Stable Local Broadcast in Multihop Wireless Networks Under SINR

Dongxiao Yu, *Member, IEEE*, Yifei Zou, Jiguo Yu, *Senior Member, IEEE, Member, ACM*, Xiuzhen Cheng, *Fellow, IEEE, Member, ACM*, Qiang-Sheng Hua, *Member, IEEE*, Hai Jin, *Senior Member, IEEE*, and Francis C. M. Lau, *Senior Member, IEEE*

***Abstract*— We present a distributed stable protocol for local broadcast in multi-hop wireless networks, where packets are injected to the nodes continuously, and each node needs to quickly disseminate the injected packets to all its neighbors within a given communication range R. We investigate the maximum packet injection rate and the minimum packet latency that can be achieved in a stable protocol. This paper assumes the signal-to-interference-plus-noise-ratio (SINR) interference model, which reflects more accurately the physical characteristics of the wireless interference, such as fading and signal accumulation, than conventional local interference models, e.g., graph-based models. More specifically, we present a stable protocol that can handle both stochastic and adversarial injection patterns. The protocol is asymptotically optimal in terms of both injection rate and packet latency. To the best of our knowledge, this paper is the first one studying the properties of stable protocols for the basic primitive of local broadcast in a multi-hop setting under SINR. Our proposed protocol utilizes a static local broadcast algorithm as a subroutine. This static algorithm is of independent interest, and it closes the $O(\log n)$ gap between the upper and lower bounds for static local broadcast. Simulation results indicate that our proposed algorithms can perform well in realistic environments.**

***Index Terms*— Multi-hop wireless networks, distributed algorithm, stable protocol, SINR, continuous local broadcast.**

I. INTRODUCTION

INFORMATION exchange between nodes in wireless networks is typically done via local broadcast (to neighboring nodes only) on a shared channel. The *Static Local Broadcast* (SLB) problem, in which all the packets are stored at the nodes before dissemination, has been extensively studied under a variety of interference models, from local graph-based ones [2], [15] to the global SINR model [4], [19], [21], [27], [33]–[35]. The primitive of local broadcast is frequently used as a buliding block in algorithms for upper-layer functions such as Neighbor Discovery [8], [9], Broadcast [16], [24], and Consensus [25]. *Continuous packet Local Broadcast* (CLB), in which packets are injected into the nodes continuously, takes into consideration the reality that packets may actually arrive over time. There have been a fair amount of work focusing on this more realistic problem. However, these work usually assume a single-hop network topology, and are based on local interference models. Little is known about CLB in a multi-hop network setting, especially under the more sophisticated interference models. We believe this is an important missing piece for the study of information exchange in wireless networks, which is the focus of this paper. Specifically, we tackle the challenges of whether a protocol can be derived for continuous local broadcast that can achieve good performance with respect to certain crucial metrics.

Particularly, we consider the three most crucial aspects of the network protocol performance: *stability*, *throughput*, and *packet latency*. A protocol is said to be stable with respect to a packet injection pattern if, in any execution of the protocol, it ensures that the number of packets stored in the local queues of the nodes is bounded at any time. The throughput of a protocol is then defined to be the highest injection rate that a stable protocol can handle, where the injection rate is the average number of packets injected to each node per round. Packet latency is the maximum time at which a packet may stay in a queue. Obviously, it is an extremely hard task to devise stable protocols attaining optimal performance in terms of both throughput and packet latency, as high injection rate can easily give rise to long queues. Then the following natural questions arise: *what is the tradeoff between throughput and packet latency for a stable local broadcast protocol? can we get a stable protocol that attains optimality (or asymptotic optimality) in terms of both throughput and packet latency?* In this paper, we answer the above two questions affirmatively by proposing a stable protocol with asymptotically optimal throughput and packet latency.

The hardness of designing stable protocols for continuous local broadcast comes from the facts that packets are injected to the nodes continuously and the transmissions are easily

Manuscript received April 9, 2017; revised October 10, 2017 and February 28, 2018; accepted March 27, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor X.-Y. Li. Date of publication May 15, 2018; date of current version June 14, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2016QY02D0302, in part by the National Natural Science Foundation of China Grant 61602195, Grant 61672321, Grant 61771289, Grant 61572216, and Grant 61373027, in part by the Natural Science Foundation of Hubei Province under Grant 2017CFB301, and in part by the Fundamental Research Funds for HUST. *(Corresponding author: Jiguo Yu.)*

D. Yu, Q.-S. Hua, and H. Jin are with the Services Computing Technology and System Lab, Big Data Technology and System Lab, Clusters and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: dxyu@hust.edu.cn; qshua@hust.edu.cn; hjin@hust.edu.cn).

Y. Zou and F. C. M. Lau are with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: yfzou@cs.hku.hk; fcmlau@cs.hku.hk).

J. Yu is with Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China, and also with the School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China (e-mail:jiguoyu@sina.com).

X. Cheng is with the Department of Computer Science, The George Washington University, Washington, DC 20052 USA (e-mail: cheng@gwu.edu).

Digital Object Identifier 10.1109/TNET.2018.2829712

interrupted by wireless interference. Hence, the modeling of packet injection and wireless interference has a strong bearing on the complexity of the protocol design. There are two broad packet injection patterns: *stochastic injection* [14] and *restricted adversarial injection* [3]. In the former pattern, the packets are injected to nodes by a stochastic process, while the latter pattern usually imposes an upper bound on the number of packets injected (by an adversary) to each node in an interval of rounds. We mainly present and analyze our protocol with respect to the stochastic packet injection, which could incur an extremely high degree of burstiness at times. In this case, an unpredictably large amount of collisions may happen in the network, making transmission coordination as well as the analysis of the stability and other performance guarantees of the protocol rather difficult. As for the restricted adversarial injection pattern, our protocol achieves the same performance guarantee as in the stochastic injection case (see the discussions in Section V-D). To model the wireless interference, we adopt the SINR (Signal-to-Interference-plus-Noise-Ratio) model, which is also known as the physical model. The SINR model captures more faithfully the physical features of wireless interference, such as fading and signal accumulation, than other simpler models including the graph-based ones and the protocol model.

We design a distributed protocol for continuous local broadcast in this paper. A local distributed algorithm is always desirable in wireless networks as it allows each node to perform its operations independently and simultaneously. But distributed algorithm design is always a hard task because each wireless node can only communicate directly with the nodes within its communication range while SINR insists on a much wider scope when interference is taken into calculation.

Our main contributions are summarized as follows.

- We present the first known distributed stable protocol for CLB in multi-hop wireless networks under the SINR model. Our protocol is asymptotically optimal in terms of both throughput (injection rate) and packet latency. Moreover, it can handle both stochastic and restricted adversarial packet injection patterns.
- The proposed continuous protocol is based on a distributed algorithm for SLB, which makes each node disseminate its packets to all neighbors within a specified local broadcast range R in $O(m + \log^2 n)$ rounds with a high probability guarantee,[1] where m is the maximum number of packets stored at the neighborhood of each node and n is the total number of nodes in the network. This algorithm matches the lower bound in [21], and closes the $O(\log n)$ gap between the upper and lower bounds in existing work [21], [34], under the setting in which geometric location information and physical carrier sensing are not available.

We conduct extensive simulations to evaluate our algorithms, and the simulation results corroborate our theoretical analysis.

Organization: The remaining part of the paper is organized as follows. Section II introduces the most related work. In Section III, we present the network model, problem definitions, and preliminary knowledge. A distributed algorithm for SLB is given in Section IV, based on which we propose our protocol for CLB in Section V. The simulation results are reported in Section VI. We conclude the paper with a future research discussion in Section VII.

[1]"with high probability", w.h.p. for short, means with probability $1 - n^{-c}$ for some constant $c > 0$.

TABLE I

TIME BOUNDS FOR SLB

Reference	Time (Knowing Δ)	Time (Without Knowing Δ)
[19]	$O(\Delta \log n)$	$O(\Delta \log^3 n)$
[35]	—	$O(\Delta \log^2 n)$
[21]	$O(\Delta + \log^2 n)^1$	$O(\Delta \log n + \log^2 n)$
[34]	—	$O(\Delta \log n + \log^2 n)$
[4]	$O(\Delta + \log n \cdot \log\log n)^1$	—
[20]	$O(\Delta(\log n + \log \Gamma)$ $+ \log \Gamma(\log n + \log \Gamma))^2$	—
[13]	$O(\Delta K^2 \log n)^3$	$O((\Delta + \log n)K^2 \log n)^3$
This work	$O(\Delta + \log^2 n)$	—

[1] The results are obtained assuming free acknowledgement.
[2] Γ is the ratio of the maximum and the minimum distances between nodes; The result is obtained assuming that nodes know the global parameter Γ.
[3] The results are obtained in the setting of arbitrary power assignment; K is the ratio of the maximum and the minimum transmission ranges of nodes; in the uniform setting, $K = 1$.

II. Related Work

The SLB problem under the SINR model has been extensively studied [4], [19]–[21], [27], [34], [35]; but these existing work mainly consider the case in which each node only has one packet to share with its neighbors, which is different from our setting. In other words, $m = \Delta$ is assumed in these existing research, where Δ is the maximum number of neighbors of each node, while $m \geq \Delta$ is under our consideration. Comparisons of our results under the setting of $m = \Delta$ with existing ones are given in Table I. Goussevskaia *et al.* [19] gave the first results for local broadcast under the SINR model, with running time $O(\Delta \log n)$ and $O(\Delta \log^3 n)$ for the cases with known and unknown Δ, respectively. For the case of knowing Δ, an $O(\Delta(\log n + \log \Gamma) + \log \Gamma(\log n + \log \Gamma))$ time algorithm was given in [20], where Γ is the ratio of the maximum and the minimum distances between nodes and this global parameter needs to be known by nodes. We improve the result to $O(\Delta + \log^2 n)$ in this paper. The latter result in [19] was improved in [21], [34], and [35], and the best known result is $O(\Delta \log n + \log^2 n)$ independently given by [21] and [34]. Under a setting different from that in this paper, where free acknowledgement is provided, the result can be improved to $O(\Delta + \log^2 \ n)$ [21]. If further Δ is available, static local broadcasting can be accomplished in $O(\Delta + \log n \cdot \log\log n)$ under the spontaneous setting [4]. The SLB problem under the setting of arbitrary power assignment was studied in [13], where algorithms that can accomplish local broadcast in $O(\Delta K^2 \log n)$ time when Δ is known and $O((\Delta + \log n)K^2 \log n)$ otherwise were given, with K being the ratio of the maximum and minimum transmission ranges of nodes. In [27], Pei and Vullikanti considered a variance of the local broadcast problem, which requires to find the maximum set of nodes such that they can perform local broadcast at the same time. This can be seen as an "independent set" version of the SLB problem.

