Protocols and Impossibility Results for Gossip-Based Communication Mechanisms

David Kempe[*]
Department of Computer Science
Cornell University, Ithaca NY 14853
Email: kempe@cs.cornell.edu

Jon Kleinberg[†]
Department of Computer Science
Cornell University, Ithaca NY 14853
Email: kleinber@cs.cornell.edu

Abstract

In recent years, gossip-based algorithms *have gained prominence as a methodology for designing robust and scalable communication schemes in large distributed systems. The premise underlying distributed gossip is very simple: in each time step, each node v in the system selects some other node w as a communication partner — generally by a simple randomized rule — and exchanges information with w; over a period of time, information spreads through the system in an "epidemic fashion".*

A fundamental issue which is not well understood is the following: how does the underlying low-level gossip mechanism *— the means by which communication partners are chosen — affect one's ability to design efficient high-level gossip-based protocols? We establish one of the first concrete results addressing this question, by showing a fundamental limitation on the power of the commonly used* uniform gossip *mechanism for solving nearest-resource location problems. In contrast, very efficient protocols for this problem can be designed using a non-uniform* spatial gossip *mechanism, as established in earlier work with Alan Demers.*

We go on to consider the design of protocols for more complex problems, providing an efficient distributed gossip-based protocol for a set of nodes in Euclidean space to construct an approximate minimum spanning tree. Here too, we establish a contrasting limitation on the power of uniform gossip for solving this problem. Finally, we investigate gossip-based packet routing as a primitive that underpins the communication patterns in many protocols, and as a way to understand the capabilities of different gossip mechanisms at a general level.

1. Introduction

Gossip-based communication. In recent years, *gossip-based algorithms* have gained prominence as a methodology for designing robust and scalable communication schemes in large distributed systems. The premise underlying distributed gossip is very simple: in each time step, each node v in the system selects some other node w as a communication partner — generally by a simple randomized rule — and exchanges information with w; over a period of time, information spreads through the system in an "epidemic fashion". Perhaps the most basic example of a gossip mechanism, and one of the most widely studied, is *uniform gossip*: in each step, each node v chooses some other node w, independently and uniformly at random, and sends a message to w. (For obvious reasons, this is sometimes called the "random phone call" or "rumor spreading" mechanism.)

Gossip-based approaches thus stand in contrast to more rigid communication structures in which nodes communicate according to a carefully synchronized protocol, and also in contrast to centralized schemes, in which a "leader node" is responsible for disseminating information. Gossip-based algorithms have the advantage that they are very easy to implement, with each node following a simple local rule in each time step; and they are highly fault-tolerant, since communication will happen in aggregate despite a fairly high level of message loss and node failures.

The pioneering work of Demers et al. [5] demonstrated the power of gossip-based algorithms in the context of a distributed database system; subsequently, the approach has been applied to tasks including failure detection [24], resource discovery [23, 10], data aggregation [8], and, following the initial work in [5], further problems in replicated database management [1]. A number of papers have also studied the dynamics of gossip from a combinatorial point of view (see e.g. [6, 7, 11, 12, 21]). In many of these settings, there is an underlying network that supports the abstraction of point-to-point communication — any node can

[*] Supported by an Intel Fellowship.

[†] Supported in part by a David and Lucile Packard Foundation Fellowship, an ONR Young Investigator Award, NSF ITR/IM Grant IIS-0081334, and NSF Faculty Early Career Development Award CCR-9701399.

contact any other node in a given step — but the nodes are also embedded in an underlying metric space, and the goals of the application are related to proximity in this space. For example, nodes may wish to find the *nearest* resource or mirror site in a network, or be alerted if there is a *nearby* event or process failure; here, "nearness" is with respect to the metric space containing the points, while gossip — thanks to the abstraction of point-to-point communication — may proceed on a network that is conceptually the complete graph.

In recent work with Alan Demers [13], we considered such a framework, in which gossiping nodes are positioned roughly uniformly in $\mathbb{R}^D$; essentially, they are placed so that each unit ball contains $\Theta(1)$ nodes. Such a setting suggests two very natural gossip mechanisms. Using uniform gossip, it can be shown [11, 21] that if nodes forward all information they receive, a piece of information originating with a given node will spread through the entire system in $O(\log n)$ steps. At the other extreme is *neighbor flooding* [17] in which each node communicates with its k nearest neighbors in a round-robin fashion. In [13], we analyzed a "hybrid" mechanism called *spatial gossip*, combining the locality properties of flooding with the exponentially fast dissemination of uniform gossip. For a parameter ρ, node v chooses to send a message to node w with probability proportional to $d_{v,w}^{-\rho D}$, where $d_{v,w}$ denotes the distance between v and w. For exponents ρ between 1 and 2, we proved that the time for a piece of information originating with a node v to reach nodes at distance d is bounded by $O(\log^{1+\epsilon} d)$ for some small ϵ, and independent of the total number of nodes.

Gossip mechanisms and protocols. Given the increasing use of gossip-based approaches, many distributed computational tasks are being solved by building application-specific protocols on top of a simple gossip-based communication mechanism. For this reason, [13] argued in favor of distinguishing between two conceptual layers in gossip-based algorithms: A *gossip mechanism*, which simply defines the communication connections that are made during the execution, and a *gossip protocol*, which defines the semantics of the messages that are exchanged over these connections. For example, uniform gossip and spatial gossip are *mechanisms* — they specify, through randomization, who will talk to whom in each time step, but they do not specify what the contents of the messages should be. We say that a protocol for a higher-level task (e.g. resource discovery, database replication) is *based on* such a mechanism if this sequence of communications — who talks to whom — is determined by the mechanism.

A fundamental question which is very little understood is the following: How does the choice of gossip mechanism affect one's ability to design efficient protocols? For example, are there natural problems for which it is provably impossible to design an efficient protocol based on the uniform gossip mechanism? Can we tell when there exists an efficient protocol based on one mechanism but not based on another?

