UNIVERSITY OF CALIFORNIA
RIVERSIDE

Multi-Broadcasting in Ad-Hoc Radio Networks

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Computer Science

by

Jordan Samuel Kuschner

March 2024

Thesis Committee:

Dr. Marek Chrobak, Chairperson
Dr. Yan Gu
Dr. Silas Richelson

The Thesis of Jordan Samuel Kuschner is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am grateful to my advisor, Dr. Marek Chrobak, for I would not have been able to publish this thesis without his endless dedication and support. I am also grateful to my co-chairpeople, Dr. Yan Gu and Dr. Silas Richelson who have supported me throughout my academics and have taken the time to hear my dissertation. Additionally, I am grateful for the authors of all cited papers, for this thesis could not have been completed without their publications. Lastly, I am grateful for my colleagues, Yugarshi Shashwat, Chao Wu, and Sarthak Yadav. Although I did not work directly with them on this thesis, I am thankful for all the help and support they have provided as fellow researchers in Professor Chrobak's laboratory.

I dedicate my thesis to my parents, Douglas, Suzanne, my stepfather, Chris, my brothers, Benjamin and Joshua, my girlfriend, McKenna, my cat, Woodford, my grandfathers, Marshall and Frank, my Uncle Dan, my Aunt Hong, my aforementioned colleagues, and some unnamed friends, all of whom supported me throughout my academics and research work in pursuit of my Master's Degree.

ABSTRACT OF THE THESIS

Multi-Broadcasting in Ad-Hoc Radio Networks

by

Jordan Samuel Kuschner

Master of Science, Graduate Program in Computer Science
University of California, Riverside, March 2024
Dr. Marek Chrobak, Chairperson

The challenges of gossiping and broadcasting in ad-hoc radio networks have been well-studied. Broadcasting, or one-to-all exchange, is dissemination of a message from a single source node to all other nodes in the network. Gossiping, or all-to-all exchange, is the dissemination of a message from each node to all other nodes in the network. Many-to-all exchange, on the other hand, has gone relatively unstudied in ad-hoc radio networks. We refer to this challenge as *multi-broadcasting.* Due to the unstudied nature of multi-broadcasting, the fastest previously known deterministic algorithm was the $\tilde{O}(n^{4/3})$ proposed in [10]. If a network has a set Q of nodes with messages where $|Q| = q < n$, we provide three different deterministic multi-broadcasting protocols under three different models. The first assumes knowledge of both the set Q and the value q and runs in time $O(qn \log n \log \log n)$. The second only assumes knowledge of the value of q, and runs in time $O(qn \log^3 n)$. The final model has the same assumptions as the second model, with the additional knowledge that

the nodes of Q are at most a distance σ apart. Under this model, multi-broadcasting can be completed in time $O(q^2\sigma\Delta\log^2 n)$. Lastly, we provide a protocol that can determine the correct value of q in $O(\log n)$ time, thus the need for knowledge of q under the second and third models is not necessary at a small additional cost in runtime.

Contents

List of Figures

Chapter 1

Introduction

In a radio network, the devices act as both transmitters and receivers. If a device is within the transmission range of another device, it will receive transmissions from that device. However, if two devices transmit simultaneously, a *collision* at the receiving device occurs and the transmission is not received. We assume there is no collision detection, so neither transmitters nor the receivers are aware of the collision. With this obstacle, the ad-hoc radio network provides two main challenges: *broadcasting* and *gossiping*. In broadcasting, a single device wants to share its message with all other devices in the network. In gossiping, also known as *total information exchange*, all devices want to share their own message to all other devices in the network. These two challenges further create another challenge: a scenario where many, but not all, devices want to distribute their message throughout the network. We coin this challenge as *multi-broadcasting*.

Related work. Naively, both broadcasting and gossiping can be solved in time $O(n^2)$ by performing n rounds of RoundRobin transmission where for each time slot $i = 1, 2, \cdots, n$ the node with the label i is given the sole opportunity to transmit. In [2], Chlebus et al. proposed a broadcasting algorithm that runs in time $O(n^{11/6})$, the first sub-quadratic upper bound on this problem. DeMarco and Pelc subsequently reduced the upper bound to $O(n^{5/3} \log^3 n)$ in [9] before Chlebus et al. further improved the upper bound to $O(n^{3/2})$ in [1]. A significant breakthrough in broadcasting occurred when Chrobak et al. proposed a nearly optimal $O(n \log^2 n)$ time broadcasting algorithm in [3]. Some time later, DeMarco improved the upper bound to $O(n \log n \log \log n)$ in [8] which is currently the best known upper bound for deterministic broadcasting.

With their nearly linear time broadcasting algorithm, Chrobak et al. proposed the first sub-quadratic gossiping algorithm that ran in time $O(n^{3/2} \log^2 n)$ in [3]. Gossiping later saw some improvement by Xu in [13] by dropping the poly-logarithmic factor for a runtime of $O(n^{3/2})$. This upper bound was again improved to $\tilde{O}(nD^{1/2})$ in [11] where they also proposed a $\tilde{O}(D\Delta^{3/2})$ protocol, where Δ is the maximum in-degree of the network ($\tilde{O}$ notation is a shorthand for asymptotic notation that hides logarithmic factors). Gasieniec, Radzik, and Xin proposed an $\tilde{O}(n^{4/3})$ time gossiping algorithm in [10] which is the fastest deterministic gossiping algorithm to date.