Stable protocols for CLB were studied in single-hop networks. Early work include the popular protocols such as Aloha [1] and binary exponential backoff [10]. We refer the interested readers to [7] for an overview. More recent work include [17], [18], [22], [28] under the stochastic packet injection model and [3], [5], [6] under the adversarial injection model. All these results are obtained by considering the graph-based interference models, where it is assumed that a packet can be successfully disseminated on a channel iff there is only one node transmitting in the network. Although these oversimplified assumptions can help derive protocols with high throughput and low packet latency, the designed protocols may perform dramatically different in practice from theoretical analysis, as they ignore the transmission collisions caused by the interference from non-neighboring transmitters, which occur in the multi-hop setting when considering a global interference model.

In multi-hop networks, local broadcast with continuous packet arrivals has been studied under both the graph-based interference model [29]–[32] and the SINR model [26]. These work however do not consider stable protocols. Instead, they develop randomized algorithms achieving a constant throughput on a jamming channel, where the throughput is defined as the average number of packets a node can receive per round. To the best of our knowledge, there is no known stable results under the multi-hop setting, and this gap is filled by the distributed stable protocol for CLB in multi-hop wireless networks under the SINR model proposed in this paper.

III. Problem Definition and Models

We consider a network of n nodes placed arbitrarily on a plane, possibly in a worst-case fashion. Denote by V the set of nodes, and each node $v \in V$ has a unique identifier ID_v. Note that the IDs are only used for a receiver to recognize a sender. In networks where nodes do not feature any kind of unique identifications, a node can randomly and uniformly choose an ID from $\{1, \ldots, n^3\}$. This can guarantee that each node has a unique ID with a high probability. Time is split into discrete rounds, and the length of a round is long enough to send a message, e.g., a multiple of the $50\mu s$ unit in 802.11, depending on the packet size. Each node is equipped with a half-duplex transceiver, i.e., each node can transmit or listen on the channel at each round, but cannot do both concurrently. Note that the designed algorithms can be easily applied in networks with both half-duplex and full-duplex transceivers equipped.

Interference Model

Nodes communicate on a shared channel in which simultaneous transmissions interfere with each other. We adopt the practical SINR model to depict the interference. Let $d(u,v)$ denote the distance of u and v. A message sent by node u to node v can be correctly received by v if the following SINR formula holds.

$$\frac{\frac{P_u}{d(u,v)^\alpha}}{N + \sum_{w \in V \setminus \{u,v\}} \frac{P_w}{d(w,v)^\alpha}} \geq \beta. \tag{1}$$

In the above inequality, P_u (P_w) denotes the transmission power of node u (w); α is called the path-loss exponent, whose value is normally in the range $(2, 6)$; $\beta > 1$ is the decoding threshold determined by hardware; N denotes the ambient noise; and $\sum_{w \in V \setminus \{u,v\}} \frac{P_w}{d(w,v)^\alpha}$ is the interference experienced by the receiver v caused by all simultaneously transmitting nodes in the whole network.

The *transmission range* R_T of a node u is defined as the maximum distance at which a node v can receive a clear transmission from u when there is no other simultaneous transmissions in the network. From the SINR inequality (1), $R_T = (\frac{P}{\beta \cdot N})^{1/\alpha}$ for the given power level P.

Continuous Packet Local Broadcast

We study the problem of *local broadcast*, i.e., each node disseminates the stored packets to its neighbors within a specified *local broadcast range* R. We consider the continuous scenario, where nodes are injected with packets continuously. Particularly, we take into account both the stochastic injection pattern and the restricted adversarial injection pattern.

In stochastic injection, packets are injected to each node at each round based on a stochastic injection process. Note that we do not set any special injection probability distribution, but assume that the injection probability distribution is identical and independent among different nodes or at different rounds. This assumption is common in related literature such as [23]. The injection rate λ is defined as the expected number of packets injected to a node per round.

For restricted adversarial packet injection, we consider a typical window-type model. Specifically, the packet injection, determined by an adversary, is restricted by the injection rate λ and window size ω as follows: during any continuous interval of ω rounds, the number of packets injected to each node is upper bounded by $\lambda\omega$.

Notations

Given a distance d and a node v, a node u is called a *d-neighbor* of v if $d(u,v) \leq d$. The set of d-neighbors of v is called the *d-neighborhood* of v, denoted by $N_d(v)$. Let $N_d[v] = N_d(v) \cup \{v\}$ and $\Delta = \max\{N_R(v) : v \in V\}$, where R is the local broadcast range.

A set of nodes S is called an *independent set* with respect to a distance d if for each pair of nodes $u, v \in S$, $d(u,v) > d$. An independent set S is *maximal* w.r.t. d if for each node $w \notin S$, there exists a node $v \in S$ such that $d(v,w) \leq d$.

A coloring on a node set is *proper* with respect to a distance d if the nodes in the set with the same color constitute an independent set w.r.t. d.

Some important notations are summarized in Tab. II.

Knowledge of Nodes

We assume that each node knows n and Δ. For the parameter n, a polynomial estimate (with the form of n^c for some constant $c > 0$) suffices as it will not affect the asymptotic bounds on the running time of the algorithms.

In our algorithms, nodes do not need to know the precise values of the SINR parameters α, β, and N. Instead, an upper

TABLE II
NOTATIONS

n	#nodes	$d(u,v)$	distance of u and v
λ	packet injection rate	R_T	transmission range
N	background noise	R	local broadcast range
α	path-loss exponent	Δ	the max #nodes within distance R from any node
β	threshold for successful transmissions	$N_d(v)$	the set of nodes within distance d from v
$N_d[v]$	$N_d(v) \cup \{v\}$	$\mathcal{A}(n)$	SLB algorithm with n nodes

and a lower bound of these parameters suffice, i.e., α_{min} and α_{max}, β_{min} and β_{max}, and N_{min} and N_{max}. This also indicates that our algorithms can work well in a network where the parameters may vary at different regions. For simplicity, in this version of the paper, we perform calculations assuming that exact values of these parameters are known. In order to take into account the uncertainty regarding these parameters, we choose their maximal/minimal values depending on whether their upper or lower estimates are provided.

Preliminaries

We present the preliminary results that will be used in our algorithm analysis. Two sufficient conditions for successful packet disseminations are given in Lemmas 1 and 2.

Lemma 1: Let r_1 and r_2 be distance parameters with $r_1 \geq \left(\frac{\alpha-2}{48\beta(\alpha-1)}\right)^{-\frac{1}{\alpha}} \cdot r_2$ and $r_2 = 2^{-1/\alpha} R_T$. Let S be a set of transmitting nodes that form an independent set w.r.t. r_1. Then each transmission of a node $w \in S$ can be properly received by every node in $N_{r_2}(w)$.

Proof: Since S is an independent set w.r.t. r_1, it satisfies that $d(u,v) > r_1$ for any pair of nodes $u, v \in S$. For a node $w \in S$, we compute the interference experienced by a node $x \in N_{r_2}(w)$. Let C_t be the annulus with distance from w in the range $(tr_1, (t+1)r_1]$ for $t \geq 1$. Without confusion, C_t is also used to denote the set of nodes in C_t. Because any two transmitting nodes are separated by at least r_1, the disks centered at these nodes with radii $r_1/2$ are disjoint. Notice that these disks are in the range $((t-\frac{1}{2})r_1, (t+\frac{3}{2})r_1]$. Then we can get

$$|C_t| \leq \frac{\pi((t+\frac{3}{2})r_1)^2 - \pi((t-\frac{1}{2})r_1)^2}{\pi(\frac{1}{2}r_1)^2} \leq 8(2t+1) \quad (2)$$

The above area argument is demonstrated in Fig. 1. Based on this argument, we can bound the interference at a node $x \in N_{r_2}(w)$ caused by other transmitters in S as follows:

$$I_x = \sum_{y \in S \setminus \{w\}} \frac{P}{d_{yx}^{\alpha}} \leq \sum_{t=1}^{\infty} \frac{N\beta R_T^{\alpha}}{(tr_1)^{\alpha}} \cdot 8(2t+1)$$
$$\leq 24 r_1^{-\alpha} N\beta R_T^{\alpha} \sum_{t=1}^{\infty} t^{-\alpha+1} \leq 24 r_1^{-\alpha} N\beta R_T^{\alpha} \cdot \frac{\alpha-1}{\alpha-2} \leq N. \quad (3)$$

Then according to the SINR condition, x can receive the messages sent by w. □

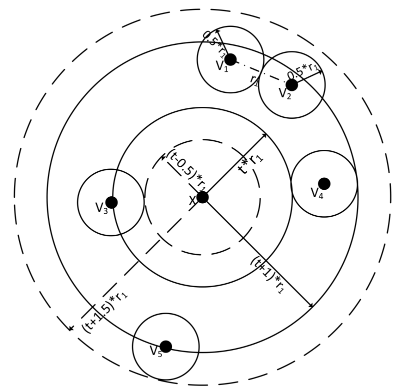


Fig. 1. Area argument: $\{v_1, \ldots, v_5\}$ are transmitting nodes that are separated by a distance larger than r_1; all transmitting nodes are in the annulus between the two solid lines with distance tr_1 and $(t+1)r_1$ to node x; hence, the disks centered at the transmitting nodes with radii $r_1/2$ are disjoint and are located in the annulus between the two dotted lines (with distance $(t-1/2)r_1$ and $(t+3/2)r_1$ from node x).

Our algorithms are randomized, i.e., each node determines whether or not to transmit with a probability. For randomized transmissions, we have the following Lemma 2, which has been implicitly proved by previous work such as [19] (Lemma 4.1 and Lemma 4.2).

Lemma 2: Given that all nodes have the same transmission power $P = \eta N\beta R^{\alpha}$ with constant $\eta > 1$. If for each node v, the transmission probability sum of the nodes in $N_{R/2}(v)$ is bounded by a constant ρ, then, with a constant probability ζ, a node u can send its message to all its R-neighbors if it transmits.

Finally, we present a result proved in [23], which will be used in the analysis of our continuous algorithm.

Lemma 3 [23]: Given two independent non-negative integer random variables X and Y, with X being distributed as follows: X takes only values $-1, 0, i \cdot H + 1$ with $\Pr[X = -1] = q$, $\Pr[X = 0] = 1 - a - q$, and $\Pr[X = i \cdot H + 1] = \frac{a}{1-b} \cdot b^i$, where $b \leq \frac{1}{8}$ and $a < \frac{q}{4H}$. If we have $\Pr[Y \geq k] \leq (1 - \frac{1}{H})^k$ for Y, then this bound also holds for $\max\{Y + X, 0\}$.

