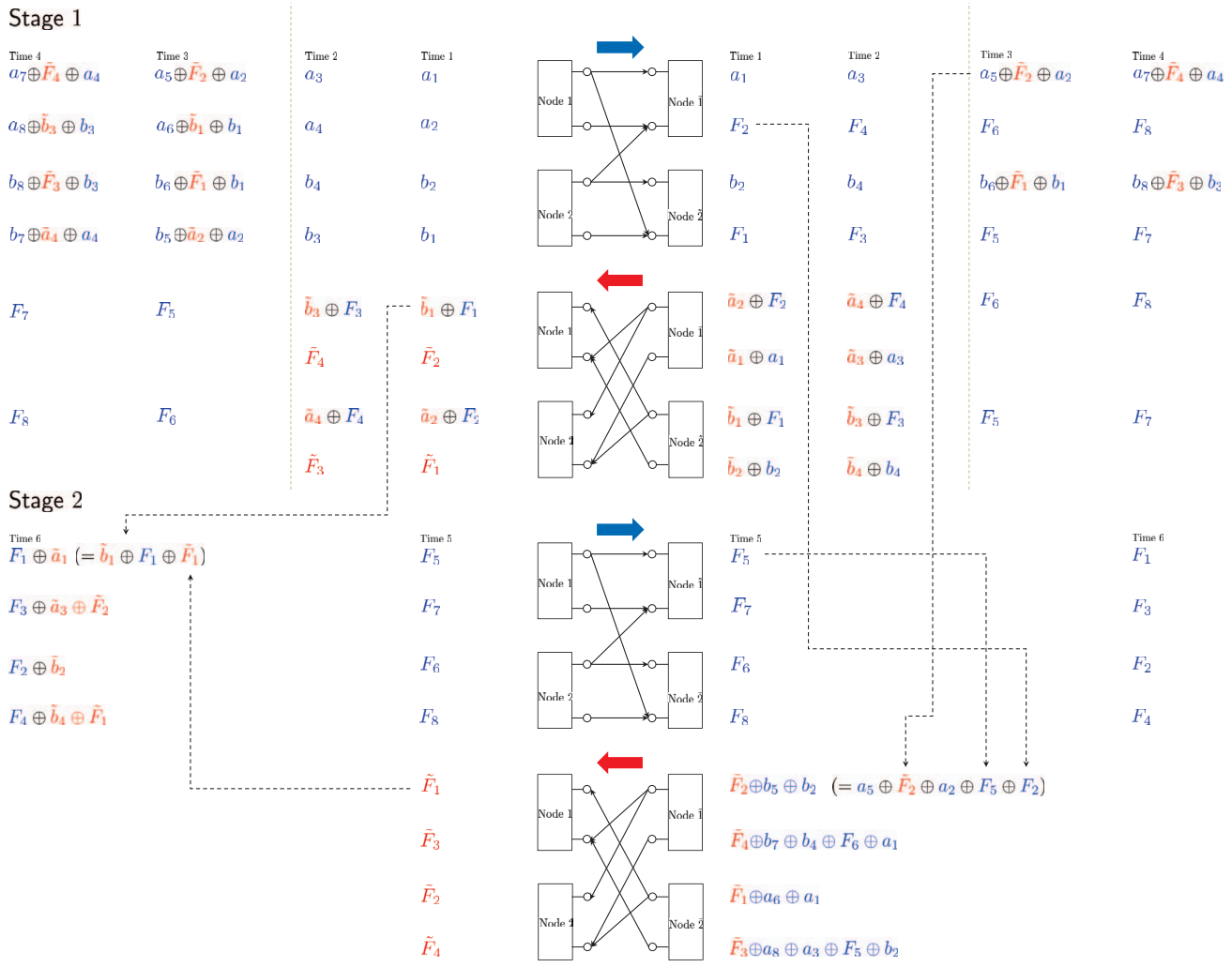

Fig. 4. An achievable scheme for $(m,n)=(1,2)$, $(\tilde{m},\tilde{n})=(2,1)$, and $L=2$.

Similar to the first time slot, node 1 and 2 utilize their own symbols as side information to obtain $\tilde{F}_4$ and $\tilde{F}_3$ respectively.

