Dynamic Multiple-Message Broadcast: Bounding Throughput in the Affectance Model

Dariusz R. Kowalski
University of Liverpool
Dept. of Computer Science
Liverpool, UK
D.Kowalski@liv.ac.uk

Miguel A. Mosteiro
Kean University
Dept. of Computer Science
Union, NJ
mmosteir@kean.edu

Tevin Rouse
Kean University
Dept. of Computer Science
Union, NJ
rousete@kean.edu

ABSTRACT

We study a dynamic version of the Multiple-Message Broadcast problem, where packets are continuously injected in network nodes for dissemination throughout the network. Our performance metric is the ratio of the throughput of such protocol against the optimal one, for any sufficiently long period of time since startup. We present and analyze a dynamic Multiple-Message Broadcast protocol that works under an affectance model, which parameterizes the interference that other nodes introduce in the communication between a given pair of nodes. As an algorithmic tool, we develop an efficient algorithm to schedule a broadcast along a BFS tree under the affectance model. To provide a rigorous and accurate analysis, we define two novel network characteristics based on the network topology, the affectance function and the chosen BFS tree. The combination of these characteristics influence the performance of broadcasting with affectance (modulo a polylogarithmic function). We also carry out simulations of our protocol instantiating affectance in the Radio Network model. To the best of our knowledge, this is the first dynamic Multiple-Message Broadcast protocol that provides throughput guarantees for continuous injection of messages and works under the affectance model.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*sequencing and scheduling*

Keywords

Multiple-Message Broadcast, Radio Network, Affectance

1. INTRODUCTION

We study the dynamic Multiple-Message Broadcast problem in wireless networks under the *affectance* model. This model subsumes many communication-interference models studied in the literature, such as Radio Network (cf., [6]) and models based on the Signal to Interference and Noise Ratio (SINR) (cf. [19, 30]). The notion of affectance was first introduced in [19] in the context of link scheduling in the more restricted SINR model of wireless networks, in an attempt to formalize the combination of interferences from a subset of links to a selected link under the SINR model. Later on, other realizations of affectance were defined and abstracted as an independent model of interference in wireless networks [23,24]. The conceptual idea of this model is to parameterize the interference that transmitting nodes introduce in the communication between a given pair of nodes.

Our results. In the *dynamic Multiple-Message Broadcast* problem considered in this work, packets arrive at nodes in an online fashion and need to be delivered to all nodes in the network. We are interested in the *throughput*, i.e., the number of packets delivered in a given period of time. In particular, we measure *competitive throughput* of *deterministic distributed algorithms* for the dynamic Multiple-Message Broadcast problem. We analyse our algorithms in the (general) affectance model, in which there is a given undirected communication graph G of n nodes and diameter D, together with the affectance function $a(\cdot)$ of nodes of distance at least 2 on each of the communication links. The affectance function has a *degradation parameter* α, being a distance after which the affectance is negligible. Our contribution is two fold.

First, we introduce new model characteristics — based on the underlying communication network, the affectance function, and a chosen BFS tree — called *maximum average tree-layer affectance* (denoted by K) and *maximum fast-paths affectance* (denoted by M), see Section 2 for the definitions, and show how they influence the time complexity of broadcast. More precisely, if one uses a specific BFS tree, called GBST (cf., [16]), that minimizes the product $M \cdot (K + M)$ of the two above characteristics, then a single broadcast can be done in time $D + O(M(K + M)\log^3 n)$,[1] cf., Corollary 3 in Section 3.

Second, we extend this method of analysis to a dynamic packet arrival model and the Multiple-Message Broadcast problem, and design a new algorithm reaching competitive throughput of $\Omega(1/(\alpha K \log n))$. In particular, in the Radio Network model it implies a competitive throughput of $\Omega(1/(\log^2 n))$. For details, see Section 4. Our deterministic results are existential, that is, we show the existence of a deterministic schedule by applying a probabilistic ar-

[1]Throughout, we denote $\log_2$ simply as log, unless otherwise stated.

FOMC'14, August 11, 2014, Philadelphia, PA, USA.

http://dx.doi.org/10.1145/2634274.2634279 .

gument to a protocol that includes a randomized subroutine for layer to layer dissemination. Given that we measure competitive throughput in the limit, preprocessing (communication infrastructure setup, topology information dissemination, etc.) can be carried out initially without asymptotic impact. Thus, the protocol presented is distributed, and it works for *every* network after learning its topology. The protocol can also be applied to mobile networks, if the movement is slow enough to recompute the structure. Our rigorous asymptotic analysis is further complemented by simulations done for the Radio Network model, c.f., Section 5.

To the best of our knowledge, ours is the first work on the dynamic Multiple-Message Broadcast problem in wireless networks under the general affectance model.

Previous and related work. There is a rich history of research on broadcasting dynamically arriving packets on a *single-hop radio network*, also called a *multiple access channel*. Most of the research focused on *stochastic arrivals*, cf., a survey by Chlebus [8]. In the remainder of this paragraph, we focus on the on-line adversarial packet arrival setting. Bender et al. [5] studied stability, understood as throughput being not smaller than the packet arrival rate, of randomized backoff protocols on multiple access channels in the *queue-free model*, in which every packet is handled independently as if it has been a standalone station (thus avoiding queuing problems). Kowalski [26] considered a dynamic broadcast on the channel in the setting where packets could be combined in a single message, which again avoids various important issues related with queuing. Anantharamu et al. [3] studied packet latency of deterministic dynamic broadcast protocols for arrival rates *smaller* than 1. Stability, understood as bounded queues, of dynamic deterministic broadcast on multiple access channels against adversaries bounded by arrival rate 1 was studied by Chlebus et al. [10], and for arrival rates smaller than 1 by Chlebus et al. [11]. In particular, in [10] a protocol Move-big-to-front (MBTF) was designed, achieving stability but not fairness (as both these properties are impossible to achieve simultaneously); we use this algorithm as a subroutine in our dynamic Multiple-Message Broadcast protocol.

