Optimal Learning Rate of Sending One Bit Over Arbitrary Acyclic BISO-Channel Networks

Chih-Chun Wang and David J. Love
Purdue ECE, Email: {chihw,djlove}@purdue.edu

***Abstract*—This work considers the problem of sending a 1-bit message over an acyclic network, where the "edge" connecting any two nodes is a memoryless binary-input/symmetric-output (BISO) channel. For any arbitrary acyclic network topology and constituent channel models, a min-cut-based converse of the learning rate, denoted by $\boldsymbol{r}^*$, is derived. It is then shown that for any $\boldsymbol{r} < \boldsymbol{r}^*$, one can design a scheme with learning rate $\boldsymbol{r}$. Capable of approaching the optimal $\boldsymbol{r}^*$, the proposed scheme is thus the *asymptotically fastest* for sending one bit over any acyclic BISO-channel network. The construction is based on a new concept of *Lossless Amplify-&-Forward*, a sharp departure from existing multi-hop communication scheme designs.**

I. Problem Formulation

We model a network by a finite directed acyclic graph (V, E). A source $s \in V$ likes to communicate a 1-bit message $\Theta \in \{0, 1\}$ to a destination $d \in V$. The "channel" along each edge $e = (u, v)$ is a memoryless *point-to-point* binary-input/symmetric-output (BISO) channel [1], [2]. There is no other restriction in our model, e.g., the network may be a line network [3]–[7] or other configurations [8]–[11]. The channel may be a Binary Symmetric Channel (BSC) for one edge [2] and be a binary-input additive white Gaussian channel for the other edge(s). While the input is always binary, the output of the channel can be a high dimensional vector if desired.

For each discrete time $t \in [1, \infty)$, source s transmits a bit

$$X_{(s,w)}(t) = f_{t,(s,w)}(\Theta) \tag{1}$$

to its downstream neighbor w based on the message Θ. Any node $v \neq s$ transmits a bit to its downstream neighbor w:

$$X_{(v,w)}(t) = f_{t,(v,w)}(\{Y_{(u,v)}(\tau) : \forall \tau < t, u\}) \tag{2}$$

based on $Y_{(u,v)}(\tau)$, the signal received at v from node u at time $\tau < t$. For any $e = (u, v)$, the distribution of $Y_e(t)$ given $X_e(t)$ follows the specified BISO channel models. Destination d then finds the Maximum Likelihood (ML) detector of Θ by

$$\hat{\Theta}(t) = \arg\max_{\theta \in \{0,1\}} \mathbb{P}\Big(\{Y_{(u,d)}(\tau) : \forall \tau \leq t, u\}\Big| \Theta = \theta\Big) \tag{3}$$

Define the error probability (of destination d) at time t by

$$p_\epsilon(t) \triangleq \max_{\theta=0,1} \mathbb{P}(\hat{\Theta}(t) \neq \theta | \Theta = \theta) \tag{4}$$

The achievable *learning rate* of a scheme $\{f_{t,e} : \forall t, e\}$ is

$$\boldsymbol{r} \triangleq \liminf_{\Delta \to \infty} \frac{-\log(p_\epsilon(\Delta))}{\Delta}. \tag{5}$$

The *optimal learning rate* $\boldsymbol{r}^*$ is the supremum of the achievable learning rates $\boldsymbol{r}$ of all possible schemes $\{f_{t,e} : \forall t, e\}$.

Denote the delay needed to achieve a target error probability p_ϵ by $\Delta(p_\epsilon)$. By (5), we have

$$\Delta(p_\epsilon) = \frac{-\log(p_\epsilon)}{\boldsymbol{r}} + o(-\log(p_\epsilon)) \quad \text{when } p_\epsilon \to 0, \tag{6}$$

As a result, we say any scheme that achieves or approaches the largest $\boldsymbol{r}^*$ is the *asymptotically fastest* (when $p_\epsilon \to 0$). With the goal of designing ultra-low-latency multi-hop communication schemes for delay sensitive applications, e.g., remote surgery, this work characterizes the $\boldsymbol{r}^*$ value.

A. Existing Results

If the network is a single hop from s to d, it is known that the optimal learning rate of a BISO channel $e = (s, d)$ is

$$\boldsymbol{r}_e^* = -\log\left(\sum_{y \in \mathcal{Y}_e} \sqrt{\mathbb{P}_{Y_e|X_e}(y|0) \cdot \mathbb{P}_{Y_e|X_e}(y|1)}\right) \tag{7}$$

if the output alphabet $\mathcal{Y}_e$ is discrete [12]. The formula can be readily generalized for continuous alphabet $\mathcal{Y}_e$ as well.

If the network is a 2-hop line network $s \to r \to d$ and both channels are discrete, [4] showed that for any

$$\boldsymbol{r} < \min\Big(\boldsymbol{r}^*_{(s,r)}, \boldsymbol{r}^*_{(r,d)}\Big), \tag{8}$$

where $\boldsymbol{r}_e^*$ is the optimal learning rate over a single hop e as defined in (7), one can design an achievability scheme of learning rate $\boldsymbol{r}$. The results [4] were a major breakthrough after several earlier attempts of designing (suboptimal) achievability schemes [5], [7].