Time ℓ: For time $\ell=3,\cdots,2L$, the transmission signals at node 1 and 2 are:

$$\text{node } 1: \begin{bmatrix} a_{2\ell-1} \\ a_{2\ell} \end{bmatrix} + \begin{bmatrix} \tilde{F}_{2(\ell-2)} \oplus a_{2(\ell-2)} \\ \tilde{b}_{2(\ell-2)-1} \oplus F_{2(\ell-2)-1} \oplus a_{2(\ell-2)-1} \end{bmatrix},$$

$$\text{node } 2: \begin{bmatrix} b_{2\ell} \\ b_{2\ell-1} \end{bmatrix} + \begin{bmatrix} \tilde{F}_{2(\ell-2)-1} \oplus b_{2(\ell-2)-1} \\ \tilde{a}_{2(\ell-2)} \oplus F_{2(\ell-2)} \oplus b_{2(\ell-2)} \end{bmatrix}.$$

Similarly, for time $\ell=3,\cdots,2L-2$, node $\tilde{1}$ and $\tilde{2}$ deliver:

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

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

Note that the transmitted signal of each node consists of two parts: Fresh symbols, e.g., $(a_{2\ell-1},a_{2\ell})$ at node 1; and feedback signals, e.g., $(\tilde{F}_{2(\ell-2)}\oplus a_{2(\ell-2)}, \tilde{b}_{2(\ell-2)-1}\oplus F_{2(\ell-2)-1}\oplus a_{2(\ell-2)-1})$.

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

Note that until time $2L$, a total of $4L$ forward symbols are delivered $(a_{2\ell-1},a_{2\ell},b_{2\ell-1},b_{2\ell})$, for $\ell=1,\cdots,2L$. Similarly, a total of $4(L-1)$ backward symbols are delivered.

One can check that node $\tilde{1}$ and $\tilde{2}$ can obtain $\{F_{2\ell}\}_{\ell=1}^{2L}$ and $\{F_{2\ell-1}\}_{\ell=1}^{2L}$ respectively. Similarly, node 1 and 2 can obtain $\{\tilde{F}_{2\ell}\}_{\ell=1}^{2(L-1)}$ and $\{\tilde{F}_{2\ell-1}\}_{\ell=1}^{2(L-1)}$ respectively. Recall that among the total $4L$ and $4L-4$ forward and backward functions, $\{F_{2\ell-1}\}_{\ell=1}^{2L}$ and $\{F_{2\ell}\}_{\ell=1}^{2L}$ are not delivered yet to node $\tilde{1}$ and $\tilde{2}$ respectively. Similarly $\{\tilde{F}_{2\ell-1}\}_{\ell=1}^{2(L-1)}$ and $\{\tilde{F}_{2\ell}\}_{\ell=1}^{2(L-1)}$ are missing at node 1 and 2 respectively.

For ease of understanding, Fig. 4 illustrates a simple case of $L=2$. In time 3, node $\tilde{1}$ gets $(a_5\oplus\tilde{F}_2\oplus a_2, F_6\oplus\tilde{a}_1)$; and node $\tilde{2}$ gets $(b_6\oplus\tilde{F}_1\oplus b_1, F_5\oplus\tilde{b}_2)$. Note that using their own symbols $\tilde{a}_1$ and $\tilde{b}_2$, node $\tilde{1}$ and $\tilde{2}$ can obtain F_6 and F_5 respectively. In time 4, we repeat the same w.r.t. new symbols. As a result, node $\tilde{1}$ and $\tilde{2}$ obtain $(a_7\oplus\tilde{F}_4\oplus a_4, F_8)$ and $(b_8\oplus\tilde{F}_3\oplus b_3, F_7)$. In the last two time slots (time 3 and

4), node 1 and 2 get (F_5, F_7) and (F_6, F_8) respectively.

