Exact Implementation of Abstract MAC Layer via Carrier Sensing

Dongxiao Yu*, Yong Zhang†, Yuyao Huang*, Hai Jin*, Jiguo Yu‡ and Qiang-Sheng Hua*§

* Services Computing Technology and System Lab, Big Data Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, P.R. China
Email: {dxyu,M201672844,hjin,qshua}@hust.edu.cn
†Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, P.R. China
Email: zhangyong@siat.ac.cn
‡School of Information Science and Engineering, Qufu Normal University, Rizhao, 276826, P.R. China
Email: jiguoyu@sina.com

Abstract*—In this paper, we present the first algorithm for exactly implementing the abstract MAC (absMAC) layer in the physical SINR model. The absMac layer, first presented by Kuhn et al. in [15], provides reliable local broadcast communication, with timing guarantees stated in terms of a collection of abstract delay functions, such that high-level algorithms can be designed in terms of these functions, independent of specific channel behavior. The implementation of absMAC layer is to design a $distributed$ algorithm for the local broadcast communication primitives over a particular communication model that defines concrete channel behaviors, and the objective is minimizing the bounds of the abstract delay functions. Halldórsson et al. [10] have shown that in the standard SINR model (synchronous communication, without physical carrier sensing or location information), there cannot be efficient *exact* implementations. In this work, we show that physical carrier sensing, a commonly seen function performed by wireless devices, can help get efficient exact implementation algorithms. Specifically, we propose an algorithm that exactly implements the absMAC layer. The algorithm provides asymptotically optimal bounds for both $acknowledgement$ and $progress$ functions defined in the absMAC layer. Our algorithm can lead to many new faster algorithms for solving high-level problems in the SINR model. We demonstrate this by giving algorithms for problems of Consensus, Multi-Message Broadcast and Single-Message Broadcast. It deserves to point out that our implementation algorithm is designed based on an optimal algorithm for a General Local Broadcast (GLB) problem, which takes the number of distinct messages into consideration for the first time. The GLB algorithm can handle much more communication scenarios apart from those defined in the absMAC layer. Simulation results show that our proposed algorithms perform well in reality.

I. Introduction

In distributed algorithm domain, most wireless algorithms nowadays are directly for the physical network. This way efficiently avoids the problem faced by the traditional manner assuming message-passing models (reliable message exchange between neighbors in each round) that realistic performance may dramatically differ from theoretical analysis. But on the other hand, algorithm designers have to consider issues of message dissemination at higher levels together with contention management at the physical level. Consequently, algorithm design and analysis becomes extremely complicated, even for simple tasks, prohibiting elegant solution studies for complex high-level problems.

To overcome the difficulty, Kuhn et al. [15] proposed a new approach, the *abstract MAC layer* service, which expresses key guarantees of real MAC layers with respect to local broadcast. These guarantees include two message delivery latency bounds: the *acknowledgement* bound f_{ack} that is the time for a sender's message to be received by all its neighbors, and the *progress* bound f_{prog} that is the time for a receiver to receive one message when there is at least one neighbor sending. The absMAC layer decomposes wireless algorithm design and analysis into two independent and manageable pieces, i.e., implementing the absMAC layer over a physical network and solving higher-level problems based on the local broadcast services and time guarantees provided by the absMAC layer. Benefiting from the absMAC layer approach, many new efficient algorithms developed for some fundamental problems, such as Single-Message Broadcast and Multiple-Message Broadcast [10], [13], [8], and Consensus [19].

In this work ,we focus on the low-level task, i.e., implementing the absMAC layer over concrete communication models. Obviously, the reality of the physical network model where the absMAC layer is implemented in determines how well the designed higher-level algorithms perform when deploying in real networks. Recently, Halldórsson et al. [10] made efforts to implement the absMAC layer in the Signal-to-Interference-plus-Noise (SINR) model. The SINR model well depicts the accumulated and fading natures of physical interference, and hence is more realistic than graph-based models commonly used in wireless algorithm studies. However, because of the defined global interference, it is very hard to directly design and analyze local distributed algorithms for higher-level problems. Halldórsson et al. showed that absMAC layer can help mask the complexity and hence the algorithm can be designed in an easier way. But they also showed that it is

§The corresponding author is Qiang-Sheng Hua (qshua@hust.edu.cn).

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

impossible to get exact efficient implementation algorithms (on the communication graph) under the standard SINR model (synchronous communications, without physical carrier sensing or location information), and they tackled the difficulty by introducing *approximate implementation*, by implementing the absMAC layer on a graph constructed by connecting each pair of nodes within a distance smaller than the communication range. In other words, the approximate implementation can only ensure the message dissemination in a range smaller than the communication range defined by the SINR model. Clearly, this approximate implementation may greatly increase the dissemination delay of messages in the network. A natural question will then be: *is it possible to get efficient "exact" implementation of the absMAC layer with the facilitation of some functions commonly possessed by wireless devices?* We answer this question affirmatively by employing the function of physical carrier sensing in the algorithm design.

In this work, instead of directly focusing on the two local broadcast primitives defined in the absMAC layer, we intend to solve a more general local broadcast problem defined as follows.

Definition 1 (General Local Broadcast): We are given a network with n nodes. Each node has a message that it wants to deliver to all its neighbors within the communication range R (refer to the definition in Sec. III). The number of distinct messages at nodes in the neighborhood of each node is upper bounded by a parameter k (k can be a non-constant). The problem is to make each node receive all distinct messages stored at neighbors, and the objective is minimizing the accomplishment time.

The GLB problem generalizes the local broadcast problem studied in previous work, which assumes each node has a distinct message. With this generalization, we can study the key factor that determines the time for local broadcast: the number of distinct messages or the local contention. Clearly, the acknowledgement and the progress bounds correspond to the accomplishment time of GLB in the cases of $k = \Delta$ and $k = 1$ respectively, where Δ is the maximum number of neighbors of nodes.

Our contributions are summarized as follows. The algorithms are randomized ones, whose performances are guaranteed with high probability (w.h.p.), i.e., with probability $1 - n^{-c}$ for some constant $c > 0$.

- We present a randomized distributed algorithm for GLB, with running time of $O(k + \log n)$. The algorithm is asymptotically optimal comparing with the lower bound $\Omega(k+\log n)$ [20]. Our algorithm shows that it is the number of distinct messages that determines the complexity of locally broadcasting messages, rather than the contention around a node as illustrated in previous local broadcast results.
- Using the GLB algorithm as subroutine, we propose the first algorithm for exactly implementing the absMAC layer. The algorithm is asymptotically optimal in terms of both acknowledgement and progress bounds. Comparing with the result in [10], our algorithm not only improves the delay bounds, but also needs less global information.[1]
- Based on the implementation algorithm, faster algorithms for solving several fundamental problems are given, including Consensus, Multiple-Message Broadcast (MMB) and Single-Message Broadcast (SMB).

The comparisons of our results with previous ones are summarized in Table I.

