

Quick, Decentralized, Energy-Efficient One-Shot Max Function Computation Using Timer-Based Selection

Arjun Anand and Neelesh B. Mehta, *Senior Member, IEEE*

Abstract—In several wireless sensor networks, it is of interest to determine the maximum of the sensor readings and identify the sensor responsible for it. We propose a novel, decentralized, scalable, energy-efficient, timer-based, one-shot max function computation (TMC) algorithm. In it, the sensor nodes do not transmit their readings in a centrally pre-defined sequence. Instead, the nodes are grouped into clusters, and computation occurs over two contention stages. First, the nodes in each cluster contend with each other using the timer scheme to transmit their reading to their cluster-heads. Thereafter, the cluster-heads use the timer scheme to transmit the highest sensor reading in their cluster to the fusion node. One new challenge is that the use of the timer scheme leads to collisions, which can make the algorithm fail. We optimize the algorithm to minimize the average time required to determine the maximum subject to a constraint on the probability that it fails to find the maximum. TMC significantly lowers average function computation time, average number of transmissions, and average energy consumption compared to approaches proposed in the literature.

Index Terms—Max function computation (TMC), wireless sensor networks (WSN), selection, timer, one-shot.

I. INTRODUCTION

WIRELESS sensor networks (WSN) are increasingly being deployed in industrial, aerospace, environmental monitoring, and smart home applications [1]. Unlike a data network that is evaluated by how much data gets transported between nodes, a WSN is evaluated by the efficacy with which it carries out the specific sensing task that it is deployed for. Constraints on the amount of energy consumed by each node for sensing, computation, and communication, on bandwidth, and on the software and hardware complexities of the nodes make the design of WSNs challenging.

In some applications, it is of interest to determine the maximum of the sensor readings in the network and identify the sensor responsible for it. For example, this helps in early

Manuscript received July 13, 2014; revised November 5, 2014; accepted December 16, 2014. Date of publication December 29, 2014; date of current version March 13, 2015. This work was partially supported by a research grant from the Aerospace Networking Research Consortium (ANRC). The associate editor coordinating the review of this paper and approving it for publication was H. Dai.

A. Anand is with the Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712 USA (e-mail: arjunanand1989@gmail.com).

N. B. Mehta is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India (e-mail: nbmehta@ece.iisc.ernet.in).

Digital Object Identifier 10.1109/TCOMM.2014.2386330

detection of an impending event such as a fire and identifying its source, and helps monitor and identify an egregious source of pollution. The following framework formalizes the max function computation problem in a WSN that consists of n nodes. Each node i has a real-valued local measurement or metric μ_i that is known only to it. The *best node* is defined as the node with the highest metric. A sink or a fusion node needs to identify the best node, $\arg \max\{\mu_1, \dots, \mu_n\}$, and its metric, $\max\{\mu_1, \dots, \mu_n\}$.

Max function computation is a special case of in-network function computation, which has attracted significant interest in the literature. Specifically, it comes under the general class of data aggregation problems that are studied in [2]. It is a special class of type-threshold symmetric functions [3], symmetric functions [4], and fully-multiplexable functions [5].

A. Literature Survey

The literature on in-network function computation can be classified on the basis of the type of network, computational paradigm, and channel access control scheme as follows.

Network Types and Channel Models: Broadcast or co-located networks are considered in [3], [6]–[8] and multi-hop networks in [2]–[5], [7]–[12]. The papers also differ in the channel models assumed. Noiseless links are assumed in [2]–[4], [6], [12], noisy or binary symmetric channels in [4], [7], [8], [10], [11], and capacity-constrained links in [5].

Channel Access Scheme: Oblivious and non-oblivious schemes have been considered. In oblivious schemes, the transmission schedule of the nodes is pre-defined before the computation is performed [2], [4], [7], [12]. Whereas, in non-oblivious schemes, the transmission schedule depends on previous transmissions [3]. A randomized channel access scheme for a structure-free network is instead used in [10].

Paradigms: Three paradigms, namely, one-shot, pipelined, and block computation, have been studied. In one-shot computation, the function is computed once within the coherence interval of measurements [2], [4], [10], [12]. In pipelined computation, the computations of function values at several time instants are pipelined over the network [5], [10], [12]. In block computation, the sensor nodes take measurements for several time instants, buffer them, and then compute the function for each of these time instants in one go [3], [7], [9].

In [3], block computation strategies over a noiseless, random planar multi-hop and co-located networks for symmetric

type-threshold functions are characterized. The nodes' transmissions occur according to a pre-communicated schedule and depend on the contents of the previous transmissions, which are decoded by the nodes within transmission range. A one-shot computation with message passing between nodes for type-threshold functions is studied in [12]. Scaling laws for the optimal computation time, transmit energy consumption, and achievable rate of communication are derived for tree, multi-hop, and ripple algorithms. Pipelining is also treated in the paper as an extension of the one-shot paradigm. In [4], the optimal rate of computation of symmetric functions with binary sensor measurements over noise-free channels and over binary symmetric channels is studied. In [7], a two-stage oblivious scheme for max function computation over a random planer network with binary symmetric channels, which is order-optimal in the number of time slots and number of transmissions required, is studied. In [10], a randomized slotted Aloha protocol over a noisy network with binary symmetric channels is studied; it is order-optimal in the number of time slots required for max function computation.

B. Contributions

We propose a novel timer-based *max* function computation (TMC) algorithm for one-shot computation. It incorporates new ideas that are inspired by opportunistic multiple access (MAC) selection algorithms [13]–[17]. We focus on one-shot computation because it is practically appealing. While block computation reduces the number of transmissions significantly and can exploit correlations between readings of different nodes, it has two limitations that can make it impractical in WSNs constrained by delay or complexity [9]. Firstly, the block size needs to be large. Secondly, the block size grows with n , which causes a large delay because every node has to collect the entire block of measurements.

TMC is based on the distributed timer-based selection scheme, in which each node sets a timer as a function of its metric, and starts counting its timer down. A node transmits a small timer packet when its timer expires. The key idea is that the metric-to-timer mapping is a *monotonic non-increasing* function, which ensures that the node with the maximum metric transmits first. Consequently, the other nodes need not transmit once they have sensed a transmission in the channel. While the timer scheme has been employed for opportunistic selection in [13]–[15], the focus was on selecting the node for data transmission in a one-hop star network. To the best of our knowledge, this is the first time that it has been effectively employed for max function computation.

Another novelty in our approach is the introduction of a reliability parameter η , which requires that the best node and its metric are identified by the sink with a probability that is at least $1 - \eta$. It permits a new trade-off between the time or energy required for computation and its reliability.

1) *Design of TMC and Challenges*: In TMC, the n sensor nodes are grouped into clusters, with each cluster having a cluster-head. The nodes in a cluster can sense transmissions by other nodes within the cluster. The cluster-heads have a longer transmission range and their transmissions can be sensed

by other cluster-heads. Computation takes place in two stages: (i) Intra-cluster stage, in which nodes within each cluster contend with each other using the timer scheme, which enables each cluster-head to determine the maximum in its cluster, and (ii) Inter-cluster stage, in which the cluster-heads contend with each other using the timer scheme, which enables the sink to determine the maximum in the network. Unlike the algorithms in the literature, in which every node transmits at least once, we shall see that far fewer nodes transmit in TMC, which translates into significant time and energy savings.

The use of the timer scheme gives rise to collisions. In the intra-cluster stage, the timer packet transmitted by the best node in a cluster will not be decoded by the cluster-head if the timer of the second best node in the cluster expires within a *vulnerability window* Δ of its transmission [13]. Collisions can occur in the inter-cluster stage as well. Here, Δ depends on the capabilities of the physical layer [13]. It includes the maximum propagation delay, maximum delay spread in the channels seen by the nodes, receive-to-transmit switching time, and time synchronization errors among the nodes.

2) *Optimization*: We determine the optimal parameters of the timer schemes used by the two stages that jointly minimize the average selection time subject to the aforementioned reliability constraint. To this end, we develop a novel bounding approach that decomposes the constrained, optimization problem at hand into two separable and solvable sub-problems pertaining to the intra-cluster and inter-cluster stages. We also study the asymptotic $n \rightarrow \infty$ regime to glean further insights.

3) *Benchmarking and Robustness*: We benchmark the expected selection time and bandwidth savings of our scheme with adaptations of the tree and the ripple algorithms [12], which are amenable to our model. We observe that TMC reduces the average number of transmissions by up to two orders of magnitude, the average energy consumption and the average max function computation time by up to one order of magnitude, except when η is very small. Furthermore, TMC incorporates reuse constraints and is simpler to implement than the oblivious algorithms, which assume a centralized scheduler that determines when each node senses and transmits. Making the nodes aware of the centralized schedule can incur considerable overhead [9].

We also evaluate the performance of TMC when some assumptions behind its design no longer hold, such as when the number of sensor nodes in a cluster is random and unknown. Such a scenario arises, for example, when the sensor nodes autonomously sleep to conserve energy [18].