Stage 2 : During the next L time slots in the second stage, we accomplish the computation of desired functions that each node did not obtain yet. Recall that the transmission strategy in the perfect-feedback scenario is simply to forward all of the received signals at each node. And 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. 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 take a recently developed approach [1]: *Retrospective decoding.* The key feature of this approach is that the successive refinement is done in a retrospective manner and this leads us to resolve the issue mentioned above. The outline of the strategy is as follows: Node $\tilde{1}$ and $\tilde{2}$ start to decode (F_{4L-3}, F_{4L-1}) and (F_{4L-2}, F_{4L}) respectively. Here one key point to emphasize is that these decoded functions act as *side information.* And it turns out that this information enables the other-side nodes to obtain the desired functions w.r.t. the past symbols. Specifically the decoding order reads:

$$\begin{aligned}&(F_{4L-3}, F_{4L-2}, F_{4L-1}, F_{4L})\\ &\to (\tilde{F}_{4(L-1)-3}, \tilde{F}_{4(L-1)-2}, \tilde{F}_{4(L-1)-1}, \tilde{F}_{4(L-1)})\\ &\to \cdots \to (F_5, F_6, F_7, F_8) \to (\tilde{F}_1, \tilde{F}_2, \tilde{F}_3, \tilde{F}_4)\\ &\to (F_1, F_2, F_3, F_4)\,.\end{aligned}$$

With the refinement in time $2L + \ell$ ($\ell = 1, \cdots, L$), i.e., the ℓth time of Stage 2, node $\tilde{1}$ and $\tilde{2}$ can decode:

$$\begin{aligned}&\text{node } \tilde{1} : (F_{4(L-(\ell-1))-3}, F_{4(L-(\ell-1))-1}),\\ &\text{node } \tilde{2} : (F_{4(L-(\ell-1))-2}, F_{4(L-(\ell-1))}).\end{aligned}$$

Subsequently, node 1 and 2 decode:

$$\begin{aligned}&\text{node } 1 : (\tilde{F}_{4(L-\ell)-3}, \tilde{F}_{4(L-\ell)-1}),\\ &\text{node } 2 : (\tilde{F}_{4(L-\ell)-2}, \tilde{F}_{4(L-\ell)}).\end{aligned}$$

For ease of illustration, we elaborate how the decoding works in the case of $L = 2$. We exploit the received signals at time 3 ($= 2L - 1$) and 4 ($= 2L$) at node 1 and 2. As they get modulo-2 sums of forward symbols directly, the transmission strategy of node 1 and 2 at time 5 ($= 2L + 1$) is exactly the same as that in the perfect-feedback scheme: Forwarding (F_5, F_7) and (F_6, F_8) respectively. Then node $\tilde{1}$ 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.

Now in the backward channel, with the newly decoded F_5, F_2 (received at time 1) and $a_5 \oplus \tilde{F}_2 \oplus 2$ (received at time 3), node $\tilde{1}$ can construct:

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

This constructed signal is sent on the top level.

Furthermore, with the newly decoded F_7, (a_1, F_4, F_6) (received at time $1, 2$ and 3) and $a_7 \oplus \tilde{F}_4 \oplus 4$ (received at time 4), node $\tilde{1}$ can construct:

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

This is sent on 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, node 1 and 2 then get $(\tilde{F}_1 \oplus a_6 \oplus a_1, \tilde{F}_3 \oplus \tilde{F}_2 \oplus a_5)$ and $(\tilde{F}_2 \oplus b_5 \oplus b_2, \tilde{F}_4 \oplus \tilde{F}_1 \oplus b_6)$ respectively.

Observe that from the top, node 1 can finally extract $\tilde{F}_1$ of interest by using its own symbols (a_6, a_1). From the bottom, it can obtain $\tilde{F}_3$. From $\tilde{F}_3 \oplus \tilde{F}_2 \oplus a_8 \oplus a_3 \oplus a_5$ (received at time 5 on bottom), node 1 utilizes $\tilde{F}_2$ (received at time 1) and (a_8, a_3, a_5) to decode $\tilde{F}_3$. Similarly, node 2 can decode $(\tilde{F}_2, \tilde{F}_4)$.