This work generalizes the characterization of $\boldsymbol{r}^*$ from 2-hop networks [4] to arbitrary acyclic multi-hop networks, a non-trivial extension that requires many new analysis tools and innovations. In addition, our result is based on a new concept called Lossless Amplify-&-Forward (AF). *The philosophy of lossless AF is to effectively eliminate error accumulation when concatenating two channels via AF, which allows us to harvest the ultra-low-latency benefits of AF without its fatal drawback* and is fundamentally different from existing multi-hop schemes like decode-&-forward, compress-&-forward [13], [14], quantize-&-forward [15], compute-&-forward [16].

II. The Converse

Proposition 1: The achievable learning rate $\boldsymbol{r}$ of any scheme over an acyclic network must satisfy

$$\boldsymbol{r} \leq \boldsymbol{r}^* \triangleq \min_{\mathsf{cut}(s,d)} \sum_{e \in \mathsf{cut}(s,d)} \boldsymbol{r}_e^* \tag{9}$$

where the minimization is over all *edge cuts* $\text{cut}(s, d)$ separating s and d, and $\boldsymbol{r}_e^*$ was defined in (7).

The proof of Proposition 1 is a standard reduction-based *cutset bound* argument. The detailed proof is thus omitted.

III. Achievability For $\boldsymbol{r}^*$-uniform Line Networks

The greatest challenge of designing an achievability scheme is to devise a systematic construction that is applicable to *any* acyclic network topology while still admitting provable performance. To that end, we first design an achievability scheme for any *L-hop line network* with edges e_1 to e_L satisfying

$$\boldsymbol{r}_{e_1}^* = \boldsymbol{r}_{e_2}^* = \cdots = \boldsymbol{r}_{e_L}^* = \boldsymbol{r}^* \tag{10}$$

and show that the scheme can achieve learning rate $\boldsymbol{r}$ for any $\boldsymbol{r} < \boldsymbol{r}^*$. We call such a network $\boldsymbol{r}^*$-uniform L-hop line network. Note that we require the same $\boldsymbol{r}_e^*$ for each hop e but the individual channel models can still be different. In Sec. IV, we discuss how the "modularity" of our construction can be used to achieve the converse (9) for general acyclic networks.

A. Construction of Abstract Binary-Input/$\overline{\gamma}$-Output Channels

We denote any BISO channel with distribution $\mathbb{P}_{Y_e|X_e}$ by CH_e. For any integer $\overline{\gamma} \geq 1$, we first discuss how we can convert m uses of CH_e to a binary-input/$\overline{\gamma}$-output (BI-$\overline{\gamma}$-out) abstract channel. This "abstract channel" is a key building block of our scheme.

To that end, we define a "$\mathcal{Y}_e^m$-to-$\overline{\gamma}$-ary-output quantizer" (or just "$\overline{\gamma}$-ary quantizer" as shorthand) $\boldsymbol{\pi}_m$, which is a mapping[1] from $\mathcal{Y}_e^m$ to $\{0, 1, \cdots, \overline{\gamma}-1\}$. The construction is as follows. For any given $\boldsymbol{\pi}_m$, suppose the abstract channel receives an input bit b. We first repeat the input bit b by m times, and send the resulting $\vec{0}$ or $\vec{1}$ vector over m uses of CH_e. After receiving the m outputs $\vec{y} \in \mathcal{Y}_e^m$, we pass it through the $\overline{\gamma}$-ary quantizer and output $\Gamma = \boldsymbol{\pi}_m(\vec{y})$. The construction is complete. It is worth noting that regardless of the m value, the abstract channel is always of binary input and $\overline{\gamma}$-ary output.

Definition 1: A quantizer $\boldsymbol{\pi}_m$ is *symmetric* if the resulting BI-$\overline{\gamma}$-out channel is of binary-input symmetric-output (BISO).

B. LLR of the New Abstract Channels

Suppose the output of the abstract channel is $\Gamma = \gamma$, we compute the LLR of the abstract (abs) channel by

$$L_{\text{abs}}^{(m)} \triangleq \log\left(\frac{\mathbb{P}(\Gamma=\gamma|b=0)}{\mathbb{P}(\Gamma=\gamma|b=1)}\right) \tag{11}$$

$$= \log\left(\frac{\mathbb{P}_{Y_e^m|X_e^m}\left(\boldsymbol{\pi}_m(\vec{Y})=\gamma\middle|\vec{0}\right)}{\mathbb{P}_{Y_e^m|X_e^m}\left(\boldsymbol{\pi}_m(\vec{Y})=\gamma\middle|\vec{1}\right)}\right) \tag{12}$$

where (12) is due to the construction of the abstract channel. When evaluating (12), we define $\log(p/0) = \infty$ and $\log(0/p) = -\infty$ for any $p > 0$. Since the event $L_{\text{abs}}^{(m)} = \log(0/0)$ is of probability zero, it is ignored/excluded from our discussion.