In *multi-hop* Radio Networks, the previous research concentrated on time complexity of single instances (i.e., from a single source) of broadcast and multi-message broadcast. For directed networks, the best deterministic solution is a combination of the $O(n \log n \log \log n)$-time algorithm by De Marco [15] and the $O(n \log^2 D)$-time algorithm by Czumaj and Rytter [13]. In undirected networks, the best up to date deterministic broadcast in $O(n \log(n/D))$ rounds was given by Kowalski [26]. The lower bounds for deterministic broadcast in directed and undirected radio networks are $\Omega(n \log(n/D))$ [12] and $\Omega(n \log_D n)$ [27], respectively. Deterministic multi-message broadcast, group communication and gossip were also considered (again, in a single instance). Chlebus et al. [9] showed a $O(k \log^3 n + n \log^4 n)$ time deterministic multi-broadcast algorithm for k packets in undirected radio networks. Single broadcast can be done optimally in $\Theta(D \log(n/D) + \log^2 n)$, as proved in [2, 29] (lower bounds) and in [13,27] (matching upper bound). Bar-Yehuda et al. [4], and recently Khabbazian and Kowalski [25] and Ghaffari et al. [18], studied randomized multi-broadcast protocols; the best results obtained for k-sources single-instance multi-broadcast is the amortized $O(\log \Delta)$ rounds per packet w.h.p. in [25], where Δ is the maximum node degree, and $O(D + k \log n + \log^2 n)$ w.h.p. to broadcast the k packets, for settings with known topology in [18]. For the same problem, Ghaffari et al. showed a throughput upper bound of $O(1/\log n)$ for any algorithm in [17]. Although this bound is worst-case, it can be compared with our $1/O(\alpha K \log n)$ that applies even under affectance.

Chlebus et al. [10] gave various deterministic and randomized algorithms for group communication, all of them being only a small polylogarithm away of the corresponding lower bounds on time complexity.

In the *SINR model*, single-hop instances of broadcast in the ad-hoc setting were studied by Jurdzinski et al. [21, 22] and Daum et al. [14], who gave several deterministic and randomized algorithms working in time proportional to the diameter multiplied by a polylogarithmic factor of some model parameters. In the SINR model with restricted sensitivity, so called weak-sensitivity device model, Jurdzinski and Kowalski [20] designed an algorithm spanning an efficient backbone sub-network, that might be used for efficient implementation of multi-broadcast.

The *generalized affectance* model was introduced and used only in the context of one-hop communication, more specifically, to link scheduling by Kesselheim [23]. He also showed how to use it for dynamic link scheduling in batches. This model was inspired by the affectance parameter introduced in the more restricted SINR setting [19]. They give a characteristic of a set of links, based on affectance, that influence the time of successful scheduling these links under the SINR model. In our paper, we generalize this characteristic, called the maximum average tree-layer affectance, to be applicable to multi-hop communication tasks such as broadcast, together with another characteristic, called the maximum fast-paths affectance. For details see Section 2.

2. PRELIMINARIES

Model. We study a model of network consisting of n nodes, where communication is carried out through radio ***transmissions*** in a shared channel. Time is discretized in a sequence of time slots $1, 2, \dots,$ which we call the ***global time***. The network is modeled by the underlying ***connectivity graph*** $G = \{V, E\}$, where V is the set of nodes and E the set of links among nodes. Let a link $\ell \in E$ between two nodes $u, v \in V$ be the set $\{u, v\}$. The network is assumed to be connected but ***multihop***. That is, not all possible links are present in E, but any pair of nodes may communicate, possibly through multiple hops.

Messages to be broadcast to the network through radio transmissions are called ***packets***. Packets are ***injected*** at nodes at the beginning of time slots, and each time slot is long enough to transmit a packet to a neighboring node. Any given node can either transmit or listen (in order to receive, if possible) in a time slot. Two or more transmissions received at a third node simultaneously are garbled. This event is called a ***collision***. Nodes cannot distinguish between a collision and the background noise in the channel, that is, collisions cannot be detected.

Additional interference on a link due to transmissions at more than one hop is modeled as affectance. We use a model of affectance that subsumes other communication-interference models, such as the Radio Network model (c.f., [6]) and the SINR model (c.f., [19]). Specifically, we realize ***affectance*** as a value $a_i(j) \leq 1$ that quantifies the interference that a transmitting node i introduces to the communication through link j. We do not restrict ourselves to any particular affectance function, as long as its effect is additive. That is, denoting $a_{V'}(j)$ as the affectance of a set of nodes on a link, for any $V' \subseteq V$ and $j \in E$, it is $a_{V'}(j) = \sum_{i \in V'} a_i(j)$. For a link (u, v), where u is the transmitter, we define $a_u((u, v)) = 0$ and $a_v((u, v)) = 1$, to model the positive (resp. negative) impact of a transmission from the transmitter (resp. receiver). Also, for $N(v)$ being the set of neighbors of v, we define $a_w((u, v)) = 1$ for each $w \neq u$ such that $w \in N(v)$.

Under the affectance model, we define a ***successful transmission*** as follows. For any pair of nodes $u, v \in V$ such that $\{u, v\} \in E$, a transmission from u is received at v in a time slot t if and only if: u transmits and v listens in time slot t, and $a_{\mathcal{T}}(u, v) < 1$, where $\mathcal{T}$ is the set of nodes transmitting in time slot t. We also denote the affectance of a set of nodes V' on a set of links E' as $a_{V'}(E')$, for any $V' \subseteq V$ and $E' \subseteq E$.