The challenge of multi-broadcasting has gone relatively unstudied, particularly under the ad-hoc radio network model. To date, the only known algorithm for this model was proposed by Clementi et al. in [5] and runs in time $O(D\Delta^2 \log^3 n)$, however this algorithm does not leverage the fact that we have fewer than n rumors to disseminate and so it is essentially equivalent to a naive gossiping algorithm. Prior to this paper, the best way to accomplish multi-broadcasting was to simply use the $\tilde{O}(n^{4/3})$ algorithm for full gossiping proposed by [10].

Our Contribution. In this paper, we explore various methods to improve the upper bound of $\tilde{O}(n^{4/3})$ for distributing a set, Q of rumors such that $|Q| = q < n$, under a variety of different conditions. First, we introduce a naive algorithm for where the source set, Q, is known by all nodes in the network. Then, we modify the naive algorithm to work when the size of Q is known, but not the actual members of Q. Under our final model, we assume the nodes of Q are within some distance σ of each other. For this model, we first show an $O(\sigma\Delta \log n \log \Delta)$ time bounded-radius broadcasting algorithm that transmits a source message μ_v from a node v to all nodes within distance σ of v. For all models, we assume that the maximum in-degree of the network is bounded by $\Delta \leq n$. Later, we describe a protocol that can find the correct value of q through a repeated doubling process such that knowledge of q is not necessary for the second and third models.

Organization. The rest of the paper is organized as follows: in Chapter 2, we review the basic definition of the radio network model and selectors, as well as full [3] and limited [4] broadcasting algorithms. Additionally, we discuss some practical applications of selectors outside the field of radio networks. In Chapter 3, we propose a bounded-radius broadcasting algorithm that can disseminate a rumor to all nodes within a distance σ from v and prove its correctness and runtime of $O(\Delta\sigma \log n \log \Delta)$. Then, in Chapter 4, we describe three different protocols under the previously mentioned environments and prove the correctness and runtime of each. We also describe a procedure to "guess" the value of q using a repeated doubling procedure and prove its correctness. Lastly, in Chapter 5, we wrap up our thoughts and discuss some open ideas for further improvement.

Chapter 2

Preliminary Knowledge

In this section, we provide the basic definition of an Ad-Hoc Radio Network model. We then review the definition of selectors. After, we discuss previous broadcasting algorithms. Lastly, we explain a couple of the many practical applications of selectors outside the scope of radio networks.

Ad-Hoc Radio Network Model. We can model a radio network as a directed graph, $G = (V, E)$ where nodes represent processors and an edge, $e = (u, v)$, denotes the fact that v is within range of u to receive a transmission. We define u to be an *in-neighbor* of v, and we refer to the set of all in-neighbors of v as *NB(v)*. Particularly in our setting, we state that for all $v \in V$, $|NB(v)| \leq \Delta \leq n$. In the Ad-Hoc model, nodes are only aware of their own label, a unique identifier in the set, $\{1, 2, \ldots, n\}$, as well as the upper bound on the number of other nodes, n. Our model allows for an unbounded message size, so a node will aggregate any messages it has already received into a single message and transmit its entire knowledge base in a single transmission. Time is discrete and divided into equal length time-steps, where in each time-step, a node determines whether or not it should transmit. We assume nodes can perform any necessary internal computations in constant time and do not account for these computations in the runtime of our protocols. Additionally, we operate under the assumption that the nodes do not have *collision detection*. That is, if two nodes, u_1 and u_2 transmit along edges (u_1, v), and (u_2, v) in the same time step, a *collision* will occur and node v will receive no transmission. Furthermore, v cannot distinguish a collision impacted transmission from a complete lack of transmission and so collided transmissions are lost. The absence of collision detection necessitates the need for a protocol to efficiently schedule transmissions from each processor in order to either avoid collisions entirely or only allow collisions on nodes who do not need to receive the colliding messages. Note that there exists other radio network models, ad-hoc or not, that do have collision detection. See [12], which explicitly talks about how collision detection can accelerate protocols in the ad-hoc model. For broadcasting, we assume that every node is reachable from the source node, v. For gossiping and multi-broadcasting, we further assume that G is a strongly connected graph.

Selectors. For a given value of k, a *selector*, or a selective family of sets, can be represented as $\bar{S} = S_1 S_2 \ldots S_m$, where for any size-$k$ set $X \subset U$, some set $S_i \in \bar{S}$ intersects with X on exactly one element. For $\{S_i \cap X\} = \{x\}$, we say that S_i *isolates* x. If $\bar{S}$ isolates exactly 1 element from any size-k subset of n, we refer to

$\bar{S}$ as a *weak selector*, and if $\bar{S}$ isolates all k elements from any size-k subset of n, we say that $\bar{S}$ is a *strong selector*.