The remaining part of the paper is organized as follows. In Section II, some closely related results will be introduced. The network model and definitions will be given in Section III. The GLB algorithm and the absMAC layer implementation algorithm will be presented in Section IV and Section V, respectively. Simulation results on these two algorithms will be analyzed in Section VI. We illustrate the applications of our absMAC layer implementation in Section VII. The whole paper will be concluded in Section VIII.

II. Related Work

Abstract MAC Layer. The absMAC layer was proposed by Kuhn et al. in [15]. Thereafter, several variants of the basic absMAC layer model have been proposed for different deployment scenarios, such as the conditional absMAC layer [4], the enhanced absMAC layer [8] and the probabilistic absMAC layer [13]. Based on the abstraction of absMAC layer, several fundamental problems have been studied and efficient algorithms were proposed, such as Single-Message Broadcast, Multiple-Message Broadcast [10], [13], [8], and Consensus [19].

For the implementation of absMAC layers, basic implementations of a probabilistic absMAC layer were given by Khabbazian et al. [13] using the classical Decay strategy and in [14] using Analog Network Coding. Lynch and Newport studied the implementation problem in a network model considering unreliable links in [18]. But all these implementation algorithms are devised under the graph-based models, where the interference is oversimplified to be a local and binary phenomenon. Halldórsson et al. first studied the implementation of probabilistic absMAC layer under the SINR model in [10]. They first gave a negative result which implies that it is impossible to get efficient implementations in the standard SINR model (synchronous communication, without physical carrier sensing or location information), and then presented algorithms for approximate implementation.

Local Broadcast. The local broadcast problem has been extensively studied in both graph-based radio network models [1], [5], [7], [24] and the SINR model [3], [11], [9], [20], [21], [22], [25]. All these results assume that each node have a distinct message, and do not consider the impact of distinct message number on local broadcast time. In the SINR model, the best known result is $O(\Delta \log n + \log^2 n)$ time [11], [20], which improves to $O(\Delta + \log^2 n)$ [11] with free acknowledgments. When Δ is known, this was further

[1] The algorithm in [10] needs to know the ratio Λ of the communication range and the minimum range between any two nodes, which is not needed in this work.

TABLE I

Task/Bound	Lower bound	Upper bound presented here	Upper bound in [10]†
f_{ack}	$\Omega(\Delta + \log n)$ [20]	$O(\Delta + \log n)$	$O(\Delta(\log n + \log \Lambda) + \log \Lambda(\log n + \log \Lambda))$
f_{prog}	$\Omega(\log n)$ [20]	$O(\log n)$	$O(\Delta(\log n + \log \Lambda) + \log \Lambda(\log n + \log \Lambda))$
f_{approg}	—	—	$O((\log^\alpha \Lambda + \log^* n) \log \Lambda \log n)$
GLB$^+$	$\Omega(k + \log n)^*$	$O(k + \log n)$	—
CONSENSUS	—	$O(D(\Delta + \log n))$	$O(D(\Delta + \log \Lambda)(\log n + \log \Lambda))$
MMB$^{(\sharp)}$	$\Omega(D \log(\frac{n}{D}) + k \log n + \log^2 n)^\ddagger$	$O(D \log n + k(\Delta + \log^2 n + \log n \log k))$	$O(D \log^{\alpha+1} \Lambda + k(\Delta + polylog(nk\Lambda))(\log n + \log k))$
SMB$^{(\S)}$	$\Omega(D \log(\frac{n}{D}) + \log^2 n)^\ddagger$	$O(D \log n + \log^2 n)$	$O((D + \log n) \log^{\alpha+1} \Lambda)$

† f_{approg} is the delay bound for approximate progress defined in [10]; Notice that the results for SMB, MMB and CONSENSUS are given based on the approximate implementation in [10], and hence the diameter D is larger than that in our result; Λ is the ratio of the communication range and the minimum distance between any two nodes, which can be exponential in n; $+$ GLB is General Local Broadcast; $*$ Combination of trivial lower bound k and the one given in [20]; § SMB is Single-Message Broadcast; $\sharp$ MMB is Multi-Message Broadcast; ‡ Combinations of lower bounds of [2], [6], [16] for graph based models.

improved recently to $O(\Delta + \log n \cdot \log \log n)$ in the spontaneous setting [3].

III. Model and Problem Definition

We consider a network of n nodes, which are deployed on a plane arbitrarily. The time is divided into synchronous rounds, each of which might contain a constant number of slots. Each node is equipped with a half-duplex transceiver. This means that in each slot, a node can only transmit or listen on the shared channel, but cannot do both. For two nodes u, v, let $d(u, v)$ denote the distance between u and v.

Interference. The Signal-to-Interference-plus-Noise-Ratio (SINR) model is used to depict the interference between concurrent transmissions. Specifically, it defines that a receiver v can successfully receive a message transmitted by node u if

$$SINR_{u,v} = \frac{P_u/d(u,v)^\alpha}{N + \sum_{w \in S \setminus \{u\}} P_w/d(w,v)^\alpha} \geq \beta. \quad (1)$$

where P_u (P_w) is the transmission power of u (w), N is the ambient noise, $\beta \geq 1$ denotes the minimum signal to interference ratio required for decoding a message, and $\alpha > 2$ is the path loss exponent.

We assume the *uniform* power assignment, i.e., all nodes take the same transmission power P. Uniform power assignment is one of the most common power assignments in practice. The nodes can perform physical carrier sensing, i.e., they can detect the interference when listening on the channel. And when a node receives a message, it knows the distance of the transmitter.

By $R_T = (P/\beta N)^{1/\alpha}$ we denote the *transmission range*, i.e. the maximum distance at which two nodes can communicate assuming no other nodes are sending at the same time. Because the transmission has to tolerate some amount of interference, the links that are used for communication have to be 'stronger' than those defined by the transmission range. Hence, we define a *communication range* R with $R = (1 - \epsilon) R_T$, where $\epsilon \in (0, 1)$ is a constant determined by the environment. We consider only communications between nodes within distance R. Strong connectivity is a common assumption in related literatures [10], [12], [23]. Furthermore, we define a constant $\hat{\epsilon}$ such that $(1 + 2\hat{\epsilon})R = (1 - \frac{\epsilon}{2})R_T$. $\hat{\epsilon}$ will be used in the algorithm design and analysis.

With the communication range R, we can define the *communication graph* $G = (V, E)$, by connecting each pair of nodes within distance R. Let D denote the diameter of G.

Abstract MAC layer. As introduced before, there have been several models presented for absMAC layer, satisfying both deterministic and randomized algorithm studies. Because we focus on randomized solutions, here we adopt the probabilistic absMAC layer model [13], [10], which is defined for the communication graph $G = (V, E)$ and provides acknowledgement and progress primitives for communications in G.

