Computation Over Multi-Access Channels: Multi-Hop Implementation and Resource Allocation

Fangzhou Wu, Li Chen, Nan Zhao, *Senior Member, IEEE,* Yunfei Chen, *Senior Member, IEEE,* F. Richard Yu, *Fellow, IEEE,* and Guo Wei

***Abstract*—For future wireless networks, enormous numbers of interconnections are required, creating a multi-hop topology and leading to a great challenge on data aggregation. Instead of collecting data individually, a more efficient technique, computation over multi-access channels (CoMAC), has emerged to compute functions by exploiting the signal-superposition property of wireless channels. However, it is still an open problem on the implementation of CoMAC in multi-hop wireless networks considering fading channel and resource allocation. In this paper, we propose multi-layer CoMAC (ML-CoMAC) by combining CoMAC and orthogonal communication to compute functions in the multi-hop network. Firstly, to make the multi-hop network more tractable, we reorganize it into a hierarchical network with multiple layers that consists of subgroups and groups. Then, in the hierarchical network, the implementation of ML-CoMAC is given by computing and communicating subgroup and group functions over layers, where CoMAC is applied to compute each subgroup function and orthogonal communication is adopted for each group to obtain the group function. The general computation rate is derived and the performance is further improved through time allocation and power control. The closed-form solutions to optimization problems are obtained, which suggests that orthogonal communication and existing CoMAC schemes are generalized.**

***Index Terms*—Achievable computation rate, data aggregation, function computation, hierarchical networks, resource allocation.**

I. Introduction

5G and the Internet of Things will lead to a revolution in wireless networks [1], [2]. With such enormous numbers of nodes, it is impractical to wirelessly aggregate a large amount of distributed data by using conventional multi-access schemes since this would result in excessive network latency and low spectrum utilization efficiency. Thus, how to aggregate data efficiently from distributed nodes is of great importance.

This work was supported by the National Key Research and Development Program of China (Grant No. 2018YFA0701603), and USTC Research Funds of the Double First-Class Initiative (Grant No. YD3500002001). This paper was presented in part at the IEEE Wireless Communications and Networking Conference (WCNC), Seoul, South Korea, May 2020. (*Corresponding author: Li Chen*)

F.R. Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, K1S 5B6, Canada. (email: richard.yu@carleton.ca).

Recently, computation over multi-access channels (CoMAC) has emerged as a promising solution that merges computation and communication by exploiting the signal-superposition property of wireless channels. Instead of collecting individual data, it collects a relevant function of the measurements from distributed nodes via concurrent node transmissions [3]–[21]. These functions computed by CoMAC belong to a class of nomographic functions such as averaging and geometric mean, and are widely used in data aggregation [3]. As a straightforward use of CoMAC, nodes in wireless sensor networks can simultaneously transmit their readings over the air to compute a function value of the sensor readings (e.g., arithmetic mean, polynomial or the number of active nodes) instead of requiring individual readings. Besides, in federated learning, to obtain the average learning model from the local models of distributed nodes, CoMAC is deployed to significantly reduce the aggregation latency [4].

CoMAC was traced back to [5]–[7], where the capacity bounds of the different types of networks were analyzed. In [5], a source coding problem involving communicating a function of two variables in a simple two-node network with side information at the receiver has been solved. Considering the multi-source network, [6] described it as one of communicating possibly correlated sources over a multi-terminal wireless network. For collocated networks and random planar networks, the capacity bounds are derived in [7]. The above works indicated that transmitting functions, instead of individual data, is a significant way to improve the network capacity from an information-theoretic perspective.

Inspired by the information-theoretic conclusion, CoMAC were first proposed via some analog approaches [8]–[13], where pre-processing at each node and post-processing at the fusion center were carried out against channel fading. The designs of pre-processing and post-processing used to compute linear and non-linear functions have been proposed in [8], and the design of multi-function CoMAC was proposed in [9] by using multiple antennas. The impact of channel estimation error for the analog CoMAC was characterized in [10] and to avoid the overhead of channel estimation, [11] proposed the blind CoMAC via provable Wirtinger Flow. To verify the feasibility of the analog CoMAC in practice, software-defined radio was built in [12], and the authors in [13] implemented a cooperative wide-band spectrum sensing system using CoMAC. For robustness to noise, the digital CoMAC was further proposed using joint source-channel coding in [14]–[21] to improve the equivalent signal-to-noise ratio (SNR). The potential of linear source coding was discussed in [14], and its application for

CoMAC was presented in [15]. Compared with linear source coding, nested lattice coding could approach the performance of a standard random coding [16]. The lattice-based CoMAC was extended to a general framework in [17] for networks with linear channels and additive white Gaussian noise. In [18], the authors derived the exact achievable computation rate of the digital CoMAC considering channel fading. Compared with the traditional communication-based computation scheme, the performance gain of CoMAC was derived in [19]. To improve the performance, some CoMAC schemes based on function division was given in [20], [21] through theoretical analysis.

Most of CoMAC works only considered single-hop networks through direct communications. Compared with single-hop networks, the emergent of large-scale networks, e.g., wireless community networks and sensor networks, makes the multi-hop networks a trend [22] and the deployment of multi-hop networks is an important approach to fully utilize the potential capacity of the network [23]. This implies that the implementation of CoMAC in multi-hop networks is of great importance for the upcoming 5G and the Internet of Things. It is worth mentioning that, in [7], [18], only the capacity bounds of multi-hop networks transmitting functions were derived considering the non-fading and fading channel, respectively. However, they still lack the general implementation for multi-hop networks to achieve CoMAC, the exact performance analysis, and the resource allocation to fully use the flexibility of the network resources. Thus, for multi-hop networks, it is still an open problem on how to practically implement CoMAC and how to derive the exact performance considering the fading channel and resource allocation.

Motivated by these observations, we try to extend the CoMAC implementation of single-hop networks to the one of multi-hop networks. The multi-layer CoMAC (ML-CoMAC) is proposed to compute functions over layers. More specifically, we first recast the multi-hop network into a hierarchical network with multiple layers consisting of subgroups and groups. Then, the implementation of ML-CoMAC is given based on the hierarchical network. Exact expressions of achievable computation rates are derived based on nested lattice coding. Furthermore, resource allocation is considered to improve the computation rate, and the corresponding closed-form solutions are given. Our contributions are summarized as follows:

- *ML-CoMAC implementation*. To easily characterize a series of multi-hop networks, we reorganize the multi-hop network into the hierarchical network with multiple layers by introducing two components, i.e., groups and subgroups. In the hierarchical network, we rule the behaviors of subgroups and groups, and propose ML-CoMAC by computing and communicating subgroup and group functions over layers and reconstructing the desired function at the fusion center. Each subgroup function is obtained by using CoMAC and the group function is obtained by using orthogonal communication.
- *General computation rate*. The theoretical expression of the computation rate of ML-CoMAC is derived, and it suggests that the subgroup with the worst computation rate plays an important role in the network. Also, the orthogonal communication and existing CoMAC schemes in [18] are generalized.
- *Time allocation and power control*. We formulate two optimization problems considering time allocation with fixed power control and adaptive power control, respectively. Both closed-form solutions are derived, which suggests that compared with fixed power control, the performance with adaptive power control is further improved as expected.

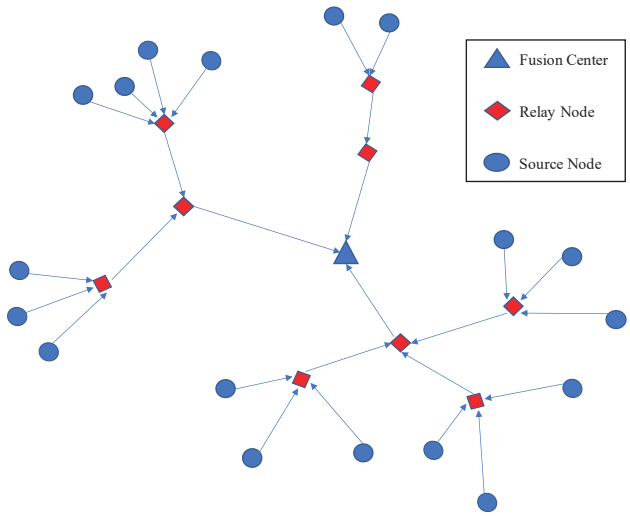


Fig. 1. The topology of the multi-hop network.

The rest of the paper is organized as follows. Section II introduces the considered the multi-hop network and the classical schemes of function computation. The proposed ML-CoMAC is presented in Section III. Based on ML-CoMAC, we derive the computation rate of ML-CoMAC in Section IV. Section V focuses on the analysis of the performance of the proposed ML-CoMAC, which includes power control and time allocation. Simulation results and the corresponding discussion are presented in Section VI, and conclusions are given in Section VII.

Notations : Throughout this paper, we define $\mathsf{C}(x) = \log(1+x)$ and $\mathsf{C}^+(x) = \max\left\{\frac{1}{2}\log(x), 0\right\}$. Let $[1:n]$ denote a set $\{1, 2, \cdots, n\}$. For a set $\mathcal{A}$, $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$. Let the entropy of a random variable A be $H(A)$ and the expectation of it be $\mathsf{E}[A]$. A set $\{x_1, x_2, \cdots, x_N\}$ is written as $\{x_i\}_{i\in[1:N]}$ or $\{x_i\}_{i=1}^N$ for short.

II. System Model

In this section, we first present the topology of the multi-hop network and then introduce two classical schemes for function computation, i.e., CoMAC and orthogonal communication. At the end of this section, we raise some open problems that need to solve.

A. Multi-hop Networks

In practical terms, the topology of a wireless network is arbitrary. With different deployment environments, the network

would be different. Thus, we follow [7] considering the multi-hop network that is the tree topology with designated roots as data collectors and only the leaves appearing to have data sources. More specifically, it consists of source nodes at the edge, relay nodes, and one fusion center as the destination. In the multi-hop network, the fusion center wishes to compute the desired function concerning the data of all the source nodes. With a given topology, the network is demonstrated as Fig. 1. We assume that the number of the source nodes is K and define a set $\mathcal{K}$ including the indexes of all the source nodes. The i-th source node N_i draws data from the corresponding random source S_i for T_d times and then provides a length-T_d data vector as $\mathbf{s}_i = [s_i[1], \cdots, s_i[j], \cdots s_i[T_d]]$.

Let $\mathbf{b_v} = [S_1, S_2, \cdots, S_K]$ be the random source vector associated with a joint probability mass function $p_{\mathbf{b_v}}(\cdot)$. The desired function determined by the random source vector $\mathbf{b_v}$ is expressed as $f(\mathbf{b_v})$, and its definition is given as follows.

Definition 1 (Desired Function)**.** For all $j \in [1 : T_d]$, the function with variables $\{s_1[j], s_2[j], \cdots, s_K[j]\}$ is called the desired function with the form as

$$f(s_1[j], s_2[j], \cdots, s_K[j]) = f(\mathbf{s}[j]), \tag{1}$$

where $\mathbf{s}[j] = [s_1[j], s_2[j], \cdots, s_K[j]]$ is drawn from $p_{\mathbf{b_v}}(\cdot)$. Every function $f(\mathbf{s}[j])$ is seen as a realization of $f(\mathbf{b_v})$. Thus, the fusion center computes T_d desired functions when each source node gets data from each random source for T_d times.