With the help of the decoded functions, node 1 and 2 can then construct signals that can aid decoding the desired functions at the other-side nodes. Node 1 uses $\tilde{b}_1 \oplus F_1$ (received at time 1) and $\tilde{F}_1$ to generate $F_1 \oplus \tilde{a}_1$ on top; 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 bottom. 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, node $\tilde{1}$ and $\tilde{2}$ get $(F_1 \oplus \tilde{a}_1, F_3 \oplus \tilde{a}_3 \oplus \tilde{a}_2)$ and $(F_2 \oplus \tilde{b}_2, F_4 \oplus \tilde{b}_4 \oplus \tilde{b}_1)$ respectively. Here using their own symbols, node $\tilde{1}$ and $\tilde{2}$ can obtain (F_1, F_3) and (F_2, F_4).

Consequently, during 6 time slots, 8 modulo-2 sum functions w.r.t. forward symbols are computed, while 4 backward functions are computed. This gives $(R, \tilde{R}) = (\frac{4}{3}, \frac{2}{3})$. Actually one can easily extend this to an arbitrary L to show that $(R, \tilde{R}) = (\frac{4L}{3L}, \frac{4(L-1)}{3L}) = (\frac{4}{3}, \frac{4L-4}{3L})$ is achievable. Note that as $L \to \infty$, we get the desired rate pair, i.e., $(R, \tilde{R}) \to (\frac{4}{3}, \frac{4}{3}) = (C_{\mathsf{pf}}, \tilde{C}_{\mathsf{pf}})$.

Remark 2 (How to achieve the perfect-feedback bound?): As in the two-way interference channel [1], the key point in our achievability lies in exploiting the following three types of information as *side information:* (1) past received signals; (2) its own symbols; and (3) future decoded functions. Recall our achievability in Fig. 4 that the encoding strategy is to combine fresh symbols with past received signal, e.g., in time 1 node $\tilde{1}$ encodes $(\tilde{a}_2 \oplus F_2, \tilde{a}_1 \oplus a_1)$, which is the mixture of the fresh symbols $(\tilde{a}_2, \tilde{a}_1)$ and the received signals (F_2, a_1). And the decoding strategy is to utilize past received signals, e.g., in 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 [1] 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 in time 6. Here the signal is the summation of $\tilde{b}_1 \oplus F_1$ and $\tilde{F}_1$. And $\tilde{F}_1$ is actually the function that node 1 wishes to decode in the end; it can be viewed

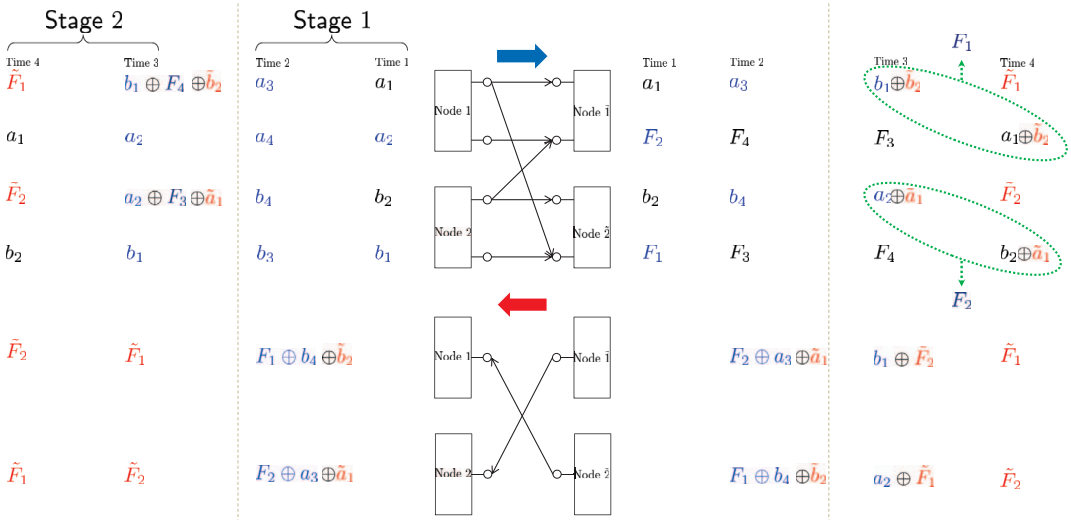

Fig. 5. An achievable scheme for $(m,n)=(1,2)$, $(\tilde{m},\tilde{n})=(1,0)$, and $(L,M)=(1,1)$.