The abstract MAC layer provides an interface to higher layer with input $bcast(m)_i$ and outputs $ack(m)_i$, $rcv(m)_i$ for any node $i \in V$ and message $m \in M$. When a node $u \in V$ broadcasts a message m, the model delivers the message to all its neighbors in G. If all neighbors of u receive the message, the abstract MAC returns an acknowledgement $ack(m)_u$ to higher layer informing that the broadcasting of u is completed. Similarly, it returns $rcv(m)_v$ for higher layer that v receives message m. The model provides two timing bounds, the *acknowledgement bound* f_{ack} and the *progress bound* f_{prog}. In particular, the acknowledgement bound guarantees each node's broadcast can be completed and acknowledged within f_{ack} time. The progress bound bounds the time for a node to receive a message when there is at least one neighbor sending. More formally, let $(u, v) \in E$ and u broadcasts a message m during an interval of length f_{prog}. It ensures that v receives some message (not necessarily m) during the interval. f_{prog} is much smaller comparing with f_{ack} [10], [13]. Further details about the definition of the absMAC layer and motivations for these delay bounds, please refer to [8], [13], [15] .

In the probabilistic absMAC layer, two parameters ξ_{prog} and ξ_{ack} are defined to indicate the error probabilities for satisfying the delay bounds f_{prog} and f_{ack}, respectively. In particular, the MAC layer guarantees that progress is made with probability $1 - \xi_{prog}$ within f_{prog} time, and with probability $1 - \xi_{ack}$ the absMAC layer correctly outputs an acknowledgment within f_{ack} time steps. In this paper, we require that the progress and acknowledgement primitives can be accomplished with high probability, i.e., $\xi_{prog}, \xi_{ack} \in n^{-c}$ for some constant $c > 0$.

General Local Broadcast. Our implementation of absMAC layer is based on an algorithm for solving the GLB problem as defined in Definition 1. Considering the reality that in an ad hoc network, it is hard to make each node know how many distinct messages stored at its neighbors, we studied the GLB problem in the harsh scenario that the parameter k or any nontrivial upper bound on k is unknown to nodes, though this harsh case is clearly more challenging for algorithm design.

Notations. For a node v, a node u is called a r-neighbor of v if $d(u,v) \le r$. Denote by $N_c(v)$ the set of v's cR-neighbors. Specifically, if two nodes are R-neighbors, i.e., they are neighbors on the communication graph, we simply say they are neighbors. A set of nodes is called an r-independent set if for any pair of nodes $u, v \in S$, $d(u,v) > r$. And, if for any node $w \notin S$, there exists a node $u \in S$, such that $d(w,u) \le r$, then S is called an r-maximal independent set.

IV. GLB Protocol

In this section, we present an algorithm for solving the GLB problem. This algorithm will be used as a subroutine in the absMAC layer implementation given later.

A. Algorithm

The GLB algorithm is given in Algorithm 1.

In the algorithm, in each round t, every node v holds a probability $p_t(v)$ with which it determines to transmit on the channel. Let $D_c(v)$ be the disk with radius cR that is centered at v. Denote by $P_t(v)$ be the sum of the transmission probabilities of nodes in $D_{\hat{\epsilon}/2}(v)$, i.e., $P_t(v) = \sum_{u \in N_{\hat{\epsilon}/2}(v)} p_t(u)$. We will also consider the contention in a larger region. Specifically, let $P_t^{\rho}(v)$ be the sum of transmission probabilities of nodes in $N_{\rho}(v)$ for a constant $\rho > 1$ that will be specified later.

Basically, in the algorithm, each node uses an exponential backoff manner to adjust its transmission probability, based on detected interference. With the exponential backoff adaption strategy, it can be shown that for a node v, it satisfies in most rounds that 1) (*Bounded Contention.*) $P_t^{\rho}(v) \le g$ for specified constants ρ and g, and 2) (*Bounded Interference*) the expected interference at v is upper bounded by T for a specified constant T. In each of these rounds, it can be then shown that a successful local broadcast occurs with some constant probability guarantee.

In the algorithm, each round consists of three slots: the first one is for message transmission (Slot $\mathcal{T}$); the second one is for acknowledgement transmission (Slot $\mathcal{A}$), to inform the transmitter that it has successfully perform local broadcast within distance $(1+\hat{\epsilon})R$; and the third one is for a successful transmitter (that received the ack in Slot $\mathcal{A}$) to inform nearby nodes (within distance $\hat{\epsilon}R$) with the same message to halt.

In Slot $\mathcal{T}$ of a round, every node v determines to transmit with a probability $p_t(v)$. $p_t(v)$ is doubled after the slot if v senses an idle channel and halves in other cases. A node detects a *busy* channel if the interference it sensed exceeds the threshold $T_b = P \cdot (\hat{\epsilon}R)^{-\alpha}$.

Define $\mathcal{E}$ as the event that a node v receives a message from a neighbor u with $d(u,v) \le \hat{\epsilon}R$ and detects that the interference is upper bounded by $T_a = \min\{(3/4)^{\alpha} \cdot ((1-\frac{\epsilon}{2})^{-\alpha}-1)N, P \cdot (4R(1+2\hat{\epsilon}))^{-\alpha}\}$. In Slot $\mathcal{A}$ of a round, every node v that has received a message will try to transmit an ack message to the transmitter if $\mathcal{E}$ occurs. In this case, it can be shown that the message is also received by $(1+\hat{\epsilon})R$-neighbors of the transmitter. And if the received message is the same with v's own, it does not need to transmit by itself, as all its neighbors has received this message. Hence, v will halt after the slot.

In Slot $\mathcal{I}$ of a round, every node v that transmitted in Slot $\mathcal{T}$ and received an ack message in Slot $\mathcal{A}$ transmits again. This transmission is to inform the nodes within distance $\hat{\epsilon}R$ that have the same message with v of stopping the execution. Because in the stated case, v's message has been received by all its neighbors within distance $(1+\hat{\epsilon})R$, i.e., the R-neighbors of v's $\hat{\epsilon}R$-neighbors has received the message, v's $\hat{\epsilon}R$ neighbors are no longer necessary to transmit, if they possess the same message.

Other parameters in Algorithm 1 are set as follows: $p_{max} = \frac{8\hat{\epsilon}^2\hat{p}}{\rho^2}$ with $\hat{p} = \log_4 \frac{10}{9}$ and ρ is a sufficiently large constant determined in the analysis; $p_{min} = \frac{1}{4n}$;$\gamma \ge 40/(p_{max} \cdot \kappa_1)$, where $\kappa_1 = \frac{1}{2} \cdot (1/4)^g$ and g is sufficiently large constant determined in the analysis.

B. Analysis

We next analyze the time complexity of the algorithm.

Overview. Let $\bar{I}_t^{\rho}(v)$ be the expected interference at node v that are caused by nodes outside $N_t^{\rho}(v)$, i.e., $\bar{I}_t^{\rho}(v) = \sum_{u \notin N_t^{\rho}(v)} S_{u,v} \cdot p_t(u)$, where $S_{u,v} = P/d(u,v)^{\alpha}$.

