Coded Distributed Computing over Packet Erasure Channels

Dong-Jun Han, Jy-yong Sohn, and Jaekyun Moon
School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST)
Email: {djhan93, jysohn1108}@kaist.ac.kr, jmoon@kaist.edu

***Abstract*—Coded computation is a framework which provides redundancy in distributed computing systems to speed up large-scale tasks. Although most existing works assume error-free scenarios, the link failures are common in current wired/wireless networks. In this paper, we consider the straggler problem in distributed computing systems with link failures, by modeling the links between the master node and worker nodes as packet erasure channels. We first analyze the latency in this setting using an (n, k) maximum distance separable (MDS) code. Then, we consider a setup where the number of retransmissions is limited due to the bandwidth constraint. By formulating practical optimization problems related to latency, bandwidth and probability of successful computation, we obtain achievable performance curves as a function of packet erasure probability.**

I. Introduction

Solving massive-scale computational tasks for machine learning algorithms and data analytics is one of the most rewarding challenges in the era of big data. Enabling large-scale computations, distributed computing systems such as MapReduce and Apache Spark have received significant attention in recent years. In distributed computing systems, large-scale computational tasks are divided into several subtasks, which are computed by different workers in parallel. This parallelization helps to reduce the overall run-time to finish the task. However, workers that are significantly slower than the average, called *stragglers*, critically slow down the computing time in practical distributed computing systems.

To alleviate the effect of stragglers, the authors of [1] proposed a new framework called *coded computation*, which uses error correction codes to provide redundancy in distributed matrix multiplications. More recently, coded computation has been applied to various other tasks such as high-dimensional matrix multiplications [2], [3], linear transforms [4], gradient [5], convolution [6] and Fourier transform [7]. The authors of [8] proposed an algorithm for speeding up distributed computing over heterogeneous clusters. In [9], a hierarchical coding scheme is proposed for practical systems with multiple racks. The authors of [10] also considered a setting where the workers are connected wirelessly to the master. In [11], [12], the authors proposed a scheme to exploit the work completed by stragglers, by sequentially allocating of multiple encoded tasks to each worker.

In most of the existing studies, it is assumed that the links between the master node and worker nodes are error-free. However, link failures are common in current wired/wireless networks. It is reported in [13] that the transmitted packets often get lost in current data centers by various factors such as load balancer issues. In wireless networks, the transmitted packets are lost by channel fading. The transmission failure at a specific worker necessitates packet retransmission, which would in turn increase the overall run-time. Moreover, when the bandwidth of the system is limited, the master would not be able to collect timely results from the workers due to link failures. A reliable solution to deal with these problems is required in practical distributed computing systems.

In this paper, we model the links between the master node and worker nodes as packet erasure channels, which is inspired by the link failures in practical systems. We consider a matrix-vector multiplication which is a key computation in many machine learning algorithms. We first separate the run-time at each worker into computation and communication time. This approach is also taken in the previous work [10] on coded computation over wireless networks under the error-free assumption. Compared to the setting in [10], in our work, link failures and retransmissions are considered. Thus, the communication time at each worker becomes a random variable which depends on the packet erasure probability ϵ. We analyze the latency in this setting using an (n, k) MDS code. For maximum k (when each node computes a single inner product), we obtain the overall run-time by analyzing a continuous-time Markov chain. We also find the lower and upper bounds on the latency in closed-forms. Then for general k, we give guidelines to design the MDS code depending on the erasure probability ϵ. Finally, we consider a setup where the number of retransmissions is limited due to the bandwidth constraint. By formulating practical optimization problems related to latency, bandwidth and probability of successful computation, we obtain achievable performance curves as a function of packet erasure probability.

Notations: For n random variables, the k^{th} order statistic is denoted by the k^{th} smallest one of n. We denote the set $\{1, 2, ...n\}$ by $[n]$ for $n \in \mathbb{N}$, where $\mathbb{N}$ is the set of positive integers. We also denote the k^{th} order statistics of n independent random variables $(X_1, X_2, ...X_n)$ by $X_{(k)}$.