Selectors have been well studied across the literature and it is known that for a given k, a weak selector has size $m = O(k \log n)$ while a strong selector has size $m = O(k^2 \log n)$. In [7], DeBonis et al. introduced (k, p, n)-selectors which guarantee isolation of at least p elements from a size-k subset of n nodes. They additionally showed that for any $p = \alpha k$ where $\alpha < 1$, a (k, p, n)-selector is larger than a weak selector by a constant factor, thus asymptotically its size is equal to a weak selector, $O(k \log n)$. Using (k, p, r)-notation, a weak selector is a $(k, 1, n)$-selector while a strong selector is a (k, k, n)-selector. We will use this notation throughout the remainder of this paper.

Interleaved Sequences of Selectors. Consider the following group of sets: $X = (x_1, x_2)$, $Y = (y_1, y_2, y_3)$, and $Z = (z_1, z_2, z_3, z_4)$. We can *interleave* these sets into a single, infinitely repeating sequence as shown below:

$$W = (x_1, y_1, z_1, x_2, y_2, z_2, x_1, y_3, z_3, x_2, y_1, z_4, x_1, y_2, z_1, x_2, y_3, z_2, \dots)$$

Such a sequence is called an *interleaved sequence.* The period of an interleaved sequence is the least common multiple, or LCM of the lengths of all individual sequences. In the above example, the period of W is $LCM(2, 3, 4) = 12$. A stage of an interleaved sequence comprised of β individual sequences is the next β steps of the infinite sequence. For our application to selectors, we can interleave $(2^j, 2^{j-1}, n)$-

selectors for $j = 1, 2, \ldots, \log \Delta$ by interleaving the sets $S_i \in \bar{S}_j$ for each j. Because the size of each $(2^j, 2^{j-1}, n)$-selector varies by an increasing power of 2, the period of the interleaved sequence of selectors is simply the size of the largest selector, $O(\Delta \log n) \cdot \beta$, where the β factor accounts for the interleaving. For $\beta = \log \Delta$ the sequence has a period length of $O(\Delta \log n \log \Delta)$. In [10], they show that an interleaved sequence of geometrically increasing strong selectors can transmit a message along a length-k path in time $O(k^2 \log^3 n)$. They call this interleaved sequence of selectors a *path selector*.

Limited and Full Broadcast. In [3], they propose an $O(n \log^2 n)$ time full broadcasting algorithm that will successfully deliver a message from a source node, v, to all other nodes in the network. It is implemented using stages of interleaved $(2^j, 1, n)$-selectors for $j = 1, 2, \ldots, \log n$, where each stage iterates over the next $\log n$ sets of the interleaved sequence of selectors. In the proof of runtime, they show that, over the course of $O(n \log n)$ stages, a source message originating from a node v will reach all other nodes. In [4], they propose an $O(k \log^2 n)$ time limited broadcasting algorithm that successfully delivers the source message to at least k nodes by using $O(k \log n)$ stages of interleaved selectors. We derive our radius broadcasting algorithm in a similar style to these broadcasting algorithms. The currently fastest known deterministic broadcasting algorithm was proposed by DeMarco in [8] and runs in time $O(n \log n \log \log n)$.

Applications of Selectors. It is well-known that the study of selectors has practical applications in many different areas such as resolving multiple-access channel (MAC) contention, group testing, coding, bioinformatics, and of course, radio networks. We discuss here the applications to multiple-access channels and group testing. We will also heavily utilize both $(k, k/2, n)$-selectors and (k, k, n)-selectors in our multi-broadcasting protocols.

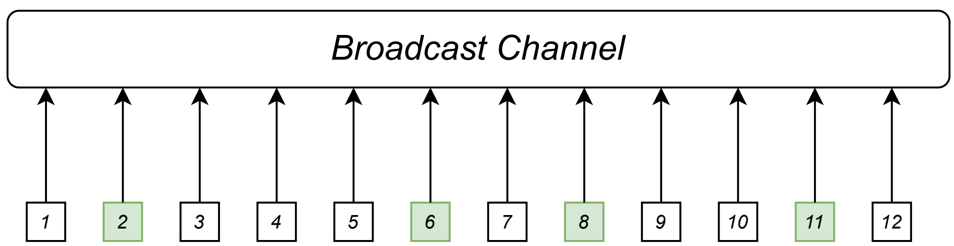

Figure 2.1: A MAC Channel with $n = 12$ nodes connected to it, and a size-k subset of nodes that want to transmit, $\{2, 6, 8, 11\}$.

The MAC contention challenge involves a group of n nodes connected to a broadcast channel, some k of which want to transmit to the channel. If a node successfully transmits to the channel, then the channel will relay that message to all other nodes attached to the channel. If multiple nodes transmit to the channel simultaneously, the messages will collide and be lost in the channel, and the transmitting nodes will have no knowledge that the collision occurred. Figure 2.1 above represents the MAC contention challenge where $n = 12$ nodes are connected to the MAC. If only 1 non-specific node must transmit to the channel then a $(k, 1, n)$-selector can isolate a node from the size-k set in time $O(k \log n)$. For any p that is a constant fraction of k, a

(k, p, n)-selector will isolate p nodes from the size-k set still in $O(k \log n)$ time. If all k nodes must transmit to the channel, then a (k, k, n)-selector can facilitate these k transmissions in time $O(k^2 \log n)$. To target a specific node or group of nodes, we must also use a (k, k, n)-selector. Selectors for MAC contention resolution were studied in [1].

