On Distributed Function Computation in Structure-Free Random Wireless Networks

Sudeep Kamath, D. Manjunath, and Ravi Mazumdar, *Fellow, IEEE*

***Abstract*— We consider in-network computation of MAX and the approximate histogram in an n-node structure-free random multihop wireless network. The key assumption that we make is that the nodes do not know their relative or absolute locations and that they do not have an identity. For the Aloha MAC protocol, we first describe a protocol in which the MAX value becomes available at the origin in $O(\sqrt{n/\log n})$ slots (bit-periods) with high probability. This is within a constant factor of that required by the best coordinated protocol. A minimal structure (knowledge of hop-distance from the sink) is imposed on the network and with this structure, we describe a protocol for pipelined computation of MAX that achieves a rate of $\Omega(1/(\log n)^2)$. Finally, we show how the protocol for computation of MAX can be modified to achieve approximate computation of the histogram. The approximate histogram can be computed in $O(n^{7/2}(\log n)^{1/2})$ bit-periods with high probability.**

***Index Terms*— Communication and computation rate, distributed computation, in-network computation, structure free networks, wireless sensor networks.**

I. Introduction

THIS PAPER is motivated by the following scenario. n sensor nodes are distributed randomly in a sensor field. Each sensor node makes a measurement of a physical variable at a rate called the refresh rate. A collector, also called a sink node, needs to obtain some function, say $f(\cdot)$, of the measurements and not the actual value of the measured variables themselves. The sensor nodes communicate over a wireless channel. Our interest is in distributed protocols—sequence of communication and computation steps—that efficiently obtain $f(\cdot)$ at the sink node with little or no knowledge of the network structure and without a central control. Energy expended in the computation, the delay between the measurement instant and the time at which $f(\cdot)$ is available, and the rate at which $f(\cdot)$ can be refreshed are the performance measures of interest. See [9] for a more comprehensive discussion of the motivation of the problem.

Manuscript received July 27, 2009; revised May 31, 2013; accepted September 25, 2013. Date of publication November 27, 2013; date of current version December 20, 2013.

S. Kamath is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: sudeep@eecs.berkeley.edu).

D. Manjunath is with the Department of Electrical Engineering, IIT Bombay, Mumbai 400076, India, and also with the Bharti Centre for Communication, IIT Bombay, Mumbai 400076, India (e-mail: dmanju@ee.iitb.ac.in).

R. Mazumdar is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: mazum@ece.uwaterloo.ca).

Communicated by R. D. Yates, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2013.2293214

A. Previous Work

Early work on computation of functions of binary data over wireless networks focused on computing over noisy, time-slotted, broadcast (single-hop or collocated) networks of n nodes, see [7], [6], [15], and [16]. With increasing interest in wireless sensor networks, which can have a dense deployment, it is natural to exploit spatial reuse for increased performance. To obtain limit laws, the alternative model of an 'extended network' (the area of the sensor field is increased while maintaining a fixed density of the nodes) is also considered in the literature. Recent research has concentrated on 'in-network' computation over multihop wireless networks, see [8], [9], and [14]. In [8] and [9], the links are assumed to be noisefree, i.e., there are no link errors and all bits are received correctly by all the intended receivers. Alternatively, block coding is assumed to provide error free communication over possibly noisy links. Of course, block coding incurs delays corresponding to accumulating the data for a block. In [14], protocols to compute the MAX in a pipelined manner in a network with noisefree links are described and analyzed. A protocol to compute the histogram in an n-node multihop network with noisy links is described in [19]. This protocol requires $O(n \log \log n)$ bit transmissions to obtain the correct histogram at the sink with high probability. [13] describes a protocol to calculate the MAX in a multihop network with noisy links in $O(n)$ transmissions with high probability. Using the results from [10], it was shown in [5] that the protocol described in [19] is order optimal over all possible protocols.

In all of the above literature, the network is assumed to be time slotted and the primary focus is to define an oblivious protocol that identifies the nodes that are to transmit in each slot and the value of the data that is to be transmitted. This implies that the nodes have organized themselves into a network and also have their clocks synchronized. Both of these require significant communication and computation effort. In this paper, we do away with the assumption that the nodes can organize themselves. In fact, we will make an extreme assumption—the nodes do not even have an identity. This of course implies that we cannot define a structure in the network and hence we cannot define a collision-free protocol. We will say that such a network is *structure-free.* Nodes with minimal knowledge of the topology of the network has been previously considered in the literature, in [1]–[4], the nodes in the multihop radio network know only the neighbourhood topology. The interest in these works is in the analysis and design of broadcasting protocols in which data from a source needs to reach all the nodes.

The rest of this paper is organized as follows. In Section II we describe the problem setting and the notation. In Section III we describe a protocol for in-network computation of MAX in a structure-free network, that uses the Aloha MAC protocol. We first describe the One-Shot MAX protocol for one-shot computation of the MAX and its analysis. We show that, with high probability (w.h.p.), the sink will have the result in a time that is within a constant factor of that required by a structured network. Next we consider continuous computation of $f(\cdot)$. For this we impose a minimal structure on the network and describe the Pipelined MAX protocol and its analysis. We show that the rate of computing the MAX in this network is $\Omega(1/(\log n)^2)$. In Section IV we describe the One-Shot Histogram protocol which uses the One-Shot MAX to compute the fractional histogram in $O(n^{7/2}(\log n)^{1/2})$ bit periods w.h.p. We conclude in Section V with a discussion. The proofs are detailed in the Appendix.

II. Problem Definition and Notation

Nodes $1, 2, \ldots n$ are assumed to be distributed over the unit square $[0, 1]^2$ in the two-dimensional Euclidean plane. Every node in the network knows the value of n. Node i is located at x_i and has data Z_i. Define $\mathbf{x} := [x_1, \ldots, x_n]$ and $\mathbf{Z} := [Z_1, \ldots, Z_n]$. We will assume that Z_i belongs to a finite set and, without loss of generality, we will assume that Z_i are binary data, i.e., $Z_i \in \{0, 1\}$. $f(\mathbf{Z})$ needs to be computed and made available at a sink node, say Node s. We will assume that x_i are uniformly and independently distributed in $[0, 1]^2$, the sink node is at the origin, and Z_i are arbitrary binary data. x_i is not known to Node i and nodes are not labelled or have an identity. The indexing of the nodes is used only for ease of description. All logarithms in this paper are natural logarithms.

In most sensor networks, a set of specific functions will be of interest and we exploit the nature of $f(\cdot)$ in the design of efficient protocols. The primary interest for distributed computation in sensor networks is for separable functions. Separable functions can be computed by partitioning the elements of $\mathbf{Z}$, performing intermediate computations on each partition, and suitably combining the results from the individual computations on the partitions of $\mathbf{Z}$ to obtain $f(\mathbf{Z})$. Another classification identifies symmetric functions in which the value of the function is invariant to the permutation of the elements of $\mathbf{Z}$. Type-sensitive and type-threshold are a further classifications of symmetric functions that are of interest in sensor networks. See [8] for a more detailed discussion on the classification of the functions of interest to sensor networks. In this paper, we will focus on the computation of two functions—MAX (a type-threshold function) and histogram (a type-sensitive function).

As was mentioned earlier, we will define a protocol to perform a distributed computation of $f(\mathbf{Z})$. Our interest is in developing an oblivious protocol in which, at any time, the decision by a node to transmit does not depend on the data (or partially computed function) values that it has received earlier.

III. MAX in a Structure Free Multihop Aloha Network

Since the nodes are not organized and do not know their neighbours, we cannot define a deterministic protocol; a random access MAC protocol is an obvious choice. For pedagogical convenience, we will assume slotted-Aloha at the MAC layer with each slot accommodating one data bit. The analysis easily extends to the case of pure Aloha MAC where we can also assume that the nodes are not time synchronized to the slot boundaries.

Spatial reuse is modeled using the following variant of the protocol model of interference of [11]. Consider a transmitter at location x_1 transmitting in a slot t. A receiver at location x_2, can successfully decode this transmission if the following two conditions are satisfied: (1) $\|x_2 - x_1\| < r_n$, and (2) $\|x_2 - x_3\| > (1 + \Delta')r_n$. Here r_n is called the transmission radius and is determined by the transmission power, $\Delta' \geq 0$, is a constant and is determined by the receiver sensitivity to interference, and x_3 is the location of any other node transmitting in slot t. The choice of r_n is discussed later in this section. In this paper, a transmission in slot t is deemed successful if all nodes within r_n of the transmitter decode it successfully. It is easy to see that the following is a sufficient condition for successful transmission by a node located at x in a slot: $\|x - x'_k\| > (1 + \Delta)r_n$, $\Delta = 1 + \Delta'$, for every x'_k where x'_k is the location of the k-th transmitter in the slot; see Fig. 1 for an illustration. Note that the requirement for a successful transmission defined above is stricter than that in the protocol model.