Basically, we first show that in most rounds of an interval $I \in \Omega(\log n)$, there exists constants ρ and g such that for each node v, the bounded contention and bounded interference properties hold. Formally, in most rounds, 1) $P_t^{\rho}(v)$ is upper bounded by a constant g and 2) $\bar{I}_t^{\rho}(v)$ is upper bounded by T, where $T = \min\{P(4(1+2\hat{\epsilon})R)^{-\alpha}, \frac{P((1+\hat{\epsilon})R)^{-\alpha}-\beta N}{2\beta(4/3)^{\alpha}}, T_a \cdot 2^{-1}(\frac{1+\hat{\epsilon}}{1+\frac{5}{8}\hat{\epsilon}})^{-\alpha}, T_b/10\}$. If the bounded contention and bounded interference properties hold for a node v in a round t, the round t is called *good* for v.

We then divide the algorithm execution into phases of $\gamma \log n$ rounds. The phases are further classified into two types: *high* and *low*. A phase is called *high* if in at least $1/10$-fraction of the good rounds, $P_t^{\rho}(v) \ge \hat{p}$; Otherwise, the phase is called *low*. We say a node *successfully transmits* if its transmission is received by all nodes within distance $(1+\hat{\epsilon})R$ in Slot $\mathcal{T}$ of a round t and it receives an *ack* message in the subsequent Slot $\mathcal{A}$. With a proved sufficient condition, it can be shown that for each node v, 1) in a high phase, there are $\Omega(\log n)$ nodes in $N_{\rho}(v)$ successfully transmitting and making nodes within distance $\hat{\epsilon}R$ that have the same message stop executing the algorithm, and 2) v will halt in a low phase. Hence, with 1), we can get that there are at most $O(k/\log n)$ high phases, as all nodes in $N_{\rho}(v)$ have stopped the execution after these phases, and then a low phase emerges by the end of which v halts. Hence, the total running time for each node is $O(k + \log n)$. Formally, we have the following result.

Theorem 1: For each node v, w.h.p., its message will be received by all its neighbors within distance R in $O(k+\log n)$ rounds.

We next give the detailed analysis for the time complexity and prove Theorem 1 at the end of this section.

Detailed Analysis. We first show a result on the contention balancing strategy. This result shows that for each node, most

Algorithm 1: GLB

Initially, $p_0(v) = p_{max}$;
In round t, v does:
In Slot $\mathcal{T}$
Let $X \leftarrow 1$ or 0 with probability $p_t(v)$ and $1 - p_t(v)$, respectively.
if $X = 1$ **then**
    Transmit the message; $p_{t+1}(v) \leftarrow \max\{\frac{p_t(v)}{2}, p_{min}\}$;
else
    Listen to the channel:
    if *detected* busy *channel* **then**
        $p_{t+1}(v) \leftarrow \max\{\frac{p_t(v)}{2}, p_{min}\}$;
    else
        $p_{t+1}(v) \leftarrow \min\{2p_t(v), p_{max}\}$
In Slot $\mathcal{A}$
if *in Slot* $\mathcal{T}$, $\mathcal{E}$ *occurs* **then**
    Let $Y \leftarrow 1$ or 0 with probability $p_t(v)$ and $1 - p_t(v)$, respectively;
    if $Y = 1$ **then**
        Transmit *ack*.
    if *the received message is the same with its own one* **then**
        Halt.
if *transmitted in Slot* $\mathcal{T}$ **then**
    Listen to the channel;
In Slot $\mathcal{I}$
if *transmitted in Slot* $\mathcal{T}$ *and received an ack for its Slot* $\mathcal{T}$*-transmission* **then**
    Transmit;
    Halt.
if *(had* $p_t(v) = p_{max}$ *in* $\frac{1}{10}$*-th of last* $\gamma \log n$ *rounds)* **then**
    Halt.
if *received a message from a neighbor within distance* $\hat{\epsilon}R$ *and same with its own* **then**
    Halt.

rounds are good. The proof of Lemma 1 is very technical. Due to space limitation, we put the proof in the full version [26].

Lemma 1: Given an interval I with $|I| \in \Omega(\log n)$ sufficiently large, then there exists constants g and ρ such that for any constant $\sigma > 0$, w.h.p., for a node v, at least $(1 - \sigma)$ fraction of rounds in I are good.

Based on Lemma 1, we can analyze successful transmissions during the algorithm execution.

Lemma 2: If a node u receives a message from an $\hat{\epsilon}R$-neighbor v and the interference at u is T_a, then all $(1+\hat{\epsilon})R$-neighbors of v also receive the transmission of v.

Proof: By the value of T_a, it can be obtained that there are no other transmitters in $N_{4(1+2\hat{\epsilon})}(u)$. Let w be a node simultaneously transmitting with v. For each node $u' \in N_{1+\hat{\epsilon}}(v)$,

$$\begin{aligned} d(w,u') &\geq d(w,u) - d(u,u') \\ &\geq d(w,u) - (d(u,v) + d(v,u')) \\ &\geq d(w,u) - (1+2\hat{\epsilon})R \\ &\geq \frac{3}{4} d(w,u). \end{aligned} \tag{2}$$

Denote by $S_{u,v}$ as the interference at v caused by node u, i.e., $S_{u,v} = P/d(u,v)^\alpha$. The interference at u' caused by w is bounded as follows.

$$S_{w,u'} = S_{w,u} \cdot \frac{d(w,u)^\alpha}{d(w,u')^\alpha} \leq S_{w,u} \cdot (4/3)^\alpha. \tag{3}$$

Then, the interference experienced by u' is

$$I_{u'} = \sum_{w \in V \setminus \{v\}} S_{w,u'} \leq \sum_{w \in V \setminus \{v\}} S_{w,u} \cdot (4/3)^\alpha = (4/3)^\alpha \cdot T_a.$$

Then we can compute whether u' can receive the transmission of v as follows.

$$\begin{aligned} \frac{P/d(v,u')^\alpha}{N + I_{u'}} &\geq \frac{P/d(v,u')^\alpha}{N + (4/3)^\alpha \cdot T_a} \\ &\geq \frac{N\beta R_T^\alpha/((1-\hat{\epsilon})R_T)^\alpha}{(1-\epsilon/2)^{-\alpha} N} \\ &\geq \frac{N\beta/(1-\epsilon/2)^\alpha}{(1-\epsilon/2)^{-\alpha} N} \\ &= \beta. \end{aligned} \tag{4}$$

The Lemma then follows. ■

Lemma 3: For a node v, if a node $v' \in N_{\hat{\epsilon}/2}(v)$ transmits in a good round for v, with constant probability $\kappa_1 = \frac{1}{2} \cdot (1/4)^g$, v' can make all nodes within distance $(1+\hat{\epsilon})R$ receive its message, and all nodes within distance $\hat{\epsilon}R$ detect interference not larger than T_a.

Proof: Consider a good round t. In round t, $P_t^\rho(v) \leq g$ and $\bar{I}_t^\rho(v) \leq T$. We claim that if there is not any other transmitter in $N_\rho(v)$, with probability at least 1/2, all nodes in $N_{1+\hat{\epsilon}}(v')$ will receive its message, and all nodes in $N_{\hat{\epsilon}}(v')$ detect interference not larger than T_a.