Remark 1 (Typical Desired Functions)**.** As studied in [18], [24], CoMAC is designed to compute different types of desired functions. There are two typical functions that we focus on. The function $f(\mathbf{s}[j])$, with values in the set $\{\sum_{i=1}^{K} a_{1,i}s_i[j], \cdots, \sum_{i=1}^{K} a_{L_s,i}s_i[j]\}$, is called the arithmetic sum function, where $a_{l,i} \in \mathbb{R}$ is the weighting factor for the node $\mathrm{N}_{1,i}$, and L_s is the number of available weighting factors and belongs to $\mathbb{N}$. The arithmetic sum function is a weighted sum function, which includes the mean function $f(\mathbf{s}[j]) = \frac{1}{K_1}\sum_{i=1}^{K} s_i[j]$ and the function for the active node only $f(\mathbf{s}[j]) = \{s_1[j], s_2[j], \cdots, s_K[j]\}$ as special cases. Otherwise, the function $f(\mathbf{s}[j])$, with values in the set of $\{\sum_{i=1}^{K} \mathbf{1}_{s_i[j]=0}, \cdots, \sum_{i=1}^{K} \mathbf{1}_{s_i[j]=p}\}$, is regarded as the type function where $\mathbf{1}_{(\cdot)}$ denotes the indicator function and $p \in \mathbb{N}$. As pointed out in [7], any symmetric function such as mean, variance, maximum, minimum and median can be attained from the type function.

To attain reliable computations against noise, a block code, named sequences of nested lattice codes [17], is used. With the length-n block code, the computation rate is used as performance metrics [3], [17], [18], [21], [24], of which the definition is given as follows.

Definition 2 (Computation rate [18], [20])**.** The computation rate specifies how many function values can be computed per channel use within a predefined accuracy. It can be written as $R = \lim_{n\to\infty} \frac{T_d(n)}{n} H(f(\mathbf{b_v}))$ where T_d is the number of function values, n is the length of the block code and $H(f(\mathbf{b_v}))$ is the entropy of $f(\mathbf{b_v})$. Apart from this, R is achievable only if there is a length-n block code so that the probability $\Pr\left(\bigcup_{j=1}^{T_d}\left\{\hat{f}(\mathbf{s}[j]) \neq f(\mathbf{s}[j])\right\}\right) \to 0$ as n increases[1], where $\hat{f}(\mathbf{s}[j])$ is the estimated desired function.

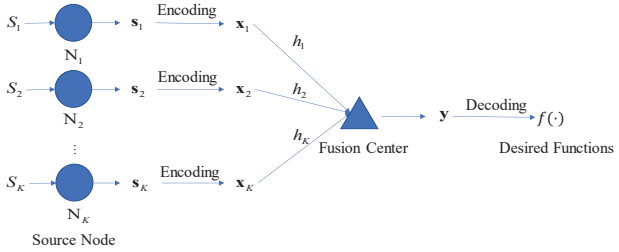


Fig. 2. The classical CoMAC for the single-hop network.

Remark 2 (Distribution of Samples)**.** The computation rate is determined by the information entropy $H(f(\mathbf{b_v}))$ of $f(\mathbf{b_v})$, not every realization. Hence, the identically and independently distributed (i.i.d.) or non-i.i.d. samples only lead to different information entropy, which is a constant with a given desired function and a given distribution. In our design, the distribution of the random source vector follows the joint probability mass function $p_{\mathbf{b_v}}(\cdot)$ without loss of generality, which includes the i.i.d. case and the non-i.i.d. case. With a given function (e.g., the arithmetic sum function and the type function), the information entropy is regarded as a constant that can be derived or estimated, which implies that the influence of the distribution of the samples on the computation rate is included.

B. Schemes of Function Computation

There exist two classical schemes, namely CoMAC and orthogonal communication, to achieve function computation in a single-hop wireless network.

- ***CoMAC.*** CoMAC has been well investigated as the efficient aggregation scheme in the single-hop network and its classical framework is given in Fig. 2. Different from the multi-hop network, in Fig. 2, all the source nodes and the fusion center can be communicated with each other directly.
  Referring to the existing CoMAC works [3], [17], [18], [20], [21], [24], let $\mathbf{s}_i$ represent the data vector of the node N_i whose length is T_d. Denote $\mathbf{x}_i = [x_i[1], x_i[2], \cdots, x_i[n]]$ as the length-n transmitted vector of the node N_i. The univariate function $\boldsymbol{\mathcal{E}}_i(\cdot)$ which generates $\mathbf{x}_i = \boldsymbol{\mathcal{E}}_i(\mathbf{s}_i)$ is an encoding function of N_i. This means that $\mathbf{s}_i$ with length T_d is mapped to a transmitted vector $\mathbf{x}_i$ with length n for N_i. Then, the received signal for the m-th channel use can be expressed as
  $$y[m] = \sum_{i=1}^{K} v_i[m]h_i[m]x_i[m] + w[m], \tag{2}$$
  where $h_i[m]$ is the channel from N_i to the fusion center at the m-th channel use, $x_i[m]$ is the m-th element of

[1]Note that $T_d(n)$ denotes the number of reliably computable function values with a length-n block code, which is a function of n.

the transmitted vector $\mathbf{x}_i$, $v_i[m] = \frac{|h_i[m]|}{h_i[m]}\sqrt{P_i[m]}$ is the power factor of N_i, $P_i[m]$ is the transmitted power of N_i and $w[m]$ is i.i.d. complex Gaussian random noise following $\mathcal{CN}(0,1)$.

After n channel uses, the received vector $\mathbf{y}$ is obtained at the fusion center. The decoding function $\boldsymbol{\mathcal{D}}_j(\cdot)$ is used to estimate the j-th desired function $f(\mathbf{s}[j])$, which satisfies $\hat{f}(\mathbf{s}[j]) = \boldsymbol{\mathcal{D}}_j(\mathbf{y})$. This implies that the fusion center obtains T_d desired functions depending on the received vector with length n. Its computation rate [18, Theorem 3] is given as

$$R = \mathsf{C}^+\left(\frac{1}{K} + \mathsf{E}\left[\min_{i\in[1:K]} |h_i|^2\right] P\right), \tag{3}$$

where K is the number of source nodes in the network, P is the transmitted power of each node and $|h_i|^2$ is the channel gain from N_i to the fusion center.

- ***Orthogonal Communication.*** The other solution to computing functions uses orthogonal resource blocks (e.g., channel uses, code sequences, and sub-carriers) to transmit all the individual data to the fusion center and then finish the computation. To compute the j-th desired function $f(\mathbf{s}_1[j])$, the fusion center should first obtain the individual data $\{s_i[j]\}_{i=1}^K$ during K channel uses. Then, the corresponding desired function $f(\mathbf{s}[j])$ is calculated. It is also known as the time-sharing technique, which achieves a computation rate of

$$R = \frac{1}{K}\mathsf{C}\left(\mathsf{E}\left[|h|^2\right] P\right), \tag{4}$$

where $|h|^2$ is the channel gain of each node without loss of generality.

Remark 3 (Implementation of Digital CoMAC)**.** The digital CoMAC is based on the digital communication schemes (see [24, Chapter. 4], also [14]–[18], [20]), and is more easily implemented in our current digital base-bands and more effectively against the noise. In our design, the implemented framework is the digital CoMAC based on the nested lattice coding so that we evaluate the performance of the proposed scheme by the computation rate. The unique difference between the digital CoMAC and the analog CoMAC is the added computation code (sequences of nested lattice codes are usually employed for the digital framework) to protect the function value against the noise. Firstly, each data vector $\mathbf{s}_i$ is converted to a bit sequence by a quantizer. Then, each encoder maps the bit sequence to a code-word $\mathbf{x}_i$. After the pre-equalization to control power and remove the fading channel, each node transmits its vector simultaneously. At the fusion center, the received vector $\mathbf{y}$ is decoded and recovered by the inverse quantizer. By carefully choosing the parameters for the nested lattice code, there exists an achievable computation rate such that the function value can be recovered without error.

C. Open Problems

Neither orthogonal communication nor CoMAC can be implemented directly in the multi-hop network. For orthogonal communication, the collection of individual data from nodes results in excessive latency as the number of nodes increases. As for CoMAC, different nodes may have a different next hop in the multi-hop network, which is different from the case in the single-hop network with direct communications. Besides, some practical issues, i.e., channel fading and limited network resources, need to consider. Thus, we expect to implement CoMAC in the multi-hop network and expand the analysis of single-hop networks to the one of multi-hop networks considering the practical case. Since the analysis of multi-hop networks is more general but also challenging, several issues need to be solved.

1. The multi-hop network needs to be reorganized to make further design possible. It needs to be considered that how to recast the multi-hop network into a hierarchical network without topology modification and how to design a scheme that ensures the reliable computation of the desired function at the fusion center.
2. With the proposed scheme, it becomes important to evaluate the performance through computation rate and further optimize it. Thus, the corresponding computation rate should be derived and the resource allocation should be discussed.

III. The Proposed ML-CoMAC

For the multi-hop network shown as Fig. 1, the network's structure makes the analysis difficult. Before proposing the scheme, in Section III-A, we first reorganize the multi-hop network into a hierarchical network with multiple layers, which consists of groups and subgroups. In Section III-B, we rule the behaviors of these subgroups and groups to achieve the division and reconstruction of the desired functions. At last, in Section III-C, the procedure of ML-CoMAC is given.

A. Hierarchical Networks

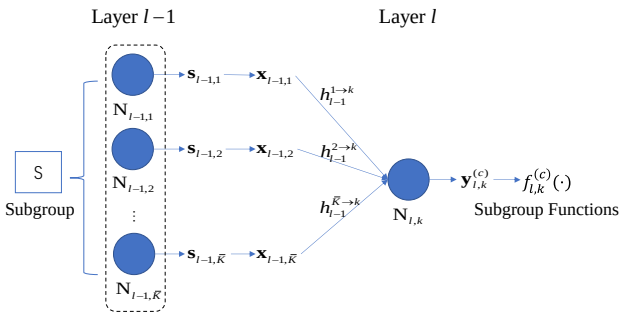
We introduce two components to reorganize the multi-hop network, which are given as follows.

- ***Subgroups.*** Assume there are two layers, namely the $(l-1)$-th layer and the l-th layer. We place $\bar{K}$ nodes[2] in the $(l-1)$-th layer and one node in the l-th layer as $\mathrm{N}_{l,k}$, where these $\bar{K}$ nodes wish to transmit their data to $\mathrm{N}_{l,k}$. The subgroup, whose index is c, consists of these $\bar{K}$ nodes where the i-th node is denoted as $\mathrm{N}_{l-1,i}$, and the node $\mathrm{N}_{l,k}$ computes the subgroup function $f_{l,k}^{(c)}(\cdot)$ associated with the subgroup c.