II. Problem Statement

Consider a master-worker setup with n worker nodes connected to a single master node. We study a matrix-vector multiplication problem, where the goal is to compute the output $\mathbf{y} = \mathbf{A}\mathbf{x}$ for a given matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$ and an input vector $\mathbf{x} \in \mathbb{R}^{d \times 1}$. The matrix $\mathbf{A}$ is divided into k submatrices

to obtain $\mathbf{A}_i \in \mathbb{R}^{\frac{m}{k} \times d}$, $i \in [k]$. Then, an (n, k) MDS code is applied to construct $\tilde{\mathbf{A}}_i \in \mathbb{R}^{\frac{m}{k} \times d}$, $i \in [n]$, which are distributed across the n worker nodes ($n > k$). For a given input vector $\mathbf{x}$, each worker i computes $\tilde{\mathbf{A}}_i\mathbf{x}$ and sends the result to the master. By receiving k out of n results, the master node can recover the desired output $\mathbf{A}\mathbf{x}$.

We assume that the required time at each worker for computing a single inner product of vectors with size d follows an exponential distribution with rate μ_1. Since $\frac{m}{k}$ inner products are computed at each worker, we model X_i, the computation time of worker i, as an exponential distribution with rate $\frac{k\mu_1}{m}$, i.e., $\Pr[X_i \leq t] = 1 - e^{-\frac{k}{m}\mu_1 t}$. After completing the given task, each worker transmits its results to the master through a memoryless packet erasure channel where the packets are erased independently with probability ϵ. For simplicity, we assume that each packet contains a single inner product but our analysis is applicable to general settings with any packet size. When a packet fails to be detected at the master, the corresponding packet is retransmitted. Otherwise, the worker transmits the next packet to the master. The entire work is finished when the master successfully detects the results of k out of n workers. The communication time for j^{th} transmission of the r^{th} packet of each worker, $Y_{j,r}$, is modeled as an exponential distribution with rate μ_2. Since each worker transmits $\frac{m}{k}$ inner products, the total communication time at worker i, denoted by S_i, is written as

$$S_i = \sum_{r=1}^{\frac{m}{k}} \sum_{j=1}^{N_r} Y_{j,r} \tag{1}$$

where $\Pr[Y_{j,r} \leq t] = 1 - e^{-\mu_2 t}$ and N_r is a geometric random variable with the success probability of $1 - \epsilon$, i.e., $\Pr[N_r = l] = (1-\epsilon)\epsilon^{l-1}$. We define the overall run-time of the given task as the sum of computation and communication time. Now the overall run-time using (n, k) MDS code is written as

$$T = \underset{i \in [n]}{k^{\text{th}}\min}(X_i + S_i). \tag{2}$$

We are interested in the expected overall run-time $\mathbb{E}[T]$.

III. Latency Analysis for $k = m$

We start by reviewing some useful results on the order statistics. The expected value of k^{th} order statistic of n exponential random variables with rate μ is $\frac{H_n - H_{n-k}}{\mu}$, where $H_k = \sum_{i=1}^{k} \frac{1}{i}$. For $k = n$, the k^{th} order statistic becomes $\frac{H_n}{\mu}$. The value H_n can be approximated as $H_n \simeq \log(n) + \eta$ for large n where η is a fixed constant.

In this section, we assume that each worker is capable of computing a single inner product, i.e., $k = m$. This is a realistic assumption especially when a massive number of Internet of Things (IoT) devices having small storage size coexist in the network for wireless distributed computing [14]. The case for general k will be discussed in Section IV.

A. Expected Overall Run-Time: Markov Chain Analysis

We derive the expected overall run-time based on μ_1, μ_2, ϵ, n. We first state the following result on the communication time.

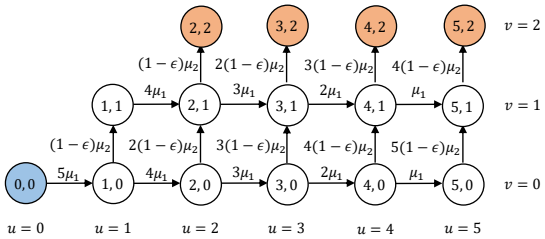


Fig. 1: State transition diagram example with $n = 5$, $k = 2$.

Lemma 1. *Assuming $k = m$, random variable S_i follows an exponential distribution with rate $(1 - \epsilon)\mu_2$.*

Proof: See [15] for the proof. ∎

Based on Lemma 1, we obtain the following theorem which shows that $\mathbb{E}[T]$ can be obtained by analyzing the hitting time of a continuous-time Markov chain.

