Coded Computation Against Straggling Channel Decoders in the Cloud for Gaussian Channels

Jinwen Shi, Cong Ling
Imperial College London
{jinwen.shi12, c.ling}@imperial.ac.uk

Osvaldo Simeone
King's College London
osvaldo.simeone@kcl.ac.uk

Jörg Kliewer
New Jersey Institute of Technology
jkliewer@njit.edu

***Abstract*—The uplink of a Cloud Radio Access Network (C-RAN) architecture is studied, where decoding in the cloud takes place at distributed decoding processors. To mitigate the impact of straggling decoders in the cloud, the cloud re-encodes the received frames via a linear code before distributing them to the decoding processors, which estimate linear combinations of the codewords. Focusing on Gaussian channels, and assuming the use of lattice codes at the users, we derive the computational rates and frame error probabilities at the cloud. The approach differs from Compute-and-Forward in that the combination of codewords is not caused by the channel but purposefully created in the cloud by encoding the received signals to reduce the decoding delay.**

I. Introduction

A Cloud Radio Access Network (C-RAN) architecture can leverage network function virtualization (NFV) in order to implement baseband functionalities on commercial off-the-shelf (COTS) hardware, such as general purpose servers. An important challenge of this solution is to ensure a prescribed latency performance despite the variability of the servers' runtimes [1].

The problem of straggling processors, that is, processors lagging behind in the execution of a certain function, has been widely studied in the context of distributed computing [2]. Reference [1] demonstrates the effectiveness of decomposing tasks in parallel runnable small jobs over a distributed computing architecture in terms of latency while avoiding overhead.

For distributed computing, it has been recently shown in [3], [4] that parallel processing can be improved by carrying out linear precoding of the data prior to processing, as long as the function to be computed is linear. The key idea is that, by employing a proper linear block code over fractions of size $1/K$ of the original data, a function may be completed as soon as a number of K or more processors have finalized their operation, irrespective of their identity.

The NFV-based C-RAN model considered in this paper is illustrated by Fig. 1. The packets sent by a user in the uplink are received by the remote radio head (RRH) through an additive white Gaussian noise (AWGN) channel and forwarded to a cloud over a RRH-to-cloud link. Decoding is carried out on a distributed architecture consisting of COTS servers $1, \ldots, N$.

We investigate the use of linear coding on the received packets as a means to improve over parallel processing in order to mitigate the impact of straggling decoders at the cloud. The idea was first studied in [5], [6] where the packets are received by the RRH via a binary symmetric channel (BSC). In this paper, we tackle the problem of extending the design and analysis to Gaussian channels.

With Gaussian channels, the use of coding at the cloud yields a set-up that is similar to Compute-and Forward (C&F) for Gaussian relay networks [7]. In that set-up, the relays attempt to decode their received signals into integer linear combinations of codewords, which they then forward to the destinations. The main difference is that, in the C&F set-up, transmitted signals are mixed by the channel, while in our model linear combining is purposefully applied at the cloud on the received signals. Accordingly, in the NFV scenario under study, the linearly combined received packets contain an accumulated noise term (i.e., $\tilde{\mathbf{y}}_i = \sum_{j=1}^{K} a_{ij} (\mathbf{x}_j + \mathbf{z}_j)$, see Fig. 1), while this is not the case in C&F setting (i.e., $\tilde{\mathbf{y}}_i = \sum_{j=1}^{K} a_{ij}\mathbf{x}_j + \mathbf{z}_j$).

The accumulated noise terms (i.e., $\sum_{j=1}^{K} a_{ij}\mathbf{z}_j$) affect the functions of the servers in terms of the following two aspects. First, noise powers are accumulated, which affects the decoding error probability of each individual server compared to the C&F problem. Second, the common terms in the sums $\sum_{j=1}^{K} a_{ij}\mathbf{z}_j$ make the noise terms seen by different servers statistically dependent. To cope with the latter aspect, we adopt the dependency graph of the linear NFV code as introduced in [5]. Based on this, we derive two analytical large-block approximations on the frame error rate (FER) as a function of the decoding latency.

Notation: Let $+$, $\sum$ and $\oplus$, $\bigoplus$ denote addition and summation over reals and finite fields, respectively. Let $\|\mathbf{h}\| \triangleq \sqrt{\sum_{i=1}^{N} |h_i|^2}$ denote the norm of a vector $\mathbf{h}$. $[K]$ denotes the set $\{1, 2, \cdots, K\}$. All logarithms are of base two. Let $\log^+(x) \triangleq \max(\log(x), 0)$. $|\mathcal{F}|$ denote cardinality of $\mathcal{F}$.

