On The Optimal Delay Growth Rate of Multi-hop Line Networks: Asymptotically Delay-Optimal Designs And The Corresponding Error Exponents

Dennis Ogbe, *Member, IEEE,* Chih-Chun Wang, *Senior Member, IEEE,* and David J. Love, *Fellow, IEEE*

***Abstract*—Multi-hop line networks have emerged as an important abstract model for modern and increasingly dense communication networks. In addition, the growth of real-time and mission-critical services has created high demand for and increased research interest in low-latency communications. The combination of these facts motivates a new investigation of data transmission schemes for L-hop line networks from a delay-vs-throughput perspective. To this end, this work defines a metric called the *delay amplification factor* for a target throughput R, denoted by $\mathsf{DAF}(R)$, which characterizes the growth rate of the (asymptotic) delay with respect to the number of hops. We show that all existing relay schemes, e.g., Decode-&-Forward (DF), have $\lim_{R\nearrow C}\mathsf{DAF}(R) = \Omega(L)$, which is consistent with the decades-old perception that delay grows linearly with respect to L. We then design a scheme satisfying $\lim_{R\nearrow C}\mathsf{DAF}(R) = 1$, if the bottleneck hop is the last hop, i.e., its asymptotic delay *does not* grow with respect to L. The results imply that this linearly growing delay is an artifact of the existing DF designs, and it is possible to surpass it and attain the true fundamental limit with a new delay-centric solution. In the second half of this work, we further show that if variable-length coding and one-bit *stop-feedback* are allowed, we can relax the condition bottleneck being the last hop and attain $\lim_{R\nearrow C}\mathsf{DAF}(R) = 1$ for any arbitrary line networks.**

***Index Terms*—Low-latency communication, relay channel, line network, error exponent, transcoding, delay-throughput tradeoff, finite-length analysis**

I. Introduction

A. Reliability function

DEFINE $p_e(R,n)$ as the message error probability of a communication channel under encoding rate R and block length n. The reliability function of a communication channel $E(R)$:

$$E(R) \triangleq \lim_{n\to\infty} -\frac{1}{n}\log\big(p_e(R,n)\big) \tag{1}$$

then describes the asymptotic tradeoff between the error probability versus the length of the coded messages. Along with the concept of capacity, the reliability function of a communication channel has played a significant role in the development of information theory. This emphasis on the *error probability vs. codeword length* tradeoff is particularly relevant for modern ultra-reliable ultra-low-latency (URLLC) communications [1], massive machine-type communications (mMTC) [2], and rural communications [3] since long codeword lengths translate directly to a long *transmission delay* between the start of the transmission at the sender and the actual extraction of the messages at the receiver, even if we assume that the underlying encoding/decoding algorithms can be carried out and finished instantaneously (with infinite hardware clock rate).

The reliability function of point-to-point channels is a well-studied subject. In 1959, Shannon discovered upper and lower bounds on the error exponent of the Additive White Gaussian Noise Channels (AWGNCs) [4], which spearheaded numerous follow-up works in the next decades, including but not limited to [5]–[14]. More advanced studies beyond the reliability function have received significant attention in recent years under the new framework of *finite-length analysis*, see e.g., [15]–[22]. These latter works focus on the more practically relevant *communication rate vs. codeword length* tradeoff under a fixed error probability requirement as opposed to the more traditional *error probability vs. codeword length* tradeoff under a fixed communication rate requirement. Essentially, both the reliability function and finite-length analysis study the joint relationship between error probability, throughput, and delay; and this work falls under the same umbrella as these important prior results.

B. Multi-hop line networks

One signature trait of modern wireless communication systems is the overall densification of the network due to novel infrastructure devices such as femto- or pico-cells. With the continuing densification in 5G and beyond-5G networks, these "small cells" are increasingly not directly connected to fiber-optic networks due to cost or other site-specific constraints. Instead, they rely on, potentially multiple, wireless connections before reaching a wired connection, with work in 5G NR done for this scenario under the Integrated Access and Backhaul (IAB) framework [23]–[25], Cloud Radio Access Networks (C-RAN) [26], or distributed transmission and reception [27]–[29]. This has kindled renewed interest in classical relay channels.

The history of the relay channel dates back to the general three-terminal channels by van der Meulen [30] in the 1970s,

D. Ogbe is with the NASA Jet Propulsion Laboratory, Pasadena, CA 91109, USA (e-mail: do@ogbe.net). D. J. Love and C.-C. Wang are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA (e-mail: {chihw,djlove}@purdue.edu).

A version of this paper was presented at the IEEE International Symposium on Information Theory 2019 in Paris, France. This work was supported in parts by NSF under Grants CCF-1618475, CNS-1642982, CCF-1816013, EEC-1941529, CCF-2008527, CNS-2107363, CNS-2212565, and CNS-2225577.

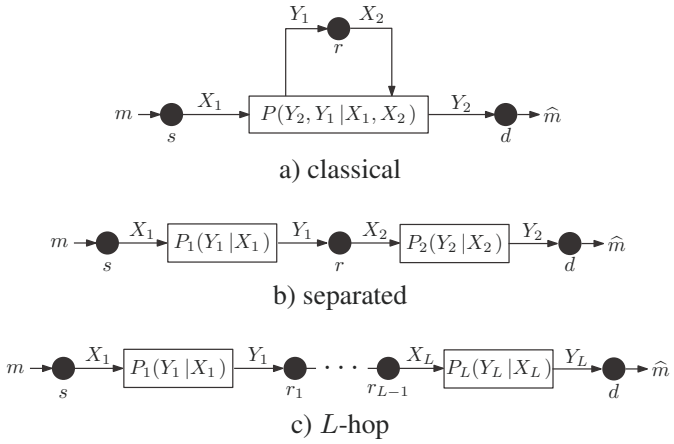

Fig. 1. Variations on the relay channel model.

and a comprehensive review can be found in [31]. Fig. 1a describes the most general relay channel model, for which the relay channel is modeled as the joint conditional probability $P(Y_2, Y_1 | X_1, X_2)$, where Y_1 and Y_2 represent the received signals at the relay and the destination, respectively, and X_1 and X_2 represent the signals transmitted at the source and the relay, respectively. While the relay channel has continued to attract research interest throughout the years [32]–[41], the capacity of the most general relay model remains unknown.[1] The difficulty of characterizing the general relay channel capacity lies in the fact that with an arbitrarily given conditional distribution $P(Y_2, Y_1 | X_1, X_2)$, a relay may "assist" with the communication task between the source and the destination by judicially selecting its transmitted value X_2. Various ideas exploring this possibility have been developed, including Compress-&-Forward (CompressF) [31], [42], Compute-&-Forward [43], Noisy Network Coding [44], etc.

As challenging as the general relay channel study can be, almost all practical relay systems are operated under the so-called *separated relay channel* in Fig. 1b., for which there is no direct link between the source and the destination. Specifically, the joint distribution admits a simpler form $P(Y_2, Y_1 | X_1, X_2) = P_1(Y_1 | X_1) P_2(Y_2 | X_2)$ that separates the destination from the source. By the max-flow/min-cut theorem [31], the capacity of the separated relay channel is the bottleneck hop capacity: $C = \min(C_1, C_2)$, where C_1 (resp. C_2) is the capacity of the source-to-relay (resp. relay-to-destination) channel, and the capacity is achieved by Decode-&-Forward (DF) [31]. The separated relay channel model and its capacity analysis can be easily extended to the multi-hop line network depicted in Fig. 1c. While being one of the simplest communication networks, multi-hop line networks are arguably the most widely used relay model for any wireless/wireline network, for which the sole task of the intermediate nodes is to "relay" the messages from source to destination, not to actively alter/assist the direct transmission through $P(Y_2, Y_1 | X_1, X_2)$.

[1]A short, non-comprehensive list of the types of relay channels for which the capacity is known includes: separated relay channel (see Sec. II), degraded and reversely degraded relay channel [31], semi-deterministic relay channel [32], permuting relay channel [33], deterministic relay channel [34], two-way relay channel [36], diamond channel [37], and (genie-aided) non-causal relay channel [39].

C. Reliability function of multi-hop line networks

This work studies the reliability function of multi-hop line networks, i.e., the asymptotic relationship of delay versus error probability. In particular, we study the following questions:

> *Question 1:* For any given multi-hop line network, does there exist an upper bound on the (largest) error exponent for any transmission scheme we can possibly design?
> *Question 2:* If such a bound on the error exponent exists, can we design a new scheme from scratch that approaches said bound?

Our goal of characterizing and approaching the optimal error exponent among *all possible schemes* separates this work from the existing results. For example, [11] derived the error exponents of the existing (partial) DF scheme, the Compress-F scheme, etc., but no new scheme was developed. In contrast, this work is not bound by any existing design philosophy and aims to directly optimize the end-to-end error exponent by proposing new analytical approaches and achievability ideas.

D. Our contributions

Before we quantitatively define the optimality of the proposed schemes, we first formulate a general class of relay schemes for multi-hop line networks with L hops, which has the following desirable features: (i) It includes any existing relay schemes as special cases and enables fair comparison that takes into account various important techniques like block Markov coding, pipelining, and/or full-duplex capability; (ii) It is flexible for any $L \geq 1$; (iii) If $L = 1$ the class of schemes (and its definitions of rates and block-lengths) naturally coincides with the traditional definitions of block codes for point-to-point channels [4]; and (iv) If $L = 2$, the proposed new definition is identical to the specialization of traditional block-based relay schemes [31], originally devised for the general $P(Y_2, Y_1 | X_1, X_2)$, to the 2-hop separated relay setting in this work.

Based on the new problem formulation, for any arbitrarily given scheme, we introduce a new metric dubbed the *delay amplification factor*, denoted by $\mathsf{DAF}(R)$, which measures the ratio of the *asymptotic delay* of applying the scheme over the L-hop line network versus the asymptotic delay of the (optimal) random-coding delay over just the bottleneck hop. For example, over a 5-hop line network, a scheme achieving $\mathsf{DAF}(R) = 3.5$ has an asymptotic delay roughly 3.5 times higher than the random-coding delay over just the bottleneck hop and incurs only $\frac{3.5}{5} = 70\%$ delay when compared to a uniform time-division transmission scheme.[2]

Using the new problem formulation and the new metric $\mathsf{DAF}(R)$, we now summarize our main contributions:

[2]A uniform time-division transmission scheme over an L-hop line network will take L times the delay experienced in the bottleneck hop for the packets to traverse from the source to destination over the L hops. Therefore such a scheme will have $\mathsf{DAF}(R) = L$.

1) A new achievability scheme for the open-loop setting: It is intuitive[3] that the $\mathsf{DAF}(R) \geq 1$ for any scheme and $\mathsf{DAF}(R) = \Omega(L)$ for DF schemes because their delay grows linearly with respect to the number of hops. In this work we show that by designing a new scheme from scratch, we can attain an $\mathsf{DAF}(R)$ satisfying $\lim_{R\nearrow C} \mathsf{DAF}(R) = 1$ if the bottleneck hop is the L-th hop (the last hop) of the line network.

Analytically, the results show that the common belief that the delay over an L-hop line network grows linearly with respect to L is not a fundamental limit, but rather an artifact of the delay-suboptimal DF schemes. In fact, when operating at rate $R \to C$, the end-to-end delay over L hops can be made comparable to the delay over the single bottleneck hop, i.e., $\mathsf{DAF}(R) \to 1$. Also see [45] for some system-level design ideas and numerical verifications on lowering the end-to-end delay beyond what is possible in the traditional DF-based paradigm.

2) A new achievability scheme for the stop-feedback setting: The previous contribution is based on a feedback-free setting. However, for practical wireless multi-hop communications, we almost always have (some form of) ACK feedback for each of the L hops. Even in the simpler point-to-point channel ($L = 1$), the use of ACK feedback has led to significant performance improvements in the form of hybrid ARQ [46]–[54]. In this contribution, we thus consider the setting of *stop-feedback* [16] of the system, and show that with the help of the stop-feedback, we can relax the "bottleneck hop being the last hop" condition and design a scheme with $\lim_{R\nearrow C} \mathsf{DAF}(R) = 1$ for *arbitrary line networks*. This finding establishes that with the one-time stop-feedback, the asymptotic delay of a multi-hop line network can be made as small as the asymptotic delay of its bottleneck hop regardless of the bottleneck hop position.

Remark: It is known that the stop-feedback can shorten the (expected) delay of variable length coding [16]. For fair comparison, we define the $\mathsf{DAF}(R)$ in Sec. II-C as the ratio of the expected delay of the given scheme over the *improved* expected delay over the bottleneck hop. The discussion of $\lim_{R\nearrow C} \mathsf{DAF}(R) = 1$ with stop-feedback is based on this new definition.

E. Remarks on the setting of sending a 1-bit message

Broadly speaking, this work analyzes the optimal error exponent of sending a message of rate $R > 0$ (i.e., the cardinality of the message set being e^{nR} where n is the block length) over an L-hop line network. The closest related works are [55]–[58]. Specifically, these works consider 2-hop line networks and analyze the optimal error exponent of the error probability. However, they focus exclusively on sending a 1-bit message (the cardinality of the message set being two). The optimal error exponent has been characterized if both hops are binary symmetric channels (BSCs) but remains open for the general setting of arbitrary channel distributions. Our results can be viewed as the counterpart of [55]–[58] for which the messages are of strictly positive rate since the 1-bit message can be viewed as an asymptotically zero-rate message. Such a generalization from the bounded alphabet setting (i.e., with asymptotically zero-rate messages) [55]–[58] to the setting of capacity-approaching rates would guide future research for scenarios more complicated than the one-bit case.

The remainder of this paper is structured as follows. We give the channel model and all necessary definitions in Secs. II-A to II-C. Secs. II-D to II-F provides some intuition for the new framework and discusses the application of our analysis to the DF and any block Markov coding schemes. Sec. III is dedicated to the open-loop setting, with Secs. III-A and III-B containing the main results; Sec. III-C containing a detailed description of the proposed transmission scheme; and Sec. III-D covering the $\mathsf{DAF}(R)$ analysis of said scheme. Sec. IV is dedicated to the stop-feedback setting, with Secs. IV-A and IV-B containing a high-level and a detailed description of the proposed transmission scheme, respectively. Sec. IV-C covers the $\mathsf{DAF}(R)$ analysis of it. Finally, we conclude the paper in Sec. V.

II. Problem Formulation

A. The multi-hop line network channel model

We define the stationary memoryless L-hop line network as follows, also see Fig. 1c. Denote the source node as s, the destination node as d and the $L-1$ intermediate relay nodes as r_ℓ for all $\ell = 1, \ldots, L-1$, respectively. The first hop connects (s, r_1); the ℓ-th hop, $\ell \in [2, L-1]$, connects $(r_{\ell-1}, r_\ell)$; and the L-th hop connects (r_{L-1}, d). Consider slotted transmission for $t = 1, 2, \ldots$. One symbol is sent in each time slot, which is sometimes called a *channel use*[4]. The channel of each hop is discrete, stationary, and memoryless, and we denote the input and output symbols of the ℓ-th hop at time slot t as $X_\ell(t)$ and $Y_\ell(t)$, respectively. We denote the input alphabet, output alphabet, and conditional distribution of the ℓ-th channel as $\mathcal{X}_\ell$, $\mathcal{Y}_\ell$, and $P_\ell(y_\ell|x_\ell)$, respectively. We refer to the ℓ-th channel as $\mathcal{W}_\ell = (\mathcal{X}_\ell, \mathcal{Y}_\ell, P_\ell)$.

Denote the Shannon capacity (or simply capacity) of the ℓ-th hop by C_ℓ (unit: nats/slot):

$$C_\ell \triangleq \max_{P_{X_\ell}} I(X_\ell; Y_\ell) \tag{2}$$

where $I(X_\ell; Y_\ell)$ is the mutual information, and the maximization is over all possible distributions of the finite input alphabet $\mathcal{X}_\ell$. It is well known that the end-to-end capacity of a line network is $C = \min_\ell C_\ell$.

Technical Assumptions. *Assumption 1:* All our results assume exclusively that there exists a unique hop ℓ^* with the lowest capacity, i.e., $\exists \ell^*$ such that $C_\ell > C_{\ell^*}\ \forall \ell \neq \ell^*$. We refer to this hop as the *bottleneck hop* and assume $C_{\ell^*} > 0$. In practice, the corner case where there are two hops satisfying $C_{\ell_1} = C_{\ell_2} = \min_\ell \{c_\ell\}$ with infinite precision is unlikely. Furthermore, we can always apply some infinitesimal perturbation to break the tie if it happens. This assumption is thus not too restrictive for practical applications.

[3]We will formalize this part of discussion in Sec. II.

[4]In practice, the time for each channel use may vary from hop to hop. For simplicity, our model assumes all hops sharing a common channel use time. Our results can be easily revised to handle heterogeneous slot duration as well.

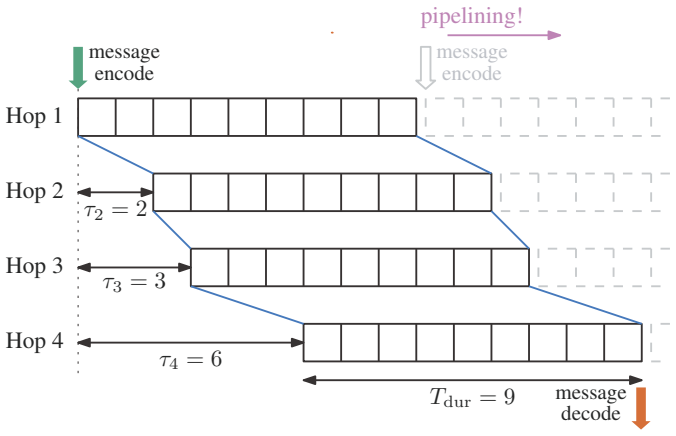

Fig. 2. Starting times, transmission duration, and encoding/decoding time-points for an example open-loop scheme (unit: slots).

Assumption 2: We assume that all transmission probabilities are non-zero, i.e., $P_\ell(y_\ell|x_\ell) > 0 \ \forall x_\ell, y_\ell, \ell$. This assumption is to ensure that the error exponent of any of the L hops is bounded away from infinity. This assumption can be relaxed in ways similar to Conditions (a) and (b) in [59, Sec. 1].

The source s wishes to send an integer message m, drawn uniformly randomly from $\mathcal{M} = \{1, 2, \ldots, |\mathcal{M}|\}$ to destination d using a *transmission scheme* Φ. A transmission scheme consists of the following elements depending on whether it is an open-loop setting or if stop-feedback is allowed.

B. The open-loop setting

A transmission scheme Φ in the open-loop setting consists of the following elements:

Starting times and duration. A sequence of L deterministic, non-decreasing time points

$$\tau_1 = 0 \leq \tau_2 \leq \cdots \leq \tau_L < \infty \tag{3}$$

determines the starting times (unit: slots) of data transmission for the corresponding hops. The maximum duration[5] of the transmission at each node is denoted as $T_{\rm dur}$ (unit: slots). Fig. 2 gives an example for $T_{\rm dur} = 9$ and starting times $\tau_1 = 0, \tau_2 = 2, \tau_3 = 3$, and $\tau_4 = 6$ over $L = 4$ hops.

Sequential encoders at the relay nodes. We assume full-duplex relays with strictly causal encoding. That is,

$$X_1(t) = f_t^{(1)}(m), \quad \forall t \in (\tau_1, \tau_1 + T_{\rm dur}] \tag{4}$$

$$X_\ell(t) = f_t^{(\ell)}([Y_{\ell-1}]_*^{t-1}), \quad \forall \ell \geq 2, \ \forall t \in (\tau_\ell, \tau_\ell + T_{\rm dur}] \tag{5}$$

where $f_t^{(\ell)}$ is the encoder of the ℓ-th hop at time slot t, and

$$[Y_{\ell-1}]_*^{t-1} \triangleq \{Y_{\ell-1}(\tau) : \tau \in (\tau_{\ell-1}, \min(t-1, \tau_{\ell-1} + T_{\rm dur})]\} \tag{6}$$

denotes all strictly causally received[6] observations from the upstream hop. The definition $[Y_{\ell-1}]_*^{t-1}$ imposes that the observation at the transmitter of the ℓ-th hop is always a subset of the upstream hop's "active period" $(\tau_{\ell-1}, \tau_{\ell-1} + T_{\rm dur}]$.

Block decoder at the destination. The final block-based decoding function is given as

$$\widehat{m} = g([Y_L]_*^{\tau_L + T_{\rm dur}}). \tag{7}$$

Our definition of the starting time instants τ_1 to τ_L and the maximum allowable[5] duration $T_{\rm dur}$ is motivated by the concept of *pipelined transmission* in practice. That is, once the source finishes transmitting the current message at time $\tau_1 + T_{\rm dur} = T_{\rm dur}$, it can immediately inject[7] the next message at time $T_{\rm dur} + 1$, even though the current message is still being transmitted to the destination by the rest of the network. This behavior is illustrated with the dotted lines in Fig. 2.

A consequence of the above pipelining assumption, also see the discussion in footnote 7, is that messages arrive at the destination once every $T_{\rm dur}$ slots, which is less than the end-to-end delay $\tau_L + T_{\rm dur}$. This observation leads to the following classification of open-loop transmission schemes, in particular the throughput definition in (9).

Definition 1. *An L-hop open-loop transmission scheme* attains *a delay-throughput-error-probability tuple (T, R, ϵ) if it satisfies*

$$T \geq \tau_L + T_{\rm dur} \tag{8}$$

$$R \leq \frac{\ln(|\mathcal{M}|)}{T_{\rm dur}} \tag{9}$$

$$\epsilon \geq \Pr(\widehat{m} \neq m). \tag{10}$$

The latency definition in (8) counts the delay from the time that the first (coded) symbol of the message is sent by the source to the time that the destination has received the last (coded and potentially corrupted) observed symbol. See Fig. 2. It does not take into account any queueing delay at the source nor any computation delay of the decoding algorithm.

Let $\mathcal{A}_\Phi$ denote the set of all (T, R, ϵ)-tuples attained by scheme Φ, i.e.,

$$\mathcal{A}_\Phi \triangleq \{(T, R, \epsilon) : \Phi \text{ attains } (T, R, \epsilon)\}. \tag{11}$$

Definition 2. *The* end-to-end *error exponent of an open-loop scheme Φ is defined as*

$$E_\Phi(R) \triangleq \liminf_{T \to \infty} \sup_{\epsilon : (T,R,\epsilon) \in \mathcal{A}_\Phi} \frac{-\ln(\epsilon)}{T}. \tag{12}$$

[5] $T_{\rm dur}$ is the maximum number of slots that can be used by each hop. A scheme can instruct some of its relays to use less than $T_{\rm dur}$ slots if desired.