Selectors are also helpful tools in group testing. Consider a case where we know that in a group of n people, k of them are known to be positive for some disease, like COVID-19. Each person has a unique identifier and their own sample. To save tests, a $(k+1, k+1, n)$-selector can be used to isolate the k positives. An administrator will collect n samples, one from each person. They will mix the samples according to the groups specified by a $(k+1, k+1, n)$-selector and apply each mixed sample to a test. If this test reads negative, then the administrator knows that this whole group of people is negative and can eliminate all of them as potential positives. If a test reads positive, then the administrator can eliminate known negative people who also participated in a negative test. Once all negative people are eliminated, only the k positives will remain. Thus, the positive people have been identified using $O(k^2 \log n)$ tests. Group testing algorithms with selectors are described further in [7].

Chapter 3

Bounded-Radius Broadcasting

For networks where the diameter of Q is at most some value, σ, it may be advantageous to perform a broadcast from a source node v to all nodes within a distance σ from v. For networks that have a bounded in-degree of Δ, one could use Phase I of the $\tilde{O}(n\Delta)$-time gossiping algorithm described in [10], however with a cost of using a $O(\Delta^2 \log n)$ strong selector, it is only viable for very small values of Δ. However, the runtime of Phase I in [10] is independent of the size of our message set, Q. As such, our goal is to design a new radius broadcasting algorithm that can efficiently perform a partial broadcast to all nodes within a distance σ from the source node, v. We can modify the broadcasting algorithm described in [3] to achieve this task. In their protocol, they interleave $(2^j, 1, n)$-selectors for $j = 1, 2, \ldots, \log n$ and they define a stage as the next $\log n$ steps of the interleaved sequence of selectors. With the in-degrees of the nodes bounded by $\Delta \leq n$, we can instead interleave $(2^j, 2^{j-1}, n)$-

selectors for $j = 1, 2, \ldots, \log \Delta$. We assume that $\Delta\sigma < n$, otherwise we can use the $O(n \log n \log \log n)$ time full-broadcasting algorithm instead. Formally, the modified procedure works as follows:

Algorithm BrBroadcast(Δ, σ): In each stage, s, for $j = 1, 2, \ldots, \log \Delta$, the transmission set in the j-th step of stage s is $\bar{S}_{j, s \bmod m_j}$.

Theorem 3.1. BrBroadcast(Δ, σ)*, successfully broadcasts a message,* μ_v*, from a source node* v *to all nodes within distance* σ *of* v *in time* $O(\sigma\Delta \log n \log \Delta)$*.*

Proof. Consider any arbitrary fixed path, $P = \langle v_0, v_1, \ldots, v_\sigma \rangle$, where $v_0 = v$. To prove the runtime and correctness of BrBroadcast(Δ, σ), we show that for each $i = 1, 2, \ldots, \sigma$, the delay between message μ_v's arrival at v_i and its arrival at v_{i+1} is at most $2c \cdot \Delta \log n \log \Delta$, where c is the constant in the asymptotic upper bound on the size of a $(2^j, 2^{j-1}, n)$-selector. Figure 3.1 below depicts the path P as part of a subset of nodes in the entire network, where nodes v_i, h, and c have μ_v.

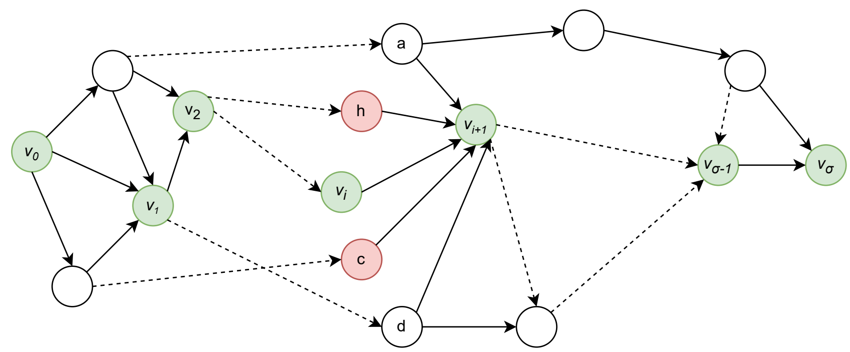

Figure 3.1: A fixed path, P, in green, of length σ originating from v_0. At time t_i, v_i and nodes shaded in red have μ_v. Nodes that are not labeled v_i are other nodes in the network that are not members of P. Dashed edges indicate a multi-hop connection.

Let X be the set of in-neighbors of v_{i+1} such that every node in X has received μ_v. X is dynamic and can grow to size 2^b for $0 \leq b \leq \log \Delta$, where $b = \log(|NB(v_{i+1})|)$. Let t_i be the time that v_i receives μ_v. Therefore, at time t_i, $|X| > 0$. In Figure 3.1, $X = \{v_i, h, c\}$ at time t_i.