4) *Comparisons With Other Approaches*: We note that there are several differences between our model and approach and those in the literature. The approaches in [3]–[7], [9], [11], [12], study collision-free or graph-theoretic models in which the transmission by a node is scheduled in such a manner as to not be corrupted by interference from other transmissions, if any. On the other hand, our approach explicitly provisions for collisions between simultaneous transmissions by different nodes. The collision model is widely assumed in the literature on multiple access protocols [18], [19] and selection schemes [13]–[17]. Secondly, [3], [6], [10] assume that a node can decode all transmissions in the network that precede its transmission,

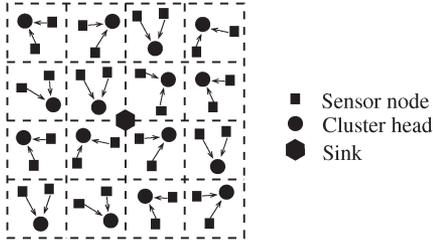


Fig. 1. A two-tier tree model for the system in which the sink is the root, k_1 cluster-heads constitute the first level, and $n = k_1 k_2$ sensor nodes constitute the leaves.

while we only require nodes within a cluster in the intra-cluster stage and the cluster-heads in the inter-cluster stage to be able to sense the presence of a previous transmission. Our algorithm is not oblivious. Thirdly, while several of the aforementioned papers focus on scaling laws for $\eta \rightarrow 0$, we allow for a pre-specified, non-zero selection failure probability and optimize TMC for any given number of nodes. We also strive to identify the sensor node responsible for the max reading.

The paper is organized as follows. Section II sets up the system model. Section III presents and optimizes TMC. Section IV presents the asymptotic scaling of the average selection time. Simulation results are presented in Section V followed by our conclusions in Section VI.

II. SYSTEM MODEL

Consider a system with n sensor nodes and a sink. For ease of exposition, the nodes are distributed in a square region, as shown in Fig. 1. The entire region is divided into k_1 clusters. Each cluster has k_2 nodes located within it. Thus, $n = k_1 k_2$. Every cluster has a cluster-head, which can transmit directly to the sink node. It can be located anywhere within the cluster. (The choice of the cluster-head itself can be optimized using algorithms such as LEACH [20]. We do not delve into this aspect in this paper.) Its transmissions can be sensed by other cluster-heads. On the other hand, the nodes in a cluster need only sense transmissions by other nodes in the same cluster, and not in other clusters. We note that the broadcast network is a special case that corresponds to $k_2 = 1$.

The metrics of the nodes are assumed to be independent and identically distributed (i.i.d.). The independence assumption is justified when the sensor readings decorrelate over distance [15], [21]. The identicalness assumption helps make the analysis tractable, and is widely used in the selection literature [13], [15], [22]. While correlated sensor readings is beyond the scope of this paper, we note that TMC also works with correlation, except that the optimization that follows below does not apply.

Let C denote the cumulative distribution function (CDF) of μ_i . Then, the random variable (RV) $v_i = C(\mu_i)$ can be shown to be uniformly distributed in $[0,1]$. Since the CDF is a monotonically non-decreasing function, the node with the largest μ_i is also the one with the largest v_i . Therefore, without loss of generality, the metric is henceforth assumed to be uniformly distributed over $[0,1]$. Knowing C is practically feasible since it changes at a rate that is several orders of magnitude slower than the metrics. We note that prior knowledge of C is not assumed in [3], [7], [8], [10], [12], while it is assumed in [23].

Timer Scheme: Before we specify TMC, we first describe the timer scheme that will be used by its two stages. The timer scheme uses a discrete metric-to-timer mapping in which the timers expire only at $0, \Delta, \dots, N\Delta$, where N is called the number of *timer levels*. This mapping is used because it maximizes the probability of selecting the best node and also minimizes the average selection time [15]. It is completely characterized by $N + 1$ positive real numbers, called *interval lengths*, $\alpha_N[0], \dots, \alpha_N[N]$ as follows. If a node's metric lies in the interval $(1, 1 - \alpha_N[0])$, then its timer expires immediately at 0. In general, if its metric lies in $(1 - \sum_{l=0}^i \alpha_N[l], 1 - \sum_{l=0}^{i-1} \alpha_N[l])$, then its timer expires at time $i\Delta$. When its timer expires, a node transmits a timer packet of duration T_p . If the metric lies in $[0, 1 - \sum_{l=0}^N \alpha_N[l])$, then its timer does not expire. We shall refer to $[\alpha_N[0], \dots, \alpha_N[N]]$ as an interval length vector.

Intra-cluster Stage: Each cluster uses a timer scheme with N_2 timer levels and interval length vector

$$\beta_{N_2} = [\beta_{N_2}[0], \beta_{N_2}[1], \dots, \beta_{N_2}[N_2]]. \quad (1)$$

A node whose timer expires transmits a packet containing its metric and identity to its cluster-head. The total time required for a cluster is $N_2\Delta + T_p$ because, in the worst case, a node may transmit a timer packet of duration T_p at time $N_2\Delta$.

Reuse Constraints: Transmissions in all the clusters cannot occur simultaneously because of interference constraints. Let r denote the number of clusters in which transmissions can occur simultaneously. Therefore, with k_1 clusters, the intra-cluster stage takes place in $\lceil k_1/r \rceil$ sub-stages, with the intra-cluster stage in r clusters occurring simultaneously. Here, $\lceil \cdot \rceil$ denotes the ceiling function. Since the cluster-heads are not required to communicate with each other, each sub-stage in the intra-cluster stage takes $N_2\Delta + T_p$ time. Hence, the duration $\Gamma_2(\beta_{N_2})$ of the intra-cluster stage is

$$\Gamma_2(\beta_{N_2}) = \lceil k_1/r \rceil (N_2\Delta + T_p). \quad (2)$$

Inter-Cluster Stage: At the start of this stage, each cluster head knows the maximum metric within its cluster, which we shall refer to as its *priority*. Now, the cluster-heads use a timer scheme with N_1 timer levels and interval length vector

$$\alpha_{N_1} = [\alpha_{N_1}[0], \alpha_{N_1}[1], \dots, \alpha_{N_1}[N_1]]. \quad (3)$$

When its timer expires, the cluster-head transmits its priority and the identity of the node it had selected in the intra-cluster stage. Since the cluster-heads can sense each other's transmissions, the inter-cluster stage ends as soon as the first transmission occurs or the maximum time available for it, which is $N_1\Delta + T_p$, runs out.

Special attention must be paid to the following two events in which the TMC cannot select the best node, which we shall refer to as *selection failures*.

Selection Failure in Intra-Cluster Stage: The cluster-head in the cluster with the best node fails to select the best node in the intra-cluster stage. This happens when: (i) no node's timer expires within the duration $N_2\Delta$, or (ii) a timer of at least one other node expires at the same time as the best node, as a result of which a collision occurs. In case the timer scheme does not

yield a success in a cluster, the respective cluster-head sets its metric as the least value possible for the metric, which is zero. This ensures that the chances of the best node being selected in the inter-cell stage are not affected in case it belongs to another cluster.

Selection Failure in Inter-Cluster Stage: The sink node fails to select the best cluster-head in the inter-cluster stage. Akin to the intra-cluster stage, this happens when either none of the cluster-heads transmit within duration $N_1\Delta$ or the transmission by the cluster-head with the highest priority suffers a collision at the sink node.

Note that the reuse constraints do not affect the selection failure probability.

The priority M_j , which drives the inter-cluster stage, is not uniformly distributed over $[0,1]$ because it is the maximum of k_1 metrics in the j^{th} cluster and $\Pr(M_j = 0) > 0$, where $\Pr(\cdot)$ denotes probability. Its CDF F_{M_j} , which is derived in Appendix A, is given in closed-form as:

$$F_{M_j}(x) = p, \quad \text{if } 0 \leq x \leq 1 - \sum_{l=0}^{N_2} \beta_{N_2}[l],$$

and, for $0 \leq i \leq N_2$,

$$F_{M_j}(x) = p + \sum_{l=i+1}^{N_2} s_l \beta_{N_2}[l] + s_i \left(x - 1 + \sum_{l=0}^i \beta_{N_2}[l] \right),$$

$$\text{if } 1 - \sum_{l=0}^i \beta_{N_2}[l] < x \leq 1 - \sum_{l=0}^{i-1} \beta_{N_2}[l], \quad (4)$$

where $p = 1 - k_2 \sum_{l=0}^{N_2} \beta_{N_2}[l] \left(1 - \sum_{q=0}^l \beta_{N_2}[q] \right)^{k_2-1}$ and $s_i = k_2 \left(1 - \sum_{l=0}^i \beta_{N_2}[l] \right)^{k_2-1}$.

The j^{th} cluster-head then applies the following transformation to its priority M_j to get an RV Y_j , which is uniformly distributed in $[0,1]$:

$$Y_j = \begin{cases} F_{M_j}(M_j), & M_j \neq 0, \\ U_j, & M_j = 0, \end{cases} \quad (5)$$

where U_j is a uniformly distributed RV in the interval $[0,p]$. It uses Y_j as its metric in the inter-cluster stage. It is easy to see that Y_1, \dots, Y_{k_1} are i.i.d.