Definition 2: For any given $\boldsymbol{\pi}_m$, we define the sample space of the LLR $L_{\text{abs}}^{(m)}$ by $\mathcal{L}_m$. Since there are exactly $\overline{\gamma}$ possible output values $\Gamma \in \{0, \cdots, \overline{\gamma}-1\}$, we can assume that $\mathcal{L}_m$ is discrete and contains *exactly* $\overline{\gamma}$ *distinct elements* without loss of generality.[2] We denote $\mathcal{L}_m$ by

$$\mathcal{L}_m = \{\ell_\gamma : \gamma \in \{0, 1, \cdots, \overline{\gamma}-1\}\} \tag{13}$$

where ℓ_γ is the LLR value in (11) if $\Gamma = \gamma$. Without loss of generality, we also assume the elements of $\mathcal{L}_m$ satisfying

$$\ell_0 > \ell_1 > \cdots > \ell_{\overline{\gamma}-1} \tag{14}$$

by relabeling the γ indices according to the order of ℓ_γ.

Many stochastic properties of a binary-input channel can be characterized by its LLR distributions. For our BI-$\overline{\gamma}$-out abstract channel, the probability mass function (pmf) of $\mathbb{P}(L_{\text{abs}}^{(m)} = \ell_\gamma|b)$, conditioning on the input $b \in \{0, 1\}$, can be computed by finding the value of $\mathbb{P}_{Y_e^m|X_e^m}(\vec{Y} \in \boldsymbol{\pi}_m^{-1}(\gamma)|\vec{b})$, see (12). Assuming $\boldsymbol{\pi}_m$ is symmetric, we introduce an alternative way of describing[3] the conditional pmf of $L_{\text{abs}}^{(m)}$.

Definition 3: For any symmetric $\boldsymbol{\pi}_m$, define the *LLR Rate Function* (LLR.rf) of the abstract channel as

$$\mathsf{LLR.rf}_m(\rho) \triangleq \frac{-\log\Big(\mathbb{P}(L_{\text{abs}}^{(m)} = \ell|b=1)\Big)}{m}$$
$$\text{if } \rho = \frac{\ell}{m} \text{ for some } \ell \in \mathcal{L}_m \tag{15}$$

and we say $\mathsf{LLR.rf}_m(\rho)$ is undefined for other ρ values.

The intuition behind the LLR.rf definition is that since we are interested in the learning rate, we take $-\log(\cdot)$ of the pmf values. Because each abstract channel consists of m uses of CH_e, we normalize both the LLR and the $-\log(\text{pmf})$ values by $1/m$ to quantify *the average impact of each* CH_e *usage.*

Because both $-\log(\cdot)$ and the normalization $1/m$ are bijective mappings, knowing the $\mathsf{LLR.rf}_m(\cdot)$ function is equivalent to knowing the conditional pmf of $L_{\text{abs}}^{(m)}$ given $b = 1$.

C. Analysis of the New Abstract Channels

With all the definitions of $L_{\text{abs}}^{(m)}$ and $\mathsf{LLR.rf}_m(\rho)$, we now analyze the properties of this BI-$\overline{\gamma}$-out abstract channel. Firstly, we quantify the *learning rate*, see (5), of using this abstract channel (repeatedly) to send a single-bit message:

Lemma 1: For any symmetric $\overline{\gamma}$-ary quantizer $\boldsymbol{\pi}_m$, the m-normalized learning rate of the resulting abstract channel is

$$\bar{\boldsymbol{r}}_m = \frac{-\log\Big(\sum_{\gamma=0}^{\overline{\gamma}-1}\sqrt{\mathbb{P}(\Gamma=\gamma|b=0)\mathbb{P}(\Gamma=\gamma|b=1)}\Big)}{m} \tag{16}$$

[1]The mapping can be either deterministic or randomized. The most rigorous definition is to treat $\boldsymbol{\pi}_m$ as a "channel" from $\mathcal{Y}_e^m$ to $\{0, 1, \cdots, \overline{\gamma}-1\}$. However, because the purpose of $\boldsymbol{\pi}_m$ is mainly to quantize $\vec{Y} \in \mathcal{Y}_e^m$, we describe it herein as a deterministic mapping for the ease of notation.

[2]Sometimes $\mathcal{L}_m$ contains strictly less than $\overline{\gamma}$ elements if $L_{\text{abs}}^{(m)} = \log(0/0)$ for some $\Gamma = \gamma$. If two $\gamma_1 \neq \gamma_2$ result in the same $L_{\text{abs}}^{(m)}$ value, the number of *distinct* elements of $\mathcal{L}_m$ also decrements. Our results still hold for those corner cases after some detailed and straightforward case discussion.

[3]With symmetric $\boldsymbol{\pi}_m$, the resulting abstract channel is BISO. Therefore, Definition 3 focuses exclusively on the conditional distribution given $b = 1$. The conditional distribution given $b = 0$ can be obtained/defined by symmetry.

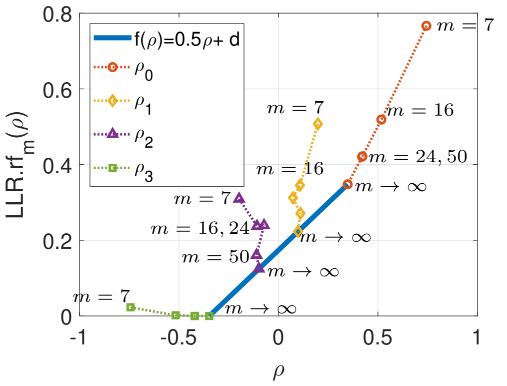

Fig. 1. Trajectories of $\mathsf{LLR.rf}_m(\rho)$ for $m = 7$, 16, 24, 50, and ∞, respectively. When $m = 7$, the four points spread the widest. As m increases, the trajectories move inward and eventually converge to the line $f(\rho) = 0.5\rho + 0.174$.

