

SINR-Based Scheduling for Minimum Latency Broadcast

Clement Kam*, Sastry Kompella*, Anthony Ephremides[†] and Ira S. Moskowitz*

*Information Technology Division, Naval Research Laboratory, Washington, DC 20375

[†]Electrical and Computer Engineering Department, University of Maryland, College Park, MD 20742

Abstract—We study the minimum latency broadcast scheduling problem, in which a single source has a quantity of data that must be transmitted to all other nodes in a multi-hop network in minimum time. Aside from the obvious application to classical communications, this problem also relates to some more general problems in the field of network science. Previous approaches to scheduling have assumed a simplistic collision model of interference, while others have studied the more realistic physical model of total received interference power. Existing suboptimal approaches for transmitting the data typically assume a collision-free, fixed-rate, single packet transmission. In this work, we devise an optimal approach for broadcast under a physical interference model with fixed-rate, single packet transmission by converting it to a shortest path problem for an unweighted, undirected graph. Since this optimal approach does not scale well, we also consider a suboptimal layered approach which separates the routing and scheduling functions, but relaxes the fixed-rate, single packet assumption. This goes beyond the signal-to-interference-plus-noise (SINR) threshold model to allow for rate adaptation as a function of SINR. We include improvements on previous routing approaches, and we formulate a linear programming approach to the variable-rate scheduling for broadcast. Simulations show that in some special cases, this variable-rate layered approach can even outperform the optimal fixed-rate, single packet approach.

I. INTRODUCTION

Network-wide broadcast is understood to be a fundamental operation in ad hoc networks, necessary in communication protocols such as route or service discovery, as well as information dissemination for military surveillance or emergency disaster relief. Latency is an important measure of performance for these applications and should be minimized to streamline the user experience. In some applications, a simple flooding algorithm may be sufficient, but in other cases, it may be necessary to optimize the broadcast schedule at time scales finer than that which a human operator can perceive, for applications in which every fraction of a second matters. For example, a network may require a fast triggered automated response to defend itself against cyber attacks, so that all other nodes in the network can be made aware of the attack in minimum time. This problem is of interest to the field of network science, for minimizing the effectiveness of attacks in some large, complex network. The present work provides some insight into this problem by proposing a complex model of data broadcast through a wireless network, and then decomposing the model for reduced complexity.

Early studies on minimum latency broadcast focused on collision-based models of interference, where the network

is represented as a graph and simultaneous reception from different neighbors leads to a collision. The broadcast problem for this graph-based approach has been shown to be NP-hard, both for general [1] and unit disk graphs [2], and many approximation algorithms have been proposed (see [3] and references therein). The problem has also been studied under the more realistic *physical interference model* as defined in [4], which assumes a successful transmission provided that the signal-to-interference-plus-noise-ratio (SINR) is above some threshold. In Huang et al [5], the authors propose tiling the plane on which the network lies to construct a backbone network, and coloring the tiles so that backbone nodes in the same colored tiles can transmit simultaneously without too much interference. This tiling approach has been used in other works to manage interference [6], [7], but tiling may be overly conservative in avoiding interference, depending on the exact location of the nodes. Also, these approaches did not consider the possibility of rate adaptation, in which a higher SINR allows for a potentially higher rate transmission. Rate adaptation was considered in [8], where the tradeoff was between transmitting at a higher power/rate and having more simultaneous transmissions at lower power (less interference). The optimal approach was shown to be NP-hard, and a heuristic approach was developed under the less realistic *protocol interference model* [4], in which non-receiving nodes within some interference range cannot receive from their intended transmitter. None of the above works consider the physical interference model with rate adaptation.

Our contributions in this work are as follows. First, we solve for the optimal fixed-rate minimum latency broadcast under the physical model of interference, based on framing the schedule as a path in a state diagram. For a basic network, we compute the complexity of solving for the shortest path of its state diagram, which yields the minimum latency schedule, and the computational complexity is shown to be exponential in the number of nodes. Second, due to the poor scalability of the optimal fixed-rate approach, we consider a layered approach that first generates a routing tree and then schedules the (identical) data from one layer of the tree to the next, but rate adaptation is now allowed. We propose a novel routing enhancement (re-layering), and we adapt the column generation approach used in optimal link scheduling approaches [9], [10] to solve the layer-by-layer broadcast schedule. Simulations are provided to compare the performance of the two approaches.

