

# Maximizing Broadcast Throughput Under Ultra-Low-Power Constraints

Tingjun Chen, Javad Ghaderi, Dan Rubenstein, and Gil Zussman

**Abstract**—Wireless object tracking applications are gaining popularity and will soon utilize emerging ultra-low-power device-to-device communication. However, severe energy constraints require much more careful accounting of energy usage than what prior art provides. In particular, the available energy, the differing power consumption levels for listening, receiving, and transmitting, as well as the limited control bandwidth must all be considered. Therefore, we formulate the problem of maximizing the throughput among a set of heterogeneous broadcasting nodes with differing power consumption levels, each subject to a strict ultra-low-power budget. We obtain the oracle throughput (i.e., maximum throughput achieved by an oracle) and use Lagrangian methods to design **EconCast** – a simple asynchronous distributed protocol in which nodes transition between sleep, listen, and transmit states, and dynamically change the transition rates. **EconCast** can operate in groupcast or anycast mode to respectively maximize two alternative throughput measures. We show that **EconCast** approaches the oracle throughput. The performance is also evaluated numerically and via extensive simulations and it is shown that **EconCast** outperforms prior art by  $6\times - 17\times$  under realistic assumptions. Moreover, we evaluate **EconCast**'s latency performance and consider design tradeoffs when operating in groupcast and anycast modes. Finally, we implement **EconCast** using the TI eZ430-RF2500-SEH energy harvesting nodes and experimentally show that in realistic environments it obtains 57% – 77% of the achievable throughput.

**Index Terms**—Internet-of-Things, energy harvesting, ultra-low-power, wireless communication

## I. INTRODUCTION

Object tracking and monitoring applications are gaining popularity within the realm of Internet-of-Things [2]. One enabler of such applications is the growing class of ultra-low-power wireless nodes. An example is active tags that can be attached to physical objects, harvest energy from ambient sources, and communicate tag-to-tag toward gateways [3], [4]. Relying on node-to-node communications will require less infrastructure than traditional (RFID/reader-based) implementations. Therefore, as discussed in [3]–[6], it is envisioned that such ultra-low-power nodes will facilitate tracking applications

This research was supported in part by ARO grant 9W911NF-16-1-0259, NSF grant ECCS-1547406, and the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement n°[PIIF-GA-2013-629740].11. A partial and preliminary version of this paper appeared in ACM CoNEXT'16, Dec. 2016 [1], and some results were presented in IEEE WCNC'17 Workshop on Energy Harvesting and Remotely Powered Wireless Communications for the IoT (invited), Mar. 2017.

T. Chen, J. Ghaderi, and G. Zussman are with the Department of Electrical Engineering, Columbia University, New York, NY, USA (email: {tingjun, jghaderi, gil}@ee.columbia.edu).

D. Rubenstein is with the Department of Computer Science, Columbia University, New York, NY 10027, USA (e-mail: danr@cs.columbia.edu).

in healthcare, smart building, assisted living, manufacturing, supply chain management, and intelligent transportation.

A fundamental challenge in networks of ultra-low-power nodes is to schedule the nodes' sleep, listen/receive, and transmit events without coordination, such that they communicate effectively while adhering to their strict power budgets. For example, energy harvesting tags need to rely on the power that can be harvested from sources such as indoor-light or kinetic energy, which provide 0.01 – 0.1 mW [7], [8] (for more details see the review in [9] and references therein). These power budgets are much lower than the power consumption levels of current low-power wireless technologies such as Bluetooth Low Energy (BLE) [10] and ZigBee/802.15.4 [11] (usually at the order of 1 – 10 mW). On the other hand, BLE and ZigBee are designed to support data rates (up to a few Mbps) that are higher than required by the applications our work envisages supporting (less than a few kbps).

In this paper, we formulate the problem of *maximizing broadcast throughput among energy-constrained nodes*. We design, analyze, and evaluate **EconCast**: **E**nergy-**C**onstrained **B**roadcast. **EconCast** is an asynchronous distributed protocol in which nodes transition between sleep, listen/receive, and transmit states, while maintaining a power budget. The nodes and network we focus on have the following characteristics:

**Broadcast**: A transmission can be heard by all listening nodes within the communication range of the transmitting node.

**Severe power constraints**: The power budget is so limited that each node needs to spend most of its time in sleep state and utilize ultra-lower-power communication, resulting in supporting data rates of a few kbps [7]. Traditional approaches that spend energy in order to improve coordination through slotting and synchronization based on accurate clocks, or form some sort of structure (e.g., routing tables and clusters) are too expensive given limited energy and bandwidth.

**Unacquainted**: Nodes should not require pre-existing knowledge of their environment (e.g., properties of neighboring nodes). This can result from the restricted power budget or from unanticipated environment changes due to altered energy sources and/or node mobility.

**Heterogeneous**: The power budgets and the power consumption levels can differ among the nodes.

Efficiently operating such structureless and ultra-low-power networks requires nodes to make their sleep, listen, or transmit decisions in a distributed manner. Therefore, we consider the fundamental problem of maximizing the rate at which the messages can be delivered (the actual content of the transmitted messages depends on the application). Namely, we

focus on maximizing the broadcast throughput and consider two alternative definitions:

- **Groupput** – the total rate of successful bit transmissions to *all the receivers* over time. Groupput directly applies to tracking applications in which nodes utilize a neighbor discovery protocol to identify neighbors which are within wireless communication range [12]–[14]. In such applications, broadcasting information (e.g., a node’s ID) to all other nodes in the network is important, allowing the nodes to transfer data more efficiently under the available power budgets. Groupput can also be applied to data flooding applications where the data at each node is sent to all the other nodes in a network.
- **Anyput** – the total rate of successful bit transmissions to *at least one receiver* over time. It applies to delay-tolerant environments that utilize gossip-style methods to disseminate information. In traditional gossip communication, a node selects a communication partner in a deterministic or randomized manner. Then, it determines the content of the message to be sent based on a naive store-and-forward, compressive sensing [15], [16], or decentralized coding [17]. Another example is delay-tolerant and surveillance applications where urgent data transmission (such as abnormal object status, a potential fire alarm, etc.) may get disrupted or lost due to the limits of communication range, sparsity of mobile nodes, or limited energy resources. In such an environment, a node may wish to send its data to any available receiver.

First, we derive oracle groupput and anyput (i.e., maximum throughput achieved by an oracle) and provide methods to efficiently compute their values. Then, we use Lagrangian methods and a Q-CSMA (Queue-based Carrier Sense Multiple Access) approach to design EconCast. To respectively maximize the two alternative throughput measures, EconCast can operate in groupput or anyput mode. Nodes running EconCast dynamically adapt their transition rates between sleep, listen, and transmit states based on (i) the energy available at the node and (ii) the number (or existence) of other active listeners. To support the latter, a listening node emits a low-cost informationless “ping” which can be picked up by other listening nodes, allowing them to estimate the number (or existence) of active listeners. We briefly discuss how this method helps increasing the throughput and the implementation aspects. We analyze the performance of EconCast and prove that, in theory, it converges to the oracle throughput.

We evaluate the throughput performance of EconCast numerically and via extensive simulations under a wide range of power budgets and listen/transmit consumption levels, and for various heterogeneous and homogeneous networks. Specifically, numerical results show that EconCast outperforms prior art (Panda [14], Birthday [18], and Searchlight [19]) by a factor of  $6\times - 17\times$  under realistic assumptions. In addition, we consider the performance of EconCast in terms of burstiness and latency. We also consider the design tradeoffs of EconCast when operating in groupput and anyput modes.

We implement EconCast using the TI eZ430-RF2500-SEH energy harvesting nodes and experimentally show that in practice it obtains 57% – 77% of the achievable throughput.

Moreover, we compare the experimental throughput to analytical throughput of Panda [14] (where the analytical throughput is usually better than the experimental performance) and show that, for example, EconCast outperforms Panda by  $8\times - 11\times$ .

We note that EconCast is not designed based on specific assumption, regarding the topology and that nodes do not need to know the properties of other nodes. Yet, in this paper, we mainly focus on a clique topology (i.e., all nodes are within the communication range of each other), since it lends itself to analysis. We briefly extend the analytical results to non-clique topologies and also evaluate the performance of EconCast in such networks.

To summarize, the main contributions of this paper are: (i) a distributed asynchronous protocol for a heterogeneous collection of energy-constrained wireless nodes, that can obtain throughput that approaches the maximum possible, (ii) efficient methods to compute the oracle throughput, and (iii) extensive performance evaluation of the protocol.

The rest of the paper is organized as follows. We discuss related work in Section II and formulate the problem in Section III. In Section IV, we present methods to efficiently compute the oracle throughput. We present EconCast in Section V and the proof of the main theoretical result in Section VI. In Section VII, we evaluate EconCast numerically and via simulations. We then discuss the experimental implementation and evaluation of EconCast in Section VIII. We conclude in Section IX. Due to space constraints, some of the proofs are omitted and can be found in a technical report [20].

## II. RELATED WORK

There is vast amount of related literature on sensor networking and neighbor discovery that tries to limit energy consumption. Most of the protocols do not explicitly account for different listen and transmit power consumption levels of the nodes, or do not account for different power budgets [12], [13], [18], [19], [21]–[26]. They mostly use a duty cycle during which nodes sleep to conserve energy and when nodes are simultaneously awake, a pre-determined listen-transmit sequence (e.g., prime number- or quorum-based approaches) with an unalterable power consumption level is used. However, for ultra-low-power nodes constrained by severe power budgets, the appropriate amount of time a node sleeps should explicitly depend on the relative listen and transmit power consumption levels. These prior approaches achieve throughput levels which are much below optimal (and hence much below what EconCast can achieve). Additionally, these protocols often require some explicit coordination (e.g., slotting [12], [18] or exchange of parameters [14], [19]), which are not suitable for emerging ultra-low-power nodes.

From the theoretical point of view, our approach is inspired by the prior work on network utility maximization (e.g., [27]), and queue-based CSMA literature (e.g., [28]–[32]). However, the problem considered in this paper is not a simple extension of the prior work for two reasons. First, in the past work on CSMA and network utility maximization, nodes or links make decisions based on the relative sizes of queues. Often, a queue is a backlog of data to send or the available energy. Prior work

TABLE I: Nomenclature

|                                      |   |
|--------------------------------------|---|
| $\mathcal{N}, N$                     | Set of nodes, number of nodes   |
| $L_i, \mathbf{L}$                    | Node $i$ 's listen power consumption (Watt), $\mathbf{L} = [L_i]$   |
| $X_i, \mathbf{X}$                    | Node $i$ 's transmit power consumption (Watt), $\mathbf{X} = [X_i]$                                       |
| $\rho_i, \boldsymbol{\rho}$          | Node $i$ 's power budget (Watt), $\boldsymbol{\rho} = [\rho_i]$   |
| $b_i$                                | Energy storage level of node $i$ (Joule)  |
| $\mathbf{w}, \mathcal{W}$            | Network state, the set of collision-free states   |
| $\alpha_i, \boldsymbol{\alpha}$      | Fraction of time node $i$ listens, $\boldsymbol{\alpha} = [\alpha_i]$                                     |
| $\beta_i, \boldsymbol{\beta}$        | Fraction of time node $i$ transmits, $\boldsymbol{\beta} = [\beta_i]$                                     |
| $\gamma, \hat{\gamma}$               | Indicator if existing some nodes listening, its estimated value   |
| $c, \hat{c}$                         | Number of nodes listening, its estimated value  |
| $\nu$                                | Indicator if there is exactly one node transmitting   |
| $\pi_{\mathbf{w}}, \boldsymbol{\pi}$ | Fraction of time the network is in $\mathbf{w} \in \mathcal{W}$ , $\boldsymbol{\pi} = [\pi_{\mathbf{w}}]$ |
| $\mathcal{T}_{\mathbf{w}}$           | Throughput of state $\mathbf{w} \in \mathcal{W}$  |
| $\mathcal{T}, \mathcal{T}^*$         | Throughput and oracle throughput  |
| $\mathcal{T}_g, \mathcal{T}_g^*$     | Groupput and oracle groupput  |
| $\mathcal{T}_a, \mathcal{T}_a^*$     | Anyput and oracle anyput  |
| $\eta_i, \boldsymbol{\eta}$          | Lagrange multiplier of node $i$ , $\boldsymbol{\eta} = [\eta_i]$  |

that considers the latter (e.g., [33]) uses the energy only for transmission, while listening is “free”, which is a very different paradigm than the one considered in this paper. Second, in our setting, the queue “backlogs” energy but there is no clear mapping as previously assumed from energy to successful transmission. A node’s listen or transmit events will relieve the backlog, but *do not* increase utility (throughput) unless other nodes are appropriately configured (i.e., transmitting when no listening nodes exist or listening when no transmitting nodes exist does not increase the throughput). This coordination of state among nodes to utilize their energy makes the considered problem more challenging.

Finally, we note that our approach should be amenable to emerging physical layer technology such as backscatter [6].