In the following we describe two protocols—One-shot MAX will compute the MAX for one instance of the data, and Pipelined MAX which will compute the MAX function for a sequence of data values.

A. One-Shot Computation of MAX Using Aloha

Let Z_i be the value of the one-bit data at Node i and $\mathcal{Z} := \max_{1 \leq i \leq n} Z_i$. The protocol One-Shot MAX is as follows. Like in all wireless networks, we assume that a node cannot receive when it is transmitting. Thus in slot t, Node i will either transmit with probability p, or listen with probability $(1 - p)$, independently of all other transmissions in the network. Let $X_i(t)$ be the value of the bit received (i.e., correctly decoded in the absence of a collision) by Node i in slot t, $t = 1, 2, \ldots$. If Node i transmits in slot t or if it senses a collision or idle in the slot, then it sets $X_i(t) = 0$. Define $Y_i(0) = Z_i$ and $Y_i(t) := \max\{Y_i(t-1), X_i(t)\}$ for $t = 1, 2, \ldots$. $Y_i(t)$ is the 'running MAX' at Node i in slot t. If Node i transmits in slot t, it will transmit $T_i(t) = Y_i(t-1)$. This is summarized in Algorithm 1.

It is easy to see that the value of $\mathcal{Z}$ will 'develop and diffuse' through the network. Our interest is in the time for the correct value to have diffused through the network. The performance of the protocol, that is, the time for the correct value to diffuse to the edges of the network, depends on p. The choice of p is discussed in the Appendix.

To study the progress of the diffusion, we will consider a tessellation of the unit square into square cells of side $s_n = \left(\left\lceil \sqrt{n/(2.75 \log n)} \right\rceil\right)^{-1}$. This will result in $l_n := 1/s_n =$

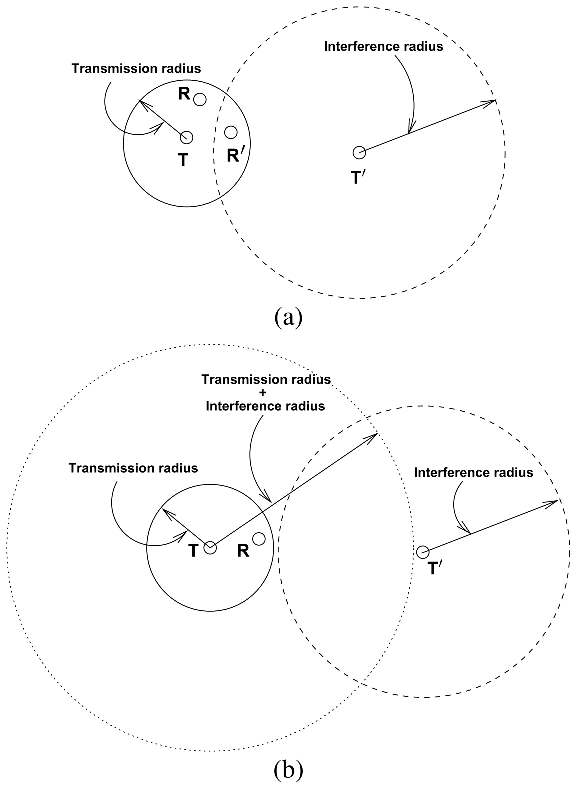

Fig. 1. Illustrating the difference between the original protocol model and the stricter condition used in this paper for a successful transmission. (a) The protocol model of interference. When the node at T' is transmitting, the node at R' cannot receive successfully from the node at T but the node at R can. (b) A sufficient condition for successful transmission by node at T as defined in Section III is that another transmitting node (e.g., T') should be outside the dotted circle. This is stricter than the sufficient condition of the protocol model of [12].

$\left\lceil \sqrt{n/(2.75 \log n)} \right\rceil$ rows (and columns) of cells in $[0, 1]^2$. There will be a total of $M_n := 1/s_n^2 = \left(\left\lceil \sqrt{n/(2.75 \log n)} \right\rceil\right)^2$ cells. Let $\mathcal{C}$ denote the set of cells under this tessellation, S_c the set of nodes in Cell c and N_c the number of nodes in Cell c. Under this tessellation, two cells are said to be *adjacent* if they have a common edge. Let the transmission radius of all the nodes be $r_n = \sqrt{(13.75 \log n)/n} \approx \sqrt{5} s_n$. For this value of r_n the network is connected w.h.p. [11]. The expected number of nodes in a cell is $n s_n^2 \approx 2.75 \log n$. Further, from Lemma 3.1 of [18], for our choice of r_n and s_n,

$$\Pr\left(c_1 \log n \le N_c \le c_2 \log n \quad \text{for} \quad 1 \le c \le M_n\right) \;\to\; 1 \qquad (1)$$

where $c_1 = 0.091$ and $c_2 = 5.41$.

The results that we describe in this paper will hold for networks that satisfy the condition $c_1 \log n \le N_c \le c_2 \log n$ for all cells c. From the choice of r_n, i.e., $r_n \ge \sqrt{5} s_n$, a successful transmission (as defined earlier) by any node from Cell c is correctly decoded by all nodes in Cell c as well as by all nodes in cells adjacent to Cell c.

Recall that we need to find the time that it will take for $\mathcal{Z}$ to diffuse to Node s. The value of $\mathcal{Z}$ can reach the sink along any of the many possible trees rooted at Node s. For our analysis, we will divide the progress of the diffusion into the following three phases and analyze each of the three phases separately. We reiterate that this sequence of phases is for the purpose of analysis of the time to complete the diffusion. The nodes do not perform any such organization.

Algorithm 1: One-Shot MAX: The Following Protocol Is Executed at Node i in Each Slot. We Assume That th Received Bit $X_i = 0$ if There Is a Collision in a Slot or if There Is No Reception in That Slot

$r = \sqrt{(13.75 n \log n)/n}$
$p = 1/(c_1 \log n)$
input: Z_i
$Y_i \leftarrow Z_i$
for $t = 1$ ***to*** ∞ **do**
  Toss coin with bias p
  if *heads* **then**
    Transmit $Y_i(t-1)$
    $X_i(t) \leftarrow 0$
  end
  else
    $X_i(t) \leftarrow$ Received bit
  end
  $Y_i(t) \leftarrow \max(Y_i(t-1), X_i(t))$
end

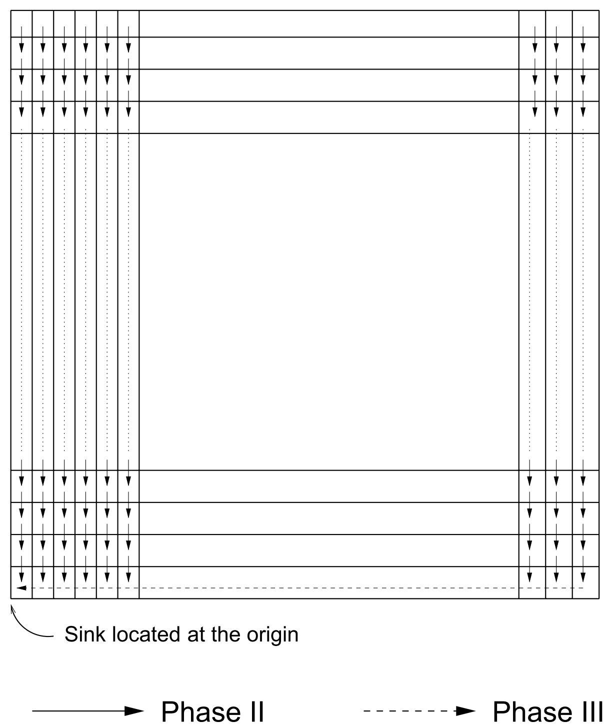

Fig. 2. Direction of diffusion during Phase II and Phase III of the analysis of the protocol One-Shot MAX.

- *Phase I for data aggregation within each cell.* Since we assume error free links, this phase is completed when every node of the network has transmitted successfully at least once. At the end of this phase every node will have received the data from every other node in its cell and hence know the MAX for its cell.
- *Phase II for progress to the bottom of the square.* In this phase, the locally computed values of the MAX diffuse

into the cells on one side of the unit square as shown in Fig. 2. At the end of this phase, at least one node in each cell of the bottom row has the value of the MAX of the cells in its column.

- *Phase III for progress into the sink.* In this phase, the value of MAX reaches the sink at the origin in the manner shown in Fig. 2.