II. BROADCAST NETWORK MODEL

The networks we consider in this work are modeled as geometric networks, such that the set of nodes in a network is represented by a finite point set V in a two-dimensional plane. We assume all nodes have a uniform transmission power K . For the analysis in this work, we assume a simple channel model that only assumes path loss, but extension to other channel models (e.g., fading) is straightforward. The received power at node v from a transmission from node u is given by $\Phi(v, u) = \frac{K}{d^{\alpha}(v, u)}$, where α is the path loss exponent and $d(v, u)$ is the Euclidean distance between nodes u and v . We assume a physical interference model, where reception at a node depends on the Signal-to-Interference-plus-Noise-Ratio (SINR) at node v :

$$\text{SINR}(v, u, \mathcal{S}) = \frac{\Phi(v, u)}{\gamma + \sum_{w \in \mathcal{S}} \Phi(v, w)} \quad (1)$$

where γ is the background noise, and \mathcal{S} is the set of senders transmitting simultaneously with node u . A single source node $s \in V$ is in possession of a packet at time 0, and the goal is to broadcast the packet from s to all other nodes in V in minimum time. It is possible that some nodes in V cannot be reached due to limitations of the transmission range, so we are only concerned with those that can be reached, potentially over multiple hops, from s . We call those nodes *connected* nodes.

III. OPTIMAL FIXED-RATE BROADCAST

We begin by studying a system in which the data rate of transmissions is fixed, and the data is transmitted as a single packet. Transmissions are considered successful if the SINR (1) is greater than some threshold β . In the absence of any other transmitters, this threshold defines the maximum transmission range of a node, which can be used to determine the connectivity between nodes. Given this setup, the schedule length can be viewed as the number of slots needed to broadcast the packet from the source to all other nodes.

To represent the possible broadcast schedules, we construct a graph, which we call a *virtual graph*, in which the vertices represent network states that correspond to the set of nodes that are in possession of the packet. Directed edges between vertices in the virtual graph are generated as follows: for some previous state, a directed edge to a next state exists when the set of nodes with the packet in the next state includes those in the previous state, plus nodes that could be added via one slot of transmission from the original previous state's nodes. Those added nodes can be determined by selecting a set of transmitting nodes (from those in the original previous state) and computing the SINR at all of the other nodes to see if reception was successful.

An example of this kind of virtual graph is represented in Fig. 1. For simplicity, we use a graph representation of a network and assume a collision model of interference. Specifically, a transmitting node has its packet received by its one-hop neighbors in the graph error-free, provided no other nodes are transmitting to the same neighbor at the same time.

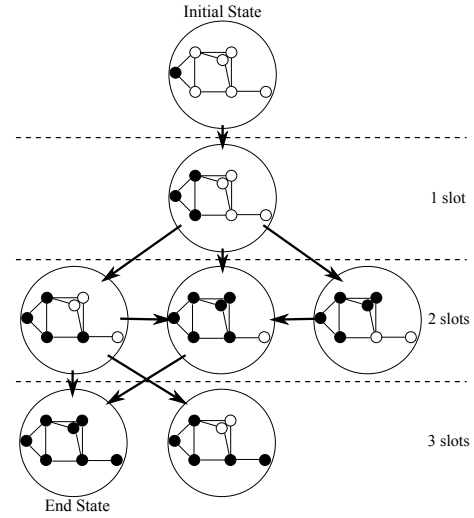


Fig. 1. Virtual graph for optimal fixed-rate approach.

In Fig. 1, the black nodes are in possession of the packet, and the white nodes have not yet received the packet. The virtual graph in the figure is organized in breadth-first-search (BFS) order, so that the states at the same level can be reached in the same minimum number of slots (but more slots are possible). Any path through the virtual graph starting from the initial state (where only the source has the packet) to the end state (where all connected nodes have the packet) is the sequence of states that results from some valid schedule. The shortest length path through the virtual graph is equal to the shortest schedule for broadcasting to all nodes. The shortest path for this unweighted graph can be found using a breadth-first-search (BFS) algorithm. Thus, for the example in Fig. 1, the shortest path is of length 3 slots.