II. System Model

As illustrated in Fig. 1, we focus on the uplink of a C-RAN system with a multi-server cloud decoder connected to an RRH via a dedicated fronthaul link. As detailed next, the

This work has been supported in part by the Engineering and Physical Sciences Research Council (EPSRC), the European Research Council (ERC) under the European Union Horizon 2020 research and innovation program (grant agreements 725731), the U.S. NSF through grant CCF-1525629, and the U.S. NSF grant CNS-1526547.

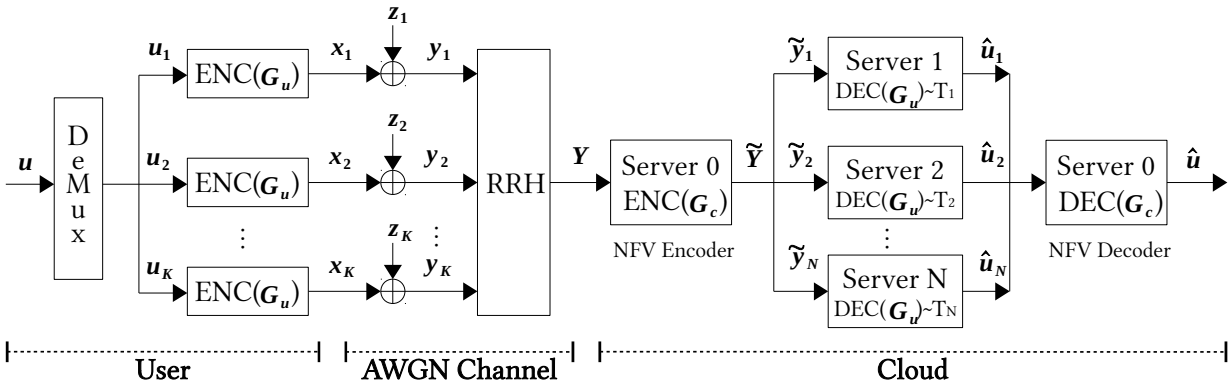

Figure 1. Distributed uplink decoding in C-RAN over an AWGN channel. $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K]$ is a $n \times K$ matrix.

model follows reference [5], but it considers the more realistic AWGN channel for the user-RRH link, requiring a redesign of the operation at the cloud.

The user encodes a file $\mathbf{u}$ of length L over a finite field $\mathbb{F}_p$ for uplink transmission, where $p > 0$ is a prime in $\mathbb{Z}$. Each symbol is drawn independently and uniformly over the finite field. Before encoding, the file is divided into K blocks $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_K$ of equal length $k \triangleq L/K$ symbols. The user's lattice encoder, $\mathcal{E} : \mathbb{F}_p^k \to \mathbb{R}^n$ with generator matrix $\mathbf{G}_u$, then maps each length-k block to a length-n real valued codeword, $\mathbf{x}_j = \mathcal{E}(\mathbf{u}_j)$ (readers are referred to [7] for details). The encoder is subject to the power constraint $\mathbb{E}[\|\mathbf{x}_j\|^2] \leq nP$ where P is the power. The transmission rate R of the user is the length of its message normalized by the number of channel uses, i.e., $R = \frac{k}{n} \log p$.

At the output of the user-RRH AWGN channel, the length-n received packet for the j-th block at the RRH is given as

$$\mathbf{y}_j = \mathbf{x}_j + \mathbf{z}_j, \tag{1}$$

where $\mathbf{z}_j$ is a vector of i.i.d. Gaussian random variables with zero-mean and variance N_0. For convenience, we define the signal-to-noise ratio (SNR) as $\text{SNR} \triangleq P/N_0$. The K packets $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K)$ are transmitted by the RRH to the cloud over a fronthaul link. Decoding is carried out at the cloud.

To this end, the cloud consists of N available servers, namely, Server $1, \ldots, N$, and a master server, i.e., Server 0. Each server can decode a packet within a random time $T_i = T_{1,i} + T_{2,i}$, where times $\{T_1, \ldots, T_N\}$ are mutually independent. Time $T_{1,i}$ accounts for the unavailability of the processor, and is independent of the workload, while $T_{2,i}$ models the execution runtime and it grows as the size n of the packet. The variable $T_{1,i}$ follows an exponential distribution with mean $1/\mu_1$, while $T_{2,i}$ is a shifted exponential with shift equal to $a \geq 0$ and average equal to $a + 1/\mu_2 \times n$ so that $1/\mu_2$ is the time required for an input symbol. The probability that a given set of l out of N servers has finished decoding by time t is given as $\Pr(l, t) = F(t)^l (1 - F(t))^{N-l}$, where $F(t)$ is the cumulative distribution function of T_i.