Claim 1. $t_{i+1} \leq t_i + 2c \cdot \Delta \log n \log \Delta$.

Claim 1 will show that node v_{i+1} receives μ_v by time $t_i + 2c \cdot \Delta \log n \log \Delta$. Thus, for $|P| = \sigma$, v_σ will receive μ_v after $\sigma \cdot 2c \cdot \Delta \log n \log \Delta = O(\sigma \Delta \log n \log \Delta)$ time steps, so Claim 1 will prove Theorem 3.1.

Let $t' = t_i + 2c \cdot \Delta \log n \log \Delta$. At time t', $|X| \leq 2^b$. We prove Claim 1 by showing that in the interval, $[t_i, t']$, a node in X will transmit μ_v to v_{i+1}. We can divide $[t_i, t']$ into intervals based on the growth of X. For $j = 0, 1, 2, \ldots, b$, we define I_j to be the interval where $2^{j-1} < |X| \leq 2^j$ holds true. Let $X' \subset X$ be the set of nodes in X at the start of an interval, I_j. If the length of I_j is shorter than the length of the appropriate $(2^j, 2^{j-1}, n)$-selector, that is, $|I_j| < c \cdot 2^j \log n \log \Delta$, then the $(2^j, 2^{j-1}, n)$-selector may not isolate any nodes in X' because X will exceed size 2^j before the protocol finishes iterating over the $(2^j, 2^{j-1}, n)$-selector. However, if there exists j such that an interval I_j has length $|I_j| \geq c \cdot 2^j \log n \log \Delta$, then the protocol will iterate over the entire $(2^j, 2^{j-1}, n)$-selector within the first $c \cdot 2^j \log n \log \Delta$ steps of I_j and isolate 2^{j-1} in-neighbors of v_{i+1}, regardless of whether they contain μ_v or not. Since $|X'| > 2^{i-1}$, then, by the pigeonhole principle, at least one node in X' will be isolated by the selector, and this node will successfully μ_v transmit to v_{i+1}.

Claim 2. *There exists j such that an interval, I_j has length $|I_j| \geq c \cdot 2^j \log n \log \Delta$*

We first show that by proving Claim 2, Claim 1 will also hold. Consider the first j where $|I_j| \geq c \cdot 2^j \log n \log \Delta$ holds. Let $t'' = \sum_{\kappa=0}^{j-1} |I_\kappa|$ be the sum of the interval-lengths from I_0 to I_{j-1}. Since these intervals are all shorter than $c \cdot 2^\kappa \log n \log \Delta$, the sum of these intervals is less than $2c \cdot 2^{j-1} \log n \log \Delta$. Thus, $t'' < c \cdot 2^j \Delta \log n \log \Delta$. Then, during the first $c \cdot 2^j \log n \log \Delta$ steps of I_j, the protocol will iterate over the full $(2^j, 2^{j-1}, n)$-selector which will isolate a node from X' and node v_{i+1} will receive μ_v at time t_{i+1} with

$$\begin{aligned} t_{i+1} &\leq t_i + t'' + c \cdot 2^j \log n \log \Delta \\ &\leq t_i + c \cdot 2^j \log n \log \Delta + c \cdot 2^j \log n \log \Delta \\ &= t_i + 2c \cdot \Delta \log n \log \Delta \end{aligned}$$

We now prove Claim 2 by contradiction. Assume that all intervals I_j for $j = 0, 1, 2, \ldots, b$ have length $|I_j| < c \cdot 2^j \log n \log \Delta$. Then the following must hold true:

$$t' - t_i = \sum_{j=0}^{b} |I_j| < c \cdot \sum_{j=0}^{b} 2^j \log n \log \Delta = 2c \cdot 2^b \log n \log 2^b \leq 2c \cdot \Delta \log n \log \Delta$$

showing that $t' - t_i < 2c \cdot \Delta \log n \log \Delta$, which is a contradiction. Therefore, there must be some I_j such that $|I_j| \geq c \cdot 2^j \log n \log \Delta$, proving Claim 2. This completes the proof of Theorem 3.1. □

Chapter 4

Protocols Under Various Models

In this chapter, we propose multi-broadcasting under three different models. The first model assumes full knowledge of Q: that is, all nodes in the network know which nodes are in Q and which nodes are not, and consequently all nodes know the size, $|Q| = q$, as well. The second model assumes the nodes know the size q but do not have any knowledge on the specific members of Q. Nodes that are in Q are aware that they are part of Q by virtue of having a message, but they are not aware of other members of Q. Under the third model, Q has a diameter at most σ. That is, any two nodes in Q are reachable via a path of length at most σ. We then describe a different protocol that can guess the correct value, q in $\log n$ time, thus enabling protocols under the second or third model to work without knowledge of the size of Q. Under all three of models, we assume that the network has a bounded maximum in-degree of $\Delta \leq n$.

4.1 Full Knowledge of Q

We start with a naive algorithm under the assumption that all nodes in the network have knowledge of both the size of Q and the individual members of Q. We utilize the $O(n \log n \log \log n)$ time broadcasting algorithm to distribute the messages of Q throughout the network.