A. Graph Construction/Shortest Path Algorithm

The straightforward approach to the general broadcast scheduling problem would be to construct the virtual graph in its entirety (now using the SINR criterion), and then running a shortest path algorithm to find the minimum schedule length. However, it is not necessary to construct the entire virtual graph to find the shortest schedule, but rather the virtual graph can be constructed on the fly in conjunction with the BFS shortest path algorithm (optimal for unweighted graphs). For the example in Fig. 1, the states can be generated left-to-right, starting from the top (i.e., in BFS order), until reaching the end state, and stopping before reaching the last state in the figure. This approach exhaustively generates all states resulting from schedules of length 1 before moving to length 2, where all states from schedules of length 2 are generated before moving to length 3, etc. Therefore, once the end state is generated, the algorithm can terminate, and the minimum length schedule is found.

The algorithm is presented in Fig. 2 and summarized as follows. First, the connectivity of nodes is determined using the SINR threshold β in the absence of any other transmitters.

```

1: Initialize:
2: Generate graph of connectivity in the network, with neighbors based on SINR threshold  $\beta$ 
3: Initialize ordered array of states with State 1 (only the source in Received List (RL))
4: Initialize corresponding array of distances with  $d(1)=0$ 
5: Add to array State 2 (add source's 1-hop neighbors in RL),  $d(2)=1$ 
6:  $n \leftarrow 2$ 
7: while Not all connected nodes in State  $n$ 's RL and  $n$  not at end of array do
8:   for All combinations of nodes in State  $n$ 's RL that have neighbors not in RL do
9:     for All neighbors not in RL do
10:      if Neighbor's SINR exceeds threshold  $\beta$  then
11:        Add neighbor to temporary RL
12:      end if
13:    end for
14:    if Resulting state after adding temporary RL to RL does not exist in the array then
15:      Add new state to the end of the array
16:       $d(\text{end of array})=d(n)+1$ 
17:    end if
18:  end for
19:   $n \leftarrow n + 1$ 
20: end while

```

Fig. 2. Graph Construction/Shortest Path Algorithm

Starting with the initial state (only the source has the packet), states are generated for each possible combination of simultaneously transmitting nodes (called a transmission set) with neighbors that do not have the packet. Based on the resulting state for each of those transmission sets, a new state is added to the virtual graph if it has not already been generated. As the states are generated, they are placed in an ordered array. The states in the array are processed in order, which coincides with a breadth-first traversal of the virtual graph. Once all of the connected nodes have the packet, the algorithm terminates, and the distance measure of the end state is equal to the minimum schedule length in slots. The schedule can be determined by maintaining a record of transmission sets that lead to each state and traversing the breadth-first-search tree backwards from the end state to the source.

B. Algorithm Complexity

For a general network geometry, it is difficult to analyze the computational complexity of the algorithm above. To get an idea, we consider a “star” topology for the network, in which the source at the center has m multi-hop branches extending outward from the center, each branch having an equal number of nodes in tandem (hop length is equal to the transmission range), and the branches do not interfere with one another. This is possible for an arbitrary number of branches even for a two-dimensional network since, under the assumptions of our model, the hop length can be made arbitrarily small to

avoid interference. For a network with n destination nodes, where n is a multiple of m , the number of states that must be traversed is at least $(n/m - 1)^m$, which is the number of combinations of nodes that can have the packet in each branch, excluding the outermost nodes. For each of these states, there are $2^m - 1$ possible transmission sets, and the number of potential receivers to be checked is equal to m (Line 10 in the algorithm). The resulting complexity is at least equal to $m(2^m - 1)(n/m - 1)^m$. If, for example, we fix the number of nodes in each branch to be 3, we have $m = n/3$. The complexity in this case is at least equal to $(n/3)(2^{n/3} - 1)2^{n/3}$, which is exponential in n . Therefore, the algorithm does not scale well, even for this “star” network, where each node has very few neighbors and interferers.

IV. LAYERED RATE ADAPTATION BROADCAST

As a computationally more efficient alternative to the optimal fixed-rate broadcast, we consider a layered approach¹, in which we first construct a routing tree for the network and divide the tree into layers. Then we schedule transmissions layer-by-layer, by constructing a schedule where each node in layer a has identical data, and layer a needs to transmit this data to each node in layer $a + 1$. When this transmission is completed, each node in layer $a + 1$ has the identical data, and only then is it ready to begin transmitting to layer $a + 2$. Optimality is sacrificed in this approach by determining the routing tree independent of the schedule, as well as scheduling the layers in a serial manner. However, there is also room for gain over the previous fixed-rate approach, in that we now allow for rate adaptation based on SINR in the scheduling optimization between layers.