In order to mitigate the effect of straggling decoders, we adapt the NFV coding scheme in [5] to the AWGN channel. NFV coding operates as follows. The K packets are first linearly encoded by Server 0 into $N \geq K$ coded blocks of the same length n, as depicted in Fig. 1. Each block is then forwarded to a different server in the cloud for decoding. NFV coding uses an (N, K) linear code $\mathcal{C}_c$ with $K \times N$ generator matrix $\mathbf{G}_c \in g(\mathbb{F}_{p'})^{N \times K}$, where $p' > 0$ is a prime and $g(\cdot)$ is the natural map from $\mathbb{F}_{p'}$ to the integers $\{0, 1, 2, \ldots, p' - 1\}$. Note that the prime p' may be different from the prime p used to define the user code. Accordingly, the encoded packets are obtained as

$$\tilde{\mathbf{Y}} = \mathbf{Y}\mathbf{G}_c, \tag{2}$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K]$ is a $n \times K$ matrix, and $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \ldots, \tilde{\mathbf{y}}_N]$ is a $n \times N$ matrix. From (1), the encoded packet $\tilde{\mathbf{y}}_i$ can be written as

$$\tilde{\mathbf{y}}_i = \sum_{j=1}^{K} \mathbf{y}_j g_{c,ji} = \sum_{j=1}^{K} \mathbf{x}_j g_{c,ji} + \sum_{j=1}^{K} \mathbf{z}_j g_{c,ji}, \tag{3}$$

where $g_{c,ji}$ is the (j, i) entry of matrix $\mathbf{G}_c$.

Each server $i \in [N]$ aims at decoding a linear combination of the messages

$$\tilde{\mathbf{u}}_i = \bigoplus_{j=1}^{K} \tilde{g}_{c,ji} \mathbf{u}_j, \tag{4}$$

where $\tilde{g}_{c,ji} = g^{-1}([g_{c,ji}] \bmod p)$ are coefficients taking values in $\mathbb{F}_p$. To this end, Server i is equipped with a decoder, $\mathcal{D}_i : \mathbb{R}^n \to \mathbb{F}_p^k$, that maps the observed output $\tilde{\mathbf{y}}_i$ to an estimate $\hat{\mathbf{u}}_i = \mathcal{D}_i(\tilde{\mathbf{y}}_i)$ of the equation $\tilde{\mathbf{u}}_i$.

Let $d_{\min}$ be the minimum distance of the NFV code $\mathcal{C}_c$. Server 0 is able to decode the message $\mathbf{u}$, or equivalently the K packets $\mathbf{u}_j$ for $j \in [K]$, as soon as $N - d_{\min} + 1$ servers have decoded successfully. The output $\hat{\mathbf{u}}_i(t)$ at the ith Server at time t is $\hat{\mathbf{u}}_i(t) = \hat{\mathbf{u}}_i$, if $T_i \leq t$; and $\hat{\mathbf{u}}_i(t) = \emptyset$, otherwise. The output $\hat{\mathbf{u}}(t)$ of the decoder at Server 0 at time t is a function of $\hat{\mathbf{u}}_i(t)$ for $i \in [N]$. The frame error rate (FER) at time t is defined as

$$P_e^{\text{FER}}(t) = \Pr(\hat{\mathbf{u}}(t) \neq \mathbf{u}). \tag{5}$$

III. Performance Analysis

In order to evaluate the FER, we first derive the computation rate, which gives the maximum rate for each Server i to decode the desired equation $\tilde{\mathbf{u}}_i$ with average probability of error approaching zero when the block n goes to infinity. Based on this auxiliary result, we then employ the error exponent given in [8, Theorems 8-11] to characterize large-block approximations of the FER.

A. Computation Rate

In order to allow servers to decode the desired equations in a manner similar to C&F, we assume that the user adopts a nested lattice code. In this subsubsection, we derive conditions on the NFV code that enable the servers to decode the desired equations when n goes to infinity.