Comments: The operation of TMC requires only loose network-wide synchronization. Firstly, the design of the timer scheme is based on Δ . As mentioned, various time synchronization errors can all be incorporated into Δ . Secondly, the timer schemes in the two stages can be launched by broadcasting a beacon from the cluster-head (in the intra-cluster stage) or sink (in the inter-cluster stage) to the respective contending nodes to indicate the start time. Thirdly, since the $\lceil k_1/r \rceil$ intra-cluster sub-stages run in a time orthogonal manner, a limited number of guard intervals can cope with any synchronization error between the sub-stages.

III. OPTIMAL DESIGN OF TMC

Our objective is to determine the optimal $\alpha_{N_1} \in (\mathbb{R}^+)^{N_1+1}$, $\beta_{N_2} \in (\mathbb{R}^+)^{N_2+1}$, and $N_1, N_2 \in \mathbb{Z}^+$ that minimize the expected

function computation time $\Gamma(\alpha_{N_1}, \beta_{N_2})$ over the network subject to (s.t.) the probability of selection failure $F(\alpha_{N_1}, \beta_{N_2})$ not exceeding η . The optimization problem can be stated as follows:

$$OP_1 : \min_{\alpha_{N_1}, \beta_{N_2}, N_1, N_2} \Gamma(\alpha_{N_1}, \beta_{N_2}), \quad (6)$$

$$\text{s.t. } F(\alpha_{N_1}, \beta_{N_2}) \leq \eta, \quad (7)$$

$$0 \leq \alpha_{N_1}[i] \leq 1, \quad i = 0, \dots, N_1, \quad (8)$$

$$0 \leq \beta_{N_2}[j] \leq 1, \quad j = 0, \dots, N_2, \quad (9)$$

$$\sum_{i=0}^{N_1} \alpha_{N_1}[i] \leq 1 \text{ and } \sum_{j=0}^{N_2} \beta_{N_2}[j] \leq 1, \quad (10)$$

$$N_1, N_2 \in \mathbb{Z}^+. \quad (11)$$

The constraints (8), (9), and (10) ensure that the timer interval lengths are positive and together lie in $[0,1]$.

OP_1 is an intractable combinatorial, stochastic optimization problem. However, we shall see that imposing a tighter reliability constraint that is based on the union bound, begets a tractable and insightful solution. For this, let $F_1(\alpha_{N_1})$ denote the probability that inter-cluster stage fails in selecting the cluster-head with the highest priority. And, let $F_2(\beta_{N_2})$ denote the probability that in the intra-cluster stage, the cluster containing the best node fails to select it. Recall that it does not matter if the intra-cluster timer scheme fails to select a node in a cluster that does not contain the best node.

Applying the union bound, we get

$$F(\alpha_{N_1}, \beta_{N_2}) \leq F_1(\alpha_{N_1}) + F_2(\beta_{N_2}). \quad (12)$$

Replacing (7) with (12) yields the following tighter constrained optimization problem:

$$OP_2 : \min_{\alpha_{N_1}, \beta_{N_2}, N_1, N_2} \Gamma(\alpha_{N_1}, \beta_{N_2}), \quad (13)$$

$$\text{s.t. } F_1(\alpha_{N_1}) + F_2(\beta_{N_2}) \leq \eta, \quad (14)$$

along with the constraints in (8), (9), (10), and (11). Any solution that satisfies (14) satisfies (7) as well.

We note that failure events due to noise or fading are not modeled. These can be incorporated into our framework by including the probability of their occurrence in the two stages in the upper bound on selection failure probability in (12).

To solve OP_2 , we first derive $F_1(\alpha_{N_1})$ and $F_2(\beta_{N_2})$.

Result 1: The intra-cluster selection failure probability $F_2(\beta_{N_2})$ is

$$F_2(\beta_{N_2}) = 1 - \frac{k_1 k_2}{k_1 k_2 - k_2 + 1} \sum_{l=0}^{N_2} \left(1 - \sum_{i=0}^l \beta_{N_2}[i] \right)^{k_2-1}$$

$$\times \left(\left[1 - \sum_{i=0}^{l-1} \beta_{N_2}[i] \right]^{k_1 k_2 - k_2 + 1} \right.$$

$$\left. - \left[1 - \sum_{i=0}^l \beta_{N_2}[i] \right]^{k_1 k_2 - k_2 + 1} \right). \quad (15)$$

The selection failure probability $F_1(\alpha_{N_1})$ in the inter-cluster stage is

$$F_1(\alpha_{N_1}) = 1 - k_1 \sum_{l=0}^{N_1} \alpha_{N_1}[l] \left(1 - \sum_{j=0}^l \alpha_{N_1}[j] \right)^{k_1-1}. \quad (16)$$

Proof: The proof is given in Appendix B. ■

Next, we evaluate the expected selection time $\Gamma(\alpha_{N_1}, \beta_{N_2})$. It is the sum of the expected duration $\Gamma_2(\beta_{N_2}) = \lceil k_1/r \rceil (N_2\Delta + T_p)$ of the intra-cluster stage (cf. (2)) and the expected duration $\Gamma_1(\alpha_{N_1})$ of the inter-cluster stage, which is derived below. Let the timer of the cluster-head with the highest priority expire at time $i\Delta$, in which case the duration of the inter-cluster stage is $T_p + i\Delta$. Therefore, we have

$$\Gamma_1(\alpha_{N_1}) = T_p + \Delta \mathbb{E}[i] = T_p + \Delta \sum_{l=0}^{N_1-1} \Pr(i > l), \quad (17)$$

where $\mathbb{E}[\cdot]$ denotes expectation. Since the metrics are uniform and i.i.d., we have $\Pr(i > l) = \left(1 - \sum_{j=0}^l \alpha_{N_1}[j] \right)^{k_1}$. Substituting this in (17) yields

$$\Gamma_1(\alpha_{N_1}) = T_p + \Delta \sum_{l=0}^{N_1-1} \left(1 - \sum_{j=0}^l \alpha_{N_1}[j] \right)^{k_1}. \quad (18)$$

The following key result that helps solve OP_2 then follows.

Result 2: Solving OP_2 is equivalent to separately solving the following two sub-problems:

$$SP_1: \min_{\alpha_{N_1}, N_1} \Gamma_1(\alpha_{N_1}) + \lambda F_1(\alpha_{N_1}), \quad (19)$$

$$\text{s.t. } 0 \leq \alpha_{N_1}[i] \leq 1, \quad i = 0, 1, \dots, N_1, \quad (20)$$

$$\sum_{i=0}^{N_1} \alpha_{N_1}[i] \leq 1, \quad N_1 \in \mathbb{Z}^+, \quad (21)$$

and

$$SP_2: \min_{\beta_{N_2}, N_2} \frac{k_1}{r} (N_2\Delta + T_p) + \lambda F_2(\beta_{N_2}), \quad (22)$$

$$\text{s.t. } 0 \leq \beta_{N_2}[i] \leq 1, \quad i = 0, 1, \dots, N_2, \quad (23)$$

$$\sum_{i=0}^{N_2} \beta_{N_2}[i] \leq 1, \quad N_2 \in \mathbb{Z}^+. \quad (24)$$

There exists $\lambda \geq 0$ such that the optimum solutions to SP_1 and SP_2 , which are $\alpha_{N_1}^*$ and $\beta_{N_2}^*$, respectively, always meet (14) with equality.

Proof: The proof is given in Appendix C. ■

Notice that SP_1 and SP_2 deal with the inter-cluster and intra-cluster stages, respectively. They are coupled through the constant λ , which is determined numerically but only once.

A. Solving SP_1

The optimal interval lengths $\alpha_{N_1}^*[j]$, $j = 0, \dots, N_1$, for a given N_1 are as follows.

Lemma 1: Given N_1 , the optimal interval lengths $\alpha_{N_1}^*[j]$, $j = 0, \dots, N_1$, are given by the following recursion:

$$\alpha_{N_1}^*[j] = \begin{cases} \frac{\Delta + \lambda + Q_{N_1-1}^*(\lambda)}{\Delta + \lambda k_1 + Q_{N_1-1}^*(\lambda)}, & j = 0, \\ \left(1 - \alpha_{N_1}^*[0] \right) \alpha_{N_1-1}^*[j-1], & 1 \leq j \leq N_1, \end{cases} \quad (25)$$

where $\alpha_0^*[0] = 1/k_1$ and

$$Q_{N_1}^*(\lambda) = \Delta \sum_{l=0}^{N_1-1} \left(1 - \sum_{q=0}^l \alpha_{N_1}^*[q] \right)^{k_1} - \lambda k_1 \sum_{l=0}^{N_1} \alpha_{N_1}^*[l] \left(1 - \sum_{q=0}^l \alpha_{N_1}^*[q] \right)^{k_1-1}. \quad (26)$$

Proof: As shown in [15], SP_1 is equivalent to optimizing the timer interval lengths such that the expected time to select the best node is minimized subject to a constraint on the probability of selecting the best node. The solution follows from [15, Theorem 3]. ■

Thus, only N_1 remains to be optimized in SP_1 . It is easy to see that $\Gamma_1(\alpha_{N_1})$ in (18) is a monotone non-increasing function of N_1 because increasing N_1 gives more variables for solving SP_1 . Therefore, given η , N_1 should be made as large as possible. When $N_1 \rightarrow \infty$, the expressions in (25) and (26) simplify considerably, as we show below.