### III. MODEL AND PROBLEM FORMULATION

We consider a network of  $N$  energy-constrained nodes whose objective is to distributedly maximize the broadcast throughput among them. The set of nodes is denoted by  $\mathcal{N}$ . Table I summarizes the notations.

#### A. Basic Node Model

**Power consumption:** A node  $i \in \mathcal{N}$  can be in one of three states: *sleep* ( $s$ ), *listen/receive*<sup>1</sup> ( $l$ ), and *transmit* ( $x$ ), and the respective power consumption levels are 0,  $L_i$  (W), and  $X_i$  (W).<sup>2</sup> These values are based on hardware characteristics. **Power budget:** Each node  $i$  has a *power budget* of  $\rho_i$  (W). This budget can be the rate at which energy is harvested by an energy harvesting node or a limit on the energy spending rate such that the node can maintain a certain lifetime. In practice, the power budget may vary with time [7], [8] and the distributed protocol should be able to adapt. For simplicity, we assume that the power budget is constant with respect to time, but our analysis in [20] can be easily extended to scenarios with time-varying power budget with the same constant mean. Each node  $i$  also has an energy storage (e.g., a battery or a capacitor) whose level at time  $t$  is denoted by  $b_i(t)$ .

<sup>1</sup>We refer the *listen* and *receive* states synonymously as the power consumption in both states is similar.

<sup>2</sup>The actual power consumption in *sleep* state, which may be non-zero, can be incorporated by reducing  $\rho_i$ , or increasing both  $L_i$  and  $X_i$ , by the sleep power consumption level.

**Severe Power Constraints:** Intermittently connected energy-constrained nodes cannot rely on complicated synchronization or structured routing approaches.

**Unacquainted:** Low bandwidth implies that each node  $i$  must operate with very limited (i.e., no) knowledge regarding its neighbors, and hence, does not know or use the information  $(\rho_j, L_j, X_j)$  of the other nodes  $j \neq i$ .

#### B. Architecture Assumptions

We assume that there is only *one frequency channel* and a *single transmission rate* is used by all nodes in the transmit state. Similar to CSMA, nodes perform *carrier sensing* prior to attempting transmission to check the availability of the medium. Energy-constrained nodes can only be awake for very short periods, and therefore, the likelihood of overlapping transmissions is negligible. We also assume that *a node in the listen state can send out low-cost, informationless “pings”* which can be picked up by other listening nodes, allowing them to estimate the number (or existence) of active listeners. We explain in Section V how this property will help us develop a distributed protocol. In practice, a short *pinging interval* is introduced after each packet transmission, during which the transmitter collects pings from the recipients of the previous packet and performs the estimation (see Section VIII).

#### C. Model Simplifications

At any time  $t$ , the network state can be described as a vector  $\mathbf{w}(t) = [w_i(t)]$ , where  $w_i(t) \in \{s, l, x\}$  represents the state of node  $i$ . While the distributed protocol EconCast (described in Section V) can operate in general scenarios, for analytical tractability, we make the following assumptions:

- The network is a *clique*.<sup>3</sup>
- Nodes can perform *perfect carrier sensing* in which the propagation delay is assumed to be zero.

These assumptions are suitable in the envisioned applications where the distances between nodes are small. Under these assumptions, the network states can be restricted to the set of *collision-free* states, denoted by  $\mathcal{W}$  (i.e., states in which there is *at most* one node in transmit state). This reduces the size of the state space from  $3^N$  to  $(N+1)2^{N-1}$ .

Let  $\gamma_{\mathbf{w}} \in \{0, 1\}$  indicate whether there *exists* some nodes listening in state  $\mathbf{w}$  and let  $c_{\mathbf{w}}$  be the number of listeners in state  $\mathbf{w}$ . We use  $\nu_{\mathbf{w}} \in \{0, 1\}$  as an indicator which is equal to 1 if there is *exactly* one transmitter in state  $\mathbf{w}$  and is 0 otherwise. Based on these indicator functions, two measures of broadcast throughput, *groupput* and *anyput*, and the throughput of a given network state  $\mathbf{w}$  are defined below.

**Definition 1 (Groupput):** The groupput, denoted by  $\mathcal{T}_g$ , is the average throughput of the transmissions received by *all* the receivers, where each transmitted bit is counted once per receiver to which it is delivered, i.e.,

$$\mathcal{T}_g = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t=0}^T \nu_{\mathbf{w}(t)} c_{\mathbf{w}(t)} dt. \quad (1)$$

<sup>3</sup>We also investigate non-clique networks in Section IV-C.

*Definition 2 (Anyput):* The anyput, denoted by  $\mathcal{T}_a$ , is the average throughput of the transmissions that are received by at least one receiver, i.e.,

$$\mathcal{T}_a = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t=0}^T \nu_{\mathbf{w}(t)} \gamma_{\mathbf{w}(t)} dt. \quad (2)$$

*Definition 3 (Network State Throughput):* The throughput associated with a given network state  $\mathbf{w} \in \mathcal{W}$ , denoted by  $\mathcal{T}_{\mathbf{w}}$ , is defined as

$$\mathcal{T}_{\mathbf{w}} = \begin{cases} \nu_{\mathbf{w}} c_{\mathbf{w}}, & \text{for Groupput,} \\ \nu_{\mathbf{w}} \gamma_{\mathbf{w}}, & \text{for Anyput.} \end{cases} \quad (3)$$

Note that without energy constraints, the oracle (maximum) groupput is  $(N - 1)$  and is achieved when some node always transmits and the remaining  $(N - 1)$  nodes always listen and receive the transmission. Similarly, the oracle (maximum) anyput without energy constraints is 1 and is achieved when some node always transmits and some other node always listens and receives the transmission.

#### D. Problem Formulation

Define  $\pi_{\mathbf{z}}$  as the fraction of time the network spends in a given state  $\mathbf{z} \in \mathcal{W}$ , i.e.,

$$\pi_{\mathbf{z}} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t=0}^T \mathbf{1}_{\{\mathbf{w}(t)=\mathbf{z}\}} dt, \quad (4)$$

where  $\mathbf{1}_{\{\mathbf{w}(t)=\mathbf{z}\}}$  is the indicator function which is 1, if the network is with state  $\mathbf{z}$  at time  $t$ , and is 0 otherwise. Correspondingly, denote  $\boldsymbol{\pi} = [\pi_{\mathbf{w}}]$ .

Below, we define the *energy-constrained throughput maximization problem (P1)* where the fractions of time each node spends in sleep, listen, and transmit states are assigned while the node maintains the power budget. Define variables  $\alpha_i, \beta_i \in [0, 1]$  as the fraction of time node  $i$  spends in listen and transmit states, respectively. The fraction of time it spends in sleep state is simply  $(1 - \alpha_i - \beta_i)$ . In view of (1)–(4), (P1) is given by

$$(P1) \quad \max_{\boldsymbol{\pi}} \quad \sum_{\mathbf{w} \in \mathcal{W}} \pi_{\mathbf{w}} \mathcal{T}_{\mathbf{w}} \quad (5)$$

$$\text{subject to} \quad \alpha_i L_i + \beta_i X_i \leq \rho_i, \quad \forall i \in \mathcal{N}, \quad (6)$$

$$\alpha_i = \sum_{\mathbf{w} \in \mathcal{W}_i^l} \pi_{\mathbf{w}}, \quad \beta_i = \sum_{\mathbf{w} \in \mathcal{W}_i^x} \pi_{\mathbf{w}}, \quad (7)$$

$$\sum_{\mathbf{w} \in \mathcal{W}} \pi_{\mathbf{w}} = 1, \quad \pi_{\mathbf{w}} \geq 0, \quad \forall \mathbf{w} \in \mathcal{W}, \quad (8)$$

where  $\mathcal{W}_i^l$  and  $\mathcal{W}_i^x$  are the sets of states  $\mathbf{w} \in \mathcal{W}$  in which  $w_i = l$  and  $w_i = x$ , respectively. Each node is constrained by a power budget, as described in (6), and (8) represents the fact that at any time, the network operates in one of the collision-free states  $\mathbf{w} \in \mathcal{W}$ .

Based on the solution to (P1), the maximum throughput is achievable by an *oracle* that can schedule nodes' sleep, listen, and transmit periods, in a centralized manner. Therefore, we define the maximum value obtained by solving (P1) as the *oracle throughput*, denoted by  $\mathcal{T}^*$ . Respectively, we define the *oracle groupput* and *oracle anyput* as  $\mathcal{T}_g^*$  and  $\mathcal{T}_a^*$ .

To evaluate EconCast, it is essential to compare its performance to the oracle throughput. However, (P1) is a Linear

Program (LP) over an exponentially large number of variables (i.e.,  $|\mathcal{W}|$  is exponential in  $N$ ) and is computationally expensive to solve. In Section IV, we show how to convert (P1) to another optimization problem with only a linear number of variables. Note that the solution to (P1) only provides the optimal fraction of time each node should spend in sleep, listen, and transmit states, but *does not* indicate how the nodes can make their individual sleep, listen, and transmit decisions locally. Therefore, in Section V, we focus on the design of EconCast that makes these decisions based on (P1).

#### IV. ORACLE THROUGHPUT

In this section, we present an equivalent LP formulation for (P1) in a clique network which only has a linear number of variables. We also derive both an upper and a lower bound for the oracle groupput in non-clique topologies which will be used later for evaluating the performance of EconCast in non-clique topologies.

Recall that  $\alpha_i$  and  $\beta_i$  are the fraction of time node  $i$  spends in listen and transmit states, respectively. We can rewrite the constraints in (P1) as follows

$$\alpha_i L_i + \beta_i X_i \leq \rho_i, \quad \forall i \in \mathcal{N}, \quad (9)$$

$$\alpha_i + \beta_i \leq 1, \quad \forall i \in \mathcal{N}, \quad (10)$$

$$\sum_{i \in \mathcal{N}} \beta_i \leq 1. \quad (11)$$

Specifically, (9) is the usual power constraint on each node  $i \in \mathcal{N}$ , and (10) is due to the fact that a node can only operate in one state at any time. We remark that energy-constrained nodes can only be awake for very small fractions of time (i.e.,  $\alpha_i + \beta_i \ll 1$ ), and therefore (10) may be redundant. Finally, collision-free operation in a clique network where *at most* one transmitter can be present at any time imposes (11), which bounds the sum of the transmit fractions by 1.

##### A. Oracle Groupput in a Clique

To maximize the groupput (1), it suffices that any node only listens when there is another transmitter, since listening when no one transmits wastes energy. Namely, the fraction of time node  $i$  listens cannot exceed the aggregate fraction of time all other nodes transmit, i.e.,

$$\alpha_i \leq \sum_{j \neq i} \beta_j, \quad \forall i \in \mathcal{N}. \quad (12)$$

Since a node only listens when there exists *exactly one* transmitter, every listen counts as a reception, and the groupput of a node (i.e., the throughput it receives from all other nodes) is simply the fraction of time it spends in listen state,  $\alpha_i$ . Therefore, the groupput in a clique network simplifies to  $\sum_{i \in \mathcal{N}} \alpha_i$ . The *oracle groupput*, denoted by  $\mathcal{T}_g^*$ , can be obtained by solving the following maximization problem

$$(P2) \quad \mathcal{T}_g^* := \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \sum_{i \in \mathcal{N}} \alpha_i, \quad \text{subject to (9) – (12)}. \quad (13)$$

(P2) is an LP consisting of  $2N$  variables and  $(3N + 1)$  constraints (i.e., solving for  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  given inputs of  $N$ ,  $\boldsymbol{\rho}$ ,  $\mathbf{L}$ , and  $\mathbf{X}$ ). On a conventional laptop running Matlab, this computation for thousands of nodes takes seconds. Moreover, we show that the oracle groupput obtained by solving (P2)

is indeed *achievable* by an oracle which can schedule nodes' listen and transmit periods. This result is summarized in the following lemma.

*Lemma 1:* The (rational-valued) solution  $(\alpha^*, \beta^*)$  to (P2) can be feasibly scheduled by an oracle in a fixed-size slotted environment via a periodic schedule, (perhaps) after a one-time energy accumulation interval.