To proceed, the following definitions are useful. An n-dimensional lattice is a discrete subgroup of $\mathbb{R}^n$ which can be described by

$$\Lambda = \{\lambda = \mathbf{B}\mathbf{z} : \ \mathbf{z} \in \mathbb{Z}^n\}, \tag{6}$$

where $\mathbf{B}$ is the full rank generator matrix, assumed to be square for convenience. The Voronoi region $\mathcal{V}$ of a lattice Λ is

$$\mathcal{V} \triangleq \{\mathbf{z} : Q_\Lambda(\mathbf{z}) = \mathbf{0}\}, \tag{7}$$

where $Q_\Lambda(\mathbf{z}) \triangleq \arg\min_{\lambda\in\Lambda} \|\mathbf{z} - \lambda\|$, *i.e.*, the set of all $\mathbf{z}$'s in Euclidean space that are closer to the origin than to any other point of the lattice. Let $\mathrm{Vol}(\mathcal{V})$ denote the volume of $\mathcal{V}$ and $\mathrm{Vol}(\mathcal{V}) = |\det(\mathbf{B})|$ [7]. The second moment of a lattice Λ is defined as

$$\sigma_\Lambda^2 \triangleq \frac{1}{n\mathrm{Vol}(\mathcal{V})} \int_{\mathcal{V}} \|\mathbf{z}\|^2 d\mathbf{z}, \tag{8}$$

and the normalized second moment (NSM) is defined as

$$G(\Lambda) \triangleq \frac{\sigma_\Lambda^2}{(\mathrm{Vol}(\mathcal{V}))^{2/n}}. \tag{9}$$

A lattice Λ is said to be nested in a lattice Λ_f if $\Lambda \subseteq \Lambda_f$. Refer Λ_f as the fine lattice and Λ as the coarse lattice. Set $\sigma_\Lambda^2 = P$.

The following theorem provides a condition on the transmission rate R that guarantees reliable decoding of given equations at the servers.

Theorem 1. *For a given NFV code matrix $\mathbf{G}_c = [\mathbf{g}_{c,1}, \mathbf{g}_{c,2}, \ldots, \mathbf{g}_{c,N}]$ and n large enough, there exists a nested lattice code $\Lambda \subseteq \Lambda_f$ with rate R, such that any Server $i \in [N]$ can recover the linear combination of messages $\tilde{\mathbf{u}}_i$ given in (4) with average probability of error ϵ as long as the inequality*

$$R < \min_{i:g_{c,ji}\neq 0} \frac{1}{2}\log^+\left(\frac{SNR}{\|\mathbf{g}_{c,i}\|^2\left(\alpha_i^2 + SNR(\alpha_i - 1)^2\right)}\right) \tag{10}$$

holds for some choice of parameters $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$.

Proof: See Appendix A. ∎

Based on Theorem 1, we define the computation rate for each Server i as

$$\mathcal{R}^*(\mathbf{g}_{c,i}) = \max_{\alpha_i\in\mathbb{R}} \frac{1}{2}\log^+\left(\frac{\mathrm{SNR}}{\|\mathbf{g}_{c,i}\|^2\left(\alpha_i^2 + \mathrm{SNR}(\alpha_i - 1)^2\right)}\right). \tag{11}$$

By Theorem 1, this is the rate that guarantees correct decoding at Server i.

Theorem 2. *The computation rate (11) is uniquely maximized by choosing α_i to be the minimum mean square error (MMSE) coefficient $\alpha_{MMSE} = \frac{SNR}{1+SNR}$ which results in a computation rate of*

$$\mathcal{R}^*(\mathbf{g}_{c,i}) = \frac{1}{2}\log^+\left(\frac{1+SNR}{\|\mathbf{g}_{c,i}\|^2}\right). \tag{12}$$

Proof: Straightforward and omitted. ∎

B. Large-Blocklength Approximations for the FER

In order to analyze the FER, we need to first evaluate the decoding error probability for each Server i, for $i \in [N]$, as a function of the vector $\mathbf{g}_{c,i}$ defined by the NFV code.

To this end, define the gap to the computation rate as

$$\Delta = \frac{1}{2}\log^+\left(\frac{1+\mathrm{SNR}}{\|\mathbf{g}_{c,i}\|^2}\right) - R, \tag{13}$$

and let $\mu \triangleq 2^{2\Delta}$. Assuming maximum likelihood (ML) decoding, an upper bound on the decoding error probability is given by $P_e^{\mathrm{ML}}(\mathbf{g}_{c,i})$ [8, Theorems 8-11], where

$$P_e^{\mathrm{ML}}(\mathbf{g}_{c,i}) \cong \begin{cases} e^{-nE_r(\mu)}\frac{1}{\sqrt{2\pi n}}, & \mu > 2 \\ e^{-nE_r(\mu)}\frac{1}{\sqrt{8\pi n}}, & \mu = 2 \\ \frac{e^{-nE_r(\mu)}(n\pi)^{-\frac{\mu}{2}}}{(2-\mu)(\mu-1)}, & 2 > \mu > 1, \end{cases} \tag{14}$$

where $a \cong b$ indicates that $\frac{a}{b} \to 1$, and $E_r(\cdot)$ is the Poltyrev random coding exponent defined as [9]

$$E_r(\mu) = \begin{cases} \frac{1}{2}\left[\ln(\mu) + \ln(e/4)\right], & \mu \geq 2 \\ \frac{1}{2}\left[\mu - 1 - \ln(\mu)\right], & 2 \geq \mu \geq 1 \\ 0, & \mu \leq 1. \end{cases} \tag{15}$$