Protocol NaiveMB(Q, n): The node in Q with the smallest label initiates a broadcast. Once this broadcast is complete, the node in Q with the next smallest label initiates a broadcast. This continues until the node with the largest label in Q initiates a broadcast. When this final broadcast finishes, the protocol terminates.

Theorem 4.1. NaiveMB(Q, n) *successfully completes multi-broadcasting in time* $O(qn \log n \log \log n)$.

Proof. With full knowledge of Q, each node in Q will know exactly at what time step it can initiate a full broadcast without contention in the network. Nodes not in Q will not initiate a broadcast and act only as participating members in broadcasts initiated by nodes in Q. After a single execution of broadcasting from v, all nodes in the network have message μ_v. Then, after q executions of broadcasting from all nodes in Q, all nodes in the network will have all messages of Q. □

4.2 Knowledge of $|Q|$

Without knowledge of the members of Q, the nodes in Q can not coordinate their broadcasts by labels. Instead, we use a (q, q, n)-selector of size $m = O(q^2 \log n)$ to coordinate broadcasts. Additionally, rather than performing a full broadcast from each node in Q, they will instead perform q limited broadcasts using LimitedBroadcast(ρ) from [4]. The value of ρ will be determined later. To facilitate limited broadcasting via the selector, we iterate over the m sets of the selector in phases, where the length of each phase is equal to the length of LimitedBroadcast(ρ), $O(\rho \log^2 n)$. After the limited broadcast phases, full dissemination of each message will be completed using a set cover concept, described in Phase II of DoGossip in [3], which we will refer to as SetCoverFlush(). Formally, the protocol works as follows:

Protocol StrongMB(q, n):

for $i = 1, 2, \ldots, m$

– **during** Phase i

—– **if** $v \in S_i$, v initiates LimitedBroadcast(ρ).

SetCoverFlush()

Theorem 4.2. *For $\rho = (n/q) \log^{1/2} \log n$, StrongMB($q, n$) completes multi-broadcasting in time $O(qn \log^3 n \log^{1/2} \log n)$.*

Proof. LimitedBroadcast(ρ) disseminates a message μ_v from a source node v to ρ nodes in time $O(\rho \log^2 n)$. Using a (q, q, n)-selector of size $O(q^2 \log n)$ will isolate each node in Q, so it takes time $O(q^2 \rho \log^3 n)$ to perform q limited broadcasts, one from each node in Q.

Once the limited broadcasting portion of the protocol is complete, each message originating from a node in Q is now in at least ρ nodes. SETCOVERFLUSH() will repeatedly find the node in the network with the most messages and broadcast from that node, flushing the messages contained in this broadcast. A message is *flushed* once all nodes in the network have received it, and flushed messages will not be counted in a node's message count during the FINDMAX portion of SETCOVERFLUSH().

With all messages in at least ρ nodes, SETCOVERFLUSH() will perform at most $O((n/\rho)\log n)$ broadcasts. On each broadcast, it takes time $O(n \log^2 n \log\log n)$ for FINDMAX and it takes an additional time $O(n \log n \log\log n)$ for the actual broadcast. Therefore, SETCOVERFLUSH() takes time $O((n^2/\rho)\log^3 n \log\log n)$ to flush all q messages from the network. After SETCOVERFLUSH() completes, all messages from Q have been flushed so STRONGMB(q, n) completes multi-broadcasting in time $O(q^2\rho \log^3 n + (n^2/\rho)\log^3 n \log\log n)$. For $\rho = (n/q)\log^{1/2}\log n$, STRONGMB$(q, n)$ runs in time $O(qn \log^3 n \log^{1/2} \log n + qn \log^2 n \log^{1/2}\log n) = O(qn \log^3 n \log^{1/2}\log n)$.

□

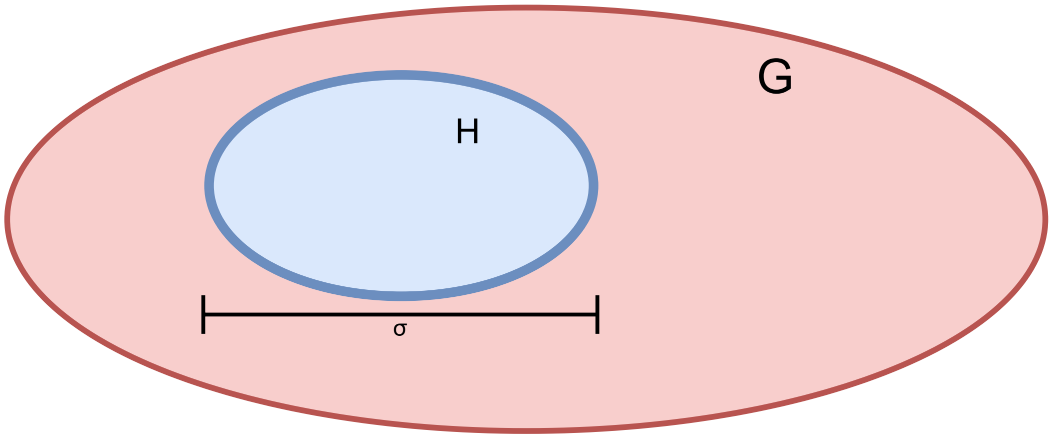

Figure 4.1: A model where $H \subset V$ has diameter σ. All nodes of Q are contained within H.