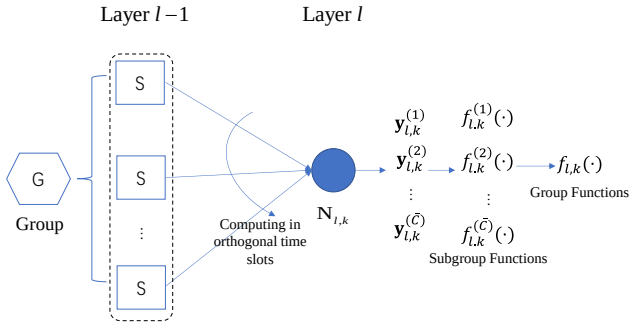
  As given in Fig. 3a, each node $\mathrm{N}_{l-1,i}$ in the $(l-1)$-th layer owns a data vector $\mathbf{s}_{l-1,i}$ of length T_d. In the l-th layer, the node $\mathrm{N}_{l,k}$ computes the subgroup function $f_{l,k}^{(c)}(\cdot)$ via concurrent node transmissions. Based on Section II-B, CoMAC is applied to compute the subgroup function.
- ***Groups.*** As shown in Fig. 3b, one group, which is allocated to the node $\mathrm{N}_{l,k}$ in the l-th layer, consists of several subgroups in the $(l-1)$-th layer. Assume the number of subgroups is $\bar{C}$[3], then the node $\mathrm{N}_{l,k}$ needs to

[2]Since the number of nodes in the $(l-1)$-th layer can be arbitrary, we assume that it is $\bar{K}$ without loss of generality.

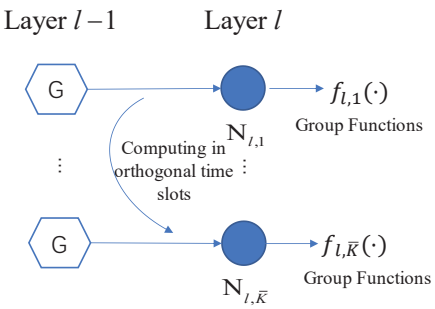
[3]Since the number of subgroups for a group can be arbitrary, we assume that it is $\bar{C}$ without loss of generality.



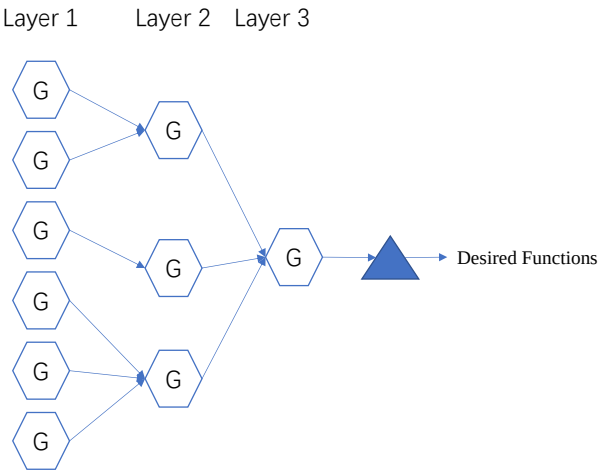
(a) The structure of one subgroup.



(b) The structure of one group.



(c) From one layer to another layer.



(d) The topology of the hierarchical network.

Fig. 3. Reorganization of the multi-hop network.

reconstruct the group function $f_{l,k}(\cdot)$ using $\bar{C}$ subgroup functions.

To reconstruct the group function $f_{l,k}(\cdot)$ at $\mathrm{N}_{l,k}$, all the subgroup functions must be obtained first. Based on Section II-B, orthogonal communication is applied by communicating the subgroup functions to $\mathrm{N}_{l,k}$ during the given channel uses, where different subgroup is active to compute different subgroup function $f_{l,k}^{(c)}(\cdot)$ at different channel use. After obtaining all the subgroup functions $\{f_{l,k}^{(c)}(\cdot)\}_{c=1}^{\bar{C}}$, the node $\mathrm{N}_{l,k}$ can reconstruct the group function $f_{l,k}(\cdot)$.

Remark 4 (Generalization of Classical Function-Computation Schemes)**.** With the description of two components, one can observe that these two classical schemes are combined in our scheme, where the subgroup function is obtained by CoMAC whereas the group function is obtained by orthogonal communication. Also, the structure of the group shown as Fig. 3b is a single-hop network in a general way. By setting the number of subgroups $\bar{C}$ to one, CoMAC is generalized, which implies that the only one subgroup is treated as a group. Also, by setting the number of subgroups $\bar{C}$ to the number of nodes $\bar{K}$, orthogonal communication is generalized, which implies that each node as a subgroup transmits its data individually.

With the help of subgroups and groups, the aggregation from the $(l-1)$-th layer to the l-th layer is given as Fig. 3c. Each node $\mathrm{N}_{l,k}$ in the l-th layer serves a group to reconstruct the corresponding group function $f_{l,k}(\cdot)$. Similar to orthogonal communication, different group in the $(l-1)$-th layer is allocated some orthogonal channel uses to compute the subgroup functions and reconstruct the group function at the corresponding node in the l-th layer. Based on Fig. 3c, we further expand it to the case with multiple layers. Then, the hierarchical network is obtained as shown in Fig. 3d, where the orthogonal transmission is used for each group since the groups in the $(l-1)$-th layer have to communicate with different relay nodes in the l-th layer.

Hierarchical Networks. The multi-hop network in Fig. 1 is reorganized into the hierarchical network, which consists of L $(L \geq 2)$ layers where the l-th layer includes K_l nodes and the indexes of them belong to a set $\mathcal{K}_l$. Compared with the multi-hop network, in the hierarchical network, K_1 nodes in the first layer are the source nodes, the nodes from the second layer to the $(L-1)$-th layer are relay nodes, and the only one node in the L-th layer is regarded as a fusion center. Finally, the desired function will be computed at the fusion center over layers.

Remark 5 (Topology-Modification-Free Operation)**.** In both multi-hop networks and hierarchical networks, the routing path of each node does not be changed, which implies that the next hop of one node in the multi-hop network is the same as the one in the corresponding hierarchical network. Further, the condition, where each node only owns one next hop, is satisfied in both multi-hop networks and hierarchical networks. Thus, we can always find an equivalent hierarchical network by changing the parameters of the hierarchical network to replace the multi-hop one. This implies that our design does not break the network topology. Protecting the topology can avoid the cost of physically reorganizing the network, especially for a network with massive numbers of nodes.

B. Division and Reconstruction of Functions

In the hierarchical network including L layers, the fusion center computes the desired function $f\left(\{s_{1,k}[j]\}_{k\in\mathcal{K}_1}\right)$, where $s_{1,k}[j]$ is the data sampled by the node $\mathrm{N}_{1,k}, k \in \mathcal{K}_1$. Depending on a given topology, all these nodes in $\mathcal{K}_{l-1}$ are divided into K_l groups and allocated to K_l nodes in the l-th layer for $l \geq 2$. We define the set $\mathcal{K}_{\mathrm{N}_{l,k}}$ including the indexes of the nodes in the group allocated to $\mathrm{N}_{l,k}$. Thus, the desired function is divided into group functions and each group function, associated with the data $\{s_{l-1,i}[j]\}_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}}$, is computed at $\mathrm{N}_{l,k}$. The detailed definition of the group function is given as follows.

Definition 3 (Group Function)**.** For $l \geq 2$, let

$$\mathcal{K}_{\mathrm{N}_{l,k}} = \{x : x \in \mathcal{K}_{l-1}\} \tag{5}$$

denote a set including these indexes of the nodes as a group allocated to $\mathrm{N}_{l,k}$. Each element x in $\mathcal{K}_{\mathrm{N}_{l,k}}$ is the index of a node from the set $\mathcal{K}_{l-1}$. Suppose that $\bigcup_{k\in\mathcal{K}_l}\mathcal{K}_{\mathrm{N}_{l,k}} = \mathcal{K}_{l-1}$ and $\mathcal{K}_{\mathrm{N}_{l,u}}\bigcap\mathcal{K}_{\mathrm{N}_{l,v}} = \emptyset$ for all $u, v \in \mathcal{K}_l$. A function $f_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}})$ is said to be a group function if and only if there exists a function $g_l(\cdot)$ satisfying

$$\begin{aligned} f(\mathbf{s}_1[j]) = g_l(&f_{l,1}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}_{\mathrm{N}_{l,1}}}), f_{l,2}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}_{\mathrm{N}_{l,2}}}),\\ &\cdots, f_{l,K_l}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}_{\mathrm{N}_{l,K_l}}})) \end{aligned} \tag{6}$$

for $l \geq 2$.

Definition 3 suggests that the group functions in each layer can reconstruct the desired function, even though the desired function only needs to be reconstructed at the fusion center in the last layer by these group functions in the $(L-1)$-th layer.

To attain the computation of the group function at $\mathrm{N}_{l,k}$, all these subgroup functions should be obtained at $\mathrm{N}_{l,k}$ first since a group function is further divided into several subgroup functions[4]. The function computed by a subgroup is called a subgroup function, and its definition is given as follows.

Definition 4 (Subgroup Function)**.** Assume the nodes in $\mathcal{K}_{\mathrm{N}_{l,k}}$ as a group is divided into $C_{\mathrm{N}_{l,k}}$ subgroups. The set $\mathcal{C}_{\mathrm{N}_{l,k}}$ includes indexes of these $C_{\mathrm{N}_{l,k}}$ subgroups satisfying that $\bigcup_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}} = \mathcal{K}_{\mathrm{N}_{l,k}}$, where $\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}} \subseteq \mathcal{K}_{\mathrm{N}_{l,k}}$ and $\mathcal{K}^{(u)}_{\mathrm{N}_{l,k}}\bigcap\mathcal{K}^{(v)}_{\mathrm{N}_{l,k}} = \emptyset$ for all $u, v \in \mathcal{C}_{\mathrm{N}_{l,k}}$. A function $f^{(c)}_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}})$ is said to be a subgroup function if and only if there exists a function $g_{l,k}(\cdot)$ satisfying

$$\begin{aligned} &f_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}})\\ &= g_{l,k}(f^{(1)}_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}^{(1)}_{\mathrm{N}_{l,k}}}, \cdots,\\ &\quad f^{(C_{\mathrm{N}_{l,k}})}_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}^{(C_{\mathrm{N}_{l,k}})}_{\mathrm{N}_{l,k}}})). \end{aligned} \tag{7}$$

The property of subgroup functions is similar to the one of group functions, which shows that a group function $f_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}})$ can be reconstructed at $\mathrm{N}_{l,k}$ after $\mathrm{N}_{l,k}$ obtains $C_{\mathrm{N}_{l,k}}$ subgroup functions.