IV. Static Local Broadcast

Our continuous local broadcast algorithm employs a static local broadcast (SLB) algorithm as a subroutine. Specifically, in our continuous algorithm, the execution is divided into stages; and at each stage, the nodes intend to disseminate the packets injected in the previous stage. In other words, the packet dissemination at each stage can be seen as a static local broadcast, as the nodes only disseminate packets that have been received before the beginning of the current stage. Formally, we define the SLB problem as follows:

Definition 1 (Static Local Broadcast (SLB)): Given a local broadcast range R. Assume that for each node v, there exist at most $m \leq n$ packets stored at the nodes (a node may possess more than one packet) in $N_R[v]$ initially, and each message transmitted by the nodes can contain one packet. Then the SLB problem seeks to make each node deliver its packets to all its R-neighbors with minimum accomplishment time.

We next present an almost asymptotically optimal algorithm for SLB. At first, we present a version of the algorithm which can disseminate each packet with high probability. And then we adapt the algorithm to be a Las Vegas one such that it can be used in our continuous protocol.

A. A SLB Algorithm Without Physical Carrier Sensing

Our SLB algorithm consists of three stages, with the first two being employed for initialization, in which nodes are first clustered and then the clusters (actually their nodes) are colored. At the third stage, nodes execute a 4-slot scheme to accomplish packet local broadcast. Each cluster consists of a dominator and a number of dominatees; the dominator adjusts the contention within its cluster (the transmission probabilities of its dominatees) and arranges the transmissions of its dominatees. Using a coloring process on the clusters, we design a TDMA-like scheme to coordinate the transmissions of the nodes in neighboring clusters, such that the interference among the neighboring clusters can be avoided. The TDMA-like scheme ensures that when a dominator transmits, all its dominatees can successfully receive the transmission. Hence, the dominator can efficiently control the contention within its cluster and effectively adjust the transmissions of its dominatees, which makes the algorithm achieve a more efficient running time compared to the ones given in previous work. In the following we introduce the algorithm in detail.

Clustering Stage: In this stage, the nodes are assigned to clusters. Each cluster consists of a dominator and the corresponding dominatees. The dominators are selected by letting the nodes compute a Maximal Independent Set (MIS). Specifically, nodes execute the MIS algorithm given in [35] to compute an MIS w.r.t. the local broadcast range R. The length of this stage is set to $\Theta(\log^2 n)$ rounds, which is the time complexity of the MIS algorithm. The nodes in the computed MIS become dominators. Every other node selects the first MIS node from which it has received a controlling message (to stop the competition) as its dominator and joins the corresponding cluster.

We denote by C_u the cluster with dominator u.

Coloring Stage: In this stage, the dominators are properly colored w.r.t. the distance $R_C = (((\alpha-2)/48\beta(\alpha-1))^{-\frac{1}{\alpha}} + 2) \cdot R$. The color of a dominator is also called the color of its cluster. R_C is selected based on Lemma 1, to ensure that when only the nodes in the clusters with the same color transmit and at each cluster there is at most one node transmitting, all transmitters can successfully disseminate their packets within distance R.

The proper coloring is obtained as follows. The whole stage consists of $(\kappa+1)\gamma\log^2 n$ rounds, where κ is an upper bound on the maximum number of dominators within the R_C-neighborhood of each node, and γ is a sufficiently large constant such that $\gamma\log^2 n$ rounds are enough to execute the MIS algorithm in [35] whose running time is $\Theta(\log^2 n)$. Using a similar area argument as that employed by the proof of Lemma 1, one can obtain that $\kappa = (2R_C/R+1)^2$, which is a constant. The coloring stage is then split into $\kappa+1$ phases, and each phase contains $\gamma\log^2 n$ rounds. At each phase, the dominators that have not been colored execute the MIS algorithm in [35] with respect to distance R_C, and the nodes elected into the MIS in phase i get color i.

After this stage, the dominators are colored with colors $1, 2, \ldots, \kappa, \kappa+1$. The dominators then execute a $(\kappa+1)$-round scheme to inform their dominatees of their colors: at round i, the dominators with color i transmit a message containing its assigned color. Lemma 2 ensures that after this informing procedure, each node is aware of the color of its cluster (dominator).

Local Broadcast Stage: In this stage, the nodes execute an algorithm in a TDMA fashion to avoid the interference among neighboring clusters. In particular, time is divided into phases of $\kappa+1$ rounds. At the i-th round of each phase, the nodes with cluster color i execute the 4-slot scheme given in Algorithm 1. The TDMA approach ensures that the nodes whose clusters have the same color execute the algorithm simultaneously.

Each node v stores its packets in a queue Q_v. Each round contains four slots for the execution of the 4-slot scheme: the first slot is used by the dominators to disseminate their packets, while the other three slots are used by the corresponding dominatees, who need to get authorization from their dominators first before packet dissemination. Hence, in the second slot, each dominatee transmits an $AskRight$ message with a specified probability to its dominator; then in the third slot, the dominator sends a $Grant$ message to the dominatee from which it receives the $AskRight$ message; and in the fourth slot, the dominatees receiving authorization in the third slot transmit their messages.

Algorithm 1 4-Slot Scheme

Initially, $p_v = \frac{1}{2n}$; $\kappa = (2R_C/R+1)^2$; ζ is the constant given in Lemma 2;

The 4-slot scheme for a dominator u:

if *Q_u is not empty* **then**
  transmit the first packet in Q_u; discard the transmitted packet from Q_u;
listen;
if *received $AskRight_v$ from a node v in its cluster* **then**
  transmit $Grant_v$;
listen;

The 4-slot scheme for a dominatee v:

listen;
if *Q_v is not empty* **then**
  transmit $AskRight_v$ with probability p_v;
listen;
if *received $Grant_v$ from its dominator* **then**
  transmit the first packet in Q_v; discard the transmitted packet from Q_v;
if *received less than $12\log n$ Grant messages from its dominator and p_v does not change in the past $96\kappa\zeta^{-1}\log n$ rounds* **then**
  $p_v = 2p_v$;

Note that in Algorithm 1 a dominatee v doubles its transmission probability p_v if there are not many message deliveries in its cluster in the past $\Theta(\log n)$ rounds. At this stage, the transmission power of the nodes is set to be $P_L = 2N\beta R^\alpha$. By definition, the transmission range of the nodes is $R_T = 2^{\frac{1}{\alpha}} R$. In Algorithm 1, ζ is the constant given in Lemma 2, and $\kappa = (2R_C/R+1)^2$ defines an upper bound on the number of colors.

B. Analysis on the SLB Algorithm

Analysis of the Clustering and Coloring Stages: We first state the efficiency of the MIS algorithm. The following lemma (Lemma 4) is given in [35].

Lemma 4 [35]: An MIS can be computed in $O(\log^2 n)$ rounds w.h.p.

With the above Lemma 4, we then analyze the algorithm execution of the clustering and coloring stages. Lemma 5 is a direct corollary of Lemma 4 for the algorithm execution of the clustering stage.

Lemma 5: After the clustering stage, w.h.p., each node is assigned to a cluster and the dominators constitute an MIS w.r.t. distance R.

For the coloring stage, we have the following result:

Lemma 6: After the coloring stage, the dominators are colored properly w.r.t. R_C w.h.p., and each dominatee can acquire the color of its dominator.

Proof: For each dominator u, after each phase, either it gets a color or one of its R_C-neighboring dominators is colored by the algorithm. Because each node has at most κ R_C-neighboring dominators, after at most κ phases, either u gets a color or all its R_C-neighboring dominators are colored. If it is the second case, u gets colored in the subsequent phase and joins the MIS. The correctness of the MIS algorithm ensures that the computed coloring is proper.

For the second part of the Lemma, the proper coloring ensures that at each round of the color informing procedure, the transmitting nodes are separated by a distance R_C. According to Lemma 1, these nodes can send their colors to their dominatees simultaneously, since dominatees are R-neighbors of their dominators. □

Analysis on the Local Broadcast Stage: To proceed, we assume that the clustering and coloring stages are correctly executed. In other words, we temporarily ignore the error probabilities in Lemmas 5 and 6, and take them into consideration in the final result.

Lemma 7: For each cluster C_u with dominator u and at each round t,

(i) when the dominator u transmits in the first slot or in the third slot, the transmitted message can be received by all its R-neighbors;

(ii) if a dominatee $v \in C_u$ transmits in the fourth slot, its transmission can be received by all its R-neighbors.

Proof: The TDMA approach ensures that each pair of nodes executing the algorithm concurrently are separated by a distance of at least $R_C - 2R$ if they are in different clusters. By the algorithm, when the dominator u transmits in the first slot or in the third slot and the dominatee v transmits in the fourth slot, they are the only transmitters in C_u. Therefore this lemma holds from Lemma 1. □

We then present the following result, which employs Lemma 2 as a sufficient condition to determine a successful packet dissemination. For a dominator u, let $P(u) = \sum_{v \in C_u} p_v$, i.e., the sum of the transmission probabilities of its dominatees.

Property 1: For any cluster C_u at any round t throughout the execution of the algorithm, $P(u) \leq \frac{1}{2}$.

The proof of Property 1 is given later in Lemma 9.

With Property 1, one can see that the condition in Lemma 2 holds, as the $R/2$-neighbors of u belong to a constant number of clusters. Thus, in the following, we can use Lemma 2 to determine a successful packet dissemination.

Lemma 8: Each node v sends its packets to all its R-neighbors successfully in $O(m + \log^2 n)$ rounds, w.h.p.

Proof: Clearly, we only need to bound the time a dominatee takes to successfully disseminate its packets, as a dominator can successfully disseminate a packet when it transmits in each round, and therefore it takes $O(m)$ rounds for a dominator to accomplish the local broadcast.

Consider a dominatee $v \in C_u$. According to Algorithm 1, within every $96\kappa\zeta^{-1}\log n$ rounds, either in a constant fraction of these rounds u transmits $Grant$ messages or v doubles its transmission probability. For the convenience of analysis, we split the time into substages of $96\kappa\zeta^{-1}\log n$ rounds. By Lemma 7, for each dominatee that has received a $Grant$ message from its dominator, it can locally broadcast a packet in the fourth slot of the same round. Hence, v can receive at most m $Grant$ messages. This means that the number of substages at which the transmission probability of v is not changed is at most $m/12\log n + 1$. Then after $m/12\log n + m/\log n + \log n + 1$ substages, there are at least $m/\log n$ substages at which v attains a constant transmission probability $\frac{1}{2}$. At each round of these substages, v can send an $AskRight$ message to its dominator with constant probability $\zeta/2$ by Lemma 2. Based on the above analysis and Lemma 7, v can disseminate $48\log n$ packets in expectation during each of these substages. Using Chernoff Bound, one can show that v can send $24\log n$ packets with probability $1 - O(n^{-3})$. Then, in the substages with a constant transmission probability, v can send all packets to all its R-neighbors with probability $1 - n^{-2}$, as v has at most m packets to deliver. This result holds for all nodes by the union bound. □

Before presenting the final result, we need to show that Property 1 holds with a high probability.