4.3 Q With Diameter σ, $|Q|$ is known

We next consider a model where both the size of Q is still known, the members of Q remain unknown, but we know that the max distance between any two nodes in Q is at most σ. Figure 4.1 above depicts this model, where $Q \subset H$. Nodes of Q can be located anywhere within H, and it is important to note that H may have nodes that do not actually have messages. Although they are not members of Q, these nodes can act as intermediate nodes in broadcasts initiated by nodes in Q.

By using the bounded-radius broadcasting protocol from Chapter 3, we can first gather all q messages into all nodes of Q. We will again facilitate these bounded-radius broadcasts with a (q, q, n)-selector of size $m = O(q^2 \log n)$, iterated over in m phases, with each phase length equal to a single call of $\textsc{BrBroadcast}(\Delta, \sigma)$. Once all bounded-radius broadcasting is complete, all nodes of Q will initiate full broadcasting to complete multi-broadcasting. Since all broadcasts carry the same messages, this is equivalent to a single broadcast from any node in Q. We assume $q^2 \leq \Delta$, otherwise we can use σ repetitions of a (Δ, Δ, n)-selector as described in [10]. Formally, the protocol works as follows:

Protocol $\textsc{BdMB}(\sigma, q)$:

for $i = 1, 2, \ldots, m$

– **during** Phase i

—– **if** $v \in S_i$, v initiates $\textsc{BrBroadcast}(\Delta, \sigma)$.

all nodes in Q initiate broadcast

Theorem 4.3. $\textsc{BdMB}(\sigma, q)$ *completes multi-broadcasting in networks with a bounded diameter on* Q *in time* $O(q^2 \sigma \Delta \log^2 n \log \Delta)$.

Proof. Each execution of $\textsc{BrBroadcast}(\Delta, \sigma)$ takes time $O(\sigma \Delta \log n \log \Delta)$, as shown in Theorem 3.1, and using a (q, q, n)-selector will take time $O(q^2 \log n)$ to initiate $\textsc{BrBroadcast}(\Delta, \sigma)$ from each node in Q. After this initial phase, all nodes in Q have all q messages, thus a single broadcast from any or all nodes in Q will complete multi-broadcasting. □

4.4 Doubling to Find $|Q|$

In Sections 4.2 and 4.3, we described protocols that rely on knowledge of the size of the set Q, but not the actual members of Q. In this section, we describe a protocol that can repeatedly call a multi-broadcasting procedure with geometrically increasing values of q' passed as a parameter for q until the first instance of $q' > q$ is found. This protocol leverages the concept of *informing by silence* [3] where nodes can interpret the lack of transmission as additional knowledge. Let $P(n, q)$ be a protocol that successfully performs multi-broadcasting in time $t_{P,q} = O(P(n, q))$ if a value $q' \geq q$ is passed, and has unknown performance if $q' < q$ is passed. We can start from $q' = 1$ and repeatedly double q' until $q' \geq q$ is found.

Protocol Doubling(q'): all nodes in Q start counting from time $t = 0$ and at the same time, $P(n, q')$ is initiated. If any node $v \in Q$ counts to $t = t_{P,q'}$ without transmitting, then v broadcasts a failure signal. The protocol then initiates a call to Doubling$(2q')$. If no node broadcasts a failure signal, then the protocol terminates.

Theorem 4.4. Doubling(q') *will successfully determine the correct value of* q *within* $\lceil \log n \rceil$ *calls.*

Proof. Although the nodes in the network are unaware of the individual members of Q, each node itself knows whether or not it is part of Q by virtue of whether or not it has a message to transmit. For a successful multi-broadcasting operation to occur, all nodes in Q must transmit at least once, otherwise some message in Q must have never left its source node. Therefore, if after $t_{P,q'}$ time steps, some node has not transmitted then it knows a failure has occurred and this node will broadcast a failure signal to the rest of the network. If multiple nodes broadcast a failure signal, collisions are mitigated as the same exact message is being transmitted and can be treated as a single transmission. Thus, after $t_{P,q'}$ steps, the protocol will know whether to continue calling $\textsc{Doubling}(2q')$ or terminate the protocol. Since the previous value of q' is geometrically increasing in each call, the correct value of q will be found after $\log q$ calls. Since $q < n$, at most $\lceil \log n \rceil$ calls will occur. This completes the proof of Theorem 4.4. □

If we are not aware of a diameter bound on the nodes of Q, then $P(n, q) = \textsc{StrongMB}(q, n)$. If we are aware of a diameter bound, σ, on Q, then $P(n, q) = \textsc{BdMB}(\sigma, q)$. In either case, we can drop the requirement on the knowledge of q because we can determine it using $\textsc{Doubling}(q)$ which increases the runtime of $\textsc{StrongMB}(q, n)$ or $\textsc{BdMB}(\sigma, q)$ asymptotically by a factor of $O(\log n)$.