The *resource location* problem provides a clean setting in which to pose such questions more concretely. In the most basic version of the resource location problem, a subset X of the nodes hold copies of some desirable resource, and we want each other node to rapidly learn the identity of an (approximately) nearest resource-holder. We will say that a protocol is an *efficient* c-approximation for this problem if each individual communication connection contains at most a poly-logarithmic number of messages, and after a poly-logarithmic number of steps, each node knows of a resource-holder whose distance is within a factor c of closest, with high probability. Moreover, we will be concerned in this paper with protocols that are *atomic*; informally, this means that after a node first injects a given message into the system, other nodes can only store, copy, and forward, but not otherwise modify it. (We define atomicity precisely in Section 2.) While this restricts the class of protocols under consideration, it includes most natural gossip-based approaches and is motivated in part from a security-based standpoint, in which we want to preserve integrity of messages. In particular, if node v wants to consider node r its nearest resource-holder, it should have received a copy of a message originating with r.

In [13], we designed an efficient, atomic $(1 + o(1))$-approximation for this problem, built on top of a spatial gossip mechanism with exponent ρ between 1 and 2. Given the analysis of spatial gossip, the use of such ρ values seemed highly appropriate for the resource location problem, since the names of resource-holders should diffuse rapidly through the set of nodes near them. At the same time, we had no proof that the particular spatial gossip mechanism was in any sense "necessary" for designing such a protocol.

The present work. We begin by proving that, in fact, there is no efficient, atomic poly-logarithmic approximation for resource location based on the uniform gossip mechanism. Moreover, the result extends to show that spatial gossip with any exponent $\rho \notin [1, 2]$ cannot provide an efficient, atomic poly-logarithmic approximation for resource location.

An important point to realize about this impossibility result is the following. If we run a poly-logarithmic number of steps of uniform gossip, and then look at the resulting pattern of communication, it appears possible for an algorithm *in retrospect* to find a way of informing most nodes of their closest resource. (We conjecture this to be the case, although we have not proved it.) The obstacle to designing a protocol seems to be a computational one: for small exponents, the pattern of communications is "too random" for nodes to have any guidelines for choosing which messages to forward. While the problem here is clearly related in spirit to the area of *communication complexity* (e.g. [16, 18]), the technical framework appears to be different — in particular, due to the atomicity condition.

We next consider a more complex problem, the construction of a minimum spanning tree using spatial gossip. Specifically, among a large set of nodes roughly uniformly positioned in $\mathbb{R}^D$, suppose there is a subset X of nodes that want to organize themselves into a tree. (The members of the subset need not know each others' identities initially.) We show how spatial gossip with $\rho \in [1, 2)$ can be used to design an atomic protocol which, within a poly-logarithmic number of steps, constructs a tree on the nodes of X of total edge length within $O(\log |X|)$ of the minimum spanning tree cost. Again, we can show that no such atomic protocol exists based on the uniform gossip mechanism, or any spatial gossip mechanism with $\rho \notin [1, 2]$.

Finally, we consider a primitive underlying a number of gossip-based applications: the routing of packets from a set of source nodes to corresponding destination nodes. One can view this as an abstract communication problem through which to analyze the power of spatial gossip. We assume that we are given a set M of *messages* (also called *packets*) μ, each of which has associated with it a source s_μ and destination d_μ, such that each node v is the source and destination of $O(1)$ messages (our results actually generalize to arbitrary bounds instead of $O(1)$). The goal of a routing protocol is to use the communication connections provided by the underlying gossip mechanism to forward all of the messages to their destinations. In the process, arbitrarily many copies of the messages may be made and stored at nodes. If nodes are allowed to forward all packets they have ever received in a single step, this problem is easy; we are concerned with the case in which only β messages may be forwarded by any one node in a given step, for some upper bound β. We obtain the following trichotomy for spatial gossip with exponent ρ.

1. For $\rho < 1$, for any routing assignment M, any protocol (deterministic or randomized) will have routed at most $O(\beta n^\rho t^2)$ of the messages by time t in expectation.

2. For $\rho = 1$, there is a simple protocol routing all of the messages in time $O(\log^3 n)$ with high probability, forwarding at most one message in each time step.

3. For $\rho > 1$, there are routing assignments M such that any protocol (deterministic or randomized) will have routed at most $O(\beta n^{2-\rho} \cdot t)$ of the messages by time t.

There are two things that should be noted about this trichotomy. First, the impossibility in part (1) applies to *any* permutation, whereas the impossibility result in part (3) is significantly weaker, asserting that there *exist* bad permutations for any protocol. This is clearly critical, since for the communication patterns that arise in resource location and MST construction, we know that poly-logarithmic running time is possible using any exponent $\rho \in [1, 2)$; the "locality" in the problem leads to tractable communication patterns.

The second point is that the obstacles to delivering all messages in poly-logarithmic time are very different in nature for exponents $\rho < 1$ and for $\rho > 1$. The former case is akin to (and at the heart of) the impossibility result for resource location when $\rho < 1$: although there may be a feasible routing of messages in retrospect, there is no way to find it as the protocol proceeds. On the other hand, when $\rho > 1$, the problem is simply one of bandwidth — there are not enough long-range connections made to even allow forwarding all messages if they all have to travel a long distance in the underlying metric. Hence, even knowledge of future random choices does not help in this case.

The three-part distinction in routing ability is similar to a corresponding trichotomy in *small-world networks* [15]. The framework there is different; in [15], a set of underlying nodes is explicitly augmented with "long-range" links; here, we have a model supporting point-to-point communication, and the connections are specified in a step-by-step fashion using the underlying gossip mechanism. Moreover, we are seeking something much stronger here — rather than searching for a single path, we wish to route an entire permutation. Our routing protocol for exponent $\rho = 1$ also has a qualitative similarity to algorithms for permutation routing on networks derived from the butterfly (see e.g. [22, 19]); in our protocol as well as those, the high-level strategy is to begin the routing with "large" jumps between nodes and converge to the destinations through smaller and smaller jumps.

Finally, we show how to extend our algorithm for routing an assignment to one in which the system runs indefinitely, and new routing requests arrive continuously according to an underlying randomized process. This naturally models a setting in which the underlying gossip mechanism serves as a long-running communication substrate on which packets are continuously routed. Using a static-to-dynamic transformation in the spirit of [2, 4], together with a result of Hajek [9], we show how spatial gossip with $\rho = 1$ can form the basis of a stable routing protocol when packets are injected at an inverse poly-logarithmic rate.