We show in the Appendix that Phase I will be completed in $O((\log n)^2)$ slots w.h.p., Phase II and Phase III will each be completed in $O\left(\sqrt{n/(\log n)}\right)$ slots w.h.p. This gives us the following theorem.

Theorem 1: If all the nodes execute protocol One-Shot MAX, then for any $\alpha, k > 0$, the maximum of the binary data at the n nodes is available at the sink with probability at least $(1 - k/n^{\alpha})$ in $O\left(\sqrt{n/(\log n)}\right)$ slots.

Proof: See Appendix B. ∎

Note that the best one-shot protocol in an organized network under this choice of r_n will require $\Theta\left(\sqrt{n/(\log n)}\right)$ slots for a one-shot computation of MAX. This is because the number of hops from one end of the network to another is at least $\left\lfloor \frac{1}{r_n} \right\rfloor$. The bound on the diffusion time in Theorem 1 is therefore tight.

B. Pipelined Computation of MAX Using Aloha

If a sequence of $\mathbf{Z}$ were available, then we would want to compute $\mathcal{Z}$ repeatedly at a fast rate. Using the One-Shot MAX protocol, and $r_n = \Omega\left(\sqrt{(\log n/n)}\right)$, a throughput (also called refresh rate) of $\Omega\left(\sqrt{(\log n)/n}\right)$ can be achieved. It can be see that number of hops to the destination, which is $O\left(\sqrt{n/(\log n)}\right)$, decides this rate. We believe that some structure in the network is necessary to do better. In the following, we assume one such minimal structure, describe a protocol for use with this structure, and derive the maximum rate at which computation will occur in the network. We will also assume that all nodes have a transmission range that is exactly r_n. In One-shot MAX transmissions from a node can be decoded by nodes further than r_n and it will not affect the correctness of the protocol. In Pipelined MAX, nodes further than r_n from the transmitter *should not* decode for correct operation of the protocol. This strict requirement can be relaxed but we will keep this assumption for pedagogical convenience.

We impose the following structure in the network. Prior to the computation, each node obtains its minimum hop distance to the sink. Henceforth, we will refer to this as simply the *hop distance* of the node. From (1), with high probability, each cell in the tessellation is occupied. Since nodes in adjacent cells differ in their hop distance by at most 1, the largest hop distance of a node in the network is upper bounded by $d := 2\,l_n = 2\left\lceil \sqrt{n/(2.75 \log n)} \right\rceil$.

Let h_i be the hop distance of Node i. Since r_n is exact, a transmission by Node i can be decoded successfully by Node j only if $|h_i - h_j| \le 1$. Hence, if there is a reception by Node i in slot t, then that transmission must have been made by a node with hop distance either $(h_i - 1)$, h_i, or $(h_i + 1)$. Thus, if a node transmits its hop distance modulo 3 along with its transmitted bit, then every receiver that can decode this transmission successfully, can also, by the receiver's knowledge of its own hop distance, correctly identify the hop distance of the transmitter. Let $A_i := h_i \mod 3$. A_i requires two bits to code. A_i will be called the identification bits of Node i.

For the protocol, time is divided into *rounds* where each round consists of τ slots. It is assumed that the nodes know the value of τ; we have earlier said that they also know the value of n. For a given n, the choice of τ is derived below. Data arrives at each node at the beginning of each round, that is, at the rate of 1 data bit per round. Minimizing τ would maximize the throughput. Let the value of the bit at Node i in round r be $Z_i(r)$. $\mathcal{Z}(r) := \max_{1 \le i \le n} Z_i(r)$, for $r = 1, 2, \ldots,$ is to be made available at the sink node, Node s.

Pipelined MAX protocol works as follows. The sink only receives data and does not transmit. The other nodes in the network perform the following. It is reiterated that the naming of the nodes is for convenience in description and the nodes themselves do not have an identity.

In each slot, Node i transmits with probability p or listens with probability $(1 - p)$ independently of all other transmissions in the network. The value of p is chosen as in the One-Shot MAX protocol. Each node executes the following protocol for round r. Since each round lasts for exactly τ slots, the nodes can count the slots to keep track of the beginning and end of a round.

Transmission: If Node i transmits in slot t of round r, then it transmits $(A_i, T_i(r))$ in the slot where A_i is the two-bit identification of node i and $T_i(r)$ is the one-bit data obtained as

$$T_i(r) = \max\{Z_i(r - (d - h_i)), Y_i(r - 1)\}. \tag{2}$$

Here, by convention, $Z_i(r) = Y_i(r) = 0$ for $r \le 0$. $Y_i(r - 1)$ is computed from successful receptions in round $(r - 1)$, as described below. Thus each slot is three bit-periods.

Reception: In round r, let $Y_i(r, t)$ be the MAX of the transmissions by nodes with hop distance $(h_i + 1)$ in slots up to and including t, that Node i has decoded successfully. $Y_i(r, t)$ is computed as follows. $Y_i(r, 0)$ is initialized to 0 at the beginning of round r. In slot t of round r, if Node i successfully receives a transmission from a node with hop distance $(h_i + 1)$ (available from the identification bits), then the received data bit is assigned to $X_i(r, t)$. If Node i senses an idle or a collision in slot t, or if it receives a successful transmission from a node from $(h_i + 1)$, then it sets $X_i(r, t) = 0$. Thus, $Y_i(r, t) = \max\{Y_i(r, t - 1), X_i(r, t)\}$. Define $Y_i(r) := Y_i(r, \tau)$. The above is summarized in Algorithm 2.

The sink node, Node s, obtains the MAX as $\mathcal{Z}(r - d) = \max\{Z_s(r - d), Y_s(r)\}$, for all $r > d$. The delay of the protocol is d rounds or $d\tau$ time slots. The achievable throughput, or the refresh rate, of this protocol is described by the following theorem.

Theorem 2: If all the nodes execute the protocol Pipelined MAX, then for any $\alpha, k > 0$, there exists $\tau = \tau(\alpha, k) = \Theta((\log n)^2)$ so that the correct MAX is available at the sink in a round with probability at least $(1 - k/n^{\alpha})$. This achieves a

Algorithm 2: Pipelined MAX: The Following Protocol Is Executed at Node i in Each Slot

```
input     : h_i
A_i ← h_i  mod 3
initialize: Z_i(r) = Y_i(r) = 0 for r ≤ 0.
for r = 1 to ∞ do
    input    : Z_i(r)
    Y_i(r, 0) = 0
    for t = 1 to τ do
        Toss coin with bias p
        if heads then
            T_i(t) ← max{Z_i(r − (d − h_i)), Y_i(r − 1)}
            Transmit (A_i, T_i(r))
        end
        else
            if successful reception then
                ĥ ← 2-bit identification in received packet
                if ĥ = (h_i + 1  mod 3) then
                    X_i(r, t) ← data bit in received packet
                    Y_i(r, t) = max{Y_i(r, t − 1), X_i(r, t)}.
                end
            end
        end
    end
    Y_i(r) = Y_i(r, τ)
end
```

throughput of $\Omega\left(1/(\log n)^2\right)$ with a delay of $O(\sqrt{n(\log n)^3})$ slots.

Proof: See Appendix D. ■

Node i, with a hop distance of h_i, requires a memory of $(d - h_i + 1)$ bits to store $Z_i(r), Z_i(r-1), \ldots, Z_i(r-d+h_i)$. Thus, the protocol requires $(d+1)$ bits of memory at each of the nodes for storage of past data values.

Remark 1: The optimal pipelined protocol for MAX in an organized network that does not use block coding requires $\Theta(\log n)$ slots for each round. Thus, the penalty for minimal organization and no coordination is at most the $\log n$ overhead for the duration of each round.

Remark 2: To be able to pipeline the computations, and hence achieve higher throughput, Pipelined MAX needs to compute a spanning tree of the random network rooted at the sink. Finding the hop distance is an equivalent problem; thus knowing the hop distance is necessary and it is also a standard assumption in much of the literature of in-network computation in random spatial networks.

Remark 3: With s-Aloha protocol for the MAC layer, this time penalty cannot be improved. The argument is as follows. By the choice of the attempt probability in a slot, each transmission attempt is successful, i.e., is successfully decoded by all the one-hop neighbors of the transmitter, with a constant probability independent of n. A 'local neighbourhood' of area $O(r_n^2)$, contains $O(\log n)$ nodes and successful transmissions occur at a constant rate in this area. Thus we need at least $O(\log n)$ time for each node in the neighborhood to have attempted transmission at least once. This is a lower bound on the amount of time required for every node to have transmitted successfully at least once for any protocol. However, in the unstructured network with s-Aloha we need some more time. Since each node is transmitting in a slot with probability $p = 1/(c' \log n)$ for some constant c', the interval between the transmission attempts of a node has a geometric distribution with mean $(c' \log n)$. To complete one stage of the pipeline, every node in the network should have transmitted at least once. The time for this to happen is the maximum of n such geometric random variables. The mean of this time is $\Omega\left((\log n)/p\right) = \Omega((\log n)^2)$. The proof essentially shows that in this much time it is in fact, guaranteed that every node has transmitted successfully (not just transmitted) at least once with high probability. Thus the bound on the throughput in Theorem 2 is tight if we are to use s-Aloha as the MAC.