*Proof:* The proof can be found in Appendix B. ■

In *homogeneous* networks (i.e.,  $\rho_i = \rho$ ,  $L_i = L$ ,  $X_i = X$ ,  $\forall i \in \mathcal{N}$ ) where nodes are sufficiently energy-constrained (i.e., (9) dominates (10)), the solution to (P2) is given by<sup>4</sup>

$$\beta^* = \rho / (X + (N - 1)L), \quad \alpha^* = (N - 1)\beta^*, \quad \mathcal{T}_g^* = N\alpha^*.$$

### B. Oracle Anyput in a Clique

The oracle anyput is obtained based on the observation that a transmission only occurs when there is *at least* one listener. Define variables  $\chi_{i,j}$  as the fraction of time node  $j$  receives a transmission from node  $i$ , for the following constraints

$$\beta_i \leq \sum_{j \neq i} \chi_{i,j}, \quad \forall i \in \mathcal{N}, \quad (14)$$

$$\alpha_j = \sum_{i \neq j} \chi_{i,j}, \quad \forall j \in \mathcal{N}. \quad (15)$$

The *oracle anyput*, denoted by  $\mathcal{T}_a^*$ , can be obtained by solving the following maximization problem

$$\begin{aligned} \text{(P3)} \quad \mathcal{T}_a^* := & \max_{\alpha, \beta} \sum_{i \in \mathcal{N}} \beta_i \\ \text{subject to} \quad & (9) - (11), (14), \text{ and } (15). \end{aligned} \quad (16)$$

First, (14) ensures that when node  $i$  transmits, there is always *at least* one other node than can receive this transmission. Then, (15) makes sure that in the optimal schedule, the fraction of time node  $j$  listens is large enough to cover all the transmissions it receives. Therefore, (P3) maximizes the anyput by ensuring that every transmission is received by *at least* one node. In *homogeneous* networks, the closed-form solution to (P3) is given by

$$\beta^* = \alpha^* = \rho / (X + L), \quad \mathcal{T}_a^* = N\beta^*.$$

### C. Oracle Groupput in Non-cliques

The problem formulations (P1) – (P3) so far have assumed a clique network. Obtaining the exact maximum groupput for non-cliques (denoted by  $\mathcal{T}_{nc}^*$ ) is difficult. This is because a node may receive simultaneous transmissions from two nodes which are not within communication range of each other. As explained before, listen and transmit events are rare within energy-constrained nodes. Therefore, the likelihood of simultaneous transmissions is small and it is expected to have minimal impact on the throughput.

We present both an upper bound  $\overline{\mathcal{T}}_{nc}^*$  and a lower bound  $\underline{\mathcal{T}}_{nc}^*$  on the maximum groupput in non-clique topologies. In the scenarios where  $\overline{\mathcal{T}}_{nc}^*$  and  $\underline{\mathcal{T}}_{nc}^*$  are the same, the exact maximum groupput  $\mathcal{T}_{nc}^*$  can be obtained. The lower bound  $\underline{\mathcal{T}}_{nc}^*$  is obtained by solving (P2) but replace constraint (12) by

$$\alpha_i \leq \sum_{j \in \mathcal{N}(i)} \beta_j, \quad \forall i \in \mathcal{N},$$

<sup>4</sup>We show that in the optimal solution, the equalities hold for equations (9) and (12). The details are in [20].

TABLE II: A simple example in a heterogeneous network.

| Node                               | 1     | 2    | 3    | 4    |
|------------------------------------|-------|------|------|------|
| Power Budget: $\rho_i$ (mW)        | 0.005 | 0.01 | 0.05 | 0.1  |
| Awake(%): $\alpha_i^* + \beta_i^*$ | 0.5   | 1.0  | 5.0  | 10.0 |
| Transmit when Awake(%)             | 20.0  | 22   | 53.6 | 65.7 |

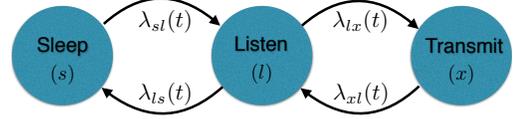


Fig. 1: The node's states and transition rates.

where  $\mathcal{N}(i)$  is the set of neighboring nodes of node  $i$ . This ensures that the fraction of time node  $i$  listens cannot exceed the sum of its neighboring nodes' fractions of transmissions. The upper bound  $\overline{\mathcal{T}}_{nc}^*$  is obtained by solving (P2) in which the constraint (11) is removed. This allows overlapping transmissions which can possibly happen in non-cliques. Numerical results show that with certain topologies,  $\overline{\mathcal{T}}_{nc}^* = \mathcal{T}_{nc}^*$  holds, resulting in the exact maximum groupput  $\mathcal{T}_{nc}^*$ . In Section VII-F, we compute  $\mathcal{T}_{nc}^*$  and evaluate the performance of EconCast in non-clique topologies.

## V. DISTRIBUTED PROTOCOL

In this section, we describe EconCast from the perspective of a single node that transitions between sleep, listen, and transmit states, under a power budget. Since we focus on a single node  $i$ , in parts of this section, we drop the subscript  $i$  of previously defined variables for notational compactness.

### A. A Simple Heterogeneous Example

To better understand the challenges faced in designing EconCast, consider a simple example of 4 nodes, all having identical listen and transmit power consumption  $L_i = X_i = 1$  mW ( $i = 1, 2, 3, 4$ ), but different power budgets  $\rho_i$ , as indicated in Table II. Table II also shows the percentage of time each node spends in listen and transmit states ( $\alpha_i^*, \beta_i^*$ ) ( $i = 1, 2, 3, 4$ ) such that the groupput is maximized by solving (P1). It also shows the percentage of time each node spends in transmit state *when awake* (i.e.,  $\frac{100 \cdot \beta_i^*}{\alpha_i^* + \beta_i^*}$  %).

If, instead, all nodes have the same power budget of  $\rho_i = 0.1$  mW, the percentage of time each node spends in transmit state when awake is 25% (with  $\alpha_i^* = 0.075, \beta_i^* = 0.025$ ,  $i = 1, 2, 3, 4$ ). Note that in the above example, the power budget of node 4 remains unchanged but changes in other nodes' power budgets shift the percentage of time it should transmit when awake from 25% to 65.7%. This clearly shows that the partitioning of a node's power budget among listen and transmit states is highly dependent on other nodes' properties. *However, we will show that if a node does not know the properties of its neighbors, an optimal configuration can be obtained without explicitly solving (P1).*

### B. Protocol Description

To clearly present EconCast, we start from a theoretical framework and slowly build on it to address practicalities. As

mentioned in Section III, a node can be in one of three states: sleep ( $s$ ), listen ( $l$ ), and transmit ( $x$ ). As depicted in Fig. 1, it must pass through listen state to transition between sleep and transmit states. The time duration a node spends in state  $u$  before transitioning to state  $v$  is exponentially distributed with rate  $\lambda_{uv}(t)$ . These *transition rates* can be adjusted over time. We remark that sending packets with exponentially distributed length (i.e., a node transitions from transmit state to listen state with a rate  $\lambda_{xl}$ ) is impractical. However, it can be shown that this is equivalent to continuously transmitting back-to-back unit-length packets with probability  $(1 - \lambda_{xl})$  if  $\lambda_{xl} \in [0, 1]$ , which is indeed the case in EconCast.

To maximize the groupput or anyput, EconCast can operate in *groupput mode* or *anyput mode*, respectively. The throughput as a function of  $\pi_w$  (see (5)) is controlled by appropriately adjusting the transition rates between different states of each node. EconCast determines in a distributed manner how these adjustments are performed over time. Roughly speaking, each node adjusts its transition rates  $\lambda_{uv}(t)$  based on limited information that can be obtained in practice, including

- Its power consumption levels in listen and transmit states,  $L$  and  $X$ , and energy storage level  $b(t)$ .
- A sensing of transmit activity of other nodes over the channel (CSMA-like carrier sensing).
- A count of other active listeners (in groupput mode),  $c(t)$ , or an indicator of whether there are any active listeners (in anyput mode),  $\gamma(t)$ . In practice,  $c(t)$  and  $\gamma(t)$  may not be accurate. We denote  $\hat{c}(t)$  and  $\hat{\gamma}(t)$  as their estimated values.

Note that in EconCast, unlike in previous work such as Panda [14], each node *does not* need to know the number of nodes in the network,  $N$ , and the power budgets and consumption levels of other nodes. Furthermore, a node *does not* need to know its power budget  $\rho$  explicitly (e.g., in the case of energy harvesting [4]), although this knowledge can be incorporated, if available.

Under EconCast, a node sets  $\lambda_{sl}(t)$  as an increasing function of the available stored energy,  $b(t)$ , to more aggressively exit sleep state. Furthermore, it sets  $\lambda_{lx}(t)$  as an increasing function of the number of listeners,  $\hat{c}(t)$ , to enter transmit state more frequently when more nodes are listening. We will describe how these functions are chosen in Section V-E.

### C. Estimating Active Listeners: Pings

We now discuss the estimation of  $\hat{c}(t)$  or  $\hat{\gamma}(t)$ . Recall from Section III that nodes can send out periodic *pings* that any other listener can receive. The pings need not carry any explicit information and are potentially significantly cheaper and shorter than control packet transmissions (e.g., an ACK). Therefore, they consume less power and take much less time than a minimal data transmission.

Consider the case in which all nodes are required to send pings at a pre-determined rate and the power consumption is accounted for in the listening power consumption  $L$ . In such a case, a fellow listener detecting such pings (e.g., using a simple energy detector) can use the count of such pings in a given period of time, or the inter-arrival times of pings, to estimate the number of active listeners  $c(t)$ . Estimating  $\gamma(t)$  is

even easier by detecting the existence of any ping. Although poor estimates are expected to affect the throughput achieved by EconCast, this performance degradation is not significant, as we will show in Section VII-E.

### D. Two Variants of EconCast

We now address the incorporation of the estimates  $\hat{c}(t)$  and  $\hat{\gamma}(t)$  into EconCast. We present two versions of EconCast which only differ when a node is in transmit state:

- EconCast-C (the capture version): a node may “capture” the channel and transmit for an exponential amount of time (i.e., several back-to-back packets). When each packet transmission is completed, the transmitter listens for pings for a fixed-length *pinging interval*. Each successful recipient of the transmission initiate one ping at time chosen uniformly at random on this interval. The transmitter then estimates  $\hat{c}(t)$  or  $\hat{\gamma}(t)$  based on the count of pings received and adjusts  $\lambda_{xl}(t)$  (as described in Section V-E). In Section VIII-C, we discuss the experimental implementation of this process.
- EconCast-NC (the non-capture version): a node always releases the channel after one packet transmission. Each node continuously pings and receives pings from other nodes when listening, estimates  $\hat{c}(t)$  or  $\hat{\gamma}(t)$ , and adjusts  $\lambda_{lx}(t)$  (as described in Section V-E).

EconCast-C is significantly easier to implement since the estimates are only needed for the transmitter right after each packet transmission. The probability that the same transmitter will continue transmitting depends on the estimates  $\hat{c}(t)$  or  $\hat{\gamma}(t)$ . Therefore, our implementation and experimental evaluations in Section VIII focus on EconCast-C.

### E. Setting Transition Rates

Consider a node running EconCast. Time is broken into intervals of length  $\tau_k$  ( $k = 1, 2, \dots$ ). The  $k$ -th interval is from time  $t_{k-1}$  to time  $t_k$  and we let  $t_0 = 0$ . EconCast takes input of two internal variables:

- $\eta$  is a multiplier which is updated at the beginning of each time interval. Let  $b[k]$  ( $k = 0, 1, \dots$ ) denote the energy storage level at the end of the  $k$ -th time interval. Let  $(\cdot)^+$  denote  $\max(0, \cdot)$  and  $\eta[k]$  is updated as follows

$$\eta[k] = \left( \eta[k-1] - \delta_k / \tau_k \cdot (b[k] - b[k-1]) \right)^+, \quad (17)$$

in which  $\delta_k \in (0, 1)$  is a step size and  $b[k] = b(t_k)$ . We use square brackets here to imply that the multiplier  $\eta[k]$  remains constant for  $t \in [t_k, t_{k+1})$ .

- $A(t)$  is the carrier sensing indicator of a node, which is 1 when the node does not sense any ongoing transmission, and is 0 otherwise. Carrier sensing forces a node to “stick” to its current state. When receiving an ongoing transmission, a node in listen state will not exit the listen state until it finishes receiving the full transmission, and a node in sleep state will not leave the sleep state (i.e., it enters the listen state but immediately leaves when it senses the ongoing transmission).

The transition rates are described as follows (the superscripts  $C$  and  $N$  denote EconCast-C and EconCast-NC), in

which the two throughput modes only differ in  $\lambda_{xl}(t)$  (for the capture version) or in  $\lambda_{lx}(t)$  (for the non-capture version). For *groupput* maximization, at any time  $t$  in the  $k$ -th interval,

$$\lambda_{sl}(t) = A(t) \cdot \exp(-\eta[k]L/\sigma), \quad (18a)$$

$$\lambda_{ls}(t) = A(t), \quad (18b)$$

$$\lambda_{lx}^C(t) = A(t) \cdot \exp(\eta[k](L - X)/\sigma), \quad (18c)$$

$$\lambda_{lx}^N(t) = A(t) \cdot \exp(\eta[k](L - X)/\sigma + \hat{c}(t)/\sigma), \quad (18d)$$

$$\lambda_{xl}^C(t) = \exp(-\hat{c}(t)/\sigma), \quad (18e)$$

$$\lambda_{xl}^N(t) = 1. \quad (18f)$$

For *anyput* maximization,  $\hat{c}(t)$  is replaced with  $\hat{\gamma}(t)$ . Theorem 1 below states the main result of this paper and the proof is in Section VI.