[6] The strict causality condition is imposed in (5) since $X_\ell(t)$ depends only on $[Y_{\ell-1}]_*^{t-1}$, i.e., the *propagation delay* being exactly one time slot. An astute reader may notice that because of the strict causality condition, we can assume, without loss generality, a strict relationship $\tau_1 < \tau_2 < \cdots < \tau_L$ instead of (3). However, it turns out that the relaxed setting that allows for $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_L$ is more flexible when considering the stop-feedback setting in Sec. II-C.

[7] The description is motivated by the *generate-at-will* model used in the network scheduling community [60]–[66], which assumes that whenever the "channel" becomes available, the source can generate a new packet (at will) to take advantage of the availability. Examples of this include sensing and IoT applications [3], [67], [68]. Whenever the channel becomes available, the sensor will measure the environment and send the latest measurement through the available channel. There is thus zero queueing delay in this model, and the end-to-end delay is equal to the time difference between the start and the end of the transmission of a message.

Denote the random-coding error exponent of the ℓ-th hop as $E_{\mathsf{rc},\ell}(R)$. Here we note that since the random-coding error exponent usually depends on the input distribution used, we assume that in all of our use cases the optimal (i.e., maximizing the exponent) input distribution is used. Put succinctly, if $E_{\mathsf{rc},\ell}(R,\mathbf{Q})$ denotes the random coding error exponent of the ℓ-th hop for channel symbols distributed according to $\mathbf{Q}$, then we always use $E_{\mathsf{rc},\ell}(R) \triangleq \sup_{\mathbf{Q}} E_{\mathsf{rc},\ell}(R,\mathbf{Q})$ as shorthand.

Definition 3. *For any fixed throughput R, the delay amplification factor $\mathsf{DAF}(R)$ of an L-hop open-loop transmission scheme Φ is defined as*

$$\mathsf{DAF}_{\Phi}(R) \triangleq \frac{E_{\mathsf{rc},\ell^*}(R)}{E_{\Phi}(R)}. \tag{13}$$

The connection between $\mathsf{DAF}_{\Phi}(R)$ and the delay experienced in a multi-hop communication scheme will be elaborated in Sec. II-E.

C. The stop-feedback setting

In the stop-feedback setting, we assume the *one-time stop-feedback* model [16]. In this model, there is a feedback channel from the destination to all other nodes (the source plus the relays). The feedback channel is assumed to be delay-free and error-free and can be used (by the destination) for one time to indicate the end of a message transmission. This one-time 1-bit stop feedback setting is different and, arguably, more realistic than the *per-slot* channel output feedback models [51], [69]–[76], which assumes that *every channel output symbol* is causally available at the transmitter with zero delay.

Technically, the stop-feedback model implies that a) the message duration T_{dur} in (4) to (7) becomes a stopping time of the filtration generated by $[Y_L]_*^t$, since the destination is the node triggering the *end-of-transmission* for each message; and b) since there is only a single stop-feedback for each message transmission, before the stop-feedback, all nodes are working simultaneously on transmitting the same message and collectively switch to the next message after receiving the end-of-transmission feedback.

As a result, we hardwire $\tau_1 = \tau_2 = \cdots = \tau_L = 0$ and modify Def. 1 for the stop-feedback setting as follows.

Definition 4. *An L-hop stop-feedback transmission scheme* attains *a delay-throughput-error-probability tuple (T,R,ϵ) if it satisfies*

$$T \geq E\{T_{\mathrm{dur}}\} \tag{14}$$

$$R \leq \frac{\ln(|\mathcal{M}|)}{E\{T_{\mathrm{dur}}\}} \tag{15}$$

$$\epsilon \geq \Pr(\widehat{m} \neq m). \tag{16}$$

where T_{dur} is now a stopping time *with respect to the filtration generated by the destination's observation $[Y_L]_*^t$, and the decision function $\widehat{m} = g(\cdot)$ in* (7) *now takes a random variable length observation $[Y_L]_0^{T_{\mathrm{dur}}}$ as input.*

Namely, T in (14) defines the average delay due to the random stopping time T_{dur}, R in (15) denotes the average throughput of the variable-length transmission, and ϵ in (16) defines the error probability for each variable-length transmission.

The definition of the end-to-end error exponent for the stop-feedback setting remains identical to Def. 2. However, due to the availability of the stop-feedback, the bottleneck error exponent changes. Results in [16] show that for a point-to-point channel, stop-feedback improves the random-coding error exponent from $E_{\mathsf{rc},\ell}(R)$ to a strictly larger value

$$E_{\mathsf{sf},\ell}(R) = (C_\ell - R)^+ \triangleq \max(C_\ell - R, 0), \tag{17}$$

which we use in the following new definition of $\widetilde{\mathsf{DAF}}(R)$. Herein we use the tilde to distinguish the stop-feedback-based definition from the open-loop one.

Definition 5. *For any fixed throughput R, the delay amplification factor $\widetilde{\mathsf{DAF}}(R)$ of an L-hop variable-length stop-feedback scheme Φ is defined as*

$$\widetilde{\mathsf{DAF}}_{\Phi}(R) \triangleq \frac{E_{\mathsf{sf},\ell^*}(R)}{E_{\Phi}(R)} = \frac{C_{\ell^*} - R}{E_{\Phi}(R)}. \tag{18}$$

D. Discussion #1: Decode-&-Forward is a special instance of this framework

To demonstrate the flexibility of our problem formulation, we notice that when $L = 1$, the term T_{dur} is equivalent to the codeword length of block coding, since we always have $\tau_1 = 0$. The new definitions in Secs. II-A and II-B are thus identical to the traditional point-to-point channel reliability functions.

An astute reader may notice that our problem formulations in Secs. II-A to II-C never define the *block length* as in many traditional information-theoretic results. The reason is that for a multi-hop relay setting, different relays may choose to be "active" at different time instants and sometimes choose different block lengths for each hop if it benefits the end-to-end transmission, also see our discussion in the end of this subsection. Therefore, using a constant scalar block length to describe a multi-hop relay scheme is too restrictive since it prohibits any heterogeneity for which that different relays may opt. Furthermore, it is also possible that the relays may choose a *streaming-code-based design*, see [77]–[79], that does not have any block-based structure. The use of starting time instants τ_1 to τ_L and the maximum allowable[5] duration T_{dur} allows us to include all possible design choices under the same analysis framework.

To further illustrate the flexibility of the new framework, we provide some intuition and discussion on how it includes DF as a special case. We restrict our focus to the open-loop setting since it is how DF was originally designed, but the insights gained apply to the stop-feedback variant of DF e.g., using the stop-feedback scheme in [16] as part of DF.

For illustration purposes, we assume that the DF scheme uses random block codes for each of the L hops. Specifically, in a DF scheme, the source s first produces a corresponding randomly-generated codeword of block length t_1 (unit: slots) and forwards it over the first channel to r_1. After the codeword is received, possibly in error, at the relay r_1, r_1 computes an estimate of the message and produces another randomly generated codeword using this estimate. Suppose that the length

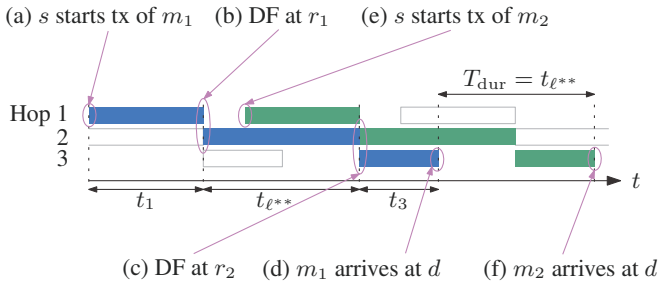

Fig. 3. Relaying a message through an example 3-hop line network using the DF scheme.

of r_1's codeword is t_2 (unit: slots). This codeword is then forwarded to r_2, and so forth. After L hops, the destination receives a codeword of length t_L and computes an estimate of the original message, denoted by $\widehat{m}$. This procedure is illustrated in items (a)–(d) of Fig. 3. The codeword lengths $t_1, t_2, \ldots, t_L$ may vary for each hop and the end-to-end delay is thus $T \triangleq t_1+t_2+\cdots+t_L$. When pipelining this DF scheme, the smallest allowed interval between two consecutive message injections is $\max_\ell t_\ell$ and the maximal sustainable throughput is thus $\log(M)/\max_\ell t_\ell$.

We now explain how this DF description fits the framework in Secs. II-A and II-B. Define $\ell^{**} \triangleq \arg\max_\ell\{t_\ell\}$. We set the starting times $\{\tau_\ell\}$ and duration $T_{\rm dur}$ to be

$$\tau_\ell = \max\left(0, -t_{\ell^{**}} + \sum_{j=1}^{\ell} t_j\right) \quad \forall \ell \in [1, L] \tag{19}$$

$$T_{\rm dur} = \max_\ell t_\ell = t_{\ell^{**}}. \tag{20}$$

With the above choices, we can easily verify that (i) $\tau_1 = 0 \le \tau_2 \le \cdots \le \tau_L$; and (ii) the *forwarding time slots* of the ℓ-th hop, i.e., $(\sum_{j=1}^{\ell-1} t_j, \sum_{j=1}^{\ell} t_j]$, are a subset of its active period $(\tau_\ell, \tau_\ell + T_{\rm dur}]$. The DF encoders and decoders can thus be rewritten as a special instance of (4) to (7).

The achievable tuple (T, R, ϵ) for DF (after strengthening the inequalities of (8)–(10) to equalities) then satisfies

$$T = \sum_{\ell=1}^{L} t_\ell = \tau_L + T_{\rm dur} \tag{21}$$

$$R = \frac{\ln(|\mathcal{M}|)}{t_{\ell^{**}}} = \frac{\ln(|\mathcal{M}|)}{T_{\rm dur}} \tag{22}$$

$$\epsilon = \Pr(\widehat{m} \neq m) = 1 - \prod_{\ell=1}^{L}(1-\epsilon_\ell) \le \sum_{\ell=1}^{L} \epsilon_\ell \tag{23}$$

where ϵ_ℓ is the random coding error probability of the ℓ-th hop. Note that our pipelining rate definition in (22) (and thus (9)) indeed matches the commonly used definition of the achievable rate of DF and the delay definition in (21) also matches the commonly used definition of the delay of DF.

Following similar reasoning, we can easily show that other schemes like AF, Compress-F, Noisy Network Coding, etc., are all special instances of the new framework, and the rate and delay definitions of ours coincide with the commonly used definitions for the individual schemes. Thus, this provides the footing for a fair comparison between different schemes.

E. Discussion #2: The physical interpretation of the $\mathsf{DAF}(R)$

We now demonstrate the relationship between the $\mathsf{DAF}_\Phi(R)$ and the end-to-end delay of an arbitrarily given scheme Φ. Specifically, by (12), for any fixed rate R and error probability ϵ, the end-to-end delay T of scheme Φ is approximately

$$T(R, \epsilon) \approx -\ln(\epsilon)/E_\Phi(R). \tag{24}$$

If we apply a rate-R random code over just the bottleneck hop ℓ^* (while ignoring all other hops), to achieve the same error probability ϵ, the corresponding delay is approximately

$$T_{\rm btl}(R, \epsilon) \approx -\ln(\epsilon)/E_{\mathsf{rc},\ell^*}(R)\,. \tag{25}$$

As a result, the delay ratio can be approximated by $\frac{T(R,\epsilon)}{T_{\rm btl}(R,\epsilon)} \approx \mathsf{DAF}_\Phi(R)$ following the definition in (13), provided we focus on the ultra-reliable (asymptotically small ϵ) regime. Note that the *finite-block-length* analysis [15] can be viewed as a strengthened version of the delay-throughput-error-probability analysis with ϵ strictly bounded away from 0. If history is any indication, the error exponent analysis in this work would be a first step for the future finite-block-length analysis of multi-hop line networks.

The main motivation behind this $\mathsf{DAF}_\Phi(R)$ definition is to provide a useful self-contained metric when measuring the performance a multi-hop scheme. For example, when comparing the latency performance of two schemes Φ_1 and Φ_2, it is important to compare the reciprocal of their error exponents $\frac{1}{E_{\Phi_1}(R)}$ versus $\frac{1}{E_{\Phi_2}(R)}$. Without relying on any external benchmark scheme, $\mathsf{DAF}_\Phi(R)$ can be viewed as a *self-benchmarked metric* as $\mathsf{DAF}_\Phi(R)$ captures the multiplicative increase of the end-to-end multi-hop delay over the optimal singe-hop delay of the *worst-hop-only* scenario.

Since modern communication schemes are targeting a rate arbitrarily close to the capacity, we are especially interested in evaluating $\lim_{R\nearrow C} \mathsf{DAF}_\Phi(R)$ for a given scheme Φ. From this respect, the new metric $\mathsf{DAF}_\Phi(R)$ is especially convenient in a sense that for any scheme Φ, we always have $E_\Phi(R)$ (and also $E_{\mathsf{rc}}(R)$) converge to zero when $R \nearrow C$. It is the ratio between them that really matters, which motivates the definition of $\mathsf{DAF}_\Phi(R)$.

To further demonstrate the new metric, we analyze the $\mathsf{DAF}_\Phi(R)$ value of the DF scheme discussed in Sec. II-D. For any rate R, define $E_{\mathsf{sp},\ell^*}(R)$, as the sphere-packing error exponent of the bottleneck hop.

Lemma 1. *Consider any rate R satisfying $E_{\mathsf{rc},\ell^*}(R) = E_{\mathsf{sp},\ell^*}(R)$, which holds for R that is sufficiently close to C [59]. For the DF scheme in Sec. II-D, regardless how we choose its parameters we always have*

$$\mathsf{DAF}_{\rm DF}(R) \ge 1 + \sum_{\ell\in[1,L]\setminus\ell^*} \frac{R}{C_\ell}. \tag{26}$$

Furthermore, when $R \to C$, we have

$$\lim_{R\nearrow C} \mathsf{DAF}_{\rm DF}(R) = \sum_{\ell=1}^{L} \frac{C_{\ell^*}}{C_\ell}. \tag{27}$$

If all C_ℓ are of comparable magnitude, then (26) lower bounds the growth rate of $\mathsf{DAF}_{\rm DF}(R) = \Omega(L)$ versus the

number of hops, which is consistent with the intuition of linearly growing delay for DF. At the same time, (27) shows that when operating at rate R sufficiently close to C, we can further strengthen the lower bound and characterize the exact value of $\mathsf{DAF}_{\rm DF}(R)$.

We note that DF represents a class of schemes that can have different choices[8] of the alphabet size $|\mathcal{M}|$ and the active periods of each hop t_ℓ, see Sec. II-D. As a result, to prove Lemma 1, we have to show (i) there exists a way of choosing $(|\mathcal{M}|, t_1, \cdots, t_L)$ such that the corresponding $\lim_{R\nearrow C} \mathsf{DAF}_{\rm DF}(R) \leq \sum_{\ell=1}^{L} \frac{C_{\ell^*}}{C_\ell}$; and (ii) regardless how we choose $(|\mathcal{M}|, t_1, \cdots, t_L)$, we always have $\mathsf{DAF}_{\rm DF}(R)$ satisfying (26). The detailed proofs of both directions are provided in Appendix A.

Remark: While the proof of Lemma 1 involves carefully applying the definitions in Secs. II-A and II-B to calculate the error exponents and its $\mathsf{DAF}(R)$ value, the results are actually quite intuitive. Take (27) as an example. Suppose the source would like to send b bits to the destination using DF. It takes roughly $\frac{b}{C_\ell}$ time slots to traverse over the ℓ-th hop. The total delay is thus $\sum_{\ell=1}^{L} \frac{b}{C_\ell}$. If we only have the bottleneck hop, then the point-to-point channel delay is $\frac{b}{C_{\ell^*}}$. The ratio of total delay versus point-to-point bottleneck delay $\sum_{\ell=1}^{L} \frac{C_{\ell^*}}{C_\ell}$ is indeed the $\mathsf{DAF}(R)$ in (27). Also see the first couple of paragraphs in Sec. II-E.

For example, if $C_1 = 5$ (nats/symbol), $C_2 = 4$, and $C_3 = 3$, the $\mathsf{DAF}(R)$ of DF is $\frac{3}{5} + \frac{3}{4} + \frac{3}{3} = 2.35$ if R is sufficiently close to C. Namely, the end-to-end delay of DF in this 3-hop example is roughly 2.35 times the delay experienced in a point-to-point system consisting of only the third hop.

F. Discussion #3: The error-exponent penalty of Block Markov Coding — A common drawback among all existing relay schemes

Existing finite-length analysis works [11], [20], [80] are based on well-known relay policies like DF, compress-&-forward, etc. It is worth emphasizing that the schemes in [11], [20], [80] are designed to have superior capacity performance in a *general* relay channel model. A common building block of these schemes is the *Block Markov Coding* (BMC) technique, for which one divides the main block into multiple microblocks, applies (near-)optimal decoding for each microblock, and the relay causally adjusts its future transmission to further improve the performance. With intelligent and innovative designs, see [31] and the references therein, BMC could significantly enlarge the achievable rate region.

In contrast, this work considers *separated* relay channels, for which the capacity is known completely. Thus, there is no need to use microblocks and BMC to enlarge the achievable rate. In fact, any use of the BMC[9] design is detrimental to the error exponent performance. For example, [11] characterizes the error exponent of 2-hop relays under DF, partial DF (PDF), and compress-&-forward (Comp-F). Block Markov Coding (BMC) is analyzed with $b \geq 2$ number of blocks. With b microblocks, the effective error protection is only applied to a micro codeword/codebook of $\frac{1}{b}$ of the overall duration. After we normalize with respect to the end-to-end total duration, the effective error exponent is reduced by a factor of $\frac{1}{b}$. With $b \geq 2$ to begin with, the the scheme in [11] achieves $\mathsf{DAF}(R) \geq 2$, which is consistent with the characterization of (27) in Lemma 1 (assuming $C_1 = C_2$) and is aligned with the intuition that the delays of DF, PDF, and Comp-F, all grow linearly with respect to L (where we have used the example case of $L = 2$ in this discussion).

Since our goal is to maximize the error exponent, if one must stick to the existing designs without any innovative improvement, it is better not to use any microblock at all (i.e., $b = 1$) and try to always group all encoding in a single block, which maximizes the error protection. In Lemma 1, we deliberately consider the DF design that does not use any microblock, which is the most powerful DF scheme from the perspective of maximizing the error exponent. As a result, it can be viewed as the optimal $\mathsf{DAF}(R)$ for any DF scheme one can possibly devise.

Note that with the setting of separated relay channels, all other existing schemes, such as PDF and Comp-F, exhibit similar behavior as DF in the sense that all of them incur delay that grows linearly with respect to the number of hops, i.e., $\mathsf{DAF}(R) = \Omega(L)$. The reason is that in those schemes the transmitter of each hop has to accumulate enough observations (in order to have a highly-reliable estimate) before it starts sending new coded symbols to its receiver. The need of waiting for having highly reliable estimate and then using that estimate to construct the next-hop transmission is the root cause of linearly growing $\mathsf{DAF}(R)$. Our results show that we can develop new schemes with significantly shorter delay than the state of the art. The main idea is to still use a large number of $b \gg 1$ microblocks but mitigate the error exponent reduction $\frac{1}{b}$ by additional code structure that offers global codeword dependence/protection that is sharply different from the local, short-ranged, Markov dependence structure in BMC.

Remark: In this work, we exclusively study discrete memoryless channels (DMCs). The extension to continuous channels is an important subject that is beyond the scope of this work. In particular, for continuous channels, we usually need to place further constraints on the input distributions, e.g., the average power constraint and/or the maximum amplitude constraint for AWGNCs. This adds another degree-of-freedom when designing the multi-hop relay schemes. We believe that the focus on DMCs in this work provides a starting point for this line of research since if we restrict our focus to the traditional modulation/constellation constraints, e.g., BPSK,

[8] In fact, the description of DF also includes how to choose the codebook for each hop. Lemma 1 and its proof in Appendix A allow the class of DF to include any possible codebook choices as well. Nonetheless, due to the (near-)optimality of random codes, it is more intuitive for readers to temporarily assume random coding over each hop and focus only on the choices of $(|\mathcal{M}|, t_1, \cdots, t_L)$.

[9] In this work, the term *BMC* exclusively refers to the schemes for which different bits in the overall coded vector exhibit a strict *block Markov* structure, which is how the term BMC is traditionally defined. For example, considering three microblocks B_1 to B_3 in sequence, the coded bits in B_1 and B_3 must be independent once we condition on the intermediate microblock B_2.

16QAM, one can always quantize the continuous channels into their DMC counterparts and all our analyses/results would easily follow suit. For the scenario in which the input alphabet is continuous and unbounded, say AWGNCs with an input power constraint, we believe the same results would hold once we include the design of the input distributions as part of the achievability scheme. The main technical challenge is that for DMCs, the input distribution is optimized over a closed set but for continuous channels, the optimization is over an open set, which warrants more careful analysis in the limiting scenarios.

III. Main Result #1 — The open-loop setting

All the results in this section are based exclusively on the open loop setting in Secs. II-A and II-B.

A. The converse of the optimal $\mathsf{DAF}(R)$

Proposition 1. *For any rate R that is sufficiently close to the capacity C such that it satisfies $E_{\mathsf{rc},\ell^*}(R) = E_{\mathsf{sp},\ell^*}(R)$ [59], regardless of the transmission scheme Φ, we always have $\mathsf{DAF}_\Phi(R) \geq 1$.*

Proof. The proof is by reduction. Suppose there exists an L-hop line network $\{P_\ell(y_\ell|x_\ell) : \ell \in [1, L]\}$ and a scheme Φ such that $\mathsf{DAF}_\Phi(R) < 1$, which implies that $E_\Phi(R) > E_{\mathsf{rc},\ell^*}(R)$. Recall that a scheme Φ is determined by $\{\tau_\ell : \ell \in [1, L]\}$, T_{dur}, and the encoding/decoding functions in (4) to (7). We will use the given scheme to construct a block code over the point-to-point (p2p) channel $P_{\ell^*}(y_{\ell^*}|x_{\ell^*})$ and show that the corresponding error exponent will be strictly larger than the sphere packing bound $E_{\mathsf{sp},\ell^*}(R)$, the needed contradiction.