Lemma 9: Property 1 holds for any cluster with probability $1 - O(n^{-1})$.

Proof: Assume that C_u is the first cluster that violates Property 1 and the round at which the violation happens is t^*. Then before t^* we can still assume that the property holds for all clusters. Next we show a result that is a little bit stronger: for each node v in C_u and at any round t, with probability $1 - O(n^{-3})$, $p_v \leq \frac{1}{2|C_u|}$. Then $\sum_{v \in C_u} \leq \frac{1}{2}$ and there is no such a violation round t^* for C_u with probability $1 - O(n^{-3})$.

Otherwise, assume that at timeslot t, for the first time, there is a node in C_u with a transmission probability larger than $\frac{1}{2|C_u|}$. By Lemma 7, u can send each $Grant$ message to nodes in C_u. Thus nodes in C_u change their transmission probabilities at the same time. Hence, they have the same transmission probability at any round. Then during the interval $I = [t - 96\kappa\zeta^{-1}\log n, t-1]$, it holds that for any node v, $\frac{1}{4|C_u|} < p_v \leq \frac{1}{2|C_u|}$, since each node in C_u only doubles its transmission probability once during I. Before t, nodes in each cluster C_w has a sum of transmission probabilities at most $\frac{1}{2}$.

From the above analysis, one can see that Lemma 2 can still be employed to determine a successful packet dissemination during I. Then if a node $v \in C_u$ transmits, with constant probability ζ, u can successfully receive this message. Then the probability that u can receive an $AskRight$ message from the nodes in C_u is at least $\sum_{v\in C_u} p_v \cdot \zeta \geq |C_u| \cdot \frac{1}{4|C_u|} \cdot \zeta = \zeta/4$.

During I, u can receive at least $96\kappa\zeta^{-1}\log n \cdot \frac{1}{\kappa} \cdot \frac{\zeta}{4} = 24\log n$ $AskRight$ messages in expectation. Then using Chernoff bound, the probability that u receives less than $12\log n$ $AskRight$ messages is at most $e^{\frac{1}{8}\cdot 24\log n} = n^{-3}$. Thus with probability at least $1 - n^{-3}$, each node can receive at least $12\log n$ $Grant$ messages during I. Then all nodes in C_u at round t do not change their transmission probabilities, which contradicts the assumption.

Finally, we bound the number of such potential violation rounds for u. Based on the above analysis, we know that before any potential violation round, there are $\Omega(\log n)$ nodes in C_u that can receive a $Grant$ message and successfully perform a local broadcast. Thus there are at most $O(\frac{m}{\log n}) \in O(\frac{n}{\log n})$ potential violation rounds. Therefore with probability $1 - O(n^{-2})$, there is no such a violation time for u. And the claim is true for all clusters with probability $1 - O(n^{-1})$. $\square$

Based on Lemmas 5, 6, and 8, we obtain the following result characterizing the algorithm performance.

Theorem 2: The proposed SLB algorithm can make all nodes accomplish packet local broadcast in $O(m + \log^2 n)$ rounds w.h.p.

C. Adapting to a Las Vegas Algorithm

In our continuous algorithm that will be given in the next section, each node needs to deterministically figure out whether a transmitted packet is disseminated to all its R-neighbors. For this purpose we need to adapt the static algorithm to a Las Vegas one, i.e., the algorithm can successfully disseminate the packets or inform about the failures of packet disseminations.

We add an explicit acknowledgement step into our SLB algorithm. Specifically, the 4-Slot Scheme in Algorithm 1 is expanded to a 5-Slot one, as shown in Algorithm 2. The difference of these two algorithms lies in the last slot: by listening in the fourth slot, a dominator u determines whether the authorized dominatee v has successfully delivered a packet by detecting whether or not the interference exceeds a threshold $\mathcal{T} = 4^{-\alpha} \cdot N$; if true, it ensures that the transmission of v is also received by all v's R-neighbors. Thus u sends back an Ack message to v in the last slot. The dominatee discards a packet only after receiving the Ack message from its dominator.

Analysis: The difference of the Las Vegas algorithm and the SLB algorithm lies in that it needs to ensure not only successful packet dissemination but also that the interference

Algorithm 2 5-Slot Scheme

Initially, $p_v = \frac{1}{2n}$; $\kappa = (2R_C/R + 1)^2$; ζ is the constant given in Lemma 2;

The 5-slot scheme for a dominator u**:**

if *Q_u is not empty* **then**
  transmit the first packet in Q_u; discard the transmitted packet from Q_u;
listen;
if *received $AskRight_v$ from a node v in its cluster* **then**
  transmit $Grant_v$;
listen;
if *received a message fron node v and detects that the interference is at most $\mathcal{T}$* **then**
  transmit Ack_v;

The 5-slot scheme for a dominatee v**:**

listen;
if *Q_v is not empty* **then**
  transmit $AskRight_v$ with probability p_v;
listen;
if *received $Grant_v$ from its dominator* **then**
  transmit the first packet in Q_v;
listen;
if *received ACK_v* **then**
  discard the transmitted packet from Q_v;
if *received less than $12\log n$ $Grant$ messages from its dominator and p_v does not change in the past $96\kappa\zeta^{-1}\log n$ rounds* **then**
  $p_v = 2p_v$;

at the dominator does not exceed the threshold $\mathcal{T}$. Using a very similar argument as that for proving Lemma 2, we can get the following result:

Lemma 10: Assume that for each node v in the network, the transmission probability sum of the nodes in $N_{R/2}(v)$ is upper bounded by a constant ρ. Then if a node u transmits, with some constant probability ζ_1, u can send its message to all its R-neighbors, and the interference at u's dominator is at most $\mathcal{T}$.

The following Lemma 11 states that the threshold $\mathcal{T}$ based condition is sufficient to determine a successful packet dissemination.

Lemma 11: When a dominator u receives a message from one of its dominatees v and detects that its experienced interference is not larger than $\mathcal{T}$, the transmission of v can be received by all v' R-neighbors.

Proof: Let X be the set of nodes transmitting simultaneously with v. By the setting of $\mathcal{T}$, the transmitters other than v have a distance at least $4R$ from u. Then for each node $w \in N_R(v)$ and each $x \in X$, we have $d(x,w) \geq d(x,u) - d(u,w) \geq d(x,u) - 2R \geq \frac{1}{2}d(x,u)$. Then the interference at w can be upper bounded as follows:

$$\sum_{x\in X} \frac{P_L}{d(x,w)^\alpha} \leq 2^\alpha \cdot \sum_{x\in X} \frac{P_L}{d(x,u)^\alpha} \leq 2^\alpha \cdot \mathcal{T} \leq N. \quad (4)$$

By the SINR condition defined in Eq. (1), it can be shown that w can successfully receive the transmission of v. $\square$

Similar to Lemma 7, it can be shown that when a dominator transmits an Ack message, all its dominatees can successfully receive the message; and Lemma 11 ensures that when a dominatee receives an Ack message from its dominator, the dominatee must have successfully disseminated a packet to all its R-neighbors. In other words, Algorithm 2 is a Las Vegas one. Then based on the above two Lemmas and an argument similar to that in the proof of Lemma 8, one can show that each node can disseminate all its packets in $O(m + \log^2 n)$ rounds w.h.p. Thus, the following result holds.

Theorem 3: There exists a Las Vegas algorithm that can make each node disseminate all packets in $O(m + \log^2 n)$ rounds w.h.p.

V. Stable Algorithm for CLB

We present our stable algorithm for CLB in this section. The algorithm is articulated and analyzed w.r.t. the stochastic packet injection, and the performance of the protocol under the adversarial packet injection will be discussed in Sec. V-D. At the end of this section, we give an upper bound on the maximum injection rate and a lower bound on the minimum packet latency with which a stable protocol can attain, to show the asymptotic optimality of our proposed protocol.

The basic idea of the algorithm is sketched as follows: the execution is divided into stages of a designated length; and at each stage, the packets injected in the past stages are disseminated by making the nodes execute the SLB algorithm in two consecutive phases, in which by elaborately setting the length of each phase one can ensure that the algorithm can disseminate the packets injected in the past stages while maintaining the stability of its execution. Indeed, we give a general framework on how to construct a CLB algorithm based on a SLB algorithm. However, the maximum packet injection rate under which the obtained stable protocol can attain is determined by the efficiency of the SLB algorithm. Our proposed SLB algorithm ensures that the maximum stable injection rate that can be attained by the obtained CLB algorithm is asymptotically optimal. Furthermore, the length of the stages determines the packet latency in the algorithm. On one hand, each packet needs to wait for the beginning of the next stage to be disseminated, and therefore the packet latency is lower bounded by the length of a stage in the worst case. This requires to set the stage as short as possible. On the other hand, if the length of the stage is set to be too short, the variance of the number of packets injected to the nodes may become too large, leading to the scenario with too many packets injected within a stage to be processed in time. In our study, we set the length of each stage to be $\Theta(\Delta + \log^2 n)$, with which it can be shown that the obtained packet latency is asymptotically optimal.

Though the basic idea of the algorithm is not very complicated, the analysis on the stability, throughput, and packet latency is non-trivial, as both the variance during the stochastic packet injection and the failures of the algorithm execution are necessary to be carefully counted.

In sequel, we denote our Las Vegas algorithm for SLB (Algorithm 2) given in the last section as $\mathcal{A}(n)$, where n is the number of nodes in the network. Furthermore, we assume that the time complexity of the Las Vegas algorithm is $\mu(m + \log^2 n)$ when the number of packets in each node's R-neighborhood is bounded by m, where μ is an upper bound of the constant hidden behind the big O notation of the running time in Theorem 3.

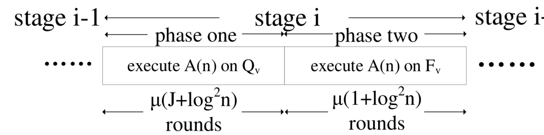


Fig. 2. A stage of the CLB protocol.

A. Stable Algorithm

We consider the packet injection rate λ that is in the range $(0, (1-\epsilon)\mu^{-1} \cdot (\Delta+1)^{-1}]$. Here ϵ can be set to be any constant in the range $[\frac{1}{2}, 1)$. To simplify our analysis, we set $\epsilon = \frac{1}{2}$. As shown later (Thoerem 5), the maximum injection rate we consider is asymptotically optimal.