*Theorem 1:* Let  $\sigma \rightarrow 0$  and select parameters  $\delta_k$  and  $\tau_k$  properly (e.g.,  $\delta_k = 1/[(k+1) \log(k+1)]$  and  $\tau_k = k$ ). Under perfect knowledge of  $c(t)$  or  $\gamma(t)$ , the average throughput of EconCast ( $\mathcal{T}_g$  or  $\mathcal{T}_a$ ) converges to the oracle throughput ( $\mathcal{T}_g^*$  or  $\mathcal{T}_a^*$ ) given by (P1).

#### F. Stability and Choice of $\sigma$ , $\delta_k$ , and $\tau_k$

EconCast is adaptive and, as expected, it must deal with the tradeoff of adapting “quickly but poorly” to “optimally but slowly”. This adaptation manifests itself into the parameters  $\sigma$ ,  $\delta_k$ , and  $\tau_k$ . When  $\sigma$  is decreased, the throughput increases, as we will describe in Section VI. However, the *burstiness* also increases with respect to decreased  $\sigma$ . The burstiness is a characteristic of communication involving multiple packets that are successfully received in bursts. In Section VII, we describe how the burstiness can be analyzed and measured.

Under a given value of  $\sigma$ , each node continuously adjusts the rates  $\lambda_{uv}(t)$  based on its multiplier  $\eta$  according to (17), which is a function of the ratio  $\delta_k/\tau_k$ . Small  $\delta_k/\tau_k$  ratios (e.g., small step sizes and large intervals) make smaller changes of  $\eta$  over time, and lead to longer convergence time to the “right” multiplier values. In contrast, larger  $\delta_k/\tau_k$  ratios make  $\eta$  oscillate more wildly near the optimal value, such that the performance of EconCast is further from the optimal. Careful choices of the parameters is required for guaranteed convergence (as stated in Theorem 1). In practice, as we set for the simulations and experimentation,  $\delta_k = \delta$  and  $\tau_k = \tau$  for some small constant  $\delta$  and large constant  $\tau$  are sufficient.

## VI. PROOF OF THEOREM 1

The proof of Theorem 1 is based on a Markov Chain Monte Carlo (MCMC) approach [29], [34] from statistical physics and consists of three parts: (i) We compute the steady state distribution of the network Markov chain under EconCast with fixed Lagrange multiplier vector  $\boldsymbol{\eta} = [\eta_i]$ ; (ii) We present an alternative concave optimization problem whose optimal value approaches that of (P1) as  $\sigma \rightarrow 0$  and show that the steady state distribution of EconCast is indeed the optimal solution to this alternative optimization problem when the Lagrange multipliers are chosen optimally; (iii) We show that nodes operating EconCast update their Lagrange multipliers locally according to a “noisy” gradient descent. Based on a stochastic approximation argument, the updated Lagrange multipliers

### Algorithm 1 Gradient Descent Algorithm

**Input parameters:**  $\sigma$ ,  $\boldsymbol{\rho}$ ,  $\mathbf{L}$ , and  $\mathbf{X}$

**Initialization:**  $\alpha_i(0) = \beta_i(0) = \eta_i(0) = 0$ ,  $\forall i \in \mathcal{N}$

1: **for**  $k = 1, 2, \dots$  **do**

2:      $\delta(k) = 1/k$ , compute  $\boldsymbol{\pi}(k)$  from (19) using  $\boldsymbol{\eta} = \boldsymbol{\eta}(k)$

3:     **for**  $i = 1, 2, \dots, N$  **do**

4:         Update  $\eta_i(k)$ ,  $\alpha_i(k)$ , and  $\beta_i(k)$  according to (23), (24)

will converge to their optimal values with proper choices of step sizes and interval lengths as given in Theorem 1.

**Part (i): Steady State Distribution.** The following lemma describes the network state distribution generated by EconCast when  $\boldsymbol{\eta}$  freezes.

*Lemma 2:* With fixed  $\boldsymbol{\eta}$ , the network Markov chain, resulted from overall interactions among the nodes according to the transition rates (18), has the steady state distribution

$$\pi_{\mathbf{w}}^{\boldsymbol{\eta}} = \frac{1}{Z^{\boldsymbol{\eta}}} \exp \left[ \frac{1}{\sigma} \left( \mathcal{T}_{\mathbf{w}} - \sum_{i:w_i=l} \eta_i L_i - \sum_{i:w_i=x} \eta_i X_i \right) \right], \quad (19)$$

where  $Z^{\boldsymbol{\eta}}$  is a normalizing constant so that  $\sum_{\mathbf{w} \in \mathcal{W}} \pi_{\mathbf{w}}^{\boldsymbol{\eta}} = 1$ .

*Proof:* The proof can be found in Appendix A. ■

**Part (ii): An Alternative Optimization.** We then present an optimization problem (P4) as follows

$$\begin{aligned} \text{(P4)} \quad & \max_{\boldsymbol{\pi}} \quad \sum_{\mathbf{w} \in \mathcal{W}} \pi_{\mathbf{w}} \mathcal{T}_{\mathbf{w}} - \sigma \sum_{\mathbf{w} \in \mathcal{W}} \pi_{\mathbf{w}} \log \pi_{\mathbf{w}} \quad (20) \\ & \text{subject to} \quad (6), (7), \text{ and } (8), \end{aligned}$$

where  $\sigma$  is the positive constant used in EconCast (the counterpart in statistical physics is the temperature in systems of interacting particles). Note that (P4) is a concave maximization problem and as  $\sigma \rightarrow 0$ , the optimal value of (P4) approaches that of (P1). To solve (P4), consider the Lagrangian  $\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\eta})$  formulated by moving the power constraint (6) into the objective (20) with a Lagrange multiplier  $\eta_i \geq 0$  for each node  $i$ , i.e.,

$$\begin{aligned} \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\eta}) = & \sum_{\mathbf{w} \in \mathcal{W}} \pi_{\mathbf{w}} \mathcal{T}_{\mathbf{w}} - \sigma \sum_{\mathbf{w} \in \mathcal{W}} \pi_{\mathbf{w}} \log \pi_{\mathbf{w}} \\ & - \sum_{i \in \mathcal{N}} [\eta_i (\alpha_i L_i + \beta_i X_i - \rho_i)]. \quad (21) \end{aligned}$$

In view of (7) and (8), it can be shown that with fixed  $\boldsymbol{\eta}$ , the optimal  $\boldsymbol{\pi}^{\boldsymbol{\eta}} = [\pi_{\mathbf{w}}^{\boldsymbol{\eta}}]$  that maximizes  $\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\eta})$  is exactly given by (19). Therefore, if EconCast knows the optimal Lagrange multiple vector  $\boldsymbol{\eta}^*$ , it can start with  $\boldsymbol{\eta}^*$  and the steady state distribution generated by EconCast will converge to the optimal solution to (P4).

In order to find  $\boldsymbol{\eta}^*$ , consider the dual function  $\mathcal{D}(\boldsymbol{\eta}) = \mathcal{L}(\boldsymbol{\pi}^{\boldsymbol{\eta}}, \boldsymbol{\eta})$  over  $\boldsymbol{\eta} \succeq \mathbf{0}$  (here  $\mathbf{0}$  is an  $N$ -dimensional zero vector and  $\succeq$  denotes component-wise inequality). Interestingly, it can be shown that the partial derivative of  $\mathcal{D}(\boldsymbol{\eta})$  with respect to  $\eta_i$  is simply given by

$$\partial \mathcal{D} / \partial \eta_i = \rho_i - (\alpha_i L_i + \beta_i X_i), \quad (22)$$

which is the difference between the power budget  $\rho_i$  and the average power consumption of node  $i$ . Therefore, the dual can be minimized by using a gradient descent algorithm with inputs of step size  $\delta_k > 0$ ,  $\boldsymbol{\rho}$ ,  $\mathbf{L}$ , and  $\mathbf{X}$ , which generates a state probability  $\boldsymbol{\pi}(k)$  ( $k = 1, 2, \dots$ ). This algorithm is described

in Algorithm 1 along with the following equations

$$\eta_i(k) = [\eta_i(k-1) - \delta_k(\rho_i - \alpha_i(k)L_i - \beta_i(k)X_i)]^+, \quad (23)$$

$$\alpha_i(k) = \sum_{\mathbf{w} \in \mathcal{W}_i^l} \pi_{\mathbf{w}}^{\eta_i(k)}, \quad \beta_i(k) = \sum_{\mathbf{w} \in \mathcal{W}_i^x} \pi_{\mathbf{w}}^{\eta_i(k)}. \quad (24)$$

Hence, with the right choice of step size  $\delta_k$  (e.g.,  $\delta_k = 1/k$ ),  $\boldsymbol{\pi}(k)$  converges to the optimal solution to (P4).

**Part (iii): Convergence Analysis.** To arrive at a distributed solution, instead of computing the quantities  $\alpha_i$  and  $\beta_i$  directly according to (24) (which is centralized with high complexity), nodes operating EconCast approximate the difference between the power budget and the average power consumption (22) by observing the dynamics of their energy storage level. Specifically, each node  $i$  can update its Lagrange multiplier  $\eta_i(k)$  based on the difference between its energy storage levels at the end and the start of an interval of length  $\tau_k$ , divided by  $\tau_k$ , as described by (17). Therefore,  $\eta_i$  is updated according to a “noisy” gradient descent. However, it follows from stochastic approximation (with Markov modulated noise) that by choosing step sizes and interval lengths as given in Theorem 1, these noisy updates will converge to  $\boldsymbol{\eta}^*$  as  $k \rightarrow \infty$  (see e.g., Theorem 1 of [35]). As mentioned in Section V-F, the choice of parameters  $\sigma$ ,  $\delta_k$ , and  $\tau_k$  will affect the tradeoff between convergence time and the performance of EconCast. The detailed proof uses similar techniques as in the proof of Theorem 1 in [35] with minor modifications and can be found in the technical report [20].

## VII. NUMERICAL RESULTS

In this section, we evaluate the throughput and latency performance of EconCast when operating in groupput and anyput modes. We use the following notation: (i)  $\mathcal{T}_g^*$  ( $\mathcal{T}_a^*$ ) is the oracle groupput (anyput) obtained by solving (P1) or, equivalently, (P2), (ii)  $\mathcal{T}_g^\sigma$  ( $\mathcal{T}_a^\sigma$ ) is the achievable groupput (anyput) of EconCast with a given value of  $\sigma$  obtained by solving (P4), and (iii)  $\widetilde{\mathcal{T}}_g^\sigma$  ( $\widetilde{\mathcal{T}}_a^\sigma$ ) is the groupput (anyput) of EconCast obtained via simulations with a given value of  $\sigma$ . For brevity, we ignore the subscripts of  $\mathcal{T}^\sigma$  when describing results that are general for both groupput and anyput.

### A. Setup

We consider clique networks<sup>5</sup> with  $\sigma \in \{0.1, 0.25, 0.5\}$ . The nodes’ power budgets and consumption levels correspond to the energy harvesting budgets and ultra-low-power transceivers in [7], [8], [36]. Note that the performance of EconCast only depends on the *ratio* between the listen or transmit power and the power budget. For example, nodes with  $\rho = 10 \mu\text{W}$ ,  $L = X = 500 \text{ mW}$  behave exactly the same as nodes with  $\rho = 1 \text{ mW}$ ,  $L = X = 50 \text{ mW}$ . Therefore, the oracle throughput applies and EconCast can operate in very general settings.

In the simulations, each node has a constant power input at the rate of its power budget, and adjusts the transition rates based on the dynamics of its energy storage level. Since nodes perform carrier sensing when waking up, there are no

<sup>5</sup>We evaluate the throughput performance of EconCast in non-clique topologies in VII-F.

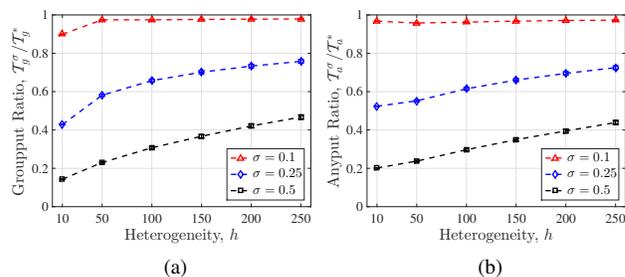


Fig. 2: Sensitivity of the achievable throughput normalized to the oracle throughput,  $\mathcal{T}^\sigma / \mathcal{T}^*$ , for: (a) groupput and (b) anyput, to the heterogeneity of the power budget,  $\rho$ , and power consumption levels,  $L$  and  $X$ .

simultaneous transmissions and collisions. We also assume that the packet length is 1 ms and that nodes have accurate estimate of the number of listeners or the existence of any active listeners, i.e.,  $\hat{c}(t) = c(t)$  or  $\hat{\gamma}(t) = \gamma(t)$ .