Theorem 1. *Consider a continuous-time Markov chain $\mathbb{C}$ defined over 2-dimensional state space $(u, v) \in \{0, 1, ..., n\} \times \{0, 1, ..., k\}$, where the transition rates are defined as follows.*

- *The transition rate from state (u, v) to state $(u+1, v)$ is $(n-u)\mu_1$ if $v \leq u < n$, and 0 otherwise.*
- *The transition rate from state (u, v) to state $(u, v+1)$ is $(u-v)(1-\epsilon)\mu_2$ if $0 \leq v < \min(u, k)$, and 0 otherwise.*

Then, the expected hitting time of $\mathbb{C}$ from state $(0, 0)$ to the set of states (u, v) with $v = k$ is equal to $\mathbb{E}[T]$.

Proof: See [15] for the proof. ∎

For an arbitrary state (u, v), the first component u can be viewed as the number of workers that finished the computation, and the second component v is the number of workers that successfully transmitted their results to the master. Fig. 1 shows an example of the transition state diagram for $n = 5$, $k = 2$. From the initial state $(0, 0)$, the task is finished when the Markov chain visits one of the states with $v = 2$ for the first time. Since the results of a worker are transmitted after the worker finishes the whole computation, $u \geq v$ always holds. The expected hitting time of the Markov chain can be obtained by performing the first-step analysis [16].

B. Lower and Upper Bounds

While the exact $\mathbb{E}[T]$ is not derived in a closed-form, we provide the closed-form expressions of the lower and upper bounds on $\mathbb{E}[T]$ in the following theorem.

Theorem 2. *Assuming $k = m$, the expected overall run-time $\mathbb{E}[T]$ is lower bounded as $\mathbb{E}[T] \geq \mathcal{L}$, where*

$$\mathcal{L} = \begin{cases} \dfrac{1}{n\mu_1} + \dfrac{H_n - H_{n-k}}{(1-\epsilon)\mu_2}, & \epsilon \geq 1 - \dfrac{\mu_1}{\mu_2} \\ \dfrac{H_n - H_{n-k}}{\mu_1} + \dfrac{1}{n(1-\epsilon)\mu_2}, & otherwise. \end{cases} \tag{3}$$

Moreover, the expected overall run-time $\mathbb{E}[T]$ is upper bounded as $\mathbb{E}[T] \leq \mathcal{U}$, where

$$\mathcal{U} = \min_{i \in [n] \setminus [k-1]} \Big(\frac{H_n - H_{i-k}}{\mu_1} + \frac{H_n - H_{n-i}}{(1-\epsilon)\mu_2} \Big). \tag{4}$$

Proof: See Section VII. ■

For $\epsilon \geq 1 - \frac{\mu_1}{\mu_2}$, the first term of the lower bound in (3) is the expected value of the minimum computation time among n workers, while the second term is the expected value of the k^{th} smallest communication time among n workers. This is the case where all the workers finish the computation by time $\frac{1}{n\mu_1}$ and then start communication simultaneously, which can be viewed as an obvious lower bound. For the upper bound, by defining $H_0 = 0$, we have $\mathcal{U} = \frac{H_n}{\mu_1} + \frac{H_n - H_{n-k}}{(1-\epsilon)\mu_2}$ by inserting $i = k$ at (4). The first term $\frac{H_n}{\mu_1}$ is the expected value of the maximum computation time among n workers and the second term is the expected value of the k^{th} smallest communication time among n workers. This is an obvious upper bound as it can be interpreted as the case where all n workers starts communication simultaneously right after the slowest worker finishes its computation. Based on the derived bounds, we show in [15] that the overall run-time of the coded scheme becomes $\Theta(\log(n))$ times faster than the uncoded scheme.