To compute the subgroup functions reliably against noise, we apply sequences of nested lattice codes. For the node $\mathrm{N}_{l-1,i}$ in the $(l-1)$-th layer, the length-T_d data vector $\mathbf{s}_{l-1,i}$ is mapped to the length-n transmitted vector $\mathbf{x}_{l-1,i} = [x_{l-1,i}[1], x_{l-1,i}[2], \cdots, x_{l-1,i}[n]]$. Then, similar to Eq. (2), the length-n received vector $\mathbf{y}^{(c)}_{l,k}$ of the c-th subgroup at $\mathrm{N}_{l,k}$ is given as

$$y^{(c)}_{l,k}[m] = \sum_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}} v^{i\to k}_{l-1}[m]h^{i\to k}_{l-1}[m]x^{i\to k}_{l-1}[m] + w[m], \tag{8}$$

where $h^{i\to k}_{l-1}[m]$ is the channel from $\mathrm{N}_{l-1,i}$ to $\mathrm{N}_{l,k}$ at the m-th channel use, $x^{i\to k}_{l-1}[m]$ is the m-th element of the transmitted vector $\mathbf{x}_{l-1,i}$, $v^{i\to k}_{l-1}[m] = \frac{|h^{i\to k}_{l-1}[m]|}{h^{i\to k}_{l-1}[m]}\sqrt{P^{i\to k}_{l-1}[m]}$ is the power factor of $\mathrm{N}_{l-1,i}$, $P^{i\to k}_{l-1}[m]$ is the transmitted power of $\mathrm{N}_{l-1,i}$ and $w[m]$ is i.i.d. complex Gaussian random noise following $\mathcal{CN}(0,1)$.

After receiving $\mathbf{y}^{(c)}_{l,k}$, the decoding function is used to unmap the length-n received vector to T_d subgroup functions $\{f^{(c)}_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}})\}^{T_d}_{j=1}$.

C. Procedure of ML-CoMAC

Recalling the hierarchical network in Section III-A, we assume that the number of the overall layers is L, the number of nodes in each layer is K_l, and the data of $\mathrm{N}_{l,k}$ is denoted as $s_{l,k}$. For $l \geq 2$, each node $\mathrm{N}_{l,k}$ is assigned $C_{\mathrm{N}_{l,k}}$ subgroups. We show the recurrence relation among these (subgroup and group) functions over layers.

In the second layer, each node $\mathrm{N}_{2,k}$ computes $C_{\mathrm{N}_{2,k}}$ subgroup functions. Then, it reconstructs the group function as

$$\begin{aligned} &f_{2,k}\left(\left\{\{s_{1,i}\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{2,k}}}\right\}_{c\in\mathcal{C}_{\mathrm{N}_{2,k}}}\right)\\ &= g_{2,k}\left(\left\{f^{(c)}_{2,k}\left(\{s_{1,i}\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{2,k}}}\right)\right\}_{c\in\mathcal{C}_{\mathrm{N}_{2,k}}}\right). \end{aligned} \tag{9}$$

By setting the data $s_{2,k} = f_{2,k}(\{\{s_{1,i}\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{2,k}}}\}_{c\in\mathcal{C}_{\mathrm{N}_{2,k}}})$ for each node in $\mathcal{K}_2$, the node $\mathrm{N}_{3,k}$ in the third layer also

[4]Shown in Fig. 3b, a group is divided into several subgroups.

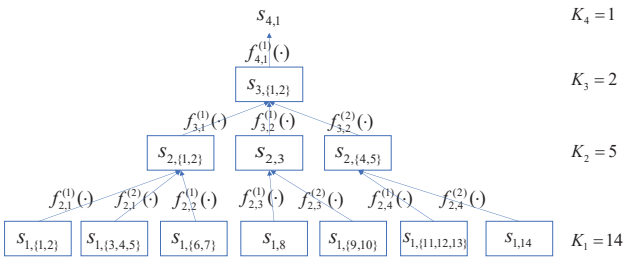


Fig. 4. An example of the implementation of ML-CoMAC.

reconstructs the corresponding group function as

$$\begin{aligned} &f_{3,k}\left(\left\{\left\{s_{2,i}\right\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{3,k}}}\right\}_{c\in\mathcal{C}_{\mathrm{N}_{3,k}}}\right) \\ &= g_{3,k}\left(\left\{f^{(c)}_{3,k}\left(\left\{s_{2,i}\right\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{3,k}}}\right)\right\}_{c\in\mathcal{C}_{\mathrm{N}_{3,k}}}\right). \end{aligned} \tag{10}$$

Thus, we can obtain the recurrence relation between the l-th layer and the $(l-1)$-th layer as

$$\begin{aligned} &f_{l,k}\left(\left\{\left\{s_{l-1,i}\right\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\right\}_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\right) \\ &= g_{l,k}\left(\left\{f^{(c)}_{l,k}\left(\left\{s_{l-1,i}\right\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\right)\right\}_{c\in\mathcal{C}_{\mathrm{N}_{l-1,k}}}\right), \end{aligned} \tag{11}$$

where $s_{l,k} = f_{l,k}(\{\{s_{l-1,i}\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\}_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}})$.

At the last layer, i.e., $l = L$, only including the fusion center, the desired function is finally computed because of

$$\begin{aligned} f_L\left(\left\{\left\{s_{L-1,i}\right\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_L}}\right\}_{c\in\mathcal{C}_{\mathrm{N}_L}}\right) &\overset{(a)}{=} f_L\left(\left\{s_{L-1,i}\right\}_{i\in\mathcal{K}_{L-1}}\right) \\ &\overset{(b)}{=} f_L\left(\left\{s_{L-2,i}\right\}_{i\in\mathcal{K}_{L-2}}\right) \\ &= \cdots \\ &\overset{(c)}{=} f_L\left(\left\{s_{1,i}\right\}_{i\in\mathcal{K}_1}\right), \end{aligned} \tag{12}$$

where the condition (a) follows since $K_L = 1$, the condition (b) follows because the values of $\{s_{L-1,i}\}_{i\in\mathcal{K}_{L-1}}$ are associated with $\{s_{L-2,i}\}_{i\in\mathcal{K}_{L-2}}$ and the condition (c) follows due to the recurrence relation Eq. (11). Eq. (12) shows that through computing and communicating subgroup and group functions, the desired function at the fusion center can be reliably reconstructed, which gives rise to ML-CoMAC.

Fig. 4[5] is an example of the hierarchical network with the given topology, which aims at computing the desired function $f\left(\{s_{1,i}\}_{i\in\mathcal{K}_1}\right)$ over 4 layers. We provide the implementation of ML-CoMAC as Algorithm 1.

[5]In Fig. 4, $\{s_{l,u}, s_{l,v}\}$ is denoted as $s_{l,\{u,v\}}$ for short.

Algorithm 1 Procedure of ML-CoMAC

1: **procedure** INITIALIZATION FOR SOURCE NODES
2: The sources nodes in $\mathcal{K}_1$ are divided into $\mathcal{K}_2$ groups.
3: Each group $\mathcal{K}_{\mathrm{N}_{2,k}}$ is allocated to the node $\mathrm{N}_{2,k}, \forall k \in \mathcal{K}_2$ in the second layer.
4: The data of $\mathrm{N}_{1,k}$ is $s_{1,k}$, which is sampled from the corresponding source.
5: **end procedure**
6: **procedure** COMPUTATION WITHIN GROUP
7: The given channel uses for the group $\mathcal{K}_{\mathrm{N}_{l,k}}$ belongs a set $\mathcal{T}_{l,k}$.
8: The group function needs to be computed at $\mathrm{N}_{l,k}$ during these channel uses.
9: To obtain this group function, $C_{\mathrm{N}_{l,k}}$ subgroup functions should first be computed at $\mathrm{N}_{l,k}$.
10: Given channel uses in $\mathcal{T}^{(c)}_{l,k}$, the subgroup function $f^{(c)}_{l,k}(\{s_{l-1,i}\}_{i\in\mathcal{K}^{(1)}_{\mathrm{N}_{l,k}}}$ is computed at the node $\mathrm{N}_{l,k}$ for each $c \in C_{\mathrm{N}_{l,k}}$ by using CoMAC.
11: After $|\mathcal{T}_{l,k}|$ channel uses, at $\mathrm{N}_{l,k}$, all subgroup functions are computed.
12: Using Definition 4, the group function $f_{l,k}(\{s_{l-1,i}\}_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}})$ can be reconstructed.
13: **end procedure**
14: **procedure** FROM ONE LAYER TO ANOTHER LAYER
15: Using the same steps in the procedure of computation within group (Line 6 in Algorithm 1), each group finishes the computation of the group function at the corresponding node $\mathrm{N}_{l,k}$.
16: **end procedure**
17: **procedure** INITIALIZATION FOR RELAY NODES
18: In the l-th layer ($l \geq 2$), the data of $\mathrm{N}_{l,k}$ is $s_{l,k}$ satisfying $s_{l,k} = f_{l,k}(\cdot)$.
19: All the nodes in $\mathcal{K}_l$ is divided into $\mathcal{K}_{l+1}$ groups that are allocated to the nodes in the $(l+1)$-th layer.
20: **end procedure**
21: **procedure** RECONSTRUCTION OF DESIRED FUNCTION
22: Using similar steps in the procedure of computation within group (Line 6 in Algorithm 1) and from one layer to another layer (Line 14 in Algorithm 1), the desired function $f(\mathbf{s}_1)$ is obtained at the fusion center using Definition 3.
23: **end procedure**

IV. ACHIEVABLE COMPUTATION RATES OF ML-COMAC

With the help of the hierarchical network, the analysis of the multi-hop network becomes tractable. With the procedure of ML-CoMAC, we derive the computation rate of ML-CoMAC based on the relationship between these different kinds of functions.

Eq. (11) shows that the computation rate of the desired function is determined by all the group functions over L layers, and each group function is reconstructed by the corresponding subgroup functions. Thus, we present the computation rates of the subgroup function, the group function and the desired function step by step.

Lemma 1 (Rate of Subgroup Function)**.** For a subgroup $\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}$ with $K^{(c)}_{\mathrm{N}_{l,k}}$ nodes, the computation rate of the subgroup function $f^{(c)}_{l,k}(\{s_{l-1,i}\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}})$ at the m-th channel use is given as

$$R^{(c)}_{l,k}[m]=\mathsf{C}^{+}\left(\frac{1}{K^{(c)}_{\mathrm{N}_{l,k}}}+\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\left[|h^{i\to k}_{l-1}[m]|^2P^{i\to k}_{l-1}[m]\right]\right), \tag{13}$$

where $|h^{i\to k}_{l-1}[m]|^2$ is the channel gain and $P^{i\to k}_{l-1}[m]$ is the transmitted power (see Eq. (8)).

Proof: Please refer to Eqs. (3) and (8) and [18, Theorem 3 and Section IV-A]. ∎

To reconstruct the group function $f_{l,k}(\{\{s_{l-1,i}\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\}_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}})$ during the given $|\mathcal{T}_{l,k}|$ channel uses where the set $\mathcal{T}_{l,k}$ includes the channel uses for $\mathrm{N}_{l,k}$, the subgroup functions should be computed first at different channel uses belonging to $\mathcal{T}_{l,k}$. Assume that the channel uses allocated to the corresponding subgroup $\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}$ are in a set $\mathcal{T}^{(c)}_{l,k}\subseteq\mathcal{T}_{l,k}$ satisfying $|\mathcal{T}^{(c)}_{l,k}|=\beta^{(c)}_{l,k}|\mathcal{T}_{l,k}|$ and $\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta^{(c)}_{l,k}=1$. After obtaining all the subgroup functions, the group function is reconstructed by Eq. (11). Thus, the computation rate of the group function is given as follows.