We now assume that the approximation (14) is close to the true probability of error, which is the case for large n, and plug it into the formulas derived in [5] to obtain upper bounds on the FER. More precisely, we use [5, Theorem 1] to obtain a large deviation bound (LDB) and [5, Theorem 2] to obtain a union bound (UB), respectively. Readers are referred to [5] for details.

IV. Numerical Results

In this section, we provide some numerical results to obtain insights into the performance of NFV codes based on the FER bounds presented in the previous section, in terms of the trade-offs between decoding latency and FER. We employ a frame length of $L = 504$ and $N = 8$ servers. The user

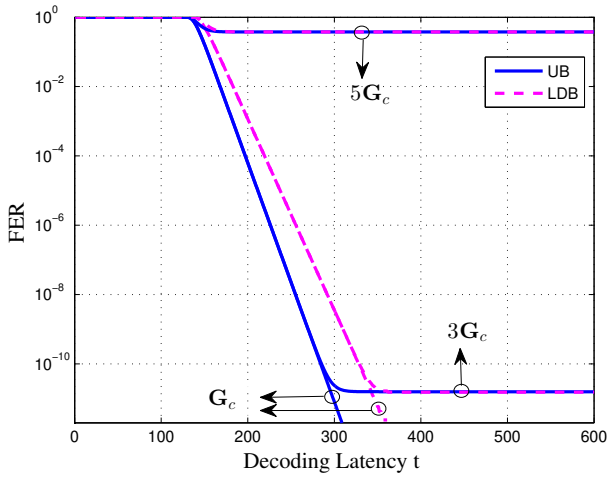

Figure 2. Comparison of LDB and UB based on ML decoding for parallel processing, whose generator matrices are set to be $\mathbf{G}_c \triangleq \mathbf{I}^{N\times N}$, $3\mathbf{G}_c$, and $5\mathbf{G}_c$. ($L = 504$, $N = 8$, $\mu_1 = 50$, $\mu_2 = 10$, $a = 1$, $p = 2$, $p' = \{2, 5, 7\}$, SNR $= 18$ dB)

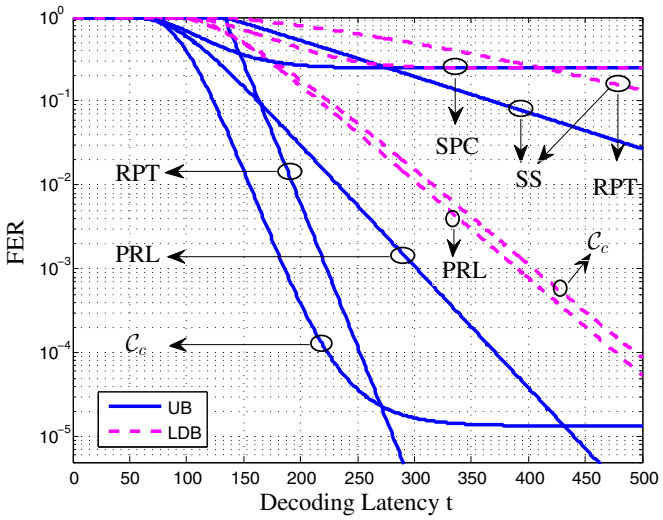

Figure 3. LDB and UB based on ML decoding for single-server decoding (SS), repetition coding (RPT), parallel processing (PRL), single parity-check code (SPC) and the NFV code $\mathcal{C}_c$ defined by $\mathbf{G}_c$ given in [5, Eq. (8)]. ($L = 504$, $N = 8$, $\mu_1 = 1/30$, $\mu_2 = 10$, $a = 0.1$, $p = 2$, $p' = 2$, SNR $= 7$ dB)

code is selected to be binary (i.e., $p = 2$) with rate $R = 0.5$. We set $\mu_1 = 50$, $\mu_2 = 10$, and $a = 1$. Unless stated, otherwise, we have $p' = p = 2$. Furthermore, we leave the performance comparison with simulated results based on specific user lattice codes to future work (see [5] for the case of binary symmetric channels).