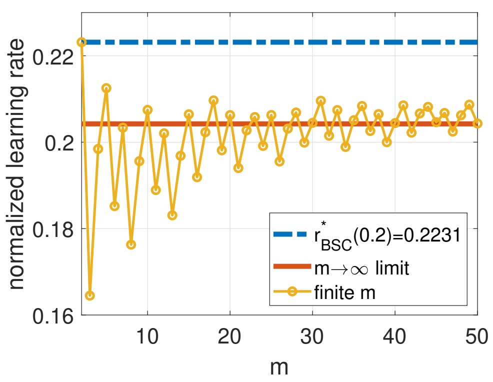

Fig. 2. The trajectory of the m-normalized learning rate $\bar{\boldsymbol{r}}_m$; and the upper bound $\boldsymbol{r}^*_{\text{BSC}}(0.2)$.

of which the numerator is a straightforward application of (7). Since our abstract channel consists of m uses of CH_e, we add the $1/m$ term to quantify the average impact of each CH_e.

Definition 4: For any fixed $d > 0$ and $\overline{\gamma}$ values, we say a class of symmetric $\overline{\gamma}$-ary quantizers $\{\boldsymbol{\pi}_m : m \in \{1, 2, \cdots\}\}$ is *asymptotically d-linear*, if all $\overline{\gamma}$ points of the limiting function $\lim_{m\to\infty} \mathsf{LLR.rf}_m(\rho)$ fall onto the line $f(\rho) = 0.5\rho + d$.

For example, suppose the constituent channel CH_e is a BSC with cross probability $p = 0.2$. We have numerically found a class of symmetric 4-ary quantizers $\{\boldsymbol{\pi}_m : m \in \mathbb{N}\}$ that is asymptotically d-linear with $d = 0.174$. Fig. 1 plots the trajectories of the $\overline{\gamma} = 4$ points of $\mathsf{LLR.rf}_m(\rho)$ for various m. As can be seen, all 4 points of $\mathsf{LLR.rf}_m(\rho)$ eventually converge to the same line $f(\rho) = 0.5\rho + 0.174$.

We now present our first main result of achievability.

Proposition 2: For any $d > 0$, any $\overline{\gamma} \geq 2$, and any class of symmetric $\overline{\gamma}$-ary quantizers $\{\boldsymbol{\pi}_m\}$ that is asymptotically d-linear, the m-normalized learning rate $\bar{\boldsymbol{r}}_m$ in (16) satisfies

$$\lim_{m\to\infty} \bar{\boldsymbol{r}}_m = d. \tag{17}$$

Continue our example of using BSC(0.2) as the constituent channel. We have numerically found a symmetric $\overline{\gamma} = 8$-ary quantizer class $\{\boldsymbol{\pi}_m : m\}$ that is asymptotically d-linear with $d = 0.204$. Fig. 2 plots the corresponding $\bar{\boldsymbol{r}}_m$ (numerically computed by (16)) for $m = 2$ to 50. As predicted, the m-normalized learning rate converges to $d = 0.204$ as $m \to \infty$.

By the data processing inequality, the m-normalized learning rate $\bar{\boldsymbol{r}}_m$ after $\overline{\gamma}$-ary quantization $\boldsymbol{\pi}_m$ must be less than the native, before-quantization learning rate of BSC, denoted by $\boldsymbol{r}^*_{\text{BSC}}(0.2) = 0.223$ as computed by (7). That said, the simple 3-bit quantization $\gamma \in \{0, \cdots, 7\}$ of the $\boldsymbol{\pi}_m$ used in Fig. 2 already achieves 92% of the upper bound, since $d = 0.204 = 0.92\boldsymbol{r}^*_{\text{BSC}}(0.2)$. In fact, we have numerically found a $\overline{\gamma} = 16$-ary symmetric quantizer class $\{\boldsymbol{\pi}_m : m\}$ that is asymptotically d-linear with $d = 0.97\boldsymbol{r}^*_{\text{BSC}}(0.2)$. This shows that *the price one pays for compressing an unboundedly large m-dimensional observation $\vec{Y} \in \mathcal{Y}_e^m$ into a mere 4-bit Γ value is only 3% of the normalized learning rate, even if we let $m \to \infty$!*

The introduction of $\{\boldsymbol{\pi}_m\}$ greatly simplifies the analysis since the resulting abstract channel is BI-$\overline{\gamma}$-out (thus having fixed input and output alphabets) regardless of m. Furthermore, the conversion is *almost lossless* in the following sense.

Proposition 3: Consider any CH_e with single-hop learning rate $\boldsymbol{r}^*_e$. For any $d < \boldsymbol{r}^*_e$ that can be arbitrarily close to $\boldsymbol{r}^*_e$, there exist a $\overline{\gamma}$ value and a symmetric $\overline{\gamma}$-ary quantizer class $\{\boldsymbol{\pi}_m : m\}$ that is asymptotically d-linear.