The simulation results show that  $\widetilde{\mathcal{T}}^\sigma$  perfectly matches  $\mathcal{T}^\sigma$  for  $\sigma \in \{0.25, 0.5\}$ . For  $\sigma = 0.1$ ,  $\widetilde{\mathcal{T}}^\sigma$  does not converge to  $\mathcal{T}^\sigma$  within reasonable time due to the bursty nature of EconCast, as will be described in Section VII-D. Therefore, we evaluate the throughput performance of EconCast by comparing  $\mathcal{T}^\sigma$  to  $\mathcal{T}^*$  with varying  $\sigma$  in both heterogeneous and homogeneous networks. Specifically, *homogeneous networks* consist of nodes with the same power budget and consumption levels, i.e.,  $\rho_i = \rho$ ,  $L_i = L$ ,  $X_i = X$ ,  $\forall i \in \mathcal{N}$ . The simulation results also confirm that nodes running EconCast consume power on average at the rate of their power budgets.

### B. Heterogeneous Networks – Throughput

One strength of EconCast is its ability to deal with *heterogeneous* networks. Fig. 2 shows the groupput and anyput achieved by EconCast normalized to the corresponding oracle groupput and anyput (i.e.,  $\mathcal{T}^\sigma / \mathcal{T}^*$ ), for heterogeneous networks with  $N = 5$  and  $\sigma \in \{0.1, 0.25, 0.5\}$ . Intuitively, higher values of  $\mathcal{T}^\sigma / \mathcal{T}^*$  indicate better performance of EconCast.

Along the  $x$ -axis, the *network heterogeneity*, denoted by  $h$ , is varied from 10 to 250 at discrete points. The relationship between the network heterogeneity and the values of  $h$  is as follows: (i) for each node  $i$ ,  $L_i$  and  $X_i$  are independently selected from a uniform distribution on the interval  $[510 - h, 490 + h]$  ( $\mu\text{W}$ ), (ii) for each node  $i$ , a variable  $h'$  is first sampled from the interval  $[-\log \frac{h}{100}, \log h]$  uniformly at random, and then  $\rho_i$  is set to be  $\exp(h')$ . Therefore, the energy budget  $\rho_i$  varies from  $100/h$  to  $h$  ( $\mu\text{W}$ ). As a result, for any  $h$ ,  $L_i$  and  $X_i$  have mean values of  $500 \mu\text{W}$ , and  $\rho_i$  has median of  $10 \mu\text{W}$  but its mean increases as  $h$  increases. Note that a homogeneous network is represented by  $h = 10$ .

The  $y$ -axis indicates for each value of  $h$ , the mean and the 95% confidence interval of the ratios  $\mathcal{T}^\sigma / \mathcal{T}^*$  averaged over 1000 heterogeneous network samples. Specifically, in each network sample, each node  $i$  samples  $(\rho_i, L_i, X_i)$  according to the processes described above. Figs. 2(a) and 2(b) show that the network heterogeneity with respect to both the nodes’ power budgets and consumption levels increases as  $h$  increases. Fig. 2 also shows that the throughput ratio  $\mathcal{T}^\sigma / \mathcal{T}^*$  increases as  $\sigma$  decreases, and approaches 1 as  $\sigma \rightarrow 0$ . Furthermore, with increased heterogeneity of the network, the throughput ratio

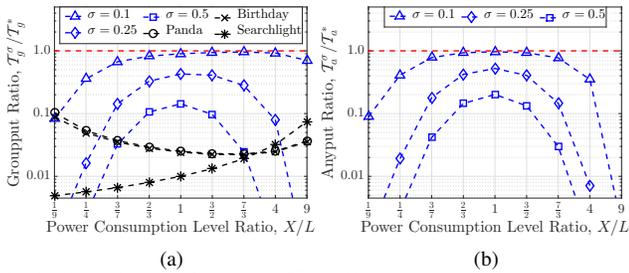


Fig. 3: Throughput performance of EconCast when operating in: (a) groupput mode and (b) anyput mode, with  $N = 5$ ,  $\rho = 10 \mu\text{W}$ ,  $L + X = 1 \text{ mW}$ , and  $\sigma \in \{0.1, 0.25, 0.5\}$ , as a function of  $X/L$ .

has little dependency on the network heterogeneity  $h$  but heavy dependency on  $\sigma$ . In general, the groupput and anyput ratios are similar except for homogeneous networks ( $h = 10$ ). In such networks, the anyput ratio is slightly higher than the groupput ratio. This is due to the fact that nodes have the same values of  $\rho_i$ ,  $L_i$ , and  $X_i$ ). Therefore, determining the existence of any active listeners,  $\gamma(t)$ , is easier than determining the number of active listeners,  $c(t)$ .

### C. Homogeneous Networks – Throughput and Comparison to Related Work

We now evaluate the throughput of EconCast in *homogeneous* networks with  $N = 5$ ,  $\rho = 10 \mu\text{W}$ ,  $L + X = 1 \text{ mW}$ , and  $\sigma \in \{0.1, 0.25, 0.5\}$ . We also compare the groupput achieved by EconCast to three protocols in related work: *Panda* [14], *Birthday* [18], and *Searchlight* [19], which operate under stricter assumptions than EconCast. In particular:

- The probabilistic protocols Panda and Birthday both require a homogeneous set of nodes and a priori knowledge of  $N$ . The throughput of Panda and Birthday is computed as described in [14] and [18], respectively.
- The deterministic protocol Searchlight is designed for minimizing the worst case pairwise discovery latency, which does not directly address multi-party communication across a shared medium. However, the discovery latency is closely related to the throughput, since the inverse of the average latency is the throughput. Hence, maximizing throughput is equivalent to minimizing the average discovery latency. We derive an upper bound on the throughput of Searchlight by multiplying the pairwise throughput by  $(N - 1)$ . This is assuming that all other  $(N - 1)$  nodes will be receiving when one node transmits. However, in practice the throughput is likely to be lower unless all the nodes are synchronized and coordinated.

Figs. 3(a) and 3(b) present, respectively, the groupput and anyput achieved by EconCast normalized to the oracle groupput and anyput, as a function of the power consumption ratio  $X/L$ , with  $N = 5$ ,  $\rho = 10 \mu\text{W}$ , and  $L + X = 1 \text{ mW}$ . Fig. 3(a) also presents the throughput achieved by Panda, Birthday, and Searchlight<sup>6</sup> protocols. The horizontal dashed lines at 1 represent the oracle groupput and anyput. Note that with  $L = X = 500 \mu\text{W}$ , the ratio  $T_g^\sigma / T_g^*$  achieved by EconCast outperforms that of Panda by 6x and 17x with  $\sigma = 0.5$  and  $\sigma = 0.25$ , respectively. In particular, the groupput ratio  $T_g^\sigma / T_g^*$

<sup>6</sup>For Searchlight protocol, we compare its throughput upper bound to  $T_g^*$ .

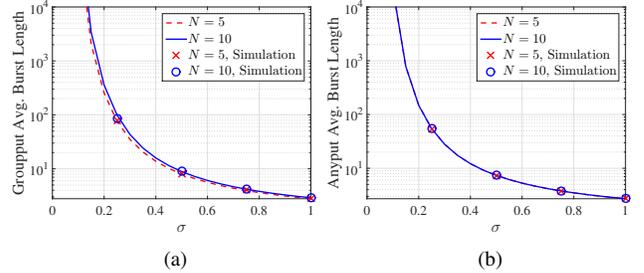


Fig. 4: Analytical (curves) and simulated (markers) average burst length of EconCast when operating in: (a) groupput mode and (b) anyput mode, with  $N \in \{5, 10\}$ ,  $\sigma \in \{0.25, 0.5\}$ ,  $\rho = 10 \mu\text{W}$ , and  $L = X = 500 \mu\text{W}$ .

significantly outperforms that of prior art for  $X \approx L$ . The simulation results, which will be discussed later, also verify this throughput improvement.

Fig. 3 shows that for a given value of  $X/L$ ,  $T^\sigma$  approaches  $T^*$  with decreasing  $\sigma$ , as expected (see Section V). Moreover, for each value of  $\sigma$ , the throughput ratio  $T^\sigma / T^*$  increases as the power consumption ratio  $X/L$  is closer to 1. This is realistic for current commercial low-power radios that have symmetric power consumption levels in listen and transmit states. This is due to the fact that (i) with small  $X/L$  values, nodes enter transmit state infrequently, since listening is expensive and they must pass the listen state to enter the transmit state, and (ii) with large  $X/L$  values, nodes spend energy transmitting even when there are no other nodes listening (e.g.,  $c(t) = 0$ ). In particular, anyput degrades with large  $X/L$  values, since anyput depends on the *existence* of any active listeners when some node is transmitting. Therefore, when listening is expensive, the fact that multiple nodes listen simultaneously does not improve anyput. We believe that any distributed protocol will suffer from such performance degradation since, unlike Panda, Birthday, and Searchlight, nodes in a fully distributed setting *do not* have any information about the properties of other nodes in the network.

### D. Burstiness and Latency

The results until now suggest allowing  $\sigma \rightarrow 0$ . While reducing  $\sigma$  improves throughput, it considerably increases the *burstiness* of communication, as mentioned in Section V. We measure the burstiness by the *average burst length*, which is defined as the average number of back-to-back packets successfully received by a node without exiting the listen state. It is not hard to see that the average burst length in the anyput mode is independent of the network size,  $N$ , and only depends on  $\sigma$ , while the average burst length in the groupput mode depends on both  $N$  and  $\sigma$  (see Appendix C for the detailed analytical derivation). In general, increased burstiness means that the long-term average throughput can be achieved under the given power budgets, but the variance of the throughput is more significant over short term intervals.

Figs. 4(a) and 4(b) show the average burst length of EconCast (in log scale) when operating in groupput and anyput modes, respectively, in homogeneous networks with  $N \in \{5, 10\}$ ,  $\rho = 10 \mu\text{W}$ ,  $L = X = 500 \mu\text{W}$ , and varying  $\sigma$ . Values are obtained using the analytical results (27)–(28) derived in Appendix C (curves) and contrasted with simulations at specific values of  $\sigma$  (markers). Aside from showing

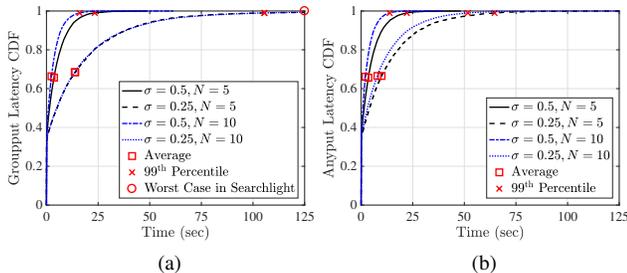


Fig. 5: The CDF, mean, and 99<sup>th</sup> percentile latency of EconCast when operating in: (a) groupput mode and (b) anyput mode, with  $N \in \{5, 10\}$ ,  $\sigma \in \{0.25, 0.5\}$ ,  $\rho = 10 \mu\text{W}$ , and  $L = X = 500 \mu\text{W}$ .

that the simulation results and the analytical results are well matched, Fig. 4 also demonstrates that reducing  $\sigma$  dramatically increases burstiness. For example, with  $\sigma = 0.25$  and  $N = 10$ , a node running EconCast in groupput mode has an average burst length of 85, and this value is increased to  $4 \times 10^5$  with  $\sigma = 0.1$ . This explains why  $\widetilde{\mathcal{T}}_g^\sigma$  does not converge within reasonable time with  $\sigma = 0.1$  (see Section VII-A). Comparing Fig. 4(a) with Fig. 4(b), it can be seen that the groupput average burst length increases more rapidly than the anyput average burst length as  $\sigma$  decreases. Moreover, Fig. 4(b) shows that the anyput average burst length is independent of  $N$ , which corresponds to the analysis in Appendix C. The reason is that the burst length of EconCast in anyput mode only depends on  $\gamma(t)$ , which always equals to 1, if the transmission is successful. We remark that reducing the burstiness is a subject of future work.

A second metric of interest is the communication *latency*, which we define as the average time interval between consecutive bursts that are successfully received by a node from some other node. This metric, together with the burstiness, can fully characterize the throughput achieved by EconCast. Particularly, we can divide time into cycles, where each cycle contains a successfully received burst and the interval followed from this burst until the start of the next successfully received burst. Then, in homogeneous networks, the throughput delivered to a single node is equal to the ratio between the average burst length and the average cycle length.

Figs. 5(a) and 5(b) present the CDF latency of EconCast when operating in groupput and anyput modes obtained via simulations, and indicate both the average and the 99<sup>th</sup> percentile latency values. The homogeneous networks considered are with  $N \in \{5, 10\}$ ,  $\sigma \in \{0.25, 0.5\}$ ,  $\rho = 10 \mu\text{W}$ , and  $L = X = 500 \mu\text{W}$ . Fig. 5(a) also shows the pairwise worst case latency of Searchlight computed from [19] under the same power budget and consumption levels.<sup>7</sup>

Fig. 5 shows that the latency of EconCast in both groupput and anyput modes (i) increases as  $\sigma$  decreases and (ii) decreases as  $N$  increases. The first observation is due to the fact that reducing  $\sigma$  increases the burstiness, which causes higher variation in the nodes' energy storage levels, and thus longer sleep durations. The second observation is due to the fact that as the network size increases, each node is more likely to receive a packet from some other node in the network (note

<sup>7</sup>This is computed with slot length of 50 ms and a beacon (packet) length of 1 ms as was done in [12].