Theorem 1 (Rate of Group Function)**.** For any group $\mathcal{K}_{\mathrm{N}_{l,k}}$ with $C_{\mathrm{N}_{l,k}}$ subgroups, the computation rate of the group function reconstructed at $\mathrm{N}_{l,k}$ is

$$\begin{aligned}R_{l,k}&=\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\frac{\beta^{(c)}_{l,k}}{|\mathcal{T}^{(c)}_{l,k}|}\sum_{m\in\mathcal{T}^{(c)}_{l,k}}\left[\mathsf{C}^{+}\left(\frac{1}{K^{(c)}_{\mathrm{N}_{l,k}}}+\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\left[|h^{i\to k}_{l-1}[m]|^2P^{i\to k}_{l-1}[m]\right]\right)\right]\\&=\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta^{(c)}_{l,k}\mathsf{E}\left[\mathsf{C}^{+}\left(\frac{1}{K^{(c)}_{\mathrm{N}_{l,k}}}+\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\left[|h^{i\to k}_{l-1}|^2P^{i\to k}_{l-1}\right]\right)\right].\end{aligned} \tag{14}$$

Proof: Based on Lemma 1, the average computation rate

$$R^{(c)}_{l,k}=\frac{1}{|\mathcal{T}^{(c)}_{l,k}|}\sum_{m\in\mathcal{T}^{(c)}_{l,k}}R^{(c)}_{l,k}[m] \tag{15}$$

is achievable for computing the subgroup function $f^{(c)}_{l,k}(\{s_{l-1,i}[j]\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}})$ during $|\mathcal{T}^{(c)}_{l,k}|$ channel uses when $|\mathcal{T}_{l,k}|$ increases. Depending on Definition 2, the number of the values of the subgroup function computed during $|\mathcal{T}^{(c)}_{l,k}|$ channel uses is $U^{(c)}_{l,k}=\frac{R^{(c)}_{l,k}|\mathcal{T}^{(c)}_{l,k}|}{H(f(\mathbf{b_v}))}$. From Eq. (11), we can observe that the group function is reconstructed by $C_{\mathrm{N}_{l,k}}$ subgroup functions, which implies that the computation rate of the group function is determined by the rates of these subgroup functions. Since the number of the values of each subgroup function $U^{(c)}_{l,k}$ is different, only $U_{l,k}=\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}U^{(c)}_{l,k}$ group functions can be reconstructed. Hence, the computation rate based on Definition 2 to compute the group function $f_{l,k}(\{\{s_{l-1,i}\}_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\}_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}})$ is

$$\begin{aligned}R_{l,k}&=\lim_{n\to\infty}\frac{U_{l,k}}{|\mathcal{T}_{l,k}|}H(f(\mathbf{b_v}))\\&\overset{(a)}{=}\lim_{n\to\infty}\frac{\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}U^{(c)}_{l,k}}{|\mathcal{T}_{l,k}|}H(f(\mathbf{b_v}))\\&\overset{(b)}{=}\lim_{n\to\infty}\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\frac{R^{(c)}_{l,k}\left|\mathcal{T}^{(c)}_{l,k}\right|}{|\mathcal{T}_{l,k}|}\\&\overset{(c)}{=}\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta^{(c)}_{l,k}R^{(c)}_{l,k},\end{aligned} \tag{16}$$

where the condition (a) follows because of $U_{l,k}=\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}U^{(c)}_{l,k}$, the condition (b) follows because the expression of $U^{(c)}_{l,k}$ and the condition (c) follows due to $\frac{\left|\mathcal{T}^{(c)}_{l,k}\right|}{|\mathcal{T}_{l,k}|}=\beta^{(c)}_{l,k}$. ∎

To reconstruct the desired function $f\left(\{s_{1,k}\}_{k\in\mathcal{K}_1}\right)$ computed at the fusion center during n channel uses over L layers, the group allocated to $\mathrm{N}_{l,k}$ is active to compute the group function in the given channel uses in a set $\mathcal{T}_{l,k}$. Assume the number of the given channel uses is given as $|\mathcal{T}_{l,k}|=\alpha_{l,k}n$ satisfying $\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\alpha_{l,k}=1$. With the help of Theorem 1, the computation rate of the desired function in the hierarchical network with L layers is given as follows.

Theorem 2 (General Rate of ML-CoMAC)**.** For any $L\in\mathbb{N}$ satisfying $L\geq 2$, the computation rate of the desired function in the hierarchical network over fading MAC is given as

$$\begin{aligned}R&=\min_{l\in[2:L]}\min_{k\in\mathcal{K}_l}\alpha_{l,k}\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta^{(c)}_{l,k}\frac{1}{|\mathcal{T}^{(c)}_{l,k}|}\sum_{m\in\mathcal{T}^{(c)}_{l,k}}\left[\mathsf{C}^{+}\left(\frac{1}{K^{(c)}_{\mathrm{N}_{l,k}}}+\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\left[|h^{i\to k}_{l-1}[m]|^2P^{i\to k}_{l-1}[m]\right]\right)\right]\\&=\min_{l\in[2:L]}\min_{k\in\mathcal{K}_l}\alpha_{l,k}\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta^{(c)}_{l,k}\mathsf{E}\left[\mathsf{C}^{+}\left(\frac{1}{K^{(c)}_{\mathrm{N}_{l,k}}}+\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}}\left[|h^{i\to k}_{l-1}|^2P^{i\to k}_{l-1}\right]\right)\right].\end{aligned} \tag{17}$$

Proof: Theorem 1 suggests that the computation rate of the group function computed at $\mathrm{N}_{l,k}$ is $R_{l,k}$. However, to reconstruct the group function at $\mathrm{N}_{l,k}$, all the nodes in $\mathcal{K}_{\mathrm{N}_{l,k}}$ need to obtain the data vector first. In the hierarchical network with L layers, the data vector of $\mathrm{N}_{l-1,i}, i\in\mathcal{K}_{\mathrm{N}_{l,k}}$ is obtained by the values of the group function computed by the group $\mathcal{K}_{\mathrm{N}_{l-1,i}}$ (see Eq. (11)). Thus, when considering the relation between layers, the number of the values of the group function computed at $\mathrm{N}_{l,k}$ is determined by not only $\mathcal{K}_{\mathrm{N}_{l,k}}$ but also $\{\mathcal{K}_{\mathrm{N}_{l-1,i}}\}_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}}$, which is expressed as

$$\bar{U}_{l,k}=\min\left\{\frac{R_{l,k}|\mathcal{T}_{l,k}|}{H(f(\mathbf{b_v}))},\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}}\bar{U}_{l-1,i}\right\}. \tag{18}$$

For the sake of simplicity, we denote $\frac{R_{l,k}|\mathcal{T}_{l,k}|}{H(f(\mathbf{b_v}))}$ as $\rho_{l,k}$. Based on the recurrence relation Eq. (18), at the fusion center ($l = L$), the number of the values of the desired function is

$$\begin{aligned}\bar{U}_{L,1} &= \min\left\{\rho_{L,1}, \min_{i\in\mathcal{K}_{\mathrm{N}_{L,1}}} \bar{U}_{l-1,i}\right\}\\ &\overset{(a)}{=} \min\left\{\rho_{L,1}, \min_{i_1\in\mathcal{K}_{L-1}} \min\left\{\rho_{L-1,i_1}, \min_{i_2\in\mathcal{K}_{\mathrm{N}_{L-1,i_1}}} \bar{U}_{L-2,i_2}\right\}\right\}\\ &\overset{(b)}{=} \min\left\{\rho_{L,1}, \min_{i_1\in\mathcal{K}_{L-1}} \rho_{L-1,i_1}, \min_{i_1\in\mathcal{K}_{L-1}} \min_{i_2\in\mathcal{K}_{\mathrm{N}_{L-1,i_1}}} \bar{U}_{L-2,i_2}\right\}\\ &\overset{(c)}{=} \min\left\{\rho_{L,1}, \min_{i_1\in\mathcal{K}_{L-1}} \rho_{L-1,i_1}, \min_{i_1\in\mathcal{K}_{L-2}} \bar{U}_{L-2,i_2}\right\}\\ &= \min_{l\in[2:L]} \min_{k\in\mathcal{K}_l} \rho_{l,k},\end{aligned} \tag{19}$$

where the condition (a) follows because of $K_L = 1$ and Eq. (18), the condition (b) follows since $\min$ operation is associative and the condition (c) follows due to $\cup_{i_1\in\mathcal{K}_{L-1}}\mathcal{K}_{\mathrm{N}_{L-1,i_1}} = \mathcal{K}_{L-2}$ (see Definition 3).

At last, the fusion center computes $\bar{U}_{L,1}$ desired functions over L layers. And, the computation rate of the desired function in the hierarchical network is given as

$$\begin{aligned}R &= \lim_{n\to\infty} \frac{\bar{U}_{L,1}}{n} H(f(\mathbf{b_v}))\\ &\overset{(a)}{=} \lim_{n\to\infty} \frac{\min_{l\in[2:L]} \min_{k\in\mathcal{K}_l} \rho_{l,k}}{n} H(f(\mathbf{b_v}))\\ &\overset{(b)}{=} \lim_{n\to\infty} \min_{l\in[2:L]} \min_{k\in\mathcal{K}_l} \frac{R_{l,k}|\mathcal{T}_{l,k}|}{n}\\ &\overset{(c)}{=} \min_{l\in[2:L]} \min_{k\in\mathcal{K}_l} \alpha_{l,k} R_{l,k},\end{aligned} \tag{20}$$

where the condition (a) follows because of Eq. (18), the condition (b) follows due to $\rho_{l,k} = \frac{R_{l,k}|\mathcal{T}_{l,k}|}{H(f(\mathbf{b_v}))}$ and the condition (c) follows as $|\mathcal{T}_{l,k}| = \alpha_{l,k} n$. ∎

The rate of Theorem 2 considers the general case and can reduce to the rate in the single-hop network by setting $L = 2$. Based on the general rate, we can apply different resource allocation to analyze the corresponding rate and to improve the performance.

V. Optimal Resource Allocation

The derived general computation rate suggests that the performance is determined by the allocated channel uses to each subgroup or each group and the power to each node. To fully use the flexibility of the total channel uses and the power of each node, we consider optimal time allocation and fixed power control as a simple case in Section V-A, and optimal time allocation and adaptive power control in Section V-B.

A. Optimal Time Allocation and Fixed Power Control

Considering the fixed power constraint for each user, we obtain the computation rate from Theorem 2 easily as

$$\begin{aligned}R &= \min_{l\in[2:L]} \min_{k\in\mathcal{K}_l} \alpha_{l,k} \min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}} \beta_{l,k}^{(c)} \frac{1}{|\mathcal{T}_{l,k}^{(c)}|} \sum_{m\in\mathcal{T}_{l,k}^{(c)}} \left[\mathsf{C}^+\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}} |h_{l-1}^{i\to k}[m]|^2 P\right)\right]\\ &\overset{(a)}{\le} \min_{l\in[2:L]} \min_{k\in\mathcal{K}_l} \alpha_{l,k} \min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}} \beta_{l,k}^{(c)} \mathsf{C}^+\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}} |h_{l-1}^{i\to k}|^2\right] P\right)\end{aligned} \tag{21}$$

by setting $P_{l-1}^{i\to k}[m] = P$, where the condition (a) follows because of the increase in n and Jensen's inequality.