We define the following parameters that will be used in the algorithm. Let $\phi = 5(\ln((1+\epsilon)^{1+\epsilon}/e^{\epsilon}))^{-1}$. We set $T = \psi(\Delta + \log^2 n)$ for some constant ψ, such that $T \geq \max\{2(1-\epsilon)^{-1}\mu \cdot \phi \ln(\phi(1+\epsilon)), 60(1-\epsilon)^{-1}\mu \ln(30(1+\epsilon))\}$ and $T \geq \max\{(1-\epsilon^2)^{-1}\mu\Delta, \frac{3}{\epsilon^2}\mu \log^2 n\}$. Let $J = (1+\epsilon)\lambda(\Delta+1)T$. It can be verified that $J \geq \Delta$.

The algorithm is divided into stages and the length of each stage is T. Each stage consists of two phases: the first phase is used to disseminate the packets injected in the last stage, and the second phase is used for the dissemination of the packets that are not successfully disseminated so far. Fig.2 illustrates one stage of the algorithm execution.

In the algorithm, each node v stores the packets it needs to disseminate in three sets: S_v, Q_v, and F_v. Specifically, v stores the packets that are just injected into S_v. At the beginning of each stage, v moves the packets in S_v into Q_v, which includes all the packets that are to be disseminated during the first phase of the stage. The first phase consists of $\mu(J + \log^2 n)$ rounds, in which v executes $\mathcal{A}(n)$ on the packets in Q_v. After the first phase, v moves the packets still in Q_v into F_v. In other words, F_v contains the packets that are not successfully disseminated during the first phases of the past stages. The second phase contains the remaining rounds in the stage. At the beginning of this phase, each node v selects the first packet in F_v with probability $1/\Delta$. If v selects a packet, it then executes $\mathcal{A}(n)$ for $\mu(1 + \log^2 n)$ rounds. If v successfully disseminates a packet to its R-neighbors, it discards the packet from F_v.

At each stage, as shown later, almost all packets in Q_v can be successfully disseminated during the first phase, and only very few packets are moved to F_v. Hence, after the first phase, the queue length $|F_v|$ for each node v does not significantly increase. Then in the second phase, it can be shown that the queue length $|F_v|$ of each node v can be effectively decreased. Combining these two aspects together, we can conclude that the protocol is stable. We next detail the performance analysis.

B. Performance Analysis

We can obtain the following result (Theorem 4) that states the stability, throughput, and packet latency of the

CLB algorithm. Note that Theorem 4 is a direct corollary of Lemma 15 and Lemma 16, to be presented later.

Theorem 4: The proposed CLB algorithm is stable w.r.t. packet injection at a rate up to $\Omega(\frac{1}{\Delta})$. The expected packet latency is $O(\Delta + \log^2 n)$.

Stability and Throughput Analysis: We first show that the algorithm for CLB is stable w.r.t. the considered injection rate $\lambda \leq (1-\epsilon)\mu^{-1} \cdot (\Delta+1)^{-1} \in \Omega(\frac{1}{\Delta})$, i.e., the expected size of $|F_v|$ for each node v is bounded at any time. The proof contains two parts: 1) bounding the increase on the expected size of F_v after the first phase of each stage; and 2) showing that the increase can be handled in the second phase.

We consider the first part of the proof. For an arbitrary node v, let m_i be the number of packets injected to the nodes in $N_R[v]$ at stage $i-1$. These packets are those that are moved to the Q_u's (the set of packets to be disseminated in the first phase of stage i) for $u \in N_R[v]$ at the beginning of stage i.

Lemma 12: For any $\delta > 0$, $\Pr[m_i \geq (1+\delta)\lambda(\Delta+1)T] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\lambda(\Delta+1)T}$.

Proof: For a node $u \in N[v]$, let I_u^t be the number of packets injected to u at round t. Denote by $\mathcal{R}$ the rounds of stage $i-1$. Then $m_i = \sum_{t\in\mathcal{R}}\sum_{u\in N_R[v]} I_u^t$. Because the packet injections on the nodes are independent, and the injections to the same node at different rounds are also independent, the expected number of packets injected to the nodes in $N_R[v]$ can be bounded as follows:

$$E[m_i] = E[\sum_{t\in\mathcal{R}}\sum_{u\in N_R[v]} I_u^t] = \sum_{t\in\mathcal{R}}\sum_{u\in N_R[v]} E[I_u^t] \leq \lambda(\Delta+1)T.$$

Using Chernoff bound, we get

$$\Pr[m_i \geq (1+\delta)\lambda(\Delta+1)T] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\lambda(\Delta+1)T}$$

□

Lemma 13: For each node v, during any stage i, the probability that the expected size of F_v increases by at least $r \cdot J + 1$ is at most J^{-4-r} for any non-negative integer r.

Proof: We distinguish the proof into two cases.

Case 1 ($r = 0$): In this case, we need to bound the probability that at least one packet is added to F_v after the first phase of stage i. Denote by P_1 this probability. There are two reasons for F_v to grow: first, the number of packets in Q_v exceeds the bound that the algorithm $\mathcal{A}(n)$ can handle in the first phase,[2] i.e., $m_i > J$; second, the inherent failure probability of the randomized algorithm is non-negligible.

In the following we consider the first increasing reason.

At first, we lower bound $\lambda(\Delta+1)T$ and claim that $\lambda(\Delta+1)T \geq \phi \cdot \ln J$. Using the inequality $x \geq 2\ln x$ for $x > 0$, we have

$$\begin{aligned}\frac{\lambda(\Delta+1)T}{\phi} &\geq \frac{\lambda(\Delta+1)T}{2\phi} + \ln\left(\frac{\lambda(\Delta+1)T}{\phi}\right)\\ &\geq \ln(\phi(1+\epsilon)) + \ln\left(\frac{\lambda(\Delta+1)T}{\phi}\right) = \ln J. \quad (5)\end{aligned}$$

[2]Notice that the first phase contains $\mu(J + \log^2 n)$ rounds. By Theorem 3 and the setting of constant μ, the algorithm $\mathcal{A}(n)$ works if $m_i \leq J$.

Now we use Lemma 12 to bound the probability that the event $m_i > J$ occurs as follows:

$$\begin{aligned}\Pr[m_i > J] &= \Pr[m_i > (1+\epsilon)\lambda(\Delta+1)T]\\ &\leq \left(\frac{e^\epsilon}{(1+\epsilon)^{1+\epsilon}}\right)^{\lambda(\Delta+1)T} \leq \left(\frac{e^\epsilon}{(1+\epsilon)^{1+\epsilon}}\right)^{\phi\ln J}\\ &= J^{-5} \quad (6)\end{aligned}$$

We next consider the second increasing reason. As stated, by appropriately tuning the constant parameters used in the Las Vegas algorithm, its failure probability can be upper bounded by $\frac{1}{2}n^{-5} \leq \frac{1}{2}J^{-4}$.

Combining the above analysis, we can conclude that $P_1 \leq J^{-5} + \frac{1}{2}J^{-4} \leq J^{-4}$.

Case 2 ($r > 0$): In this case, we only need to bound the occurrence probability of the first increasing reason.

By the value of T and using a similar argument as that in Eq. (5), one can obtain that $\lambda(\Delta+1)T \geq 30\ln J$.

Then according to Lemma 12 and Chernoff bound, we have

$$\begin{aligned}\Pr[m_i > rJ] &\leq \Pr[m_i \geq r(1+\epsilon)\lambda(\Delta+1)T]\\ &\leq e^{-\frac{1}{3}(r(1+\epsilon)-1)\cdot\lambda(\Delta+1)T}\\ &\leq e^{-\frac{1}{3}(r(1+\epsilon)-1)\cdot 30\ln J} \leq J^{-4-r}. \quad (7)\end{aligned}$$

□

We next consider the decrease of $|F_v|$ in the second phase.

Lemma 14: For a node v at the second phase of a stage i, if F_v is non-empty, the probability that the expected number of packets in F_v decreases by 1 is at least $1/2e(\Delta+1)$.

Proof: We consider the decrease of $|F_v|$ in the case that: 1) v selects a packet in F_v; and 2) other nodes in $N_R[v]$ do not select packets. The probability that v selects a packet is $\frac{1}{\Delta+1}$, and the probability that other nodes in $N_R[v]$ do not select packets is $(1-\frac{1}{\Delta+1})^\Delta \geq \frac{1}{e}$. Then, with probability at least $\frac{1}{e(\Delta+1)}$, v is the only node in $N_R[v]$ that selects a packet to transmit. Therefore, with probability at least $1/2$, v can successfully disseminate the selected packet by executing algorithm $\mathcal{A}(n)$. Combining the above analysis, we conclude that the lemma holds. □

Now we are ready to prove the stability of the algorithm.

Lemma 15: For each node v, the expected size of F_v is bounded at any time for any injection rate $\lambda \leq (1-\epsilon)\mu^{-1} \cdot (\Delta+1)^{-1}$.

Proof: In Lemma 3, by setting $a = J^{-4}$, $b = J^{-1}$, $q = \frac{1}{2e(\Delta+1)}$, $H = J$, it can be obtained that $\Pr[|F(v)| \geq k] \leq (1-J^{-1})^k$ at any time t by Lemma 13 and Lemma 14. Then,

$$\begin{aligned}E[|F(v)|] &\leq \sum_{k=1}^{\infty} k \cdot \Pr[|F(v)| \geq k-1]\\ &\leq \sum_{k=1}^{\infty} k \cdot (1-J^{-1})^{k-1} = J^2,\end{aligned}$$

which completes the proof. □

Packet Latency Bounding:

Lemma 16: The expected packet latency is $O(\Delta + \log^2 n)$.

Proof: We consider an arbitrary packet $\mathcal{P}$ that is injected to a node v. Assume that $\mathcal{P}$ is injected to v at stage i. Then according to the algorithm, $\mathcal{P}$ is moved to Q_v at the beginning of stage $i+1$. If $\mathcal{P}$ is successfully disseminated during the first phase of stage $i+1$, clearly, the packet latency is upper bounded by $2T \in O(\Delta+\log^2 n)$. Otherwise, $\mathcal{P}$ is moved to F_v and then disseminated in the second phases of the subsequent stages.