A. Routing

The first step in the layered approach is to construct a routing tree rooted at the source. Previous approaches for minimum latency broadcast typically use a breadth first search (BFS) tree and assign each node to a layer in the tree, where each layer is determined by its hop distance from the source. We also use this approach, but with some enhancements. To determine the connectivity between nodes, we first assume that in the absence of interference, there is some maximum transmission range such that any pair of nodes within range can communicate successfully. For the BFS tree, we say that the source node is in layer 0, and its one-hop neighbors, are in layer 1. The source node is the parent node of the layer 1 nodes, which are, accordingly, the source node's children. For each node in layer 1, we say its one-hop neighbors are in layer 2 if they are not already part of the tree, and the parent-child assignment between layer 1 and 2 occurs according to the ordering of nodes in the search. The process repeats for layer 2, layer 3, etc. until all of the nodes are in the tree.

¹The approach is “layered” in the sense that the routing tree is divided into layers, and also that the routing and scheduling functions occur in separate layers.

1) *Re-assigning parents*: When layers and parent-child relationships are assigned, it is possible that a node in one layer may be geographically closer to a node in the layer above other than its assigned parent, since the BFS only considers connectivity due to the maximum transmission range. To create a better assignment for each pair of adjacent layers, we assign to each child node the parent in the layer above that is of minimum distance from it.

2) *Re-layering*: By doing a BFS-based layering, nodes are likely to be in interference range of other nodes in the layer directly above, lowering the achievable rates via simultaneous transmission. To create more separation between nodes transmitting simultaneously, we introduce a re-layering algorithm that allows the layers to be more staggered without hindering the propagation of data. Our re-layering algorithm maintains the same tree structure for fast propagation, but some nodes are reassigned to lower layers with which to transmit, thus increasing the separation between nodes in the BFS layer. Note that the order of re-assigning parents and re-layering matters, and in our approach, we first re-assign parents to get the more desirable tree structure before re-layering.

When re-layering, we maintain separate labels for transmitting nodes and receiving nodes, since a node may receive from layer 0, for example, and may wait until layer 2 to transmit. We first define the *transmit layering* as follows. Starting with the nodes in the last original layer (layer 3 in Fig. 3), we label those and their direct ancestors above as in the original tree. For the layer above the last (e.g., layer 2), we label any leaf nodes (nodes without children) the same as the last transmitting layer (e.g., 3), and count down as we label the direct ancestors up the tree, and we stop when an already-labeled ancestor is reached. This process is repeated with the next higher layer of leaf nodes, until all nodes are labeled. The result is that the last transmit layer (even though it never transmits) is all of the leaf nodes, the next to last transmit layer contains all the parents of the leaf nodes (if not belonging to some longer branch), etc. For the *receive layering*, the labels come directly from the transmit layering and the parent-child relationships that have been previously established. The receive layer of a receiving node is equal to its parent's transmit layer plus one, i.e., if the parent is in transmit layer a , the child is in receive layer $a + 1$. When it comes time to schedule the layer-by-layer transmission from a to $a + 1$, layer a refers to the transmit layer, and layer $a + 1$ is the receive layer. Note that the total number of layers is not changed in the re-layering process.

B. Scheduling Formulation

After establishing the routing structure, we schedule the transmission for each pair of layers. The nodes in transmit layer a have identical data and need to propagate this data to the nodes in receive layer $a + 1$, none of which have the data. In the following, we will drop the *transmit* and *receive* terms and simply refer to them as layer a and layer $a + 1$. In contrast to the optimal fixed-rate approach, we relax the fixed rate and fixed time slot duration assumptions and allow

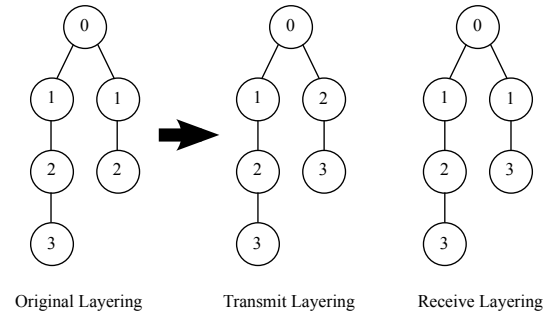


Fig. 3. Re-layering example.