Communication task. Under the above model, we study the following ***Multiple-Message Broadcast*** problem. Starting at time slot 1, packets are being dynamically injected into ***source nodes*** for dissemination throughout the network. The set of all source nodes is denoted as $S \subseteq V$. After a packet has been received by all the nodes in the network, we say that the packet was ***delivered***. The injections are adversarial, that is, packets can be injected at any time slot at any source node, but the injections are limited to be feasible. We say that an injection is ***feasible*** if there exists an optimal algorithm OPT such that the ***latency*** (i.e., the time elapsed from injection to delivery) of each packet is bounded for OPT. Given that at most one packet may be received by a node in each time slot, and that all nodes must receive the packet in order the packet to be delivered, this assumption limits the adversarial injection rate to at most 1 packet per time slot for all nodes. The goal is to find a ***broadcasting schedule***, that is, a temporal sequence of transmit/not-transmit states for each node, so that packets are delivered. We denote the period of time since a packet is transmitted from the source until it is delivered the ***length*** of the schedule.

Performance metric. We evaluate the ***ratio*** of the performance of a distributed online algorithm ALG against an optimal algorithm OPT. For one hop networks it is known [10] that no protocol is both ***stable*** (i.e., bounded number of packets in the system at any time) and ***fair*** (i.e., every packet is eventually delivered). For multihop networks the same result holds as a natural extension of the single hop model. Thus, instead of further limiting the adversary (beyond feasibility) to achieve stability or bounded latency, our goal is to prove a lower bound on the ***competitive throughput***, for any sufficiently long prefix of time slots since global time 1. Specifically, we want to prove that there exists a function f, possibly depending on network parameters, such that $\lim_{t\to\infty} d_{ALG}(t)/d_{OPT}(t) \in \Omega(f)$, where $d_X(t)$ is the number of packets delivered to all nodes by algorithm X until time slot t.

Network characterization. We characterize a network by its ***affectance degradation distance***, which is the number of hops α such that the affectance of nodes of distance bigger that α in the network G to a given link is "negligible", that is, zero. Additionally, we characterize the network with two measures of affectance based on broadcast trees, as follows. Given a network with a set of nodes V including a source node s, consider a gathering-broadcast spanning tree (GBST) [16] rooted at s. A GBST is a breadth-first-search tree with a specific node ranking, satisfying the property that no two links of senders and receivers with the same rank create collisions (i.e., the receivers are different and there is no "cross link" between the sender in one link and receiver in the other). We define a node-set partition (slightly different than the partition in [16] for convergecast) based on that ranking and the distance to the source. Specifically, for a GBST tree T, the set of nodes V is partitioned in sets $F_d^r(T)$ and sets $S_d(T)$. A node of rank r at (shortest) distance d from the source is in set $F_d^r(T)$ if it has a child of the same rank (so called ***fast nodes***), or it is in set $S_d(T)$ otherwise (so called ***slow nodes***). Let $V_d(T) = F_d(T) \cup S_d(T)$, where $F_d(T) = \bigcup_r F_d^r(T)$. That is, $V_d(T)$ is the set of all nodes at distance d from the root. Based on this partition, we define the ***maximum average tree-layer affectance***

$$K(T, s) = \max_d \max_{V' \subseteq V_d(T)} \frac{1}{|L(V')|} a_{V'}(L(V')) \ ,$$

where $L(V')$ is the set of GBST links between V' and nodes at distance $d+1$ of the source. Additionally, we define the ***maximum fast-paths affectance***

$$M(T, s) = \max_{d,r} \max_{\ell \in F_d^r(T)} a_{F_d^r(T) \setminus \ell}(\ell) \ .$$

Given a GBST tree, the former characteristic says what is the maximum average affectance of a subset of nodes in the same layer on the links to their children in the tree, while the latter characteristic says what is the maximum affectance of fast links of the same rank and originated in the same layer to one of them. Intuitively, the former characteristic indicates what might be the worst affectance to overcome when trying to broadcast from one layer to another, while the latter one indicates what is the worst affectance when trying to pipeline a packet via fast links. In the rest of the paper, the specific tree and source node s will be omitted when clear from the context.

3. A BROADCAST TREE

In this section, we show a broadcasting schedule that, under the affectance model, disseminates a packet held at a source node to all other nodes. The schedule is defined constructively with a protocol that uses randomization, thus providing only stochastic guarantees. Given that the protocol is Las Vegas, the construction also proves the existence of a deterministic broadcasting schedule.

First, we detail the construction of a ranked tree spanning the network rooted at the source node that will be used to define the broadcasting schedule that we detail afterwards. The following notation will be used.

Given a tree $T(s) \subseteq E$ rooted at $s \in V$, spanning a set of network nodes V with set of links E, let $d(v)$ be the distance in hops from a node $v \in V$ to the root of $T(s)$, let $p(\ell)$ and $c(\ell)$ be the parent and child nodes of link $\ell \in T(s)$ respectively, and let $D(T(s))$ be the maximum distance in $T(s)$ from any node to the root s. Additionally, a ***rank*** (a number in $\mathbb{N}$) will be assigned to each node. Let $r(u)$ be the rank of node $u \in V$, let $R(T(s))$ be the maximum rank in the tree, and let $F_d^r = \{u | u \in V \wedge d(u) = d \wedge \exists v \in V : v = c(u) \wedge r(v) = r(u) = r\}$, that is, the set of nodes of rank r at distance d from the root that have a child with the same rank. In the above notation, the specific tree parameter and/or source node will be omitted when clear from the context.

Then, given a graph G and a source node $s \in S$, consider the following construction of a ***Low-Affectance Broadcast Spanning Tree (LABST)***. Let T_{min} be the GBST that minimizes the following polynomial on the affectance measures. Letting $\mathcal{T}$ be the class of all GBSTs that can be defined with source s, it is $\forall T \in \mathcal{T} : M(T_{\min}, s)(M(T_{\min}, s) + K(T_{\min}, s)) \leq M(T, s)(M(T, s) + K(T, s))$. Then, using Algorithm 1, transform $T_{\min}$ into a LABST T that avoids links between nodes of the same rank with big affectance.

Algorithm 1: LABST construction.