IV. Histogram in Structure-Free Networks

Computation of the histogram in a structure-free network needs more work. Rather than compute the histogram exactly, we will obtain the histogram of fractions, i.e., we will obtain the fraction of nodes with bit '0' and the fraction of nodes with bit '1'.

In Section IV-A we obtain the background results necessary to estimate the cardinality of a set using a MAX function. In Section IV-B, we derive the results for the case when the numbers have to be represented using finite precision. In Section IV-C we describe the One-Shot Histogram protocol and analyze its performance.

A. Estimating the Cardinality of a Random Set Using a MAX Function

We begin by first considering the following hypothetical experiment. An urn has N identical balls where N is an unknown (large) number. We sample with replacement one ball per sample from the urn. A countably infinite number of samples are drawn so that with probability 1, each ball is sampled at least once. But the sampler has memory of just one number. The memory restriction and the sampling method ensure that N cannot be estimated by counting. To help estimate N, assume that each ball in the urn is labelled with an exponential random variable of unit mean. Let M be the maximum of the labels of the balls. With probability 1, this will be the same as the maximum of the balls that have been sampled. The distribution and expectation of M are

$$F_M(t) = (1 - e^{-t})^N, \quad t \geq 0 \tag{3}$$

$$\mathbb{E}[M] = \int_0^\infty (1 - F_M(t))dt \tag{4}$$

$$= \sum_{i=1}^{N} \frac{1}{i} = \log N + \gamma + \xi_N \tag{5}$$

where $\xi_N \to 0$ as $N \to \infty$ and $\gamma \approx 0.577$ is the Euler-Mascheroni constant ($\lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{i} - \log n = \gamma$).

We can have $\hat{N} := e^{M-\gamma}$ as an estimator for N. We would like $\hat{N}$ to be within a constant factor of N, i.e., $\frac{N}{1+\delta} \leq \hat{N} \leq N(1+\delta)$ or

$$|\log \hat{N} - \log N| \leq \log(1+\delta), \tag{6}$$

where δ will be the *accuracy* of the computation. Note $M - \mathbb{E}[M] = \log \hat{N} - \log N - \xi_N$. For the rest of this analysis, we will assume that N is large enough so that $\log(1+\delta) > |\xi_N|$. We repeat the above algorithm l times and let $\bar{M}_l$ be the sample mean of l independent instances of the MAX of the exponentially distributed random variables with mean 1 generated by each element. If we use the estimator $\hat{N}_l := e^{\bar{M}_l - \gamma}$, then we would have

$$|\log \hat{N}_l - \log N| \leq \log(1+\delta), \tag{7}$$

as long as

$$|\bar{M}_l - \mathbb{E}[M]| \leq \log(1+\delta) - |\xi_N|. \tag{8}$$

B. Using Finite Precision

To make the above idea useful in our problem of computation of histogram in the multihop network, we have to consider the effect of finite precision in the representation of the exponential random variable. The random variables are drawn from a continuous distribution, but the data that is to be transmitted has to be constrained to be from a finite set. This brings up the need for truncation and quantization.

1) Truncation: The following lemma bounds the probability that the maximum of n exponential random variables exceeds a threshold.

Lemma 1: Let R_i, $i = 1, 2, \ldots, N$ be independent unit mean exponential random variables. Then, for any constant $r > 0$, we have $\Pr\left(\max_{1 \leq i \leq N} R_i > (r+1)\log N\right) \leq N^{-r}$.

This enables us to make the following change in the algorithm. If an element of S generates an exponential random variable with value greater than $(r+1)\log N$, then it resets its value to 0. The requirements on N will be described in the Appendix.

2) Quantization: The range $[0, (r+1)\log N]$ may be quantized uniformly in steps of q, the choice of which will be described later. The elements of the set, upon being queried, will respond with the quantized value of the truncated exponential random variable.

The maximum of these truncated and quantized values can be used to estimate the cardinality of the set.

C. Algorithm for the Histogram

The preceding discussion motivates the following protocol to estimate the histogram in a structure-free multihop network using the One-Shot MAX protocol. In the following we will explain one iteration. The other $k-1$ iterations can be pipelined.

Protocol One-Shot Histogram: Let Z_i be the data bit at Node i. Each node generates an independent exponential random variable of mean 1. In each time slot, each node transmits with probability p and listens with probability $(1-p)$, independently of all other nodes, p being chosen as in Protocol One-Shot MAX. Let E_i be the value of the truncated exponential random variable generated by Node i. Let $\overline{E}_i$ be the quantized value of E_i. If Node i transmits in slot t, or if it detects no transmission or a collision in that slot, then it sets $(X_{0,i}(t), X_{1,i}(t)) = (0, 0)$. If it receives data $(R_{0,i}(t), R_{1,i}(t))$ in slot t, then it sets $(X_{0,i}(t), X_{1,i}(t)) = (R_{0,i}(t), R_{1,i}(t))$. If Node i transmits in slot t, then it transmits $(T_{0,i}(t), T_{1,i}(t))$ according to the following scheme.

- If $Z_i = 0$, then $T_{0,i}(t) = \max\{\max_{0<t_1<t} X_{0,i}(t_1), \overline{E}_i\}$, and $T_{1,i}(t) = \max_{0<t_1<t} X_{1,i}(t_1)$.
- If $Z_i = 1$, then $T_{0,i}(t) = \max_{0<t_1<t} X_{0,i}(t_1)$ and $T_{1,i}(t)) = \max\{\max_{0<t_1<t} X_{1,i}(t_1), \overline{E}_i\}$.

We will see later that it is sufficient to encode $T_{0,i}(t)$ and $T_{1,i}(t)$ using $\log n$ bits. Thus each slot is $\log n$ bit-periods. This protocol, illustrated in Fig. 3, will result in correct computation of $(\overline{\mathcal{E}}_0, \overline{\mathcal{E}}_1) = (\max_{i:Z_i=0} \overline{E}_i, \max_{j:Z_j=1} \overline{E}_j)$ after $O\left(n^{7/2}(\log n)^{1/2}\right)$ time bit-periods with high probability. $(\overline{\mathcal{E}}_0, \overline{\mathcal{E}}_1)$ are then used to estimate the histogram.

Theorem 3: If all the nodes execute the protocol One-Shot Histogram, then the fractional histogram $(\frac{\hat{n}_0}{n}, \frac{\hat{n}_1}{n})$ is available to all nodes in $O\left(n^{7/2}(\log n)^{1/2}\right)$ bit-periods with high probability as given by:

$$\Pr\left(\frac{n_b}{n} e^{-\frac{3}{2n}} e^{\xi_{n_b}} \leq \frac{\hat{n}_b}{n} \leq \frac{n_b}{n} e^{\frac{3}{2n}} e^{\xi_{n_b}}\right) \geq 1 - O\left(\frac{1}{n}\right), \tag{9}$$

for $b = 0, 1$ where $\xi_{n_b} \to 0$ as $n_b \to \infty$.

Proof: See Appendix F. ■

Remark 4: If the fraction of nodes with bit '0', i.e. $\frac{n_0}{n}$ is bounded away from 0 and 1, then as $n \to \infty$, the quantity $e^{\xi_{n_b}}$ approaches 1 for $b = 0, 1$.

The above algorithm can be used to compute any continuous function of the histogram. Examples of such functions are mean, sample variance, moments of any finite order. However, functions of the histogram which are discontinuous, e.g., mode and parity, may result in large errors.

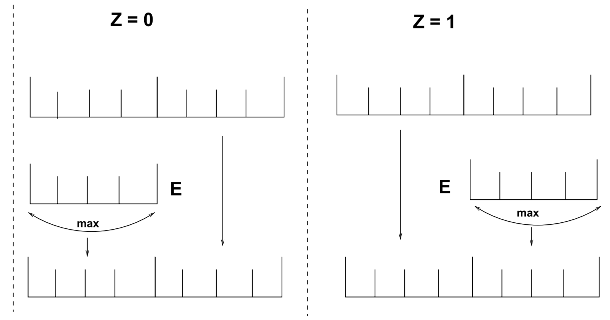

Fig. 3. Local algorithm at each node for computation of MAX. Each slot receives two numbers of $\log n$ bits each. Depending on the value of Z_i, one of the numbers is relayed and the other one is compared with its exponential random variable and the maximum of the two is transmitted.