The rest of the proof is straightforward. The block length of the p2p channel is set to $T = \tau_L + T_{\mathrm{dur}}$. The encoder of the p2p channel uses the given multi-hop scheme, encodes the message, *simulates* the operations/transmissions of the first $\ell^* - 1$ hops, and physically sends out the encoded symbols that are supposed to be sent over the ℓ^*-th hop over the physical p2p channel. Namely, it forfeits the first τ_{ℓ^*} time slots and only uses the interval $(\tau_{\ell^*}, \tau_{\ell^*} + T_{\mathrm{dur}}]$ even though the overall block length is $\tau_L + T_{\mathrm{dur}}$. The receiver of the p2p channel will simulate the remaining $L - \ell^*$ hops plus the final decoder. Per our definitions in (8) to (12), if the given multi-hop scheme achieves the error exponent $E_\Phi(R) > E_{\mathsf{rc},\ell^*}(R)$, then the new p2p scheme will attain the same error exponent which is greater than $E_{\mathsf{rc},\ell^*}(R)$. However, since we consider R being sufficiently close to C and satisfying $E_{\mathsf{rc},\ell^*}(R) = E_{\mathsf{sp},\ell^*}(R)$, this implies that the block code surpasses the sphere packing bound. By contradiction, the proof is complete. ∎

Proposition 1 effectively answers Question 1 from Sec. I-C.

B. An achievability scheme for the setting $\ell^* = L$

Proposition 2. *Consider arbitrary $L \geq 2$ and arbitrary channels $P_\ell(y_\ell|x_\ell)$ for $\ell = 1, \ldots, L$. If the unique bottleneck hop is the last hop, i.e., $\ell^* = L$, then there exists a scheme and a threshold $R_0 < C$ such that the scheme achieves*

$$\mathsf{DAF}(R) = \frac{\left\lfloor \frac{C}{C-R} \right\rfloor + L - 1}{\left\lfloor \frac{C}{C-R} \right\rfloor}, \quad \forall R \in [R_0, C). \tag{28}$$

Proposition 2 immediately implies the following corollary:

Corollary 1. *For any given L-hop line network, the scheme in Proposition 2 approaches the lower bound $\mathsf{DAF}(R) \geq 1$ when R is sufficiently close to C. That is,*

$$\lim_{R \nearrow C} \mathsf{DAF}(R) = 1. \tag{29}$$

Note that even though the $\mathsf{DAF}(R)$ in (28) still grows linearly at a rate $\frac{C-R}{C} \cdot L$, the coefficient $\frac{C-R}{C}$ diminishes when $R \nearrow C$, which is vastly different than the DF scheme, for which its DAF is uniformly bounded below by (26) regardless of the R value. By the lower bound from Proposition 1, such a scheme is delay-amplification-factor-optimal when R is sufficiently close to C. I.e., $\mathsf{DAF}(R)$ can be made arbitrarily close to 1 as long as $R \geq C - \frac{\mathsf{const}}{L}$ for some sufficiently small $\mathsf{const} > 0$.

In Sec. III-C, we describe the mechanics of the scheme in detail. We then characterize the $\mathsf{DAF}(R)$ of our proposed scheme in Sec. III-D.

C. Description of the transmission scheme

Our scheme is inspired by the transcoding design [45] and the concatenated coding structure [81], [82]. The main ideas are as follows. Each message m is mapped to K microblocks of equal length Δ symbols. The K microblocks are obtained using a concatenated code [81] consisting of a single end-to-end outer code and a set of inner codes for each individual hop. After the source generates the K microblocks, each microblock is relayed through the L hops using DF in a pipelined fashion, but the decoders at the *relays* use *only* the inner codes. After accumulating all K microblocks, the destination performs optimal *joint* inner/outer-code ML decoding. Such an inner/outer code structure is also used in the 1-bit-message optimal-learning-rate achievability scheme in [55].

We call such a scheme *transcoding* since the relay nodes do not perform full global decoding and re-encoding. Instead, the decoding and re-encoding operates on a local scale and "transforms" the signals from one inner codeword to another inner codeword along the hops. The transcoding scheme for an L-hop line network is parameterized by the number of microblocks K, the code rate R (unit: nats/slot), and the microblock length Δ. It consists of the following elements.

Partitioning the time axis as microblocks. Every Δ time slots are grouped as a microblock. That is, the k-th microblock refers to the time slots $t \in ((k-1)\Delta, k\Delta]$. All the operations are aligned in time with the microblocks. In particular, we set the starting time instant $\tau_\ell = (\ell - 1)\Delta$ for all $\ell \in [1, L]$ and the duration $T_{\mathrm{dur}} = K\Delta$.

Inner/outer code architecture.

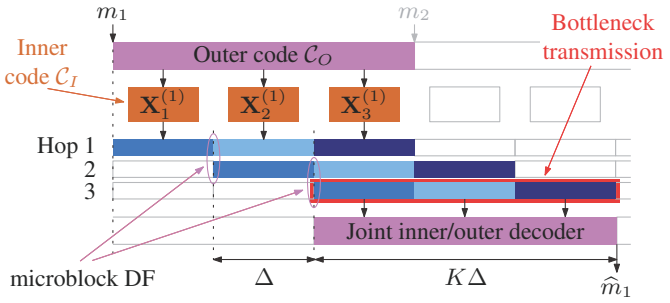

Fig. 4. An asymptotically delay-amplification-factor-optimal transmission scheme for bottleneck-terminated DMC line networks. K microblocks using concatenated code construction. ($K = 3$ and $L = 3$ pictured for illustration)

- We define the inner code rate $R_I = \frac{K}{K-1}R$, for which the physical interpretation will be clear shortly after.
- The message set is $\mathcal{M} = [1, e^{K\Delta R}] = [1, e^{(K-1)\Delta R_I}]$.
- The outer code $\mathcal{C}_O$ is a single parity check code over K *super symbols*, where each super symbol is chosen from $[0, e^{\Delta R_I})$. Namely, any message $m \in \mathcal{M}$ is first bijectively mapped to a $(K-1)$-dimensional vector $(i_1^{[m]}, \ldots, i_{K-1}^{[m]}) \in [0, e^{\Delta R_I})^{(K-1)}$. Then a parity super symbol $i_K^{[m]}$ is computed from solving the parity check equation
$$\left(\sum_{k=1}^{K} i_k^{[m]} \mod e^{\Delta R_I}\right) = 0.$$
- A set of $K \cdot L$ random inner codebooks
$$\mathcal{C}_I = \left\{\{\mathbf{X}_1^{(1)}, \ldots, \mathbf{X}_K^{(1)}\}, \ldots, \{\mathbf{X}_1^{(L)}, \ldots, \mathbf{X}_K^{(L)}\}\right\}.$$
As will be shown later, $\mathbf{X}_k^{(\ell)}$ is the k-th codebook used by the ℓ-th hop. We assume that each codebook $\mathbf{X}_k^{(\ell)}$ consists of $e^{\Delta R_I}$ codewords of length Δ symbols:
$$\mathbf{X}_k^{(\ell)} = \left\{\mathbf{x}_{k,i}^{(\ell)} \in (\mathcal{X}_\ell)^\Delta : i \in [0, e^{\Delta R_I})\right\},$$
where each coordinate of the i-th (Δ-dimensional) codeword $\mathbf{x}_{k,i}^{(\ell)}$ is sampled independently and identically (i.i.d.) from the error-exponent-achieving input distribution $P_\ell^*(X_\ell)$ for the ℓ-th channel $P_\ell(y_\ell|x_\ell)$. The random codebook is generated independently for different k and ℓ as well. Namely, even for the same ℓ, we repeat the independent random codebook construction for different k_1 and k_2.

Encoding at the source node. The source encoding function $f_1 : \mathcal{M} \mapsto (\mathcal{X}_1)^{K\Delta}$ maps a message m to a block of symbols of length $K\Delta$ by
$$f_1(m) = \left(\mathbf{x}_{1,i_1^{[m]}}^{(1)}, \mathbf{x}_{2,i_2^{[m]}}^{(1)}, \ldots, \mathbf{x}_{K,i_K^{[m]}}^{(1)}\right),$$
where the $i_k^{[m]}$ are obtained from the single-parity outer code and each microblock $\mathbf{x}_{k,i}^{(1)}$ is drawn from the random inner codebook $\mathbf{X}_k^{(1)}$. Since $\tau_1 = 0$ and $T_{\text{dur}} = K\Delta$ and all transmissions are aligned with the microblocks, the source s will transmit each inner codeword $\mathbf{x}_{k,i_k^{[m]}}^{(1)}$ during the k-th microblock.

Relaying through the network. We define a set of $K \cdot (L-1)$ relaying functions $f_k^{(\ell)} : (\mathcal{Y}_{\ell-1})^\Delta \mapsto (\mathcal{X}_\ell)^\Delta$ for all $\ell \in [2, L]$ which map the channel outputs from the $(\ell-1)$-th hop to the inputs of the ℓ-th hop at the relays. The mapping is performed by combining the maximum-likelihood (ML) *inner code* decoder of the previous hop plus the inner code encoder of the current hop, i.e.,
$$f_k^{(\ell)}(\vec{Y}_{\ell-1}[k+\ell-2]) = \mathbf{x}_{k,\widehat{i}_k^{(\ell-1)}}^{(\ell)}, \tag{30}$$
where the vector $\vec{Y}_{\ell-1}[j] \triangleq \{Y_{\ell-1}(\tau) : \tau \in ((j-1)\Delta, j\Delta]\}$ is the Δ-dimensional observation of the $(\ell-1)$-th hop over the j-th microblock; the microblock index being $k+\ell-2$ is because since the starting time of the ℓ-th hop is $\tau_{\ell-1} = (\ell-1)\Delta$, the k-th microblock of the $(\ell-1)$-th hop is occupying the $(k+\ell-2)$-th microblock in the overall time axis, also see Fig. 4. The index
$$\widehat{i}_k^{(\ell-1)} = \underset{i \in [0, e^{\Delta R_I})}{\arg\max}\ P_{\ell-1}\left(\vec{Y}_{\ell-1}[k+\ell-2] \,\middle|\, \mathbf{x}_{k,i}^{(\ell-1)}\right) \tag{31}$$
is the optimal ML inner code decoder over the just received microblock, for which we slightly abuse the notation $P_{\ell-1}(\cdot|\cdot)$ by letting
$$\begin{aligned} &P_{\ell-1}\left(\vec{Y}_{\ell-1}[k+\ell-2] \,\middle|\, \mathbf{x}_{k,i}^{(\ell-1)}\right) \\ &\quad\triangleq \Pr\left(\vec{Y}_{\ell-1}[k+\ell-2] \,\middle|\, \vec{X}_{\ell-1}[k+\ell-2] = \mathbf{x}_{k,i}^{(\ell-1)}\right). \end{aligned} \tag{32}$$

After decoding in (31) and re-encoding in (30), the k-th microblock of the ℓ-th hop will be transmitted in the $(k+\ell-1)$-th microblock in the overall time axis (recalling that the starting time $\tau_\ell = (\ell-1)\Delta$).

Decoding at the destination. We define a decoding function $g : (\mathcal{Y}_L)^{K\Delta} \mapsto \mathcal{M}$ which maps an observation of K microblocks at the destination to an estimate of the message $\widehat{m}$ using the ML joint inner/outer decoder, i.e.,
$$\widehat{m} = \underset{m \in \mathcal{M}}{\arg\max}\ P_L\left(\left[\vec{Y}_L\right]_1^K \,\middle|\, \mathbf{c}_L^{[m]}\right), \tag{33}$$
where $\left[\vec{Y}_L\right]_1^K = \left\{\vec{Y}_L[j+L-1] : j \in [1, K]\right\}$ denotes the K microblocks received through the L-th hop;
$$\mathbf{c}_L^{[m]} = \left(\mathbf{x}_{1,i_1^{[m]}}^{(L)}, \ldots, \mathbf{x}_{K,i_K^{[m]}}^{(L)}\right) \tag{34}$$
is the concatenation of the outer code with the K inner codebooks $\mathbf{X}_1^{(L)}$ through $\mathbf{X}_K^{(L)}$ designed for the L-th hop; and $P_L(\mathbf{y}\,|\mathbf{c}\,)$ is the conditional probability (likelihood) of receiving $\mathbf{y} \in (\mathcal{Y}_L)^{K\Delta}$ at the destination given that $\mathbf{c} \in (\mathcal{X}_L)^{K\Delta}$ was transmitted over the last hop, i.e., we slightly abuse the notation in a way similar to (32). The difference between (31) and (33) is that the former ML decoder uses only the inner codebook while the latter is the ML joint inner/outer decoder.

The error probability is defined as $\epsilon = \Pr(m \neq \widehat{m})$. In the next subsection, we demonstrate how to choose the parameters (K, R, Δ) of the transcoding scheme to attain $\mathsf{DAF}_\Phi(R)$ described in (28).

D. DAF(R) *analysis*

Recall that any transcoding scheme is defined by the tuple (R, K, Δ). As a result, its error exponent is determined by the pair R, K since in (12) we let $T \to \infty$ and thus the third coordinate $\Delta \to \infty$. As a result, we use the notation $E_\Phi(R, K)$ for the error exponent of the transcoding scheme $\Phi(R, K, \Delta)$. We then have

Lemma 2. *Assume $\ell^* = L$. There exists an $R_0 < C$ such that for any $R \in [R_0, C)$, we have*

$$E_\Phi(R, K^*(R)) = \frac{K^*(R)}{K^*(R) + L - 1} \cdot E_{\mathsf{rc},L}(R), \quad (35)$$

where $K^*(R)$ is the largest K still satisfying $R_I = \frac{K}{K-1}R \geq C$. The closed-form expression of $K^*(R)$ is

$$K^*(R) \triangleq \left\lfloor \frac{C}{C-R} \right\rfloor. \quad (36)$$

Proposition 2 is a direct result of Lemma 2. The remainder of this section, in which we use K^* as shorthand for $K^*(R)$, proves the above lemma.

1) Probability of error: Recall that m was the selected message at the source. The corresponding tuple of microblock messages, generated by the source, is denoted as $\mathbf{i}^{[m]} = \left(i_1^{[m]}, i_2^{[m]}, \ldots, i_{K^*}^{[m]}\right)$. During the transmission, the estimate of the k-th microblock message at the receiver of the ℓ-th hop is denoted as $\widehat{i}_k^{(\ell)}$. Define $\mathcal{A}$ as the event that there exists at least one pair $(k, \ell) \in [1, K^*] \times [1, L-1]$ satisfying $\widehat{i}_k^{(\ell)} \neq i_k^{[m]}$. Namely, $\mathcal{A}$ is the event that at least one inner decoder (among the relays but excluding the destination) is in error.

Recall that $\left[\vec{Y}_L\right]_1^{K^*}$ denotes the received symbols at the destination. Using the union bound, we can then bound the probability of message error as

$$\begin{aligned}\epsilon &\triangleq \Pr(\widehat{m} \neq m) \\ &\leq \Pr(\mathcal{A}) + \Pr\left(\widehat{m} \neq m \middle| \mathcal{A}^c\right). \end{aligned} \quad (37)$$

The intuition behind (37) is that since we transmit the message over L hops, there are two types of errors. The first type is the error caused by performing DF using only the inner codes during the first $(L-1)$ hops, and the second type is the error of the joint inner/outer decoder at the destination. Our goal is not just to show that (37) converges to zero when $\Delta \to \infty$. Instead, one needs to show that (37) converges to zero at speed no slower than $e^{-K^*\Delta E_{\mathsf{rc},L}(R)}$.

We proceed by bounding the two terms in (37) individually. For the first term, we use the individual random coding error exponents for the first $L-1$ hops and obtain the following bound for all $R \in [R_0, C)$ for some R_0 that is sufficiently close to C.

$$\begin{aligned}\Pr(\mathcal{A}) &\leq \sum_{k=1}^{K^*} \sum_{\ell=1}^{L-1} e^{-\Delta E_{\mathsf{rc},\ell}(R_I)} && (38)\\ &\leq K^* \sum_{\ell=1}^{L-1} e^{-\Delta E_{\mathsf{rc},\ell}\left(\frac{C_L + \min_{\rho \neq L} C_\rho}{2}\right)} && (39)\\ &\leq K^*(L-1) e^{-K^*\Delta E_{\mathsf{rc},L}(R)}, && (40)\end{aligned}$$

where (38) follows from the union bound, and (39) follows from the following arguments. Recall that K^* is the largest K still satisfying $R_I = \frac{K}{K-1}R \geq C$. Therefore, when $R \nearrow C$ from below, we will have $R_I \searrow C$ from above. Since $C = C_L$ has a non-zero gap to the second smallest capacity $\min_{\rho \neq L} C_\rho$, when R is sufficiently close to C, we must have

$$R_I \leq C_L + 0.5((\min_{\rho \neq L} C_\rho) - C_L). \quad (41)$$

Namely, R_I is strictly bounded away from $0.5C_L + 0.5 \min_{\rho \neq L} C_\rho$. This implies (39).

Ineq. (40) follows from the fact that when R is sufficiently close to $C = C_L$, we always have

$$E_{\mathsf{rc},L}(R) < \frac{\min_{\ell \in [1, L-1]} E_{\mathsf{rc},\ell}\left(\frac{C_L + \min_{\rho \neq L} C_\rho}{2}\right)}{K^*}. \quad (42)$$

The reason is that $E_{\mathsf{rc},L}(R) = O((C-R)^2)$ is a quadratic function of $(C-R)$ when R is sufficiently close to C [83, Ex. 5.23], but K^* is approximately $\frac{C}{C-R}$. Since the numerator of the right-hand side of (42) is a constant, when $R \nearrow C$, the left-hand side of (42) will eventually be strictly less than the right-hand side of (42). Therefore, we can choose a sufficiently large $R_0 < C$ value such that (40) holds for all $R \in [R_0, C)$.

Note that the above discussion shows that the inner-code protection for the non-bottleneck hops $\ell \neq L$ is strong enough to achieve the desired error exponent error probability $e^{-K^*\Delta E_{\mathsf{rc},L}(R)}$ *before* the message going through the last hop. We now focus on the last hop. Suppose we only use the inner decoder at the bottleneck/last hop without any outer code protection. Because the inner codeword length is Δ, the convergence speed of the error probability is $e^{-\Delta E_{\mathsf{rc},L}(R)}$, which is $\frac{1}{K^*}$ of the desired decay rate. This is the reason why we perform joint inner-outer decoding at the destination (the receiver of the bottleneck/last hop) in order to recover the full error exponent. This is also the reason that the error event $\mathcal{A}$ only considers the first $L-1$ hops and we perform the analysis separately for the bottleneck/last hop.

The following lemma upper bounds the second term in (37):

Lemma 3. *We have*

$$\Pr\left(\widehat{m} \neq m \middle| \mathcal{A}^c\right) \leq 2^{K^*} e^{-K^*\Delta \cdot E_{\mathsf{rc},L}(R)} \quad (43)$$

The proof is relegated to Appendix B.

It is worth emphasizing that the single-parity-check outer code itself is not able to correct any error when combined with a hard (separated) inner code decoder. Nonetheless, under the joint inner/outer ML decoder, our concatenated code design achieves the same error exponent as the non-concatenated classical random code ensemble.[10]

Combining (40) and (43), we then get

$$\epsilon \leq \left(K^*(L-1) + 2^{K^*}\right) e^{-K^*\Delta \cdot E_{\mathsf{rc},L}(R)}. \quad (44)$$

[10]A closely related result was first discovered in [84], which focused on a strictly more general MDS-code-based setting than the simple parity-check-code-based construction herein.

2) End-to-end latency: The length of each microblock is Δ. There are K^* total microblocks per message. Each microblock needs to traverse L hops. In total, the latency of one message is given as

$$T = (K^* + L - 1)\Delta. \tag{45}$$

3) Error exponent and $\mathsf{DAF}(R)$*:* Now, combining the results about the error probability and delay, we can derive the end-to-end error exponent of the proposed transmission scheme:

$$\begin{aligned} E_\Phi(R) &= \lim_{T\to\infty} \frac{-\ln(\epsilon)}{T} \\ &= \lim_{\Delta\to\infty} \frac{-\ln\left(\left(K^*(L-1)+2^{K^*}\right)e^{-K^*\Delta\cdot E_{\mathsf{rc},L}(R)}\right)}{(K^*+L-1)\Delta} \\ &= \frac{K^*}{K^*+L-1} E_{\mathsf{rc},L}(R). \end{aligned} \tag{46}$$

The proof of Lemma 2 is complete.

We now note that even if the bottleneck hop is not the last hop, i.e., $\ell^* \neq L$, we can iteratively apply our proposed scheme to achieve the following $\mathsf{DAF}(R)$ value.

Corollary 2. *For the case of $\ell^* \neq L$, define $\ell_0^* = 0$ and iteratively define $\ell_i^* = \arg\min_{\ell\in(\ell_{i-1}^*, L]} C_\ell$ for $i = 1, 2, 3, \cdots$ until $\ell_i^* = L$. Suppose there are I such ℓ_i^*, i.e., $\ell_I^* = L$. Also assume the minimum is unique when computing each ℓ_i^*. Then we can construct a scheme Φ such that*

$$\lim_{R\nearrow C} \mathsf{DAF}_\Phi(R) = \sum_{i=1}^{I} \frac{C_{\ell_1^*}}{C_{\ell_i^*}}. \tag{47}$$

For example, consider an 8-hop line network with $(C_1, \cdots, C_8) = (1.72, 1.05, 1.94, 1.69, 1.58, 1.14, 1.95, 1.34)$. Per our discussion, we have $\ell_1^* = 2, \ell_2^* = 6, \ell_3^* = 8$, and $I = 3$. The resulting delay amplification factor is thus $\mathsf{DAF}(R) = 2.70$ for R being sufficiently close to the capacity $C = 1.05$. For comparison, the DF scheme has $\lim_{R\to C} \mathsf{DAF}_{\mathrm{DF}}(R) = 5.68$ as computed by (27).

In Corollary 2, we can easily see that for R sufficiently close to C, our scheme has superior delay performance than the DF schemes in all scenarios except the ones for which the capacity of each hop rises in ascending order $C_1 \leq C_2 \leq \cdots \leq C_L$. Corollary 2 can be proved by combining the DF principle and the transcoding scheme from Proposition 2.