IV. Latency for General k

In this section, we give discussions on the expected latency for general k. From Lemma 1, random variable $\sum_{j=1}^{N_r} Y_{j,r}$ follows an exponential distribution with rate $(1-\epsilon)\mu_2$ for all $r \in [\frac{m}{k}]$. Therefore, $S_i = \sum_{r=1}^{\frac{m}{k}} \sum_{j=1}^{N_r} Y_{j,r}$, which is the sum of $\frac{m}{k}$ independent exponential random variables with rate $(1-\epsilon)\mu_2$, follows an erlang distribution with the shape parameter $\frac{m}{k}$ and the rate parameter $(1-\epsilon)\mu_2$, i.e., $\Pr[S_i \leq t] = 1 - \sum_{p=0}^{\frac{m}{k}-1} \frac{1}{p!} e^{-(1-\epsilon)\mu_2 t}[(1-\epsilon)\mu_2 t]^p$. Similar to the proof of Theorem 2, the lower bound $\mathcal{L}$ and the upper bound $\mathcal{U}$ on $\mathbb{E}[T]$ are obtained as follows,

$$\mathcal{L} = \max_{i\in[k]} \left(\frac{m(H_n - H_{n-k+i-1})}{k\mu_1} + \frac{\mathbb{E}[G_{(i)}]}{(1-\epsilon)\mu_2} \right) \tag{5}$$

$$\mathcal{U} = \min_{i\in[n]\setminus[k-1]} \left(\frac{m(H_n - H_{i-k})}{k\mu_1} + \frac{\mathbb{E}[G_{(i)}]}{(1-\epsilon)\mu_2} \right) \tag{6}$$

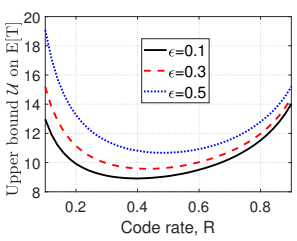
where G_i follows an erlang distribution with the shape parameter $\frac{m}{k}$ and the rate parameter 1. As illustrated in [17], the expected value of the i^{th} order statistic $\mathbb{E}[G_{(i)}]$ can be written as (7), shown at the top of next page. Here, Γ is a gamma function[1] and $a_q(x,y)$ is the coefficient of t^q in the expansion of $(\sum_{j=0}^{x-1} \frac{t^j}{j!})^y$ for any integers x and y. While finding the exact expression of $\mathbb{E}[T]$ for general k remains as an open problem, the optimal code rate R^* can be designed from Fig. 2(a) to minimize the upper bound $\mathcal{U}$ in (6) depending on the erasure probability ϵ. Fig. 2(b) shows that the performance of the MDS code with rate R^* is very close to the achievable $\mathbb{E}[T]$, showing the effectiveness of the proposed design.

V. Analysis under the Setting of Limited Number of Retransmissions

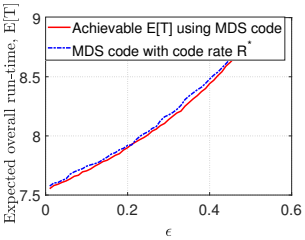
A. Probability of Successful Computation

So far, we focused on the latency analysis assuming that there is no limit on the number of transmissions at each worker. The number of transmissions can be potentially infinite on the analysis above. However, in practice, the number of transmissions can be limited due to the bandwith constraint. With the limited number of transmissions, the overall computation can fail, i.e., the master node may not receive enough results (less than k) from the workers due to link failures.

[1]For a positive integer z, $\Gamma(z) = (z-1)!$ is satisfied.



(a) Upper bound $\mathcal{U}$ versus R.



(b) Expected latency of the MDS code targeting upper bound minimization.

Fig. 2: Simulation results for $n = 100$, $m = 500$, $\mu_1 = 1$, $\mu_2 = 10$.

Thus, we consider the success probability of a given task in this section. Let γ be the maximum number of packet transmissions allowed at each worker. That is, when the number of workers is given by n, the total transmission bandwidth of the network is then limited to $n\gamma l$, where l is the packet length. For a successful computation at a specific worker node, the master node should successfully receive $\frac{m}{k}$ out of γ packets from that worker. The probability of success at a specific worker, denoted by p, is obtained as

$$p = \underbrace{(1-\epsilon)^{\frac{m}{k}}}_{0 \text{ failure}} + \underbrace{\binom{\frac{m}{k}}{1}(1-\epsilon)^{\frac{m}{k}}\epsilon}_{1 \text{ failure}} + \ldots \underbrace{\binom{\gamma-1}{\gamma-\frac{m}{k}}(1-\epsilon)^{\frac{m}{k}}\epsilon^{\gamma-\frac{m}{k}}}_{\gamma-\frac{m}{k} \text{ failures}}$$

$$= \sum_{i=0}^{\gamma-\frac{m}{k}} \binom{\frac{m}{k}+i-1}{i}(1-\epsilon)^{\frac{m}{k}}\epsilon^i. \tag{8}$$