Proposition 2 is proven by plugging (15) into (16) and using Definition 4. Proposition 3 is proven by first choosing a sufficiently large $\overline{\gamma}$ for the given $d < \boldsymbol{r}^*_e$ and then explicitly devising either a deterministic or a randomized mapping $\boldsymbol{\pi}_m$. The detailed proofs are omitted due to space constraints.

D. The New Concept of Lossless Amplify-&-Forward

We now describe how one can serially concatenate two abstract channels in an almost lossless fashion. Consider two arbitrary physical channels CH_{e_1} and CH_{e_2}, and the corresponding two symmetric quantizer classes $\{\boldsymbol{\pi}^{[1]}_m\}$ and $\{\boldsymbol{\pi}^{[2]}_m\}$, respectively. We assume that both $\{\boldsymbol{\pi}^{[1]}_m\}$ and $\{\boldsymbol{\pi}^{[2]}_m\}$ are asymptotically d_0-linear with the same d_0, and the first quantizer class is $\overline{\gamma}_1$-ary and the second quantizer class is $\overline{\gamma}_2$-ary, where $\overline{\gamma}_1$ and $\overline{\gamma}_2$ may be different.

Since $\{\boldsymbol{\pi}^{[1]}_m : m\}$ is $\overline{\gamma}_1$-ary, its $\mathsf{LLR.rf}^{\text{ch1}}_m(\cdot)$ has $\overline{\gamma}_1$ "valid points" and we denote their $m \to \infty$ limits by

$$(x_0, y_0), (x_1, y_1), \cdots, (x_{\overline{\gamma}_1-1}, y_{\overline{\gamma}_1-1}). \tag{18}$$

where (x_γ, y_γ) is the x- and y-coordinates of the limiting point corresponding to the event $\{\Gamma = \gamma\}$. For example, the trajectories of $\mathsf{LLR.rf}_m(\cdot)$ in Fig. 1 converge to four different points when $m \to \infty$, and these four limiting points are denoted by (18).

By (14), $\Gamma = 0$ corresponds to the largest LLR $L^{(m)}_{\text{abs}}$ and thus the largest x-coordinate. This implies

$$x_0 > x_1 > \cdots > x_{\overline{\gamma}_1-1} \tag{19}$$

By Definition 4, we have $y_\gamma = 0.5x_\gamma + d_0$, which then implies

$$y_0 > y_1 > \cdots > y_{\overline{\gamma}_1-1} \tag{20}$$

Corollary 1: Without loss of generality, we may further assume

$$(x_0, y_0) = (2d_0, 2d_0) \text{ and } (x_{\overline{\gamma}_1-1}, y_{\overline{\gamma}_1-1}) = (-2d_0, 0).$$

Corollary 1 is derived from the explicit construction of $\{\boldsymbol{\pi}_m\}$ that is used to prove Proposition 3.

For example, the four limiting points in Fig. 1 are

$$\begin{aligned}(x_0, y_0) &= (0.348, 0.348), \quad (x_1, y_1) = (0.0986, 0.223)\\ (x_2, y_2) &= (-0.0986, 0.125), \ (x_3, y_3) = (-0.348, 0)\end{aligned} \tag{21}$$

and they satisfy Corollary 1 since $d = 0.174$.

We now describe our construction. Suppose the first abstract channel uses CH_{e_1} for m times and outputs $\Gamma = \gamma$. We then construct a bit string $b_1 b_2 \cdots b_{\overline{\gamma}_1-1}$ satisfying

$$b_i = \begin{cases} 0 & \text{if } i < \overline{\gamma}_1 - \gamma \text{ and } i \in \{1, \cdots, \overline{\gamma}_1 - 1\} \\ 1 & \text{if } i \geq \overline{\gamma}_1 - \gamma \text{ and } i \in \{1, \cdots, \overline{\gamma}_1 - 1\} \end{cases} \tag{22}$$

Recall that we have been given a class of symmetric $\overline{\gamma}_2$-ary quantizers $\{\boldsymbol{\pi}_m^{[2]} : m\}$ for the physical channel CH_{e_2}. We then send bit b_i over an abstract channel created by m_i uses of CH_{e_2} using the symmetric quantizer $\boldsymbol{\pi}_{m_i}^{[2]}$, where

$$m_i \triangleq \left\lfloor \frac{m \cdot (y_{i-1} - y_i)}{y_0} \right\rfloor \tag{23}$$

Namely, we "split" a single abstract channel that uses CH_{e_2} for m times to $(\overline{\gamma}_1 - 1)$ abstract (sub-)channels, each uses CH_{e_2} for m_i times to send a single bit b_i. This is possible since $\sum_{i=1}^{\overline{\gamma}_1-1} m_i \leq m$ by (23). This concludes our concatenation of m uses of physical channel CH_{e_1} and $\sum m_i$ uses of physical channel CH_{e_2} to create an end-to-end (e2e) new abstract channel.