E. Comparison to Existing Works and Other Short Remarks

Concatenated coding for line networks was also used in [82] under a half-duplex, Gaussian channel setting. Nonetheless, a suboptimal two-stage hard decoding scheme (i.e., decode the inner codes first and then use the hard decisions to decode the outer code) was used in [82], which significantly decreases the error exponent (also see discussion in Forney's thesis [81, Eq. (101)]) and is thus strictly suboptimal. In contrast, we prove that by adopting the concatenated coding structure over multi-hop relays and by using the optimal joint inner/outer decoding at the destination d, we can achieve $\mathsf{DAF}(R)$ that is arbitrarily close to the lower bound of 1 when R is sufficiently close to the capacity C.

The optimal joint inner/outer decoder at the destination is crucial for achieving the near-optimal $\mathsf{DAF}(R)$, which is doable only because of the special assumption that the bottleneck hop is the last hop. In particular, if the last hop is not the bottleneck hop, then the receiver of the bottleneck hop can still forward the *hard-decoded* inner code decisions to its next hop neighbor. The destination then has to infer the original message $\hat{m}$ based on the hard-decoded inner code messages $(\hat{i}_1, \cdots \hat{i}_K)$. Because the outer code being used is a single parity-check code, no error correction is possible when having only the hard-decoded messages $(\hat{i}_1, \cdots \hat{i}_K)$. As a result, the outer/inner code structure provides no additional error correction if the bottleneck hop is not the last hop. Only when the bottleneck hop is the last hop, the destination can perform the optimal joint inner/outer decoding based directly on the noisy "observations" of the bottleneck hop, which is the key component how our scheme achieves the optimal $\lim_{R\nearrow C} \mathsf{DAF}(R) = 1$.

If the bottleneck hop receiver is not the final destination, another possibility is that it waits until it has received all microblocks, and then performs joint decoding and forwards the decoded codeword to the final destination. In this way, the decoded codeword will have high reliability but the action of "waiting" increases the end-to-end delay and it becomes suboptimal from the delay's perspective. In fact, this design of letting the bottleneck hop receiver wait for the entire codeword and then forward it is how we design the high performance but still suboptimal achievability scheme for Corollary 2.

One may be tempted to conjecture that one should always be able to achieve $\lim_{R\to C} \mathsf{DAF}(R) = 1$ regardless of the location of the bottleneck hop. Whether such a conjecture is true remains an open problem. However, a recent result [55] in the highly related *teaching-and-learning-under-uncertainty* model shows that "the order of the hops" matters greatly when characterizing the optimal learning rate (similar to our $\mathsf{DAF}(R)$ metric). Specifically, if the first hop is a Z-channel and the second hop is a BSC, the cut-set bound (similar to our DAF lower bound $\mathsf{DAF}(R) \geq 1$) is provably loose [55, Corollary 2]. That is, it is impossible to achieve/approach the bottleneck performance (similar to $\mathsf{DAF}(R) = 1$) regardless how one designs the 2-hop relay scheme [55]. However, no such result was shown in [55] when the order is reversed (first hop being a BSC and second hop being a Z-channel). Collectively, the lessons learned in [55] suggest that one cannot easily rule out the possibility that $\mathsf{DAF}(R)$ may be strictly bounded away from 1 if the bottleneck hop is not the last hop.

Finally, findings similar to this work have been found in some other multi-hop line network settings. For example, [77]–[79] study the setting of multi-hop line networks with the constituent channels being *adversarial packet erasure channels*. Similar to our discoveries, [77] shows that DF is suboptimal in a delay-aware 2-hop setting since the relay needs to wait for the entire message to be decodable before it starts forwarding the message. A new 2-hop scheme is proposed in [77] where each message is divided into smaller symbols, which echoes the construction in this work. In [79], this is further relaxed by the relay forwarding symbols that have

not been entirely decoded yet, and instead forwarding so-called "estimates" as soon as possible, with the idea that the destination can "sort this out" by the delay deadline. Similar findings and designs for $L \geq 2$ hops have been provided in [78]. The collective findings of [77]–[79] suggest that, in a multi-hop setting, relays should forward small pieces of information as soon as possible, rather than waiting for a reliable decode of the entire message.

In the next section, we will show that if a 1-bit stop feedback is allowed, then we can adopt some networking/control ideas and devise a scheme that attains $\mathsf{DAF}(R)$ close to 1 regardless of the location of the bottleneck hop.

IV. Main Result 2: The stop-feedback setting

In this section, we present a transmission scheme which achieves $\lim_{R\nearrow C} \widetilde{\mathsf{DAF}}(R) = 1$ in the stop-feedback setting. For the stop-feedback setting, there is no guaranteed optimality for random codes. As a result, unlike Proposition 1 for the open-loop setting, our investigation has not established the converse $\widetilde{\mathsf{DAF}}(R) \geq 1$ for the stop-feedback setting. That said, random coding currently achieves the highest error exponent out of all existing stop-feedback solutions [16] and we thus use it as the benchmark in our $\widetilde{\mathsf{DAF}}(R)$ definition.

Proposition 3. *Consider arbitrary $L \geq 2$ and arbitrary channels $P_\ell(y_\ell|x_\ell)$ for $\ell = 1, \ldots, L$. In the stop-feedback setting described in Sec. II-C, there exists a scheme such that for any $\alpha \in (0.5, 1)$, there exists an $R_0(\alpha)$ such that for all $R \in [R_0(\alpha), C)$, we have*

$$\widetilde{\mathsf{DAF}}(R) = \frac{1}{2\alpha - 1}. \tag{48}$$

By choosing α sufficiently close to 1, one can make $\widetilde{\mathsf{DAF}}(R) \to 1$ when operating at rate R sufficiently close to C. We proceed according to the same recipe as in Sec. III and provide the high-level and detailed descriptions of the scheme in Secs. IV-A and IV-B, respectively. We then prove in Sec. IV-C that our scheme achieves $\lim_{R\nearrow C} \widetilde{\mathsf{DAF}}(R) = 1$ regardless of whether $\ell^* = L$ or not.

A. High-level description of the transmission scheme

Recall that we refer to the hop with the minimum capacity as the *bottleneck hop* ℓ^*. We refer to the transmitter and receiver of the bottleneck hop as the *bottleneck transmitter* and *bottleneck receiver*, respectively. All relays between the source and the bottleneck transmitter are referred to as the *pre-bottleneck relays* while all relays between the bottleneck receiver and the destination are referred to as the *post-bottleneck relays*.

The proposed scheme is a variable-length scheme with termination, i.e., all of the nodes in the network work on the same message until the destination signals an *end-of-transmission* (EOT) message over the feedback link.

In the subsequent discussion, we will use Fig. 5 as illustration, which shows an example transmission over an $L = 3$ hop line network where the bottleneck hop is the second hop, i.e., $\ell^* = 2$. Our scheme also uses an inner/outer code architecture. Comparing to the scheme in Sec. III, the open-loop scheme in Sec. III uses its outer code to generate a fixed number of

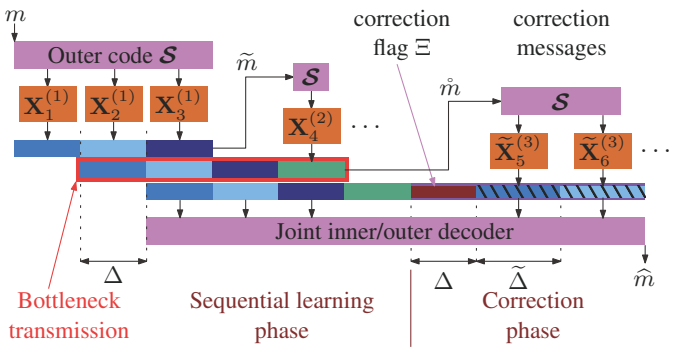

Fig. 5. An asymptotically delay-amplification-factor-optimal transmission scheme for DMC line networks with stop-feedback. Illustration for $L = 3$ hops and $K = 3$ microblocks. The bottleneck transmitter continues to send encoded microblocks after the initial K microblocks are forwarded. The sequential learning phase at the bottleneck receiver lasts $K + 1$ microblock in this example. During the correction phase, the bottleneck receiver sends correction messages to the destination.

microblock messages. The outer code of our stop-feedback scheme generates a *variable-length sequence* of microblock messages for each message. However, as illustrated in Fig. 5, the source node and all the transmitters of the pre-bottleneck hops generate and transmit only the first K microblocks of the infinite-length sequence generated by the outer code. The additional microblock messages are used/sent only by the bottleneck transmitter and all the transmitters of the post-bottleneck hops until the destination declares EOT.

The superior delay performance of our scheme lies in the unique design of the bottleneck receiver. For comparison, in a traditional DF scheme, only when the bottleneck receiver has high confidence[11] of the source message, does it start to re-encode and forward the message to its downstream node(s). However, obtaining a high-confidence estimate requires accumulating a lot of received symbols, which incurs long delay. The main idea of our new scheme is for the bottleneck receiver to start forwarding even before it has deduced a high-confidence estimate, thus the reduced delay. The cost of this early forwarding is that at the time when the bottleneck receiver finally has a high-confidence estimate, it has already forwarded many "low-confidence" microblocks and some of them may be in error, which leads to higher error probability.

To address this drawback, we note that when the bottleneck receiver finally has enough observations to deduce a high-confidence estimate of the source message, it also has compiled the records of all low-confidence microblock messages that have been forwarded to downstream nodes prematurely (before having the high-confidence source message). As a result, we let the bottleneck receiver use the high-confidence estimate of the source message to recompute the set of high-confidence microblocks that *should have been forwarded* to its downstream nodes if it could go back in time. Then it compares those high-confidence microblocks to the low-confidence microblocks that were physically transmitted. If there is any mismatch, the bottleneck receiver *forwards ad-*

[11] In this discussion, we use the term *high confidence* by its intuitive definition. The concept of high confidence will be made rigorous once we provide the complete proof in Sec. IV-C.

ditional microblocks to correct the ones it now knows were relayed in error.

To realize the above idea, we design two separate phases of the message transmission at the bottleneck receiver. In the *sequential learning phase*, the bottleneck receiver aims to obtain the high-confidence message estimate from its observations. In the *correction phase*, the bottleneck receiver aims to correct any mistake(s) due to premature forwarding. Also see the illustration in Fig. 5.

Note that the lengths of both phases are random (and correlated), where the duration of the first phase depends on the channel realization of the bottleneck hop; and the duration of the second phase depends on how many previously transmitted microblocks were in error. Since only the bottleneck receiver knows when the first phase ends and the second phase starts, it has to "instruct" the downstream nodes how to correctly interpret the traffic. To that end, we thus insert a *correction flag* in the forward traffic. The correction flag Ξ can take values in $\{0, 1, \cdots, K\}$. The value $\Xi = 0$ indicates that there is no correction needed and $\Xi = j \geq 1$ indicates that we will send an additional j microblocks that are meant to replace j of the previously transmitted microblocks. We limit the maximum number of replacements to be K, the reason of which will be made clear in the detailed analysis.

The post-bottleneck hops mainly perform DF to help relay the initial microblocks (during the sequential learning phase) and the correction microblocks (during the correction phase) to the destination. After the destination receives the correction flag Ξ, it terminates the transmission of the message for the entire network using the stop-feedback if $\Xi = 0$. Or, if $\Xi = j \geq 1$, it terminates the transmission after receiving j additional microblocks.

B. Detailed description of the transmission scheme

We now provide a detailed construction of the proposed scheme. Specifically, the scheme has four deterministic parameters (K, Δ, R, α), where Δ is the microblock length and $\alpha \in (0.5, 1)$ is a tuning parameter that will be used during the construction. The deterministic parameter K is the *target number of microblocks* used in the transmission since the actual number of microblocks used by the network is a random number. R is directly related to the message alphabet size via $R = \frac{\log(|\mathcal{M}|)}{K\Delta}$ (unit: nats/slot) and can be interpreted as the *target end-to-end throughput*. Again, the actual throughput depends on the expected duration of the variable-length scheme, see (15), and will be analyzed later.

Since we always operate within the capacity, we are exclusively interested in the set of parameter values satisfying $R < C = C_{\ell^*}$. For given values of (K, Δ, R, α), we compute the following constants:

$$R_I \triangleq R + \frac{\alpha + 1}{2}(C_{\ell^*} - R) \tag{49}$$

$$K_{\mathsf{max}} = K e^{K\Delta\alpha(C_{\ell^*} - R)} \tag{50}$$

$$\eta = e^{-K\Delta\alpha(C_{\ell^*} - R)}. \tag{51}$$

Intuitively, R_I is the inner code rate; K_{max} is the upper bound on the length of our variable-length scheme which keeps the scheme from running indefinitely; and η is the *target error probability* of the given scheme.

Outer code at the source. The scheme uses a *sequential random permutation outer code (SRPOC)* $\boldsymbol{\mathcal{S}}$. Given any (K, Δ, R, R_I) tuple, the SRPOC is a rateless code consisting of

1) The message set $\mathcal{M} = [1, e^{K\Delta R}]$.
2) A finite sequence of permutations $\{\pi_k : k \in [1, K_{\mathsf{max}}]\}$.

- A permutation π on $\mathcal{M}$ is a bijective mapping from $\mathcal{M}$ to $\mathcal{M}$. In total, there are $|\mathcal{M}|!$ different permutations.
- Each π_k is drawn independently and uniformly randomly from the set of all $|\mathcal{M}|!$ possible permutations.

3) A finite sequence of encoding functions $\{f_k\}$.

- It is best to interpret k as the *microblock index*.
- For any $k \in [1, K_{\mathsf{max}}]$ and the randomly chosen permutation π_k, the k-th encoding function $f_k : \mathcal{M} \mapsto [1, e^{\Delta R_I}]$ maps a message m to an outer-code symbol $i \in [1, e^{\Delta R_I}]$.
- Specifically, given any $m \in \mathcal{M}$, the output $i_k^{[m]}$ is the unique integer satisfying

$$(i_k^{[m]} - 1) \cdot \frac{|\mathcal{M}|}{e^{\Delta R_I}} < \pi_k(m) \leq i_k^{[m]} \cdot \frac{|\mathcal{M}|}{e^{\Delta R_I}}. \tag{52}$$

- This mechanism partitions the message set $\mathcal{M}$ into buckets of $e^{\Delta R_I}$ messages and selects the corresponding bucket index $i_k^{[m]}$ for a given message $m \in \mathcal{M}$ after applying the permutation π_k.

Inner codes. We use the standard random block code construction and choose the coordinate values of each inner codeword i.i.d. according to the capacity-achieving input distribution. For mathematical rigor, we never reuse any codebook and all codewords/codebooks are generated independently. We now describe how many inner codebooks are used in our scheme.

1) For each of the pre-bottleneck hops $\ell \in [1, \ell^*)$ and for all $k \in [1, K]$, we have a random inner codebook $\mathbf{X}_k^{(\ell)}$ for which the codeword length is Δ (unit: slots) and the total number of codewords in each codebook is $e^{\Delta R_I}$. In total, there are $K(\ell^* - 1)$ of them and these codebooks are used for microblock-based DF in the pre-bottleneck hops.

2) For the bottleneck hop, we have K_{max} inner codebooks $\mathbf{X}_k^{(\ell^*)}$ for all $k \in [1, K_{\mathsf{max}}]$ for which the codeword length of each codebook is again Δ (unit: slots) and the total number of codewords in each codebook is $e^{\Delta R_I}$. The only difference between the bottleneck hop and the pre-bottleneck hops is that the bottleneck hop needs to support variable-length encoding and thus we need to prepare a larger number of codebooks $K_{\mathsf{max}} \geq K$.

3) For each of the post-bottleneck hops $\ell \in [\ell^* + 1, L]$, we have *two* finite sequences of codebooks and we denote them by $\{\mathring{\mathbf{X}}_k^{(\ell)} : k \in [1, K_{\mathsf{max}}]\}$ and $\{\widetilde{\mathbf{X}}_k^{(\ell)} : k \in [1, K]\}$, respectively. The constructions of $\mathring{\mathbf{X}}_k^{(\ell)}$ and $\widetilde{\mathbf{X}}_k^{(\ell)}$ have the following subtle but important differences.

- Each $\mathring{\mathbf{X}}_k^{(\ell)}$ codebook has codeword length Δ but the total number of codewords is $e^{\Delta R_I} + K + 1$. Namely, compared to the pre-bottleneck codebooks, $\mathring{\mathbf{X}}_k^{(\ell)}$ has $K + 1$ additional codewords which are used to represent the correction flag values $\Xi = 0$ to K discussed previously.

- Each $\widetilde{\mathbf{X}}_k^{(\ell)}$ codebook has codeword length $\widetilde{\Delta} = \Delta + \frac{\log(K)}{R_I}$ and the total number of codewords is $e^{\widetilde{\Delta} R_I} = K e^{\Delta R_I}$. Namely, the effective rate of the codebook $\widetilde{\mathbf{X}}_k^{(\ell)}$ is still R_I (unit: nats/slot), but we slightly elongate the codeword length to accommodate a factor of K more codewords than the pre-bottleneck codebooks $\mathbf{X}_k^{(\ell)}$. The new codebook $\widetilde{\mathbf{X}}_k^{(\ell)}$ is used to carry the *correction microblocks*. More specifically, to correct a microblock, we not only have to specify the new value of the previously incorrect microblock but also which of the previous microblocks is being corrected (i.e., the microblock index). Thus, we expand the number of codewords by a factor of K so that we can specify, out of the K previously sent microblocks, which one needs to be corrected by which value. More details can be found shortly after.

We are now ready to describe how to assemble the outer and inner codes for the final scheme.

Encoding at the source node. The source first maps the message m to a sequence of K microblock messages $\{i_k^{[m]}\}_{k=1}^K$ using the outer code $\boldsymbol{\mathcal{S}}$. Then it transmits $\mathbf{x}_{k,i_k^{[m]}}^{(1)}$, one codeword for each microblock, sequentially for $k \in [1, K]$. In the end, the source transmits $K\Delta$ symbols over the first hop. After the end of its transmission, the source idles[12].

Relaying at the pre-bottleneck nodes. The transmitters of the pre-bottleneck hops ($2 \le \ell < \ell^*$) perform microblock DF using the inner-code decoders and encoders. More specifically, the transmitter of the ℓ-th hop decodes the message of the k-th microblock using the codebook $\mathbf{X}_k^{(\ell-1)}$ into an estimate $\widehat{i}_k^{(\ell-1)}$ and forwards the re-encoded codeword $\mathbf{x}_{k,\widehat{i}_k^{(\ell-1)}}^{(\ell)} \in \mathbf{X}_k^{(\ell)}$ to the next node, which is similar to the open loop setting in (30) and (31) After forwarding K microblocks, each of the pre-bottleneck nodes enters the idle state.

Operation of the bottleneck transmitter. While relaying the first K microblocks, the bottleneck transmitter mirrors the behavior of the transmitters of the pre-bottleneck hops. However, after forwarding K microblocks through the bottleneck hop, the operation of this transmitter starts to differ.

After receiving the K-th microblock, the bottleneck transmitter checks whether its past K estimates of the microblock messages $\{\widehat{i}_k^{(\ell^*-1)}\}_{k=1}^K$ correspond to a unique codeword $\widetilde{m}$ from the message set. Namely, using the SRPOC outer code, the bottleneck transmitter computes

$$\widetilde{\mathcal{M}}_{\ell^*-1} \triangleq \left\{ m : \widehat{i}_k^{(\ell^*-1)} = i_k^{[m]} \ \forall k \in [1, K] \right\}. \tag{53}$$

If the set contains a single message, i.e., $\widetilde{\mathcal{M}}_{\ell^*-1} = \{\widetilde{m}\}$, the bottleneck transmitter effectively subsumes the role of the source and continues to use SRPOC outer code to transmit microblocks of the estimated message $\widetilde{m}$ until the stop-feedback from the destination or until it has reached the maximum number of transmissions $k = K_{\mathsf{max}}$. That is, it sends $\mathbf{x}_{k,i_k^{[\widetilde{m}]}}^{(\ell^*)} \in \mathbf{X}_k^{(\ell)}$ for $k \in [K+1, K_{\mathsf{max}}]$ until EOT.

If $\left|\widetilde{\mathcal{M}}_{\ell^*-1}\right| \neq 1$, i.e., either no such $\widetilde{m}$ can be found or multiple $\widetilde{m}$ compatible to the inner code decoded codewords, the transmitter declares an error event[13] and enters the idle state. This mechanism is illustrated in Fig. 5, where the bottleneck transmitter uses its message estimate $\widetilde{m}$ to produce additional microblocks.

Operation of the bottleneck receiver. The operation of the bottleneck receiver consists of two distinct phases: the sequential learning phase and the correction phase. We describe both phases separately below.

1) The sequential learning phase. During the sequential learning phase, the bottleneck receiver computes a running log-likelihood ratio (LLR) for each of the messages in the message set $\mathcal{M}$. When any of the ratios exceeds a certain threshold, it switches to the correction phase. More precisely, recall that $\vec{Y}_{\ell^*}[k]$ denotes the Δ-dimensional received signals of the k-th microblock and let $\left[\vec{Y}_{\ell^*}\right]_1^k = \left\{\vec{Y}_{\ell^*}[j+\ell^*-1] : j \in [1,k]\right\}$ denote the vector of all observed channel outputs at the bottleneck receiver after the receiving the k-th microblock. Then, for every message $m \in \mathcal{M} = [1, e^{K\Delta R}]$, the bottleneck receiver computes the LLR

$$Z_m(k) = \ln\left(\frac{\Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m\right)}{\sum_{m' \neq m} \Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m'\right)}\right), \tag{54}$$

where the conditional probability functions are computed by

$$\Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m\right) \triangleq \Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| \mathbf{c}_{\ell^*}^{[m]}\right) \tag{55}$$

$$= \prod_{i=1}^{k\Delta} P_{\ell^*}(y_i | c_i) \tag{56}$$

and

$$\mathbf{c}_{\ell^*}^{[m]} = \left(\mathbf{x}_{1,i_1^{[m]}}^{(\ell^*)}, \mathbf{x}_{2,i_2^{[m]}}^{(\ell^*)}, \dots\right). \tag{57}$$

That is, the conditional probability function is based on the assumption that the sequence of transmitted channel symbols (over the bottleneck hop) is produced directly by the message m using the joint outer/inner encoder. Note that this assumption is false since the symbols transmitted over the ℓ^*-th hop are based either on the inner codeword estimate $\widehat{i}_k^{(\ell^*-1)}$ of the previous hop (for those $k \in [1, K]$) without the help of the outer codebook, or on the re-encoded versions based on $\widetilde{m}$ (for those $k \in [K+1, K_{\mathsf{max}}]$), cf. the discussion around (53). Hence (55) should be viewed as a way of computing the "score" of each message m, not the actual likelihood of receiving m.