V. Discussion

We conclude with some remarks.

1) Our focus has been on obtaining the throughputs possible in a structure free network. There will of course be an energy penalty. The total number of transmissions (successful as well as unsuccessful) in one execution of One-Shot MAX is $\Theta((n^{3/2})/(\log n)^{3/2})$. In the Pipelined MAX, a total of $\Theta(n \log n)$ transmissions are made per round. Note that the corresponding number is $\Theta(n)$ with a coordinated protocol for both cases.

2) In computing the histogram, rather than use the maximum of the exponential random variables, we could also use the minimum. See [17] for a discussion on the truncation and the quantization when the minimum is used for the approximate histogram.
3) Broadcast networks have been considered in [8] and a structure-free broadcast network is also possible. It is fairly straightforward to show that in a noiseless, structure-free broadcast network, the histogram can be computed in $\Theta(n)$ slots w.h.p. In the noisy broadcast network, by a simple modification of the protocol of [7], we can show that the histogram can be computed in $\Theta(n \log\log n)$ slots w.h.p.
4) The protocol One-Shot Histogram can be modified along similar lines to obtain protocol Pipelined Histogram which would achieve a throughput of $\Omega(1/(n^3(\log n)^3))$.
5) Our analysis can be extended to the case where the nodes use pure Aloha as the MAC. We need to use a transmission rate rather than a transmission probability. The success probabilities are calculated similarly except that we now have a collision window that is twice the packet length. All other calculations are analogous.
6) Rather than use s-Aloha, the MAC protocol can be made adaptive. This is future work.

Appendix

A. Preliminaries

1) Bounding the Number of Interfering Neighbors: Define the interfering neighborhood of Node i by $\mathcal{N}_i^{(I)} := \{j : 0 < \|x_i - x_j\| \le (1+\Delta)r_n\}$, where x_j is the location of Node j. As discussed earlier, a transmission from Node i in slot t is deemed successful if all nodes within r_n of Node i can decode this transmission without a collision. A sufficient condition for Node i to be successful in transmitting in slot t is that no node belonging to $\mathcal{N}_i^{(I)}$ must transmit in slot t.

From the protocol model, the choice of s_n and (1), the set of nodes that interfere with a transmission from a node in Cell c, i.e., $\bigcup_{i\in S_c} \mathcal{N}_i^{(I)}$, is contained within an interference square centered at Cell c. This square contains $k_1 = (2\lceil((1+\Delta)r_n)/s_n\rceil + 1)^2$ cells. From (1),

$$|\mathcal{N}_i^{(I)}| \le k_1 c_2 \log n - 1 \tag{10}$$

Observe that k_1 is a constant for large enough n.

2) Probability of a Successful Transmission from a Cell: Let P_i be the probability that Node i transmits successfully in a slot and $P^{(c)}$, the probability that some node in Cell c transmits successfully in a slot. $P_i \ge p(1-p)^{|\mathcal{N}_i^{(I)}|}$, and from (10), we have $P_i \ge p(1-p)^{k_1c_2\log n - 1}$. Successful transmissions by nodes from Cell c are mutually disjoint events, and hence, $P^{(c)} = \sum_{i\in S_c} P_i \ge N_c p(1-p)^{k_1c_2\log n-1}$. From (1), we have $N_c \ge c_1 \log n \ \forall c \in \mathcal{C}$ and hence, $P^{(c)} \ge c_1 \log n \ p(1-p)^{k_1c_2\log n -1}$. Choosing $p = \frac{1}{k_1c_2\log n}$ maximizes the lower bound in this inequality and yields

$$\begin{aligned} P^{(c)} &\ge \frac{c_1}{k_1c_2}\left(1 + \frac{1}{k_1c_2\log n - 1}\right)^{-(k_1c_2\log n - 1)} \\ &\ge \frac{c_1}{k_1c_2e} =: p_S \end{aligned} \tag{11}$$

Thus, with p chosen as above, the probability of successful transmission from a cell is lower bounded by a constant p_S, independent of the number of nodes in the network. This will be crucial to our analysis.

B. Proof of Theorem 1

We will prove Theorem 1 by obtaining bounds on the total time required for each of phases I, II and III.

1) Phase I: Data aggregation within each cell: Consider Cell c. Let $\mathcal{T}_c$ be the total number of slots required for every node in Cell c to have transmitted successfully at least once. Recall that $p = (k_1c_2\log n)^{-1}$. We will bound $\mathcal{T}_c$ by stochastic domination.

Consider a bin with N_c white balls numbered 1 through N_c, each of weight $p(1-p)^{k_1c_2\log n-1}$ and 1 black ball of weight $1 - N_c p(1-p)^{k_1c_2\log n -1}$. Consider a sequence of samples with replacement drawn from this bin, where the probability of picking up a ball of weight w is equal to w. (Note that the weights add up to 1, so this is fine.) Let T_c' be the number of samples drawn with replacement until each white ball is picked up at least once. This is essentially the well known coupon collector problem except that we can also draw a blank.

Let R_c' denote the number of white balls (not necessarily distinct) sampled up to time T_c'. By the standard coupon collector analysis, R_c' is the sum of independent geometrically distributed random variables with parameters $(1-(l-1)/N_c)$, i.e., $R_c' = \sum_{l=1}^{n} U_l$ where U_l are independent and $U_l \sim$ Geom $(1-(l-1)/N_c)$. (U_l can be interpreted as the number of white balls picked between the sampling of the $(l-1)$-th and the l-th new white balls. Geom(a) is the probability mass function $a(1-a)^{k-1}$ for $k \ge 1$.)

Let $t_{c,j}'$ denote the number of balls sampled after the arrival of the $(j-1)$-th new white ball up to the arrival of the j-th new white ball, for $1 \le j \le R_c'$. Then $t_{c,j}' \sim \text{Geom}(P_E)$ where $P_E := N_c p(1-p)^{k_1c_2\log n-1}$. Note that $t_{c,j}'$ are mutually independent across $1 \le j \le R_c'$ and further, they are also mutually independent of R_c'. We thus have,

$$T_c' = \sum_{j=1}^{R_c'} t_{c,j}' \tag{12}$$

Now compare the following two events: (1) Event $\mathcal{A}$ defined as the successful transmission from Cell c resulting from a successful transmission by Node i in Cell c and (2) Event $\mathcal{B}$ defined as the sampling of a white ball in any particular sample. Observe that $\Pr(\mathcal{A}) \ge \Pr(\mathcal{B})$. From this comparison, we see that $\mathcal{T}_c$ will be stochastically dominated by T_c' i.e. $\Pr(\mathcal{T}_c \ge z) \le \Pr(T_c' \ge z) \ \forall z \in \mathbb{N}$. Further, T_c' will be stochastically dominated by the random variable $T_c = \sum_{j=1}^{R_c} t_{c,j}$, where $t_{c,j} \sim \text{Geom}(p_S)$ and $R_c \sim \sum_{l=1}^{m} \text{Geom}(1 - \frac{l-1}{m})$ with

$m = \lceil c_2 \log n \rceil$ which is an upper bound on N_c from (1). We therefore, have

$$\mathsf{Pr}(\mathcal{T}_c \geq z) \leq \mathsf{Pr}(T_c \geq z) \;\; \forall z \in \mathbb{N} \tag{13}$$

It is convenient to work with the random variable T_c because it is independent of the parameters of Cell c. We will obtain the moment generating functions (mgf) of the distributions of the integer-valued random variables involved. Let the mgf of each random variable be denoted by the same character in sans serif font. For a random variable F, $\mathsf{F}(z) = \sum_{j \in \mathbb{Z}} \mathsf{Pr}(F = j)\, z^{-j}$. The region of convergence (RoC) of the mgf is also specified.

$$\begin{aligned}
\mathsf{t_{c,j}}(z) &= \frac{p_S z^{-1}}{1-(1-p_S)z^{-1}} := S(z) \\
&\qquad \text{with RoC } (|z| > 1 - p_S) \\
\mathsf{R_c}(z) &= \Pi_{l=1}^{m} \frac{(1-\frac{l-1}{m})z^{-1}}{1-\frac{l-1}{m}z^{-1}} \\
&\qquad \text{with RoC } \left(|z| > 1 - \frac{1}{m}\right) \\
\mathsf{T_c}(z) &= \sum_{r \in \mathbb{N}} \mathsf{Pr}(R_c = r)\,[S(z)]^r \; = \; \mathsf{R_c}\left(\frac{1}{S(z)}\right) \\
&= \frac{m!\,p_S^m}{\Pi_{l=1}^{m}\left(m[z-(1-p_S)] - (l-1)p_S\right)} \\
&\qquad \text{with RoC } \left(|z| > 1 - \frac{p_S}{m}\right)
\end{aligned}$$