1) Asymptotic Simplifications: Let $\lim_{N_1 \rightarrow \infty} \alpha_{N_1}^*[j] = \alpha_\infty^*[j]$.

Applying the limit $N_1 \rightarrow \infty$ to (25), we get $\alpha_\infty^*[j] = (1 - \alpha_\infty^*[0]) \alpha_\infty^*[j-1]$. Substituting these in (16) and (18), we get

$$\Gamma_1(\alpha_\infty^*) = T_p + \frac{\Delta (1 - \alpha_\infty^*[0])^{k_1}}{1 - (1 - \alpha_\infty^*[0])^{k_1}}, \quad (27)$$

$$F_1(\alpha_\infty^*) = 1 - \frac{k_1 \alpha_\infty^*[0] (1 - \alpha_\infty^*[0])^{k_1-1}}{1 - (1 - \alpha_\infty^*[0])^{k_1}}. \quad (28)$$

Note that (27) and (28) are a function of only $\alpha_\infty^*[0]$. Therefore, when $N_1 \rightarrow \infty$, the inter-cluster stage is completely characterized by a single parameter $\alpha_\infty^*[0]$. As derived in Appendix D, $\alpha_\infty^*[0]$ is the unique fixed point of the following function $f_\lambda(x)$ in the interval $[0, 1]$:

$$f_\lambda(x) = 1 - \frac{(1-x)^{k_1} + k_1 - 1}{(\Delta/\lambda) + k_1}. \quad (29)$$

It is efficiently computed using bisection search.

B. Solving SP_2

We now determine the optimal interval length vector β_{N_2} that solves SP_2 .

Result 3: Given any N_2 , the optimal interval lengths $\beta_{N_2}^*[j]$, $j = 0, \dots, N_2$, are recursively given by

$$\beta_{N_2}^*[j] = \begin{cases} 1 - \left(\frac{\kappa(k_2-1)}{k_1 k_2 [\kappa-1 + F_2(\beta_{N_2-1}^*)]} \right)^{\kappa/(k_1 k_2)}, & j = 0, \\ \left(1 - \beta_{N_2}^*[0] \right) \beta_{N_2-1}^*[j-1], & 1 \leq j \leq N_2, \end{cases} \quad (30)$$

where $\kappa = \frac{k_1 k_2}{k_1 k_2 - k_2 + 1}$ and $\beta_0^*[0] = 1 - \left(\frac{k_2-1}{k_1 k_2} \right)^{\kappa/(k_1 k_2)}$.

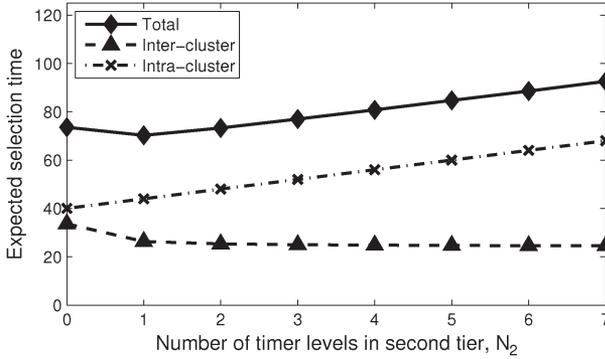


Fig. 2. Expected total selection time $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$, inter-cluster time $\Gamma_1(\alpha_{N_1}^*)$, and intra-cluster selection time $\Gamma_2(\beta_{N_2}^*)$ as a function of N_2 ($\eta = 0.05$, $k_1 = 200$, $k_2 = 10$, $r = \lceil k_1/4 \rceil = 50$, and $T_p = 10\Delta$).

Proof: The proof is given in Appendix E. ■

A key difference between \mathcal{SP}_1 and \mathcal{SP}_2 is that, unlike N_1 , a larger value of N_2 need not reduce $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$. This is shown in Fig. 2, which plots $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$, $\Gamma_1(\alpha_{N_1}^*)$, and $\Gamma_2(\beta_{N_2}^*)$ as a function of N_2 for $\eta = 0.05$. We see that the optimal value for N_2 is one in this example. The reason for this is as follows. As N_2 increases, $F_2(\beta_{N_2}^*)$ decreases and $\Gamma_2(\beta_{N_2}^*)$ increases linearly (cf. (2)). Given η , decreasing $F_2(\beta_{N_2}^*)$ increases $F_1(\alpha_{N_1}^*)$ since they must sum to η . However, increasing the selection failure probability $F_1(\alpha_{N_1}^*)$ in the inter-cluster stage reduces the time for selection $\Gamma_1(\alpha_{N_1}^*)$. Thus, a trade-off exists between increasing $\Gamma_2(\beta_{N_2}^*)$ and decreasing $\Gamma_1(\alpha_{N_1}^*)$. The optimum value of N_2 is determined using a simple one-dimensional bisection search. In general, as η decreases, the optimal value of N_2 will increase.

Comments: The recursions in Section III-A1 and Result 3 can be implemented locally at the cluster-heads and sensor nodes, respectively. In principle, k_1 , k_2 , and η need to be broadcast to the cluster-heads and nodes to enable this. In addition, N_1 and N_2 may also be broadcast to simplify computation. As we shall observe later in Section V-D, TMC is also robust to uncertainty in the number of nodes per cluster. Thus, it can handle a dynamic clustering environment.

IV. ASYMPTOTIC INSIGHTS

To gain more insights about the optimal design of TMC, we study the asymptotic regime $n \rightarrow \infty$. We shall show that $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ tends to a positive constant, which depends on the reliability parameter η . Thus, TMC is scalable.

The value of $n = k_1 k_2$ is increased by increasing the number of clusters k_1 , while keeping the number of nodes per cluster k_2 and the area covered by the network constant. This increases the number of nodes per unit area. First, we derive the asymptotic scaling of the expected selection time in the intra-cluster stage.

Lemma 2: For $N_2 = 0$, $F_2(\beta_{N_2}^*) \rightarrow 0$ as $k_1 \rightarrow \infty$ and k_2 is kept constant.

Proof: The proof is given in Appendix F. ■

The above result shows that the best node is selected with probability one in the intra-cluster stage for $N_2 = 0$ in the asymptotic regime. For $N_2 = 0$, the selection time in the intra-cluster stage is the lowest and equals T_p . Therefore, $N_2 = 0$ is the optimal solution for the intra-cluster stage in the asymptotic

regime. Since r is proportional to k_1 , the intra-cluster stage takes a constant time of $\lceil k_1/r \rceil T_p$ slots. This proportionality between r and k_1 is justifiable when a constant number of adjacent cells are disabled due to interference from a given cell [7].

For the inter-cluster stage, we saw in Section III-A that the optimum solution is obtained when $N_1 \rightarrow \infty$. Applying the limit $k_1 \rightarrow \infty$ to (27), (28), and (29), yields the following.

Result 4: When $k_1 \rightarrow \infty$, we have

$$\Gamma_1(\alpha_{N_1}^*) = T_p + \frac{\Delta \exp(-d_\lambda)}{1 - \exp(-d_\lambda)}, \quad (31)$$

where d_λ is a positive constant that depends on λ . It is the unique solution of $x = 1 + (\Delta/\lambda) - \exp(-x)$.

Proof: The proof is given in Appendix G. ■

Thus, $\Gamma_1(\alpha_{N_1}^*)$ depends only d_λ . This depends on λ , which, in turn, depends on η (cf. Result 2). Hence, $\Gamma_1(\alpha_{N_1}^*)$ tends to a constant and so does the total selection time.

V. NUMERICAL RESULTS AND PERFORMANCE EVALUATION

We now compare the average number of transmissions, average energy consumed, and average selection time of TMC with other algorithms proposed in the literature that can be adapted and simulated for one-shot max function computation. For the simulations, a square field of unit area is considered, as shown in the Fig. 1. The reuse constraints require that four neighboring clusters cannot transmit simultaneously. Hence, $r = \lceil k_1/4 \rceil$. Further, $T_p = 10\Delta$.

A. Energy Model

To compare energy consumption, we use the following propagation and transmission model. If l and b are the length and breadth, respectively, of a cluster area, then the transmit power P_{tx} for the intra-cluster stage is set such that any two nodes in the cluster – including the worst case in which nodes that are diagonally opposite to each other – can hear each other. It is calculated as follows. The receive signal-to-noise-ratio (SNR) at the worst case distance of $\sqrt{l^2 + b^2}$ is $\frac{P_{tx}}{\sigma^2} (\frac{d_0}{d})^\zeta$, where σ^2 is the noise power, d_0 is a reference distance, and ζ is the pathloss exponent. In order that the SNR equals the decoding threshold of γ , the transmit energy $E^{\text{intra}} = P_{tx} T_p$ is equal to $E^{\text{intra}} = \gamma \sigma^2 T_p (\sqrt{l^2 + b^2} / d_0)^\zeta$.

Similarly, in the intra-cluster stage, all cluster-heads must hear each other. Therefore, the transmission energy E^{inter} for a cluster-head that transmits directly to the sink is $E^{\text{inter}} = \gamma \sigma^2 T_p (\sqrt{L^2 + B^2} / d_0)^\zeta$, where L and B are the length and breadth, respectively, of the measurement field. We focus on transmit energy consumption, as has been done in [7], [8].