[12]Due to the lack of a special "idle" symbol in the channel input alphabet $\mathcal{X}_\ell$, in the subsequent discussion, we assume that a node in the idle state arbitrarily picks one symbol from its channel input alphabet and transmits it repeatedly until the EOT feedback.

[13]Note that the transmitter does not need to send any "error flags" to the downstream nodes and only needs to idle. The error-event declaration is only used in the analysis.

Using this running LLR value for each k, the bottleneck receiver performs what is essentially a sequential probability ratio test (SPRT) [85] to determine the end of the phase and an estimate of the current message. In other words, the sequential learning phase ends whenever there exists one m such that

$$Z_m(k) > \ln\left(\frac{1}{\eta}\right), \tag{58}$$

where η is the target error probability parameter defined in (51). Since $\log\left(\frac{1}{\eta}\right) > 0$ per our choice of η, by (54) we can have at most one m satisfying (58) at any given k. Namely, if there exists any m satisfying (58), such m must be unique.

If no m satisfying (58) after K_{max} microblocks, we declare a "sequential learning failure", terminate the sequential learning phase, and give up transmission.

In addition to performing the described probability ratio test, during the sequential learning phase, the bottleneck receiver, (i.e., the transmitter of the (ℓ^*+1)-th hop) continues to relay microblocks using the familiar microblock DF scheme. However, it uses the codebooks $\mathring{\mathbf{X}}_k^{(\ell^*+1)}$ defined earlier, which have the same microblock length Δ but contain $e^{\Delta R_I}+K+1$ codewords. The additional $K+1$ codewords represent the value of the correction flag Ξ and are not used at all during the sequential learning phase.

Other than using a slightly modified codebook, the microblock DF operation for the (ℓ^*+1)-th hop (performed by the bottleneck receiver) mirrors the operation of the pre-bottleneck hops: The k-th microblock is mapped to an estimate of the microblock message $\widehat{i}_k^{(\ell^*)}$ using ML decoding and then used to generate the microblock codeword $\mathring{\mathbf{x}}_{k,\widehat{i}_k^{(\ell^*)}}^{(\ell^*+1)} \in \mathring{\mathbf{X}}_k^{(\ell^*+1)}$, which is forwarded to the next node.

2) The correction phase. The description herein assumes that the sequential learning phase is properly terminated, i.e., there exists a unique $\mathring{m}$ satisfying (58). The operations under the forced termination scenario $k = K_{\mathsf{max}}$ are inconsequential since they are lumped under the "error event". During the correction phase, the bottleneck receiver transmits up to K additional microblocks that aim to correct the first K microblocks it transmitted. For this, it must first compile the set of microblocks which were forwarded in error. Let $\tilde{k}$ denote the microblock index for which the sequential learning phase ends and let $\mathring{m}$ denote the message which triggered the end of the sequential learning phase. Next, let $\mathbf{i}^{[\mathring{m}]}$ denote the corresponding sequence of microblock messages generated by $\mathring{m}$ for the first K microblocks, i.e.,

$$\mathbf{i}^{[\mathring{m}]} = \left\{ i_1^{[\mathring{m}]}, \ldots, i_K^{[\mathring{m}]} \right\}. \tag{59}$$

Since the bottleneck receiver observes the transmission over the ℓ^*-th hop, we let $\widehat{\mathbf{i}}^{(\ell^*)}$ denote the sequence of microblock messages that were decoded (using only the inner code decoder of the ℓ^*-th hop) and forwarded to the (ℓ^*+1)-th hop during the sequential learning phase, i.e.,

$$\widehat{\mathbf{i}}^{(\ell^*)} = \left\{ \widehat{i}_1^{(\ell^*)}, \ldots, \widehat{i}_K^{(\ell^*)} \right\}. \tag{60}$$

We call $\mathring{m}$ the *high-confidence estimate* since it is based on the sequential log-likelihood ratio test that jointly considers the inner and outer codes while (60) is of (relatively) low confidence due to the use of only the inner decoders. Finally, denote the set of microblock messages forwarded *in error* as

$$\mathcal{K}_{\mathrm{err}} = \left\{ k \in [1,K] : \widehat{i}_k^{(\ell^*)} \neq i_k^{[\mathring{m}]} \right\}. \tag{61}$$

The transmission of the correction microblocks during the correction phase starts immediately after the $\tilde{k}$-th microblock is received and $\mathcal{K}_{\mathrm{err}}$ is computed. As described earlier, to correct a single microblock, the bottleneck receiver needs to send (i) a correction flag $\Xi = |\mathcal{K}_{\mathrm{err}}|$ that instructs the downstream nodes to expect $|\mathcal{K}_{\mathrm{err}}|$ additional correction microblocks; (ii) the new/corrected content of the microblock message; and (iii) the index of the microblock that is being corrected. As a result, after $\tilde{k}$ microblock transmissions from $\mathring{\mathbf{X}}_k^{\ell^*+1} : k \in [1,\tilde{k}]$, the next microblock $k = \tilde{k}+1$ will still be chosen from the inner codebook $\mathring{\mathbf{X}}_{\tilde{k}+1}^{(\ell^*+1)}$ but this time we transmit one of the additional $(K+1)$ codewords of $\mathring{\mathbf{X}}_{\tilde{k}+1}^{(\ell^*+1)}$ that indicates the value of $\Xi \in [0,K]$.

After transmitting the Ξ value, the bottleneck receiver picks one microblock index j from $\mathcal{K}_{\mathrm{err}}$ that has not been "corrected" and maps the index/payload pair $(j, i_j^{[\mathring{m}]})$ to the corresponding codeword from $\widetilde{\mathbf{X}}_k^{(\ell^*+1)}$. It then repeats this process until all $k \in \mathcal{K}_{\mathrm{err}}$ have been "corrected". Effectively, the correction phase consumes $\Delta + |\mathcal{K}_{\mathrm{err}}|\widetilde{\Delta}$ slots, where Δ of which are used to transmit the flag Ξ and $|\mathcal{K}_{\mathrm{err}}|\widetilde{\Delta}$ of which are used to correct all microblocks in $\mathcal{K}_{\mathrm{err}}$.

Once all microblocks in $\mathcal{K}_{\mathrm{err}}$ have been corrected, the bottleneck receiver remains idle. This mechanism is illustrated in Fig. 5, where two correction microblocks are displayed.

Relaying at the post-bottleneck nodes. The operation of the post-bottleneck nodes mostly mirrors that of the pre-bottleneck nodes. That is, DF is performed on all microblocks. The main difference is, as described, how to coordinate the use of the two codebooks of different sizes. Specifically, the post-bottleneck receivers initially perform ML decoding for every Δ received channel symbols based on the length-Δ inner decoders of $\mathring{\mathbf{X}}_k^{(\ell)}$ for each k. However, once its ML decoder outputs a correction flag $\Xi \in \{0,\cdots,K\}$, say it decodes $\Xi = j$, the post-bottleneck receivers then anticipate the next j microblocks to be based on the length-$\widetilde{\Delta}$ inner codebooks $\widetilde{\mathbf{X}}_k^{(\ell)}$ and thus use the inner codebooks $\widetilde{\mathbf{X}}_k^{(\ell)}$ to perform DF (instead of the pre-Ξ codebooks $\mathring{\mathbf{X}}_k^{(\ell)}$).

Decoding at the destination. The destination operates in the same way as the other post-bottleneck receivers with some minor additions. That is, after its ML decoder $\mathring{\mathbf{X}}_k^{(L)}$ outputs a correction flag $\Xi = j$, the destination starts anticipating the final j correction microblocks and will decode them based on the length-$\widetilde{\Delta}$ inner decoder $\widetilde{\mathbf{X}}_k^{(L)}$. After the final j correction microblocks, it sends the stop-feedback signal through the feedback channel and the entire network stops transmission.

To decode the original message m, the destination takes the very first K microblock messages it previously decoded using the inner code ML decoders, and then replaces a subset of those estimates by the final j correction microblocks received after receiving $\Xi = j$. Let $\left\{\widetilde{\imath}_k \in [1, e^{\Delta R_I}] : k \in [1,K]\right\}$

denote the resulting K microblock messages after correction/replacement. The destination uses the SRPOC outer code $\boldsymbol{\mathcal{S}}$ to compute the *compatible set*

$$\hat{\mathcal{M}} \triangleq \left\{ m \in \mathcal{M} : \breve{\imath}_k = i_k^{[m]} \right\}. \tag{62}$$

The destination then proceeds as follows. If $\hat{\mathcal{M}}$ contains exactly one message $\hat{m}$, then output such $\hat{m}$. If $\hat{\mathcal{M}}$ contains zero or multiple messages, then declare error.

It is possible that the destination's ML decoder $\mathring{\mathbf{X}}_k^{(L)}$ never outputs a correction flag even after receiving K_{max} microblocks, since the channels are noisy. In this case, the destination aborts the transmission of the entire network once the waiting time for the correction flag exceeds the upper limit K_{max} (unit: microblocks). The upper limit K_{max} is critical to attaining a bounded expected delay since if the scheme runs indefinitely with some non-zero probability, no matter how small the probability is, the expectation is always infinite.

C. $\widetilde{\mathsf{DAF}}(R)$ analysis

Recall that our proposed scheme is defined by the four deterministic parameters (K, Δ, R, α). Analogous to the analysis in Sec. III-D, the second coordinate $\Delta \to \infty$ since we let $T \to \infty$ in (12). The error exponent of our proposed scheme is thus determined by the tuple (K, R, α) and we use the notation $E_\Phi(K, R, \alpha)$. We then have

Lemma 4. *For any $\alpha \in (0.5, 1)$, there exists an $R_0(\alpha) < C$ such that for all $R \in [R_0(\alpha), C)$, we have*

$$E_\Phi(K^*, R, \alpha) \geq (2\alpha - 1) \cdot (C_{\ell^*} - R), \tag{63}$$

where K^* is the largest integer K satisfying

$$\alpha(C_{\ell^*} - R) < \frac{\min_{\ell \neq \ell^*} E_{\mathsf{rc},\ell}\left(C_{\ell^*}\right)}{K} \tag{64}$$

and $E_{\mathsf{rc},\ell}(R)$ is the open-loop random coding error exponent (even though this lemma is analyzing a stop-feedback scheme).

By (63) in Lemma 4 and the definition of $\widetilde{\mathsf{DAF}}(R)$ in (18), the proposed scheme thus achieves $\widetilde{\mathsf{DAF}}(R) \leq \frac{1}{2\alpha - 1}$ for all $R \in [R_0(\alpha), C)$. The proof of Proposition 3 is thus complete.

The proof of Lemma 4 is relegated to Appendix C.

D. The Relationship to Multi-hop Network Protocols

Our construction is highly motivated by existing networking algorithms. For example, the new inner codebooks $\mathring{\mathbf{X}}_k^{(\ell)}$ and $\widetilde{\mathbf{X}}_k^{(\ell)}$ for the post-bottleneck hops can be viewed as adding a small *header* to the regular codebook $\mathbf{X}_k^{(\ell)}$. The main difference between our results and existing networking protocols is that in almost all networking protocols, the smallest data unit is *a packet*. Each packet is assumed to have a sufficiently large number of bits and a packet can be either perfectly received or completely erased during transmission. Under this model, the control overhead of adding a header to each packet is negligible when compared to the length of the payload. Essentially, networking protocols are allowed to use the header to send control messages for free or with very small cost. Under these assumptions, one can design many ingenious solutions to lower the end-to-end delay. Two important designs under this packet-based model are the rateless codes [86] and the BATS codes [87]. Both exhibit significantly better delay performance than simple per-packet DF algorithms.

In a broad sense, our results answer the following question: If the smallest data unit is a symbol (e.g., BPSK or 16QAM in physical-layer wireless communications) and if the control overhead cannot be ignored and all control messages are subject to the same noisy channel model as the regular symbols, can we still harvest significant end-to-end delay benefits for multi-hop transmission similar to what was first demonstrated in the coarser, packet-level network-based designs [86], [87]? The answer is a resounding yes and we show that even with all the control overhead being carefully accounted for in a symbol-level setting, we can still achieve provably optimal asymptotic delay when $R \nearrow C$.

V. Conclusion

A new metric called *Delay Amplification Factor* $\mathsf{DAF}(R)$ is proposed, which benchmarks the multi-hop end-to-end error exponent against the worst single-hop random coding exponent, which allows for an intuitive meaning for numerical comparison as it characterizes the multiplicative increase of the asymptotic delay caused by the multi-hop transmission.

For the open-loop setting, we have shown that by judicially combining (i) the concept of microblock-based relay designs, (ii) concatenated coding for point-to-point channels, and (iii) the ML joint inner/outer decoder, we can approach the optimal $\lim_{R \nearrow C} \mathsf{DAF}(R) = 1$ if the bottleneck hop is the last hop of the line network.

For the one-time stop feedback setting, the scheme presented in Sec. IV lifted this restriction and can provably approach $\lim_{R \nearrow C} \widetilde{\mathsf{DAF}}(R) = 1$ regardless of the position of the bottleneck hop. The design combines several new components: (i) the microblock-based designs; (ii) the use of a rateless code as the outer code; (iii) the sequential probability ratio test at the bottleneck receiver based on the ML joint inner/outer decoder; (iv) the bottleneck receiver sending correction packets; and (v) carefully incorporating the network-layer concepts of the *timer* K_{max} and the correction flag Ξ into the physical-layer code design.

Appendix A
Proof of Lemma 1

We first prove $\lim_{R \nearrow C} \mathsf{DAF}_{\mathrm{DF}}(R) \leq \sum_\ell \frac{C_{\ell^*}}{C_\ell}$. To this end, we will construct explicitly the t_1 to t_L and analyze its performance, assuming random coding for each hop.

For any arbitrarily given $\delta > 0$, we define $\delta_{\ell^*} = \delta$ and for any $\ell \neq \ell^*$, we define

$$\delta_\ell \triangleq \inf \left\{ x > 0 : \frac{E_{\mathsf{rc},\ell}\left(C_\ell - x\right)}{C_\ell - x} \geq 2 \cdot \frac{E_{\mathsf{rc},\ell^*}\left(C_{\ell^*} - \delta\right)}{C_{\ell^*} - \delta} \right\} \tag{65}$$

It is well known that any random coding error exponent must satisfy $E_{\mathsf{rc},\ell}\left(C_\ell - x\right) \to 0$ when $x \to 0$ (including $\ell = \ell^*$) since the error exponent is upper bounded by the sphere

packing error exponent $E_{\mathsf{sp},\ell}(R)$ and the latter converges to 0 when $R \to C_\ell$, see [59]. Therefore, we have $\delta_\ell \to 0$ for all ℓ when $\delta \to 0$. We then choose t_ℓ, the active duration of the ℓ-th hop, as

$$t_\ell = \frac{\ln(|\mathcal{M}|)}{C_\ell - \delta_\ell}, \quad \forall \ell \in [1, L]. \tag{66}$$

With random coding on each hop, the description of the DF scheme is thus complete, once we fix the values of $|\mathcal{M}|$ and δ. We now analyze the $\mathsf{DAF}(R)$ value under this construction.

Since $\delta_\ell \to 0$ if $\delta \to 0$, we have $\ell^{**} \triangleq \arg\max_\ell\{t_\ell\} = \ell^* = \arg\min_\ell\{C_\ell\}$ when δ is sufficiently small. By the discussion in Sec. II-D, $T_{\rm dur} = t_{\ell^{**}} = t_{\ell^*}$. The tuple (T, R, ϵ) of this DF scheme thus satisfies

$$T = \sum_{\ell=1}^{L} t_\ell = \ln(|\mathcal{M}|) \cdot \sum_{\ell=1}^{L} \frac{1}{C_\ell - \delta_\ell} \tag{67}$$

$$R = \frac{\ln(|\mathcal{M}|)}{T_{\rm dur}} = \frac{\ln(|\mathcal{M}|)}{t_{\ell^*}} = C_{\ell^*} - \delta_{\ell^*} = C - \delta \tag{68}$$

$$\epsilon \le \sum_{\ell=1}^{L} \epsilon_\ell \le \sum_{\ell=1}^{L} \exp\left\{-t_\ell E_{\mathsf{rc},\ell}\left(\frac{\ln(|\mathcal{M}|)}{t_\ell}\right)\right\} \tag{69}$$

$$= \sum_{\ell=1}^{L} \exp\left\{-\frac{\ln(|\mathcal{M}|)}{C_\ell - \delta_\ell} E_{\mathsf{rc},\ell}\left(C_\ell - \delta_\ell\right)\right\} \tag{70}$$

$$\le \exp\left\{-\frac{\ln(|\mathcal{M}|)}{C_{\ell^*} - \delta_{\ell^*}} E_{\mathsf{rc},\ell^*}\left(C_{\ell^*} - \delta_{\ell^*}\right)\right\} + (L-1)\exp\left\{-\frac{\ln(|\mathcal{M}|)}{C_{\ell^*} - \delta_{\ell^*}} 2 \cdot E_{\mathsf{rc},\ell^*}\left(C_{\ell^*} - \delta_{\ell^*}\right)\right\} \tag{71}$$

where (67) follows from the construction of t_ℓ; (68) follows from $t_{\ell^{**}} = t_{\ell^*}$; (69) follows from the random coding reliability function; and (71) follows from the construction of δ_ℓ in (65).

Letting $|\mathcal{M}| \to \infty$, by (12) we have

$$E_{\rm DF}(R) = E_{\rm DF}(C_{\ell^*} - \delta_{\ell^*}) \ge \frac{\frac{1}{C_{\ell^*} - \delta_{\ell^*}}}{\sum_{\ell=1}^{L} \frac{1}{C_\ell - \delta_\ell}} \cdot E_{\mathsf{rc},\ell^*}\left(C_{\ell^*} - \delta_{\ell^*}\right). \tag{72}$$

which implies

$$\mathsf{DAF}_{\rm DF}(R) \le \sum_{\ell=1}^{L} \frac{C_{\ell^*} - \delta_{\ell^*}}{C_\ell - \delta_\ell}. \tag{73}$$

Finally, we notice that $R = C - \delta$ in (68). Letting $R \nearrow C$ is equivalent to letting $\delta \to 0$. By (13) and (73), we then have (27).

We now prove that regardless how $(|\mathcal{M}|, t_1, \cdots, t_L)$ is chosen, $\mathsf{DAF}(R)$ is always lower bounded by (26). Suppose a given choice of $(|\mathcal{M}|, t_1, \cdots, t_L)$ attains the tuple (T, R, ϵ). Assuming sufficiently small ϵ, we must have the following inequalities:

$$t_\ell \ge \frac{\ln(|\mathcal{M}|)}{C_\ell}, \quad \forall \ell \in [1, L] \tag{74}$$

$$T = \sum_{\ell=1}^{L} t_\ell \ge t_{\ell^*} + \sum_{\ell \in [1,L]\setminus \ell^*} \frac{\ln(|\mathcal{M}|)}{C_\ell} \tag{75}$$

$$R \triangleq \frac{\ln(|\mathcal{M}|)}{T_{\rm dur}} = \frac{\ln(|\mathcal{M}|)}{\max_\ell t_\ell} \le \frac{\ln(|\mathcal{M}|)}{t_{\ell^*}} \le C \tag{76}$$

$$\epsilon \ge \epsilon_{\ell^*} \ge \exp\left\{-t_{\ell^*} E_{\mathsf{sp},\ell^*}\left(\frac{\ln(|\mathcal{M}|)}{t_{\ell^*}}\right) - o(t_{\ell^*})\right\} \tag{77}$$

$$\ge \exp\left\{-t_{\ell^*} E_{\mathsf{sp},\ell^*}(R) - o(t_{\ell^*})\right\} \tag{78}$$

$$= \exp\left\{-t_{\ell^*} E_{\mathsf{rc},\ell^*}(R) - o(t_{\ell^*})\right\} \tag{79}$$

where we have (74) since in order to achieve small ϵ, the coding rate per hop $\frac{\ln(|\mathcal{M}|)}{t_\ell}$ must be less than the capacity C_ℓ; (75) follows from (74); (76) follows from the definition of $T_{\rm dur}$; (77) follows from the fact that the end-to-end error probability is lower bounded by the error probability of the ℓ^*-th hop, which is lower bounded later by the sphere packing bound; (78) follows from (76) and $E_{\mathsf{sp},\ell^*}(\cdot)$ being non-increasing; and (79) follows from the assumption that R satisfies $E_{\mathsf{sp},\ell^*}(R) = E_{\mathsf{rc},\ell^*}(R)$.

By (79) and (75), we have

$$\frac{-\ln(\epsilon)}{T} \le \frac{t_{\ell^*} E_{\mathsf{rc},\ell^*}(R) + o(t_{\ell^*})}{t_{\ell^*} + \sum_{\ell \in [1,L]\setminus \ell^*} \frac{\ln(|\mathcal{M}|)}{C_\ell}} \iff$$

$$\frac{E_{\mathsf{rc},\ell^*}(R)}{\frac{-\ln(\epsilon)}{T}} \ge \frac{t_{\ell^*} + \sum_{\ell \in [1,L]\setminus \ell^*} \frac{\ln(|\mathcal{M}|)}{C_\ell}}{t_{\ell^*} + o(t_{\ell^*})} \tag{80}$$

$$\ge \frac{1 + \sum_{\ell \in [1,L]\setminus \ell^*} \frac{R}{C_\ell}}{1 + o(1)} \tag{81}$$

where (81) follows from (76); and $o(1)$ goes to 0 when $t_{\ell^*} \to \infty$. By letting $|\mathcal{M}| \to \infty$ while fixing the R value, the left-hand side of (80) becomes $\mathsf{DAF}(R)$ and we have thus proven that (26) holds regardless how one chooses the parameters of the DF scheme.

APPENDIX B
PROOF OF LEMMA 3

Recall that $\mathcal{A}$ is the event that at least one inner decoder (among the relays but excluding the destination) is in error. Under the event $\mathcal{A}^c$, all $(L-1)$ upstream hops are error-free. For any non-empty subset $\emptyset \ne S \subseteq \{1, 2, \cdots, K^*\}$, we say *two messages m_1 and m_2 differ by S* if their outer coded vectors $\mathbf{i}^{[m_1]} = \left(i_1^{[m_1]}, i_2^{[m_1]}, \ldots, i_{K^*}^{[m_1]}\right)$ and $\mathbf{i}^{[m_2]} = \left(i_1^{[m_2]}, i_2^{[m_2]}, \ldots, i_{K^*}^{[m_2]}\right)$ satisfy $i_k^{[m_1]} \ne i_k^{[m_2]}$ *if and only if* $k \in S$. Namely, the set S contains the coordinates for which the outer codewords of m_1 and m_2 differ.