We consider the first claim. Using Markov Inequality, with probability at least $1/2$, the interference at node v is at most $2T$. Then by $T \leq \frac{1}{2}T_a$ and Lemma 2, all nodes in $N_{1+\hat{\epsilon}}(v')$ receives the transmission of v'.

We next prove the second claim. For a node $u' \in N_{\hat{\epsilon}}(v')$, using triangle inequality argument as above, we can show that for each transmitter w, $d(w,u') \geq d(w,v) - d(v,u') \geq d(w,v) - \frac{3\hat{\epsilon}/2}{4(1+\hat{\epsilon})} \cdot d(w,v) = \frac{1+\frac{5}{8}\hat{\epsilon}}{1+\hat{\epsilon}} \cdot d(w,v)$. Hence, the interference at u' can be upper bounded as follows.

$$\begin{aligned} \sum_{w \in V \setminus \{v'\}} S_{w,u'} &= \sum_{w \in V \setminus \{v'\}} S_{w,v} \cdot \frac{d_{w,v}^\alpha}{d_{w,u'}^\alpha} \\ &\leq \left(\frac{1+\hat{\epsilon}}{1+\frac{5}{8}\hat{\epsilon}}\right)^\alpha \cdot 2T \end{aligned} \tag{5}$$

Hence, u' senses an interference not larger than T_a.

The above results are obtained based on the condition that the interference at v is at most $2T$ and there are no other transmitters in $N_\rho(v)$ except v'. This happens with probability $\frac{1}{2} \cdot \prod_{x \in N_\rho(v) \setminus \{v'\}} (1 - p_t(x)) \geq \frac{1}{2} \cdot (1/4)^{\sum_{x \in N_\rho(v) \setminus \{v'\}} p_t(x)} \geq \frac{1}{2} \cdot (1/4)^g$. ■

We now start analyzing successful transmissions in high and low phases.

Lemma 4: In a high phase I for a node v, at least $\Omega(\log n)$ nodes in $N_\rho(v)$ successfully transmit w.h.p.

Proof: Let S be an $\frac{\hat{\epsilon}}{2}R$-maximal independent set in $N_\rho(v)$. Using an area argument, it can be obtained that $|S| \in O(1)$. Let $\sigma = \frac{1}{20|S|}$. By Lemma 1, by setting g and ρ large enough, we can get that for each node u, the number of good rounds in I is at least $(1-\sigma)|I|$. Then, we can get that for all nodes in S, there are at least $\frac{1}{10} - |S| \cdot \frac{1}{20|S|} = \frac{1}{20}$-fraction of rounds in I that are good for all nodes in S and has high contention w.r.t. v. We denote the set of these rounds as I'.

Now consider a round $t \in I'$. Because it is a high contention round w.r.t. v, which means that $P_t^\rho(v) \geq \hat{p}$. Then, we can get that there is a node $\hat{v} \in S$ such that $P_t(\hat{v}) \geq \frac{\hat{p}}{|S|}$. We next consider the transmissions of nodes in $N_{\hat{\epsilon}/2}(\hat{v})$.

Because t is a good round for $\hat{v}$, by Lemma 3, if a node $v' \in N_{\hat{\epsilon}/2}(\hat{v})$ transmits, the $(1+\hat{\epsilon})R$-neighbors of v' can receive the message of v' with constant probability κ_1. The probability that there is a node in $N_{\hat{\epsilon}/2}(\hat{v})$ transmitting is $\frac{\hat{p}}{|S|}$. Then, with probability $\kappa_1\hat{p}/|S|$, v' can send its message to all $(1+\hat{\epsilon})R$-neighbors, and all nodes within distance $\hat{\epsilon}R$ detect an interference not larger than T_a.

We still need to show that v' can receive an *ack* message from its neighbors in Slot $\mathcal{A}$ of round t. By the algorithm, the $\hat{\epsilon}R$-neighbors of v' that detect an interference not larger than T_a will transmit *ack* in Slot $\mathcal{A}$. Because with constant probability, all nodes in $N_{\hat{\epsilon}}(\hat{v})$ detect an interference not larger than T_a in Slot $\mathcal{T}$, these nodes will try to send back *ack* messages. Using a similar argument as in Lemma 3, it can be shown that if there is a node in $N_{\hat{\epsilon}}(v')$ transmitting, the probability that v' can receive the *ack* message is constant. Because $N_{\hat{\epsilon}/2}(\hat{v}) \subseteq N_{\hat{\epsilon}}(v')$, there is a transmitter in $N_{\hat{\epsilon}}(v')$ with probability $P_t(v) - p_t(v') \geq \frac{\hat{p}}{|S|} - p_{max} \geq \hat{p}/2|S|$. Hence, with constant probability, v' will receive an *ack* message.

In above, we have shown that in a high contention and in a good round, with some constant probability, there is a node in $N_\rho(v)$ successfully transmitting. Denote by this constant probability as κ_2. Then during a high phase, the expected number of successfully transmitting nodes is $(1-\sigma)\gamma \log n \cdot \frac{1}{20} \cdot \kappa_2$. Using Chernoff bound, it can be shown that w.h.p, the number of successfully transmitting nodes in $N_\rho(v)$ is $\Omega(\log n)$ if γ is sufficiently large. ■

Lemma 5: A node v will halt after a low phase I w.h.p.

Proof: We only need to show that during the low phase, v can attain the maximum transmission probability in $\frac{1}{10}$-fraction of rounds.

Let $\sigma = \frac{1}{10}$. Then by Lemma 1, there are at least $(1-\sigma)$-fraction of rounds in I that are good for v. By the definition of low phase, we can then obtain that in $\frac{4}{5}$-fraction of rounds in I that are both good and low contention. Denote the set of these rounds as I'. We next consider the rounds in I'.

For a round $t \in I'$, denote by $\mathcal{E}_1$ the event that the interference at v from nodes outside $N_\rho(v)$ is at most T_b, and by $\mathcal{E}_2$ the event that there is not any transmitter within distance ρR from v. When $\mathcal{E}_1$ and $\mathcal{E}_2$ occur, v will sense an idle channel, and make its transmission probability double. We next bound the probability that these two events occur.

Because t is a good round, the expect interference at v from nodes outside $N_\rho(v)$ is at most $T \leq T_b/10$. Using Markov Inequality,

$$Pr(\mathcal{E}_1) \geq 9/10. \tag{6}$$

In t, we know that $P_t^\rho(v) \leq \hat{p}$. Hence,

$$Pr(\mathcal{E}_2) = \prod_{u \in N_\rho(v)} (1 - p_t(u)) \geq (\frac{1}{4})^{\sum_{u \in N_\rho(v)} p_t(u)} \geq (\frac{1}{4})^{\hat{p}}.$$

Hence, $Pr(\mathcal{E}_2) \geq 9/10$.