Thus,

$$\mathbb{E}[e^{sT_c}] = \frac{m!\,p_S^m}{\Pi_{l=1}^{m}\left(m[e^{-s}-(1-p_S)] - (l-1)p_S\right)}$$

for $s < \log\left(\frac{1}{1-\frac{p_S}{m}}\right)$. Choose $s_1 = \log\left(\frac{1}{1-\frac{p_S}{2m}}\right)$. After some algebra, we can show the following.

$$\begin{aligned}
\mathbb{E}[e^{s_1 T_c}] &= \frac{m!\,p_S^m}{m^m}\Pi_{l=1}^{m}\left(e^{-s_1} - 1 + \frac{m-l+1}{m}p_S\right)^{-1} \\
&= c_m\sqrt{\pi m}.
\end{aligned}$$

Here $c_m = \frac{2^{2m}}{\binom{2m}{m}\sqrt{\pi m}} \to 1$ as $m \to \infty$ by the Stirling approximation. From the Chernoff bound we get $\mathsf{Pr}(\mathcal{T}_c \geq V_1) \leq \mathsf{Pr}(T_c \geq V_1) \leq c_m\sqrt{\pi m}\left(1-\frac{p_S}{2m}\right)^{V_1}$. From the union bound, we have

$$\mathsf{Pr}\left(\max_{c \in \mathcal{C}} \mathcal{T}_c \geq V_1\right) \leq M_n c_m \sqrt{\pi m}\left(1 - \frac{p_S}{2m}\right)^{V_1} \tag{14}$$

To achieve $\mathsf{Pr}(\max_{c\in\mathcal{C}} \mathcal{T}_c \geq V_1) \leq \frac{k}{n^\alpha}$, it is sufficient to have $(1-\frac{p_S}{2m})^{V_1} \leq \frac{k}{n^\alpha M_n c_m \sqrt{\pi m}}$ or

$$V_1 \geq \frac{\frac{1}{2}\log m + \log M_n + \alpha \log n - \log k + \frac{1}{2}\log \pi + \log c_m}{-\log(1-\frac{p_S}{2m})} \tag{15}$$

Here, $m = \lceil c_2 \log n \rceil$, $M_n = \lceil \sqrt{\frac{n}{2.75 \log n}} \rceil^2$. Writing

$$-\log\left(1 - \frac{p_S}{2m}\right) = \frac{p_S}{2m} + \frac{p_S^2}{2(2m)^2} + \ldots,$$

we have $-\log\left(1-\frac{p_S}{2m}\right) \geq \frac{p_S}{2m}$ we can see that there exists a choice of $V_1 = O((\log n)^2)$ which would be sufficient for the completion of Phase I, i.e., every node in every cell of the network would have successfully transmitted at least once in V_1 slots, with probability at least $\left(1 - \frac{k}{n^\alpha}\right)$.

2) Phase II: Progress to the bottom of the square: Let the columns of cells shown in Fig. 2 be numbered $C_1, C_2, \ldots, C_{l_n}$. Let the l_n cells in each column be numbered from 1 to l_n from top to bottom. In this phase, we are concerned with transmissions in the top $w := l_n - 1$ cells of each column. In Phase I, each node has successfully received the transmissions by every other node in its cell. Hence, Phase II will be completed if the following sequence of events occurs for each column C: A successful transmission by some node in the first cell of the column, followed by a successful transmission by some node in the second cell of the column and so on until a successful transmission by some node in the w-th cell of the column.

Let the number of slots required for this sequence of events be $\mathcal{T}^{(C)}$ for column C. We can see that $\mathcal{T}^{(C)}$ will be stochastically dominated by $T^{(C)} := \sum_{j=1}^{w} t_j^{(C)}$, where $t_j^{(C)} \sim \text{Geom}(p_S)$. We can thus derive the following.

$$\begin{aligned}
\mathsf{T^{(C)}}(z) &= \frac{p_S^w z^{-w}}{(1-(1-p_S)z^{-1})^w} \\
&\qquad \text{for } (|z| > 1 - p_S) \\
\mathbb{E}[e^{sT^{(C)}}] &= \frac{p_S^w}{(e^{-s}-(1-p_S))^w} \\
&\qquad \text{for } s < \log\left(\frac{1}{1-p_S}\right) \\
\mathsf{Pr}\left(T^{(C)} \geq V_2\right) &\leq \frac{\mathbb{E}[e^{s_2 T^{(C)}}]}{e^{s_2 V_2}} = 2^w\left(1 - \frac{p_S}{2}\right)^{V_2} \\
\mathsf{Pr}\left(\max_{1 \leq j \leq l_n} \mathcal{T}^{(C_j)} \geq V_2\right) &\leq l_n 2^w \left(1 - \frac{p_S}{2}\right)^{V_2}
\end{aligned}$$

where we have used $s_2 = \log(\frac{1}{1-\frac{p_S}{2}})$ in the Chernoff bound. Thus, to achieve $\mathsf{Pr}\left(\max_{1\leq j \leq l_n} T^{(C_j)} \geq V_2\right) \leq \frac{k}{n^\alpha}$, it suffices to have $(1-\frac{p_S}{2})^{V_2} \leq \frac{k}{n^\alpha l_n 2^w}$ or

$$V_2 \geq \frac{\alpha \log n + \log l_n + w \log 2 - \log k}{-\log(1-\frac{p_S}{2})}$$

Now, $l_n = \lceil \sqrt{n/(2.75 \log n)} \rceil = w + 1$, and hence, $V_2 = O\left(\sqrt{n/\log n}\right)$ slots are sufficient for the completion of Phase II with probability at least $(1 - k/n^\alpha)$.

3) Phase III: Progress into the sink: Phase III comprises of diffusion of the MAX into the cell containing the sink. Let the time required for this to happen be the random variable T_s. It is easily seen from the analysis of the sequence of transmission for Phase II that $\mathsf{Pr}(T_s \geq V_3) \leq 2^w(1-\frac{p_S}{2})^{V_3}$ where w is as defined before. Calculations similar to those in the analysis for Phase II show that $V_3 = O\left(\sqrt{n/\log n}\right)$ slots are sufficient for completion of this phase with probability at least $(1 - k/n^\alpha)$.

4) Bound on the overall time: Since each of phases I, II and III get completed in $O\left(\sqrt{n/(\log n)}\right)$ time slots with probability at least $\left(1 - k'/n^\alpha\right)$, for appropriate constants k', the protocol One-Shot MAX achieves computation of the MAX at the sink in $O\left(\sqrt{n/(\log n)}\right)$ slots with probability at least $(1 - k/n^\alpha)$.

Algorithm 3: Hop Distance Compute: There are $d+1$ Frames Numbered $0, 1, \ldots, d$. The Sink Node Transmits 0 in Each Slot of Frame of 0. Node i Executes the Following Protocol in Each Slot of Each Frame

```
for frame=1 to d + 1 do
    for slot=1 to τ do
        if successful reception then
            n = received number
            h_i = n + 1
        end
        if frame=h_i then
            Transmit h_i with probability p
        end
    end
end
```

C. Obtaining the Hop Distance

The algorithm Hop Distance Compute described below obtains the hop distance for each node in the network. τ slots form a frame and $\lceil \log d \rceil$ bits are transmitted in each slot. $\tau = V_1 = \Theta((\log n)^2)$ and is obtained as in the analysis of Phase I of protocol One-Shot MAX. The protocol completes in $(d+1)$ frames.

Let the frames be numbered $0, 1, \ldots, d$. A node either transmits with probability p in every slot of a frame or it does not transmit in any slot of the frame. Each transmission is a $\lceil \log d \rceil$ bit number. At the beginning of the algorithm, the sink transmits 0 in each slot of frame 0. Each node of the network other than the sink executes the following algorithm. Node i makes no transmission till it has decoded a transmission successfully. Let the first successful reception by Node i happen in a slot from frame g_i and let the decoded transmission be the number n_i. Node i sets its hop distance $h_i = (n_i + 1)$ and ignores other successful receptions in slots from frame g_i. During the τ slots from frame $(g_i + 1)$, Node i transmits h_i (expressed in $\lceil \log d \rceil$ bits) with probability p, independently of all the other transmissions; the node does not transmit with probability $(1-p)$. Node i does not transmit in frames $(g_i + 2), \ldots, d+1$. Node i makes no more transmissions. This is summarized in Algorithm 3. The protocol completes after $(d+1)\tau\lceil \log d \rceil$ bit-periods.