To complete the overall task, k out of n workers should successfully deliver their results to the master. Based on p, the probability of success for the overall computation, denoted by P_s, can be written as

$$P_s = \sum_{i=k}^{n} \binom{n}{i}(1-p)^{n-i}p^i. \tag{9}$$

When applying $p = (1-\epsilon)^{\frac{m}{k}}$ at (9), the P_s value corresponds to the success probability when no retransmission is allowed. Now the question is, how to design k? How about choosing the appropriate γ? We address these issues in Section V-C.

B. Latency

The overall run-time to finish the task should be reconsidered when γ is finite. Let T' be the overall run-time for a given γ. Since the term γ only affects the communication time at each worker (not the computation time), T' can be written as $T' = \underset{i\in[n]}{k^{\text{th}}\min}(X_i + S_i')$. Here S_i' denotes the communication time at worker i which is written as $S_i' = \sum_{j=1}^{C_i} Y_j$, where Y_j follows an exponential distribution with rate μ_2 and C_i is the number of transmissions at a specific worker until it

$$\mathbb{E}[G_{(i)}] = \frac{n!}{(i-1)!(n-i)!\Gamma(\frac{m}{k})} \sum_{p=0}^{i-1} (-1)^p \binom{i-1}{p} \sum_{q=0}^{(\frac{m}{k}-1)(n-i+p)} a_q(\frac{m}{k}, n-i+p) \frac{\Gamma(\frac{m}{k}+q+1)}{(n-i+p+1)^{\frac{m}{k}+q+1}} \tag{7}$$

successfully transmit $\frac{m}{k}$ number of packets. With probability $1-p$, the worker fails to transmit the $\frac{m}{k}$ packets so that C_i is undefined and S_i' becomes infinity. Random variable C_i is first written as

$$C_i = \begin{cases} \frac{m}{k} + j \text{ w.p. } \binom{\frac{m}{k}+j-1}{j} (1-\epsilon)^{\frac{m}{k}} \epsilon^j \\ \text{undefined w.p. } 1-p \end{cases} \tag{10}$$

for $j = 0, ..., \gamma - \frac{m}{k}$. Then, random variable S_i' becomes

$$S_i' \begin{cases} \sim \text{erlang}(\frac{m}{k}+j, \mu_2) \text{ w.p. } \binom{\frac{m}{k}+j-1}{j} (1-\epsilon)^{\frac{m}{k}} \epsilon^j \\ = \infty \text{ w.p. } 1-p. \end{cases} \tag{11}$$

for $j = 0, ..., \gamma - \frac{m}{k}$.

C. Optimal Resource Allocation for Practical Scenarios

For given constraints on γ and P_s, we would like to minimize $T^{(\alpha)}$ which is defined as the minimum overall runtime that we can guarantee with probability $1-\alpha$:

$$T^{(\alpha)} = \min\{\tau : \Pr[T' \leq \tau] \geq 1-\alpha\}. \tag{12}$$

For a given n, the optimization problem is formulated as

$$\min_{k,\gamma} T^{(\alpha)} \tag{13}$$

$$\text{subject to } \gamma \leq \gamma_t, \ P_s \geq 1-\delta, \tag{14}$$

where γ_t is the transmission bandwidth limit and δ shows how P_s is close to 1 ($0 \leq \delta \leq 1$). Here, the optimal k and γ can be found by combining the results of Figs 3 and 4. We first observe Fig. 3, which shows P_s as a function of γ. Since there are $\frac{m}{k}$ packets to be transmitted at each worker, we observe that $P_s = 0$ for $\gamma < \frac{m}{k}$. The optimal bounds shown in Fig. 3 are the minimum γ values that satisfy $P_s \geq 1-\delta$, where $\delta = 0.01$. Given a success probability constraint $P_s \geq 1-\delta$ and bandwidth constraint $\gamma \leq \gamma_t$, one can find the candidates of k values that satisfy these constraints. For example, if $\gamma_t = 13$, $k = 30$ and $k = 40$ become possible candidates for the optimal solution while $k = 10$ does not.