2. Preliminaries

In our model, we have a set of n nodes which are located in a metric space that defines a distance $d_{u,v}$ for any pair of nodes. For the most part, we will think of the metric space as $\mathbb{R}^D$, with points spread roughly uniformly in the sense defined in Section 1. However, the results hold for a wider class of metric spaces as defined in [13], essentially any metric space in which balls have uniformly polynomial growth. In particular, we will also consider the metric space in which the nodes correspond to the leaves of a "virtual" balanced binary tree, and the distance between two nodes is the length of the unique path between them in that tree.

We assume that there is an underlying mechanism for point-to-point communication which allows every pair of nodes to communicate, regardless of their distance. Furthermore, we assume that each node can initiate at most one such communication during any one round. Using this point-to-point communication, a gossip mechanism provides a distribution on the connections that are actually made. For two nodes u, v at distance $d = d_{u,v}$, we let $B_u(d) = \{w \mid d_{u,w} \leq d\}$ denote the ball of radius d around u. Then, in the spatial gossip scheme with exponent ρ, node v contacts u with probability proportional to $|B_u(d)|^{-\rho}$. Notice that for points spread uniformly in $\mathbb{R}^D$, this distribution is identical to the one described in the introduction.

In [13], it was suggested that the "output" of the underlying gossip mechanism be considered as an entity of interest in itself, called a *temporal network* [14]. A temporal network is a graph $G = (V, E)$ with edge labels $\lambda(e)$ that denote the time at which the two endpoints of the edge "communicated" (hence, the graph may contain parallel edges). Given a temporal network, we are particularly interested in the existence of *time-respecting paths* [14, 7] — paths P such that the labels on the edges are increasing — since these are the paths along which information could have moved from one endpoint to the other. A gossip mechanism can then be considered as a distribution on temporal networks.

The protocols we develop in this paper are all *atomic*, in the sense discussed informally in the introduction. To formalize this notion, we want to capture the idea that messages in the protocol can only be stored, copied, and forwarded, but not modified in other ways. Specifically, we suppose that before the start of the protocol, each node v generates a message $\mu_\tau(v)$ for each time step τ. Then, in step τ of the protocol, v may send a subset (up to a fixed size bound β) of the set consisting of all messages it has received up to that point and the messages $\mu_{\tau'}(v)$ for all $\tau' \leq \tau$.

3. Basic impossibility results: message routing and resource location

3.1. Impossibility of routing with $\rho < 1$.

We begin with a general theorem that implies our impossibility results for resource location and MST approximation. We consider a *routing* problem in which there is a set of messages M that must be delivered; for each message $\mu \in M$, there is a source node s_μ where it originates and a *set* of possible destinations D_μ. The message is routed successfully when it reaches any node $v \in D_\mu$. (This includes the natural special case in which all sets D_μ have size 1; we invoke this greater generality because it will be useful in proving impossibility results for approximate minimum spanning trees in a subsequent section.) Now, for an underlying gossip mechanism, we consider whether there exists an atomic protocol to solve this message routing problem in a small number of steps. Note that there may be additional messages sent; however, we will be concerned only with the delivery of messages in the set M.

Theorem 3.1 *Consider an arbitrary set M of m distinct messages μ with sources s_μ and destination sets D_μ, such that $|D_\mu| \leq \Delta$ for all messages μ. Suppose the underlying gossip mechanism has the property that for any pair (u, v) of nodes, the probability that they communicate (either because u calls v or vice versa) is bounded by p, and the number of messages that can be exchanged during any one connection is bounded by β.*

Then, the expected number of messages that have been delivered to any one of their destinations within t rounds is $O(\Delta \cdot p \cdot t \cdot (\beta t n + m))$.

Before giving the actual proof, we outline a simpler and more intuitive version, giving a weaker result for the following special case: each message μ has a unique destination, the underlying gossip mechanism is uniform gossip, and only one message can be forwarded during any communication connection. That is, each node in each round calls another node chosen uniformly at random, and forwards a copy of some message it holds.

Throughout the execution, there are n distinct messages, but they may exist in multiple copies, and at different nodes. Initially, there is a total of n copies (over all messages), and since at most n copies are added in each round, there are at most τn copies at the beginning of round τ, and tn at the end of the protocol. By the Pigeon Hole Principle, there are at least $(1 - \epsilon)n$ messages such that there are at most $\frac{t}{1-\epsilon}$ copies of each of those messages (for any $\epsilon > 0$). Hence, there are at most $\frac{t}{1-\epsilon}$ nodes having one of these *rare* messages at any point in the protocol.

In order for a rare message to reach its destination, the destination must at some point during the t rounds have communicated with one of the at most $\frac{t}{1-\epsilon}$ owners of the rare message. By the Union Bound, the probability of this is at most $t \cdot \frac{t}{(1-\epsilon)n}$, and by linearity of expectation, the expected number of rare messages reaching their destinations is at most $t^2 = o(n)$ for poly-logarithmic time bounds t. Hence, the expected number of messages reaching their destination in poly-logarithmic time is at most $\epsilon n + o(n)$.

Proof of Theorem 3.1. Let $\mathcal{K}_\tau(u,\mu)$ the random event that at time τ or earlier, node u has received the message μ, and $K_\tau(u,\mu)$ the 0-1 indicator variable for that event. Also, we let $\mathcal{C}_\tau(u,v)$ denote the event that nodes u and v communicate in round τ, either because u called v, or because v called u. Finally, we write $\mathcal{A}_\mu = \bigcup_{u \in D_\mu} \mathcal{K}_t(u,\mu)$ for the event that the message μ has reached any of its destinations by time t, and A_μ for its 0-1 indicator variable. Then, the number N of messages having reached any of their destinations at time t can be expressed as $N = \sum_\mu A_\mu$.