Lemma 2: The nodes of the network correctly compute their minimum hop distance from the sink, using Hop Distance Compute in $O(\sqrt{n(\log n)^5})$ bit periods with probability at least $(1 - k/n^{\alpha})$ for any positive α and some constant k.

Proof: The algorithm Hop Distance Compute does the following. Each node, after computing its true hop distance in some frame transmits it in the next frame. These transmissions enable those nodes with hop distance one greater than itself to compute their true hop distance.

Consider any frame g. For a suitable $\tau = c(\log n)^2$, by the proof of the bound on phase I of One-Shot MAX, the event of at least one successful transmission by every node that is attempting to transmit in frame g, will occur with probability at least $(1 - k/n^{\alpha})$. Therefore, the probability that some node does not transmit successfully in any slot is upper bounded by $(k/n^{\alpha})d$, where $d = 2\lceil \sqrt{n/(2.75 \log n)} \rceil$. Since α can be made as large as necessary, this upper bound on the probability can be made arbitrarily small (by choosing $\alpha > 1/2$). The total number of slots used is at most $d\tau$, and hence the total number of bits transmitted is at most $d\tau\lceil \log d \rceil$, which is $O(\sqrt{n(\log n)^5})$ bit-periods. ■

D. Proof of Theorem 2

We just need to obtain the probability that the result obtained after a specified number of slots is correct. Let the set of nodes at hop distance h be G_h. Let $t_{i,r}$ be the first slot in round r that Node i transmits successfully in. The number of slots in a round is $\tau = \Theta((\log n)^2)$ ($\tau = V_1$ from Phase I). Every node in the network would have transmitted successfully at least once in each round of τ slots w.h.p. Let $h_{\max} \leq d$ be the largest hop distance of a node in the network. In the proof, we will assume that each node of the network transmits successfully in each round at least once. We claim that

$$\max_{i \in G_h} T_i(r, t_{i,r}) = \max_{j \in \bigcup_{h \leq f \leq d} G_f} Z_j(r - d + h) \tag{16}$$

for $0 \leq h \leq h_{\max}$ and $r > d - h$. Since the sink is at hop distance 0, proving the preceding claim will complete the proof. Assume that the claim is true for $h_0 < h \leq h_{\max}$, $r > d - h$. We shall show that the claim will then be true for $h = h_0$ and for $r > d - h_0$. Consider transmissions by the nodes at hop distance h_0 in round $(r+1)$.

$$\max_{i \in G_{h_0}} T_i(r+1, t_{i,r+1}) = \max_{i \in G_{h_0}} \{\max\{Z_i(r+1-d+h_0), Y_i(r)\}\}$$

Since each node at hop distance $(h_0 + 1)$ transmits successfully at least once in round r, the transmission of each such node is decoded successfully by some node at hop distance h_0. Hence,

$$\begin{aligned} \max_{i \in G_{h_0}} Y_i(r) &= \max_{j \in G_{h_0+1}} T_j(r, t_{j,r}) \\ &= \max_{j \in \bigcup_{h_0+1 \leq f \leq d} G_f} Z_j(r - d + h_0 + 1) \end{aligned}$$

where the second equality follows from the induction hypothesis. Hence,

$$\begin{aligned} \max_{i \in G_{h_0}} T_i(r+1, t_{i,r+1}) &= \max\{\max_{i \in G_{h_0}} Z_i(r+1-d+h_0), \\ &\quad \max_{j \in \bigcup_{h_0+1 \leq f \leq d} G_f} Z_j(r - d + h_0 + 1)\} \\ &= \max_{j \in \bigcup_{h_0 \leq f \leq d} G_f} Z_j(r - d + h_0 + 1)\} \end{aligned}$$

which proves the claim for hop distance h_0 for round $(r+1)$. By induction, the claim is true for each h and each round $r > d - h$. Therefore, the sink node s correctly sets $\mathcal{Z}(r-d) = \max\{Z_s(r-d), Y_s(r)\}$. The delay of the protocol is $d\tau = \Theta(\sqrt{n(\log n)^3})$ bit-periods.

As transmissions by different nodes are independent, the analysis in the diffusion of phase I of One-Shot MAX carries over. The probability that the computed value of $\mathcal{Z}(r)$ is incorrect for any given round is upper bounded by $\frac{k}{n^{\alpha}}$ for any constants $\alpha, k > 0$.

E. Proof of Lemma 1

Let $R_1, R_2, \ldots, R_N$ be i.i.d. exponential random variables each exponentially with unit mean. Let $R = \max_{1\leq i\leq N} R_i$.

$$\begin{aligned}\Pr(R \leq m) &= \left(1-e^{-m}\right)^N\\ &\geq 1-\frac{N}{e^m}\\ \Pr(R > m) &\leq N\ e^{-m}\end{aligned}$$

The second inequality is obtained from $(1-x)^N \geq 1 - Nx$ for $x \in [0, 1]$. Set $m = (r + 1)\log N$ to get the lemma.

F. Proof of Theorem 3

Let $S_0 = \{i : Z_i = 0\}, |S_0| = n_0$ and $S_1 = \{i : Z_i = 1\}, |S_1| = n_1$. Let the M_0 (resp. M_1) denote the maximum of the random variables generated by all nodes with data bit 0 (resp. 1).

Let us have $I_n = n^3$ iterations of generating exponential random variables and computing their MAX. The total number of exponential random variables generated in the entire algorithm is $N := nI_n$. We will use this value of N and $r = 1$ in Lemma 1 to get the probability of any exponential random variable in our algorithm falling outside our range $[0, (r+1)\log N] = [0, 8\log n]$ to be at most $\frac{1}{N} = \frac{1}{n^4}$. Let us declare that the algorithm fails if any of the exponential random variables generated in the entire algorithm exceeds $8\log N$. Our calculations guarantees that, with high probability, this kind of an error does not occur and in the subsequent calculations we will assume that this error does not occur.

Let the MAX of the exponential random variables generated by nodes in S_0 and S_1 in the j-th iteration be $\mathcal{E}_0^j$ and $\mathcal{E}_1^j$ respectively and their quantized versions be $\overline{\mathcal{E}}_0^j$ and $\overline{\mathcal{E}}_1^j$ respectively. For $b = 0, 1$, observe that $|\overline{\mathcal{E}}_b^j - \mathcal{E}_b^j| \leq \frac{q}{2}$ and hence,

$$\mathcal{E}_b^j - \frac{q}{2} \leq \overline{\mathcal{E}}_b^j \leq \mathcal{E}_b^j + \frac{q}{2}. \tag{17}$$

It is easy to show that

$$\mathrm{Var}(\mathcal{E}_b^j) = \sum_{j=1}^{n_b} \frac{1}{j^2}. \tag{18}$$

By the Chebyshev inequality,

$$\Pr\left(\left|\frac{\sum_{j=1}^{I_n}\mathcal{E}_b^j}{I_n} - \mathbb{E}[M_b]\right| > \epsilon\right) \leq \frac{\sum_{j=1}^{n_b}\frac{1}{j^2}}{\epsilon^2 I_n} \leq \frac{\pi^2}{6\epsilon^2 I_n}. \tag{19}$$

Since we are quantizing in steps of q, we have

$$\frac{\sum_{j=1}^{I_n}\mathcal{E}_b^j}{I_n} - \frac{q}{2} \leq \frac{\sum_{j=1}^{I_n}\overline{\mathcal{E}}_b^j}{I_n} \leq \frac{\sum_{j=1}^{I_n}\mathcal{E}_b^j}{I_n} + \frac{q}{2}. \tag{20}$$

Choose $\epsilon = \frac{1}{n}$ and as mentioned before, $I_n = n^3$. Let $\hat{M}_0$ and $\hat{M}_1$ be the estimates of $\mathbb{E}[M_0]$ and $\mathbb{E}[M_1]$ respectively obtained as $\hat{M}_b = \frac{\sum_{j=1}^{I_n}\overline{\mathcal{E}}_j^b}{I_n}$, for $b = 0, 1$. Then, $\hat{M}_b$ is a good estimate of $\mathbb{E}M_b$ as given by

$$\Pr\left(|\hat{M}_b - \mathbb{E}[M_b]| > \frac{q}{2} + \frac{1}{n}\right) \leq \frac{\pi^2}{6n}. \tag{21}$$

Let the quantization step be chosen as $q = \frac{1}{n}$. Then,

$$\Pr\left(|\hat{M}_b - \mathbb{E}[M_b]| > \frac{3}{2n}\right) \leq \frac{\pi^2}{6n}. \tag{22}$$