Three observations are in order. Firstly, the new e2e abstract channel is of binary input since the m uses of CH_{e_1} create a *binary-input* abstract channel. The output of the e2e abstract channel is of order $(\overline{\gamma}_2)^{\overline{\gamma}_1-1}$. The reason is that sending each b_i through a BI-$\overline{\gamma}_2$-out abstract channel (that uses CH_{e_2} for m_i times) will have an output $\Gamma_i \in \{0, \cdots, \overline{\gamma}_2 - 1\}$. We have $(\overline{\gamma}_1 - 1)$ bits b_i and the overall output is thus a $(\overline{\gamma}_1 - 1)$-dimensional vector in $\{0, \cdots, \overline{\gamma}_2 - 1\}^{\overline{\gamma}_1-1}$. Secondly, it is possible that $m > \sum m_i$. Note that the m uses of CH_{e_2} can be viewed as a *budget* of how many times we are allowed to use the physical channel CH_{e_2}. If $m > \sum m_i$, then we do not use CH_{e_2} to its full extent and simply discard/ignore the excess $m - \sum m_i$ channel uses. Thirdly, we have

Lemma 2: The new e2e abstract channel is BISO.

The proof of this lemma is done by verifying the symmetry between sending $b = 0$ versus sending $b = 1$.

To demonstrate our construction, suppose $\overline{\gamma}_1 = 4$ and the four limiting points of $\{\boldsymbol{\pi}_m^{[1]} : m\}$ are as described in (21). For any m, say $m = 1000$, we use the symmetric quantizer $\boldsymbol{\pi}_{1000}^{[1]}$ to create a BI-4-out abstract channel from 1000 uses of CH_{e_1}. Suppose the output of the abstract channel is $\Gamma = 1$, then we construct a 3-bit string $b_1 b_2 b_3 = 001$ according to (22). Bit $b_1 = 0$ will be sent through an abstract channel that uses CH_{e_2} for $m_1 = \lfloor 1000 \cdot (0.348 - 0.223)/0.348 \rfloor = 359$ times based on $\boldsymbol{\pi}_{359}^{[2]}$. Bit $b_2 = 0$ will be sent by using CH_{e_2} for $m_2 = \lfloor 1000 \cdot (0.223 - 0.125)/0.348 \rfloor = 281$ times based on $\boldsymbol{\pi}_{281}^{[2]}$. Bit $b_3 = 1$ will be sent by using CH_{e_2} for $m_3 = \lfloor 1000 \cdot (0.125 - 0)/0.348 \rfloor = 359$ times based on $\boldsymbol{\pi}_{359}^{[2]}$.

Suppose both symmetric quantizer classes $\{\boldsymbol{\pi}_m^{[1]} : m\}$ and $\{\boldsymbol{\pi}_m^{[2]} : m\}$ are asymptotically d_0-linear. We compute, using[4] (11)–(15), the $\mathsf{LLR.rf}_m^{\mathrm{e2e}}(\rho)$ function of our new e2e abstract channel. We then have:

Proposition 4: When $m \to \infty$, all $(\overline{\gamma}_2)^{\overline{\gamma}_1-1}$ limiting points of $\mathsf{LLR.rf}_m^{\mathrm{e2e}}(\rho)$ will fall onto the same line $f(\rho) = 0.5\rho + d_0$ as the limiting points of $\mathsf{LLR.rf}_m^{\mathrm{ch1}}(\rho)$ and $\mathsf{LLR.rf}_m^{\mathrm{ch2}}(\rho)$ of the individual channels. Herein we slightly abuse the notation and say the e2e abstract channel is also asymptotically d_0-linear.

We call this new design *lossless Amplify-&-Forward* (lossless AF). The reason is that the output of abstract channel 1 is converted to a bit string $b_1 \cdots b_{\overline{\gamma}_1-1}$ by (22) in a *reversible* fashion. The relay then sends each b_i directly over a split version of abstract channel 2 with zero additional processing. Therefore, it is along the spirit of AF, for which the received signal from the previous hop is directly used as the input to the next hop without any active processing that cleans up the noise (such as decode-&-forward) or judiciously compresses[5] the observations (such as compress-&-forward, quantize-&-forward, compute-&-forward).

The reason that our new design is called lossless is that Propositions 2 and 4 jointly imply:

$$\lim_{m\to\infty} \bar{\boldsymbol{r}}_m^{[\mathrm{e2e}]} = d_0 = \lim_{m\to\infty} \bar{\boldsymbol{r}}_m^{[\mathrm{ch1}]} = \lim_{m\to\infty} \bar{\boldsymbol{r}}_m^{[\mathrm{ch2}]}. \tag{24}$$

Therefore, our AF design is (asymptotically) lossless in terms of the m-normalized learning rate.

E. From Abstract Channels To Physical Implementation

We now explain how this exercise of creating abstract channels and the corresponding lossless AF can be converted to a new scheme for any $\boldsymbol{r}^*$-uniform L-hop line network with learning rate $\boldsymbol{r}$ arbitrarily close to the min-cut $\boldsymbol{r}^*$, see (10).