For notational simplicity, we use m_0 to denote the message that was actually transmitted. We denote

$$D_S(m_0) \triangleq \{m' : m' \text{ and } m_0 \text{ differ by } S\} \tag{82}$$

as the collection of all possible m' that differs from m_0 by S. By the union bound, we have

$$\Pr\left(\widehat{m} \neq m_0 \middle| \mathcal{A}^c\right) \leq \sum_{\forall S \neq \emptyset} \sum_{\forall m' \in D_S(m_0)} \Pr\left(\widehat{m} = m' \middle| \mathcal{A}^c\right). \tag{83}$$

We now bound each summation

$$\sum_{\forall m' \in D_S(m_0)} \Pr\left(\widehat{m} = m' \middle| \mathcal{A}^c\right).$$

by treating those m' in $D_S(m_0)$ as a (sub) random codebook. Totally, we will have $2^{K^*} - 1$ sub-codebooks since we have $2^{K^*} - 1$ different S to consider

We notice that for all $m' \in D_S(m_0)$, the outer coded vector of m' differs from the outer coded vector of m_0 in and only in the locations of $k \in S$. We now investigate the inner coded codewords of m' and m_0. Specifically, we note that the inner encoder randomly maps each coordinate $i_k^{[m']}$ value of the outer coded vector to the actual *transmitted coded symbol* $\mathbf{x}_{k,i_k^{[m']}}^{(L)}$. Since m' and m_0 differ in the coordinates of S, for those coordinates, the values of the transmitted coded symbols $\mathbf{x}_{k,i_k^{[m']}}^{(L)}$ of m' will be independently and randomly chosen when compared to the transmitted coded symbols $\mathbf{x}_{k,i_k^{[m_0]}}^{(L)}$ of m_0. On the other hand, since m' and m_0 have the same outer coded vector values for those coordinates in $\{1, \cdots, K^*\} \backslash S$, the values of the transmitted coded symbols $\mathbf{x}_{k,i_k^{[m']}}^{(L)}$ of m' will be identical to the transmitted coded symbols $\mathbf{x}_{k,i_k^{[m_0]}}^{(L)}$ of m_0 for the coordinates in $\{1, \cdots, K^*\} \backslash S$.

Because of the above observations, the *effective* codeword length of the random inner code construction of $m' \in D_S(m_0)$, when compared to the inner codeword corresponding to m_0, is reduced from K^* microblocks ($K^*\Delta$ symbols) to $|S|$ microblocks ($|S|\Delta$ symbols) since only those coordinates in S are randomly chosen and are (likely to be) different from m_0.

The next step is to figure out how many m' are in the set $D_S(m_0)$. Namely, what is the number of codewords in the sub-codebook $D_S(m_0)$. Because we use a parity-check outer code, the number of m' within the set $D_S(m_0)$ is upper bounded by

$$|D_S(m_0)| \leq e^{\Delta R_I(|S|-1)} \tag{84}$$

The reason is that each outer-coded symbol is of cardinality $e^{\Delta R_I}$. There are $|S|$ outer coded symbols of m'. However, because the outer coded symbols must satisfy the parity check equation, the *degree of freedom* is reduced from $|S|$ to $|S|-1$. The total number of m' in $D_S(m_0)$ is thus upper bounded by (84). Using the number of codewords and the effective codeword length, the effective code rate in $D_S(m_0)$ is thus upper bounded by

$$\frac{\ln(e^{\Delta R_I(|S|-1)})}{|S|\Delta} = R_I \cdot \frac{|S|-1}{|S|}.$$

Using the effective codeword length $|S|\Delta$ and the effective rate $R_I \cdot \frac{|S|-1}{|S|}$, we can apply the standard random coding error exponent upper bound argument to the sub-codebook $D_S(m_0)$, which implies

$$\sum_{\forall m' \in D_S(m_0)} \Pr\left(\widehat{m} = m' \middle| \mathcal{A}^c\right) \leq e^{-|S|\Delta \cdot E_{\mathsf{rc},L}\left(R_I \cdot \frac{|S|-1}{|S|}\right)} \tag{85}$$

To further upper bound the right-hand side of (85), we notice that the definitions $R_I = \frac{K^*}{K^*-1}R$ and $|S| \leq K^*$ plus some basic algebraic simplification imply the following statement:

$$\begin{aligned} \text{if } R_I \geq C, \text{ then } C - R_I \cdot \frac{|S|-1}{|S|} &\geq \left(C - R_I \frac{K^*-1}{K^*}\right)\frac{K^*}{|S|} \\ &= (C-R)\frac{K^*}{|S|} \end{aligned} \tag{86}$$

Since the K^* value is chosen such that $R_I \geq C$, (86) must hold in our construction.

Since for any $x \leq C$, the error exponent $E_{\mathsf{rc},L}(x)$ is a convex, non-increasing function of x satisfying $E_{\mathsf{rc},L}(C) = 0$, we thus have

$$\frac{E_{\mathsf{rc},L}(C-a) - 0}{a} \geq \frac{E_{\mathsf{rc},L}(C-b) - 0}{b} \tag{87}$$

for all $a \geq b > 0$. We then have

$$E_{\mathsf{rc},L}\left(R_I \cdot \frac{|S|-1}{|S|}\right) \geq E_{\mathsf{rc},L}\left(C - (C-R)\cdot\frac{K^*}{|S|}\right) \tag{88}$$

$$\geq \left(E_{\mathsf{rc},L}\left(C - (C-R)\right)\right)\cdot\frac{K^*}{|S|}. \tag{89}$$

where (88) follows from (86) and from the fact that $E_{\mathsf{rc},L}(x)$ is a non-increasing function of x; and (89) follows from (87) by choosing $a = (C-R)\frac{K^*}{|S|}$ and $b = (C-R)$.

Ineq. (89) is then used to upper bound the right-hand side of (85) by

$$e^{-|S|\Delta \cdot E_{\mathsf{rc},L}\left(R_I \cdot \frac{|S|-1}{|S|}\right)} \leq e^{-K^*\Delta \cdot E_{\mathsf{rc},L}(R)} \tag{90}$$

Combining (83), (85), and (90), we have proven (43) in Lemma 3. The proof is complete.

APPENDIX C
PROOF OF LEMMA 4

We analyze the error probability and the expected end-to-end latency of the scheme in Sec. IV-B in Appendices C-A and C-B, respectively. We then combine them to derive the corresponding error exponent in Appendix C-C.

A. Error probability

In this proof, we denote the message sent by the source as m_0.

Definition 6. *We say that the SRPOC outer code is* reversible *(with respect to m_0) if there exists no other $m \in \mathcal{M}\backslash\{m_0\}$ such that*

$$\left(i_1^{[m]}, i_2^{[m]}, \ldots, i_K^{[m]}\right) = \left(i_1^{[m_0]}, i_2^{[m_0]}, \ldots, i_K^{[m_0]}\right). \tag{91}$$

Namely, by observing the first K coded outer code messages $(i_1^{[m_0]}, i_2^{[m_0]}, \ldots, i_K^{[m_0]})$, we can uniquely recover the original sent message m_0.

Recall that $\tilde{k}$ is defined as the time when the sequential probability test ends (i.e., when (58) holds for some m or when we reach the time limit K_{max}). We can further formalize this discussion by the following definition.

Definition 7. *Define $\tilde{k}_m$ as the microblock index at which the LLR for a specific message m first crosses the threshold in* (58)*, assuming we run the sequential probability test indefinitely without the maximum constraint. As a result, we can rewrite $\tilde{k}$ as*

$$\tilde{k} \triangleq \min\left(K_{\mathsf{max}}, \min_{m\in\mathcal{M}} \tilde{k}_m\right). \tag{92}$$

We then note that the following conditions are sufficient for an error-free end-to-end message transmission.

C1: There are no microblock decoding errors at the pre-bottleneck relays.

C2: The encoding function of the SRPOC $\boldsymbol{\mathcal{S}}$ is reversible. When both C1 and C2 hold, the bottleneck transmitter can correctly decode the message m_0 after K microblocks, successfully subsume the role of the source, and continue to transmit the correct microblocks across the bottleneck hop.

C3: The message decision at the end of the sequential learning phase at the bottleneck receiver is within the hard time limit (i.e., $\min_m \tilde{k}_m \le K_{\max}$) and is correct (i.e., $\mathring{m} = m$).

C4: There is no microblock decoding error over all the post-bottleneck hops.

When C1 to C4 all hold, the destination can recover the first K microblock messages generated at the source correctly either during the sequential learning phase or during the correction phase. Note that since C2 ensures reversibility of the outer SRPOC, when C1 to C4 hold, the destination can recover the message correctly.

We can distill a set of useful error events from these observations. First, let N_{pre} and N_{post} denote the total number of microblock errors over all pre- and post-bottleneck hops, respectively. We then define the following (error) events:

$$\mathcal{A}_1 = \Big\{N_{\mathsf{pre}} > 0\Big\} \tag{93}$$

$$\mathcal{A}_2 = \Big\{\text{the SRPOC is not reversible}\Big\} \tag{94}$$

$$\mathcal{A}_3 = \Big\{\min_m \tilde{k}_m > K_{\max}\Big\} \tag{95}$$

$$\mathcal{A}_4 = \Big\{\text{the output of the sequential LLR test } \mathring{m} \neq m_0\Big\} \tag{96}$$

$$\mathcal{A}_5 = \Big\{N_{\mathsf{post}} > 0\Big\} \tag{97}$$

By mapping the above events to the previous discussion of sufficient conditions C1 to C4, the error probability of our scheme must satisfy

$$\begin{aligned}\epsilon \triangleq \Pr(\widehat{m} \neq m_0) &\le \Pr\left(\bigcup_{i=1}^{5} \mathcal{A}_i\right)\\ &\le \Pr(\mathcal{A}_1) + \Pr(\mathcal{A}_2) + \Pr(\mathcal{A}_3|\mathcal{A}_1^c\mathcal{A}_2^c)\\ &\quad + \Pr(\mathcal{A}_4|\mathcal{A}_1^c\mathcal{A}_2^c\mathcal{A}_3^c) + \Pr(\mathcal{A}_5|\mathcal{A}_1^c\mathcal{A}_2^c)\,,\end{aligned} \tag{98}$$

To continue the analysis, we must bound each of the five terms in (98) separately. We begin with $\mathcal{A}_1$. Using the individual random coding error exponents for the pre-bottleneck hops, we can bound

$$\Pr(\mathcal{A}_1) \le K \sum_{\ell=1}^{\ell^*-1} e^{-\Delta E_{\mathrm{rc},\ell}(R_I)}. \tag{99}$$

From the code construction of the SRPOC $\boldsymbol{\mathcal{S}}$, we know that the probability of the event $\mathcal{A}_2$ satisfies the following union bound.

$$\begin{aligned}\Pr(\mathcal{A}_2) &\le \sum_{m\neq m_0}\prod_{k=1}^{K}\Pr\Big(i_k^{[m]} = i_k^{[m_0]}\Big) &(100)\\ &= \sum_{m\neq m_0}\prod_{k=1}^{K}\left(\frac{\frac{|\mathcal{M}|}{e^{\Delta R_I}} - 1}{|\mathcal{M}| - 1}\right) &(101)\\ &= (|\mathcal{M}|-1)\left(\frac{|\mathcal{M}|e^{-\Delta R_I} - 1}{|\mathcal{M}| - 1}\right)^K &(102)\\ &\le (|\mathcal{M}|-1)e^{-K\Delta R_I} &(103)\\ &\le e^{-K\Delta(R_I - R)}, &(104)\end{aligned}$$

where (101) is due to the uniform random choices of the permutation π_k of $\boldsymbol{\mathcal{S}}$.

To continue, the following lemma provides a bound on the third term of (98).

Lemma 5. *The third term of* (98) *satisfies*

$$\Pr(\mathcal{A}_3|\mathcal{A}_1^c\mathcal{A}_2^c) \le (1+\varsigma_1(K))e^{-K\Delta\alpha(C_{\ell^*}-R)}, \tag{105}$$

where the term $\varsigma_1(K) \to 0$ as $K \to \infty$.

For the proof refer to Appendix D.

For the fourth term of (98), we note that at the moment[14] that the sequential decision rule from (58) reaches the threshold $\ln(1/\eta)$, the conditional error probability is upper bounded by η. Since we chose the target error probability η by (51), we have

$$\Pr(\mathcal{A}_4\,|\mathcal{A}_1^c\mathcal{A}_2^c\mathcal{A}_3^c) \le \eta = e^{-K\Delta\alpha(C_{\ell^*}-R)}. \tag{106}$$

Next, the following lemma provides a bound on the final term of (98).

Lemma 6. *The fifth term of* (98) *satisfies*

$$\Pr(\mathcal{A}_5\,|\mathcal{A}_1^c\mathcal{A}_2^c) \le K\cdot(2+\varsigma_2(K))\sum_{\ell=\ell^*+1}^{L} e^{-\Delta E_{\mathrm{rc},\ell}(R_I)}, \tag{107}$$

where the term $\varsigma_2(K) \to 0$ as $K \to \infty$.

For the proof refer to Appendix G.

We now make the following observations.

1) Due to (64) and (99), we have

$$\Pr(\mathcal{A}_1) \le e^{-K\Delta\alpha(C_{\ell^*}-R)} \tag{108}$$

[14]It needs to be carefully argued whether there is a non-zero probability that the sequential value in (58) never reaches the threshold, i.e., $\Pr\Big(\min_m \tilde{k}_m = \infty\Big) > 0$. That is why we upper bound $\Pr\big(\mathcal{A}_3|\,\mathcal{A}_1^c\mathcal{A}_2^c\big)$ separately from $\Pr\big(\mathcal{A}_4|\,\mathcal{A}_1^c\mathcal{A}_2^c\mathcal{A}_3^c\big)$.

and

$$\Pr(\mathcal{A}_5|\mathcal{A}_1^c\mathcal{A}_2^c) \leq e^{-K\Delta\alpha(C_{\ell^*}-R)} \tag{109}$$

for sufficiently large Δ.

2) Due to (49) and (104), we have

$$\Pr(\mathcal{A}_2) \leq e^{-K\Delta\alpha(C_{\ell^*}-R)} \tag{110}$$

for sufficiently large Δ.

In other words, with the selected parameters, all of the terms in (98) decay exponentially with the rate $K\Delta\alpha(C_{\ell^*} - R)$. Combining all terms, we can thus write

$$\epsilon \leq (5 + \varsigma_1(K))e^{-K\Delta\alpha(C_{\ell^*}-R)} \tag{111}$$

for sufficiently large Δ.

B. End-to-end latency

We next derive a bound on the *expected* end-to-end delay T for one message. For this, we will first bound the expectation of the *total duration of message reception* at the destination, denoted by D. This time, measured from the time slot in which the first microblock arrives until the end of the last microblock, does not include the amount of time it takes to relay all microblocks from the source to the destination. As a result, D and T are directly related by $T = D + (L-1)\Delta$, where the incremental term accounts for the time it takes for the first microblock to be relayed to the destination. We then have

Lemma 7. *The expected total duration of message reception satisfies*

$$E\{D\} \leq K\left(1 + \varsigma_3(K,\Delta)\right)\Delta \tag{112}$$

where $\varsigma_3(K,\Delta) \to 0$ if we let $\Delta \to \infty$ and then $K \to \infty$ in this order.

For the proof refer to Appendix H.

Using this result, we can thus write

$$E\{T\} \leq \left[K\left(1 + \varsigma_3(K,\Delta)\right) + (L-1)\right]\Delta. \tag{113}$$

C. Error exponent and $\widetilde{\mathsf{DAF}}(R)$

Using (111) and (113), we can bound the error exponent as

$$\begin{aligned} E_\Phi(K,R,\alpha) &= \lim_{\Delta\to\infty} \frac{-\ln(\epsilon)}{E\{T\}} \\ &\geq \lim_{\Delta\to\infty} \frac{-\ln\left[(5+\varsigma_1(K))e^{-K\Delta\alpha(C_{\ell^*}-R)}\right]}{\left[K\left(1+\varsigma_3(K,\Delta)\right)+(L-1)\right]\Delta} \\ &= \frac{K\cdot\alpha(C_{\ell^*}-R)}{K\left(1+\lim_{\Delta\to\infty}\varsigma_3(K,\Delta)\right)+(L-1)} \end{aligned} \tag{114}$$

where (114) follows from taking the limit of $\Delta \to \infty$ and only focusing on the dominant terms. Define

$$\zeta(K) \triangleq \frac{K}{K\left(1+\lim_{\Delta\to\infty}\varsigma_3(K,\Delta)\right)+(L-1)}. \tag{115}$$

We then have

$$E_\Phi(K,R,\alpha) \geq \zeta(K)\cdot\alpha\cdot(C_{\ell^*}-R). \tag{116}$$

Finally, since $\lim_{K\to\infty}\lim_{\Delta\to\infty}\varsigma_3(K,\Delta) = 0$ and since $K^* \to \infty$ when $R \nearrow C$, we have $\zeta(K^*) \to 1$ when $R \nearrow C$, which implies that we can find a sufficiently large R (still less than C) which satisfies $\zeta(K^*)\cdot\alpha \geq (2\alpha - 1)$. This thus proves the statement of Lemma 4.

APPENDIX D
PROOF OF LEMMA 5

Let $\check{k} = \min_m \tilde{k}_m$, and note that

$$\Pr\left(\mathcal{A}_3\middle|\mathcal{A}_1^c\mathcal{A}_2^c\right) \leq \Pr\left(\check{k} \geq K_{\mathsf{max}}\middle|\mathcal{A}_1^c\mathcal{A}_2^c\right). \tag{117}$$

For notational simplicity, we denote the expectation conditioned on the event $\mathcal{A}_1^c\mathcal{A}_2^c$ as $E_*\left\{\ \cdot\ \right\} \triangleq E\left\{\ \cdot\ \middle|\mathcal{A}_1^c\mathcal{A}_2^c\right\}$. Using Markov's inequality, we then have

$$\Pr\left(\check{k} \geq K_{\mathsf{max}}\middle|\mathcal{A}_1^c\mathcal{A}_2^c\right) \leq \frac{E_*\left\{\check{k}\right\}}{K_{\mathsf{max}}}. \tag{118}$$

To bound $E_*\left\{\check{k}\right\}$, we first note that the definition of $Z_m(k)$ in (54) holds only for the range of $k \geq 1$. To facilitate our discussion, we define $Z_m(0) = \ln\frac{e^{-K\Delta R}}{1-e^{-K\Delta R}} > -K\Delta R$. Namely, before we receive any observations (i.e., $k = 0$), the LLR value is computed using the uniform prior $P(m) = e^{-K\Delta R}$ over $\mathcal{M} = [1, e^{K\Delta R}]$. We then introduce the following Lemmas.

Lemma 8. *There exists a constant B such that $|Z_m(k+1) - Z_m(k)| < \Delta\cdot B$ with probability one regardless of the values of $m \in \mathcal{M}$, $k \geq 0$, and $\Delta \geq 1$.*

For the proof of this lemma, see Appendix E.

Lemma 9. *For sufficiently large but fixed Δ and assuming m_0 is the transmitted message, the random process*

$$Z_{m_0}(k) - k\Delta\left(R + \alpha(C_{\ell^*}-R)\right) \tag{119}$$

is a submartingale with respect to the time index $k \geq 0$.

For the proof of this lemma, see Appendix F.

We then notice that

$$\begin{aligned} &\Pr\Big(Z_{m_0}(j) < \ln\left(1/\eta\right)\Big) \\ &= \Pr\Big(Z_{m_0}(j) < K\Delta\alpha(C_{\ell^*}-R)\Big) \\ &\leq \Pr\Big(Z_{m_0}(j) - j\Delta(R+\alpha(C_{\ell^*}-R)) - Z_{m_0}(0) \\ &\qquad < (K-j)\Delta(R+\alpha(C_{\ell^*}-R))\Big) \qquad (120) \\ &\leq c_1\cdot e^{-jc_2} \qquad (121) \end{aligned}$$

for some positive constants $c_1, c_2 > 0$, where (120) follows from $Z_{m_0}(0) \geq -K\Delta R$, and (121) is by applying Lemma 8 and Azuma's inequality to the submartingale $Z_{m_0}(j) - j\Delta\left(R + \alpha(C_{\ell^*}-R)\right)$ in Lemma 9. We then have

$$\begin{aligned} E_*\left\{\tilde{k}_{m_0}\right\} &= \sum_{j=0}^{\infty}\Pr\Big(\tilde{k}_{m_0} > j\Big) \qquad (122) \\ &\leq \sum_{j=0}^{\infty}\Pr\Big(Z_{m_0}(j) < \ln(1/\eta)\Big) < \infty \qquad (123) \end{aligned}$$

where (122) follows from basic probability equality and (123) follows from the fact that the event $\{\tilde{k}_{m_0} > j\}$ implies that at time j, the probability ratio $Z_{m_0}(j)$ is still less than the threshold $\ln(1/\eta)$. The above inequality implies that the stopping time $\tilde{k}_{m_0}$ has finite expectation. With bounded $E_*\Big\{\tilde{k}_{m_0}\Big\}$ and the globally bounded variation established in Lemma 8, we can then apply Doob's optional stopping theorem and get

$$\begin{aligned} E_*\Big\{Z_m(\tilde{k}_{m_0}) - \tilde{k}_{m_0}\Delta\left(R+\alpha(C_{\ell^*}-R)\right)\Big\} &\geq E_*\{Z_m(0)\} \\ &\geq -K\Delta R \end{aligned}$$

which implies

$$E_*\Big\{\tilde{k}_{m_0}\Big\} \leq \frac{E_*\Big\{Z_m(\tilde{k}_{m_0})\Big\} + K\Delta R}{\Delta(R+\alpha(C_{\ell^*}-R))}. \tag{124}$$

To further upper bound the numerator of (124), we note that

$$\begin{aligned} E_*\Big\{Z_{m_0}(\tilde{k}_{m_0})\Big\} &\leq E_*\Big\{Z_{m_0}(\tilde{k}_{m_0}-1)\Big\} + \Delta\cdot B & (125)\\ &\leq K\Delta\alpha(C_{\ell^*}-R) + \Delta\cdot B, & (126)\end{aligned}$$

where the first inequality is due to Lemma 8 and the second inequality is because at time $k=\tilde{k}_{m_0}-1$, the term $Z_{m_0}(k)$ has not hit the threshold $K\Delta\alpha(C_{\ell^*}-R)$ yet.