1 $T \leftarrow T_{\min}$
2 **foreach** *rank* $r = R(T), R(T) - 1, \ldots, 2, 1$ **do**
3 $\quad r \leftarrow r\lceil M(T) \rceil$
$\quad$ //now it is $R(T) = R(T_{\min})\lceil M(T) \rceil$
4 update all sets F_d^r.
5 **foreach** *distance* $d = D(T), \ldots, 2, 1$ **do**
6 $\quad$ **foreach** *rank* $r = 1, 2, \ldots, R(T)$ **do**
7 $\quad\quad$ **foreach** *link* ℓ *such that* $p(\ell) \in F_d^r$ **do**
8 $\quad\quad\quad$ **if** $a_{F_d^r \setminus \ell}(\ell) \geq 1$ **then** $r(p(\ell)) \leftarrow r + 1$
9 $\quad$ update all sets F_d^r.

The broadcasting schedule is defined using the LABST T obtained. Being a radio-broadcast network, transmissions might be received using other links or time slots, but the LABST and broadcasting schedule defined provide the communication guarantees. Each node follows certain broadcasting schedule, but using only time slots reserved for itself. Specifically, let a node $v \in V$ be called ***fast*** if it belongs to the set $F_{d(v)}^{r(v)}(T)$, and ***slow*** otherwise. Then, for each node $v \in V$, if v is fast, it uses each time slot t such that $t \equiv d(v) + 2h(R(T) - r(v)) \pmod{2hR(T)}$, where $h = \max\{3, \alpha\}$ and α is the affectance degradation distance. Otherwise, if v is slow, it uses each time slot t such that $t \equiv d(v) + h \pmod{2h}$. (The reason for this particular choice of reserved slots will become clear in Theorem 2.)

The ***broadcasting schedule*** for fast nodes is simple: upon receiving a packet for dissemination, transmit in the next time slot reserved. For slow nodes, the schedule is determined by a randomized contention resolution protocol that can be run in the reserved time slots. The protocol is simple: upon receiving a packet for dissemination, each slow node transmits repeatedly with probability $1/(4K(T_{\min}, s))$, until the packet is delivered. In the rest of this section, we bound the length of the broadcasting schedule. The following upper bound will be used.

LEMMA 1. *The maximum rank of a LABST on a network of n nodes with source node s is*

$$R(T) \leq \lceil \log n \rceil \lceil M(T_{\min}, s) \rceil.$$

PROOF. Consider the construction of a LABST T. The initial GBST $T_{\min}$ guarantees that the maximum rank is $R(T_{\min}) \leq \lceil \log n \rceil$(cf. [16]). Consider Algorithm 1, after Line 3, it is $R(T) \leq \lceil \log n \rceil \lceil M(T_{\min}) \rceil$. We show here that such overhead is enough for all the updates in Line 8.

Consider any path p from root to leaf in $T_{\min}$ defined by its set of links in the path (the order is implicit). Let $p' \subseteq p$ be the set of all links in a maximal subpath of p where all nodes have the same rank. The maximum number of ranks needed for the updates Line 8 is $\left\lceil \sum_{\ell \in p'} a_{F_{d(p(\ell))}^{r(p(\ell))} \setminus \ell}(\ell)/|p'| \right\rceil$. The bound holds because each time that a link is removed from such path, a value ≥ 1 is reduced from the total affectance of the path, and fast nodes continue being fast (possibly in a different set) even after updating the rank. Also, because fast nodes are still fast after the update, no new collisions appear and the links do not need to be updated. Given that $a_{F_{d(p(\ell))}^{r(p(\ell))} \setminus \ell}(\ell) \leq M(T_{\min})$, it is $\left\lceil \sum_{\ell \in p'} a_{F_{d(p(\ell))}^{r(p(\ell))} \setminus \ell}(\ell)/|p'| \right\rceil \leq \left\lceil \sum_{\ell \in p'} M(T_{\min})/|p'| \right\rceil = \lceil M(T_{\min}) \rceil$. Thus, the rank overhead with respect to $T_{\min}$ is enough. $\square$

THEOREM 2. *For any given network of n nodes with a source node, diameter D, and affectance degradation distance α, there exists a broadcasting schedule of length*

$$D + 2h\lceil \log n \rceil^2 \left(\lceil M(T_{\min}) \rceil^2 + 16\lceil M(T_{\min}) \rceil K(T_{\min})\right),$$
$$\textit{where } h = \max\{3, \alpha\}.$$

PROOF. First we show that the broadcasting schedule is correct. Consider any pair of nodes $u, v \in V$ transmitting in the same time slot. If $d(u) = d(v)$ and they are both fast nodes with the same rank, the affectance on each other's links is low by definition of the LABST. If $d(u) = d(v)$ and they are both slow nodes, the contention resolution protocol will disseminate the packet to the next layer. Otherwise, given the slot reservation, $|d(u) - d(v)| \geq h$. Given that $h \geq \alpha$, the affectance on each other's links is negligible, and given that $h \geq 3$, there are no collisions between their transmissions.

To prove the schedule length, consider any path p from root to leaf in the LABST T. The path p can be partitioned into consecutive maximal subpaths according to rank. In each maximal subpath $p' \in p$ of consecutive nodes of the same rank, the first node may have to wait up to $2hR(T)$ slots for the next reserved time slot, but after that all nodes except the last one transmit in consecutive time slots. Given that there are at most $R(T)$ such maximal subpaths and that their aggregated length is at most $D(T)$, the schedule length in the fast nodes of path p is at most $D(T) + 2hR(T)^2 \leq D + 2hR(T)^2$, where the latter inequality holds because T is a BFS tree.

Consider now any link $\ell \in p$ where the rank changes, that is $r(p(\ell)) \neq r(c(\ell))$ and $p(\ell) \in S_{d(p(\ell))} \subseteq V_{d(p(\ell))}$. Recall that the schedule in such link is defined by a randomized contention resolution protocol where each node transmits with probability $1/(4K(T_{\min}))$, where

$$K(T_{\min}) = \max_{d} \max_{V' \subseteq V_d(T_{\min})} \frac{1}{|L(V')|} a_{V'}(L(V')),$$

where $L(V')$ is the set of GBST links between V' and nodes at distance $d + 1$ of the source, and $V_d(T_{\min})$ is the set of nodes at distance d from the source in $T_{\min}$.