variable rates. Our goal for each pair of layers is to minimize the total time duration required to broadcast the data to layer $a + 1$. The total schedule length is the sum of the broadcast time durations for all pairs of layers in the network. We again assume that each node has a fixed transmission power, and the maximum rate that we allow at some receiver v from transmitter u in the presence of transmission set \mathcal{S} is given by $R = W \log_2(1 + \text{SINR}(v, u, \mathcal{S}))$, where W is the bandwidth. A node u transmitting at rate R can be successfully received by some node v provided $\text{SINR}(v, u, \mathcal{S}) \geq 2^{R/W} - 1$. Again, we assume a distance-based path loss for the channel model (i.e., no fading).

In previous link scheduling approaches, different combinations of links (called *matchings*) are activated simultaneously, with each link transmitting at a rate determined by the given matching. The goal is to determine the time duration that each matching needs to be activated to transmit all of the data, while minimizing the total time duration. We adapt this approach for the broadcast scheduling problem, in which each transmitter may have multiple children destination nodes, so it can choose to transmit to all of them at one low rate, or it can choose to transmit to a subset of its children within some closer distance, at a higher rate. We therefore have multiple activations that we call *rate-matchings*, which are determined by the combination of transmitting nodes and their intended receiving children.

Given the rate-matchings, we can formulate the scheduling problem as a linear program in a manner similar to that of the link scheduling problems [9], [10]:

$$\begin{aligned}
 & \min. && \sum_{k \in \mathcal{M}} t^k \\
 & \text{subject to} && \sum_{k \in \mathcal{M}} r_j^k t^k \geq V \quad \forall j \in \mathcal{D} \\
 & && t^k \geq 0
 \end{aligned} \tag{2}$$

where \mathcal{D} is the set of child nodes in layer $a + 1$, \mathcal{M} is the set of all possible rate-matchings, t^k is the time duration of activation of the k th rate-matching, r_j^k is the rate of the j th child node in \mathcal{D} for the k th rate-matching, and $V > 0$ is the size of the packet to be received. Since this linear program is a large problem with as many as $2^{|\mathcal{D}|}$ rate-matchings, we resort to a reduced-complexity algorithm known as *column*

generation to generate solutions with good performance.

C. Column Generation Scheduling Algorithm

Column generation is an iterative algorithm for solving large linear programming problems. It has been applied in many scheduling algorithms, and in our problem, each iteration can be broken into two parts: an *activation duration module* and a *group generation module* [10]. The activation duration module first solves a reduced linear program (using a subset of rate-matchings to reduce the size of the constraint matrix and cost function) to get activation times that are optimal for the reduced set. Then a metric based on the dual variables is fed to the group generation module, which selects a new rate-matching using the metric. This new rate-matching is added to the linear program as a column in the constraint matrix (and in the cost function), and the next iteration starts with the activation time module with the new augmented linear program.

Similar to the “CG-Heuristic” approach in [10], we apply column generation to the broadcast scheduling problem in the following manner. We start by formulating the reduced linear program, where only the rate-matchings with one transmitter active are included. We call this set \mathcal{M}^1 . For each transmitter, a separate column (corresponding to a rate-matching) in the constraint matrix is included for each child node, with a different subset of children receiving at the rate of the farthest child in that subset.

As an example, we consider two transmitters (A, B), each having two children ($A1, A2; B1, B2$) at different distances ($\text{dist}(A, A1) < \text{dist}(A, A2); \text{dist}(B, B1) < \text{dist}(B, B2)$). The constraints for the single-activation rate matchings are given by

$$\begin{aligned} r_{A1}^1 t^1 + r_{A2}^2 t^2 &\geq V \\ r_{A2}^2 t^2 &\geq V \\ r_{B1}^3 t^3 + r_{B2}^4 t^4 &\geq V \\ r_{B2}^4 t^4 &\geq V \end{aligned}$$

For child node $A1$, the highest rate at which it can receive is equal to r_{A1}^1 , and $A2$ cannot receive at such a rate because its SINR is not high enough. Node $A1$ can also receive at the lower rate r_{A2}^2 , which is the highest rate at which $A2$ can receive. It turns out that the solution of this initial problem has $t^1 = t^3 = 0, t^2 = V/r_{A2}^2, t^4 = V/r_{B2}^4$. This corresponds to only sending to the farthest child of each transmitter. This result can be understood intuitively: for A (B) to complete transmission to $A2$ ($B2$), it can transmit for t^2 (t^4) to offload all of the data, but in the process, $A1$ ($B1$) receives the complete packet as well, so rate-matching 1 (3) does not need to be activated. Although the rate-matchings 1 and 3 are not necessary to the initial solution, they may be needed in later iterations when more rate-matchings are added, since they may provide a complementary rate-matching that best satisfies the constraints.