B. Benchmark Algorithms

We benchmark TMC with the following algorithms proposed in [12], as they can be adapted for one-shot computation.

1) *Tree Algorithm With Direct Transmission:* It uses two stages. In the intra-cluster stage, the k_2 nodes in a cluster transmit to their cluster-head in a round-robin manner. This requires a total time of $4k_2 T_p$ after accounting for reuse constraints.

The energy consumed is E^{intra} per transmission. Next, in the inter-cluster stage, the k_1 cluster-heads transmit their priorities directly to the sink in a round-robin manner. This requires an additional time of $k_1 T_p$. The energy consumed is E^{inter} per transmission.

2) *Tree Algorithm With Multi-Hop Transmission*: The cluster-heads route their priorities column-wise and then row-wise to the sink. First, the cluster-heads of the left-most and right-most columns transmit to the cluster-heads in the same row of the neighboring columns. Each cluster-head transmits the maximum of its priority and the priority that it has received thus far from its neighbors to its inner neighbor, and so on. Due to the interference constraints, a column is divided into two sets of alternate clusters, and only one set transmits at a time for a duration of T_p . Hence, $2T_p$ time is required by all the cluster-heads in one column to transmit to the corresponding ones in the neighboring column. This process repeats until the priorities are routed to the central column. Thereafter, they are routed row-wise to the central sink node. Since the cluster-heads have to transmit to their neighboring cluster-heads, which could be located anywhere in the neighboring cluster, the energy required turns out to be $2E^{\text{intra}}$.

3) *Ripple Algorithm*: The computation is divided into rounds. In a round, each node broadcasts a packet with its metric and identity exactly once. Due to reuse constraints, only the nodes in $\lceil k_1/4 \rceil$ non-adjacent clusters can transmit at a time. In each of these non-interfering clusters, the nodes take turns to broadcast their metrics. Hence, each round takes $4k_2 T_p$ time. As before, each transmission consumes E^{intra} energy. Upon decoding a packet, a node updates its metric as the maximum of its metric and the metric in the received packet. The metric of the best node, thus, propagates by one hop in each round. The algorithm stops when the metric from the farthest node reaches the sink. The number of rounds is the maximum hop distance from the sink to the edge of the square field.

We note that it is difficult to directly compare with the algorithms studied in [3], [5], [7], [9] because they use an altogether different block computation model. It is also difficult to compare with the algorithms studied in [2], [4], since their focus is on proving order-optimal bounds on the rate at which the function computation or energy consumption occur. The message propagation for the tree algorithms above is similar to that considered in [2, Alg. 2], [4, Thm. III.4] once the tree is specified.

C. Numerical Results

Fig. 3 plots the expected selection time as a function of the target selection failure probability η for $k_1 = 10$ and $k_2 = 20$. The selection times of the tree and ripple algorithms are independent of η because all nodes transmit at least once as per a pre-defined schedule. However, the expected selection time for TMC decreases as η increases. A larger value of η means a weaker constraint, which enables TMC to allocate less time to select the best node. For $\eta = 0.04$, TMC is 8.1x, 8.5x, and 37.9x faster than the tree algorithm with multi-hop transmission, tree algorithm with direct transmission, and ripple algorithm, respectively. Note that these significant reductions are not a straight-forward outcome of the fact that TMC is designed for

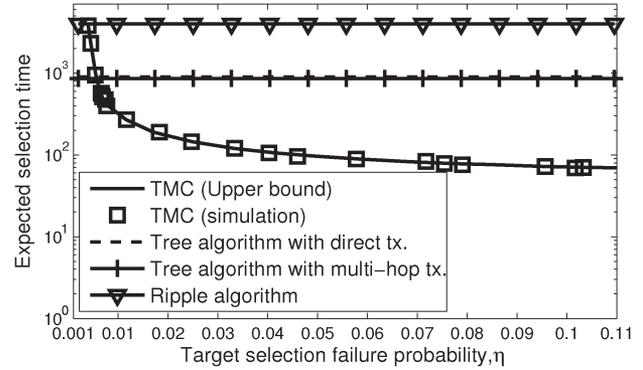


Fig. 3. Expected selection time $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ as a function of the target selection failure probability η ($k_1 = 10$ and $k_2 = 20$).

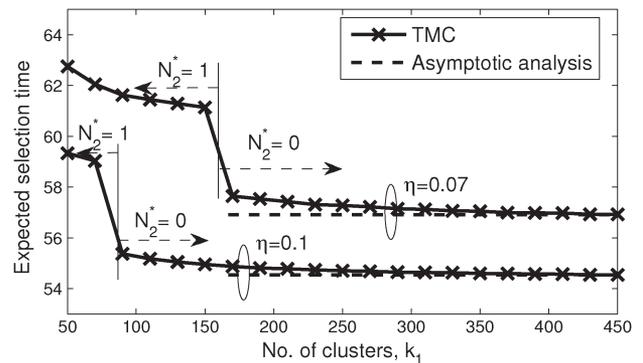


Fig. 4. Zoomed-in view of asymptotic behavior of $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ as a function of k_1 ($k_2 = 20$).

a non-zero selection failure probability while the benchmark algorithms are not. Instead, they arise because the timer scheme incentivizes nodes with higher metrics to transmit earlier, which prevents unnecessary transmissions from nodes with smaller metrics. We see that the upper bound in (12) that was used in the formulation of OP_2 is quite close to the simulation results. For $\eta < 0.005$, TMC no longer outperforms the benchmark algorithms. As n increases, this cross-over point decreases.

Asymptotics and Scalability: Fig. 4 plots $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ as a function of k_1 for $k_2 = 20$ for two values of η . We see that as k_1 increases, $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ decreases and approaches the asymptotic value, which is derived in Section IV. The trends for $\eta = 0.1$ are similar to those for $\eta = 0.07$, except that the selection time is lesser. As k_1 increases, the sudden dips in the expected selection time correspond to a change in the optimal value of N_2 . Further, if n is kept fixed and the number of clusters k_1 is increased, $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ decreases; the corresponding figure is not shown due to space constraints. Thus, if every node is capable of serving as a cluster-head and communicating directly with the sink, then the optimal system configuration is to have all the nodes act as cluster-heads. However, such a configuration may not be achievable in practice because the nodes might not be capable of transmitting directly to the sink. The two-tier configuration also enables the use of sensor nodes and cluster-heads with different capabilities, as is the case in Zigbee [24].

Fig. 5 plots the average number of transmissions, which is a measure of the time-frequency resources required by the network to compute the maximum, as a function of η . Both

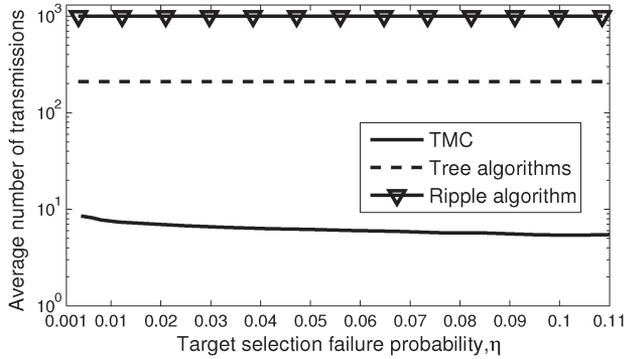


Fig. 5. Average number of transmissions as a function of the target selection failure probability η ($k_1 = 10$ and $k_2 = 20$).

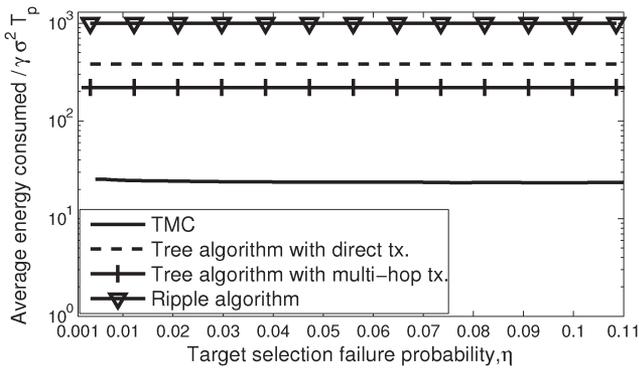


Fig. 6. Average energy consumed as a function of η ($k_1 = 10$, $k_2 = 20$, $E_0 = 1$, $l = 0.5$, $b = 0.2$, $L = 1$, and $B = 1$).

variants of the tree algorithm require the same number of transmissions since all nodes transmit exactly once in a round of computation. The number of transmissions in TMC is up to two orders of magnitude lower than the benchmark algorithms because, in TMC, at most one transmission occurs per cluster in the intra-cluster stage and one transmission typically occurs in the inter-cluster stage. For example, for $\eta = 0.04$, TMC requires 33.3x and 158.6x fewer transmissions than the tree and ripple algorithms, respectively.

Fig. 6 plots the average energy consumed as a function of η for $\zeta = 3$ and $d_0 = \sqrt{l^2 + b^2}$. The cluster size is such that the energy at the cluster edge is E_0 , which is the minimum energy required to decode a packet. TMC consumes 9.2x, 16.1x, and 42.1x less average energy than the tree algorithm with multi-hop transmission, the tree algorithm with direct transmission, and the ripple algorithm, respectively, for $k_1 = 10$ and $k_2 = 20$. The reason behind the significant reduction in energy consumed is the same as that in Fig. 5.