as a *future* function since it is not available at the moment. So the approach is to defer the decoding procedure for F_1 until $\tilde{F}_1$ becomes available at node 1; see 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 node $\tilde{1}$ and $\tilde{2}$ similarly follows: Deferring the decoding of these functions until $\tilde{F}_2$ and $(\tilde{F}_1,\tilde{F}_3)$ becomes available at node 1 and 2 respectively. The decoding of (F_5,F_7) and (F_6,F_8) at node $\tilde{1}$ and $\tilde{2}$ precedes that of $(\tilde{F}_1,\tilde{F}_3)$ and $(\tilde{F}_2,\tilde{F}_4)$ at node 1 and 2 respectively. The idea of deferring the refinement together with the retrospective decoding plays a key role to achieve the perfect-feedback bound in the limit of L.

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

Similar to the previous example, we first review the perfect-feedback scheme, which we will use as a baseline for comparison to our achievable scheme. We focus only on the case of $(\tilde{m},\tilde{n})=(1,0)$, as the one for $(m,n)=(1,2)$ was already covered.

1) Perfect-feedback strategy: The perfect feedback scheme for $(\tilde{m},\tilde{n})=(1,0)$ consists of three time slots. In time 1, we send backward symbols $\tilde{a}_1$ and $\tilde{b}_2$ at node $\tilde{1}$ and $\tilde{2}$ respectively. Then node 1 and 2 get $\tilde{b}_2$ 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 get $\tilde{a}_1$ from node 2.

In time 2, with the feedback signals, node $\tilde{1}$ and $\tilde{2}$ can construct $\tilde{F}_2$ and $\tilde{F}_1$ respectively; and send them over the backward channel. Then node 1 and 2 get $\tilde{F}_1$ 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 get $\tilde{F}_2$ 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, node 1 and 2 obtain $(\tilde{F}_1,\tilde{F}_2)$ during three time slots. This gives a rate of $\frac{2}{3}$ $(=\tilde{C}_{\mathsf{pf}})$.

As mentioned in the previous example, in the two-way setting of our interest, a challenge arises due to the tension between feedback transmission and the traffic w.r.t. the other direction. Here is an achievable scheme that addresses this challenge.

2) Achievability: The underlying idea is similar to that for $(m,n)=(1,2)$, $(\tilde{m},\tilde{n})=(2,1)$: Exploiting effectively future information as side information via retrospective decoding. A key distinction relative to the prior example is that here the retrospective decoding occurs in a nested manner.

Our achievability introduces now the concept of multiple layers, say M layers. Each layer consists of two stages as in the previous example. Hence there are $2M$ stages in total. For each layer, the first stage consists of $2L$ time slots; and the second stage consists of $L+1$ time slots.

Until the end of $[M-(L-1)]$th layer, in the first stage, $4L$ and $2L$ of fresh symbols are transmitted through the forward and backward channels respectively. From $[M-(L-2)]$th layer to Mth layer, $4L$ fresh symbols are transmitted over the forward channel, while no fresh symbols are transmitted over the backward channel. The operation in the second stage for each layer is the same as that for the other layers: No fresh forward and backward symbols are transmitted.

We claim that the proposed scheme can achieve $(R,\tilde{R})=(\frac{4LM}{(3L+1)M},\frac{2L(M-2(L-1))}{(3L+1)M})$. Set $M=L^2$. As L tends to infinity, the rate pair becomes $(R,\tilde{R})=(\frac{4}{3},\frac{2}{3})=(C_{\mathsf{pf}},\tilde{C}_{\mathsf{pf}})$. Instead of describing schemes for arbitrary (L,M), we will illustrate our scheme for the single layer case, i.e., $(L,M)=(1,1)$ where part of the idea can be presented. In fact, from this single-layer case, one cannot see how the key distinctive idea of employing the multi-layer concept works. Actually the simplest multi-layer case is the one in which $(L,M)=(2,4)$ due to our setting $M=L^2$. As you can imagine, however, this case requires the use of $(3L+1)M$ $(=28)$ time slots, which makes presentation a bit challenging given the page-limit constraint. Hence, we leave more detailed and instructive explanations together with a multi-layer example in the full version [19]. In the case of $(L,M)=(1,1)$, the claimed achievable rate pair reads:

$(1, \frac{1}{2})$.