Combining above results together, the probability that v senses an idle channel is at least $Pr(\mathcal{E}_1) \cdot Pr(\mathcal{E}_2) \geq 4/5$. Then, in expectation, there are at least $\frac{4}{5}$-fraction of rounds in I' in which v senses an idle channel. Using Chernoff bound, it can be shown that w.h.p., v senses an idle channel in at least $\frac{7}{10}$-fraction of rounds in I'. This means that v doubles its transmission probability in at least $\frac{7}{10} \cdot \frac{4}{5} = \frac{28}{50}$ fraction of rounds in I. Notice that at the beginning of I, $p_t(v) \geq p_{min} = \frac{1}{4n}$. And for other rounds, the worst case is that in all these rounds, $p_t(v)$ is halved. Then if γ is sufficiently large, there will be $\frac{28}{50} - \frac{22}{50} - \frac{1}{50} = \frac{1}{10}$ fraction of rounds in I in which v attains the maximum transmission probability p_{max}, where the $\frac{1}{50}|I|$ doubling is used for increasing the initial transmission probability to the maximum one.

As above, we have shown that in a low phase, w.h.p., v will attain the maximum transmission probability in $\frac{1}{10}$-fraction of rounds in I. Here, we did not consider the case that v may also halt after receiving *ack* messages. But this clearly make v halt earlier. Hence, by the algorithm and the above analysis, w.h.p., v will halt in a low phase. ■

In the algorithm, a node v halts when satisfying either of the three conditions: (i) node v receives an *ack* message; (ii) node v attains the maximum transmission probability in $\frac{1}{10}$-fraction of rounds in the past $\gamma \log n$ rounds; (iii) v receives a message from a neighbor within distance $\hat{\epsilon}R$ that is the same with v's in Slot $\mathcal{I}$. For condition (i), we have shown that v can send the message to all its neighbors within distance $(1+\hat{\epsilon})R$. For condition (iii), it also has been shown that the neighbors of v within distance R receive the same message with v. We still need to show that if v halts when satisfying condition (ii), it have also sent its message to all neighbors within distance $(1+\hat{\epsilon})R$.

Lemma 6: When a node v halts because that condition (ii) is satisfied, it has sent its message to all neighbors within distance $(1+\hat{\epsilon})R$ and made all nodes within $\hat{\epsilon}R$ stop the algorithm execution, w.h.p.

Proof: Consider the time interval in which condition (ii) is satisfied for v, and denote the interval as I. Let $\sigma = \frac{1}{20}$. By Lemma 1, there are $(1-\sigma)$-fraction of rounds in I that are good. This means that in at least $\frac{1}{20}$-fraction of good rounds in I in which v attains the maximum transmission probability. Denote by I' the set of these rounds. By Lemma 3, in each good round, if v transmits, it can send its message to $(1+\hat{\epsilon})R$-neighbors and make all nodes within distance $\hat{\epsilon}R$ stop executing the algorithm with constant probability κ_1. Hence, in each round $t \in I'$, with probability $p_{max} \cdot \kappa_1$, the Lemma holds. Then during I', the probability that the Lemma does not hold is reduced to $(1 - p_{max} \cdot \kappa_1)^{\frac{1}{20}\gamma \log n} \leq n^{-2}$. ■

To finally get our result, we next prove that when a node receives an ack message, it can make all nodes within distance $\hat{\epsilon}R$ stop the algorithm execution.

Lemma 7: For a node v, if it received an ack message in Slot $\mathcal{A}$ of a round t, it can send its message to all nodes within distance $(1+\hat{\epsilon})R$ in Slot $\mathcal{I}$.

Proof: Because v receives the ack message, there must be an $\hat{\epsilon}R$-neighbor u receiving the message of v in Slot $\mathcal{T}$, and detects an interference not larger than T_a. By Lemma 2, all nodes in $N_{1+\hat{\epsilon}}(v)$ have also received the message of v. Notice that in Slot $\mathcal{I}$, only nodes that transmit in Slot $\mathcal{T}$ transmit. So in Slot $\mathcal{I}$, the interference at every node will not increase. Then all nodes in $N_{1+\hat{\epsilon}}(v)$ can still receive the transmission of v in Slot $\mathcal{I}$. The Lemma follows. ■

Now we are ready to prove the main result.

Proof of Theorem 1: As shown before, we have proved that when a node v halts, w.h.p., its stored message has been received by all nodes within distance R. Hence, we only need to bound the time each node stays in the algorithm execution. Specifically, for each node v, we bound the number of phases that are high and low respectively. As shown in Lemma 5, there can be at most one low phase. We next bound the number of high phases.

By Lemma 4, in a high phase, there are $\Omega(\log n)$ nodes in $N_\rho(v)$ successfully transmitting w.h.p. By the definition of successful transmission, each of these nodes u can send its message to their $(1+\hat{\epsilon})R$-neighbors, and the $\hat{\epsilon}R$-neighbors of u will not transmit the same message again. This means that for a particular message, its successful transmitters constitute an $\hat{\epsilon}R$-independent set. Because in the R-neighborhood of each node, there are at most k distinct messages to disseminate, and the ρR-neighborhood of a node can be covered by a constant number of nodes' R-neighborhoods, there are at most $O(k)$ distinct messages to disseminate in the ρR-neighborhood of v. Hence, at most $O(k)$ $\hat{\epsilon}R$-independent sets are used in the dissemination of these messages. Finally, notice that the size of an $\hat{\epsilon}R$-independent set in the ρR-neighborhood of a node is a constant. This concludes that in $N_\rho(v)$, there are at most $O(k)$ nodes successfully transmitting. Based on this, we can get that the number of high phases is at most $O(\frac{k}{\log n})$ w.h.p.

Combining all above together, after $O(\frac{k}{\log n}+1)\cdot O(\log n) = O(k+\log n)$ rounds, w.h.p., the message stored at v must have been disseminated to all its R-neighbors. ■

V. Implementing AbsMAC Layer

By the definition of GLB and the primitives given in the absMAC layer, the acknowledgement and progress primitives correspond to the cases that nodes have distinct messages and all nodes send the same message respectively. Based on this observation, we propose the implementation algorithm for absMAC layer.

In particular, the algorithm execution is divided into dual rounds, each of which consists of two rounds, round $\mathbb{A}$ and round $\mathbb{P}$, for implementing the acknowledgement primitive and the progress primitive respectively. The acknowledgement primitive is implemented by letting nodes execute the GLB algorithm given in Algorithm 1, where the steps judging whether its message is the same with received message (Lines 15-16 and 25-26) is deleted, as all nodes have distinct messages. The progress primitive is implemented by adapting the GLB algorithm in Algorithm 1 in the way that whenever a node receive a message in Line 15, it stops the algorithm execution, as all nodes transmit the same message.

Notice that the dual round design can increase the acknowledgement and progress bounds by a factor of at most 2. Then based on Theorem 1, we can get the following result for absMAC layer implementation.

Theorem 2: The absMAC layer can be exactly implemented with $f_{ack} = O(\Delta + \log n)$, $f_{prog} = O(\log n)$, w.h.p.