D. Robustness Evaluation

We now study the performance of TMC when the number of nodes in each cluster is an RV that is not known *a priori*. TMC is now designed assuming that the nodes per cluster is equal to its mean \bar{k}_2 . Note that due to its distributed, non-oblivious nature, it can easily adapt to such variations.

We present simulation results for two distributions on k_2 : (i) Binomial: the probability that the number of nodes in a clus-

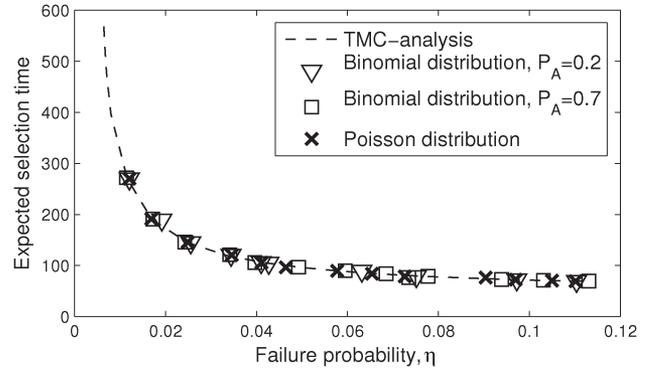


Fig. 7. Expected selection time $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ as a function of η when k_2 is a binomial or Poisson distributed RV ($k_1 = 10$ and $\bar{k}_2 = 20$).

ter is k_2 is $\binom{k_2^{\max}}{k_2} P_A^{k_2} (1 - P_A)^{k_2^{\max} - k_2}$, where P_A is the probability that a sensor node takes part in the algorithm and k_2^{\max} is the maximum number of nodes possible in a cluster. We shall refer to P_A as *active probability*. Thus, $\bar{k}_2 = k_2^{\max} P_A$. (ii) Poisson: the probability that the number of nodes in a cluster is k_2 is $\frac{\bar{k}_2^{k_2}}{k_2!} \exp(-\bar{k}_2)$. The number of clusters is k_1 and is assumed to be known *a priori* since the cluster-heads can easily register with the sink. Fig. 7 plots $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ as a function of η when k_2 is drawn from the above distributions. In both cases, we see that $\Gamma(\alpha_{N_1}^*, \beta_{N_2}^*)$ is within 2% of the case when TMC is designed for \bar{k}_2 .

VI. CONCLUSION

We proposed a new, two stage algorithm called TMC for one-shot max function computation in a WSN. Its use of the timer-based selection scheme enabled nodes with higher metrics to transmit earlier and prevented unnecessary transmissions by nodes with smaller metrics. We jointly optimized the intra-cluster and inter-cluster stages to minimize the expected selection time subject to a cap on the probability of selection failure. We saw that TMC was scalable and robust. It delivered significant reductions in the average selection time, number of transmissions, and energy consumed compared to approaches pursued in the literature except for very small values of η .

We note that TMC can be generalized to a multi-hop network as follows. First, the nodes form a tree. In each layer/hop of the tree network, the timer scheme is run. The leaf nodes with a common parent first contend using the timer scheme. Thereafter, these parents with common grandparents then contend using the timer scheme, and so on. The optimization of the scheme is also amenable to such a model as the upper bound in (12) would now sum over the selection failure probabilities of the different levels of the tree.

APPENDIX

A. CDF of M_j in (4)

We treat the cases $M_j = 0$ and $M_j > 0$ separately.

$M_j = 0$: This happens in case running the timer scheme in the j^{th} cluster does not result in a success. Therefore, $F_{M_j}(0)$ is equal to one minus the probability of success in

the j^{th} cluster. Success occurs if the metric of one node in the cluster lies in $\left(1 - \sum_{q=0}^l \beta_{N_2}[q], 1 - \sum_{q=0}^{l-1} \beta_{N_2}[q]\right)$ and the metrics of all the other $k_2 - 1$ nodes are less than $1 - \sum_{q=0}^l \beta_{N_2}[q]$, for any $l = 0, \dots, N_2$. This happens with probability $k_2 \beta_{N_2}[l] \left(1 - \sum_{q=0}^l \beta_{N_2}[q]\right)^{k_2-1}$. Summing up the probabilities over $l = 0, 1, \dots, N_2$, we get

$$F_{M_j}(0) = 1 - k_2 \sum_{l=0}^{N_2} \beta_{N_2}[l] \left(1 - \sum_{q=0}^l \beta_{N_2}[q]\right)^{k_2-1}. \quad (32)$$

$M_j > 0$: In the intra-cluster stage, when the best metric in the j^{th} cluster is less than $1 - \sum_{l=0}^{N_2} \beta_{N_2}[l]$, no transmission occurs, which leads to the cluster-head setting M_j as 0. Therefore, $M_j \notin \left(0, 1 - \sum_{l=0}^{N_2} \beta_{N_2}[l]\right)$. Hence, for $0 \leq x \leq 1 - \sum_{l=0}^{N_2} \beta_{N_2}[l]$, we have $F_{M_j}(x) = F_{M_j}(0)$.

Now, consider the case in which $1 - \sum_{l=0}^i \beta_{N_2}[l] < x \leq 1 - \sum_{l=0}^{i-1} \beta_{N_2}[l]$. In this case,

$$\begin{aligned} F_{M_j}(x) &= \Pr\left(1 - \sum_{l=0}^i \beta_{N_2}[l] < M_j \leq x\right) + F_{M_j}(0) \\ &+ \Pr\left(0 < M_j \leq 1 - \sum_{l=0}^i \beta_{N_2}[l]\right). \end{aligned} \quad (33)$$

For M_j to lie in $\left(1 - \sum_{l=0}^i \beta_{N_2}[l], x\right)$, one node must lie in this interval and the metrics of the remaining $k_2 - 1$ nodes must lie in $\left[0, 1 - \sum_{l=0}^i \beta_{N_2}[l]\right)$. Since the metrics are uniformly distributed in $[0, 1]$, we get

$$\begin{aligned} \Pr\left(1 - \sum_{l=0}^i \beta_{N_2}[l] < M_j \leq x\right) &= \left(1 - \sum_{l=0}^i \beta_{N_2}[l]\right)^{k_2-1} \\ &\times k_2 \left(x - 1 + \sum_{l=0}^i \beta_{N_2}[l]\right). \end{aligned} \quad (34)$$

Similarly, we can show that

$$\begin{aligned} \Pr\left(0 < M_j \leq 1 - \sum_{l=0}^i \beta_{N_2}[l]\right) &= \sum_{l=i+1}^{N_2} k_2 \left(1 - \sum_{j=0}^l \beta_{N_2}[j]\right)^{k_2-1} \beta_{N_2}[l]. \end{aligned} \quad (35)$$

Substituting (32), (34), and (35) in (33) yields (4).

B. Proof of Result 1

1) *Expression for $F_2(\boldsymbol{\beta}_{N_2})$* : Let the nodes be labeled such that the first k_2 nodes form the first cluster, the next set of k_2 nodes form the second cluster, and so on. Without loss of generality, let node 1 be the best node. Let S_1 denote the event that node 1 is selected by its cluster in the intra-cluster stage. Therefore,

$$F_2(\boldsymbol{\beta}_{N_2}) = 1 - \Pr(S_1 | \mu_1 = \max\{\mu_1, \dots, \mu_{k_1 k_2}\}). \quad (36)$$

As the metrics are i.i.d., $\Pr(\mu_1 = \max\{\mu_1, \dots, \mu_{k_1 k_2}\}) = 1/(k_1 k_2)$. Therefore, from Bayes' rule, we get

$$F_2(\boldsymbol{\beta}_{N_2}) = 1 - k_1 k_2 \Pr(S_1, \mu_1 \geq \mu_2, \dots, \mu_1 \geq \mu_{k_1 k_2}). \quad (37)$$

To evaluate (37), we condition with respect to μ_1 . Selection of node 1 in the intra-cluster stage depends only on the metrics of the $k_2 - 1$ other nodes in its cluster, and is independent of the metrics of the remaining $k_1 k_2 - k_2$ nodes. Since the metrics are i.i.d. and are uniformly distributed in $[0, 1]$, we get

$$\begin{aligned} F_2(\boldsymbol{\beta}_{N_2}) &= 1 - k_1 k_2 \mathbb{E}_{\mu_1} \left[\mu_1^{k_1 k_2 - k_2} \right. \\ &\quad \left. \times \Pr(S_1, \mu_1 \geq \mu_2, \dots, \mu_1 \geq \mu_{k_2} | \mu_1) \right]. \end{aligned} \quad (38)$$

Next, we evaluate $\Pr(S_1, \mu_1 \geq \mu_2, \dots, \mu_1 \geq \mu_{k_2} | \mu_1)$. If $\mu_1 \in \left(1 - \sum_{i=0}^l \beta_{N_2}[i], 1 - \sum_{i=0}^{l-1} \beta_{N_2}[i]\right)$, then node 1 is selected only if $\mu_q \leq 1 - \sum_{j=0}^l \beta_{N_2}[j]$, for $q = 2, \dots, k_2$. Otherwise, there would be a collision. Therefore, if $\mu_1 \in \left(1 - \sum_{i=0}^l \beta_{N_2}[i], 1 - \sum_{i=0}^{l-1} \beta_{N_2}[i]\right)$, we have

$$\Pr(S_1, \mu_1 \geq \mu_2, \dots, \mu_1 \geq \mu_{k_2} | \mu_1) = \left(1 - \sum_{i=0}^l \beta_{N_2}[i]\right)^{k_2-1}. \quad (39)$$

Substituting (39) in (38) for the different timer intervals that μ_1 can lie in, and computing the expectation using the fact that μ_1 is uniformly distributed in $[0, 1]$, we get (15).