We then note that since $\breve{k} \leq \tilde{k}_{m_0}$, we have $E_*\Big\{\breve{k}\Big\} \leq E_*\Big\{\tilde{k}_{m_0}\Big\}$. Then, after combining (124) and (126), we get

$$E_*\Big\{\breve{k}\Big\} \leq K + \frac{B}{R+\alpha(C_{\ell^*}-R)}. \tag{127}$$

Finally, after combining (127) and (118), we get

$$\Pr\Big(\mathcal{A}_3\Big|\mathcal{A}_1^c\mathcal{A}_2^c\Big) \leq (1+\varsigma_1(K))e^{-K\Delta\alpha(C_{\ell^*}-R)}, \tag{128}$$

where the term $\varsigma_1(K) = \frac{B}{K(R+\alpha(C_{\ell^*}-R))}$ goes to zero as $K\to\infty$. ■

APPENDIX E
PROOF OF LEMMA 8

Recall (54) and note that we can write

$$Z_m(k+1) = \ln\left(\frac{\Pr\Big(\Big[\vec{Y}_{\ell^*}\Big]_1^k\Big|m\Big)\cdot\Pr\Big(\vec{Y}_{\ell^*}[k+\ell^*]\Big|\mathbf{x}^{(\ell^*)}_{k+1,i^{[m]}_{k+1}}\Big)}{\sum_{m'\neq m}\Pr\Big(\Big[\vec{Y}_{\ell^*}\Big]_1^k\Big|m'\Big)\cdot\Pr\Big(\vec{Y}_{\ell^*}[k+\ell^*]\Big|\mathbf{x}^{(\ell^*)}_{k+1,i^{[m']}_{k+1}}\Big)}\right), \tag{129}$$

where we recall that $\vec{Y}_{\ell^*}[k+\ell^*]$ denotes the channel output symbols of the $(k+1)$-th microblock at the bottleneck receiver. Next, we define

$$p_{\mathsf{min}} \triangleq \min_{X\in\mathcal{X}_{\ell^*},Y\in\mathcal{Y}_{\ell^*}} P_{\ell^*}(Y|X) \tag{130}$$

and

$$p_{\mathsf{max}} \triangleq \max_{X\in\mathcal{X}_{\ell^*},Y\in\mathcal{Y}_{\ell^*}} P_{\ell^*}(Y|X). \tag{131}$$

Since $\mathcal{X}_{\ell^*}$ and $\mathcal{Y}_{\ell^*}$ are finite and $P_\ell(y|x)>0$ for all $\ell\in[1,L], x\in\mathcal{X}_\ell, y\in\mathcal{Y}_\ell$, we know that p_{min} and p_{max} exist and are >0.

We then notice that for any fixed Δ, for any choice of m, and for any micro-block index k, we have

$$(p_{\mathsf{min}})^\Delta \leq \Pr\Big(\vec{Y}_{\ell^*}[k+\ell^*]\Big|\mathbf{x}^{(\ell^*)}_{k+1,i^{[m]}_{k+1}}\Big) \leq (p_{\mathsf{max}})^\Delta. \tag{132}$$

We thus have

$$\begin{aligned} Z_m(k+1) &\leq \ln\left(\left(\frac{p_{\mathsf{max}}}{p_{\mathsf{min}}}\right)^\Delta\cdot\frac{\Pr\Big(\Big[\vec{Y}_{\ell^*}\Big]_1^k\Big|m\Big)}{\sum_{m'\neq m}\Pr\Big(\Big[\vec{Y}_{\ell^*}\Big]_1^k\Big|m'\Big)}\right) \\ &= \Delta\ln\left(\frac{p_{\mathsf{max}}}{p_{\mathsf{min}}}\right) + Z_m(k) & (133)\end{aligned}$$

and similarly

$$Z_m(k+1) \geq \Delta\ln\left(\frac{p_{\mathsf{min}}}{p_{\mathsf{max}}}\right) + Z_m(k). \tag{134}$$

Applying the bounds (133) and (134) to the absolute difference $|Z_m(k+1)-Z_m(k)|$, we get

$$|Z_m(k+1)-Z_m(k)| \leq \Delta\cdot\ln\left(\frac{p_{\mathsf{max}}}{p_{\mathsf{min}}}\right) \triangleq \Delta\cdot B, \tag{135}$$

which completes the proof. ■

APPENDIX F
PROOF OF LEMMA 9

We first describe the filtration $\mathcal{F}_k$ on which the submartingale is defined. Let $\mathbf{Y}^k \triangleq \Big[\vec{Y}_{\ell^*}\Big]_1^k$ denote the history of all observed channel outputs at the bottleneck receiver after the k-th microblock, let $[\pi]^k$ denote all permutations π_1 to π_k for the outer code, and let $[\mathbf{X}^{(\ell^*)}]^k$ denote all existing inner codebooks until time k. The filtration $\mathcal{F}_k$ is then generated by the tuple $(\mathbf{Y}^k, [\pi]^k, [\mathbf{X}^{(\ell^*)}]^k)$. To prove Lemma 9, we thus have to prove (i) $E\{|Z_{m_0}(k)|\}<\infty$ and (ii)

$$E_*\Big\{Z_{m_0}(k+1)-Z_{m_0}(k)\Big|\mathcal{F}_k\Big\} \geq \Delta\left(R+\alpha(C_{\ell^*}-R)\right). \tag{141}$$

The finite expectation can be quickly proven by iteratively applying Lemma 8. To prove the second statement, let $\mathbf{y}_{k+1} \triangleq \vec{Y}_{\ell^*}[k+\ell^*]$ denote the channel outputs of the bottleneck hop corresponding to only the $(k+1)$-th microblock. Without loss of generality, we assume that the transmitted message m_0 results in $i^{[m_0]}_{k+1}=1$ based on the randomly chosen permutation π_{k+1} from (52). This can be achieved by renaming whatever the output $i^{[m_0]}_{k+1}$ is as the first symbol. For ease of exposition, for the remainder of this section, we drop the explicit mention of the (ℓ^*)-th hop when discussing the microblock codewords, i.e., we will set $\mathbf{x}^{(\ell^*)}_{k,i} \triangleq \mathbf{x}_{k,i}$ for the rest of this section. We continue by writing

$$Z_{m_0}(k+1) = \ln\left(\frac{\Pr\Big(\mathbf{Y}^k\Big|m_0\Big)\cdot\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)}{\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\cdot\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,i^{[m]}_{k+1}}\Big)}\right). \tag{142}$$

$$E_*\Big\{Z_{m_0}(k+1)-Z_{m_0}(k)\Big|\mathbf{x}_{k+1,1},\pi_{k+1},\mathbf{y}_{k+1},\mathcal{F}_k\Big\}$$
$$\geq \ln\left(\frac{\left(\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\right)\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)}{\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\,E_*\Big\{\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,i_{k+1}^{[m]}}\Big)\Big|\mathbf{x}_{k+1,1},\pi_{k+1},\mathbf{y}_{k+1}\Big\}}\right) \tag{136}$$

$$E_*\Big\{Z_{m_0}(k+1)-Z_{m_0}(k)\Big|\mathbf{x}_{k+1,1},\pi_{k+1},\mathbf{y}_{k+1},\mathcal{F}_k\Big\}$$
$$\geq \ln\left(\frac{\left(\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\right)\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)}{\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\left(\mathbf{1}_{\left\{m\in\mathcal{M}_{k+1}^{[m_0]}\right\}}\cdot\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)+\mathbf{1}_{\left\{m\in\mathcal{M}\setminus\mathcal{M}_{k+1}^{[m_0]}\right\}}\cdot\Pr\Big(\mathbf{y}_{k+1}\Big)\right)}\right) \tag{137}$$

$$E_*\Big\{Z_{m_0}(k+1)-Z_{m_0}(k)\Big|\mathbf{x}_{k+1,1},\mathbf{y}_{k+1},\mathcal{F}_k\Big\}$$
$$\geq \ln\left(\frac{\left(\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\right)\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)}{\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\left(\Pr\Big(m\in\mathcal{M}_{k+1}^{[m_0]}\Big)\cdot\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)+\Pr\Big(m\notin\mathcal{M}_{k+1}^{[m_0]}\Big)\cdot\Pr\Big(\mathbf{y}_{k+1}\Big)\right)}\right) \tag{138}$$
$$=\ln\left(\frac{\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)}{\frac{|\mathcal{M}|e^{-\Delta R_I}-1}{|\mathcal{M}|-1}\cdot\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)+\frac{|\mathcal{M}|-|\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}|-1}\cdot\Pr\Big(\mathbf{y}_{k+1}\Big)}\right) \tag{139}$$

$$E_*\Big\{Z_{m_0}(k+1)-Z_{m_0}(k)\Big|\mathcal{F}_k\Big\}$$
$$\geq -\epsilon\cdot\Delta\cdot B+(1-\epsilon)\ln\left(\frac{e^{-\Delta(H(Y|X)+\epsilon)}}{\frac{|\mathcal{M}|e^{-\Delta R_I}-1}{|\mathcal{M}|-1}e^{-\Delta(H(Y|X)-\epsilon)}+\frac{|\mathcal{M}|-|\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}|-1}e^{-\Delta(H(Y)-\epsilon)}}\right) \tag{140}$$

The difference between (142) and (54) thus becomes

$$Z_{m_0}(k+1)-Z_{m_0}(k)=$$
$$\ln\left(\frac{\left(\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\right)\cdot\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,1}\Big)}{\sum_{m\neq m_0}\Pr\Big(\mathbf{Y}^k\Big|m\Big)\cdot\Pr\Big(\mathbf{y}_{k+1}\Big|\mathbf{x}_{k+1,i_{k+1}^{[m]}}\Big)}\right). \tag{143}$$

Since our scheme is based on a concatenated inner/outer code construction, the corresponding analysis, as will be seen, is much more involved than the traditional single random code construction. Specifically, we notice that the above difference depends on the realizations of the following sets of random variables.

(a) With the inner/outer code construction, any message $m\in\mathcal{M}$ will be encoded as $i_{k+1}^{[m]}\in[1,e^{\Delta R_I}]$ based on the randomly chosen permutation π_{k+1}, see the definition in (52). The first random variable to consider is thus the random outer code permutation π_{k+1}.

(b) For any outer code message $i\in[1,e^{\Delta R_I}]$, the corresponding inner codeword $\mathbf{x}_{k,i}$ is chosen randomly. The second set of random variables is the random choices of $\mathbf{x}_{k,i}$ for all $i\in[2,e^{\Delta R_I}]$. Namely, the choices of inner codewords that are not selected by the actual transmitted message m_0. (Recall that we assume $i_{k+1}^{[m_0]}=1$.)

(c) The $(k+1)$-th microblock codeword $\mathbf{x}_{k+1,1}$, i.e., the codeword choice of the transmitted outer code message $i_{k+1}^{[m_0]}$.

(d) The channel output symbols corresponding to the $(k+1)$-th microblock $\mathbf{y}_{k+1}$ when the input codeword is $\mathbf{x}_{k+1,1}$.

In the following, we take a sequence of conditional expectations until we derive the desired inequality in (141).

Our first step is to take the expectation over the randomness

in (b) while conditioning on (a), (c), and (d). Specifically, we note that $\ln\left(\frac{a}{x}\right)$ is a convex function of x and apply Jensen's inequality to (143) while conditioning on (a), (c), and (d). This leads to the bound in (136), which uses the fact that given $\mathbf{x}_{k+1,1}$, π_{k+1}, $\mathbf{y}_{k+1}$, and $\mathcal{F}_k$, the terms $\Pr\left(\mathbf{Y}^k\middle|m\right)$ and $\Pr\left(\mathbf{y}_{k+1}\middle|\mathbf{x}_{k+1,1}\right)$ become deterministic.

To continue, we define the set

$$\mathcal{M}_{k+1}^{[m_0]} = \left\{m \in \mathcal{M} : i_{k+1}^{[m]} = i_{k+1}^{[m_0]=1}\right\}, \tag{144}$$

which contains all outer code messages m that result in the same microblock message $i_{k+1}^{[m]} = 1$ for the $(k+1)$-th microblock as that of the transmitted message m_0. It is clear that $\mathcal{M}_{k+1}^{[m_0]}$ is a function of the permutation π_{k+1}.

Now, continuing from (136), we observe the following. First, for all messages $m \in \mathcal{M}_{k+1}^{[m_0]}$, since the microblock messages are equal, the microblock codewords must be equal too and thus

$$E_*\left\{\Pr\left(\mathbf{y}_{k+1}\middle|\mathbf{x}_{k+1,i_{k+1}^{[m]}}\right)\middle|\mathbf{x}_{k+1,1},\pi_{k+1},\mathbf{y}_{k+1}\right\} = \Pr\left(\mathbf{y}_{k+1}\middle|\mathbf{x}_{k+1,1}\right) \tag{145}$$

for all $m \in \mathcal{M}_{k+1}^{[m_0]}$. Second, for all messages $m \notin \mathcal{M}_{k+1}^{[m_0]}$, taking the expectation over the distribution of the microblock codewords gives the marginal distribution, and

$$E_*\left\{\Pr\left(\mathbf{y}_{k+1}\middle|\mathbf{x}_{k+1,i_{k+1}^{[m]}}\right)\middle|\mathbf{x}_{k+1,1},\pi_{k+1},\mathbf{y}_{k+1}\right\} = \Pr\left(\mathbf{y}_{k+1}\right) \tag{146}$$

for all $m \notin \mathcal{M}_{k+1}^{[m_0]}$. Using these observations, (136) can be rewritten in the equivalent form in (137), where $\mathbf{1}_{\mathcal{A}}$ is the indicator function of the event $\mathcal{A}$.

By further averaging over π_{k+1} and applying Jensen's inequality again to the denominator of (137), we obtain the bound in (138). The term in (138) can be further simplified to the bound in (139) by noting that regardless of the value of $m \neq m_0$, we always have $\Pr\left(m \in \mathcal{M}_{k+1}^{[m_0]}\right) = \frac{|\mathcal{M}|e^{-\Delta R_I}-1}{|\mathcal{M}|-1}$ and $\Pr\left(m \notin \mathcal{M}_{k+1}^{[m_0]}\right) = \frac{|\mathcal{M}|-|\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}|-1}$ due to the uniform random permutation of π_{k+1}.

To further bound (139), we notice that by the asymptotic equipartition property of $\mathbf{y}_{k+1}$ and $\mathbf{x}_{k+1,1}$, for any $\epsilon > 0$, there exists a sufficiently large Δ such that the event

$$\Pr\left(\mathbf{y}_{k+1}\right) < e^{-\Delta(H(Y)-\epsilon)} \text{ and}$$
$$e^{-\Delta(H(Y|X)+\epsilon)} < \Pr\left(\mathbf{y}_{k+1}\middle|\mathbf{x}_{k+1,1}\right) < e^{-\Delta(H(Y|X)-\epsilon)}$$

has probability $\geq 1-\epsilon$. Using this property and Lemma 8, we obtain (140). By upper bounding $\frac{|\mathcal{M}|e^{-\Delta R_I}-1}{|\mathcal{M}|-1} \leq e^{-\Delta R_I}$ and $\frac{|\mathcal{M}|-|\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}|-1} \leq 1$ in the denominator of (140), we have

$$E_*\left\{Z_{m_0}(k+1) - Z_{m_0}(k)\middle|\mathcal{F}_k\right\} \geq -\epsilon\cdot\Delta\cdot B + (1-\epsilon)\ln\left(\frac{1}{e^{-\Delta(R_I-2\epsilon)}+e^{-\Delta(C_{\ell^*}-2\epsilon)}}\right). \tag{147}$$

Now, since our choice of R_I in (49) satisfies $R_I < C_{\ell^*}$ and ϵ can be made arbitrarily small as long as a sufficiently large Δ is used, we can rewrite (147) as

$$E_*\left\{Z_{m_0}(k+1) - Z_{m_0}(k)\middle|\mathcal{F}_k\right\} \geq \Delta\cdot(R_I - \varsigma(\Delta)) \tag{148}$$

where $\varsigma(\Delta) \to 0$ for sufficiently large Δ. Finally, the choice of R_I from (49) always satisfies $R_I > R+\alpha(C_{\ell^*}-R)$. Using a sufficiently large Δ, (148) implies (141), finishing the proof. ■

APPENDIX G
PROOF OF LEMMA 6

In this section, for notational simplicity, we denote the expectation conditioned on the event $\mathcal{A}_1^c\mathcal{A}_2^c$ as $E_*\left\{\cdot\right\} \triangleq E\left\{\cdot\middle|\mathcal{A}_1^c\mathcal{A}_2^c\right\}$.

We now bound the probability of the event $\mathcal{A}_5\,|\mathcal{A}_1^c\mathcal{A}_2^c$. We first notice that the bottleneck receiver will transmit at most $\breve{k} = \min_m \tilde{k}_m$ microblocks of size Δ during the sequential learning phase, 1 microblock of size Δ for the correction flag Ξ, and at most K additional microblocks of size $\tilde{\Delta}$. We then observe that whether a microblock in any of the post-bottleneck hops is in error is determined by the post-bottleneck hop channel realizations and is thus independent of the number of microblocks transmitted by the bottleneck receiver. As a result, by Wald's lemma, we can employ the same union bound argument as in (99) using the expected number of microblock transmissions and obtain

$$\Pr(\mathcal{A}_5\,|\mathcal{A}_1^c\mathcal{A}_2^c) \leq \sum_{\ell=\ell^*+1}^{L}\left(E_*\left\{\breve{k}\right\}+1\right)e^{-\Delta E_{\mathrm{rc},\ell}(R_I)} + Ke^{-\widetilde{\Delta}E_{\mathrm{rc},\ell}(R_I)}. \tag{149}$$

Next, since $\widetilde{\Delta} > \Delta$, we can write

$$Ke^{-\widetilde{\Delta}E_{\mathrm{rc},\ell}(R_I)} \leq Ke^{-\Delta E_{\mathrm{rc},\ell}(R_I)} \tag{150}$$

which, when used in (149), results in

$$\begin{aligned}\Pr(\mathcal{A}_5\,|\mathcal{A}_1^c\mathcal{A}_2^c) &\leq \left(E_*\left\{\breve{k}\right\}+1+K\right)\cdot\sum_{\ell=\ell^*+1}^{L}e^{-\Delta E_{\mathrm{rc},\ell}(R_I)}\\ &\leq K\cdot\left((1+\varsigma_1(K))+\frac{1}{K}+1\right)\cdot\sum_{\ell=\ell^*+1}^{L}e^{-\Delta E_{\mathrm{rc},\ell}(R_I)}\end{aligned} \tag{151}$$

where (151) follows from $E_*\left\{\breve{k}\right\} \leq K\cdot(1+\varsigma_1(K))$ as proven in the discussion around (127). Since $\varsigma_1(K) \to 0$ as $K \to \infty$. This completes the proof. ■

APPENDIX H
PROOF OF LEMMA 7

In this section, for notational simplicity, we denote the expectation conditioned on the event $\mathcal{A}_1^c\mathcal{A}_2^c$ as $E_*\left\{\cdot\right\} \triangleq$

$E\Big\{ \cdot \Big| \mathcal{A}_1^c \mathcal{A}_2^c \Big\}$. We start with the observation that we can write the expected duration as

$$E\{D\} = E\{D \,|\mathcal{A}_1 \cup \mathcal{A}_2\} \Pr(\mathcal{A}_1 \cup \mathcal{A}_2) + E_*\{D\} \Pr(\mathcal{A}_1^c \mathcal{A}_2^c). \tag{152}$$

For the first term in (152), we note that the maximum possible duration of D (unit: slots) is given by

$$D_{\mathsf{max}} = (K_{\mathsf{max}} + 1)\Delta + K\widetilde{\Delta}. \tag{153}$$

We can thus bound this term by

$$E\{D \,|\mathcal{A}_1 \cup \mathcal{A}_2\} \Pr(\mathcal{A}_1 \cup \mathcal{A}_2) \le D_{\mathsf{max}} \left(\Pr(\mathcal{A}_1) + \Pr(\mathcal{A}_2)\right), \tag{154}$$

where, after carefully applying (99), (104), (49), (50), and (64), we note that the right-hand side of (154) decays exponentially as a function of Δ.

For the second term in (152), define D_{ℓ^*} as the combined duration of the learning and correction phase of the bottleneck receiver. We can then bound $E_*\{D\}$ as

$$E_*\{D\} \le E_*\big\{(D - D_{\ell^*})^+\big\} + E_*\{D_{\ell^*}\} \tag{155}$$

and note that $(D - D_{\ell^*})^+ > 0$ implies that the event $\mathcal{A}_5$ must be true, since it is only possible for the bottleneck receiver's duration to be longer than the destination's if any of the post-bottleneck transmissions are in error, which destroys the synchronization between the bottleneck receiver and the destination. In addition, it is also trivially true that $(D - D_{\ell^*})^+ \le D_{\mathsf{max}}$. We thus have

$$E_*\big\{(D - D_{\ell^*})^+\big\} \le D_{\mathsf{max}} P(\mathcal{A}_5 | \mathcal{A}_1^c \mathcal{A}_2^c). \tag{156}$$

Using similar arguments as the ones used to show how (154) decays exponentially as a function of Δ, we can use (151), (49), (50), and (64) to show that the right-hand side of (156) also decays exponentially as a function of Δ.

Combining the results in (152), (154), (155), and (156), we have

$$\begin{aligned} E\{D\} &\le E_*\{D_{\ell^*}\} P(\mathcal{A}_1^c \mathcal{A}_2^c) + \varsigma(\Delta) \\ &\le E_*\{D_{\ell^*}\} + \varsigma(\Delta) \\ &= \left(1 + E_*\Big\{\breve{k}\Big\}\right)\Delta + E_*\big\{|\mathcal{K}_{\mathrm{err}}| \cdot \mathbf{1}_{\mathcal{A}_3^c}\big\}\,\widetilde{\Delta} + \varsigma(\Delta), \end{aligned} \tag{157}$$

where $\varsigma(\Delta)$ signifies a term satisfying $\lim_{\Delta\to\infty} \varsigma(\Delta) = 0$. (157) follows from the fact that the bottleneck receiver first transmits $\tilde{k}+1$ microblocks of length Δ, and if a decision for one message m can be made before time K_{max}, it then sends additional $|\mathcal{K}_{\mathrm{err}}|$ microblocks of length $\widetilde{\Delta}$. Herein $\mathbf{1}_{\mathcal{A}_3^c}$ is the indicator function of the complement of event $\mathcal{A}_3$ in (95). The term $E_*\Big\{\breve{k}\Big\}$ is upper bounded previously in (127). We now upper bound $E_*\{|\mathcal{K}_{\mathrm{err}}|\}$.