Fig. 4 shows $\Pr[T' \leq \tau]$ versus both k and γ. Among the candidates obtained from Fig. 3, one can find the optimal k that minimizes $T^{(\alpha)}$ from Fig. 4(a). For example, by setting $\alpha = 0.03$ and $\tau = 8.6$, it can be seen from Fig. 4(a) that $\Pr[T' \leq \tau] \geq 1-\alpha$ holds only for $k = 20$, which means that $k = 20$ is the optimal solution. Note that k that minimizes γ to satisfy $P_s \geq 1-\delta$ (in Fig. 3) does not always minimize $T^{(\alpha)}$, which motivated us to formulate the problem.

For a given n, one can also minimize γ by formulating

$$\min_{k} \gamma \tag{15}$$

$$\text{subject to } P_s \geq 1-\delta, \ T^{(\alpha)} \leq \tau_t \tag{16}$$

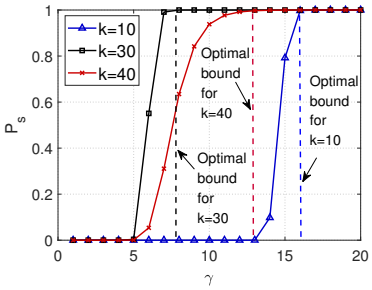
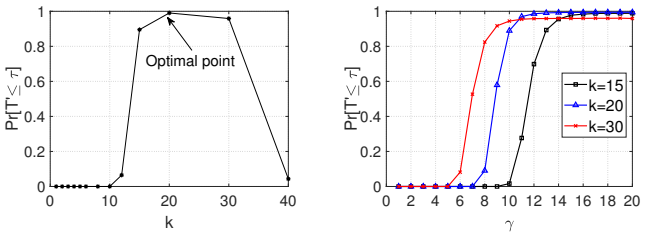


Fig. 3: P_s versus γ assuming $n = 40$, $m = 120$, $\epsilon = 0.3$. The figure shows the minimum γ satisfying $P_s \geq 1-\delta$ for different k values, where δ is set to 0.01.



(a) $\Pr(T' \leq \tau)$ versus k, $\tau = 8.6$ (b) $\Pr(T' \leq \tau)$ versus γ, $\tau = \tau_t$

Fig. 4: Plots on $\Pr[T' \leq \tau]$ for $n = 40$, $m = 120$, $\gamma_t = 13$, $\tau_t = 8.6$, $\epsilon = 0.3$, $\mu_1 = 1$, $\mu_2 = 5$.

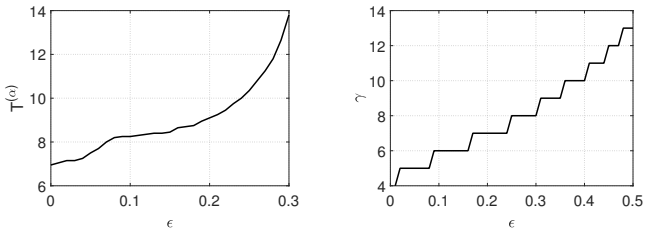
where τ_t is the latency constraint. It can be seen that the constraint $T^{(\alpha)} \leq \tau_t$ is equivalent to $\Pr[T' \leq \tau_t] \geq 1-\alpha$. From Fig. 3, candidates of (k, γ) pairs satisfying $P_s \geq 1-\delta$ can be first obtained. Among these candidates, one can find the best k that minimizes γ under the constraint $T^{(\alpha)} \leq \tau_t$ from Fig. 4(b), which shows $\Pr[T' \leq \tau_t]$ versus γ for $\tau_t = 8.6$. By setting $\alpha = 0.05$, it can be seen that the minimum γ that satisfies $\Pr[T' \leq \tau_t] \geq 1-\alpha$ becomes $\gamma = 11$, which can be achieved for $k = 20$. As γ grows, we observe that the probability $\Pr[T' \leq \tau_t]$ converges to $\Pr[T \leq \tau_t]$ for each k in Fig. 4(b).

D. Achievable Curves

Based on the solutions to the optimization problems, Fig. 5 illustrates the optimal curves as a function of ϵ. Fig. 5(a) shows the achievable $T^{(\alpha)}$ in (13) as a function of ϵ. We observe that the achievable $T^{(\alpha)}$ increases as ϵ grows. From Fig. 5(b), which shows the optimal $\gamma - \epsilon$ curve, we observe that the achievable γ in (15) increases as ϵ grows, which indicates that a larger transmission bandwidth is required for a higher erasure probability.