We want to bound the probability of the event $\mathcal{A}_\mu$. In order for a node u to receive μ, it must at some time τ communicate with a node v that had received μ earlier, so

$$\mathcal{K}_t(u,\mu) \quad \subseteq \quad \bigcup_{\tau,v} \mathcal{C}_\tau(u,v) \cap \mathcal{K}_\tau(v,\mu).$$

Applying the Union Bound, the fact that communication partners are chosen independently of the data held, and the upper bound of p on the probability of the events $\mathcal{C}_\tau(u,v)$, this yields

$$\begin{aligned} \text{Prob}[\mathcal{K}_t(u,\mu)] &\leq \sum_{\tau,v} \text{Prob}[\mathcal{C}_\tau(u,v)] \cdot \text{Prob}[\mathcal{K}_\tau(v,\mu)] \\ &\leq p \cdot \sum_{\tau,v} \text{E}\left[K_\tau(v,\mu)\right]. \end{aligned}$$

Taking the Union Bound over all nodes $u \in D_\mu$, we obtain that $\text{Prob}[\mathcal{A}_\mu] \leq \Delta \cdot p \cdot \sum_{\tau,v} \text{E}\left[K_\tau(v,\mu)\right]$.

By linearity of expectation, the number of messages μ that have reached a destination by time t is the sum of the probabilities for the events $\mathcal{A}_\mu$, taken over all messages μ.

$$\begin{aligned} \text{E}[N] &= \sum_\mu \text{Prob}[\mathcal{A}_\mu] \\ &\leq \Delta \cdot p \cdot \sum_{\mu,\tau,v} \text{E}\left[K_\tau(v,\mu)\right] \\ &= \Delta \cdot p \cdot \text{E}\left[\sum_{\tau,\mu,v} K_\tau(v,\mu)\right]. \end{aligned}$$

At time 0, each message μ is only known to its source s_μ, so $\sum_{v,\mu} K_0(v,\mu) = m$. During each round of communication, exactly n calls are made, and each call transmits at most β messages. Hence, there are at most $n \cdot \beta$ pairs (μ, v) such that $K_\tau(v,\mu) = 0$ and $K_{\tau+1}(v,\mu) = 1$. By induction, it is easy to see that for all times $\tau \leq t$,

$$\textstyle\sum_{v,\mu} K_\tau(v,\mu) \quad \leq \quad \beta\tau \cdot n + m \quad \leq \quad \beta t \cdot n + m$$

Hence, the above expectation can be bounded as

$$\begin{aligned} \text{E}[N] &\leq \Delta \cdot p \cdot \text{E}\left[\sum_\tau (\beta t \cdot n + m)\right] \\ &= \Delta \cdot p \cdot t \cdot (\beta t n + m), \end{aligned}$$

completing the proof. ■

As a corollary, we obtain a result for spatial gossip with exponent $\rho < 1$, packets with a unique destination, and the restriction that at most one message is forwarded in each communication step.

Corollary 3.2 *For spatial gossip with exponent $\rho < 1$ and the number of messages restricted to 1 per communication, we obtain that in $\log^\kappa n$ rounds, the expected number of successfully routed messages is $O(n^\rho \log^{2\kappa} n)$, and in particular $o(n)$.*

3.2. Resource Location with $\rho < 1$ or $\rho > 2$.

For exponent $\rho > 2$, the impossibility of resource location results from messages traveling too slowly: with high probability, it takes time $\Omega(n^{\rho-2})$ for a message to travel distance n along the line (this was proved in [14]). Hence, if there is only one resource at node 1, node n will only find out about *any* resource after time $\Omega(n^{\rho-2})$.

For the rest of this section, we consider the resource location problem for $\rho < 1$. Let X be the subset of nodes consisting of the resource-holders. We will say that an atomic resource location protocol is a t-round c-approximation if after t rounds of communication, each node v has received with high probability a copy of a message that originated at a resource-holder r' such that $d_{v,r'} \leq c \cdot \min_{r \in X} d_{v,r}$.

We now show how to construct an instance of the resource location problem that essentially contains an instance of the message routing problem described above. Let $\epsilon < 1 - \rho$ and $c = n^\epsilon$, and consider a set of n nodes in $\mathbb{R}$ positioned at the points $\{1, 2, \ldots, n\}$. There is a resource-holder r_j at each point of the form $2jc$ for natural numbers $j = 1, 2, \ldots, n^{1-\epsilon}/2$. Consider the node x_j at the point $2jc - 1$. The protocol will only be a c-approximation if x_j receives a forwarded copy of the message originating at r_j, and hence we have an instance of the message routing problem with messages $\mu_{r_1}, \ldots, \mu_{r_{n^{1-\epsilon}/2}}$; here, message μ_{r_j} has source r_j and destination x_j.

From Theorem 3.1, applied to spatial gossip with parameter $\rho < 1$, $m \leq nt$, and β and t poly-logarithmic in n, we obtain that the expected number of nodes x_j having received a copy of a message originating at their nearest resource is $O(n^\rho \log^\kappa n) = o(n^{1-\epsilon})$. For the remaining $\Omega(n^{1-\epsilon})$ nodes x_j, the distance to their second-closest resource is at least by a factor $c = n^\epsilon$ larger than the distance to the closest resource, proving the following corollary.

Corollary 3.3 *Fix $\rho < 1$. For any $\epsilon < 1 - \rho$, there is no atomic resource location protocol based on spatial gossip with exponent ρ that is a t-round n^ϵ-approximation, where t and the number of messages β per communication step are both poly-logarithmic in n.*

4. Approximate Minimum Spanning Trees

In this section, we investigate protocols for a more complex problem in a metric space: the construction of an approximate minimum spanning tree on a set of N *terminal nodes*. In addition to designing an efficient protocol for this problem, we are interested, as before, in the influence of the underlying gossip mechanism. We first present a protocol based on spatial gossip with exponent $\rho \in [1, 2)$ that constructs a spanning tree within an $O(\log N)$ factor of minimum. We then show that there is no atomic protocol based on spatial gossip with exponent $\rho < 1$ that constructs a spanning subgraph whose expected cost is within an $O(N^{\epsilon})$ factor of minimum, for any $\epsilon < 1 - \rho$.

As in Section 2, we begin with a set of n nodes located at points spaced approximately uniformly in $\mathbb{R}^D$. A subset X of these nodes of size $N \leq n$, called *terminals*, wishes to construct an approximate minimum spanning tree T on X, using the underlying gossip mechanism. (Note that it will not be necessary for all the terminals to be known to one another at the start.) The output of the protocol is a spanning tree T on X such that for each edge $e = (x, y)$ of T, at least one of the ends x or y has stored e. Hence, the tree is represented in a distributed fashion among all the terminals. Notice that we are *not* seeking to construct a Steiner tree using the non-terminal nodes as Steiner nodes. The non-terminals participate in the communication and distributed computation effort, but the final tree contains only terminal nodes.