For any given $\varepsilon > 0$, we first construct L symmetric quantizer classes $\{\boldsymbol{\pi}_m^{[l]} : m\}$, one for each physical channel CH_{e_l} such that all of them are asymptotically d_0-linear satisfying $d_0 \triangleq \boldsymbol{r}^* - 0.5\varepsilon$, which is feasible by Proposition 3. We then concatenate the first two abstract channels using lossless AF. The combined abstract channel is asymptotically d_0-linear by Proposition 4. We then apply lossless AF again to combine the new channel with the third abstract channel. This is possible since our construction is a black-box design that uses only the locations of the limiting points (x_γ, y_γ), see (22)–(23), and is oblivious of the construction of the abstract channel. By repeatedly applying lossless AF, the e2e abstract channel is

[4]Eq. (11) was described for scalar Γ. Since the e2e abstract channel is of vector output, one can first relabel each vector output by an integer in $\{0, \cdots, (\overline{\gamma}_2)^{\overline{\gamma}_1-1} - 1\}$ and then apply the formulas.

[5]Our abstract channels do involve quantizers $\boldsymbol{\pi}_m$. However, the quantizers $\boldsymbol{\pi}_m$ in our designs are part of the abstract channels, not part of the concatenation mechanism. The mechanism in (22) that "concatenates two abstract channels" is bijective and reversible, a defining feature of AF schemes.

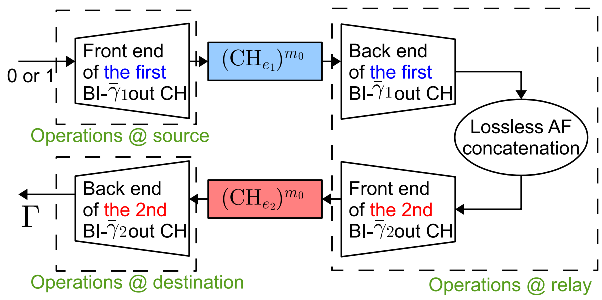

Fig. 3. Illustration for a 2-hop line network.

still asymptotically d_0-linear. We then choose an m_0 such that the e2e abstract channel's normalized learning rate satisfies

$$\bar{\boldsymbol{r}}_{m_0}^{[\text{e2e}]} > d_0 - 0.5\varepsilon = (\boldsymbol{r}^* - 0.5\varepsilon) - 0.5\varepsilon = \boldsymbol{r}^* - \varepsilon \quad (25)$$

which is doable because of Proposition 2. The chosen m_0 value is then fixed throughout the rest of the discussion.

The next step is to let the physical L-hop network "simulate" the concatenation of the abstract channels we have just designed. See Fig. 3 for illustration. E.g., the l-th relay will carry out the "back-end" computation of the l-th BI-$\bar{\gamma}_l$-out channel, followed by the lossless AF concatenation, followed by the "front-end" of the $(l+1)$-th BI-$\bar{\gamma}_{l+1}$-out channel. For simplicity, Fig. 3 does not illustrate the detailed *channel splitting* operations (22) and (23) of lossless AF concatnation.

Each simulation will take m_0 channel uses, and we group the corresponding m_0 channel uses (also known as time slots) as a "sub-block". Because of the strict causality requirement in (2), the beginning of a sub-block of the l-th hop can only start *after* the end of the corresponding sub-block of the previous hop. This can be achieved by shifting the time axis of the l-th relay by $l \cdot m_0$ time slots. The operation is repeated and *pipelined* for K times. That is, each node will spend a sub-block of m_0 slots and collectively they (one source plus $(L-1)$ relays) will simulate one copy of the e2e abstract channel. We then let the network simulate K copies of the e2e channel, i.e., s uses slots $[1, K \cdot m_0]$ and the l-th relay uses slots $[(l-1)m_0+1, (K+l)m_0]$ for all $l \in [1, L-1]$. The destination will finish receiving the observations by time $(K + (L-1))m_0$. This pipelined, sub-block-based operations are standard in the literature, see the detailed description in [3], [4].

Since the optimal scheme (over these K copies of the e2e abstract channel) is just a simple repetition code, we let source s repeat the message bit $\Theta \in \{0, 1\}$ for K times, and send each of them over the K parallel copies of the e2e abstract channels, respectively. Destination d will receive K outputs, one from each copy of the e2e abstract channel, and it then performs ML decoding described in (3). By letting $K \to \infty$ while keeping m_0 fixed, this scheme thus achieves the desired $\bar{\boldsymbol{r}}_{m_0}^{[\text{e2e}]}$ in (25), which has already been normalized over m_0 uses of the physical channels, see (16). The slight delay increase from Km_0 to $(K + (L-1))m_0$ caused by the time-shift at each relay is negligible in our learning-rate analysis, which has $K \to \infty$ while using a fixed m_0. Comparing (25) and Proposition 1, the converse bound of the $\boldsymbol{r}^*$-uniform L-hop network can thus be approached arbitrarily closely regardless how large L is.