Stage 1 : We now illustrate the scheme for $(L, M) = (1, 1)$, which consists of a single layer, 2 stages, and 4 time slots. We noted that the first stage consists of 2 $(= 2L)$ time slots as we focus on the case of $L = 1$. In time 1, node 1 sends (a_1, a_2); node 2 sends (b_2, b_1). Then node $\tilde{1}$ gets (a_1, F_2). Similarly node $\tilde{2}$ gets (b_2, F_1). On the other hand, node $\tilde{1}$ and $\tilde{2}$ keep silent.

The transmission strategy in time 2 is pretty much the same as that in time 1. Node 1 and 2 transmit (a_3, a_4) and (b_4, b_3) so that node $\tilde{1}$ and $\tilde{2}$ obtain (a_3, F_4) and (b_4, F_3). Through the backward channel, node $\tilde{1}$ and $\tilde{2}$ send $\tilde{a}_1 \oplus F_2 \oplus a_3$ and $\tilde{b}_2 \oplus F_1 \oplus b_4$ respectively. Note that these signals contain each node's own symbol and received signals. See Fig. 5. Note that node 1 and 2 then get $\tilde{b}_2 \oplus F_1 \oplus b_4$ and $\tilde{a}_1 \oplus F_2 \oplus a_3$ respectively. Note that until the end of time 2, (F_2, F_4) and (F_1, F_3) are not delivered yet to node $\tilde{1}$ and $\tilde{2}$ respectively, while $(\tilde{F}_1, \tilde{F}_2)$ are missing at both node 1 and 2.

Stage 2 : The transmission strategy in the second stage is to accomplish the computation of desired functions that each node did not obtain yet. The stage consists of 2 time slots. In time 3, from the received symbol in time 2, node 1 cancels out its odd-index symbol a_1 and adds even-index symbol a_4, thus encoding $b_1 \oplus F_4 \oplus \tilde{b}_2$. In a similar manner, node 2 encodes $a_2 \oplus F_3 \oplus \tilde{a}_1$. The transmission strategy for each node is to forward the encoded signal on top. Furthermore, through the bottom level, each node forwards its own symbol whose index is the other transmitting node's symbol index canceled out. For example, node 2 forwards b_1, i.e., its own symbol of index 1, because node 1 cancels out a_1 at time 3. Similarly, node 1 forwards a_2. Node $\tilde{1}$ and $\tilde{2}$ then get $(b_1 \oplus \tilde{b}_2, F_3)$ and $(a_2 \oplus \tilde{a}_1, F_4)$ respectively.

From the received signal on top, node $\tilde{1}$ and $\tilde{2}$ can generate $b_1 \oplus \tilde{F}_2$ and $a_2 \oplus \tilde{F}_1$ respectively. Note that sending back them allows node 1 and 2 to obtain $\tilde{F}_1$ and $\tilde{F}_2$ respectively.

In time 4, through the top level, node 1 and 2 forward what they just computed: $\tilde{F}_1$ and $\tilde{F}_2$. Furthermore, node 1 and 2 forward a_1 and b_2 on bottom level. Node $\tilde{1}$ and $\tilde{2}$ then obtain $(\tilde{F}_1, a_1 \oplus \tilde{b}_2)$ and $(\tilde{F}_2, b_2 \oplus \tilde{a}_1)$. Observe that node $\tilde{1}$ can now obtain F_1 by adding $b_1 \oplus \tilde{b}_2$ (received on top in time 3) and $a_1 \oplus \tilde{b}_2$ (received on bottom at time 4). Similarly, node $\tilde{2}$ can obtain F_2.

Through the backward channel, node $\tilde{1}$ and $\tilde{2}$ simply feed back $\tilde{F}_1$ and $\tilde{F}_2$. Then node 1 and 2 get $\tilde{F}_2$ and $\tilde{F}_1$ respectively.