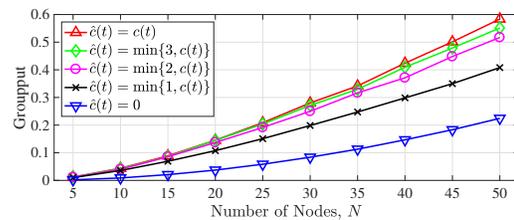


Fig. 6: Simulated throughput of EconCast when operating in groupput mode based on both accurate and inaccurate estimates of the number of listeners,  $\hat{c}(t)$ , with  $\sigma = 0.5$ ,  $\rho = 10 \mu\text{W}$ ,  $L = X = 500 \mu\text{W}$ , and varying  $N$ .

that we do not specify the source of the packet).

Comparing Fig. 5(a) with Fig. 5(b), it is observed that EconCast operating in anyput mode has slightly lower average latency than in groupput mode. However, with a smaller  $\sigma$  value (i.e.,  $\sigma = 0.25$ ), the 99<sup>th</sup> percentile latency of EconCast when operating in anyput mode is significantly lower than that in groupput mode. This results from the fact that the average burst length of EconCast in anyput mode depends on the existence of any listening nodes, whose value is always less than or equal to the number of listening nodes considered in groupput mode. For all parameters considered, the 99<sup>th</sup> percentile groupput latency is within 110 seconds, outperforming the Searchlight pairwise worst case latency bound of 125 seconds. Note that although EconCast has a non-zero probability of having any latency, its latency is below the worst case latency of Searchlight in most cases (over 99%).

### E. Groupput Sensitivity to the Estimated Number of Listeners

As described in Section V-C, the performance of EconCast in groupput mode depends on the estimated number of listeners,  $\hat{c}(t)$ . Estimating  $c(t)$  becomes more challenging in larger networks, due the possibility of more pings being sent during the pinging interval (see Section III-B). However, inaccurate estimates do not significantly degrade the groupput performance of EconCast. To explore the effect of inaccurate estimate  $\hat{c}(t)$ , we evaluate EconCast in the groupput mode while  $\hat{c}(t) \neq c(t)$ . Fig. 6 presents the groupput achieved by EconCast via simulations based on both accurately and inaccurately estimated number of listeners. The homogeneous networks considered are with varying  $N$ ,  $\sigma = 0.5$ ,  $\rho = 10 \mu\text{W}$ , and  $L = X = 500 \mu\text{W}$ .

Fig. 6 shows that the groupput achieved by EconCast using estimate  $\hat{c}(t) = \min\{2, c(t)\}$  is very close to the groupput,  $\widetilde{\mathcal{T}}_g^\sigma$ , achieved using the ground truth  $\hat{c}(t) = c(t)$ . Moreover, the groupput is within 30% of  $\widetilde{\mathcal{T}}_g^\sigma$  with  $\hat{c}(t) = \min\{1, c(t)\}$  for all values of  $N$  considered. This implies that EconCast still has good groupput performance with very rough estimates.<sup>8</sup> Note that such a rough estimate can be obtained easily by a simple energy detector even when multiple pings collide since the transmitter does not need to correctly decode each ping individually. Therefore, the groupput performance of EconCast is not very sensitive to the estimated number of listeners,  $\hat{c}(t)$ .

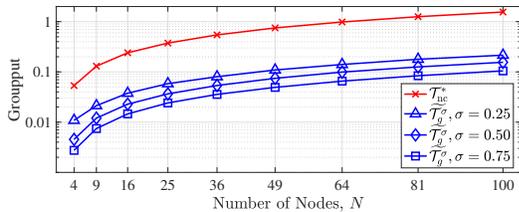


Fig. 7: The oracle groupput,  $\mathcal{T}_{nc}^*$ , and the throughput of EconCast when operating in groupput mode obtained via simulations,  $\mathcal{T}_g^\sigma$ , in grid topologies with varying  $N$ ,  $\sigma \in \{0.25, 0.5, 0.75\}$ ,  $\rho = 10 \mu\text{W}$ , and  $L = X = 500 \mu\text{W}$ .

### F. Groupput Evaluation in Non-clique Topologies

We now compute the oracle groupput for non-clique topologies (derived in Section IV-C) and evaluate the groupput achieved by EconCast in such scenarios via simulations with accurate estimate  $\hat{c}(t) = c(t)$ . Since simultaneous transmissions can happen in non-clique topologies, none of the transmissions will be counted as throughput in the simulations.

We use grid topologies with varying number of nodes,  $N$ , in which each node has at most 4 neighbors. For example,  $N = 25$  represents a  $5 \times 5$  grid. Fig. 7 presents the oracle groupput,  $\mathcal{T}_{nc}^*$ , for grid topologies, and the throughput achieved by EconCast in groupput mode obtained via simulations with varying  $\sigma$  and  $N$ . Note that for all the grid topologies considered, the upper and lower bounds of  $\mathcal{T}_{nc}^*$  (see Section IV-C) are the same, providing the exact oracle groupput.

Fig. 7 shows that EconCast achieves 14% – 22% of the oracle groupput,  $\mathcal{T}_{nc}^*$ , with  $\sigma = 0.25$ . Although increasing  $\sigma$  leads to lower groupput, it can be observed that as  $N$  increases, the groupput approaches 10% of  $\mathcal{T}_{nc}^*$  with  $\sigma = 0.5$ . Despite the fact that the groupput cannot be obtained for  $\sigma = 0.1$ , achieving 10% – 20% of  $\mathcal{T}_{nc}^*$  is remarkable given the fact that EconCast operates in a distributed manner.

## VIII. EXPERIMENTAL EVALUATION

To experimentally evaluate the performance of EconCast-C,<sup>9</sup> we implement it using the Texas Instruments eZ430-RF2500-SEH node [37].<sup>10</sup> In this section, we first describe the energy measurements performed on the nodes running EconCast-C. Then, we describe the method by which nodes can estimate the number of listening nodes. Finally, we experimentally evaluate the performance of EconCast-C.

### A. Experimental Setup

The TI eZ430-RF2500-SEH node is equipped with: (i) an ultra-low-power MSP430 microcontroller and a CC2500 wireless transceiver operating at 2.4 GHz at 250 kbps, (ii) a solar energy harvester (SEH-01) that converts ambient light into electrical energy, and (iii) a 1 mF capacitor to power up the transceiver board. Despite its drawbacks which will be discussed below, it can be used for evaluation by extending the length of the shortest allowable data transmission.

<sup>8</sup>Setting  $\hat{c}(t) = \min\{1, c(t)\}$  is similar to the anyput mode in which the transmitter only detects the existence of any listeners.

<sup>9</sup>See Section V-D the reasons for only implementing EconCast-C.

<sup>10</sup>A demonstration of the testbed is presented in [38].

We consider power budgets of  $\rho \in \{1 \text{ mW}, 5 \text{ mW}\}$ . From our measurements, a node spends  $L = 67.08 \text{ mW}$  in the listen state and  $X = 56.29 \text{ mW}$  in the transmit state.<sup>11</sup> The power consumption levels are very similar from node to node. Recall from Section VII that the performance of EconCast depends on the ratio between the power consumption levels and budget. Therefore, our experimental results will be similar to experiments when both the power consumption levels and budget are scaled down (e.g., a network of nodes with  $\rho \in \{10 \mu\text{W}, 50 \mu\text{W}\}$ ,  $L = 670 \mu\text{W}$ , and  $X = 560 \mu\text{W}$ ).

Each node is programmed with its  $\rho$ ,  $L$ , and  $X$  as the input of EconCast-C. The nodes' main drawbacks include (i) inaccurate readings of the energy storage level (i.e., the voltage of the on-board capacitor) which are sensitive to the environment, and (ii) the fact that the 1 mF capacitor cannot support multiple packet transmissions. Due to these drawbacks, we implement (via software) a virtual battery at each node. The virtual battery emulates the node's energy storage level based on its sleep, listen, transmit activities, as well as the energy consumption of sending and receiving the real pings, and is used for updating the Lagrange multiplier according to (17). We show in the following section that in practice, a node running EconCast-C using this virtual battery is indeed consuming power at a rate close to its power budget.

### B. Energy Consumption Measurements

To accurately measure the power consumption of the nodes, we disable the on-board solar cell, and attach a large pre-charged capacitor ( $C_{\text{cap}} = 5 \text{ F}$ ) that stores energy in advance (similar power consumption measurements were used in [39]). The energy consumed is computed by

$$E_{\text{consumed}} = 0.5 C_{\text{cap}} \cdot (V_{t_0}^2 - V_{t_1}^2), \quad (25)$$

where  $V_{t_0}$  and  $V_{t_1}$  are the measured power voltage values of the capacitor at  $t_0$  and  $t_1$ . The empirical average power consumption,  $\bar{P}(\text{mW})$ , is then computed by

$$\bar{P} = E_{\text{consumed}} / (t_1 - t_0). \quad (26)$$

Note that even with such a big capacitor, a node with a power budget of 1 mW (5 mW) has a lifetime of only 135 (27) minutes with  $V_{t_0} = 3.6 \text{ V}$  and  $V_{t_1} = 3.0 \text{ V}$ , which represent its stable working voltage range.

To measure the power consumption of the nodes, we charge the capacitor to  $V_{t_0} = 3.6 \text{ V}$  and log the readings of  $V_{t_1}$  after 30 minutes using a multimeter. The empirical average power consumption is computed from (25) and (26) for  $\sigma \in \{0.25, 0.5\}$  and is averaged using 60 runs. Because  $L$  and  $X$  do not account for some additional energy usage,<sup>12</sup> the *actual power consumption*,  $\bar{P}$ , is in fact a small fraction higher than the *target power budget*,  $\rho$ . Irrespective of  $\sigma$ , the measurement results show that  $\bar{P}$  exceeds  $\rho$  by 11% for  $\rho = 1 \text{ mW}$ , and by 4% for  $\rho = 5 \text{ mW}$ . Note that the capacitor used has an equivalent energy storage as a 0.6 mAh battery

<sup>11</sup>This corresponds to a  $-16 \text{ dBm}$  transmission power, at which nodes within the same room typically have little or no packet loss.

<sup>12</sup>The additional energy usage includes the energy consumption of sending and receiving the real pings, and the energy consumed in powering up the regulator circuitry, etc.

and EconCast works well at this battery size. Such a battery can be scaled down if the nodes' power budget is scaled down.

Observing the empirical power consumption of the nodes, we compute the achievable throughput by solving (P4) using both the actual power consumption,  $\bar{P}$ , and the target power budget,  $\rho$ , denoted by  $\bar{\mathcal{T}}_g^\sigma$  and  $\mathcal{T}_g^\sigma$ , respectively. In Section VIII-D, we compare the experimental throughput to both  $\bar{\mathcal{T}}_g^\sigma$  and  $\mathcal{T}_g^\sigma$ . Having verified the power consumption of the nodes, we replace the capacitor with AAA batteries,<sup>13</sup> allowing the experiments to run for longer times.

### C. Practical Pinging

To enable practical pinging in EconCast-C, a short, fixed-length *pinging interval* is introduced after each packet transmission. During this interval, the transmitter listens for pings and recipients of the previous packet send a short ping at a random time uniformly distributed within the interval. The transmitter then estimates the number of listeners,  $\hat{c}(t)$ , by counting the pings it receives, and adjusts the transition rate,  $\lambda_{xi}^C(t)$ , according to (18e).

Ideally, each ping should be much shorter than both the pinging interval and the packet length in order to reduce the collisions between pings, and for the transmitter to successfully receive it. Therefore, we use pings of length 0.4 ms, which is the shortest packet that can be sent by a node. Based on this, we empirically set the pinging interval to 8 ms and each data packet to 40 ms. Namely, the energy consumption per ping is 1% of the consumption for a data packet.

### D. Performance Evaluation

We consider homogeneous networks with  $N \in \{5, 10\}$ ,  $\rho \in \{1 \text{ mW}, 5 \text{ mW}\}$ ,  $\sigma \in \{0.25, 0.5\}$ , and nodes are located in proximity. One additional listening node (a 6<sup>th</sup> or 11<sup>th</sup> node) is also present but only as an observer and is connected to a PC via a USB port. Each data packet contains the node ID and information about the number of packets it has received from each other node. The observer node reports all received packets to the PC for storage and post processing. Each experiment is conducted for up to 24 hours. The experimental throughput is computed by dividing the duration of successful transmissions by the experiment duration.