VI. Simulation Results

We conduct empirical studies for our proposed algorithms in this section. Specifically, we compare the performances of our algorithms and the best known result in previous work, and evaluate the impact of network parameters on the algorithm performance.

Specifically, for the absMAC layer implementation algorithm, we conduct the following simulations: 1) comparison with the only known implementation algorithm given in [10]; 2) evaluating the impact of the network density on the acknowledgement bound f_{ack}; 3) evaluating the impact of parameter ϵ on the progress bound f_{prog}.

For the GLB algorithm, because it is the first known result, we only evaluate the impact of parameters on the algorithm performance, including the network size n, the number of distinct messages k and the model parameter ϵ.

In the simulation, nodes are uniformly and randomly distributed in a square of 150*150. Over 20 runs of the simulations were carried out for each reported result. The default setting of model parameters are given in Table II.

TABLE II
PARAMETERS IN SIMULATION

parameter	value	parameter	value
α	3	β	2
R	10	N	1
k	4	ϵ	0.5

AbsMAC Layer Implementation. The simulation results for absMAC layer implementation are given in Fig. 1∼ Fig. 5. In the figures, ack$_{cs}$ and ack$_{ncs}$ represent the implementations of the acknowledgement primitive in our algorithm and the

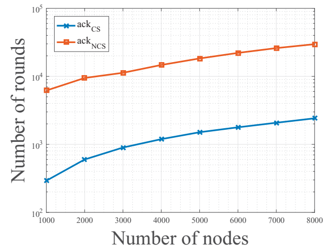

Fig. 1. Comparisons on f_{ack}

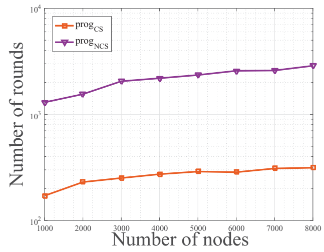

Fig. 2. Comparisons on f_{prog}

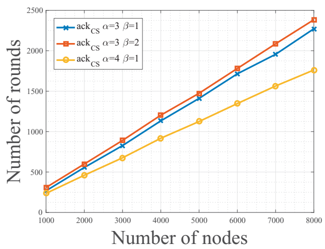

Fig. 3. Impact of SINR parameters on f_{ack}

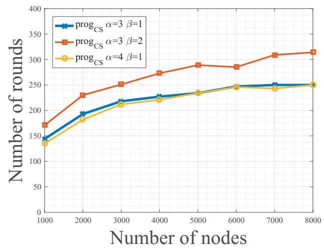

Fig. 4. Impact of SINR parameters on f_{prog}

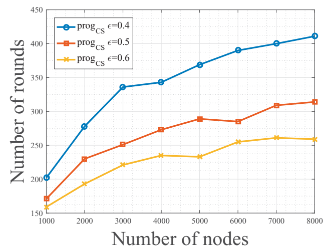

Fig. 5. Impact of ϵ on f_{prog}

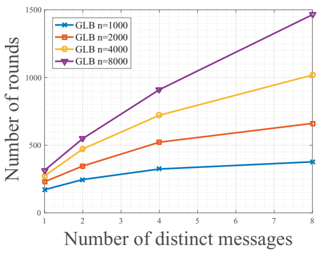

Fig. 6. Impact of number of distinct messages on GLB algorithm performance

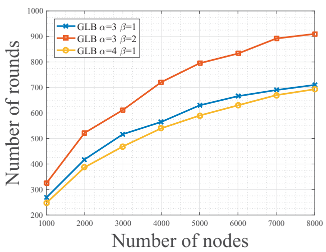

Fig. 7. Impact of SINR parameters on GLB algorithm performance

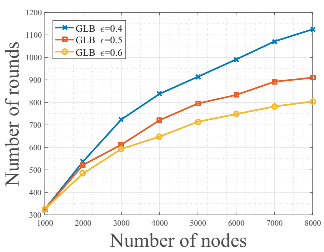

Fig. 8. Impact of ϵ on GLB algorithm performance

one presented in [10] respectively, while prog_{cs} and prog_{ncs} represent the implementations of the progress primitive in these two algorithms respectively.

The comparisons of our implementation algorithm with the one presented by Halldórsson et al. in [10] are illustrated in Fig. 1 and Fig. 2. The curves depict the acknowledgement and progress bounds of the implementation algorithm as the number of nodes in the network changes. As shown in the figures, our algorithm reduce both the acknowledgement and progress bounds for around 10 times. This corroborates our analysis that our algorithm improves for at least an $\log n$ factor in the running time over the algorithm in [10], in terms of both acknowledgement and progress bounds. The figures also illustrates that the constant behind big O notation in the running time bound is not very large, which is around 25. We can also find that the acknowledgement bound increase roughly linearly as the number of nodes changes, but the increase of the progress bound is much slower. This corroborates with our analysis, as the acknowledgement bound is mainly determined by the number of neighbors, which increases linearly with the network size, while the progress bound is logarithmic in the number of nodes.

The impact of SINR parameters are illustrated in Fig. 3 and Fig. 4. As shown in the figures, the delay bounds change very slightly when α and β change. So our algorithm is insensitive to the SINR parameters. In comparison, the acknowledgement bound tends to be affected by the path-loss exponent α, while the progress bound tends to be affected by the decoding threshold β.

Finally, we investigate the impact of parameter ϵ on the progress bound. The evaluation results are shown in Fig. 5. ϵ is used in two places in the progress implementation: the definition of communication range and determining the range for a node to quit when receiving a message. As shown in the figure, as ϵ gets larger, the progress bound decreases. This is because larger ϵ means more nodes will quit from the algorithm execution when a node successfully perform a local broadcast in our algorithm.

General Local Broadcast. We evaluate the performance of our GLB algorithm in Fig. 6$\sim$ Fig. 8.

The impact of the number of distinct messages are illustrated in Fig. 6. As shown in the figure, the time needed for accomplishing GLB increases roughly linearly with the number of distinct messages. This corroborates with our analysis. Furthermore, this figure also shows that the constant behind the big O notation in the analyzed running time bound is not large in all cases.

Fig 7 shows that α and β do not have significant impact on the performance of our GLB algorithm. In comparison, β impacts the algorithm performance heavier than α.

Fig 8 illustrates the running times of the algorithm in different settings of ϵ. It can be seen that the running time decreases as ϵ increases. This is because larger ϵ means when a node successfully perform local broadcast, nodes in a larger range with the same message will stop execution, and hence the general local broadcast can be accomplished faster.

From Fig 7 and Fig 8, it can also be found that the running time of the algorithm increases roughly logarithmically with the number of nodes, which corroborates our analysis.

VII. Applications

With our implementation and combining the algorithms designed based on the absMAC layer, faster algorithms for many fundamental problems can be devised in the physical network. We here give some typical examples to illustrate the widely application of our implementation algorithm.

Consensus.