2) *Evaluation of $F_1(\boldsymbol{\alpha}_{N_1})$* : If the priority of the cluster-head whose cluster contains the best node lies in $\left(1 - \sum_{l=0}^i \alpha_{N_1}[l], 1 - \sum_{l=0}^{i-1} \alpha_{N_1}[l]\right)$, then the priorities of all the other cluster-heads must be less than $1 - \sum_{l=0}^i \alpha_{N_1}[l]$ to avoid a collision. This happens with probability $k_1 \alpha_{N_1}[i] \left(1 - \sum_{l=0}^i \alpha_{N_1}[l]\right)^{k_1-1}$. Summing over $i = 0, \dots, N_1$ yields (16).

C. Proof of Result 2

In the proof that follows, given N_1 and N_2 , we shall say that a pair of interval length vectors for the intra-cluster and inter-cluster stages is a *feasible* solution of OP_2 if it satisfies the constraints in (14), (8), (9), and (10).

For a constant $\lambda \geq 0$, define

$$L^\lambda(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2}) = \Gamma(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2}) + \lambda (F_1(\boldsymbol{\alpha}_{N_1}) + F_2(\boldsymbol{\beta}_{N_2})). \quad (40)$$

Let $\boldsymbol{\alpha}_{N_1}^*$ and $\boldsymbol{\beta}_{N_2}^*$ minimize (40) for a given λ and let L_λ^* denote the corresponding minimum value. Suppose there exists a λ such that $F_1(\boldsymbol{\alpha}_{N_1}^*) + F_2(\boldsymbol{\beta}_{N_2}^*) = \eta$. Clearly, $\boldsymbol{\alpha}_{N_1}^*$ and $\boldsymbol{\beta}_{N_2}^*$ are feasible. Furthermore, by definition, $L_\lambda^* \leq L^\lambda(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2})$ for any $\boldsymbol{\alpha}_{N_1}$ and $\boldsymbol{\beta}_{N_2}$. Therefore,

$$\Gamma(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2}) - \Gamma(\boldsymbol{\alpha}_{N_1}^*, \boldsymbol{\beta}_{N_2}^*) \geq \lambda (\eta - F_1(\boldsymbol{\alpha}_{N_1}) - F_2(\boldsymbol{\beta}_{N_2})). \quad (41)$$

If $\boldsymbol{\alpha}_{N_1}$ and $\boldsymbol{\beta}_{N_2}$ are feasible, then $F_1(\boldsymbol{\alpha}_{N_1}) + F_2(\boldsymbol{\beta}_{N_2}) \leq \eta$. Therefore, $\Gamma(\boldsymbol{\alpha}_{N_1}^*, \boldsymbol{\beta}_{N_2}^*) \leq \Gamma(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2})$. Hence, if there exists $\boldsymbol{\alpha}_{N_1}^*$ and $\boldsymbol{\beta}_{N_2}^*$ such that $F_1(\boldsymbol{\alpha}_{N_1}^*) + F_2(\boldsymbol{\beta}_{N_2}^*) = \eta$, then it is the optimal solution.

Next, we show that there exists a $\lambda \geq 0$ such that $F_1(\boldsymbol{\alpha}_{N_1}^*) + F_2(\boldsymbol{\beta}_{N_2}^*) = \eta$.

1) When $\lambda = 0$, minimizing $L^\lambda(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2})$ in (40) reduces to minimizing $\Gamma(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2})$ without a constraint on $F(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2})$. In this case, the optimal solution is to have all the nodes transmit immediately. This also implies that a collision is inevitable in both stages. Thus, $F_1(\boldsymbol{\alpha}_{N_1}^*) = 1$ and $F_2(\boldsymbol{\beta}_{N_2}^*) = 1$. Therefore, $F_1(\boldsymbol{\alpha}_{N_1}^*) + F_2(\boldsymbol{\beta}_{N_2}^*) = 2$.

2) When $\lambda \rightarrow \infty$, minimizing $L^\lambda(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2})$ reduces to minimizing $F_1(\boldsymbol{\alpha}_{N_1}) + F_2(\boldsymbol{\beta}_{N_2})$. It can be seen that when $N_1 \rightarrow \infty$ and $N_2 \rightarrow \infty$, $F_1(\boldsymbol{\alpha}_{N_1}^*) \rightarrow 0$ and $F_2(\boldsymbol{\beta}_{N_2}^*) \rightarrow 0$. Therefore, $F_1(\boldsymbol{\alpha}_{N_1}^*) + F_2(\boldsymbol{\beta}_{N_2}^*) \rightarrow 0$.

Furthermore, from (16) and (25), $F_1(\boldsymbol{\alpha}_{N_1}^*)$ is a continuous function of λ . From (15) and (30), we see that $F_2(\boldsymbol{\beta}_{N_2}^*)$ is independent of λ . Thus, $F_1(\boldsymbol{\alpha}_{N_1}^*) + F_2(\boldsymbol{\beta}_{N_2}^*)$ is a continuous function of λ . Therefore, the existence result follows from the intermediate value theorem.

The decomposition into the two sub-problems \mathcal{SP}_1 and \mathcal{SP}_2 then follows because the objective function $L^\lambda(\boldsymbol{\alpha}_{N_1}, \boldsymbol{\beta}_{N_2})$ can be written as the sum of $\lceil \frac{k_1}{r} \rceil (N_2 \Delta + T_p) + \lambda F_2(\boldsymbol{\beta}_{N_2})$ and $\Gamma_1(\boldsymbol{\alpha}_{N_1}) + \lambda F_1(\boldsymbol{\alpha}_{N_1})$, which share no optimization variable in common.

D. Solution for $\alpha_\infty^*[0]$ in (29)

Upon substituting the expressions for $\Gamma_1(\boldsymbol{\alpha}_{N_1}^*)$ and $F_1(\boldsymbol{\alpha}_{N_1}^*)$ in (27) and (28), respectively, in (19), \mathcal{SP}_1 becomes

$$\mathcal{SP}_1 : \min_{\alpha_\infty^*[0]} T_p + \frac{\Delta(1 - \alpha_\infty^*[0])^{k_1}}{1 - (1 - \alpha_\infty^*[0])^{k_1}} - \lambda \left(1 - \frac{k_1 \alpha_\infty^*[0] (1 - \alpha_\infty^*[0])^{(k_1-1)}}{1 - (1 - \alpha_\infty^*[0])^{k_1}} \right), \quad (42)$$

$$\text{s.t. } 0 \leq \alpha_\infty^*[0] \leq 1. \quad (43)$$

Applying the first order condition, we get

$$\alpha_\infty^*[0] = 1 - \frac{(1 - \alpha_\infty^*[0])^{k_1} + k_1 - 1}{(\Delta/\lambda) + k_1}. \quad (44)$$

When $\alpha_\infty^*[0]$ increases from 0 to 1, the right hand side of (44) monotonically increases from $1 - \frac{k_1}{(\Delta/\lambda) + k_1} > 0$ to $1 - \frac{k_1 - 1}{(\Delta/\lambda) + k_1} < 1$. Therefore, the solution of (44) is unique, and lies in $(0, 1)$.