We compare the performance of the following solutions: *(i) Single-server (SS) decoding*, where there is a single server $N = 1$ at the cloud that decodes the entire frame ($K = 1$), so that we have $n = 1008$ and $\mathcal{X}(\mathbf{G}_c) = d_{\min} = 1$; *(ii) Repetition coding (RPT)*, where the entire frame is duplicated at all servers, so that we have $n = 1008$ and $\mathcal{X}(\mathbf{G}_c) = d_{\min} = 8$; *(iii) Parallel processing (PRL)*, where the frame is divided into $K = N$ disjoint parts processed by different servers in parallel, and hence we have $n = 126$ and $\mathcal{X}(\mathbf{G}_c) = d_{\min} = 1$; *(iv) Single parity check code (SPC)*, with $K = 7$, where one servers decodes a sum of all other K received packets, and hence we have $n = 144$ and $\mathcal{X}(\mathbf{G}_c) = d_{\min} = 2$; and *(v) an NFV code $\mathcal{C}_c$* with generator matrix $\mathbf{G}_c$ defined in [5, Eq. (8)] which is characterized by $K = 4$, $n = 252$ and $\mathcal{X}(\mathbf{G}_c) = d_{\min} = 3$.

In order to elaborate on the optimal computation rate in Theorem 2, Figure 2 shows the LDB and UB for three parallel coding schemes with generator matrices $\mathbf{G}_c = \mathbf{I}^{N\times N}$, $3\mathbf{G}_c$, and $5\mathbf{G}_c$. Note that all these parallel codes have the same minimum Hamming distance $d_{\min} = 1$ and the same chromatic number $\mathcal{X}(\mathbf{G}_c) = 1$, since the positions of all the non-zeros elements are the same. However, they take entries from different field sizes, e.g., $p' = 2, 5, 7$. Figure 2 confirms the main result in Theorem 2 that, under the same SNR, the NFV codes with larger norms on the column vectors of the generator matrix entails a larger equivalent noise for the server to decode the message equations, causing a larger error floor, and accordingly, a worse trade-off between latency and FER. Larger fields may offer opportunities for the design of more efficient codes, which we leave as an open problem.

To compare different NFV coding schemes, Figure 3 is obtained with parameters $\mu_1 = 1/30$, $\mu_2 = 10$, and $a = 0.1$, in which we consider the case where latency may be dominated by effects that are independent of n, i.e., $\mu_1 = 1/30$. Figure 3 shows both LDB and UB for all the five schemes under SNR $= 7$ dB. As first observation, Figure 3 confirms that UB is tighter than the LDB, and we note that leveraging multiple servers for decoding yields a better trade-off between latency and FER.

Figure 3 shows that, according to the derived upper bounds, the NFV code $\mathcal{C}_c$ provides the smallest FER for a sufficiently small latency level, improving over all schemes including parallel processing. The latter scheme is in fact very sensitive to the unavailability of the servers, requiring all servers to complete decoding, and hence it needs a longer latency in order to achieve a low FER. As for the SPC scheme, although it has an extra parity-check server as compared to parallel processing, its performance is limited by the large equivalent noise determined by its coding matrix. We emphasize that these conclusions are drawn based solely on the derived upper bound, but simulation results for practical codes are expected to show a similar behavior (see [3]).

Appendix A

The user's encoder $\mathcal{E}$ maps its finite field message vector $\mathbf{u}_j$ to a lattice point $\mathbf{t}_j \in \Lambda_f \cap \mathcal{V}$, using the function ϕ from [7, Lemma 5], i.e., $\mathbf{t}_j = \phi(\mathbf{u}_j)$. In order to recover $\tilde{\mathbf{u}}_i$, each Server i needs to decode the lattice equation

$$\mathbf{v}_i = \left[\sum_{j=1}^{K} \mathbf{t}_j g_{c,ji}\right] \mod \Lambda \tag{16}$$

of the lattice points $\mathbf{t}_j$ for $j \in [K]$, where mod is the modulo operation defined as [7]

$$\mathbf{y} = \mathbf{x} \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}). \tag{17}$$

Dither vectors $\mathbf{d}_j$ are generated independently by a uniform distribution over the Voronoi region $\mathcal{V}$ of the coarse lattice Λ. All dither vectors are available at the servers. The user transmits

$$\mathbf{x}_j = [\mathbf{t}_j - \mathbf{d}_j] \bmod \Lambda. \tag{18}$$

By [7, Lemma 7], the vector $\mathbf{x}_j$ is uniform over $\mathcal{V}$, so we have the equality $\mathbb{E}[\|\mathbf{x}_j\|^2] = nP$, where the expectation is over all dithers. Furthermore, it is argued in [7] that there exist fixed dithers that meet the power constraint $\|\mathbf{x}_j\|^2 \leq nP$.