For a probability of transmission

$$q \leq \frac{1}{4 \max_{S \subseteq V_{d(p(\ell))}} a_S(L(S))/|L(S)|},$$

it was proved in [24] that the probability that there is still some link in S where no transmission was successful after $4c \ln |V_{d(p(\ell))}|/q$ time slots running Algorithm 1 in [24], is at most $|V_{d(p(\ell))}|^{1-c}$, $c > 1$. Given that $1/(4K(T_{\min}))$ verifies such condition, we know that after

$$16cK(T_{\min}) \ln |V_{d(p(\ell))}| \leq 16cK(T_{\min}) \ln n$$

(reserved) time slots, the transmission in link ℓ has been successful with positive probability. Given that there are at most $R(T)-1$ links where the rank changes, using the union bound, we know that after $(R(T)-1)16cK(T_{\min}) \ln n$ (reserved) time slots all slow nodes have delivered their packets with some positive probability, which shows the existence of a deterministic schedule of such length[2]. The time slots reserved for slow nodes appear with a frequency of $2h$. Thus, the schedule length in the slow nodes of path p is at most $2h(R(T)-1)16cK(T_{\min}) \ln n \leq 32hR(T)K(T_{\min}) \ln n$, for $c = R(T)/(R(T)-1)$.

Adding both schedule lengths we have

$$D + 2hR(T)^2 + 32hR(T)K(T_{\min}) \ln n$$

Replacing the bound on $R(T)$ in Lemma 1, the claim follows. □

For networks with affectance degradation distance $\lceil \log n \rceil$, Theorem 2 yields the following corollary.

Corollary 3. *For any given network of $n \geq 8$ nodes, diameter D, and affectance degradation distance $\lceil \log n \rceil$, there exists a broadcasting schedule of length*

$$D + O(\log^3 n(M(T_{\min})(M(T_{\min}) + K(T_{\min})))).$$

For comparison, for less contentious networks where affectance is not present (Radio Network model), using a GBST a broadcast schedule of length $D + O(\log^3 n)$ was shown in [16] and of length $O(D + \log^2 n)$ was proved in [28].

4. A DYNAMIC Multiple-Message Broadcast PROTOCOL

In this section, we present our Multiple-Message Broadcast protocol and we bound its competitive throughput. The protocol uses the LABST[3] presented in Section 3.[4] The intuition of the protocol is the following. Each source node has a (possibly empty) queue of packets that have been injected for dissemination. Then, starting with an arbitrary source node $s \in S$ with "large enough" number of packets in its queue, packets are disseminated through a LABST rooted at s. If the number of packets in the queue of s becomes "small", s stops sending packets and, after some delay to clear the network, another source node $s' \in S$ starts disseminating packets through a LABST rooted at s'. The procedure is repeated following the order of a list of source nodes, which is dynamically updated according to queue sizes to guarantee good throughput. Packets from any given source are pipelined with some delay to avoid collisions and affectance. Being a radio broadcast network, packets might be received earlier than expected using links or time slots other than those defined by the LABST. If that is the case, to guarantee the pipelining, nodes ignore those packets.

The following notation will be also used. The LABST rooted at $s \in S$ is denoted as $T(s)$. We denote the length of the broadcast schedule (time to deliver to all nodes) from s as $\Delta(s)$, and $\Delta = \max_{s \in S} \Delta(s)$. Let the pipeline delay (the time separation needed between consecutive packets to avoid collisions and affectance) from s be $\delta(s)$, and $\delta = \max_{s \in S} \delta(s)$. Given a node $i \in S$ and time slot t, the length of the queue of i is denoted $\ell(i,t)$. Let the length of all queues at time t be $\ell(t) = \sum_{i \in S} \ell(i,t)$. We say that, at time t, a node i is ***empty*** if $\ell(i,t) < \Delta$, ***small*** if $\Delta \leq \ell(i,t) < n\Delta$, and ***big*** if $\ell(i,t) \geq n\Delta$.

Consider the following ***Multiple-Message Broadcast Protocol***.

1. For each source node $s \in S$ define a LABST rooted at s.

2. Define a Move-big-to-front (MBTF) list [10] of source nodes, initially in any order. According to this list, source nodes circulate a token. While being disseminated, the token has a time-to-live counter of Δ, maintained by all nodes relaying the token. A source node s receiving the token has to wait for the token counter to reach zero before starting a new transmission. Let the time slot when the counter reaches zero be t. Then, node s does the following depending on the length of its queue.

   (a) If s is empty at t, it passes the token to the next node in the list. We call this event a ***silent round***.

   (b) If s is small at t, it broadcasts Δ packets pipelining them in intervals of δ slots. After δ more slots, it passes the token to the next node in the list.

   (c) If s is big at t, it moves itself to the front of the list. We call this event a ***discovery***. Then, s broadcasts packets pipelining them in intervals of δ slots as long as it is big, but a minimum of Δ packets. With the first of these packets s broadcasts the changes in the list. δ more slots after transmitting these packets, it passes the token to the next node in the list.

The following theorem shows an upper bound on the number of packets in the system at any time, which allows to prove the competitive throughput of our protocol. The proof structure is similar to the proof in [10] for MBTF, but many details have been redone to adapt it to a multihop network.

Theorem 4. *For any given network of n nodes, at any given time slot t of the execution of the Multiple-Message Broadcast protocol defined, the overall number of packets in queues is $\ell(t) < (t\delta/(1+\delta)) + 2\Delta n^2$.*

Proof. For the sake of contradiction, assume that there exists a time t such that the overall number of packets in the system is $\ell(t) \geq (t\delta/(1+\delta)) + 2\Delta n^2$. The number of packets in queues at the end of any given period of time is at most the number of packets in queues at the beginning of such period, plus the number of time slots when no packet is delivered, given that at most one packet is injected in each time slot. We arrive to a contradiction by upper bounding

[2] In settings with collision detection and where the affectance on any given link is $O(n)$, a big enough constant $c > 1$ yields a randomized protocol that succeeds with probability $1 - 1/n$.