As a result, during 4 time slots, node $\tilde{1}$ and $\tilde{2}$ obtain $\{F_\ell\}_{\ell=1}^4$ while node 1 and 2 obtain $(\tilde{F}_1, \tilde{F}_2)$. This gives $(1, \frac{1}{2})$. As mentioned earlier, we find that this idea can be extended to arbitrary values of (L, M) by employing the retrospective decoding strategy over multi-layers, which yields: $(R, \tilde{R}) \to (\frac{4}{3}, \frac{2}{3}) = (C_{\mathsf{pf}}, \tilde{C}_{\mathsf{pf}})$. See the full version for details [19].

C. Generalization

We now prove the achievability for general $(m, n), (\tilde{m}, \tilde{n})$. Our proof is categorized into three regimes.

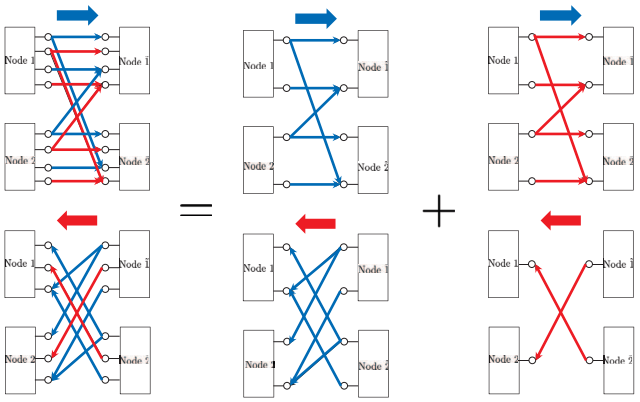

Fig. 6. A network decomposition example of $(m, n) = (2, 4), (\tilde{m}, \tilde{n}) = (3, 1)$ model. We see that the decomposition is given by $(2, 4), (3, 1) \longrightarrow (1, 2), (2, 1) \times (1, 2), (1, 0)$.

1) Regimes in which interaction provides no gain: See a non-interactive scheme [18]. The channel regimes of this category are: $0 \le \alpha \le \frac{2}{3}, 0 \le \tilde{\alpha} \le \frac{2}{3}$; $(\frac{2}{3} \le \alpha < 1, 1 < \alpha \le \frac{3}{2})$, $(\frac{2}{3} \le \tilde{\alpha} < 1,\ 1 < \tilde{\alpha} \le \frac{3}{2})$; and $\alpha \ge \frac{3}{2}, \tilde{\alpha} \ge \frac{3}{2}$.

2) Regimes in which interaction helps only either in forward or backward direction: It turns out the achievability in this case is a simple combination of the non-interactive scheme [18] and an interactive scheme in [16]. While it requires a tedious calculation, it contains no new ingredient, hence, we omit the detailed proof. The channel regimes of this category are: $(0 \le \alpha < \frac{2}{3}, \alpha > \frac{3}{2}), (\frac{2}{3} < \tilde{\alpha} < 1, 1 < \tilde{\alpha} < \frac{3}{2})$; and $(\frac{2}{3} < \alpha < 1, 1 < \alpha < \frac{3}{2}), (0 \le \tilde{\alpha} < \frac{2}{3}, \tilde{\alpha} > \frac{3}{2})$.

3) Regimes in which interaction helps both in forward and backward directions: As mentioned earlier, the key idea is to employ the retrospective decoding (potentially employing the concept of multiple layers). For ease of generalization to arbitrary channel parameters in the regime, here we employ network decomposition in [18] where an original network is decomposed into elementary orthogonal subnetworks and achievable schemes are applied separately into the subnetworks. See Fig. 6 for an example of such network decomposition. The idea is to use graph coloring. The figure in Fig. 6 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: $(m, n) = (1, 2), (\tilde{m}, \tilde{n}) = (2, 1)$ model (blue color); and $(m, n) = (1, 2), (\tilde{m}, \tilde{n}) = (1, 0)$ model (red color). Note that the original network is simply the concatenation of these two subnetworks. We denote the decomposition as $(2, 4), (3, 1) \longrightarrow (1, 2), (2, 1) \times (1, 2), (1, 0)$.