IV. Achievability For General Acyclic Networks

For a general acyclic network, let $\boldsymbol{r}^*$ denote the min-cut bound in Proposition 1. By the max-flow/min-cut theorem, one can find a finite set of s-to-d directed paths $\{\mathsf{path}_1, \cdots, \mathsf{path}_F\}$, where each path_f is associated with a rate $\boldsymbol{r}_{\mathsf{path}_f} \geq 0$ satisfying

$$\forall e \in E, \quad \sum_{\mathsf{path}_f : \mathsf{path}_f \ni e} \boldsymbol{r}_{\mathsf{path}_f} \leq \boldsymbol{r}_e^* \quad (26)$$

$$\text{and} \quad \sum_{\mathsf{path}_f} \boldsymbol{r}_{\mathsf{path}_f} = \boldsymbol{r}^*. \quad (27)$$

We now consider two cases. **Case 1:** All F paths are *edge disjoint*. In this case, each path_f can be viewed as an independent $\boldsymbol{r}_{\mathsf{path}_f}$-uniform L_f-hop line network. By the discussion in Sec. III-E, each line network can "simulate" an e2e BISO abstract channel, of which the normalized learning rate is arbitrarily close to $\boldsymbol{r}_{\mathsf{path}_f}$. *Since we have F independent e2e BISO abstract channels, the learning rate of the combined F-dimensional vector abstract channel is the sum of the individual learning rates.* By the same sub-block-based implementation in Sec. III-E that sends a repetition code over K copies of the F-dimensional e2e vector abstract channel for a sufficiently large K, our scheme achieves a learning rate $\boldsymbol{r}$ arbitrarily close to $\boldsymbol{r}^*$, also see (27).

Case 2: Some of the F paths share an edge e. We address this case by *temporal multiplexing*. For example, suppose path_1 and path_2 share an edge e and they happen to have $\boldsymbol{r}_{\mathsf{path}_1} = \boldsymbol{r}_{\mathsf{path}_2} = 0.5\boldsymbol{r}_e^*$. We then use the odd (resp. even) time slots of CH_e to carry out the traffic over path_1 (resp. path_2). Since only $0.5m$ time slots of CH_e are used to carry out the traffic over path_i, when normalized over the original m value, it is as if we are creating a new edge $\tilde{e}_i$ with learning rate $\boldsymbol{r}_{\tilde{e}_i}^* = 0.5\boldsymbol{r}_e^*$ for $i \in \{1, 2\}$. Since lossless AF is a black-box design that uses only the locations of the limiting points of $\mathsf{LLR.rf}(\rho)$ and is oblivious of the actual construction of the abstract channel, the new $\tilde{e}_i$ can be readily concatenated with the next hop of path_i. Case 2 is essentially reduced back to Case 1 via this temporal multiplexing technique. By carefully formulating it for general acyclic networks and by reusing the arguments of Case 1, one can design a scheme that achieves a learning rate $\boldsymbol{r}$ arbitrarily close to $\boldsymbol{r}^*$ in Case 2 as well.

V. Conclusion

This work has characterized the optimal learning rate $\boldsymbol{r}^*$ of sending a one-bit message over any arbitrary acyclic BISO-channel network.

Acknowledgement

This work was supported in parts by NSF under Grant CCF-1816013, EEC-1941529, CCF-2008527, and CNS-2107363.

REFERENCES

[1] Y. Geng, C. Nair, S. Shamai, and Z. Wang, "On broadcast channels with binary inputs and symmetric outputs," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 6980–6989, November 2013.

[2] C.-C. Wang, S. Kulkarni, and H. Poor, "Finite-dimensional bounds on $\mathbf{Z}_m$ and binary LDPC codes with belief propagation decoders," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 56–81, January 2007.

[3] D. Ogbe, C.-C. Wang, and D. Love, "On the optimal delay amplification factor of multi-hop relay channels," in *Proc. IEEE Int'l Symp. Information Theory (ISIT)*. Paris, France, July 2019.

[4] Y. H. Ling and J. Scarlett, "Optimal rates of teaching and learning under uncertainty," *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 7067–7080, November 2021.

[5] W. Huleihel, Y. Polyanskiy, and O. Shayevitz, "Relaying one bit across a tandem of binary-symmetric channels," in *Proc. IEEE Int'l Symp. Inform. Theory*, July 2019.

[6] C.-C. Wang, D. Love, and D. Ogbe, "Transcoding: A new strategy for relay channels," in *Proc. 55th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, Illinois, USA, October 2017.

[7] V. Jog and P.-L. Loh, "Teaching and learning in uncertainty," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, January 2021.

[8] C.-C. Wang, "Pruning network coding traffic by network coding — a new class of max-flow algorithms," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1909–1929, April 2010.

[9] C.-C. Wang and M. Chen, "Sending perishable information: Coding improves delay-constrained throughput even for single unicast," *IEEE Trans. Inf. Theory*, vol. 63, no. 1, pp. 252–279, January 2017.

[10] C.-C. Wang and N. Shroff, "Beyond the butterfly — a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Proc. IEEE Trans. Inform. Theory*. Nice, France, June 2007.

[11] ——, "Pairwise intersession network coding on directed networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 3879–3900, August 2010.

[12] C. Shannon, R. Gallager, and E. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels, i." *Information and Control*, vol. 10, pp. 65–103, 1967.

[13] S.-H. Lee and S.-Y. Chung, "When is compress-and-forward optimal?" in *Proc. Inform. Theory and Applications Workshop*. San Diego, USA, February 2010.

[14] M. Yassaee and M. Aref, "Generalized compress-and-forward strategy for relay networks," in *Proc. IEEE Int'l Symp. Inform. Theory*. Toronto, Canada, June 2008.

[15] V. Nagpal, I.-H. Wang, M. Jorgovanovic, D. Tse, and B. Nikolic, "Quantize-map-and-forward relaying: Coding and system design," in *Proc. 48th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, Illinois, USA, September 2010.

[16] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, October 2011.