E. Proof of Result 3

We now prove that the solution in (30) achieves a lower bound on $F_2(\boldsymbol{\beta}_{N_2})$, which proves that it is optimal. Taking out the common factor $(1 - \beta_{N_2}[0])^{k_1 k_2}$ from the terms indexed by $l = 1, \dots, N_2$ in (15), we get

$$F_2(\boldsymbol{\beta}_{N_2}) = 1 - \kappa(1 - \beta_{N_2}[0])^{k_2-1} \times \left[1 - (1 - \beta_{N_2}[0])^{k_1 k_2 - k_2 + 1} \right] - (1 - \beta_{N_2}[0])^{k_1 k_2} \times \left[1 - F_2 \left(\frac{\beta_{N_2}[1]}{1 - \beta_{N_2}[0]}, \dots, \frac{\beta_{N_2}[N_2]}{1 - \beta_{N_2}[0]} \right) \right],$$

where $\kappa = \frac{k_1 k_2}{k_1 k_2 - k_2 + 1}$. If $\boldsymbol{\beta}_{N_2-1}^*$ is the optimum solution for $N_2 - 1$ timer levels, then we must have $F_2 \left(\frac{\beta_{N_2}[1]}{1 - \beta_{N_2}[0]}, \dots, \frac{\beta_{N_2}[N_2]}{1 - \beta_{N_2}[0]} \right) \leq F_2(\boldsymbol{\beta}_{N_2-1}^*)$. Therefore,

$$F_2(\boldsymbol{\beta}_{N_2}) \geq 1 - \kappa(1 - \beta_{N_2}[0])^{k_2-1} \times \left[1 - (1 - \beta_{N_2}[0])^{k_1 k_2 - k_2 + 1} \right] - (1 - \beta_{N_2}[0])^{k_1 k_2} [1 - F_2(\boldsymbol{\beta}_{N_2-1}^*)]. \quad (45)$$

Furthermore, the lower bound in (45) is achievable, and is obtained by setting $\frac{\beta_{N_2}[1]}{1 - \beta_{N_2}[0]} = \beta_{N_2-1}^*[0]$, $\frac{\beta_{N_2}[2]}{1 - \beta_{N_2}[0]} = \beta_{N_2-1}^*[1]$, \dots , $\frac{\beta_{N_2}[N_2]}{1 - \beta_{N_2}[0]} = \beta_{N_2-1}^*[N_2 - 1]$. Therefore,

$$F_2(\boldsymbol{\beta}_{N_2}^*) = \min_{0 \leq \beta_{N_2}[0] \leq 1} \left\{ 1 - (1 - \beta_{N_2}[0])^{k_1 k_2} [1 - F_2(\boldsymbol{\beta}_{N_2-1}^*)] - \kappa(1 - \beta_{N_2}[0])^{k_2-1} \left[1 - (1 - \beta_{N_2}[0])^{k_1 k_2 - k_2 + 1} \right] \right\}. \quad (46)$$

Using the first-order condition to minimize (46), we get

$$\beta_{N_2}^*[0] = 1 - \left(\frac{\kappa(k_2-1)}{\kappa-1+F_2(\boldsymbol{\beta}_{N_2-1}^*)} \right)^{\kappa/(k_1 k_2)}. \quad \text{Base Case of } N_2 = 0: \text{ Substituting } N_2 = 0 \text{ in (15), we get}$$

$$F_2(\boldsymbol{\beta}_0) = 1 - (1 - \beta_0[0])^{k_2-1} \left[1 - (1 - \beta_0[0])^{k_1 k_2 - k_2 + 1} \right]. \quad (47)$$

Using the first-order condition to minimize (47), we get $\beta_0^*[0] = 1 - ((k_2 - 1)/(k_1 k_2))^{\kappa/(k_1 k_2)}$. It can be verified that $0 < \beta_0^*[0] < 1$. Using induction, it can be shown that $\boldsymbol{\beta}_{N_2}^*$ is feasible.

F. Proof of Lemma 2

From Result 3, $\beta_0^*[0] = 1 - ((k_2 - 1)/(k_1 k_2))^{\kappa/(k_1 k_2)}$. Substituting this expression and $N_2 = 0$ in (15), we can show that $\lim_{k_1 \rightarrow \infty} F_2(\boldsymbol{\beta}_{N_2}^*) = 0$.

G. Proof of Result 4

From (44), it can be shown that $\lim_{k_1 \rightarrow \infty} k_1 \alpha_\infty^*[0]$ exists. Let it be equal to $d_\lambda > 0$, where d_λ depends on λ . Substituting this in (27) and (28), and taking limits on both sides, yields $\Gamma_1(\boldsymbol{\alpha}_\infty^*) = T_p + \frac{\Delta \exp(-d_\lambda)}{1 - \exp(-d_\lambda)}$ and $F_1(\boldsymbol{\alpha}_\infty^*) = 1 - \frac{d_\lambda \exp(-d_\lambda)}{1 - \exp(-d_\lambda)}$.

Next we solve for d_λ . Upon rearranging the terms in the expression for $\alpha_\infty^*[0]$ in (44), we get

$$(1 - \alpha_\infty^*[0])((\Delta/\lambda) + k_1) = (1 - \alpha_\infty^*[0])^{k_1} + k_1 - 1. \quad (48)$$

Taking the limit $k_1 \rightarrow \infty$ on both sides, we get

$$d_\lambda = 1 + (\Delta/\lambda) - \exp(-d_\lambda). \quad (49)$$

The right hand side of (49) is a monotonic increasing function of d_λ . It increases from Δ/λ to $(\Delta/\lambda) + 1$ when d_λ increases from 0 to ∞ . The left hand side of (49) instead starts lower at 0 but increases to ∞ when d_λ increases from 0 to ∞ . Therefore, there exists a unique solution for d_λ .

ACKNOWLEDGMENT

The second author would like to thank Prof. D. Manjunath, IIT Bombay, for introducing him to the problem.

REFERENCES

- [1] M. Ilya and I. Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. Boca Raton, FL, USA: CRC Press, 2004.
- [2] M. Li, Y. Wang, and Y. Wang, "Complexity of data collection, aggregation, selection for wireless sensor networks," *IEEE Trans. Comput.*, vol. 60, no. 3, pp. 386–399, Mar. 2011.
- [3] 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, Apr. 2005.
- [4] N. Karamchandani, R. Appuswamy, and M. Franceschetti, "Time and energy complexity of function computation over networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7671–7684, Dec. 2011.
- [5] S. Banerjee, P. Gupta, and S. Shakkottai, "Towards a queueing-based framework for in-network function computation," in *Proc. IEEE ISIT*, Jul. 2011, pp. 2542–2546.
- [6] H. Kowshik and P. R. Kumar, "Optimal computation of symmetric Boolean functions in collocated networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 639–654, Mar. 2013.
- [7] Y. Kanoria and D. Manjunath, "On distributed computation in noisy random planar networks," in *Proc. IEEE SIT*, Jun. 2007, pp. 626–630.
- [8] L. Ying, R. Srikant, and G. E. Dullerud, "Distributed symmetric function computation in noisy wireless sensor networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4826–4833, Dec. 2007.
- [9] A. Giridhar and P. R. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 98–107, Apr. 2006.
- [10] S. Kamath and D. Manjunath, "On distributed function computation in structure-free random networks," in *Proc. IEEE ISIT*, Jul. 2008, pp. 647–651.
- [11] R. Sappidi, A. Girard, and C. Rosenberg, "Maximum achievable throughput in a wireless sensor network using in-network computation for statistical functions," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1581–1594, Oct. 2013.
- [12] N. Khude, A. Kumar, and A. Karnik, "Time and energy complexity of distributed computation of a class of functions in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 617–632, May 2008.
- [13] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 659–672, Mar. 2006.
- [14] Q. Zhao and L. Tong, "Opportunistic carrier sensing for energy-efficient information retrieval in sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 2, pp. 231–241, Apr. 2005.
- [15] V. Shah, N. B. Mehta, and R. Yim, "Optimal timer based selection schemes," *IEEE Trans. Commun.*, vol. 58, no. 6, pp. 1814–1823, Jun. 2010.
- [16] Y. Xiao and L. J. Cimini, "Impact of overhead on spectral efficiency of cooperative relaying," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2228–2239, May 2013.
- [17] H. Ko, S. Pack, and W. Lee, "Timer-based push scheme for online social networking services in wireless networks," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2095–2098, Dec. 2012.
- [18] A. Kumar, D. Manjunath, and J. Kuri, *Wireless Networking*, 1st ed. San Mateo, CA, USA: Morgan Kaufmann, 2008.
- [19] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.
- [20] W. B. Heinzelman, "An Application-specific protocol architecture for wireless networks," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, Jun. 2000.
- [21] R. Talak and N. B. Mehta, "Feedback overhead-aware, distributed, fast, reliable selection," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3417–3428, Nov. 2012.
- [22] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks," in *Proc. INFOCOM*, Mar. 2004, pp. 1662–1672.
- [23] H. Kowshik and P. R. Kumar, "Zero-error function computation in sensor networks," in *Proc. 48th IEEE CDC/CCC*, Dec. 2009, pp. 3787–3792.
- [24] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-rate Wireless Personal Area Networks (WPANs)*, Tech. Rep. IEEE Std 802.15.4-2006, Jun. 16, 2006, IEEE.



Arjun Anand received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology, Calicut, India, in 2011 and the M.Eng. degree in telecommunications from the Indian Institute of Science, Bangalore, India, in 2013. Currently, he is pursuing the Ph.D. degree at the University of Texas, Austin, Texas, USA. From July 2013 to July 2014, he worked as Staff-I Engineer in Broadcom Communication Technologies Pvt. Ltd., Bangalore, India, and was involved in the development of products based on IEEE 802.11 ac

standard. His research interests include resource allocation in data networks, queuing theory, and game theory.



Neelesh B. Mehta (S'98-M'01-SM'06) received his B.Tech. degree in electronics and communications engineering from the Indian Institute of Technology (IIT), Madras, India, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, California, USA, in 1997 and 2001, respectively. He is now an Associate Professor in the Department of Electrical Communication Engineering, Indian Institute of Science (IISc), Bangalore, India. He serves as an Executive Editor of the IEEE TRANSACTIONS

ON WIRELESS COMMUNICATION, and as an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE WIRELESS COMMUNICATION LETTERS. He currently serves as a Member-at-Large on the Board of Governors of IEEE ComSoc.