Let Z be 1 if $\mathcal{P}$ is moved to F_v and 0 otherwise. Let D be the number of stages used for $\mathcal{P}$ to be disseminated. Then we have

$$\begin{aligned} E[D] &= \Pr[Z=1]E[D|Z=1] + \Pr[Z=0]E[D|Z=0] \\ &\le \Pr[Z=1]E[D|Z=1] + 2. \end{aligned}$$

We next bound $\Pr[Z = 1]E[D|Z = 1]$. As shown in Lemma 13, $\Pr[Z=1] \le J^{-4}$. Now we condition on $Z=1$, i.e., $\mathcal{P}$ is added to F_v. Let X be the number of packets in F_v before $\mathcal{P}$. By Lemma 14, from stage $i+1$, there is a packet in F_v being disseminated at each of the subsequent stages with probability $\frac{1}{2e(\Delta+1)}$. Then in expectation, $\mathcal{P}$ can be disseminated after $2e(\Delta+1)\cdot\ (E[X]+1) \le 2e(\Delta+1)\cdot E[|F_v|] \le 2e(\Delta+1)\cdot J^2$ stages.

Combining the above analysis together, we can bound $E[D]$ as follows:

$$\begin{aligned} E[D] &\le J^{-4}\cdot 2e(\Delta+1)\cdot J^2 + 2 \\ &= 2e(\Delta+1)\cdot J^{-2} + 2 \le 4eJ^{-1} + 2. \end{aligned}$$

The last inequality holds by $J \ge \Delta$. Then the expected delay of $\mathcal{P}$ is $E[D]\cdot T \le\ (4eJ^{-1}+2)T \in O(T) \in O(\Delta+\log^2 n)$. $\square$

C. Optimality of the Proposed Algorithm

With the following result, one can show that our proposed algorithm is asymptotically optimal in terms of both injection rate and packet latency.

Theorem 5: For any stable local broadcast protocol, the maximum injection rate it can handle is $O(\frac{1}{\Delta})$, and the minimum packet latency it can attain is at least $\Omega(\Delta+\log^2 n)$.

Proof: For each node v, it can receive at most one packet from its R-neighborhood within one round. Hence, the sum of the number of injected packets to the nodes of $N_R[v]$ is at most 1 at each round such that a stable protocol can be obtained. It follows that the injection rate of each node is at most $\frac{1}{\Delta+1}$.

In [21], it is shown that to make v receive packets from each of its R-neighbors, it takes at least $\Omega(\Delta+\log^2 n)$ rounds to succeed with a high probability guarantee, which is a lower bound of the packet latency. $\square$

D. Adversarial Packet Injection

Our CLB protocol can also handle adversarial packet injection. The key point to ensure the stability of our protocol is that the number of injected packets within a stage should be upper bounded by the estimate J. For adversarial injection, by bounding the injection rate to be $\lambda \le (1-\epsilon)\mu^{-1}(\Delta+1)^{-1}$ for constant $\epsilon = \frac{1}{2}$, the number of injected packets at each stage can be upper bounded. Then using a similar argument as that for the stochastic injection case, we can get the following result.

Theorem 6: The proposed CLB algorithm is stable w.r.t. the adversarial packet injection at a rate of up to $\Omega(\frac{1}{\Delta})$. The expected packet latency is upper bounded by $O(\Delta+\log^2 n)$.

TABLE III
PARAMETER SETTINGS IN THE SIMULATION STUDY

Parameter	value	Parameter	value	Parameter	value
# of nodes n	2500	Δ	≈ 500	R	40
R_T	$2^{1/\alpha}R$	R_C	$2R$	α	3.0
β	1.5	N	10		

Proof: We first consider the stability of the algorithm for injection rate $\lambda \le (1-\epsilon)\mu^{-1}(\Delta+1)^{-1}$. Recall that for an arbitrary node v, m_i denotes the the number of packets injected to the nodes in $N_R[v]$ at stage $i-1$. With a similar approach as that in the proof of Lemma 12, we can bound m_i as follows:

$$m_i = \sum_{t\in\mathcal{R}}\sum_{u\in N_R[v]} I_u^t \le \lambda(\Delta+1)T < J. \quad (8)$$

Based on the above bound, using a similar argument as that in Lemma 13, one can conclude that during any stage, the probability at which the expected size of F_v increases by at least 1 is at most J^{-5}, and the probability at which the expected size of F_v increases by at least $r\cdot J+1$ for a positive integer r is 0. Hence, Lemma 13 also holds in the adversarial injection case. Then based on similar arguments as those used in Lemma 14 and Lemma 15, one can prove the stability of the protocol under the stated injection rate.

For packet latency, notice that the proof in Lemma 16 does not depend on any packet injection pattern; thus one can get the same asymptotic packet latency bound in the case of adversarial packet injection. $\square$

VI. Simulation Results

We study the empirical performances of our continuous algorithm in this section. Specifically, we investigate (i) the range of the injection rate λ for the protocol to be stable; (ii) the influence of the injection rate λ on the queue lengths of the nodes, i.e., the length of $|F_v|$ for each node v; and (iii) the influence of the injection rate λ on the packet latency. The performances of the algorithm under uniform, normal, and exponential node distributions are evaluated. Both the stochastic and the adversarial packet injection patterns are considered. Particularly, for the stochastic packet injection case, the packets injected to the nodes are generated using a normal distribution.

We also investigate the performances of the proposed static algorithm from two aspects: (i) the performance improvement over existing ones; and (ii) the influence of SINR parameters on the algorithm's performance.

The simulations are conducted in a network whose nodes can be deployed in a square area of 150 by 150. The default settings are given in Tab. III. Over 20 runs of the simulation have been carried out for each reported result.

All experiments are conducted on a Linux machine with Intel Xeon CPU E5-2670@2.60GHz and 64 GB main memory, implemented in C++ and compiled by g++ compiler.

A. CLB Protocol

The lengths of the first and the second phases at each stage of the CLB protocol are set as 8000 and 320, respectively. To better compare the performances of our protocol with the

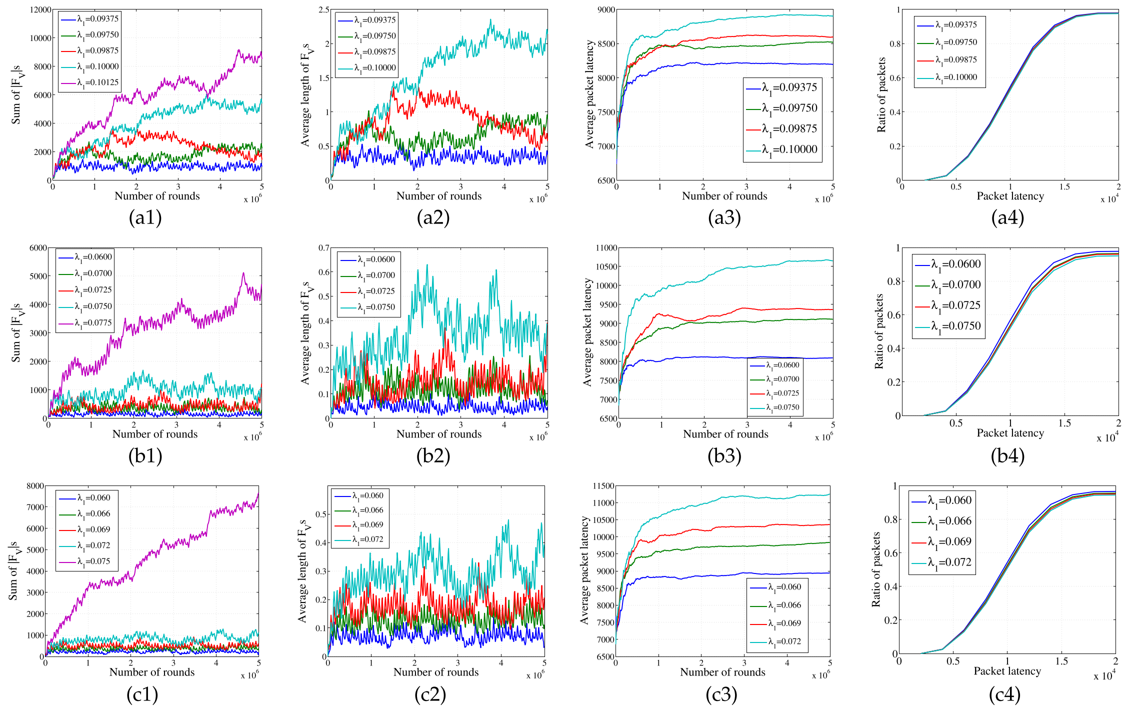


Fig. 3. Performance evaluations of the CLB Protocol under the stochastic packet injection pattern. We consider three types of node distributions: (a) Uniform; (b) Normal, and (c) Exponential, and four aspects of performances: (1) Stability; (2) Average queue length; (3) Average packet latency; and (4) Packet latency distribution. Subfigures (a1)-(a4), (b1)-(b4), and (c1)-(c4) illustrate the simulation results under the uniform, normal, and exponential node distributions, respectively.

optimal solution, we scale the injection rate by a parameter $\Delta+1$, and let $\lambda_1 = (\Delta+1)\lambda$. It is easy to see that λ_1 is an upper bound on the sum of the injection rates of a node's R-neighbors. As shown in the proof of Theorem 5, for any stable protocol, $\lambda \leq \frac{1}{\Delta+1}$. Hence, $\lambda_1 \leq 1$ holds for an optimal stable protocol. We call an injection rate stable if the protocol is stable under the packet injection with the particular injection rate.

The simulation results of the CLB algorithm under the stochastic and the adversarial packet injection patterns are illustrated in Fig. 3 and Fig. 4, respectively. We next analyze the simulation results in more detail.

1) Stochastic Packet Injection: In Fig. 3, the simulation results under the uniform, normal, and exponential node distributions are illustrated in Subfigures (a1-a4), (b1-b4), and (c1-c4), respectively.

Range of injection rates for stability: The stabilities of the protocol under different node distributions are illustrated in Subfigures (a1), (b1), and (c1) in Fig. 3. In each of these subfigures, the x-axis represents the number of rounds the algorithm has been executed. To better illustrate the stability of the protocol, in the y-axis, we count the total number of packets in the nodes' queues ($\sum_{v\in V}|F_v|$) at each round, which clearly upper bounds the maximum queue length of the nodes during the algorithm execution in our simulation. We execute the algorithm for 5×10^6 rounds. The subfigure (a1) demonstrates that under the uniform node distribution, the protocol is stable when the injection rate $\lambda_1 \leq 0.10000$, but is not stable for $\lambda = 0.10125$. Hence the maximum injection rate λ_1 for the protocol to be stable is in the range $[0.10000, 0.10125)$. Hence, our protocol is inferior to the optimal solution by a factor of less than 10 in the case of uniform node distribution. From Subfigures (b1) and (c1), it can be seen that the maximum stable injection rates λ_1 are in the ranges $[0.0750, 0.0775)$ and $[0.072, 0.075)$ under the normal and exponential node distributions, respectively, which are a bit smaller than that in the uniform distribution case. This is mainly because networks of larger density tend to be generated in the normal and exponential cases compared to uniform node distribution case. Consequently, the network suffers from more interference and conflicts.