Definition 2 (Consensus [10]): At the beginning, each n-ode is assigned with an initial value from $\{0, 1\}$. Every node can make a single irrevocable decision on a value in $\{0, 1\}$.

The consensus problem requires the decisions of nodes satisfy the following three properties: 1) *agreement*: no two nodes decide on different values; 2) *validity*: if a node decides a value $x \in \{0, 1\}$, then x must be the initial value of some node; 3) *termination*: every node eventually decides on some value.

In [10], Halldórsson et al. showed the following result.

Theorem 3: The $wPAXOS$ algorithm given in [19] can solve the consensus problem in $O(D \cdot f_{ack})$ time in the (probabilistic) abstract MAC layer model in any connected network topology, w.h.p.

With the acknowledgement bound achieved by our implementation algorithm, we can get the following result for consensus, which improves the $O(D(\Delta + \log \Lambda)(\log n + \log \Lambda))$ time result in [10].

Corollary 1: The consensus problem can be solved in $f_{CONS} = O(D(\Delta + \log n))$ time w.h.p.

Broadcast.

Definition 3 (Multi-Message Broadcast (MMB)): Given k messages that are initially stored at nodes, with k unknown to nodes. The MMB problem requires to disseminate all messages to all nodes in the network.

The BMMB protocol given in [13] can accomplish the multi-message broadcast problem as stated below.

Theorem 4: (Theorem 8.20 in [13]). The MMB problem can be solved in $O(kf_{ack} + (D + k(\log n + \log k))f_{prog})$ time, w.h.p.

Based on the bounds on f_{ack} and f_{prog}, we can get the following result. Our result improves the $O(D \log^{\alpha+1} \Lambda + k(\Delta + \text{polylog}(nk\Lambda))(\log n + \log k))$ time result given in [10].

Corollary 2: MMB can be accomplished in $O(k(\Delta + \log n) + (D + k(\log n + \log k)) \log n)$ time, w.h.p.

As for the special case that $k = 1$, which is also known as the *Single-Message Broadcast* (SMB) problem, the BSMB protocol given in [13] provides the following result.

Theorem 5: W.h.p., SMB can be performed in time $O((D + \log n) f_{prog})$.

Then with the progress bound in our implementation, we have the following result for SMB.

Corollary 3: The SMB problem can be solved in $O((D + \log n) \log n)$ time, w.h.p.

VIII. Conclusion

In this paper, we have shown how to use physical carrier sensing to get efficient exact implementation algorithms for absMAC layer. Our proposed algorithm is asymptotically optimal in terms of both acknowledgement and progress delay bounds. Hence, our implementation algorithm can support higher-level algorithm design efficiently, as illustrated. Our implementation algorithm is based on an algorithm for a general local broadcast problem, which introduces distinct messages into the problem definition for the first time. Our GLB algorithm is potentially useful for solving many other fundamental problems, such as data aggregation and collection. We make this as a future work.

IX. Acknowledgement

This work is supported by National Key Research and Development Program of China under grant No.2016QY02D0302, the National Natural Science Foundation of China Grants 61602195, 61572216, 61672321, 61433012 and U1435215, Natural Science Foundation of Hubei Province 2017CFB301, Shenzhen Research Grant JCYJ20160229195940462.

References

[1] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. On the complexity of radio communication (extended abstract). In *STOC*, 1989.
[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. Barenboim and D. Peleg. Nearly optimal local broadcasting in the SINR model with feedback. In *SIROCCO*, 2015.
[4] S. Daum, S. Gilbert, F. Kuhn, and C. Newport. Broadcast in the Ad Hoc SINR Model. In *DISC*, 2013.
[5] B. Derbel and E. Talbi. Radio network distributed algorithms in the unknown neighborhood model. In *ICDCN*, 2010.
[6] M. Ghaffari, B. Haeupler, and M. Khabbazian. A bound on the throughput of radio networks. *CoRR*, abs/1302.0264, 2013.
[7] M. Ghaffari, B. Haeupler, N. A. Lynch, and C. C. Newport. Bounds on contention management in radio networks. In *DISC*, 2012.
[8] M. Ghaffari, E. Kantor, N. Lynch, and C. Newport. Multi-message broadcast with abstract mac layers and unreliable links. In *PODC*, 2014.
[9] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local broadcasting in the physical interference model. In *DIALM-POMC*, 2008.
[10] M. M. Halldórsson, S. Holzer, and N. Lynch. A local broadcast layer for the sinr network model. In *PODC*, 2015.
[11] M. M. Halldórsson and P. Mitra. Towards tight bounds for local broadcasting. In *FOMC*, 2012.
[12] T. Jurdzinski, D. R. Kowalski, M. Rozanski, and G. Stachowiak. On setting-up asynchronous ad hoc wireless networks. In *INFOCOM*, 2015.
[13] M. Khabbazian, D. R. Kowalski, F. Kuhn, and N. A. Lynch. Decomposing broadcast algorithms using abstract MAC layers. *Ad Hoc Networks*, 12:219–242, 2014.
[14] M. Khabbazian, F. Kuhn, N. A. Lynch, M. Médard, and A. ParandehGheibi. MAC design for analog network coding. In *FOMC*, 2011.
[15] F. Kuhn, N. A. Lynch, and C. C. Newport. The abstract MAC layer. In *DISC* 2009.
[16] 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.
[17] N. Lynch, T. Radeva, and S. Sastry. Asynchronous leader election and mis using abstract mac layer. In *FOMC*, 2012.
[18] N. A. Lynch and C. Newport. A (truly) local broadcast layer for unreliable radio networks. In *PODC*, 2015.
[19] C. C. Newport. Consensus with an abstract MAC layer. In *PODC*, 2014.
[20] D. Yu, Q. Hua, Y. Wang, and F. C. M. Lau. An O(log n) distributed approximation algorithm for local broadcasting in unstructured wireless networks. In DCOSS, 2012.
[21] D. Yu, Q. Hua, Y. Wang, H. Tan, and F. C. M. Lau. Distributed multiple-message broadcast in wireless ad-hoc networks under the SINR model. In *SIROCCO*, 2012.
[22] D. Yu, Y. Wang, Q. Hua, and F. C. M. Lau. Distributed local broadcasting algorithms in the physical interference model. In *DCOSS*, 2011.
[23] D. Yu, Y. Wang, Y. Yan, J. Yu, and F. C. M. Lau. Speedup of information exchange using multiple channels in wireless ad hoc networks. In *INFOCOM*, 2015.
[24] X. Zheng, Z. Cai, J. Li and H. Gao. A Study on Application-aware Scheduling in Wireless Networks. *IEEE Transactions on Mobile Computing*, 16(7): 1787-1801, 2017.
[25] X. Zheng, Z. Cai, J. Li, and H. Gao. Scheduling Flows with Multiple Service Frequency Constraints. IEEE Internet of Things. 4(2): 496-504 (2017).
[26] Optimal algorithm for implementing abstract mac layer via carrier sensing. Technical Report. https://www.dropbox.com/home/paper?preview=absMAC.pdf.