4.1. A protocol for $\rho \in [1, 2)$.

The motivating idea of the protocol is to approximately simulate an execution of Boruvka's MST algorithm [3, 20], in which nodes repeatedly link to their nearest neighbors, contract the resulting connected components, and iterate. Unfortunately, if the protocol is to run in poly-logarithmic time using gossip, it seems difficult to implement this algorithm directly. First, using the guarantee from [13], we will only be able to obtain an *approximately* nearest neighbor for each node. This poses a problem, since the resulting set of edges from nodes to their approximately nearest neighbors need not be acyclic, and so the output may not even be a tree. Moreover, it is not clear how to implement the contraction of components in poly-logarithmic time.

We define an atomic protocol that runs in $O(\log N)$ phases in expectation. In each phase, certain terminals will be *active*, and the rest will be *inactive*. During a given phase, all active terminals try to find an approximately nearest neighbor among the other active terminals, and then add an edge based on a random symmetry-breaking test to ensure that no cycle would be obtained.

In order to obtain approximation guarantees on the distance of the neighbor that can be found using gossip, we require the following Lemma 4.1, which is an easy corollary of a result in [13]. For a path P with vertices $v_1, \ldots, v_k$, let its *path distance* be $d(P) = \sum_{i=1}^{k-1} d_{v_i, v_{i+1}}$.

Lemma 4.1 *For an exponent $\rho \in [1, 2)$, there is a constant c and a poly-logarithmic time-bound f, such that for any two nodes v, w and any time τ, the execution of spatial gossip with exponent ρ produces a time-respecting v-w path P contained in the time interval $[\tau, \tau + f(n)]$ of path distance at most $cd_{v,w}$ with high probability.*[1]

At the beginning of the protocol, all terminals are active, and the protocol ends with exactly one active terminal. Each phase lasts for $f(n)$ rounds of gossip. We maintain the invariant that each inactive terminal will know the identity of a single edge incident to it in the tree T.

In a given phase j, each node v (not only each terminal) will try to determine the *two* active terminals that are closest to it, and store this information in a set $S_\tau(v)$, where τ denotes the time step within phase j. For purposes of initialization, we will imagine that there are two "dummy terminals" $\perp_1$ and $\perp_2$, each at distance ∞ from all nodes. For active terminals x, the set $S_\tau(x)$ begins the phase initialized to $\{x, \perp_1\}$, and for all other nodes v, the set is initialized to $\{\perp_1, \perp_2\}$. Also, at the beginning of phase j, each active terminal x chooses a random number $\pi_j(x) \in \{1, 2, \ldots, n^3\}$.

All nodes then perform $f(n)$ iterations of the spatial gossip algorithm with an exponent $\rho \in [1, 2)$. When w calls v in a given step, w sends the (at most two) messages corresponding to the non-dummy elements of its current set $S_\tau(w)$. This causes v to update $S_\tau(v)$ in the obvious way: it chooses the two distinct closest nodes from among the (at most four) nodes in the union $S_\tau(v) \cup S_\tau(w)$. For each terminal $x \in S_\tau(v)$, node v also stores and forwards the random number $\pi_j(x)$ associated with x.

At the end of the $f(n)$ steps in phase j, each active terminal x has a set $S_\tau(x)$ that consists, with high probability, of its own name (at distance 0) and the name of some other active terminal y. If $\pi_j(x) > \pi_j(y)$, then x stores the edge (x, y) as part of the eventual tree T and becomes inactive for the remainder of the protocol; otherwise, x does not store any edge and it remains active in phase $j + 1$.

This defines the full protocol. We note that it is atomic, following the definition in Section 2. Theorem 4.2 provides guarantees on the protocol's performance.

Theorem 4.2 *After an expected number of phases bounded by $O(\log N)$, there is exactly one terminal left. At that point, the edges stored by inactive terminals form a spanning tree whose cost is $O(\log N \cdot C(X))$ in expectation, where $C(X)$ is the cost of a minimum spanning tree on X.*

[1] In fact, [13] shows that the length of the path is $d_{v,w} + o(d_{v,w})$ with high probability.

Proof. We first argue that the protocol produces a spanning tree. Let E denote the set of edges constructed. By induction on j, we see that every terminal $x \in X$ has a path in (X, E) to some terminal that was active in phase j. Since there is only a single active terminal at the end of the protocol, this implies that (X, E) is connected. Now, suppose that (X, E) contained a cycle Γ. We can orient each edge $e \in E$ from the terminal that stored e to the other end of e. In this orientation, at most one edge leaves any node in X, so Γ must be a directed cycle with respect to this orientation. If we consider the terminal x on Γ that was active for the maximum number of phases (say through phase j), breaking ties for the x with minimum value $\pi_j(x)$, then we obtain the following contradiction: x would not have constructed an edge to the next node on C. Thus, (X, E) is both connected and acyclic, and hence a spanning tree.

In the remainder of the proof, we will bound the expected cost of the spanning tree found by the protocol, by showing that the expected number of phases is $O(\log N)$, and the cost of edges added in any one phase is within a constant factor of the minimum spanning tree cost $C(X)$. The central fact is the following lemma.

Lemma 4.3 *Let j be a phase with at least two active terminals. Let x be a terminal active in phase j, and y the closest other active terminal to x. With high probability, at the end of phase j, $S_\tau(x)$ will contain x and another active terminal y' with the property that $d_{x,y'} \leq c \cdot d_{x,y}$ (where c is the constant from Lemma 4.1).*

Proof. Let τ^* denote the time step in which phase j begins. For purposes of the analysis, we will imagine that for every node v, there are edges to v from each of the two dummy terminals $\perp_1$ and $\perp_2$ with time label τ^*; in the context of the discussion below, it will be important that such a single edge is a time-respecting path contained in the time interval $[\tau^*, \tau^*]$.