Influence of injection rates on queue length: Subfigures (a2), (b2), (c2) in Fig. 3 illustrate the average lengths of the nodes' queues for the three different node distributions. In all these cases, the average queue length is smaller than 2.5, which is quite small. Furthermore, the subfigures also exhibit the tradeoff between injection rate and average queue length: the higher the injection rate is, the larger the average queue length will be.

Tab. IV gives a detailed distribution of the nodes' queue lengths in a round when the network is in a stable state. In the table, the rates of nodes with queue lengths $|F_v|$ in different ranges under different packet injection patterns and different node distributions are illustrated. As shown in the table, under all cases, 90% of the nodes have a queue length less than 5. Furthermore, we observe that the maximum queue length of the nodes is no more than 20.

Subfigures (a2), (b2), (c2) of Fig. 3 and Tab. IV indicate that almost all packets are disseminated in the first phase of each stage.

Influence of injection rates on packet latency: Subfigures (a3), (b3), (c3) in Fig. 3 illustrate the average packet latency as the algorithm executes under three different node distributions. In these subfigures, the x-axis and y-axis

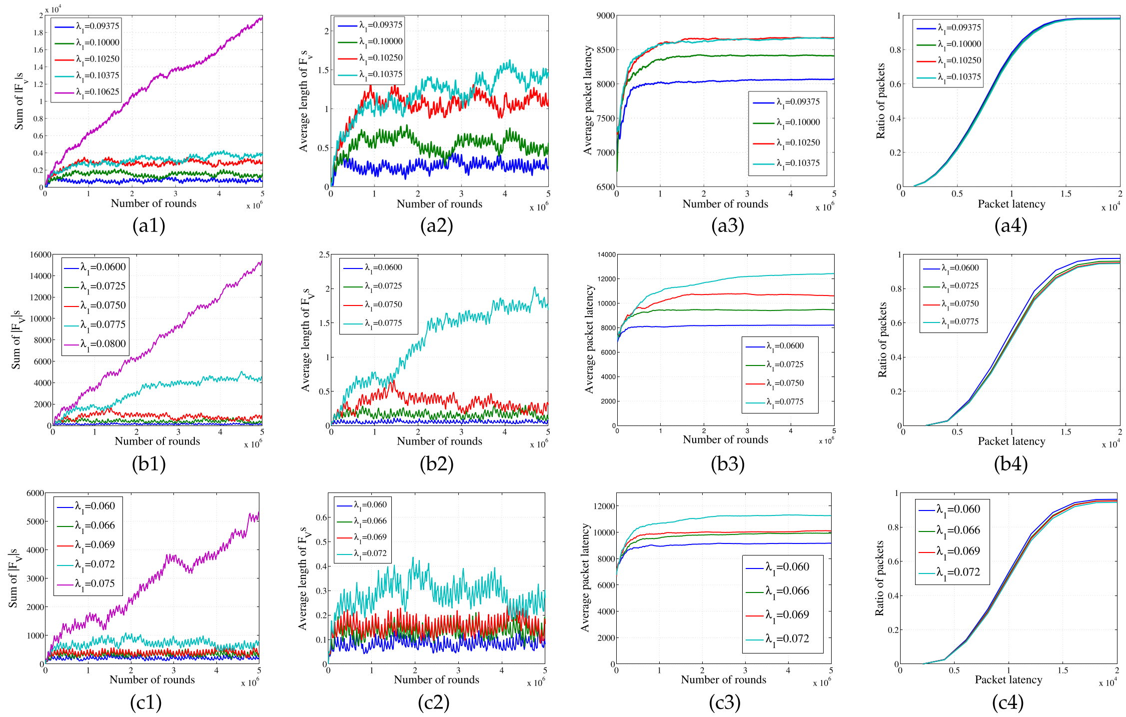


Fig. 4. Performance evaluations of the CLB Protocol under the adversarial packet injection. Similar to Fig. 3, subfigures (a1)-(a4), (b1)-(b4), and (c1)-(c4) illustrate the simulation results under the uniform, normal, and exponential node distributions, respectively.

TABLE IV

DISTRIBUTION OF THE QUEUE LENGTH UNDER DIFFERENT PACKET INJECTION PATTERNS/RATES AND NODE DISTRIBUTIONS

Case	Value of λ_1	Value of $\|F_v\|$			
		$=0$	$\in[1,2]$	$\in[3,5]$	>5
Stochastic injection, Uniform distribution.	0.09375	0.844	0.116	0.040	0.000
	0.09750	0.776	0.131	0.065	0.028
	0.09875	0.675	0.141	0.115	0.069
	0.10000	0.653	0.147	0.091	0.109
Stochastic injection, Normal distribution.	0.0600	0.950	0.048	0.002	0
	0.0700	0.850	0.136	0.014	0
	0.0725	0.820	0.155	0.023	0.002
	0.0750	0.777	0.182	0.037	0.004
Stochastic injection, Exponential distribution.	0.0600	0.896	0.099	0.005	0
	0.0660	0.898	0.089	0.013	0
	0.0690	0.881	0.106	0.013	0
	0.0720	0.838	0.128	0.031	0.003
Adversarial injection, Uniform distribution.	0.09375	0.848	0.112	0.040	0.000
	0.10000	0.778	0.130	0.064	0.028
	0.10250	0.676	0.142	0.114	0.068
	0.10375	0.656	0.148	0.088	0.108
Adversarial injection, Normal distribution.	0.0600	0.944	0.056	0	0
	0.0725	0.872	0.117	0.011	0
	0.0750	0.836	0.134	0.028	0.002
	0.0775	0.615	0.197	0.085	0.103
Adversarial injection, Exponential distribution.	0.0600	0.917	0.081	0.002	0
	0.0660	0.868	0.124	0.008	0
	0.0690	0.877	0.115	0.008	0
	0.0720	0.797	0.168	0.031	0.004

represent the number of rounds that the algorithm has been executed and the average packet latency, respectively. From the subfigures, we observe that the average packet latency is in the ranges of $[8200, 8900]$, $[8000, 10700]$, and $[8800, 11300]$, respectively, for the three distributions, which is about the length of one stage in the protocol. This is consistent with our theoretical analysis. The subfigures also show that a higher injection rate results in a larger packet latency.

Subfigures (a4), (b4), (c4) from Fig. 3 illustrate the distributions of the packet latency in a round when the network is stable. As shown in these subfigures, one can see that about 95% of the packets under the uniform distribution and 90% of the packets under the normal and exponential distributions have latency less than 15000, which is the length of around 2 stages. Because each packet starts to be disseminated in the subsequent stage after it is injected according to our algorithm, such a latency indicates that most packets are successfully disseminated in the first stage at which it starts to be disseminated.

2) Adversarial Packet Injection: The simulation results of the CLB protocol under the adversarial packet injection model are illustrated in Fig. 4, in which Subfigures (a1-a4), (b1-b4), and (c1-c4) report the simulation results under the uniform, normal, and exponential node distributions, respectively. Furthermore, similar to Fig 3, in Fig 4, the stabilities of the protocol under different node distributions are illustrated in Subfigures (a1), (b1), and (c1); Subfigures (a2), (b2), (c2) demonstrate the average lengths of the nodes' queues under the three different node distributions; the average packet latencies as the algorithm executes under the three different node distributions are illustrated in Subfigures (a3), (b3), and (c3), and Subfigures (a4), (b4), (c4) report the distributions of the packet latency in a round when the network is stable. Similarly as in the stochastic packet injection case, in Tab. IV, we present a detailed distribution of the nodes' queue lengths within a round when the network is in a stable state. Specifically, the queue lengths $|F_v|$ under different packet injection patterns/rates and different node distributions when the protocol is stable are illustrated.

From the subfigures in Fig. 4 and Tab. IV, we obtain similar observations and conclusions as those in the stochastic packet injection case. But it is worthy of pointing out that the protocol can attain a larger stable injection rate under

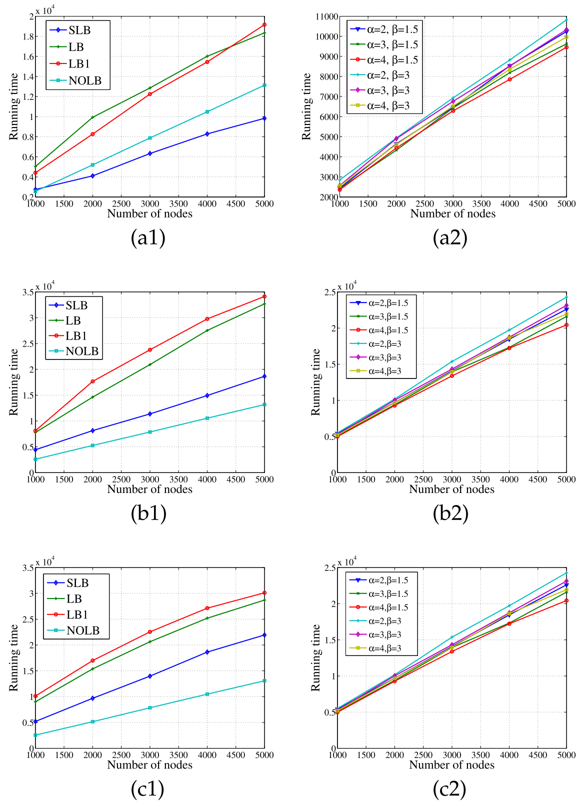


Fig. 5. Performance evaluation of the SLB algorithm. We consider three types of node distributions: (a) Uniform distribution; (b) Normal distribution; and (c) Exponential distribution.

the case of adversarial packet injection. This is because in the stochastic injection case, the injection rate refers to the expected packet injection speed, which implies that the probability under which more packets are injected is not zero, while in the adversarial case, the injection rate is the tight upper bound of the injection speed.

In summary, the simulation results reported in Fig.3 and Fig. 4 demonstrate that the protocol is stable for both stochastic and adversarial packet injections with an injection rate up to $\Omega(\frac{1}{\Delta+1})$ under three different typical node distributions, and attains asymptotically optimal packet latency. These simulation results corroborate our theoretical analysis. Furthermore, our simulation study also shows that the queue lengths of the nodes are much smaller than the analyzed bound.

B. SLB Algorithm

The simulation results for our SLB protocol are shown in Fig. 5, in which the Subfigures (a1-a2), (b1-b2), and (c1-c2) illustrate the simulation results under the uniform, normal, and exponential node distributions, respectively.