The input of Server $i \in [N]$ is given by (3). Each server computes

$$\mathbf{s}_i = \alpha_i \tilde{\mathbf{y}}_i + \sum_{j=1}^{K} \mathbf{d}_j g_{c,ji}. \tag{19}$$

Let Q_f denote the lattice quantizer for the fine lattice Λ_f. To obtain an estimation of the lattice equation $\mathbf{v}_i$, this vector is quantized onto Λ_f modulo the coarse lattice Λ.

$$\begin{aligned} \hat{\mathbf{v}}_i &= [Q_f(\mathbf{s}_i)] \bmod \Lambda \\ &= [Q_f([\mathbf{s}_i] \bmod \Lambda)] \bmod \Lambda. \end{aligned} \tag{20}$$

The following sequence of equalities shows that $[\mathbf{s}_i] \bmod \Lambda$ is equivalent to $\mathbf{v}_i$ with some added noise terms.

$$\begin{aligned} &[\mathbf{s}_i] \bmod \Lambda \\ &= \left[\sum_{j=1}^{K} g_{c,ji} \left([\mathbf{t}_j - \mathbf{d}_j] \bmod \Lambda + \mathbf{d}_j \right) \right. \\ &\qquad \left. + \sum_{j=1}^{K} g_{c,ji} \left((\alpha_i - 1)\mathbf{x}_j + \alpha_i \mathbf{z}_j \right) \right] \bmod \Lambda \\ &= \left[\sum_{j=1}^{K} g_{c,ji} \mathbf{t}_j + \sum_{j=1}^{K} g_{c,ji} \left((\alpha_i - 1)\mathbf{x}_j + \alpha_i \mathbf{z}_j \right) \right] \bmod \Lambda \\ &= \left[\mathbf{v}_i + \sum_{j=1}^{K} g_{c,ji} \left((\alpha_i - 1)\mathbf{x}_j + \alpha_i \mathbf{z}_j \right) \right] \bmod \Lambda. \end{aligned} \tag{21}$$

By [7, Lemma 7], the pair $(\mathbf{v}_i, \hat{\mathbf{v}}_i)$ has the same joint distribution as the pair $(\mathbf{v}_i, \tilde{\mathbf{v}}_i)$, where $\tilde{\mathbf{v}}_i$ is defined as

$$\tilde{\mathbf{v}}_i \triangleq \left[Q_f\left(\mathbf{v}_i + \mathbf{z}_{\mathbf{eq},i} \right) \right] \bmod \Lambda, \tag{22}$$

where

$$\mathbf{z}_{\mathbf{eq},i} \triangleq \sum_{j=1}^{K} g_{c,ji} \left((\alpha_i - 1)\mathbf{x}_j + \alpha_i \mathbf{z}_j \right), \tag{23}$$

and $\mathbf{x}_j$ is drawn independently and uniformly distributed over $\mathcal{V}$. By [7, Lemma 8], the density of $\mathbf{z}_{\mathbf{eq},i}$ can be upper bounded by an i.i.d. zero-mean Gaussian vector $\mathbf{z}_i^*$ whose variance $\sigma^2_{eq,i}$ approaches

$$N_{eq,i} = \|\mathbf{g}_{c,i}\|^2 N_0 \left(\alpha_i^2 + \text{SNR} (\alpha_i - 1)^2 \right), \tag{24}$$

as $n \to \infty$.

The probability of error $\Pr(\hat{\mathbf{v}}_i \neq \mathbf{v}_i)$ is thus equal to the probability that the equivalent noise leaves the Voronoi region surrounding the codeword, $\Pr\left(\mathbf{z}_{\mathbf{eq},i} \notin \mathcal{V}_f\right)$. Also, we design the fine lattice such that Λ_f satisfies AWGN-goodness [9], which requires that $\epsilon_i = \Pr(\mathbf{z}_i^* \notin \mathcal{V}_f)$ goes to zero exponentially in n as long as the volume-to-noise ratio is such that

$$\mu(\Lambda_f, \epsilon_i) \triangleq \frac{(\text{Vol}(\mathcal{V}_f))^{2/n}}{\sigma^2_{eq,i}} > 2\pi e. \tag{25}$$