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 $(R, \tilde{R}) = (\frac{4LM}{(3L+1)M}, \frac{2L(M-2(L-1))}{(3L+1)M})$. Setting $M = L^2$ and letting $L \to \infty$, 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 rate region: $\{(R, \tilde{R}) : R \leq C_{\mathsf{pf}} = \frac{8}{3}, \tilde{R} \leq \tilde{C}_{\mathsf{pf}} = 2\}$.□

We find that this idea can be extended to arbitrary values of $(m, n), (\tilde{m}, \tilde{n})$. The detailed proof is in the full version [19].

V. 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 not only interaction offers a net increase in capacity, it leads us to get all the way to perfect-feedback computation capacities simultaneously in both directions. In view of [1], this result is another instance in which interaction provides a huge gain. One future work of interest would be to explore a variety of network communication/computation scenarios in which the similar phenomenon occurs.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2014-0-00282, Development of 5G Mobile Communication Technologies for Hyper-connected smart services) and (No. 2017-0-00694, Coding for High-Speed Distributed Networks).

References

[1] C. Suh, J. Cho, and D. Tse, "Two-way interference channel capacity: How to have the cake and eat it too," in *Proc. IEEE International Symposium on Information Theory*, July 2017.

[2] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley Symp. Math, statics and Prob.*, 1961, pp. 611–644.

[3] ——, "The zero error capacity of a noisy channel," *IRE Transactions on Information Theory*, vol. 2, pp. 8–19, Sept. 1956.

[4] T. M. Cover and S. Pombra, "Gaussian feedback capacity," *IEEE Transactions on Information Theory*, vol. 35, pp. 37–43, Jan. 1989.

[5] Y.-H. Kim, "Feedback capacity of the first-order moving average Gaussian channel," *IEEE Transactions on Information Theory*, vol. 52, pp. 3063–3079, July 2006.

[6] N. T. Gaarder and J. K. Wolf, "The capacity region of a multiple-access discrete memoryless channel can increase with feedback," *IEEE Transactions on Information Theory*, Jan. 1975.

[7] L. H. Ozarow, "The capacity of the white Gaussian multiple access channel with feedback," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 623–629, July 1984.

[8] L. H. Ozarow and S. K. Leung-Yan-Cheong, "An achievable region and outer bound for the Gaussian broadcast channel with feedback," *IEEE Transactions on Information Theory*, vol. 30, pp. 667–671, 1984.

[9] C. Suh and D. Tse, "Feedback capacity of the Gaussian interference channel to within 2 bits," *IEEE Transactions on Information Theory*, vol. 57, pp. 2667–2685, May 2011.

[10] C. Suh, I.-H. Wang, and D. Tse, "Two-way interference channels," *Proc. IEEE International Symposium on Information Theory*, July 2012.

[11] C. Suh, D. Tse, and J. Cho, "To feedback of not to feedback," *Proc. IEEE International Symposium on Information Theory*, July 2016.

[12] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 755–764, Apr. 2005.

[13] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, pp. 4539–4551, Sept. 2010.

[14] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, pp. 476–489, Mar. 2011.

[15] C. Suh and M. Gastpar, "Interactive function computation," *Proc. IEEE International Symposium on Information Theory*, July 2013.

[16] S. Shin and C. Suh, "Two-way function computation," in *Proc. 52th Annu. Allerton Conf. Communication, Control, and Computing*, Oct. 2014.

[17] A. S. Avestimehr, S. N. Diggavi, and D. Tse, "Wireless network information flow: A deterministic approach," *IEEE Transactions on Information Theory*, vol. 57, pp. 1872–1905, Apr. 2011.

[18] C. Suh, N. Goela, and M. Gastpar, "Computation in multicast networks: Function alignment and converse theorems," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1866–1877, Feb. 2016.

[19] S. Shin and C. Suh, "Capacity of a two-way function multicast channel," [Online]. Available: https://sites.google.com/site/seiyunshin/Capacity.of.a.Two-way.Multicast.Channel.Seiyun.pdf.