**Throughput evaluation:** Fig. 8 presents the ratio of the experimentally obtained throughput,  $\bar{\mathcal{T}}_g^\sigma$ , normalized to the achievable throughput  $\mathcal{T}_g^\sigma$  and  $\bar{\mathcal{T}}_g^\sigma$  (see Section VIII-B). Separate charts represent the results for differing number of nodes,  $N$ , and power budget,  $\rho$ . Points marked "Ideal" show the experimental throughput normalized to the achievable throughput computed by solving (P4) with the target power budget  $\rho$  (i.e.,  $\bar{\mathcal{T}}_g^\sigma/\mathcal{T}_g^\sigma$ ). Points marked "Relaxed" show the experimental throughput normalized to the achievable throughput computed by solving (P4) with the actual power consumption  $\bar{P}$  (i.e.,  $\bar{\mathcal{T}}_g^\sigma/\bar{\mathcal{T}}_g^\sigma$ ). As expected,  $\bar{\mathcal{T}}_g^\sigma$  is higher than  $\mathcal{T}_g^\sigma$ , resulting in a lower throughput ratio.

Fig. 8 shows that despite the practical limitations (e.g., packet collisions, inaccurate clocks, and inaccurate estimates

TABLE III: Experimental throughput of EconCast-C compared to computed throughput of Panda (all normalized to the achievable throughput  $\mathcal{T}_g^\sigma$ ), with  $\sigma = 0.25$  and varying  $(N, \rho)$ .

| $(N, \rho(\text{mW}))$  | (5, 1) | (10, 1) | (5, 5) | (10, 5) |
|---|--------|---------|--------|---------|
| $\bar{\mathcal{T}}_g^\sigma/\mathcal{T}_g^\sigma$ (%)         | 66.78  | 77.96   | 74.84  | 80.53   |
| $\bar{\mathcal{T}}_{\text{Panda}}/\mathcal{T}_g^\sigma$ (%)   | 6.24   | 9.64    | 19.35  | 35.63   |
| $\bar{\mathcal{T}}_g^\sigma/\bar{\mathcal{T}}_{\text{Panda}}$ | 10.76  | 8.09    | 3.87   | 2.26    |

of the number of active listeners,  $\hat{c}(t)$ ) faced when running EconCast-C on real hardware, the ratio  $\bar{\mathcal{T}}_g^\sigma/\mathcal{T}_g^\sigma$  is between 57% – 77% ( $\bar{\mathcal{T}}_g^\sigma/\mathcal{T}_g^\sigma$  is between 67% – 81%) for all settings considered. Moreover, Table III shows the improvement of EconCast-C over the throughput of Panda computed according to [14], denoted by  $\bar{\mathcal{T}}_{\text{Panda}}$ , under the same power consumption levels and budget, with  $\sigma = 0.25$ . It can be seen that with power budget of  $\rho = 1 \text{ mW}$ , the experimental throughput of EconCast-C outperforms the analytically computed throughput of Panda by  $8\times - 11\times$ .

We remark that getting a higher experimental throughput ratio is limited by the following reasons. First, there is an 8 ms pinging interval (see Section VIII-C) after each packet transmission which effectively reduces the number of bits delivered. Second, collisions of pings or failed decodings of pings result in inaccurate estimates of the number of listeners. Third, the low-power clock used by a node during its sleep state drifts and additionally can be affected by its environment. **Power consumption:** In Section VIII-B, we show that the virtual battery, which is updated based on the energy consumption of both data packets and real pings, is valid for evaluating the actual power consumption of the node. Fig. 8 also presents the mean, minimum, and maximum power consumption of the virtual battery normalized to the target power budget  $\rho$ . Specifically, a value of 1 means that a node consumes power on average at the rate of its power budget throughout the experiment, and a higher value means that a node consumes power at the rate which is higher than its power budget.

The results show that nodes running EconCast-C consume power at rates which are within 7% and 3% of the target power budget with  $\sigma = 0.25$  and  $\sigma = 0.5$ , respectively. This is because smaller value of  $\sigma$  increases the communication burstiness (see Section VII-D), resulting in larger variance of the nodes' virtual battery levels.

**Collection of Pings:** An important input to EconCast-C is the estimates of number of active listeners,  $\hat{c}(t)$ , based on which the transmitter decides the probability to continuously transmit. Larger values of  $\hat{c}(t)$  lead to longer average burst length and can potentially significantly increase the throughput. For example, receiving 1 ping, the transmitter continuously transmits a packet with probability 0.8647 with  $\sigma = 0.5$ . This probability increases to 0.9817 with  $\sigma = 0.25$ , which substantially increases the burstiness. Also, with lower power budget, a successful transmission happens more rarely and it becomes harder to collect pings.

Table IV presents the distribution of number of pings (i.e., number of active listeners) received by the transmitter after each packet transmission, during experiments of  $N = 5$ ,  $\sigma = 0.25$ , and  $\rho \in \{1 \text{ mW}, 5 \text{ mW}\}$ . It can be shown that with a higher power budget, the nodes are more active and

<sup>13</sup>The constant voltage of AAA batteries limits the ability to measure the power consumption of the nodes.

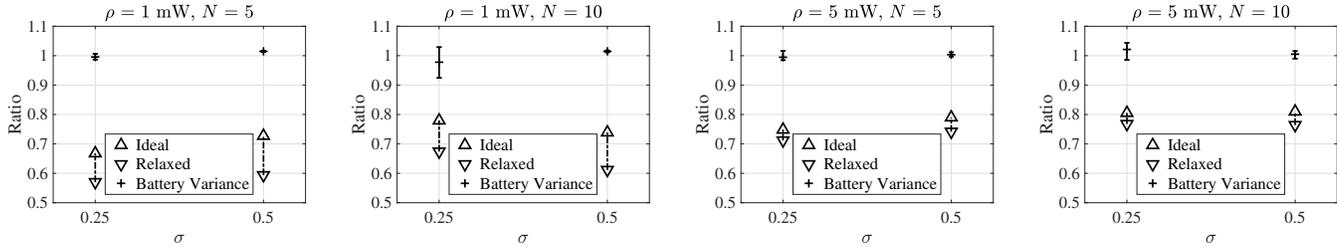


Fig. 8: Points marked as “Ideal” (“Relaxed”) represent ratio of experimental throughput normalized to the achievable throughput obtained by using the target power budget (actual power consumption) and points marked as “Battery Variance” present the average, minimum, and maximum ratios of power consumption normalized to target power budget, with  $N \in \{5, 10\}$ ,  $\rho \in \{1 \text{ mW}, 5 \text{ mW}\}$ , and  $\sigma \in \{0.25, 0.5\}$ .

TABLE IV: Distribution of number of pings (active listeners) received after each packet transmission with  $N = 5$ ,  $\sigma = 0.25$ , and varying  $\rho$ .

| # of Listeners             | 0     | 1     | 2    | 3    | 4    |
|----------------------------|-------|-------|------|------|------|
| $\rho = 1 \text{ mW} (\%)$ | 89.03 | 9.69  | 1.28 | 0.00 | 0.00 |
| $\rho = 5 \text{ mW} (\%)$ | 59.21 | 31.22 | 8.22 | 1.24 | 0.11 |

the transmitter has higher probability to receive more pings. On the other hand, with lower power budget, the transmitter almost never receives more than 3 pings in a 5 nodes experiment, resulting in lower throughput as illustrated in Fig. 8.

## IX. CONCLUSION

In this paper, we considered the problem of maximizing the broadcast groupput and anyput among a set of energy-constrained nodes with heterogeneous power budgets and listen and transmit power consumption levels. We also provided efficient methods to obtain oracle groupput and oracle anyput for a given set of heterogeneous nodes.

We developed the EconCast distributed protocols that control the nodes’ transitions among sleep, listen, and transmit states. We analytically showed that heterogeneous nodes using the protocols (without any a priori knowledge regarding the number of nodes in the network, and power budgets and consumption levels) can achieve the oracle groupput and anyput in a limiting sense (when  $\sigma \rightarrow 0$ ).

We evaluated the throughput performance of EconCast numerically and through extensive simulations, and compared it to the state of the art. We also considered the design tradeoffs in relation to  $\sigma$  and the impact of  $\sigma$  on the burstiness and throughput. Finally, we experimentally evaluated EconCast using commercial-off-the-shelf energy harvesting nodes, thereby demonstrating its practicality.

There are several open future research directions. In particular, future research will focus on extending the analysis to non-clique topologies and improving the delay performance of EconCast. Moreover, evaluation with custom-designed ultra-low-power nodes (e.g., [4]), that have improved energy awareness compared to the TI eZ430-RF2500-SEH nodes, would enable to better assess the tradeoffs related to the protocol design. Finally, considering unique application characteristics and their relation to groupput and anyput is an open problem.

## REFERENCES

- [1] T. Chen, J. Ghaderi, D. Rubenstein, and G. Zussman, “Maximizing broadcast throughput under ultra-low-power constraints,” in *Proc. ACM CoNEXT’16*, 2016.
- [2] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [3] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman, “Challenge: ultra-low-power energy-harvesting active networked tags (EnHANTs),” in *Proc. ACM MobiCom’09*, 2009.
- [4] R. Margolies, M. Gorlatova, J. Sarik, G. Stanje, J. Zhu, P. Miller, M. Szczerzak, B. Vignaham, L. Carloni, P. Kinget *et al.*, “Energy-harvesting active networked tags (EnHANTs): Prototyping and experimentation,” *ACM Trans. Sens. Netw.*, vol. 11, no. 4, p. 62, 2015.
- [5] M. Buettner, B. Greenstein, and D. Wetherall, “Dewdrop: an energy-aware runtime for computational RFID,” in *Proc. USENIX NSDI’11*, 2011.
- [6] J. Wang and D. Katabi, “Dude, where’s my card?: RFID positioning that works with multipath and non-line of sight,” in *Proc. ACM SIGCOMM’13*, 2013.
- [7] M. Gorlatova, A. Wallwater, and G. Zussman, “Networking low-power energy harvesting devices: Measurements and algorithms,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1853–1865, Sept. 2013.
- [8] M. Gorlatova, J. Sarik, G. Grebla, M. Cong, I. Kymissis, and G. Zussman, “Movers and shakers: Kinetic energy harvesting for the internet of things,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1624–1639, 2015.
- [9] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, “Energy harvesting wireless communications: A review of recent advances,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.
- [10] “Texas Instruments CC2640 SimpleLink Bluetooth smart wireless MCU,” 2015. [Online]. Available: <http://ti.com/lit/ds/symlink/cc2640.pdf>
- [11] J.-S. Lee, Y.-W. Su, and C.-C. Shen, “A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” in *Proc. IEEE IECON’07*, 2007.
- [12] W. Sun, Z. Yang, K. Wang, and Y. Liu, “Hello: A generic flexible protocol for neighbor discovery,” in *Proc. IEEE INFOCOM’14*, 2014.
- [13] L. Chen, R. Fan, K. Bian, M. Gerla, T. Wang, and X. Li, “On heterogeneous neighbor discovery in wireless sensor networks,” *arXiv preprint arXiv:1411.5415*, 2014.
- [14] R. Margolies, G. Grebla, T. Chen, D. Rubenstein, and G. Zussman, “Panda: Neighbor discovery on a power harvesting budget,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3606–3619, Dec. 2016.
- [15] C. Luo, F. Wu, J. Sun, and C. W. Chen, “Compressive data gathering for large-scale wireless sensor networks,” in *Proc. ACM MobiCom’09*, 2009.
- [16] T. Srisooksai, K. Keemarungsi, P. Lamsrichan, and K. Araki, “Practical data compression in wireless sensor networks: A survey,” *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 37–59, 2012.
- [17] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, “Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes,” in *Proc. ACM/IEEE IPSN’05*, 2005.
- [18] M. J. McGlynn and S. A. Borbash, “Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks,” in *Proc. ACM MobiHoc’01*, 2001.
- [19] M. Bakht, M. Trower, and R. H. Kravets, “Searchlight: Won’t you be my neighbor?” in *Proc. ACM MobiCom’12*, 2012.
- [20] T. Chen, J. Ghaderi, D. Rubenstein, and G. Zussman, “Maximizing broadcast throughput under ultra-low-power constraints,” *arXiv preprint arXiv:1610.04203v2*, 2017.
- [21] W. Sun, Z. Yang, X. Zhang, and Y. Liu, “Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1448–1459, 2014.