Chapter 5

Conclusions

In Chapter 3, we presented a bounded-radius broadcasting algorithm for radio networks with a maximum in-degree of Δ. We proved that BrBroadcast(Δ, σ) will successfully disperse a message from a source node v to all nodes within distance σ of v in time $O(\Delta\sigma \log n \log \Delta)$.

In Chapter 4, we discussed the challenge of multi-broadcasting under three different models: (i) A network where both Q and $|Q|$ are known, (ii) A network where only $|Q|$ is known, and (iii) A network where $|Q|$ is known and Q has a known diameter σ. We proposed a multi-broadcasting protocol for each model and proved the correctness and runtime of each protocol. For (i), we showed that NaiveMB(Q, n) completes multi-broadcasting in time $O(qn \log n \log \log n)$. For (ii), we showed that StrongMB(q, n) completes multi-broadcasting in $O(qn \log^3 n \log^{1/2} \log n)$. For (iii) we showed that BdMB(σ, q) completes multi-broadcasting in time $O(q^2\sigma\Delta \log^2 n)$.

Additionally, we described and proved a procedure that can find the correct value of $|Q| = q$ in $O(\log n)$ time so knowledge of $|Q|$ is not necessary under models (ii) and (iii) at this additional $O(\log n)$ cost. For all full broadcasts, we utilize the protocol described in [8] which runs in time $O(n \log n \log \log n)$.

In multi-broadcasting, several problems remain open due to the relatively unstudied nature of this problem. For one, under the simpler models mentioned in Chapters 4.1 and 4.2, it would be interesting to see an adaptation that allows for larger values of q. Additionally, one can observe that for $k \approx n^{2/3}$, the full gossiping algorithm in [10] uses at most $O(n/k)$ broadcasts. With only $q < n$ nodes containing messages, we suspect that it may be possible to perform multi-broadcasting with $O(q/k)$ broadcasts, possibly with a slight refactoring of k. If such an algorithm exists, it will beat the $\tilde{O}(n^{4/3})$ full gossiping algorithm for any value of $q < n$ under any ad-hoc radio network model.

In bounded-radius broadcasting, there are two main problems that remain open: The first is to perform bounded-radius broadcasting efficiently in radio networks without a bound on maximum in-degree, Δ. Also, it would be interesting to close the gap between our protocol, BrBroadcast(Δ, σ), and the runtime of the full broadcasting in [6]. Our bounded-radius broadcast protocol was derived from the full broadcasting protocol described in [3], using stages of interleaved selectors. For networks with in-degree bounded by Δ, it seems possible that one could complete bounded-radius broadcasting in time roughly equal to $O(\sigma \Delta \log n \log \log \Delta)$.

Bibliography

[1] Bogdan S. Chlebus, Leszek Gasieniec, Alan Gibbons, Andrzej Pelc, and Wojciech Rytter. Deterministic broadcasting in ad hoc radio networks. *Distributed Computing*, 15(1):27–38, 2002.

[2] Bogdan S. Chlebus, Leszek Gasieniec, Alan Gibbons, Andrzej Pelc, and Wojciech Rytter. Deterministic broadcasting in unknown radio networks. page 861–870, 2000.

[3] Marek Chrobak, Leszek Gasieniec, and Wojciech Rytter. Fast broadcasting and gossiping in radio networks. *Journal of Algorithms*, 43(2):177–189, 2002.

[4] Marek Chrobak, Leszek Gasieniec, and Wojciech Rytter. A randomized algorithm for gossiping in radio networks. *Networks*, 43(2):119–124, 2004.

[5] Andrea E. F. Clementi, Angelo Monti, and Riccardo Silvestri. Distributed broadcast in radio networks of unknown topology. *Theor. Comput. Sci.*, 302(1-3):337–364, 2003.

[6] Artur Czumaj and Peter Davies. Faster deterministic communication in radio networks. In *Proc. 43rd International Colloquium on Automata, Languages, and Programming (ICALP'16)*, pages 139:1–139:14, 2016.

[7] Annalisa De Bonis, Leszek Gasieniec, and Ugo Vaccaro. Generalized framework for selectors with applications in optimal group testing. In *Automata, Languages and Programming*, pages 81–96, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[8] Gianluca De Marco. Distributed broadcast in unknown radio networks. *SIAM Journal on Computing*, 39:2162–2175, 2010.

[9] Gianluca De Marco and Andrzej Pelc. Faster broadcasting in unknown radio networks. In *Information Processing Letters*, pages 53–56, 2001.

[10] Leszek Gasieniec, Tomasz Radzik, and Qin Xin. Faster deterministic gossiping in directed ad hoc radio networks. In *Proc. Scandinavian Workshop on Algorithm Theory (SWAT'04)*, pages 397–407, 2004.

[11] Leszek Gasieniec and Andrzej Lingas. On adaptive deterministic gossiping in ad hoc radio networks. *Information Processing Letters*, 83(2):89–93, 2002.

[12] Dariusz R. Kowalski and Andrzej Pelc. Leader election in ad hoc radio networks: A keen ear helps. *Journal of Computer and System Sciences*, 79(7):1164–1180, 2013.

[13] Ying Xu. An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks. *Algorithmica*, 36(1):93–96, 2003.