Under this condition, $\epsilon_i = \Pr\left(\mathbf{z}_{\mathbf{eq},i} \notin \mathcal{V}_f\right)$ also goes to zero exponentially in n. By the union bound, the average probability of error ϵ is upper bounded by $\epsilon \leq \sum_{i=1}^{N} \Pr\left(\mathbf{z}_{\mathbf{eq},i} \notin \mathcal{V}_f\right)$. To ensure that ϵ_i goes to zero for all desired equations, $\mathcal{V}_f$ must satisfy (25) for all servers with $g_{c,ji} \neq 0$. We set $\mathcal{V}_f$ such that the constraint

$$\text{Vol}(\mathcal{V}_f) > \left(2\pi e \max_{i: g_{c,ji} \neq 0} \sigma^2_{eq,i} \right)^{n/2} \tag{26}$$

is always met.

The rate of a nested lattice code is given by $R = \frac{1}{n} \log \frac{\text{Vol}(\mathcal{V})}{\text{Vol}(\mathcal{V}_f)}$. By (9), we derive $\text{Vol}(\mathcal{V}) = \left(\frac{P}{G(\Lambda)} \right)^{n/2}$. It follows that we can achieve any rates satisfying

$$R < \min_{i: g_{c,ji} \neq 0} \frac{1}{2} \log^+ \left(\frac{P}{G(\Lambda) 2\pi e \sigma^2_{eq,i}} \right). \tag{27}$$

Since Λ satisfies quantization-goodness [10] for n large enough by assumption, we have $G(\Lambda) 2\pi e < (1 + \delta)$ for any $\delta > 0$. Knowing that $\sigma^2_{eq,i}$ converges to $N_{eq,i}$, so for $n \to \infty$, we have $\sigma^2_{eq,i} < (1 + \delta) N_{eq,i}$. Finally, we derive that the rate of the nested lattice code should be at least

$$R < \min_{i: g_{c,ji} \neq 0} \frac{1}{2} \log^+ \left(\frac{P}{N_{eq,i}} \right) - \log(1 + \delta). \tag{28}$$

Therefore, by choosing δ small enough, we can approach the computation rate as close as we desired.

As a result, the servers can make estimates $\hat{\mathbf{v}}_i$ of lattice equations $\mathbf{v}_i$ with coefficient vectors $\mathbf{g}_{c,1}$, $\mathbf{g}_{c,2}$,..., $\mathbf{g}_{c,N} \in g(\mathbb{F}_{p'})^K$ such that $\Pr(\hat{\mathbf{v}}_i \neq \mathbf{v}_i) < \epsilon$ for $\epsilon > 0$ and large n enough as long as

$$R < \min_{i: g_{c,ji} \neq 0} \frac{1}{2} \log^+ \left(\frac{P}{\|\mathbf{g}_{c,i}\|^2 N_0 \left(\alpha_i^2 + \text{SNR} (\alpha_i - 1)^2 \right)} \right) \tag{29}$$

for some $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$. Finally, using ϕ^{-1} from [7, Lemma 6], each server can produce estimates of the desired linear combination of messages $\hat{\mathbf{u}}_i = \phi^{-1}(\hat{\mathbf{v}}_i)$ such that $\Pr\left(\bigcup_{i=1}^{N} \{\hat{\mathbf{u}}_i \neq \tilde{\mathbf{u}}_i\} \right) < \epsilon$ where

$$\tilde{\mathbf{u}}_i = \bigoplus_{j=1}^{K} \tilde{g}_{c,ji} \mathbf{u}_j. \tag{30}$$

REFERENCES

[1] V. Q. Rodriguez and F. Guillemin, "Cloud-RAN modeling based on parallel processing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 457–468, March 2018.

[2] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[3] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, March 2018.

[4] Y. Yang, P. Grover, and S. Kar, "Computing linear transformations with unreliable components," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3729–3756, June 2017.

[5] M. Aliasgari, J. Kliewer and O. Simeone, "Coded computation against processing delays for virtualized cloud-based channel decoding," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 28-38, Jan. 2019.

[6] A. Al-Shuwaili, O. Simeone, J. Kliewer, and P. Popovski, "Coded network function virtualization: Fault tolerance via in-network coding," *IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 644–647, Dec 2016.

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

[8] A. Ingber, R. Zamir, and M. Feder, "Finite-dimensional infinite constellations," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1630–1656, 2013.

[9] G. Poltyrev, "On coding without restictions for the AWGN channel," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 409–417, Mar. 1994.

[10] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Trans. Inf. Theory*, vol. 42, no. 4, pp. 1152–1159, Jul 1996.