VI. Concluding Remarks

We studied the problem of coded computation assuming link failures, which are a fact of life in current wired/wireless networks. Closed-form expressions for the lower and upper



(a) Optimal $T^{(\alpha)} - \epsilon$ curve (b) Optimal $\gamma - \epsilon$ curve

Fig. 5: Optimal curves assuming $n = 40$, $m = 120$, $\mu_1 = 1$, $\mu_2 = 5$, $\gamma_t = 7$, $\tau_t = 10$, $\alpha = 0.05$, $\delta = 0.01$.

bounds on the expected latency are obtained. Then, we show that the performance of the coding scheme which minimizes the upper bound nearly achieves the optimal run-time, which shows the effectiveness of the proposed design. Finally, under the setting of limited number of retransmissions, we formulated practical optimization problems and obtained achievable curves as a function of packet erasure probability ϵ.

VII. Proof of Theorem 2

We only provide the proof for the upper bound. The full proof is in [15]. We first prove the following proposition, which can be illustrated as in Fig. 6.

Proposition 1. *For any two sequences* $\{a_i\}_{i\in[n]}$, $\{b_i\}_{i\in[n]}$,

$$k^{\text{th}}\min_{i\in[n]}(a_i + b_i) \leq \min_{i\in[n]\setminus[k-1]}(a_{(n+k-i)} + b_{(i)}). \tag{17}$$

Proof: Due to page limit, we only provide a proof sketch here. Define the ordering vector $\boldsymbol{\pi} \in \Pi$, where Π is the set of permutations of $[n]$, i.e., $\Pi = \{\boldsymbol{\pi} = [\pi_1, ..., \pi_n] : \pi_i \in [n]\ \forall i \in [n],\ \pi_i \neq \pi_j\ for\ i \neq j\}$. Define $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2 \in \Pi$ where $\boldsymbol{\pi}_1 = [\pi_{1,1}, ..., \pi_{n,1}]$, $\boldsymbol{\pi}_2 = [\pi_{1,2}, ..., \pi_{n,2}]$. Noting that the k^{th} smallest value out of n can be always rewritten as the minimum of $n-k+1$ largest values, $k^{\text{th}}\min_{i\in[n]}(a_i + b_i)$ can be always rewritten as

$$k^{\text{th}}\min_{i\in[n]}(a_i + b_i) = \min_{i\in[n]\setminus[k-1]}(a_{(\pi_{i,1})} + b_{(\pi_{i,2})}) \tag{18}$$

by appropriately selecting $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2 \in \Pi$. Now define another set of ordering vectors $\Pi^* \subseteq \Pi$ as $\Pi^* = \{\boldsymbol{\pi} \in \Pi : \pi_i \in [n] \setminus [k-1]\ \forall i \in [n] \setminus [k-1]\}$. Then we can always construct $\boldsymbol{\pi}_1^*, \boldsymbol{\pi}_2^* \in \Pi^*$ such that $\pi_{i,1} \leq \pi_{i,1}^*$, $\pi_{i,2} \leq \pi_{i,2}^*$ for all $i \in [n] \setminus [k-1]$, which can be proved easily. Moreover, we can always find $\boldsymbol{\pi}^* \in \Pi^*$ such that (18) is upper bounded as

$$\begin{aligned}\min_{i\in[n]\setminus[k-1]}(a_{(\pi_{i,1})} + b_{(\pi_{i,2})}) &\leq \min_{i\in[n]\setminus[k-1]}(a_{(\pi_{i,1}^*)} + b_{(\pi_{i,2}^*)}) \\ &= \min_{i\in[n]\setminus[k-1]}(a_{(\pi_i^*)} + b_{(i)}).\end{aligned} \tag{19}$$