At a given time $\tau \geq \tau^*$, we define the *bundle* of paths incident to a node v, denoted $B_\tau(v)$, as follows. Let $x_1, \ldots, x_r$ be the active terminals (including $\perp_1$ and $\perp_2$) such that there is a time-respecting x_i-v path in the interval $[\tau^*, \tau]$ (note that $r \geq 2$.) For each x_i, choose a time-respecting x_i-v path P_i whose path distance is as small as possible; and assume the indexing is such that $d(P_1) \leq d(P_2) \leq \ldots \leq d(P_r)$. The bundle $B_\tau(x)$ is then the r-tuple of paths $(P_1, P_2, \ldots, P_r)$. We prove the following statement by induction on $\tau \geq \tau^*$.

> $(*)$ Let v be any node, and let $B_\tau(v) = (P_1, P_2, \ldots, P_r)$. Then the set $S_\tau(v)$ contains distinct nodes x'_1 and x'_2 such that $d_{x,x'_1} \leq d(P_1)$ and $d_{x,x'_2} \leq d(P_2)$.

Observe first that, by the initialization step, the condition $(*)$ holds at time τ^*.

For the induction step, we consider the effect of the message(s) from w to v at time $\tau + 1$. We will suppose for simplicity that these are the only messages received by v at time $\tau + 1$; however, messages from other nodes at time $\tau + 1$ are easily handled by applying the following argument sequentially over each. If the bundles $B_\tau(v)$ and $B_\tau(w)$ just before the sending of the message(s) are equal to $(P_1, P_2, \ldots, P_r)$ and $(P'_1, \ldots, P'_s)$, respectively, the bundle $B_{\tau+1}(v)$ after the call can be constructed by merging in a path $P'_i \cdot (w, v)$ for each $P'_i \in B_\tau(w)$, and subsequently eliminating the longer one among two paths originating with the same terminal. By the induction hypothesis, there is a terminal $y_1 \in S_\tau(w)$ such that $d_{y_1,w} \leq d(P'_1)$; hence, by the triangle inequality, $d_{y_1,v} \leq d(P'_1) + d_{w,v} = d(P'_1 \cdot (w, v))$. Also, the other node $y_2 \in S_\tau(w)$ satisfies $d_{y_2,w} \leq d(P'_2)$, and analogously we have $d_{y_2,v} \leq d(P'_2 \cdot (w, v))$. Thus, the smallest two distances from v to nodes in the union of $S_\tau(v)$ and $S_\tau(w)$ will be at most the corresponding minima of the path distances of $P_1, P_2, P'_1 \cdot (w, v), P'_2 \cdot (w, v)$, and so the induction step follows.

Now, for an active terminal x, let y be the closest active terminal other than x itself. By Lemma 4.1, there will, with high probability, be a time-respecting path of path-distance at most $c \cdot d_{x,y}$ by the end of the phase, so the other terminal $y' \neq x$ that is contained in $S_\tau(x)$ at the end of phase j will satisfy $d_{x,y'} \leq c \cdot d_{x,y}$, completing the proof. ■

As a consequence of Lemma 4.3, each active terminal x with high probability has another active terminal y (not a dummy terminal) in its set $S_\tau(x)$ at the end of any phase. With probability $\frac{1}{2} - o(1)$, we have $\pi_j(x) > \pi_j(y)$, and so an edge will be formed. Hence each terminal becomes inactive in each phase with probability $\frac{1}{2} - o(1)$, and so the expected number of phases is $O(\log N)$.

It remains to bound the cost of the edges added in one phase. Let X_j denote the set of active terminals in phase j. Since $X_j \subseteq X$, the Steiner ratio in metric spaces implies that $C(X_j) \leq 2C(X)$. We know that the minimum spanning tree on X_j includes a shortest edge incident to each node; thus, if each node in X_j were to construct such a shortest edge, the total edge length — counting an edge twice if it is constructed from both ends — would be at most $2C(X_j)$. However, each node in X_j actually constructs an edge that is within a factor c of shortest; hence, the total edge length added in phase j is at most $2c \cdot C(X_j) \leq 4c \cdot C(X)$, completing the proof. ■

4.2. Impossibility results for $\rho < 1$ and $\rho > 2$.

The impossibility for exponent $\rho > 2$ can be shown with a very simple example. There are n nodes at positions $\{1, \ldots, n\}$ on the line, and the set of terminals is $X = \{1, n\}$. It was proved in [14] that with high probability, it will take time $\Omega(n^{\rho-2})$ until a message originating with nodes 1 or n has traveled distance $n - 1$, and until that point, no edge can be stored at either node. Thus, with high probability, it takes time $\Omega(n^{\rho-2})$ until *any* spanning subgraph is constructed.

For the remainder of this section, we will prove the impossibility with exponents $\rho < 1$. We will say that an atomic minimum spanning tree protocol on a set of terminals X is a t-round c-approximation if after t rounds of communication the following holds: there is a spanning subgraph (X, E) of total cost at most $c \cdot C(X)$ so that, for each $e = (v, w) \in E$, at least one of v or w has received a message that originated at the other. (Note that this is in fact a much weaker condition than is satisfied by our protocol above.)

As another corollary of Theorem 3.1, we now derive an impossibility result for atomic minimum spanning tree protocols based on spatial gossip with exponent $\rho < 1$. Specifically, choose any $\epsilon < 1 - \rho$, and let $\kappa = \frac{1-\epsilon+\rho}{2\rho}$, and $\delta = N^{\kappa+\epsilon-1}$. Notice that $1 < \kappa < \frac{1-\epsilon}{\rho}$. For a given number N of terminals, let $n = N^\kappa$ be the total number of nodes, which are placed at equal distance 1 on the line. The N terminals are the nodes with coordinates $i \cdot \delta + j$, for $i = 0, \ldots, N^{1-\epsilon} - 1$ and $j = 1, \ldots N^\epsilon$, where the nodes for a particular i are said to form a *cluster*.

For an edge $e = (u, v)$ to be considered part of a spanning subgraph, at least one of the nodes u, v must have received a message originating at the other. In a good approximate spanning tree, most nodes should have edges within their cluster, so we are considering messages $\mu_\tau(v)$ whose destination set consists of all other $N^\epsilon - 1$ nodes in v's cluster. Each node v may generate at most one new such message in each time step τ, so the number of messages is $m \leq N \cdot t$.