One can observe that each $\alpha_{l,k}$ and each $\beta_{l,k}^{(c)}$ should be optimized to approach the optimal computation rate since the computation rate of each subgroup function is different. A subgroup function with higher computation rate should be allocated fewer channel uses as the number of the desired functions computed at the fusion center is determined by the minimum of the number of each subgroup function. Thus, we formulate the following optimization problem.

Problem 1.

$$\begin{aligned}\underset{\alpha_{l,k},\beta_{l,k}^{(c)}}{\text{maximize}} \quad & \min_{l\in[2:L]} \min_{k\in\mathcal{K}_l} \alpha_{l,k} \min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}} \beta_{l,k}^{(c)} \mathsf{C}^+\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}} |h_{l-1}^{i\to k}|^2\right] P\right)\\ \text{s.t.} \quad & \sum_{l=2}^{L} \sum_{k\in\mathcal{K}_l} \alpha_{l,k} = 1 \qquad (22)\\ & \sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}} \beta_{l,k}^{(c)} = 1, \forall l \in [2:L], \forall k \in \mathcal{K}_l \qquad (23)\end{aligned}$$

Although the objective function is non-convex, it can be transformed into a convex function by relaxing the bi-linear function through McCormick relaxation [25]. We introduce $p_{l,k}^{(c)} = \alpha_{l,k}\beta_{l,k}^{(c)}$. Then, the constraints (22) and (23) can be jointly rewritten as $\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}} p_{l,k}^{(c)} = 1$. Therefore, the $\max-\min$ problem can be reformed as

Problem 2.

$$\begin{aligned}\underset{p_{l,k}^{(c)},t}{\text{maximize}} \quad & t \\ \text{s.t. } & p_{l,k}^{(c)}\mathsf{C}^{+}\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}}+\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\right)\geq t, \\ & \forall l\in[2:L],\forall k\in\mathcal{K}_l,\forall c\in\mathcal{C}_{l,k} \quad (24) \\ & \sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}p_{l,k}^{(c)}=1\end{aligned}$$

Since Problem 2 is a linear programming problem, the problem can be solved by the interior-point methods or Lagrangian duality approach [26]. However, such an optimal solution requires iteratively updating Lagrange multipliers using sub-gradient methods. By exploring the special structure of Problem 2, we obtain a simple optimal solution that does not require iterations. The optimal $\left\{p^{*}{}_{l,k}^{(c)}\right\}_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}$ and t^* can be obtained as closed-form expressions though the Lagrangian function

$$\begin{aligned}\mathcal{L}=t-\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\lambda_{l,k}^{(c)}\Bigg[t-p_{l,k}^{(c)}\mathsf{C}^{+}\Bigg(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} \\ +\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\Bigg)\Bigg] \quad (25) \\ -\mu\Big(\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}p_{l,k}^{(c)}-1\Big),\end{aligned}$$

where $\left\{\lambda_{l,k}^{(c)}\right\}$ and μ are Lagrange multipliers.

By setting the first derivative of $\mathcal{L}$ with respect to t, we have $\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\lambda_{l,k}^{(c)}=1$ with the complementary slackness condition for all $c\in\mathcal{C}_{\mathrm{N}_{l,k}}$

$$\lambda_{l,k}^{(c)}\left[t-p_{l,k}^{(c)}\mathsf{C}^{+}\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}}+\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\right)\right]=0. \quad (26)$$

Also, by setting the first derivative of $\mathcal{L}$ with respect to $p_{l,k}^{(c)}$ for all $c\in\mathcal{C}_{\mathrm{N}_{l,k}}$, we have

$$\lambda_{l,k}^{(c)}\mathsf{C}^{+}\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}}+\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\right)-\mu=0 \quad (27)$$

with the complementary slackness condition $\mu(\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}p_{l,k}^{(c)}-1)=0$.

From Eq. (27), one can observe that $\lambda_{l,k}^{(c)}=0,\forall c\in\mathcal{C}_{\mathrm{N}_{l,k}}$ if $\mu=0$, which is contrary to $\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\lambda_{l,k}^{(c)}=1$. Thus, to obtain the optimal solution, $\mu\neq 0$ should hold. For each c in $\mathcal{C}_{\mathrm{N}_{l,k}}$, $p_{l,k}^{(c)}\mathsf{C}^{+}\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}}+\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\right)$ should be the same and equal to t due to $\mu\neq 0$, $\lambda_{l,k}^{(c)}\neq 0$ and Eq. (26). Using $\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\lambda_{l,k}^{(c)}=1$ and

$$p_{l,k}^{(c)}=\frac{t}{\mathsf{C}^{+}\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}}+\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\right)}, \quad (28)$$

the optimal t^* is given as

$$\begin{aligned}t^{*}=\Bigg[\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\Bigg[\mathsf{C}^{+}\Bigg(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}}+ \\ \mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\Bigg)\Bigg]^{-1}\Bigg]^{-1}\end{aligned} \quad (29)$$

and the optimal $p^{*}{}_{l,k}^{(c)}$ is given as

$$p^{*}{}_{l,k}^{(c)}=\frac{t^{*}}{\mathsf{C}^{+}\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}}+\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P\right)}. \quad (30)$$

As a result, the computation rate with optimal time allocation and fixed power control is given as t^* (Eq. (29)).

Remark 6 (Special Cases)**.** By setting $L=2$, $K_2=1$ and $C_{\mathrm{N}_{2,1}}=1$ in Eq. (29), it reduces to a simple case where K_1 nodes wish to compute a desired function at the fusion center directly as classical CoMAC mentioned in Section II-B, and the rate of it, named the rate of CoMAC with fixed power control, is the same as Eq. (3) [18]. Also, by setting $L=2$, $K_2=1$, $C_{\mathrm{N}_{2,1}}=K_1$, and $K_{\mathrm{N}_{2,1}}^{(c)}=1$ in Eq. (29), it reduces to the time-sharing case as Eq. (4).

B. Optimal Time Allocation and Adaptive Power Control

We observe that each node in the hierarchical network is active only in the corresponding channel uses. To compute the functions more efficiently, long-term power control should be considered as $\mathsf{E}\left[P_{l-1}^{i\to k}[m]\right]=P$. The transmitted power of each node is set to

$$P_{l-1}^{i\to k}[m]=\begin{cases}c\dfrac{\min_{j\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{j\to k}[m]|^2}{|h_{l-1}^{i\to k}[m]|^2} & ,m\in\mathcal{T}_{l,k}^{(c)} \\ 0 & ,\text{otherwise}\end{cases}. \quad (31)$$

To satisfy the long-term power control constrain, we have

$$\begin{aligned}\mathsf{E}\left[P_{l-1}^{i\to k}[m]\right]&=\sum_{t=1}^{n}\Pr(m=t)P_{l-1}^{i\to k}[m]|_{m=t} \\ &\overset{(a)}{=}\frac{c}{n}\sum_{t\in\mathcal{T}_{l,k}^{(c)}}\frac{\min_{j\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{j\to k}[t]|^2}{|h_{l-1}^{i\to k}[t]|^2} \quad , \quad (32) \\ &\overset{(b)}{=}c\alpha_{l,k}\beta_{l,k}^{(c)}\mathsf{E}\left[\frac{\min_{j\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{j\to k}|^2}{|h_{l-1}^{i\to k}|^2}\right]\end{aligned}$$

which should be equal to P. Then, c is obtained as

$$c = \frac{P}{\alpha_{l,k}\beta_{l,k}^{(c)}\mathsf{E}\left[\frac{\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2}{|h_{l-1}^{j\to k}|^2}\right]}. \tag{33}$$

Substituting Eqs. (31) and (33) into the rate in Theorem 2, the computation rate is expressed as

$$R = \min_{l\in[2:L]}\min_{k\in\mathcal{K}_l}\alpha_{l,k}\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta_{l,k}^{(c)}\mathsf{C}^+\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \frac{\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P}{\alpha_{l,k}\beta_{l,k}^{(c)}\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2/|h|^2\right]}\right), \tag{34}$$

where h is used as a representative coefficient without loss of generality.

Considering adaptive power control, we formulate an optimization problem as Problem 3 to maximize the computation rate.

Problem 3.

$$\begin{aligned}\underset{\alpha_{l,k},\beta_{l,k}^{(c)}}{\text{maximize}} \quad & \min_{l\in[2:L]}\min_{k\in\mathcal{K}_l}\alpha_{l,k}\min_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta_{l,k}^{(c)}\mathsf{C}^+\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \frac{\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P}{\alpha_{l,k}\beta_{l,k}^{(c)}\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2/|h|^2\right]}\right)\\ \text{s.t.} \quad & \sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\alpha_{l,k} = 1 \qquad (35)\\ & \sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\beta_{l,k}^{(c)} = 1, \forall l\in[2:L], \forall k\in\mathcal{K}_l \qquad (36)\end{aligned}$$

By introducing the convex relaxation as $p_{l,k}^{(c)} = \alpha_{l,k}\beta_{l,k}^{(c)}$, this problem is rewritten as the following form.

Problem 4.

$$\begin{aligned}\underset{p_{l,k}^{(c)},t}{\text{maximize}} \quad & t\\ \text{s.t. } & p_{l,k}^{(c)}\mathsf{C}^+\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \frac{\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P}{p_{l,k}^{(c)}\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2/|h|^2\right]}\right) \geq t,\\ & \qquad \forall l\in[2:L], \forall k\in\mathcal{K}_l, \forall c\in\mathcal{C}_{l,k} \qquad (37)\\ & \sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}p_{l,k}^{(c)} = 1\end{aligned}$$

Problem 4 now is a convex problem since the constrain, Eq. (37), is concave. Hence, the above optimization problem has a unique maximum. The Lagrangian function is given as

$$\begin{aligned}\mathcal{L} =& t - \sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}\lambda_{l,k}^{(c)}\left[t - p_{l,k}^{(c)}\mathsf{C}^+\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \frac{\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P}{p_{l,k}^{(c)}\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2/|h|^2\right]}\right)\right]\\ & - \mu\left(\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}p_{l,k}^{(c)} - 1\right)\end{aligned} \tag{38}$$

with the complementary slackness condition

$$\mu\left(\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}p_{l,k}^{(c)} - 1\right) = 0, \tag{39}$$

where $\left\{\lambda_{l,k}^{(c)}\right\}$ and μ are Lagrange multipliers.

We apply the KKT optimality conditions to the Lagrangian function to obtain the optimal factor ${p^*}_{l,k}^{(c)}$. By setting the first derivative of $\mathcal{L}$ as Eq. (38) with respect to $p_{l,k}^{(c)}$ to zero, we have

$$\ln\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \frac{\varepsilon_{l,k}^{(c)}}{p_{l,k}^{(c)}}\right) - \frac{\varepsilon_{l,k}^{(c)}}{p_{l,k}^{(c)}\left(\frac{1}{K_{\mathrm{N}_{l,k}}^{(c)}} + \frac{\varepsilon_{l,k}^{(c)}}{p_{l,k}^{(c)}}\right)} = \frac{\mu\ln(2)}{\lambda_{l,k}^{(c)}}, \tag{40}$$

where $\varepsilon_{l,k}^{(c)} = \frac{\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2\right]P}{\mathsf{E}\left[\min_{i\in\mathcal{K}_{\mathrm{N}_{l,k}}^{(c)}}|h_{l-1}^{i\to k}|^2/|h|^2\right]}$.