[3] We refer to the tree and the broadcast schedule indistinctively.

[4] Any broadcast schedule that works under the affectance model could be used.

the number of time slots when no packet is delivered within a conveniently defined period before t. Consider the period of time T such that

$$\ell(t-T) \leq n^2\Delta + \frac{(t-T)\delta}{1+\delta} \quad (1)$$

$$\forall t' \in [t-T, t] : \ell(t') \geq n^2\Delta \quad (2)$$

$$\ell(t) \geq (t\delta/(1+\delta)) + 2\Delta n^2 \quad (3)$$

From now on, the analysis refers to the period of time T. We omit to specify it for clarity. Let $C \subseteq S$ be the set of nodes that are big at some point. Due to the pigeonhole principle and Equation (2), we know that for each time slot there is at least one big source node. In other words, the token cannot be passed throughout the whole list without at least one discovery. As a worst case, assume that only nodes in C have packets to transmit. For each node $i \in C$, the token has to be passed through at most $|S \setminus C| \leq n-|C|$ nodes that are not in C before i is discovered, because after i is discovered no node in $S \setminus C$ will be before i in the list. Hence, there are at most $|C|(n-|C|)$ silent rounds, each of length Δ for token pass. So, due to passing the token through nodes in $S \setminus C$, there are at most $|C|(n-|C|)\Delta$ time slots when no packet is delivered.

We bound now the time slots when no packet is delivered due to passing the token through nodes in C before being discovered for the first time. Consider any given node $i \in C$. The argument is similar to the previous case. Any other node $j \in C$ that is discovered before i is moved to the front of the list. If i is going to be before j in the list later, it is not going to happen before i is discovered for the first time. Then, before i is discovered, it may hold the token at most $|C|-1$ times. As a worst case, assume that for each of these times i is empty. Hence, there are at most $|C|(|C|-1)$ silent rounds, each of length Δ for token pass. So, due to passing the token through nodes in C before being discovered, there are at most $|C|(|C|-1)\Delta$ time slots when no packet is delivered.

It remains to bound the time slots when no packet is delivered due to pipelining and passing the token through nodes in C after being discovered. Consider any given node $i \in C$ after being discovered. If i is big during the rest of T, it broadcasts packets pipelining them in intervals of δ slots. If instead i becomes small during T, i will have Δ packets to transmit for at least $n-1$ times that holds the token afterwards before becoming empty, because right after becoming small it has at least $(n-1)\Delta$ packets in queue. And there are at most $n-1$ nodes in C that will not be behind i in the list until i becomes big again. Hence, i always has Δ packets to transmit after being discovered the first time. After becoming small, i has to pass the token to the next node in the list introducing a delay of Δ. As a worst case scenario, we assume that upon each discovery of each node $i \in C$, only Δ packets are broadcast before passing the token. Then, for each Δ packets delivered, there are at most $\Delta + \Delta(\delta-1) = \Delta\delta$ time slots when no packet is delivered, over a period of $\Delta + \Delta\delta = \Delta(1+\delta)$ time slots. Because C is the set of nodes that are discovered in T, we can bound the number of batches of Δ packets delivered in T by $\lfloor T/(\Delta(1+\delta)) \rfloor \leq T/(\Delta(1+\delta))$. Then, there are at most $T\Delta\delta/(\Delta(1+\delta)) = T\delta/(1+\delta)$ time slots when no packet is delivered due to nodes in C after being discovered.

Combining these bounds with Equation (1), we have that there are at most

$$\begin{aligned} n^2\Delta + \frac{(t-T)\delta}{1+\delta} + |C|(n-|C|)\Delta + |C|(|C|-1)\Delta + \frac{T\delta}{1+\delta} \\ = n^2\Delta + \frac{t\delta}{1+\delta} + \Delta|C|(n-1) \\ < \frac{t\delta}{1+\delta} + 2\Delta n^2 \end{aligned}$$

time slots when no packet is delivered. Which is a contradiction. □

LEMMA 5. *There exists a Multiple-Message Broadcast protocol that achieves a competitive throughput of at least*

$$\lim_{t\to\infty} \frac{1}{1+\delta} - \frac{2\Delta n^2}{t}.$$

PROOF. A packet is delivered when it has been received by *all* nodes. The optimal algorithm delivers at most one packet per time slot, since any given node can receive at most one packet per time slot. Additionally, the injection is limited to be feasible, that is, there must exist an optimal algorithm OPT such that the latency of each packet is bounded for OPT. Thus, at most one packet may be injected in each time slot. Then, the competitive throughput is at least

$$\lim_{t\to\infty} \frac{d_{ALG}(t)}{d_{OPT}(t)} \geq \lim_{t\to\infty} \frac{t - nd_{ALG}(t)}{t},$$

where $nd_{ALG}(t)$ is the max number of packets that could not be delivered by ALG by time t. Using the bound in Theorem 4 we have that

$$\begin{aligned} \lim_{t\to\infty} \frac{d_{ALG}(t)}{d_{OPT}(t)} &\geq \lim_{t\to\infty} \frac{t - (t\delta/(1+\delta)) - 2\Delta n^2}{t} \\ &\geq \lim_{t\to\infty} \frac{1}{1+\delta} - \frac{2\Delta n^2}{t}. \end{aligned}$$

□

The following theorem shows our main result.