By applying Theorem 3.1 to the above routing problem, we obtain that the expected number of messages having reached one of their destinations by time t is bounded by $O(\beta \cdot n \cdot N^\epsilon \cdot n^{\rho-1} t^2) = O(N^{\kappa\rho+\epsilon} \beta t^2) = o(N)$ whenever both β and t are poly-logarithmic, by the upper bound on κ. This is an upper bound on the number of edges both of whose endpoints lie within the same cluster, and hence also an upper bound on the number of edges with cost less than $\delta - N^\epsilon$. Because $\kappa > 1$, this cost $\delta - N^\epsilon = N^{\kappa+\epsilon-1} - N^\epsilon$ is $\Omega(N^{\kappa+\epsilon-1})$.

Since any spanning subgraph must contain $\Omega(N)$ edges, of which in expectation only $o(N)$ can have cost less than $\Omega(N^{\kappa+\epsilon-1})$, the solution has expected cost at least $\Omega(N \cdot N^{\kappa+\epsilon-1}) = \Omega(N^{\kappa+\epsilon})$. However, the minimum spanning tree has cost at most $n - 1 = O(N^\kappa)$, and hence the solution found by the protocol cannot approximate the minimum spanning tree by a factor of $O(N^\epsilon)$ with respect to expected cost, proving the following theorem.

Theorem 4.4 *Consider any atomic protocol for computing an approximately cheapest spanning subgraph, based on spatial gossip with exponent $\rho < 1$. If each node can send only a poly-logarithmic number β of messages per round, and if the protocol runs for a poly-logarithmic number of rounds, then the expected cost of the spanning subgraph it produces will be at least a factor of $\Omega(N^\epsilon)$ larger than the cost of a minimum spanning tree, for any $\epsilon < 1 - \rho$.*

5. Routing permutations

In this section, we further investigate the problem of message routing. We show that for exponent $\rho = 1$, there is a simple protocol routing a set of messages in poly-logarithmic time, with high probability. We complete the trichotomy presented in Section 1 by showing that there are permutations that cannot be routed by any protocol building on top of spatial gossip with exponent $\rho > 1$. Finally, we show how to extend the analysis of the simple protocol with $\rho = 1$ to apply in the case of the protocol running for an indefinite time, with continuously arriving messages.

5.1. A simple protocol for exponent $\rho = 1$

We consider a protocol which uses the metric defined by the length of the unique path between two leaves in the "virtual" balanced binary tree (see Section 2). This keeps the description and proofs cleaner — however, the results extend to other metric spaces as well. Each node v has a queue Q_v in which it stores its current set of messages. The protocol is parametrized by an arbitrary queueing discipline, and in each round, each node v executes the following.

> Choose a message μ according to the queueing discipline, and a communication partner w according to the underlying spatial gossip scheme. If w is strictly closer to the destination d_μ than v, then forward μ to w; else do nothing.

We let Z denote the time at which the last message μ reaches its destination, and give high-probability bounds for Z. The bounds certainly depend on how many packets originate with or are destined for any one node. For a node v, let its *message volume* l_v be the number of messages μ such that $s_\mu = v$ or $d_\mu = v$, and let L be an upper bound on l_v for all nodes v. The guarantee for the protocol is as follows:

Theorem 5.1 *Fix a constant $a > 2L \cdot \log^3 n$. With probability at least $1 - (n \cdot e^{-\frac{(a-2L\log^3 n)^2}{16L\log^5 n}} + n \cdot L \cdot e^{-\frac{a}{8\log n}})$, all messages μ have reached their destination by time a, i.e. $Z \leq a$.*

The proof of this theorem will be given in the full version of the paper. The intuition is as follows: we first bound the number of messages that visit any one node v by $O(L \log n)$

with high probability. To this end, it is crucial to notice that the paths of different messages are created independently, which allows us to apply Chernoff Bounds. The probability of a node v seeing a message with destination w at distance i from v is at most 2^{-i}, and there are 2^i such nodes w, each the destination of at most L messages.

As a second step, we notice that any message can be delayed by at most all messages whose path it meets, and each message's waiting time at the front of the queue is distributed as a geometric variable with success probability $1/\log n$. Because each message is forwarded at most $\log n$ times, its arrival time is bounded by the sum of $O(L\log^2 n)$ such geometric variables, which has expectation $O(L\log^3 n)$. We obtain the desired high-probability result using Chernoff bounds and union bounds.

By plugging in particular values of L and a, we obtain the following corollary:

Corollary 5.2 *The routing is accomplished in $O(L\log^3 n)$ with probability at least $1-2/n$. In particular, a permutation is routed in $O(\log^3 n)$ steps with probability at least $1-2/n$.*

5.2. Impossibility of routing with exponent $\rho > 1$

We now complete the proof of the trichotomy stated in Section 1 by proving that there are routing assignments which cannot be routed in poly-logarithmic time using spatial gossip with exponent $\rho > 1$.

For simplicity, we prove this result for the metric of path distances in the "virtual" binary tree (see Section 2). In our routing assignment, each node i is the source of a message with destination $n-i$, i.e. a total of n messages have to cross into the other subtree of height $\log n - 1$.

Theorem 5.3 *Consider any protocol which is restricted to forwarding at most β messages during any one communication step. The expected number of messages that reach their destination by time t when spatial gossip with exponent $\rho > 1$ is used is $O(n^{2-\rho}\cdot\beta t)$.*

Proof. By definition of the spatial gossip mechanism, in each round, each node $u \leq n/2$ communicates with a node $v > n/2$ with probability $O(n^{1-\rho})$, hence over all of the $n/2$ such nodes u, at most $O(n^{2-\rho})$ pairs of nodes (u,v) with $u \leq n/2$ and $v > n/2$ communicate during any one round on expectation. In each such communication, at most β messages are forwarded, and this is done for at most t rounds. Hence, at most $O(n^{2-\rho}\cdot\beta t)$ messages cross into the other subtree in expectation, and crossing into the other subtree is necessary to reach the destination. ■

Notice that the limitation for exponents $\rho > 1$ is very different from the one for exponents $\rho < 1$. Here, the problem is simply one of limited "bandwidth", so even knowledge of future random outcomes is not sufficient. On the other hand, for small exponents, the problem was the lack of structure among the connections that would help in guiding messages.