1) Error in Estimate: Assume that the MAX is computed successfully in each round. From the union bound and Theorem 1, this happens with probability at least $(1-(k\times I_n \times \frac{8\log n}{q})/n^\alpha) = (1-(k\times I_n \times 8n\log n)/n^\alpha) = (1-\frac{8k\log n}{n^{\alpha-4}})$. n_0 and n_1 are estimated by $\hat{n}_b = e^{\hat{M}_b-\gamma}$. The error in this estimate is then given by the following bound:

$$\Pr\left(e^{-\frac{3}{2n}}e^{\xi_{n_b}} \leq \frac{\hat{n}_b}{n_b} \leq e^{\frac{3}{2n}}e^{\xi_{n_b}}\right) \geq 1-\frac{1}{n^4}-\frac{8k\log n}{n^{\alpha-4}}-\frac{\pi^2}{6n}. \tag{23}$$

The three errors above corresponding respectively to:

- The MAX of any of the exponential random variables generated in the algorithm exceeds $8\log n$.
- The MAX of the quantized exponential random variables is incorrectly computed in at least one round.
- The average of the MAX of the quantized exponential random variables across the $I_n = n^3$ iterations exceeds the mean of the MAX of the exponential random variables by more than $\frac{q}{2} + \frac{1}{n} = \frac{3}{2n}$.

By choosing $\alpha = 6$, we have

$$\begin{aligned}&\Pr\left(\frac{n_b}{n}e^{-\frac{3}{2n}}e^{\xi_{n_b}} \leq \frac{\hat{n}_b}{n} \leq \frac{n_b}{n}e^{\frac{3}{2n}}e^{\xi_{n_b}}\right)\\ &= \Pr\left(e^{-\frac{3}{2n}}e^{\xi_{n_b}} \leq \frac{\hat{n}_b}{n_b} \leq e^{\frac{3}{2n}}e^{\xi_{n_b}}\right) \geq 1 - O\left(\frac{1}{n}\right).\end{aligned} \tag{24}$$

2) Total Time for Computation: The total time required for one-shot histogram computation using the values as chosen would be

$$\begin{aligned}&O(n^3\sqrt{n/\log n}\log(8\log n/q))\\ &= O(n^3\sqrt{n/\log n}\log(8n\log n))\\ &= O(n^{7/2}(\log n)^{1/2})\end{aligned}$$

bit-periods.

REFERENCES

[1] B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter, “Deterministic broadcasting in unknown radio networks,” *Distrib. Comput.*, vol. 15, no. 1, pp. 27–38, 2002.

[2] M. Chrobak, L. Gasieniec, and W. Rytter, “Fast broadcasting and gossiping in radio networks,” in *Proc. 41st Symp. FOCS*, Nov. 2000, pp. 575–581.

[3] A. E. F. Clementi, A. Monti, and R. Silvestri, “Distributed broadcast in radio networks of unknown topology,” *Theoretical Comput. Sci.*, vol. 302, nos. 1–3, pp. 337–364, Jun. 2003.

[4] A. Czumaj and W. Rytter, “Broadcasting algorithms in radio networks with unknown topology,” in *Proc. 44th Symp. FOCS*, Oct. 2003, pp. 492–501.

[5] C. Dutta, Y. Kanoria, D. Manjunath, and J. Radhakrishnan, “A tight lower bound for parity in noisy communication networks,” in *Proc. 9th Annu. ACM-SIAM SODA*, San Francisco, CA, USA, Jan. 2008, pp. 1056–1065.

[6] U. Feige and J. Kilian, “Finding OR in noisy broadcast network,” *Inf. Process. Lett.*, vol. 73, nos. 1–2, pp. 69–75, Jan. 2000.

[7] R. G. Gallager, “Finding parity in simple broadcast networks,” *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 176–180, Mar. 1988.

[8] 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.

[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] N. Goyal, G. Kindler, and M. E. Saks, “Lower bounds for the noisy broadcast problem,” *SIAM J. Comput.*, vol. 37, no. 6, pp. 1806–1841, 2008.

[11] P. Gupta and P. R. Kumar, “Critical power for asymptotic connectivity in wireless networks,” in *Proc. 37th IEEE Conf. Decision Control*, vol. 1. Tampa, FL, USA, Dec. 1998, pp. 1106–1110.

[12] P. Gupta and P. R. Kumar, “Capacity of wireless networks,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[13] Y. Kanoria and D. Manjunath, “On distributed computation in noisy random planar networks,” in *Proc. IEEE ISIT*, Nice, France, Jun. 2007 pp. 1–5.

[14] N. Khude, A. Kumar, and A. Karnik, “Time and energy complexity of distributed computation in wireless sensor networks,” in *Proc. IEEE 24th Annu. Joint Conf. INFOCOM*, vol. 4. Miami, FL, USA, Nov. 2005, pp. 2625–2637.

[15] E. Kushilevitz and Y. Mansour, “An $\omega(d \log(n/d))$ lower bound for broadcast in radio networks,” *SIAM J. Comput.*, vol. 27, no. 3, pp. 702–712, 1998.

[16] I. Newman, “Computing in fault tolerance broadcast networks,” in *Proc. 19th IEEE Annu. Conf. Comput. Complex.*, Jun. 2004, pp. 113–122.

[17] S. Subramaniam, P. Gupta, and S. Shakkottai, “Scaling bounds for function computation over large networks,” in *Proc. IEEE ISIT*, Nice, France, Jun. 2007, pp. 136–140.

[18] F. Xue and P. Kumar, “The number of neighbors needed for connectivity of wireless networks,” *Wireless Netw.*, vol. 10, no. 2, pp. 169–181, Mar. 2004.

[19] L. Ying, R. Srikant, and G. Dullerud, “Distributed symmetric function computation in noisy wireless sensor networks,” *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4826–4833, Dec. 2007.

Sudeep Kamath received the Bachelor of Technology degree in Electrical Engineering from the Indian Institute of Technology Bombay in 2008. He received the M.S. degree in Electrical Engineering and Computer Science in 2011 and the M.A. degree in Statistics, and Ph.D. degree in Electrical Engineering and Computer Science in 2013, all from the University of California, Berkeley.

D. Manjunath received his BE from Mysore University, MS from IIT Madras and PhD from Rensselaer Polytechnic Inst, Troy NY in 1986, 1989 and 1993 respectively. He has worked in the Corporate Research and Development Center of GE in Scehenectady NY (1990), CIS Deptt. of the University of Delaware (1992-93) and the Computer Science Deptt. University of Toronto (1993- 84). He was on the Electrical Engineering faculty of the IIT Kanpur during 1994-98. He has been with the Electrical Engineering Dept of IIT-Bombay since July 1998 where he is now a Professor. He is also the head of the Computer Centre at IIT Bombay. He is a coauthor of two Morgan-Kaufman textbooks *Communication Networking: An Analytical Approach*, published in May 2004, and *Wireless Networking*, published in April 2008. He is also the recipient of the Best Paper award at ACM SIGMETRICS 2010. His research interests include communication networks, performance analysis and queueing systems. His recent research has concentrated on pricing and queue control, analysis of random wireless data and sensor networks and network performance measurement.

Ravi Mazumdar (F'05) was born in April 1955 in Bangalore, India. He obtained the B.Tech. in Electrical Engineering from the Indian Institute of Technology, Bombay, India in 1977, the M.Sc. DIC in Control Systems from Imperial College, London, U.K. in 1978 and the Ph.D. in Systems Science from the University of California, Los Angeles, USA in 1983.

He is a University Research Chair Professor of Electrical and Computer Engineering at the University of Waterloo, Waterloo, Canada and an Adjunct Professor of ECE at Purdue University. He has served on the faculties of Columbia University (NY, USA) and INRS-Telecommunications (Montreal, Canada) . He held a Chair in Operational Research and Stochastic Systems in the Dept. of Mathematics at the University of Essex (Colchester, U.K.), and from 1999-2005 was Professor of ECE at Purdue University (West Lafayette, USA). He has held visiting positions and sabbatical leaves at UCLA, the University of Twente (Netherlands), the Indian Institute of Science (Bangalore); Ecole Nationale Superieure des Telecommunications (Paris), INRIA, and UC Berkeley.

He is a Fellow of the IEEE and the Royal Statistical Society. He is a member of the working groups WG6.3 and 7.1 of the IFIP and a member of SIAM and the IMS. He is a recipient of the IEEE INFOCOM 2006 Best Paper Award and was runner-up for the Best Paper at INFOCOM 1998. His research interests are in applied probability, stochastic analysis, optimization, and game theory with applications to wireless and wireline networks, traffic engineering, filtering theory, and risk theory.