First, define K_{true} as the number of erroneous packet indices at the bottleneck receiver when comparing (a) the inner-coded codewords of all microblocks transmitted by the bottleneck receiver, versus (b) the inner-coded codewords of the microblocks generated directly from the true message m_0. We then have[15]

$$\begin{aligned} &E_*\big\{|\mathcal{K}_{\mathrm{err}}| \cdot \mathbf{1}_{\mathcal{A}_3^c}\big\} && (158)\\ &\le E_*\big\{(|\mathcal{K}_{\mathrm{err}}| - K_{\mathsf{true}})^+ \cdot \mathbf{1}_{\mathcal{A}_3^c}\big\} + E_*\{K_{\mathsf{true}}\} && (159) \end{aligned}$$

Note that the event $\big\{(|\mathcal{K}_{\mathrm{err}}| - K_{\mathsf{true}})^+ \cdot \mathbf{1}_{\mathcal{A}_3^c} > 0\big\}$ can happen only if that the sequential learning phase did not recover the actual transmitted message m_0, i.e., the sequentially learned $\mathring{m} \neq m_0$. This implies that the error event $\mathcal{A}_4$ must be true. Combining the fact that $(|\mathcal{K}_{\mathrm{err}}| - K_{\mathsf{true}})^+ \cdot \mathbf{1}_{\mathcal{A}_3^c} \le K$, we have

$$\begin{aligned} &E_*\big\{(|\mathcal{K}_{\mathrm{err}}| - K_{\mathsf{true}})^+ \cdot \mathbf{1}_{\mathcal{A}_3^c}\big\} \\ &= E_*\big\{(|\mathcal{K}_{\mathrm{err}}| - K_{\mathsf{true}})^+ \cdot \mathbf{1}_{\mathcal{A}_3^c} \cdot \mathbf{1}_{\mathcal{A}_4}\big\} \\ &\le K \cdot \Pr\Big(\mathcal{A}_3^c \cap \mathcal{A}_4 \Big| \mathcal{A}_1^c \mathcal{A}_2^c\Big) \\ &\le K \cdot \Pr\Big(\mathcal{A}_4 \Big| \mathcal{A}_1^c \mathcal{A}_2^c \mathcal{A}_3^c\Big) \\ &\le K e^{-K\Delta\alpha(C_{\ell^*} - R)}, \end{aligned} \tag{160}$$

where (160) is due to (106). Furthermore, since K_{true} is computed based on the actual transmitted message m_0, each of the first K microblocks that is in error (i.e., contributing to K_{true}) is an independent event[15] with probability $\le e^{-\Delta E_{\mathsf{rc},\ell^*}(R_I)}$. By the union bound, we have

$$E_*\{K_{\mathsf{true}}\} \le K \cdot e^{-\Delta E_{\mathsf{rc},\ell^*}(R_I)}. \tag{161}$$

Combining these results, we get

$$\begin{aligned} &E_*\{|\mathcal{K}_{\mathrm{err}}|\} \\ &\le K\Big(e^{-K\Delta\alpha(C_{\ell^*} - R)} + e^{-\Delta E_{\mathsf{rc},\ell^*}(R_I)}\Big). \end{aligned} \tag{162}$$

Finally, combining (162), (157), and (127), we have

$$\begin{aligned} &E\{D\} \\ &\le \left[1 + K(1 + \varsigma_1(K))\right]\Delta \\ &\quad + K\left(e^{-K\Delta\alpha(C_{\ell^*} - R)} + e^{-\Delta E_{\mathsf{rc},\ell^*}(R_I)}\right) \\ &\qquad \cdot \left(\Delta + \frac{\ln(K)}{R_I}\right) + \varsigma(\Delta) \\ &= K \cdot \Delta \cdot (1 + \varsigma_3(K, \Delta)), \end{aligned} \tag{163}$$

where

$$\begin{aligned} \varsigma_3(K, \Delta) \triangleq &\frac{1}{K} + \varsigma_1(K) + \left(e^{-K\Delta\alpha(C_{\ell^*} - R)} + e^{-\Delta E_{\mathsf{rc},\ell^*}(R_I)}\right) \\ &\cdot \left(1 + \frac{\ln(K)}{\Delta \cdot R_I}\right) + \frac{\varsigma(\Delta)}{K \cdot \Delta} \end{aligned} \tag{164}$$

Clearly, $\varsigma_3(K, \Delta) \to 0$ if we let $\Delta \to \infty$ and then $K \to \infty$ in this order. The proof is complete. ■

[15]The reason we introduce K_{true} is that K_{true} depends only on the channel realization of the first K microblock transmissions since K_{true} assumes perfect knowledge about m_0. However, $\mathcal{K}_{\mathrm{err}}$ is decided not only by the channel noise of the first K microblock transmissions, but also on the channel noise of the *entire sequential learning phase* (since it is based on the estimate $\mathring{m}$). The introduction of K_{true} thus severs the long-range dependence of $\mathcal{K}_{\mathrm{err}}$.

REFERENCES

[1] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[2] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, Sept. 2016.

[3] Y. Zhang, D. Love, J. Krogmeier, C. Anderson, R. Heath, and D. Buckmaster, "Challenges and opportunities of future rural wireless communications," *IEEE Commun. Magazine*, vol. 59, no. 12, pp. 16–22, December 2021.

[4] C. E. Shannon, "Probability of error for optimal codes in a Gaussian channel," *The Bell System Technical Journal*, vol. 38, no. 3, pp. 611–656, May 1959.

[5] R. G. Gallager, "A simple derivation of the coding theorem and some applications," *IEEE Trans. Inform. Theory*, vol. 11, no. 1, pp. 3–18, 1965.

[6] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels. II," *Information and Control*, vol. 10, no. 5, pp. 522–552, 1967.

[7] G. D. Forney, "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Transactions on Information Theory*, vol. 14, no. 2, pp. 206–220, Mar. 1968.

[8] A. E. Ashikhmin, A. Barg, and S. N. Litsyn, "A new upper bound on the reliability function of the Gaussian channel," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 1945–1961, Sept. 2000.

[9] S. Shamai and I. Sason, "Variations on the gallager bounds, connections, and applications," *IEEE Transactions on Information Theory*, vol. 48, no. 12, pp. 3029–3051, Dec. 2002.

[10] A. Barg and G. D. Forney, "Random codes: minimum distances and error exponents," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2568–2573, Sept. 2002.

[11] V. Y. F. Tan, "On the reliability function of the discrete memoryless relay channel," *IEEE Trans. Inform. Theory*, vol. 61, no. 4, pp. 1550–1573, Apr. 2015.

[12] D. Cao and V. Y. F. Tan, "Exact error and erasure exponents for the asymmetric broadcast channel," *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 865–885, Feb. 2020.

[13] N. Merhav, "A Lagrangedual lower bound to the error exponent of the typical random code," *IEEE Transactions on Information Theory*, pp. 1–1, 2019.

[14] P.-W. Su, Y.-C. Huang, S.-C. Lin, I.-H. Wang, and C.-C. Wang, "Detailed asymptotics of the delay-reliability tradeoff of random linear streaming codes," in *Proc. IEEE Int'l Symp. Inform. Theory*. Taipei, Taiwan, June 2023.

[15] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[16] ——, "Feedback in the non-asymptotic regime," *IEEE Trans. Inform. Theory*, vol. 57, no. 8, pp. 4903–4925, Aug. 2011.

[17] V. Kostina and S. Verdu, "Fixed-length lossy compression in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3309–3338, 2012.

[18] V. Y. F. Tan and O. Kosut, "On the dispersions of three network information theory problems," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 881–903, 2014.

[19] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.

[20] Y. Hu, J. Gross, and A. Schmeink, "On the capacity of relaying with finite blocklength," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1790–1794, 2016.

[21] P.-W. Su, Y.-C. Huang, S.-C. Lin, I.-H. Wang, and C.-C. Wang, "Sequentially mixing randomly arriving packets improves channel dispersion over block-based designs," in *Proc. IEEE Int'l Symp. Inform. Theory*. Espoo, Finland, June 2022.

[22] S.-C. Lin, Y.-C. Lai, Y.-C. Huang, C.-C. Wang, and I.-H. Wang, "Optimal finite-length linear codes and the corresponding channel dispersion for broadcast packet erasure channels with feedback," in *Proc. IEEE Inform. Theory Workshop*. Virtual, Oct. 2021.

[23] O. Teyeb, A. Muhammad, G. Mildh, E. Dahlman, F. Barac, and B. Makki, "Integrated access backhauled networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.

[24] Y. Li, E. Pateromichelakis, N. Vucic, J. Luo, W. Xu, and G. Caire, "Radio resource management considerations for 5G millimeter wave backhaul and access networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 86–92, 2017.

[25] R.-A. Pitaval, O. Tirkkonen, R. Wichman, K. Pajukoski, and E. Lahetkangas, "Full-duplex self-backhauling for small-cell 5G networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 83–89, 2015.

[26] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networksa technology overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, Sept. 2015.

[27] D. R. Brown III, U. Madhow, M. Ni, M. Rebholz, and P. Bidigare, "Distributed reception with hard decision exchanges," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3406–3418, June 2014.

[28] U. Madhow, D. R. Brown, S. Dasgupta, and R. Mudumbai, "Distributed massive MIMO: Algorithms, architectures and concept systems," in *2014 Information Theory and Applications Workshop (ITA)*, 2014, pp. 1–7.

[29] J. Choi, D. J. Love, and T. P. Bidigare, "Coded distributed diversity: A novel distributed reception technique for wireless communication systems," *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1310–1321, Mar. 2015.

[30] E. C. Van Der Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971.

[31] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.

[32] A. E. Gamal and M. Aref, "The capacity of the semideterministic relay channel (corresp.)," *IEEE Transactions on Information Theory*, vol. 28, no. 3, pp. 536–536, 1982.

[33] K. Kobayashi, "Combinatorial structure and capacity of the permuting relay channel," *IEEE Transactions on Information Theory*, vol. 33, no. 6, pp. 813–826, 1987.

[34] P. Vanroose and E. C. van der Meulen, "Uniquely decodable codes for deterministic relay channels," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1203–1212, 1992.

[35] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[36] W. Nam, S. Chung, and Y. H. Lee, "Capacity of the Gaussian two-way relay channel to within $\frac{1}{2}$ bit," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5488–5494, 2010.

[37] W. Kang and S. Ulukus, "Capacity of a class of diamond channels," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4955–4960, 2011.

[38] D. Gunduz, A. Yener, A. Goldsmith, and H. V. Poor, "The multiway relay channel," *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 51–63, 2013.

[39] L. Wang and M. Naghshvar, "On the capacity of the noncausal relay channel," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3554–3564, 2017.

[40] S. H. Lim, K. T. Kim, and Y. Kim, "Distributed decodeforward for relay networks," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4103–4118, 2017.

[41] X. Wu, L. P. Barnes, and A. zgr, "The Capacity of the Relay Channel: Solution to Covers problem in the Gaussian case," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 255–275, 2019.

[42] S.-H. Lee and S.-Y. Chung, "When is compress-and-forward optimal?" in *Information Theory and Applications Workshop (ITA), 2010*. IEEE, 2010, pp. 1–3.

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

[44] S. H. Lim, Y. Kim, A. E. Gamal, and S. Chung, "Noisy network coding," *IEEE Trans. Inform. Theory*, vol. 57, no. 5, pp. 3132–3152, May 2011.

[45] C.-C. Wang, D. J. Love, and D. Ogbe, "Transcoding: A new strategy for relay channels," in *55th Annual Allerton Conference on Communication, Control and Computing*, Oct. 2017.

[46] J. M. Wozencraft and M. Horstein, "Coding for two-way channels," Massachusetts Institute of Technology, Research Laboratory of Electronics, Tech. Rep., 1961.

[47] K. R. Narayanan and G. L. Stuber, "A novel ARQ technique using the turbo coding principle," *IEEE Communications Letters*, vol. 1, no. 2, pp. 49–51, 1997.

[48] S. Sesia, G. Caire, and G. Vivier, "Incremental redundancy hybrid ARQ schemes based on low-density parity-check codes," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1311–1321, 2004.

[49] J. Wang, S. Y. Park, D. J. Love, and M. D. Zoltowski, "Throughput delay tradeoff for wireless multicast using hybrid-ARQ protocols," *IEEE Transactions on Communications*, vol. 58, no. 9, pp. 2741–2751, Sept. 2010.

[50] S. Y. Park, D. Park, and D. J. Love, "On scheduling for multiple-antenna wireless networks using contention-based feedback," *IEEE Transactions on Communications*, vol. 55, no. 6, pp. 1174–1190, June 2007.

[51] M. Agrawal, Z. Chance, D. J. Love, and V. Balakrishnan, "Using channel output feedback to increase throughput in hybrid-ARQ," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6465–6480, 2012.

[52] B. Zhao and M. C. Valenti, "Practical relay networks: a generalization of hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7–18, Jan. 2005.

[53] J.-F. Cheng, "Coding performance of hybrid ARQ schemes," *IEEE Transactions on Communications*, vol. 54, no. 6, pp. 1017–1029, June 2006.

[54] Qinqing Zhang and S. A. Kassam, "Hybrid ARQ with selective combining for fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 867–880, May 1999.

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

[56] W. Huleihel, Y. Polyanskiy, and O. Shayevitz, "Relaying one bit across a tandem of binary-symmetric channels," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2928–2932.

[57] V. Jog and P.-L. Loh, "Teaching and learning in uncertainty," *IEEE Trans. Inform. Theory*, vol. 67, no. 1, pp. 598–615, August 2021.

[58] C.-C. Wang and D. Love, "Optimal learning rate of sending one bit over arbitrary acyclic BISO-channel networks," in *Proc. IEEE Int'l Symp. Inform. Theory*. Taipei, Taiwan, June 2023, 6 pages.

[59] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels. I," *Information and Control*, vol. 10, no. 1, pp. 65 – 103, 1967.

[60] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. Koksal, and N. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Dec. 2017.

[61] C.-H. Tsai and C.-C. Wang, "Unifying AoI minimization and remote estimation – optimal sensor/controller coordination with random two-way delay," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, Feb. 2022, https://doi.org/10.1109/TNET.2021.3111495.

[62] ——, "Age-of-information revisited: Two-way delay and distribution-oblivious online algorithm," in *Proc. IEEE Int'l Symp. Inform. Theory*. Los Angeles, USA, June 2020, 6 pages.

[63] ——, "Jointly minimizing AoI penalty and network cost among coexisting source-destination pairs," in *Proc. IEEE Int'l Symp. Inform. Theory*. Melbourne, Australia, July 2021, 6 pages.

[64] ——, "Distribution-oblivious online algorithms for age-of-information penalty minimization," *IEEE/ACM Trans. Netw.*, 2023, https://doi.org/10.1109/TNET.2022.3230009.

[65] ——, "Unifying AoI minimization and remote estimation — optimal sensor/controller coordination with random two-way delay," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*. Toronto, Canada, July 2020.

[66] C.-C. Wang, "How useful is delayed feedback in AoI minimization — a study on systems with queues in both forward and backward directions," in *Proc. IEEE Int'l Symp. Inform. Theory*. Espoo, Finland, June 2022, 6 pages.

[67] T. Kim, I. Kim, Y. Sun, and Z. Jin, "Physical layer and medium access control design in energy efficient sensor networks: An overview," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 1, pp. 2–15, Dec. 2014.

[68] T. Kim, D. Love, M. Skoglund, and Z.-Y. Jin, "An approach to sensor network throughput enhancement by PHY-aided MAC," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 670–684, Sept. 2015.

[69] C. Chang and C.-C. Wang, "A new capacity-approaching scheme for general 1-to-k broadcast packet erasure channels with ACK/NACK," *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 3000–3025, May 2020.

[70] W. Kuo and C.-C. Wang, "Robust and optimal opportunistic scheduling for downlink two-flow network coding with varying channel quality and rate adaptation," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 465–479, 2017.

[71] J. Han and C.-C. Wang, "General capacity region for the fully connected three-node packet erasure network," *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5503–5523, 2016.

[72] Z. Ahmad, Z. Chance, D. J. Love, and C.-C. Wang, "Concatenated coding using linear schemes for Gaussian broadcast channels with noisy channel output feedback," *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 4576–4590, 2015.

[73] C.-C. Wang and J. Han, "The capacity region of two-receiver multiple-input broadcast packet erasure channels with channel output feedback," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5597–5626, 2014.

[74] C.-C. Wang, "On the capacity of 1-to-k broadcast packet erasure channels with channel output feedback," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 931–956, 2012.

[75] M. Agrawal, D. J. Love, and V. Balakrishnan, "Communicating over filter-and-forward relay networks with channel output feedback," *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1117–1131, 2016.

[76] Z. Chance and D. J. Love, "Concatenated coding for the AWGN channel with noisy feedback," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6633–6649, 2011.

[77] S. Fong, A. Khisti, B. Li, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Optimal streaming erasure codes over the three-node relay network," *IEEE Trans. Inform. Theory*, vol. 66, no. 5, pp. 2696–2712, May 2020.

[78] E. Domanovitz, A. Khisti, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Streaming erasure codes over multi-hop relay network publisher: Ieee," in *2020 IEEE International Symposium on Information Theory (ISIT)*, June 2020.

[79] M. Krishnan, G. Facenda, E. Domanovitz, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "High rate streaming codes over the three-node relay network," in *2021 IEEE Information Theory Workshop (ITW)*, Oct. 2021.

[80] S. Fong and V. Tan, "Achievable rates for Gaussian degraded relay channels with non-vanishing error probabilities," *IEEE Trans. Inform. Theory*, vol. 63, no. 7, pp. 4183–4201, July 2017.

[81] G. Forney, *Concatenated Codes*. Cambridge, MA, USA: MIT Press, 1966.

[82] N. Wen and R. Berry, "Reliability constrained packet-sizing for linear multi-hop wireless networks," in *Proc. IEEE Int'l Symp. Information Theory*, July 2008, pp. 16–20.

[83] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: John Wiley & Sons, Inc., 1968.

[84] C. Thommesen, "Error-correcting capabilities of concatenated codes with MDS outer codes on memoryless channels with maximum-likelihood decoding," *IEEE Trans. Inform. Theory*, vol. 33, no. 5, pp. 632–640, Sept. 1987.

[85] A. Wald, "Sequential tests of statistical hypotheses," *The annals of mathematical statistics*, vol. 16, no. 2, pp. 117–186, 1945.

[86] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Compuer Science (FOCS'02)*. Vancouver, BC, Canada, Nov. 2002, pp. 271–280.

[87] S. Yang and R. Yeung, "Batched sparse codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5322–5346, Sept. 2014.

Dennis Ogbe Dennis O. Ogbe (S'13, M'20) is a member of the Reprogrammable Signal Processing group at NASAs Jet Propulsion Laboratory in Pasadena, CA. Prior to joining JPL, he was a postdoctoral associate in the Bradley Department of Electrical and Computer Engineering at Virginia Tech in Arlington, VA, followed by a stint as a software-defined radio engineer at Lynk in Falls Church, VA. He holds a B.S. in electrical engineering from Tennessee Technological University and a Ph.D. in electrical engineering from Purdue University. His research interests are in the fields of communication theory, signal processing, computer engineering, and their application to aerospace engineering problems.

Chih-Chun Wang Chih-Chun Wang (S'01 - M'05 - SM'12) is a Professor of the School of Electrical and Computer Engineering of Purdue University. He received the B.E. degree in E.E. from National Taiwan University, Taipei, Taiwan in 1999, the M.S. degree in E.E., the Ph.D. degree in E.E. from Princeton University in 2002 and 2005, respectively. He worked in Comtrend Corporation, Taipei, Taiwan, as a design engineer in 2000 and spent the summer of 2004 with Flarion Technologies, New Jersey. In 2005, he held a post-doctoral researcher position in the Department of Electrical Engineering of Princeton University. He joined Purdue University in 2006, and became a Professor in 2017. He is currently a senior member of IEEE and served as an associate editor of IEEE Transactions on Information Theory during 2014 to 2017. He served as the technical co-chair of the 2017 IEEE Information Theory Workshop. His current research interests are in the latency minimization of 5G wireless networks and the corresponding protocol design, information theory for ultra-low latency communications, network coding, and cyber-physical systems. Other research interests of his fall in the general areas of networking, optimal control, information theory, detection theory, and coding theory.

Dr. Wang received the National Science Foundation Faculty Early Career Development (CAREER) Award in 2009.

David J. Love David J. Love (S'98 - M'05 - SM'09 - F'15) received the B.S. (with highest honors), M.S.E., and Ph.D. degrees in electrical engineering from the University of Texas at Austin in 2000, 2002, and 2004, respectively. Since 2004, he has been with the Elmore Family School of Electrical and Computer Engineering at Purdue University, where he is now the Nick Trbovich Professor of Electrical and Computer Engineering. He served as a Senior Editor for IEEE Signal Processing Magazine, Editor for the IEEE Transactions on Communications, Associate Editor for the IEEE Transactions on Signal Processing, and guest editor for special issues of the IEEE Journal on Selected Areas in Communications and the EURASIP Journal on Wireless Communications and Networking. He was a member of the Executive Committee for the National Spectrum Consortium. He holds 32 issued U.S. patents. His research interests are in the design and analysis of broadband wireless communication systems, beyond-5G wireless systems, multiple-input multiple-output (MIMO) communications, millimeter wave wireless, software defined radios and wireless networks, coding theory, and MIMO array processing.

Dr. Love is a Fellow of the American Association for the Advancement of Science (AAAS) and was named a Thomson Reuters Highly Cited Researcher (2014 and 2015). Along with his co-authors, he won best paper awards from the IEEE Communications Society (2016 Stephen O. Rice Prize and 2020 Fred W. Ellersick Prize), the IEEE Signal Processing Society (2015 IEEE Signal Processing Society Best Paper Award), and the IEEE Vehicular Technology Society (2010 Jack Neubauer Memorial Award).