Then, each optimal factor is expressed as

$${p^*}_{l,k}^{(c)} = \max\left\{0, -\varepsilon_{l,k}^{(c)}K_{\mathrm{N}_{l,k}}^{(c)}\left[1 + \left(\tau_{l,k}^{(c)}\right)^{-1}\right]^{-1}\right\}, \tag{41}$$

where $\tau_{l,k}^{(c)}$ is a Lambert W function as

$$\tau_{l,k}^{(c)} = W\left(-2^{-\frac{\mu}{\lambda_{l,k}^{(c)}}}\left(K_{\mathrm{N}_{l,k}}^{(c)}\right)^{-1}\exp(-1)\right), \tag{42}$$

while ${p^*}_{l,k}^{(c)}$ satisfies

$$\begin{cases}\sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}{p^*}_{l,k}^{(c)} \leq 1, & \mu = 0\\ \sum_{l=2}^{L}\sum_{k\in\mathcal{K}_l}\sum_{c\in\mathcal{C}_{\mathrm{N}_{l,k}}}{p^*}_{l,k}^{(c)} = 1, & \mu > 0\end{cases}. \tag{43}$$

Remark 7 (Special Cases)**.** By setting $L = 2$, $K_2 = 1$ and $C_{\mathrm{N}_{2,1}} = 1$ in Eq. (34), the rate of it is the same as the rate

$$R = \mathsf{C}^+\left(\frac{1}{K_1} + \frac{\mathsf{E}\left[\min_{i\in[1:K_1]}|h_{1,i}|^2\right]P}{\mathsf{E}\left[\min_{i\in[1:K_1]}|h_{1,i}|^2/|h|^2\right]}\right) \tag{44}$$

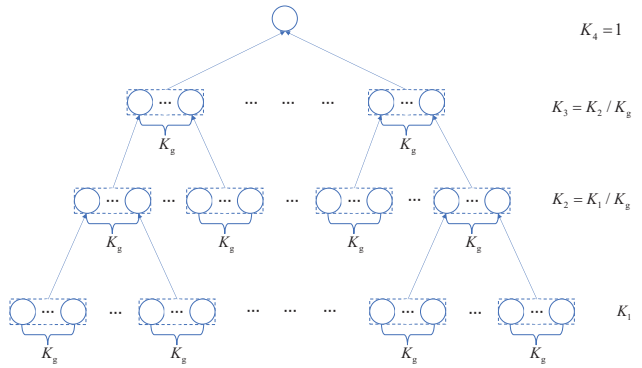


Fig. 5. The simulated topology.

in [18, Theorem 3 and Section VII-C] as the rate of CoMAC with adaptive power control. Also, by setting $L = 2$, $K_2 = 1$, $C_{\mathrm{N}_{2,1}} = K_1$, $K^{(c)}_{\mathrm{N}_{2,1}} = 1$ and $\beta^{(c)}_{l,k} = \frac{1}{K_1}$ in Eq. (34) as the time-sharing case, an improved rate is obtained as $R = \frac{1}{K_1}\mathsf{E}\left[\mathsf{C}\left(|h|^2 K_1 P\right)\right]$ compared with Eq. (4).

We provide a low-overhead signaling procedure so that the resource allocation can be carried out without collecting the global channel state information (CSI).

Algorithm 2 Global-CSI-Free Signaling Procedure for Resource Allocation

1: **procedure** Global-CSI-Free Signaling for Resource Allocation
2: Each node $\mathrm{N}_{l-1,i}$ collects its statistical CSI $\mathsf{E}\left[|h^{i\to k}_{l-1}|^2\right]$ only once for $i \in \mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}, c \in \mathcal{C}_{\mathrm{N}_{l,k}}, k \in \mathcal{K}_l, l \in [2:L-1]$.
3: Each relay $\mathrm{N}_{l,k}$ collects $C_{\mathrm{N}_{l,k}}$ statistical channel parameters $\mathsf{E}\left[\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}} |h^{i\to k}_{l-1}|^2\right]$ only once for $k \in \mathcal{K}_l, l \in [2:L-1]$
4: The above statistical parameters are transmitted to the fusion center one by one only once.
5: The fusion center calculates the optimal t^* through Eq. (29) (for fixed power control) or Problem 4 (for adaptive power control).
6: t^* is broadcast from one layer to another using L channel uses.
7: Each node obtains the optimal time factor $p^{*(c)}_{l,k}$ through Eq. (30) (for fixed power control) or the equation form of the inequality (37) (for adaptive power control).
8: Broadcasting a pilot by either the relay or the fusion center, each node can obtain its own CSI at the same time.
9: The power of each node is set to P (for fixed power control) or Eq. (31) (for adaptive power control).
10: **end procedure**

VI. Simulation Results and Discussion

In this section, we provide simulation results of the computation rates of ML-CoMAC, the time-sharing scheme Eq. (4), CoMAC with fixed power control Eq. (3), and CoMAC with adaptive power control Eq. (44). In our simulation, the average signal-to-noise ratio (SNR) is the same as P because the variance of the noise is set as one. We consider i.i.d. Rayleigh fading channel, i.e., the exponential distribution with parameter one. The abbreviations for fixed power control, adaptive power control, average time allocation, and optimal time allocation are FPC, APC, ATA, and OTA, respectively.

TABLE I
Parameters Definitions

Parameters	
$L \in \{2,3,4\}$	The number of layers
K_{g}	The number of nodes in each group
$K_{\mathrm{sub-g}}$	The number of nodes in each subgroup
K_1	The number of nodes in the first layer
$K_L = 1$	The number of nodes in the L-th layer
$K_l = \max\{1, K_{l-1}/K_g\}$, $\forall l \in [2:L-1]$	The number of nodes in the l-th layer
$N_{l,k}$	The k-th node in the l-th layer
$C_{N_{l,k}} = K_{\mathrm{g}}/K_{\mathrm{sub-g}}$	The number of subgroups for $N_{l,k}$

A. Simulated Topology Set Up

The simulated topology is given as Fig. 5 and the detailed parameters are given in Tab. I. The key parameters of this hierarchical network are the number K_{g} of nodes in each group and the number $K_{\mathrm{sub-g}}$ of nodes in each sub-group. In an existing multi-hop network, grouping is usually determined by topology parameters (i.e., K_{g} is determined by how many nodes each relay can hear) whereas sub-grouping within each group can be arbitrary since it is a topology-protection operation, i.e., sub-grouping divides the group into several sub-groups in protocol not in topology. A general guideline on how to construct groups and sub-groups is given as follows.

- **How to Construct Groups**: The groups are constructed based on the topology parameters. In other words, Ml-CoMAC is the efficient function-computation method for the existing multi-hop networks without topology modification since such a change in the topology can be very capital intensive and in many cases seems not easy to implement for existing networks. Each relay can hear the distributed nodes within a common transmission range and these distributed nodes that can be heard are regarded as a group assigned to that relay [7].
- **How to Construct Subgroups**: Since sub-grouping within each group can be arbitrary and is a topology-protection operation, we provide a (sub)optimal way to construct the subgroups, which can achieve a higher computation rate for each group. Recall the rate $R_{l,k}$ of group function (see Theorem 1) computed by the relay $\mathrm{N}_{l,k}$. Considering the independent and identically distributed (i.i.d.) Rayleigh fading, $\mathsf{E}\left[\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}} \left[|h^{i\to k}_{l-1}|^2 P^{i\to k}_{l-1}\right]\right] = \frac{P}{K^{(c)}_{\mathrm{N}_{l,k}}}$ holds since the average power of each node is the same as $\mathsf{E}\left[P^{i\to k}_{l-1}\right] = P$ and $\mathsf{E}[\min_{i\in\mathcal{K}^{(c)}_{\mathrm{N}_{l,k}}} [|h^{i\to k}_{l-1}|^2]]$ is equal to $1/K^{(c)}_{\mathrm{N}_{l,k}}$. We formulate the optimization problem to maximize the rate

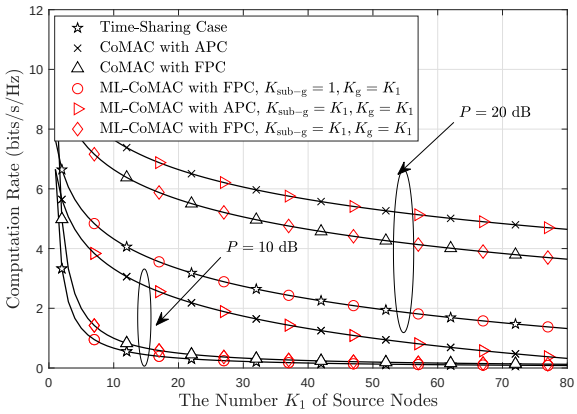


Fig. 6. Computation rates of CoMAC with different schemes with respect to the number K_1 of source nodes and P when $L = 2$.

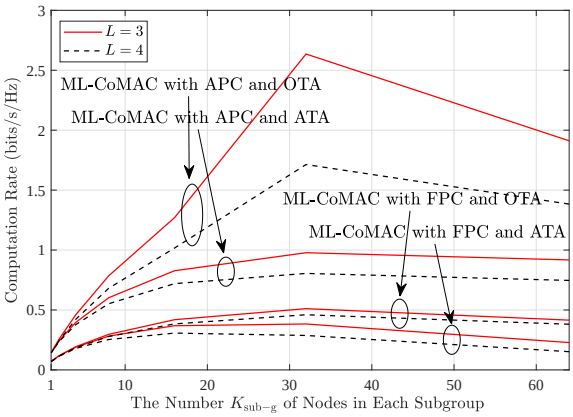


Fig. 7. Computation rates of ML-CoMAC with different schemes with respect to the number $K_{\text{sub-g}}$ of nodes in each subgroup and the number L of layers when K_1=64 and $K_{\text{g}} = 64$.

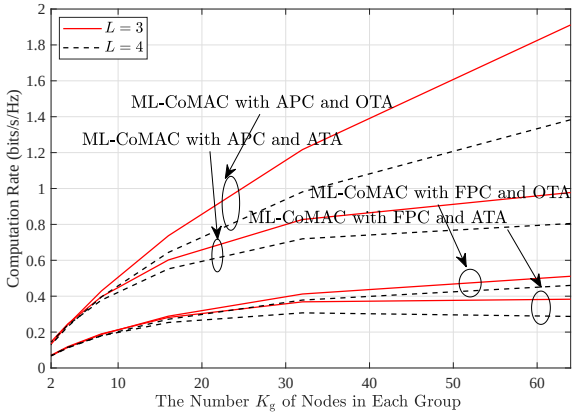


Fig. 8. Computation rates of ML-CoMAC with different schemes with respect to the number K_{g} of nodes in each group and the number L of layers when $K_1 = 64$ and $K_{\text{sub-g}} = K_{\text{g}}/2$.