In Subfigures (a1), (b1), and (c1) of Fig. 5, we compare the performance of our static algorithm with the following three best known existing ones: LocalBroadcast1 (LB1) [21], LocalBroadcast (LB) [34], and Nearly Optimal Local Broadcast (NOLB) [4]. Because the existing algorithms only consider the case under which each node has exactly one packet to disseminate, we conduct the simulation study under this setting. It can be seen that our algorithm is at least twice as fast as the LB and LB1 algorithms. NOLB achieves a bit better performance than our algorithm under the normal and exponential distributions, but it needs a strong assumption that nodes can get free feedback on its transmissions, which is impractical in real-world wireless networks. Furthermore, we observe that the constant hidden behind the big O in the running time of our algorithm is around 13.

In Subfigures (a2), (b2), and (c2) of Fig. 5, we investigate the sensitivity of our algorithm to the SINR parameters. One can see that the algorithm has almost the same running time across different settings of α and β. In other words, our algorithm is insensitive to these two parameters.

VII. Conclusion

In this paper, we initiate the study of the properties of stable protocols under the SINR model for continuous local broadcast in multi-hop networks. We propose a protocol that is asymptotically optimal w.r.t. throughput (packet injection rate) and packet latency, the two most important metrics measuring the performance of stable protocols. Our protocol can handle both stochastic and adversarial injection patterns. Furthermore, our static algorithm, which is used as a subroutine in the continuous protocol, is of independent interest. It is also asymptotically optimal, which closes the $O(\log n)$ gap between the upper and lower bounds as stated in previous work.

As a basic primitive of information exchange, local broadcasting deserves more research effort that can lead to a better understanding of the desirable performance of stable protocols for this type of or similar fundamental operations. One meaningful future research direction is to consider the proposed idea under some graph-based potential applications in dynamic networks such as uncertain networks [11] or random networks [12].

References

[1] N. Abramson, "Development of the ALOHANET," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 2, pp. 119–123, Mar. 1985.

[2] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "On the complexity of radio communication," in *Proc. STOC*, 1989, pp. 274–285.

[3] L. Anantharamu, B. S. Chlebus, D. R. Kowalski, and M. A. Rokicki, "Deterministic broadcast on multiple access channels," in *Proc. INFOCOM*, Mar. 2010, pp. 1–5.

[4] L. Barenboim and D. Peleg, "Nearly optimal local broadcasting in the SINR model with feedback," in *Proc. SIROCCO*, 2015, pp. 164–178.

[5] M. A. Bender, M. Farach-Colton, S. He, B. C. Kuszmaul, and C. E. Leiserson, "Adversarial contention resolution for simple channels," in *Proc. SPAA*, 2005, pp. 325–332.

[6] M. A. Bender, J. T. Fineman, S. Gilbert, and M. Young, "How to scale exponential backoff: Constant throughput, polylog access attempts, and robustness," in *Proc. SODA*, 2016, pp. 636–654.

[7] B. S. Chlebus, "Randomized communication in radio networks," in *Handbook Randomized Computing*. Dordrecht, The Netherlands: Kluwer, 2001, pp. 401–456.

[8] A. Cornejo, N. Lynch, S. Viqar, and J. L. Welch, "Neighbor discovery in mobile ad hoc networks using an abstract MAC layer," in *Proc. Allerton*, Sep. 2009, pp. 1460–1467.

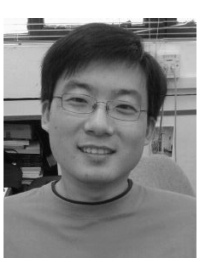
[9] A. Cornejo, S. Viqar, and J. L. Welch, "Reliable neighbor discovery for mobile ad hoc networks," *Ad Hoc Netw.*, vol. 12, pp. 259–277, Jan. 2014.

[10] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Commun. ACM*, vol. 19, no. 7, pp. 395–404, 1976.

[11] L. Fu *et al.*, "Determining source–destination connectivity in uncertain networks: Modeling and solutions," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3237–3252, Dec. 2017.

[12] L. Fu, X. Wang, and P. R. Kumar, "Are we connected? Optimal determination of source–destination connectivity in random networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 751–764, Apr. 2017.

[13] F. Fuchs and D. Wagner, “Arbitrary transmission power in the SINR model: Local broadcasting, coloring and MIS,” *Comput. Sci.*, vol. 46, pp. 252–259, Feb. 2014.
[14] R. G. Gallager, “A perspective on multiaccess channels,” *IEEE Trans. Inf. Theory*, vol. IT-31, no. 2, pp. 124–142, Mar. 1985.
[15] M. Ghaffari, B. Haeupler, N. Lynch, and C. Newport, “Bounds on contention management in radio networks,” in *Proc. DISC*, 2012, pp. 223–237.
[16] M. Ghaffari, E. Kantor, N. Lynch, and C. Newport, “Multi-message broadcast with abstract MAC layers and unreliable links,” in *Proc. PODC*, 2014, pp. 56–65.
[17] L. A. Goldberg, M. Jerrum, S. Kannan, and M. Paterson, “A bound on the capacity of backoff and acknowledgment-based protocols,” *SIAM J. Comput.*, vol. 33, no. 2, pp. 313–331, 2004.
[18] L. A. Goldberg, P. D. MacKenzie, M. Paterson, and A. Srinivasan, “Contention resolution with constant expected delay,” *J. ACM*, vol. 47, no. 6, pp. 1048–1096, 2000.
[19] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer, “Local broadcasting in the physical interference model,” in *Proc. DIALM-POMC*, 2008, pp. 35–44.
[20] M. M. Halldórsson, S. Holzer, and N. Lynch, “A local broadcast layer for the SINR network model,” in *Proc. PODC*, 2015, pp. 129–138.
[21] M. M. Halldórsson and P. Mitra, “Towards tight bounds for local broadcasting,” in *Proc. FOMC*, 2012, Art. no. 2.
[22] J. Håstad, T. Leighton, and B. Rogoff, “Analysis of backoff protocols for multiple access channels,” *SIAM J. Comput.*, vol. 25, no. 4, pp. 740–774, 1996.
[23] T. Kesselheim, “Dynamic packet scheduling in wireless networks,” in *Proc. PODC*, 2012, pp. 281–290.
[24] M. Khabbazian, D. R. Kowalski, F. Kuhn, and N. Lynch, “Decomposing broadcast algorithms using abstract MAC layers,” *Ad Hoc Netw.*, vol. 12, pp. 219–242, Jan. 2014.
[25] C. Newport, “Consensus with an abstract MAC layer,” in *Proc. PODC*, 2014, pp. 66–75.
[26] A. Ogierman, A. Richa, C. Scheideler, S. Schmid, and J. Zhang, “Competitive MAC under adversarial SINR,” in *Proc. INFOCOM*, Apr. 2014, pp. 2751–2759.
[27] G. Pei and A. K. S. Vullikanti, “Distributed approximation algorithms for maximum link scheduling and local broadcasting in the physical interference model,” in *Proc. INFOCOM*, Apr. 2013, pp. 1339–1347.
[28] P. Raghavan and E. Upfal, “Stochastic contention resolution with short delays,” in *Proc. STOC*, 1995, pp. 229–237.
[29] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, “A jamming-resistant MAC protocol for multi-hop wireless networks,” in *Proc. DISC*, 2010, pp. 179–193.
[30] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, “Competitive and fair medium access despite reactive jamming,” in *Proc. ICDCS*, Jun. 2011, pp. 507–516.
[31] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, “Self-stabilizing leader election for single-hop wireless networks despite jamming,” in *Proc. MobiHoc*, 2011, pp. 1–15.
[32] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, “Competitive and fair throughput for co-existing networks under adversarial interference,” in *Proc. PODC*, 2012, pp. 291–300.
[33] J. Yang, H. Wang, J. Ding, X. Gao, and Z. Ding, “Spectral and energy efficiency analysis for massive MIMO multi-pair two-way relaying networks under generalized power scaling,” *Sci. China Inf. Sci.*, vol. 60, no. 10, p. 102303, 2017.
[34] D. Yu, Q.-S. Hua, Y. Wang, and F. C. M. Lau, “An O(log n) distributed approximation algorithm for local broadcasting in unstructured wireless networks,” in *Proc. DCOSS*, May 2012, pp. 132–139.
[35] D. Yu, Y. Wang, Q.-S. Hua, and F. C. M. Lau, “Distributed local broadcasting algorithms in the physical interference model,” in *Proc. DCOSS*, Jun. 2011, pp. 1–8.



Dongxiao Yu received the B.Sc. degree from the School of Mathematics, Shandong University, in 2006, and the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, in 2014. He is currently an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology. His research interests include wireless networks and distributed computing.

Yifei Zou received the B.E. degree from the Computer School, Wuhan University, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science, The University of Hong Kong. His research interests include wireless networks, ad hoc networks, and distributed computing.

Jiguo Yu (SM’17) received the Ph.D. degree in operational research and control theory from Shandong University, Jining, China, in 2004. Since 2007, he was a Full Professor with the School of Information Science and Engineering, Qufu Normal University, Jining. His main research interests include wireless ad hoc and sensor networks, cognitive radio networks, and distributed computing. He is interested in designing and analyzing algorithms for many computationally hard problems in communication networks and systems. He is a member of the ACM and a senior member of the China Computer Federation.

Xiuzhen Cheng (F’15) received the M.S. and Ph.D. degrees in computer science from the University of Minnesota–Twin Cities in 2000 and 2002, respectively. She is currently a Professor with the Department of Computer Science, The George Washington University, Washington, DC, USA. Her current research interests include privacy-aware computing, wireless and mobile security, cyber physical systems, mobile computing, and algorithm design and analysis. She is a member of the ACM. She received the NSF CAREER Award in 2004. She has chaired several international conferences. She was a Program Director of the U.S. National Science Foundation in 2006 (full-time) and from 2008 to 2010 (part-time). She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops.

Qiang-Sheng Hua received the B.Eng. and M.Eng. degrees from the School of Information Science and Engineering, Central South University, China, in 2001 and 2004, respectively, and the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, China, in 2009. He is currently an Associate Professor with the Huazhong University of Science and Technology, China. He is interested in parallel and distributed computing, including algorithms and implementations in real systems.

Hai Jin received the Ph.D. degree from the Huazhong University of Science and Technology in 1994. He was a Post-Doctoral Fellow at the University of Southern California and The University of Hong Kong. He is currently a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology. His research interests include HPC, grid computing, cloud computing, and virtualization.

Francis C. M. Lau (SM’03) received the Ph.D. degree in computer science from the University of Waterloo. He is currently a Professor in computer science with The University of Hong Kong. His research interests include computer systems, networks, programming languages, and the application of computing in arts. He is a Senior Member of the IEEE Computer Society. He is the Editor-in-Chief of the *Journal of Interconnection Networks*.