Now we want to show that for any $\boldsymbol{\pi}^* \in \Pi^*$, (19) cannot be larger than $\min_{i\in[n]\setminus[k-1]}(a_{(n+k-i)} + b_{(i)})$, which is the upper bound illustrated in Fig. 6. Define $U = \min_{i\in[n]\setminus[k-1]}(a_{(n+k-i)} + b_{(i)}) = a_{(n+k-j)} + b_{(j)}$ for some $j \in [n]\setminus[k-1]$. Suppose there exist U' and an ordering vector $\boldsymbol{\pi'} = [\pi_1', ..., \pi_n'] \in \Pi^*$ such that $\min_{i\in[n]\setminus[k-1]}(a_{(\pi_i')} + b_{(i)}) = U' > U$. Then, $a_{(\pi_{j'}')} + b_{(j')} > a_{(n+k-j)} + b_{(j)}$ should be satisfied for all $j' \in [n] \setminus [k-1]$. This implies $\pi_{j'}' > n+k-j$ for all $j' \in [j] \setminus [k-1]$, which is a contradiction since one-to-one mapping between j' and $\pi_{j'}'$ cannot be constructed for $j' \in [n] \setminus [k-1]$. ■

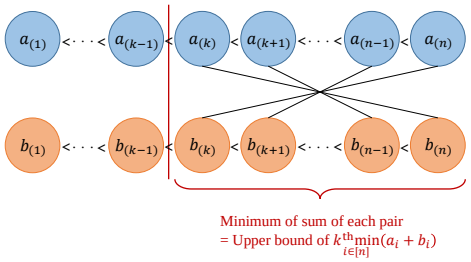


Fig. 6: Illustration of Proposition 1.

Now the upper bound is derived as

$$\mathbb{E}[T] = \mathbb{E}[k^{\text{th}}\min_{i\in[n]}(X_i + S_i)] \tag{20}$$

$$\underset{(a)}{\leq} \mathbb{E}[\min_{i\in[n]\setminus[k-1]}(X_{(n+k-i)} + S_{(i)})] \tag{21}$$

$$\underset{(b)}{\leq} \min_{i\in[n]\setminus[k-1]}(\frac{H_n - H_{i-k}}{\mu_1} + \frac{H_n - H_{n-i}}{(1-\epsilon)\mu_2}) \tag{22}$$

where (a) holds from Proposition 1 and (b) holds from the Jensen's inequality.

References

[1] 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, Mar. 2018.

[2] K. Lee, C. Suh, and K. Ramchandran, "High-dimensional coded matrix multiplication," in *Proc. IEEE ISIT*, Jun. 2017, pp. 2418-2422.

[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," in *Proc. NIPS*, Long Beach, CA, USA, Dec. 2017

[4] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," in *Proc. NIPS*, pp. 2092–2100, 2016.

[5] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc. ICML*, 2017.

[6] S. Dutta, V. cadambe, and P. Grover, "Coded convolution for parallel and distributed computing within a deadline," in *Proc. IEEE ISIT*, Jun. 2017, pp. 2403-2407.

[7] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded Fourier transform", in *Proc. Allerton Conf.*, 2017.

[8] A. Reisizadeh, S. Prakash, R. Pedarsani, and S. Avestimehr, "Coded computation over heterogeneous clusters," in *Proc. IEEE ISIT*, 2017.

[9] H. Park, K. Lee, J. Sohn, C. Suh, and J. Moon, "Hierarchical coding for distributed computing," in *Proc. IEEE ISIT*, Jun. 2018.

[10] A. Reisizadeh and R. Pedarsani, "Latency analysis of coded computation schemes over wireless networks," in *Proc. Allerton Conf.*, 2017.

[11] N. Ferdinand, and S. C. Draper, "Hierarchical Coded Computation," in *Proc. IEEE ISIT*, Jun. 2018.

[12] S. Kiani, N. Ferdinand, and S. C. Draper, "Exploitation of Stragglers in Coded Computation," in *Proc. IEEE ISIT*, Jun. 2018.

[13] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 350-361, 2011.

[14] D. Datla et al., "Wireless distributed computing: a survey of research challenges," *IEEE Communications Magazine*, vol. 50, no. 1, pp. 144-152, Jan. 2012.

[15] D.-J. Han, J. Sohn, and J. Moon, "Coded Distributed Computing over Packet Erasure Channels," *arXiv:1901.03610*, 2019.

[16] P. Bremaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Science Business Media, 2013, vol. 31.

[17] S. S. Gupta, "Order statistics from the Gamma distribution," *Technometrics*, vol. 2, no. 2, pp. 243-262, 1960.