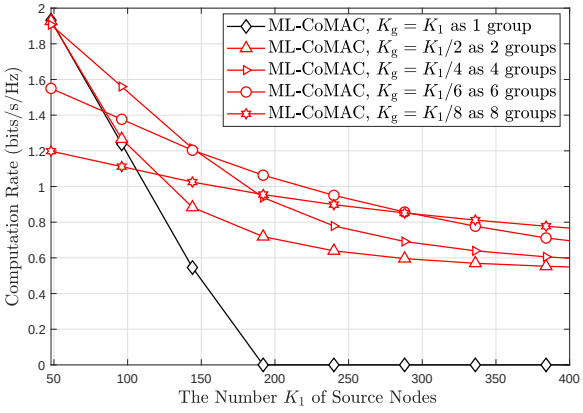


Fig. 9. Computation rates of ML-CoMAC with respect to the number K_{g} of nodes in each group and the number K_1 of source nodes when $L = 3$.

by finding the optimal $C_{\text{N}_{l,k}}$, $\beta_{l,k}^{(c)}$, and $K_{\text{N}_{l,k}}^{(c)}$. Since $R_{l,k}$ is not concave, to find some insights, we set the time factor $\beta_{l,k}^{(c)} = K_{\text{N}_{l,k}}^{(c)}/K$ to relax it. This is because the subgroup with more nodes has a slower computing rate and needs more time slots to compute the subgroup function in such a $\max-\min$ problem. Since the objective function is now concave, under the case without constraints, the optimal $K_{\text{N}_{l,k}}^{(c)}$ can be obtained directly as $K_{\text{N}_{l,k}}^{(c)} = (1+P)/e$. This implies that the optimal $K_{\text{N}_{l,k}}^{(c)}$ is always the same with each other subgroup within the group. Thus, the insightful solution considering the node constraint $\sum_{c=1}^{C_{\text{N}_{l,k}}} K_{\text{N}_{l,k}}^{(c)} = K_{\text{N}_{l,k}}$ is given as

$$C_{\text{N}_{l,k}}{}^{*} = \left\lfloor \frac{K_{\text{N}_{l,k}}}{\min\left\{\left\lceil \frac{1+P}{e} \right\rceil, K_{\text{N}_{l,k}}\right\}} - \frac{1}{2} \right\rceil, \qquad (45)$$

where $\lceil\cdot\rceil$ is the ceiling function. This solution provides a straightforward way to sub-grouping that can achieve a improved computation rate.

In our simulation, to simply the topology, we consider the case where the number of nodes in each subgroup is uniform and the number of nodes in each group is also uniform. Still, the influence of these main parameters on various aspects of the network topology can be shown.

B. Computation Rates

Since the multi-hop network is a more general case compared with the single-hop network, the rates of CoMAC Eqs. (4), (3) and (44) should be generalized by the rates of ML-CoMAC with specified parameters. Thus, in Fig. 6, their relationship is given. By setting $L = 2$, the multi-hop network reduces to the single-hop network aiming at computing the desired function associated with K_1 source nodes directly. When $K_{\text{sub-g}} = 1$ and $K_{\text{g}} = K_1$, the fusion center collects all the individual data from K_1 nodes as the time-sharing case and the rate of ML-CoMAC with FPC is the same as Eq. (4) by setting $C_{\text{N}_{2,1}} = K_{\text{g}}/K_{\text{sub-g}} = K_1$

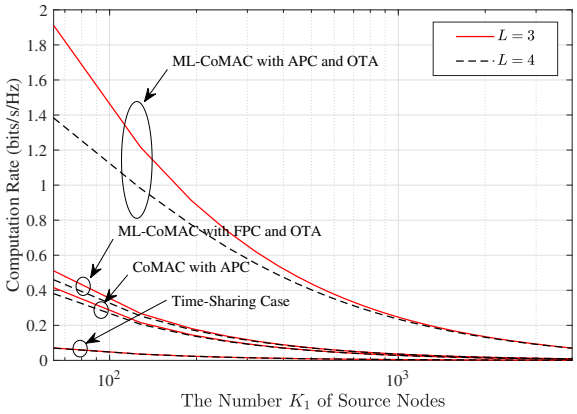


Fig. 10. Comparison between ML-CoMAC, time-sharing, and CoMAC with respect to the number K_1 of source nodes and the number L of layers when $K_g = 64$ and $K_{\mathrm{sub-g}} = K_g/2$.

in Eq. (29). When $K_{\mathrm{sub-g}} = K_1$ and $K_{\mathrm{g}} = K_1$, by setting $C_{\mathrm{N}_{2,1}} = K_{\mathrm{g}}/K_{\mathrm{sub-g}} = 1$ in Eq. (29), all the nodes transmit signals simultaneously to the fusion center. It generalizes the rate of CoMAC with FPC Eq. (3). Similarly, by setting $C_{\mathrm{N}_{2,1}} = 1$ in Eq. (34), the rate of CoMAC with APC Eq. (44) is obtained.

The computation rates of ML-CoMAC with different schemes versus the number $K_{\mathrm{sub-g}}$ of nodes in each subgroup and the number L of layers are demonstrated in Fig. 7. One can observe that the computation rate decreases as the number of layers increases. Since each layer has to be allocated some channel uses to compute the corresponding functions, i.e., subgroup functions and group functions, the increase in the number of layers causes the decrease in the number of channel uses allocated to each layer when the total of channel uses is fixed. Besides, the computation rate is improved by setting $K_{\mathrm{sub-g}} = 32$. This optimal $K_{\mathrm{sub-g}}$ also can be obtained from Eq. (45) by setting $K_{\mathrm{N}_{l,k}} = K_{\mathrm{g}} = 64$ and $P = 100$. This verifies the correctness of Eq. (45) and implies that the group function divided into several subgroup functions will improve the performance. Compared with ML-CoMAC with FPC, ML-CoMAC with APC improves the rate. Also, optimal time allocation provides further improvement.

However, the impact of the number K_{g} of nodes in each group is different from the impact of the number of subgroups. In Fig. 8, we show the computation rates of ML-CoMAC for different schemes versus K_{g} and L. The main difference from Fig. 7 is that the increase in the groups results in the worse performance since each group is allocated fewer channel uses. With fewer channel uses, the number of the group functions computed at the corresponding node is fewer. Thus, the computation rate of ML-CoMAC decreases.

Although Fig. 8 suggests that, with the fixed K_1, the number of groups in a network should be as few as possible, it does not mean that the increase in the number of groups only has the disadvantage. As shown in Fig. 9, we simulate the computation rates with respect to K_{g} and K_1. One can observe that all the rates decrease as the number of source nodes K_1 increases. Also, when K_1 is small, the relation between the rate and the number of groups is the same as that in Fig. 8. However, as K_1 becomes larger, unlike the rate of ML-CoMAC with one group decreasing rapidly, the rates of ML-CoMAC with multiple groups keep a slower decrease. Especially, ML-CoMAC with eight groups provides the slowest decrease, which implies that ML-CoMAC with more groups can support a network with more nodes. Thus, it provides a way to design a network that can afford massive numbers of nodes by increasing the number of groups in this network.

Fig. 10 shows the computation rates with different schemes, i.e., ML-CoMAC, time-sharing, and CoMAC, versus K_1 and L. For the implementation of the conventional CoMAC, we consider the case that each group computes the group function directly without sub-grouping to provide an approximate CoMAC rate since the conventional CoMAC cannot be directly implemented in the multi-hop network. For time-sharing, each node transmits its data one by one during the different channel uses. For the proposed ML-CoMAC, we choose the sub-grouping strategy (i.e., $K_{\mathrm{sub-g}} = K_g/2$). One can observe that ML-CoMAC can provide higher rates in the multi-hop network, and the rate of ML-CoMAC with APC and OTA provides the highest rate compared with the others. This implies that the sub-grouping in ML-CoMAC can not only improve the performance but also protect the existing topology. Also, the rate of time-sharing is vanishing rapidly, which implies that orthogonal communication to compute functions is not suitable for a multi-hop network.

VII. Conclusion

In this paper, we have combined the uses of CoMAC and orthogonal communication to attain the implementation of CoMAC in the multi-hop network. ML-CoMAC has been developed based on the hierarchized multi-hop network including multiple layers that consists of subgroups and groups. By computing and communicating subgroup functions and group functions over layers, the desired function is reconstructed at the fusion center. Based on the proposed scheme, we have derived the general computation rate, which suggests that the computation rate is determined by the subgroup function with the worst rate. Furthermore, we have formulated optimization problems taking time allocation and power control into account. The closed-form optimal solutions have been given with respect to different cases, which generalizes the existing CoMAC works.

References

[1] G. Fettweis and S. Alamouti, "5G: Personal mobile internet beyond what cellular did to telephony," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 140–145, 2014.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] M. Goldenbaum, H. Boche, and S. Stańczak, "Nomographic functions: Efficient computation in clustered Gaussian sensor networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2093–2105, 2015.

[4] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[5] A. Orlitsky and J. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 903–917, 2001.

[6] L.-L. Xie and P. R. Kumar, "A network information theory for wireless communication: Scaling laws and optimal operation," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 748–767, 2004.

[7] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 755–764, 2005.

[8] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," *arXiv preprint arXiv:1612.02307*, 2016.

[9] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multimodal sensing," *IEEE IoT J.*, vol. 6, no. 4, pp. 6089–6103, 2018.

[10] M. Goldenbaum and S. Stanczak, "On the channel estimation effort for analog computation over wireless multiple-access channels," *IEEE Wireless Commun. Lett.*, vol. 3, no. 3, pp. 261–264, 2014.

[11] J. Dong, Y. Shi, and Z. Ding, "Blind over-the-air computation and data fusion via provable wirtinger flow," *IEEE Trans. Signal Process.*, vol. 68, pp. 1136–1151, 2020.

[12] A. Kortke, M. Goldenbaum, and S. Stańczak, "Analog computation over the wireless channel: A proof of concept," in *SENSORS, 2014 IEEE*. IEEE, 2014, pp. 1224–1227.

[13] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Over-the-air computation for cooperative wideband spectrum sensing and performance analysis," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10 603–10 614, 2018.

[14] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, 2007.

[15] R. Appuswamy and M. Franceschetti, "Computing linear functions by linear coding over networks," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 422–431, 2014.

[16] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3401–3416, 2005.

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

[18] S.-W. Jeon, C.-Y. Wang, and M. Gastpar, "Computation over Gaussian networks with orthogonal components," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7841–7861, 2014.

[19] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Communicating or computing over the mac: Function-centric wireless networks," *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6127–6138, 2019.

[20] S.-W. Jeon and B. C. Jung, "Opportunistic function computation for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4045–4059, 2016.

[21] F. Wu, L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Computation over wide-band multi-access channels: Achievable rates through sub-function allocation," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3713–3725, 2019.

[22] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless networks*, vol. 11, no. 4, pp. 471–487, 2005.

[23] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 736–748, 2003.

[24] M. Goldenbaum, "Computation of real-valued functions over the channel in wireless sensor networks," Ph.D. dissertation, Technische Universität München, 2014.

[25] A. Mitsos, B. Chachuat, and P. I. Barton, "Mccormick-based relaxations of algorithms," *SIAM J. Optim.*, vol. 20, no. 2, pp. 573–601, 2009.

[26] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.