For the group generation step, a search-based algorithm can be used to find the optimal rate-matching, but here we

- 1: **Initialize:**
- 2: Rank the children nodes in descending order of associated dual variables ω
- 3: Choose child node 1 to be in the rate-matching m , compute $\nu = \omega_1 \cdot r_1(\text{SNR})$
- 4: **for** All remaining children nodes **do**
- 5: **if** Child is farther from its parent than any of its siblings in m **then**
- 6: Compute $\tilde{\nu} = \sum_{c \in m} \omega_c \cdot r_c(\text{SINR}_m)$
- 7: **if** $\tilde{\nu} > \nu$ **then**
- 8: Add child node c to rate-matching m
- 9: $\nu \leftarrow \tilde{\nu}$
- 10: **end if**
- 11: **end if**
- 12: **end for**

Fig. 4. Rate-Matching Generation Algorithm

simply demonstrate the approach using the greedy heuristic in Fig. 4 to iteratively choose the activations. The algorithm first ranks the child nodes according to their associated dual variables from the linear program. The first-ranked child node is included in the rate-matching m at its interference-free rate ($r_1(\text{SNR})$), and a metric ν is computed. Each node is considered sequentially in ranked order as a candidate to be added to the rate-matching. If there is a sibling node (shares same parent) already in m that is farther from their parent, the candidate node is skipped since it is already covered. Otherwise, the metric is recomputed, taking into account new levels of interference in the rates. If the metric is improved, then the child node gets added to the rate-matching m . To avoid highly suboptimal rate-matchings from the heuristic, we re-run the rate-matching algorithm but start with the second-ranked child node, and again starting with the third-ranked child node. Of the three resulting rate-matchings, we select the one with the largest final metric.

V. SIMULATION

We evaluate our layered broadcast scheduling algorithm via simulation over randomly placed networks in a two-dimensional square deployment area. We assume a fixed transmission power of $K = 23$ dBm, noise level $\gamma = -30$ dBm, path loss exponent $\alpha = 3$, and maximum transmission range equal to 78 m. The power, noise, and transmission range parameters are derived from the field test results shown in [8], and the nodes are uniformly randomly placed in the square area according to some constant node density.

A. Performance of Layered Approaches

To evaluate the effect of the various routing enhancements proposed, we simulate the broadcast scheduling for various deployment area sizes. We set the node density to $\lambda = 10^{-3}$ nodes/m² for sufficient connectivity. The results are shown in Fig. 5. We note that re-parenting yields a shorter schedule length in all cases, since it simply improves on the BFS-tree assignment that is based only on connectivity and blind

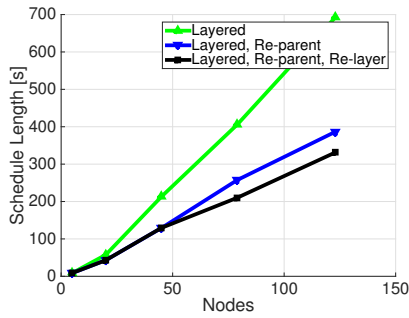


Fig. 5. Schedule length of layered approaches vs. number of nodes, fixed node density.

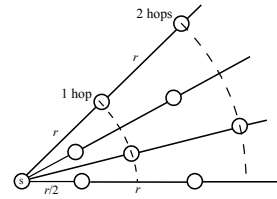


Fig. 6. Circular sector network.

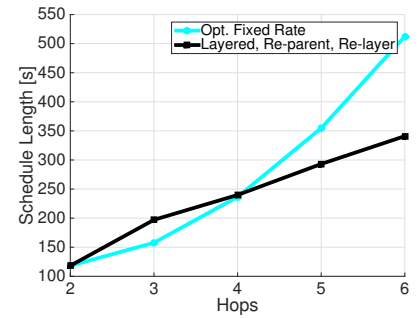


Fig. 7. Comparison of optimal fixed-rate and layered rate adaptation approaches.

to parent-child distances. We also see that the re-layering improves the schedule length more as the deployment area increases, since the links in the new layers will be more spread out, taking greater advantage of the spatial reuse. The combined improvement is about 50% across the board.