THEOREM 6. *For any given network of n nodes, diameter D, and affectance degradation distance α, there exists a Multiple-Message Broadcast protocol that achieves a competitive throughput of at least*

$$\lim_{t\to\infty} \frac{1}{1+\delta} - \frac{2\Delta n^2}{t}.$$

Where

$$\begin{aligned} \Delta &\leq D + 2\max\{3,\alpha\}\lceil \log n \rceil^2 \\ &\quad \left(\lceil M(T_{\min}) \rceil^2 + 16\lceil M(T_{\min}) \rceil K(T_{\min})\right), \\ K &= \max_{s\in S} K(T_{\min}(s), s), \\ M &= \max_{s\in S} M(T_{\min}(s), s), \\ \delta &= \max\{3,\alpha\} 16K \ln n. \end{aligned}$$

PROOF. The length $\Delta(s)$ of the broadcast schedule in a LABST rooted at s is given in Theorem 2. With respect to $\delta(s)$, as explained in the proof of Theorem 2, slow nodes at distance d from the root deliver a packet to the next node in a path of a LABST $T(s)$ within $16cK(T_{\min}(s)) \ln |V_d|$ with positive probability for any $c > 1$. This shows the existence of a deterministic schedule of that length. Additionally, packets must be separated by at least $\max\{3, \alpha\}$ to avoid collisions and affectance from nodes at different distances from the source (see the proof of Theorem 2 for further details). Then, it is $\delta(s) = \max\{3, \alpha\} 16K(T_{\min}(s)) \ln n$, for $c = \ln n / \ln |V_d|$. Replacing, the claim follows. □

The above theorem yields the following corollary that provides intuition.

Corollary 7. *For any given network of n nodes, diameter D, and affectance degradation distance α, there exists a Multiple-Message Broadcast protocol such that the competitive throughput converges to*

$$\frac{1}{O(\alpha K \log n)},$$

where $K = \max_{s \in S} K(T_{\min}(s), s)$.

To evaluate these results, it is important to notice that the competitive throughput bound was computed against a theoretical optimal protocol that delivers one packet per time slot, which is not possible in practice in a multi-hop network. For comparison, instantiating our interference model in the Radio Network model (no affectance), using the WEB protocol [7] for slow transmissions our Multiple-Message Broadcast protocol can be shown to converge to $1/O(\log^2 n)$. Furthermore, for single-instance multi-broadcast in Radio Network, Ghaffari et al. showed in [17] a throughput upper bound of $O(1/\log n)$ for any algorithm. Although this bound is worst-case, it can be compared with our $1/O(\alpha K \log n)$ that applies even under affectance. We evaluate the Radio Network case through simulations of our protocol in the following section.

5. SIMULATIONS

For simplicity, we carried out simulations of the Multiple-Message Broadcast protocol assuming the Radio Network model. That is, interference is due to collisions only. In absence of affectance, the LABST construction is simply a GBST. Furthermore, the affectance measures are zero and the broadcast tree becomes *any* GBST as defined in [16]. We simulated the tree broadcast schedule specified in Section 3, except for the protocol for small nodes transmissions from layer to layer, which in [16] is the deterministic schedule of the WEB protocol [7].

Regarding the delay δ and the schedule length Δ, using a GBST and the WEB protocol they are $\delta = \ln^2 n$ (cf. Lemma 6.2 in [7]) and $\Delta = max_{s \in S} D(s) + 6r_{\max}(s)^2 + r_{\max}(s) \ln^2 n$ (cf. [16]), where $r_{\max}$ is the maximum rank in the GBST rooted on the source node s. Using such broadcast trees from each source, we simulated the protocol in Section 4 for network sizes $n = 8, 16, 32, 64, 128, 256$, and 512. The input networks were random graphs $G(n, p)$, where $p = 1/5$, and each node was chosen to be a source at random with probability 1/3. The injection at a rate of one packet per time slot was also random with uniform distribution on the nodes. The packet queues of the source nodes were initialized to Δ packets. That is, initially all source nodes were small introducing overhead due to token passing.

The results of the simulations are illustrated by the plot in Figure 1. It can be seen that, after an initial phase, for any of the network sizes studied, the competitive throughput converges to a constant (with respect to time). Furthermore, except for the small networks, for bigger values of n it can be seen that the value of convergence decreases linearly although n grows exponentially, showing that the convergence value is approximately inverse logarithmic (with respect to n) as expected from replacing the value of δ in Lemma 5. It is important to notice that the competitive throughput was computed against a theoretical optimal protocol that delivers one packet per time slot, which is not possible in practice in a multi-hop network.

6. ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation (CCF 1114930, CCF 0937829) and Kean University UFRI grant.

7. REFERENCES

[1] Y. Afek, editor. *Distributed Computing - 27th International Symposium, DISC 2013, Jerusalem, Israel, October 14-18, 2013. Proceedings*, volume 8205 of *Lecture Notes in Computer Science*. Springer, 2013.

[2] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *J. Comput. Syst. Sci.*, 43(2):290–298, 1991.

[3] L. Anantharamu, B. S. Chlebus, D. R. Kowalski, and M. A. Rokicki. Deterministic broadcast on multiple access channels. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–5, 2010.

[4] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multihop radio networks. *SIAM J. Comput.*, 22(4):875–887, 1993.

[5] M. A. Bender, M. Farach-Colton, S. He, B. C. Kuszmaul, and C. E. Leiserson. Adversarial contention resolution for simple channels. In *Proceedings of the 17th Annual ACM Symposium on Parallel Algorithms (SPAA)*, pages 325–332, 2005.

[6] I. Chlamtac and S. Kutten. Tree-based broadcasting in multihop radio networks. *IEEE Trans. Computers*, 36(10):1209–1223, 1987.

[7] I. Chlamtac and O. Weinstein. The wave expansion approach to broadcasting in multihop networks. In *Proc. of INFOCOM*, 1987.

[8] B. S. Chlebus. *Randomized communication in radio networks*, volume I, pages 401 – 456. Kluwer Academic Publishers, 2001.

[9] B. S. Chlebus, D. R. Kowalski, A. Pelc, and M. A. Rokicki. Efficient distributed communication in ad-hoc radio networks. In L. Aceto, M. Henzinger, and J. Sgall, editors, *ICALP (2)*, volume 6756 of *Lecture Notes in Computer Science*, pages 613–624. Springer, 2011.