By choosing both β and t poly-logarithmic in n in the above theorem, we obtain that only $o(n)$ messages reach their destination in poly-logarithmic time.

5.3. Dynamically arriving messages

In a real-world system employing a gossip-style mechanism for message forwarding, the emphasis is usually on having the system run for an indefinite amount of time, whereas the above analysis implicitly assumed that packets were only being injected and forwarded starting at a specific time, and after successful delivery of all packets, the system was stopped. Here, we extend the analysis to the case of a system running for an indefinite time.

As a simple model for message arrival, we assume that in each time step, and for each pair (u,v) of nodes, a new message μ with source $s_\mu = u$ and destination $d_\mu = v$ is generated independently with probability $\gamma = O(\frac{1}{n\log^4 n})$. Note that this means that on average, each node u becomes the source of a new message every $O(\log^4 n)$ rounds. The routing protocol is essentially unchanged from the previous section, except that we specify that nodes v with more than one queued packet will use the longest-in-system (LIS) rule to select the packet to forward. This rule specifies that at any time, a node tries to forward the packet from its queue which has its injection longest in the past. In this model, we obtain the following guarantee on the protocol:

Theorem 5.4 *For any message μ, the time from its injection until its delivery is at most $O(\kappa\log^8 n)$, with probability at least $1-\frac{1}{n^\kappa}$. In particular, the system is stable.*

We use the result from Theorem 5.1 in the proof, but it is obvious that the techniques employed there will not be sufficient. Over the execution of the protocol, any queue length or waiting time will eventually be exceeded, so our goal will be to show that such bad events happen rarely, and in particular, that the system recovers quickly enough from them. The proof is left to the full version of this paper, but we will present an outline here.

Strictly for the purpose of analyzing the protocol, we consider a "batched" version of the problem, in which time is divided into *windows*. Messages arriving during one window are stored until all messages from the previous window have reached their destinations, at which point they are released into the system, and routed as in Section 5.1. By the choice of the message generation probability, the routing assignment will be sufficiently permutation-like to give an expected routing time $O(\log^4 n)$. By making the windows slightly longer than the expected routing time, the protocol has "time to spare".

Now, we consider the random variable Y_j which characterizes how much behind schedule the j^{th} batch is. The intuition of a batch having time to spare in order to catch up with delays caused by previous batches can be formalized as the sequence (Y_j) having sufficiently large negative drift whenever it exceeds a certain bound. After showing that the sequence is not too "jumpy", i.e. the sequence of differences $(Y_{j+1} - Y_j)$ has an exponentially decreasing tail, we can apply a Theorem by Hajek [9] to obtain high-probability bounds for the delay staying bounded by a poly-logarithmic time bound.

Finally, to complete the proof, we prove the intuitive fact that the delay of a batch of messages does not increase if they are actually released at their arrival times instead of the time at which the previous batch finishes. Here, it is important to notice that the LIS queueing discipline ensures that messages from later batches do not interfere with the completion of earlier batches.

Acknowledgments

We thank Alan Demers for useful discussions about the results presented here.

References

[1] D. Agrawal, A. E. Abbadi, and R. Steinke. Epidemic algorithms in replicated databases. In *Proc. 16th ACM Symp. on Principles of Database Systems*, 1997.

[2] D. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, F. Leighton, and Z. Liu. Universal stability results for greedy contention–resolution protocols. In *Proc. 37th IEEE Symp. on Foundations of Computer Science*, 1996.

[3] O. Boruvka. O jistém problému minimálním. *Práca Moravské Přírodovědecké Společnosti*, 3, 1926.

[4] A. Broder, A. Frieze, and E. Upfal. Existence and construction of edge low congestion paths on expander graphs. In *Proc. 29th ACM Symp. on Theory of Computing*, 1997.

[5] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Stuygis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proc. 7th ACM Symp. on Operating Systems Principles*, 1987.

[6] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. *Random Structures and Algorithms*, 1, 1990.

[7] F. Göbel, J. O. Cerdeira, and H. Veldman. Label-connected graphs and the gossip problem. *Discrete Mathematics*, 87:29–40, 1991.

[8] I. Gupta, R. van Renesse, and K. Birman. Scalable fault-tolerant aggregation in large process groups. In *Proc. Conf. on Dependable Systems and Networks*, 2001.

[9] B. Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Adv. Appl. Prob.*, 14, 1982.

[10] M. Harchol-Balter, F. Leighton, and D. Lewin. Resource discovery in distributed networks. In *Proc. 18th ACM Symp. on Principles of Distributed Computing*, 1999.

[11] S. Hedetniemi, S. Hedetniemi, and A. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18:319–349, 1988.

[12] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *Proc. 41st IEEE Symp. on Foundations of Computer Science*, 2000.

[13] D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. In *Proc. 33rd ACM Symp. on Theory of Computing*, 2001.

[14] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. In *Proc. 32nd ACM Symp. on Theory of Computing*, 2000.

[15] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. 32nd ACM Symp. on Theory of Computing*, 2000.

[16] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[17] M. Lin, K. Marzullo, and S. Masini. Gossip versus deterministic flooding: Low message overhead and high reliability for broadcasting on small networks. Technical Report CS99-0637, University of California at San Diego, 1999.

[18] L. Lovász. Communication complexity: A survey. In B. Korte, L. Lovász, H. Prömel, and A. Schrijver, editors, *Paths, Flows, and VLSI-Layout*. Springer, 1990.

[19] B. Maggs and B. Vöcking. Improved routing and sorting on multibutterflies. In *Proc. 29th ACM Symp. on Theory of Computing*, 1997.

[20] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1990.

[21] B. Pittel. On spreading a rumor. *SIAM J. Applied Math.*, 47, 1987.

[22] E. Upfal. An $O(\log N)$ deterministic packet routing scheme. *J. of the ACM*, 39, 1992.

[23] R. van Renesse. Scalable and secure resource location. In *33rd Hawaii Intl. Conf. on System Sciences*, 2000.

[24] R. van Renesse, Y. Minsky, and M. Hayden. A gossip-style failure-detection service. In *Proc. IFIP Intl. Conference on Distributed Systems Platforms and Open Distributed Processing*, 1998.