B. Fixed-Rate vs. Rate Adaptation

To simulate the optimal fixed-rate approach, a smaller number of nodes was used to generate results, due to the computational complexity. To vary the node density and vary the number of hops in the network for up to 24 nodes, we generated a circular sector-shaped network with an angle of $\pi/4$ radians, shown in Fig. 6. The source is located at the narrow point of the sector, and we place nodes along four branches spread across the sector. We build the network hop by hop, where each hop after the first places nodes within the transmission range r of the previous hop's nodes. Each hop includes four nodes, and the nodes are alternately staggered by $r/2$ to more evenly distribute the nodes over each hop.

We plot the performance of the optimal fixed-rate approach and the layered rate adaptation approach for this network in Fig. 7. The SINR threshold for the optimal fixed-rate approach is chosen to be $\beta \approx 0.42$, corresponding to a maximum transmission range of 78 m. Also for the fixed-rate approach, we choose a rate equal to $W \log_2(1 + \text{SINR}_{\min})$, where SINR_{\min} is the minimum SINR over all correctly received packets in the optimal schedule. For a smaller number of hops, the network is denser and the optimal fixed-rate approach outperforms the layered approach, but as more hops are added, the layered approach can take advantage of the rate adaptation, resulting in a shorter schedule. There are a couple of factors that are responsible for this result, namely the optimal approach's unlayered flexibility, as well as the layered approach's adaptability of the rate to the SINR (as opposed to the optimal approach's hard SINR constraint).

VI. CONCLUSION

In this paper, we have looked at the minimum latency broadcast problem with interference considerations, with applications to protocol functionality as well as to more general problems considered in network science. We have studied two different scheduling approaches, namely a fixed-rate optimal approach and a layered rate adaptation approach. We derived

the fixed-rate optimal approach by constructing a virtual graph of network states and finding the shortest path. Due to the high computational complexity of the fixed-rate approach, we developed a layered approach that divides the nodes into layers and schedules transmissions from one layer to the next, allowing for rate adaptation based on SINR. We introduced a novel re-layering enhancement and show approximately 50% improvement over the typical approach. Lastly, we have compared the optimal fixed-rate approach and the layered approach and showed that the layered approach performs comparably well. Furthermore, the rate adaptation in the layered approach is shown to be effective enough in some cases to outperform the optimal fixed-rate approach, particularly for sparse, multi-hop networks.

REFERENCES

- [1] I. Chlamtac and S. Kutten, "On broadcasting in radio networks—problem analysis and protocol design," *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1240–1246, Dec 1985.
- [2] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, 2003, pp. 222–232.
- [3] S. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, "Minimum-latency broadcast scheduling in wireless ad hoc networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE, May 2007, pp. 733–739.
- [4] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.
- [5] S.-H. Huang, P.-J. Wan, J. Deng, and Y. S. Han, "Broadcast scheduling in interference environment," *IEEE Transactions on Mobile Computing*, vol. 7, no. 11, pp. 1338–1348, Nov 2008.
- [6] P.-J. Wan, L. Wang, and O. Frieder, "Fast group communications in multihop wireless networks subject to physical interference," in *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, Oct 2009, pp. 526–533.
- [7] T. Jurdzinski, D. R. Kowalski, and G. Stachowiak, "Distributed deterministic broadcasting in wireless networks of weak devices," in *Automata, Languages, and Programming*. Springer, 2013, pp. 632–644.
- [8] Y. Chang, Q. Liu, and X. Jia, "A data rate and concurrency balanced approach for broadcast in wireless mesh networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3556–3566, July 2014.
- [9] S. Kompella, J. E. Wieselthier, A. Ephremides, H. D. Sherali, and G. D. Nguyen, "On optimal SINR-based scheduling in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1713–1724, Dec. 2010.
- [10] V. Angelakis, A. Ephremides, Q. He, and D. Yuan, "Minimum-time link scheduling for emptying wireless systems: Solution characterization and algorithmic framework," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1083–1100, Feb 2014.