[10] B. S. Chlebus, D. R. Kowalski, and M. A. Rokicki. Maximum throughput of multiple access channels in adversarial environments. *Distributed Computing*, 22(2):93–116, 2009.

[11] B. S. Chlebus, D. R. Kowalski, and M. A. Rokicki. Adversarial queuing on the multiple access channel. *ACM Transactions on Algorithms*, 8(1):5, 2012.

[12] A. E. F. Clementi, A. Monti, and R. Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In S. R. Kosaraju, editor, *SODA*, pages 709–718. ACM/SIAM, 2001.

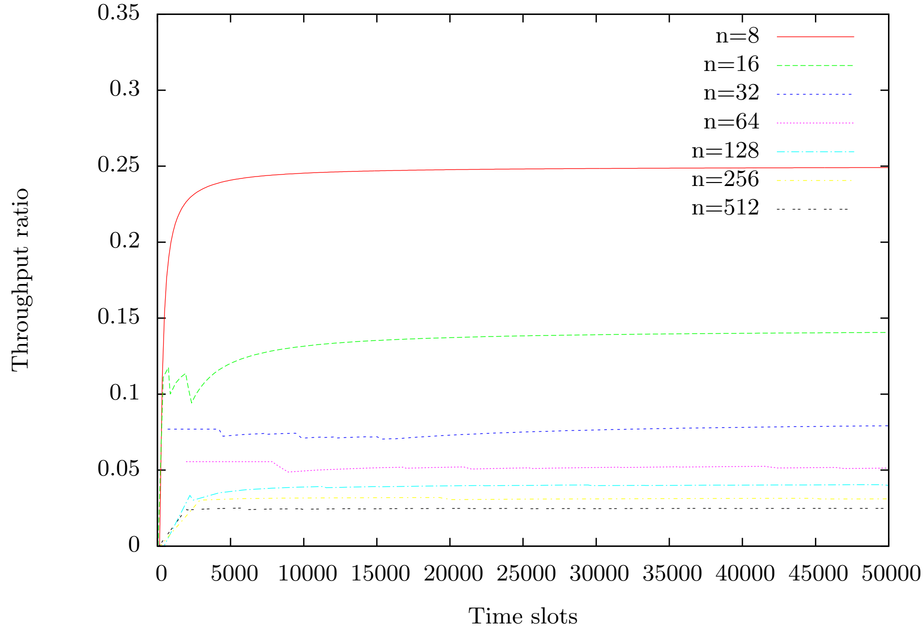

Figure 1: Competitive throughput vs. time.

[13] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *FOCS*, pages 492–501. IEEE Computer Society, 2003.

[14] S. Daum, S. Gilbert, F. Kuhn, and C. C. Newport. Broadcast in the ad hoc sinr model. In Afek [1], pages 358–372.

[15] G. De Marco. Distributed broadcast in unknown radio networks. In S.-H. Teng, editor, *SODA*, pages 208–217. SIAM, 2008.

[16] L. Gasieniec, D. Peleg, and Q. Xin. Faster communication in known topology radio networks. *Distributed Computing*, 19(4):289–300, 2007.

[17] M. Ghaffari, B. Haeupler, and M. Khabbazian. A bound on the throughput of radio networks. *CoRR*, abs/1302.0264, 2013.

[18] M. Ghaffari, B. Haeupler, and M. Khabbazian. Randomized broadcast in radio networks with collision detection. In *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, PODC '13, pages 325–334, New York, NY, USA, 2013. ACM.

[19] M. M. Halldórsson and R. Wattenhofer. Wireless communication is in apx. In *Proc. of the 36th International Colloquium on Automata, Languages and Programming, Part I*, pages 525–536, 2009.

[20] T. Jurdzinski and D. R. Kowalski. Distributed backbone structure for algorithms in the sinr model of wireless networks. In M. K. Aguilera, editor, *DISC*, volume 7611 of *Lecture Notes in Computer Science*, pages 106–120. Springer, 2012.

[21] T. Jurdzinski, D. R. Kowalski, M. Rozanski, and G. Stachowiak. Distributed randomized broadcasting in wireless networks under the sinr model. In Afek [1], pages 373–387.

[22] T. Jurdzinski, D. R. Kowalski, and G. Stachowiak. Distributed deterministic broadcasting in uniform-power ad hoc wireless networks. In L. Gasieniec and F. Wolter, editors, *FCT*, volume 8070 of *Lecture Notes in Computer Science*, pages 195–209. Springer, 2013.

[23] T. Kesselheim. Dynamic packet scheduling in wireless networks. In *Proc. of the 31st Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 281–290, 2012.

[24] T. Kesselheim and B. Vöcking. Distributed contention resolution in wireless networks. In *Proc. of the 24th International Symposium on Distributed Computing*, volume 6343 of *Lecture Notes in Computer Science*, pages 163–178. Springer-Verlag, Berlin, 2010.

[25] M. Khabbazian and D. R. Kowalski. Time-efficient randomized multiple-message broadcast in radio networks. In C. Gavoille and P. Fraigniaud, editors, *PODC*, pages 373–380. ACM, 2011.

[26] D. R. Kowalski. On selection problem in radio networks. In *Proceedings of the 24th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 158–166, 2005.

[27] D. R. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. *Distributed Computing*, 18(1):43–57, 2005.

[28] D. R. Kowalski and A. Pelc. Optimal deterministic broadcasting in known topology radio networks. *Distributed Computing*, 19(3):185–195, 2007.

[29] E. Kushilevitz and Y. Mansour. An omega(d log (n/d)) lower bound for broadcast in radio networks. *SIAM J. Comput.*, 27(3):702–712, 1998.

[30] C. Scheideler, A. W. Richa, and P. Santi. An o(log n) dominating set protocol for wireless ad-hoc networks under the physical interference model. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 91–100. ACM, 2008.