- [22] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.
- [23] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. ACM SenSys'08*, 2008.
- [24] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *Proc. IEEE ISCC'04*, 2004.
- [25] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. ACM/IEEE IPSN'10*, 2010.
- [26] S. Vasudevan, M. Adler, D. Goeckel, and D. Towsley, "Efficient algorithms for neighbor discovery in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 69–83, 2013.
- [27] M. Chiang, S. H. Low, J. C. Doyle *et al.*, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, 2007.
- [28] J. Ghaderi and R. Srikant, "On the design of efficient CSMA algorithms for wireless networks," in *Proc. IEEE CDC'10*, 2010.
- [29] J. Ghaderi, S. Borst, and P. Whiting, "Queue-based random-access algorithms: Fluid limits and stability issues," *Stochastic Systems*, vol. 4, no. 1, pp. 81–156, 2014.
- [30] P. Marbach and A. Eryilmaz, "A backlog-based CSMA mechanism to achieve fairness and throughput-optimality in multihop wireless networks," in *Proc. Allerton Conf.*, 2008.
- [31] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 960–972, 2010.
- [32] K. Xu, O. Dousse, and P. Thiran, "Self-synchronizing properties of CSMA wireless multi-hop networks," in *Proc. ACM SIGMETRICS'10*, 2010.
- [33] S. Chen, P. Sinha, N. B. Shroff, and C. Joo, "A simple asymptotically optimal joint energy allocation and routing scheme in rechargeable sensor networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1325–1336, 2014.
- [34] A. Proutiere, Y. Yi, T. Lan, and M. Chiang, "Resource allocation over network dynamics without timescale separation," in *Proc. INFOCOM'10*, 2010.
- [35] L. Jiang and J. Walrand, "Convergence and stability of a distributed CSMA algorithm for maximal network throughput," in *Proc. CDC'09*, 2009.
- [36] K. Philips, "Ultra low power short range radios: Covering the last mile of the IoT," in *Proc. IEEE ESSCIRC'14*, 2014.
- [37] "Texas Instruments eZ430-RF2500-SEH solar energy harvesting development tool user's guide," 2013.
- [38] T. Chen, G. Chen, S. Jain, R. Margolies, G. Grebla, D. Rubenstein, and G. Zussman, "Demo abstract: Power-aware neighbor discovery for energy harvesting things," in *Proc. ACM SenSys'16*, 2016.
- [39] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," in *Proc. ACM SIGCOMM'12*. ACM, 2012.

## APPENDIX A PROOF OF LEMMA 2

For the capture version EconCast-C, we prove that the transition rates (18a), (18b), (18c), and (18e) will drive the network Markov chain to a steady state with distribution (19), by checking that the *detailed balanced equations* hold. We assume  $\sigma = 1$  and drop the constant term  $1/\sigma$  for brevity. For network state  $\mathbf{w}$ , define  $\mathcal{N}_s(\mathbf{w})$ ,  $\mathcal{N}_l(\mathbf{w})$ , and  $\mathcal{N}_x(\mathbf{w})$  as the sets of nodes in sleep, listen, and transmit states, respectively, and their cardinalities as  $N_s(\mathbf{w})$ ,  $N_l(\mathbf{w})$ , and  $N_x(\mathbf{w})$ , respectively. Note that  $N_s(\mathbf{w}) + N_l(\mathbf{w}) + N_x(\mathbf{w}) = N$  and  $N_x(\mathbf{w}) \in \{0, 1\}$ . We also use  $\mathbf{w} = (\mathcal{N}_s(\mathbf{w}), \mathcal{N}_l(\mathbf{w}), \mathcal{N}_x(\mathbf{w}))$  to denote network state  $\mathbf{w}$  and  $r(\mathbf{w}, \mathbf{w}')$  to denote the transition rate from state  $\mathbf{w}$  to  $\mathbf{w}'$ . We consider the following cases.

**Case 1:** If node  $i$  is in sleep state ( $w_i = s$ ), the only transition that can happen is to transition into listen state when the channel is clear, i.e.,  $\mathbf{w} = (\mathcal{N}_s(\mathbf{w}), \mathcal{N}_l(\mathbf{w}), \emptyset) \rightarrow \mathbf{w}' =$

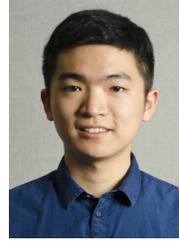
$(\mathcal{N}_s(\mathbf{w}) \setminus \{i\}, \mathcal{N}_l(\mathbf{w}) \cup \{i\}, \emptyset)$ . In this case,  $r(\mathbf{w}, \mathbf{w}') = \exp(-\eta_i L_i)$  and  $r(\mathbf{w}', \mathbf{w}) = 1$ .

**Case 2:** If node  $i$  is in listen state ( $w_i = l$ ), and transitions to sleep state, i.e.,  $\mathbf{w} = (\mathcal{N}_s(\mathbf{w}), \mathcal{N}_l(\mathbf{w}), \emptyset) \rightarrow \mathbf{w}' = (\mathcal{N}_s(\mathbf{w}) \cup \{i\}, \mathcal{N}_l(\mathbf{w}) \setminus \{i\}, \emptyset)$ . In this case,  $r(\mathbf{w}, \mathbf{w}') = 1$  and  $r(\mathbf{w}', \mathbf{w}) = \exp(-\eta_i L_i)$ .

**Case 3:** If node  $i$  is in listen state ( $w_i = l$ ) and transitions to transmit state, i.e.,  $\mathbf{w} = (\mathcal{N}_s(\mathbf{w}), \mathcal{N}_l(\mathbf{w}), \emptyset) \rightarrow \mathbf{w}' = (\mathcal{N}_s(\mathbf{w}), \mathcal{N}_l(\mathbf{w}) \setminus \{i\}, \{i\})$ . In this case,  $r(\mathbf{w}, \mathbf{w}') = \exp(\eta_i(L_i - X_i))$  and  $r(\mathbf{w}', \mathbf{w}) = \exp(-N_l(\mathbf{w}'))$ .

**Case 4:** If node  $i$  is in transmit state ( $w_i = x$ ), the only transition that can happen is to transition to listen state when its transmission is finished, i.e.,  $\mathbf{w} = (\mathcal{N}_s(\mathbf{w}), \mathcal{N}_l(\mathbf{w}), \{i\}) \rightarrow \mathbf{w}' = (\mathcal{N}_s(\mathbf{w}) \cup \{i\}, \mathcal{N}_l(\mathbf{w}) \cup \{i\}, \emptyset)$ . In this case,  $r(\mathbf{w}, \mathbf{w}') = \exp(-N_l(\mathbf{w}))$  and  $r(\mathbf{w}', \mathbf{w}) = \exp(\eta_i(L_i - X_i))$ .

For each case,  $\pi_{\mathbf{w}} \cdot r(\mathbf{w}, \mathbf{w}') = \pi_{\mathbf{w}'} \cdot r(\mathbf{w}', \mathbf{w})$  holds and similar detailed balance equations hold for the non-capture version EconCast-NC. Therefore, we complete the proof. ■



**Tingjun Chen** received the B.Eng. degree in electronic engineering from Tsinghua University in 2014 and the M.S. degree in electrical engineering from Columbia University in 2015, where he is currently pursuing the Ph.D. degree. His research interests are in PHY/MAC layer algorithms, optimization, and system design and implementation in Internet-of-Things, energy harvesting networks, full-duplex networks, and 5G networks. He received the Wei Family Private Foundation Fellowship and the ACM CoNEXT 2016 Best Paper Award.



**Javad Ghaderi** is an Assistant Professor in the Department of Electrical Engineering at Columbia University. His research interests include network algorithms, control and optimization. He received his Ph.D. from the University of Illinois at Urbana-Champaign (UIUC) in 2013. He is the recipient of the Mac Van Valkenburg Graduate Research Award at UIUC, Best Student Paper Finalist at the 2013 American Control Conference, Best Paper Award at ACM CoNEXT 2016, and NSF CAREER Award in 2017.



**Dan Rubenstein** is an Associate Professor in the Department of Computer Science at Columbia University. He received a PhD in computer science from University of Massachusetts, Amherst. His research interests are in network technologies, applications, and performance analysis. He has received a NSF CAREER Award, IBM Faculty Award, the four paper awards from ACM SIGMETRICS 2000, IEEE ICNP 2003, and ACM CoNEXT 2008 and 2016. He was a visiting scientist at Google in 2011, and Chief Scientist at Silver Lining Systems in 2012.



**Gil Zussman** received the Ph.D. degree in electrical engineering from the Technion in 2004 and was a postdoctoral associate at MIT in 2004–2007. He is an Associate Professor of Electrical Engineering at Columbia University. He is a co-recipient of 7 paper awards including the ACM SIGMETRICS'06 Best Paper Award, the 2011 IEEE Communications Society Award for Advances in Communication, and the ACM CoNEXT'16 Best Paper Award. He received the Fulbright Fellowship, the DTRA Young Investigator Award, and the NSF CAREER Award.

APPENDIX B  
PROOF OF LEMMA 1

We describe here a schedule that works assuming each packet transmission has a fixed transmit length of  $\theta$ , though the proof can be extended to varying transmission lengths. Therefore, time can be broken into slots of length  $\theta$  (by the oracle) and nodes sleep, listen, or transmit on a per-slot basis.

Assume that the optimal solution  $(\alpha^*, \beta^*)$  to (P2) yields rational values for all  $\alpha_i^*$  and  $\beta_i^*$ . The period size of the oracle schedule  $P$  is set to the least common denominator over all solution variables in  $(\alpha^*, \beta^*)$ . Hence, during each period, node  $i$  listens for  $\alpha_i^*P$  slots and transmits for  $\beta_i^*P$  slots. The slots during the period can be assigned arbitrarily by the oracle to transmitters (e.g., weighted round-robin or in-order) and (11) ensures that there will be sufficient slots. Once the slots for the transmitters are assigned, each listener can then choose their  $\alpha_i^*P$  slots in which they listen from the set of transmit slots assigned to other transmitters, and (12) ensures these are sufficient as well (note that multiple listeners are permitted for a single transmitter slot).

If the periodic scheduler is launched immediately, some nodes may not have the harvested (or budgeted) energy to perform all listen and transmit tasks within the first period (e.g., it may be assigned to transmit and listen early on, and recoup the energy during later slots). If such a case, we simply delay the initial iteration, allowing all nodes to harvest (or budget)  $\rho_i P \theta$  amount of energy for that initial period. Then the nodes have enough energy to repeat all the subsequent periods since the energy node  $i$  spends during the  $k$ -th period is  $(\alpha_i L_i P \theta + \beta_i X_i P \theta)$ , and the energy it accumulates during

this period is  $\rho_i P \theta$ . Hence, it follows from (9) that no more energy is spent than is accumulated (budgeted). Therefore, there is sufficient energy to repeat the period. ■

APPENDIX C  
BURSTINESS ANALYSIS OF ECONCAST

To derive the average burst length (denoted by  $B$ ) of EconCast-C, we use  $\pi_{\mathbf{w}}^*$  to denote the optimal solution to (P4) and define  $\mathcal{W}' = \{\mathbf{w} \in \mathcal{W} : \nu_{\mathbf{w}} = 1, c_{\mathbf{w}} \geq 1\}$ , i.e., the set of states with successfully received bursts. Recall that for a given value of  $\sigma$ , the optimal value of (P4) is exactly  $\mathcal{T}^\sigma$ . According to (18e), for a given state  $\mathbf{w} \in \mathcal{W}'$ , the average burst length of EconCast-C in groupput mode is  $\exp(c_{\mathbf{w}}/\sigma)$ . Therefore, during a (long enough) time duration of  $T$ , the average number of bursts received by all the nodes can be computed by  $\sum_{\mathbf{w} \in \mathcal{W}'} \frac{T \cdot \pi_{\mathbf{w}}^*}{\exp(c_{\mathbf{w}}/\sigma)}$ , and the average burst length of EconCast in groupput mode,  $B_g$ , is given by

$$\begin{aligned} B_g &= (\text{Avg. Total Burst Length}) / (\text{Avg. Number of Bursts}) \\ &= \frac{T \cdot \sum_{\mathbf{w} \in \mathcal{W}'} \pi_{\mathbf{w}}^*}{\sum_{\mathbf{w} \in \mathcal{W}'} \frac{T \cdot \pi_{\mathbf{w}}^*}{\exp(c_{\mathbf{w}}/\sigma)}} = \frac{\sum_{\mathbf{w} \in \mathcal{W}'} \pi_{\mathbf{w}}^*}{\sum_{\mathbf{w} \in \mathcal{W}'} \frac{\pi_{\mathbf{w}}^*}{\exp(c_{\mathbf{w}}/\sigma)}}. \end{aligned} \quad (27)$$

Similarly, the average burst length of EconCast-C in anyput mode is computed by replacing  $c_{\mathbf{w}}$  with  $\gamma_{\mathbf{w}}$  in (27). Since  $\gamma_{\mathbf{w}} = 1$  always holds for  $\mathbf{w} \in \mathcal{W}'$ ,  $B_a$  simplifies to

$$B_a = \frac{\sum_{\mathbf{w} \in \mathcal{W}'} \pi_{\mathbf{w}}^*}{\sum_{\mathbf{w} \in \mathcal{W}'} [\pi_{\mathbf{w}}^* \cdot \exp(-1/\sigma)]} = \exp(1/\sigma). \quad (28)$$

This shows that the anyput average burst length is independent of the number of nodes,  $N$ , and only depends on  $\sigma$ . ■