Multi-flow Glossy: Physical-layer Network Coding Meets Embedded Wireless Systems

Abdelrahman Abdelkader*, Johannes Richter*, Eduard A. Jorswieck*, Marco Zimmerling†
*Communications Laboratory, Faculty of Electrical and Computer Engineering, TU Dresden, Germany
†Networked Embedded Systems Group, Center for Advancing Electronics Dresden, TU Dresden, Germany
{abdelrahman.abdelkader, johannes.richter, eduard.jorswieck, marco.zimmerling}@tu-dresden.de

***Abstract*—Emerging applications like wireless control or drone swarms require low-latency communication across multiple hops among a large number of both static and mobile devices. Recent protocols based on synchronous transmissions can meet most of these requirements. In particular, Glossy comes extremely close to the minimum lower latency bound for flooding a *single* packet. However, when *multiple* packets need to be exchanged, the overall latency increases linearly since each packet must be mapped onto a distinct Glossy flood. This paper explores the opportunities and challenges of physical-layer network coding to exchange more packets per unit of time. To this end, we present *Multi-flow Glossy (MF-Glossy)*, a communication scheme that simultaneously floods different packets from multiple sources to all nodes in the network. We determine upper bounds on the performance of Glossy and MF-Glossy using a communication-theoretic analysis. Further, we show by simulation that MF-Glossy has the potential to achieve several-fold improvements in goodput and latency across a wide spectrum of network configurations at lower energy costs and comparable packet reception rates. On the road to harnessing this potential in real low-power networks, we demonstrate a proof-of-concept implementation on software-defined radios, and discuss the key research challenges and possible solutions lying ahead.**

I. Introduction

Embedded wireless networks are a key factor to innovation in cyber-physical systems (CPS). Using wireless radios instead of wires cuts costs and clears physical barriers, allowing to tap into previously inaccessible information and to create applications with unprecedented opportunites, from industrial wireless control [1] to emergency response via aerial drones [2].

Challenges. These emerging applications often rely on multi-hop communication with stringent requirements on latency and reliability [1], [2]. For instance, coordination and control tasks require short end-to-end delays of 10–250 ms [3] and tolerate only small packet loss rates [4]. Further, industrial applications and drone swarms typically need to exchange messages among a large number of static and/or mobile devices [1], [5], which are often also subject to energy constraints [3], [6].

Given the state of the art in low-power wireless networking, one possible solution to meet these requirements is to map the communication demands onto a sequence of Glossy floods [7]. Similar to the Low-Power Wireless Bus (LWB) [8], each node that wishes to transmit is allocated a time slot in which it can flood one packet to all other nodes in the network. For flooding an *individual packet* using half-duplex radios, Glossy achieves almost the theoretical minimum communication latency, while providing a reliability higher than 99.9 % in diverse scenarios regardless of whether the devices are static or mobile [7], [8]. However, the overall latency increases linearly with the number of packets if *multiple packets from different source nodes* are exchanged using Glossy. In order to avoid interference between consecutive Glossy floods with different packets, the next flood can only start after the previous one is over [8]. This operation bounds the number of packets that can be exchanged within, for example, the cycle time of a control or coordination task, which may limit the applicability of low-power wireless technology to control systems with slow-changing dynamics and to swarms with not many more than a handful of drones.

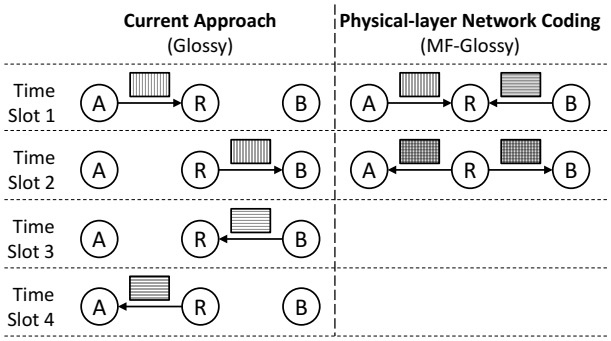


Figure 1: Two-hop relay network, where nodes A and B intend to exchange a message with each other via relay R. Using physical-layer network coding, A and B send simultaneously. R receives and decodes the sum of both packets, which it encodes and transmits. A and B substract their own packet to get the other packet. This doubles the goodput compared with the current approach.

Contribution. In this paper, we ask the question whether it is possible to overcome this limitation while retaining the merits of Glossy, including its seamless support for communication to all nodes in the presence of mobile devices. In particular, we explore the use of physical-layer network coding (PLNC) to enable *multiple sources simultaneously flood different packets to all nodes*. This way, we aim at significantly increasing the goodput (*i.e.*, the amount of successfully exchanged data in the network per unit of time) over the state of the art for the same transmit bit rate (*e.g.*, 250 kbps for IEEE 802.15.4 radios).

Like Glossy, PLNC is based on synchronous transmissions. Unlike Glossy, however, the transmitted packets have *different* payloads. As an example, consider the classical two-hop relay network in Figure 1, where nodes A and B want to exchange a message with one another through relay R. Using Glossy, node A "floods" its packet via R to B, and then node B "floods" its packet via R to A. So exchanging two packets takes 4 time

slots. PLNC achieves the same in 2 time slots. In particular, nodes A and B transmit their packets simultaneously in the first time slot. Due to the coding that naturally occurs when electromagnetic waves come together within the same physical space, relay R receives and decodes the sum (*i.e.*, a linear combination) of both packets, which it then encodes and transmits in the second time slot. Because A and B know the packets they transmitted, they can subtract them from the received sum to get the other packet. Thus, using PLNC, we double the goodput in this example compared with Glossy.

Introduced in 2006 by Zhang et al. [9], the first PLNC implementation on software-defined radios (SDRs) was demonstrated by Katti et al. [10] in 2007. They adopt an *amplify-and-forward* approach, where a relay does not decode the received superposition but simply amplifies and forwards the signal. In 2011, Nazer and Gastpar showed that a *compute-and-forward* approach (see Section IV-B for more details), where relays de- and encode linear combinations of packets, can achieve higher goodput than amplify-and-foward [11].

We investigate the use of the compute-and-forward approach by Nazer and Gastpar [11] to enable the simultaneous flooding of multiple packets in a multi-hop embedded wireless network. We refer to this many-to-all communication scheme as *Multi-flow Glossy (MF-Glossy)*. Using MF-Glossy, a given number of souces K initates the flood by simultaneously transmitting K different source packets. At the end of a MF-Glossy flood, every node in the network is aware of all K source packets.

To the best of our knowledge, we are the first to consider PLNC in a low-power wireless setting. We realize that a lot of research needs to be done before MF-Glossy runs on off-the-shelf devices just like Glossy. As such, our goal in this paper is two-fold: We intend to (*i*) shed light on the performance gains we can hope to achieve over Glossy by using the compute-and-forward approach, which has thus far never been implemented on real hardware, and (*ii*) identify the key challenges that must by solved to harness these performance gains in a MF-Glossy implementation on embedded wireless devices.

To this end, this paper contributes the following:

- We introduce MF-Glossy, a many-to-all communication scheme that exploits PLNC based on the compute-and-forward approach [11] to simultaneously flood multiple packets in a multi-hop embedded wireless network.
- By conducting a communication-theoretic analysis, at the most fundamental level, we provide thus far unknown upper bounds on the performance of the widely used state-of-the-art Glossy and our proposed MF-Glossy scheme.
- We show in simulation that MF-Glossy has the potential to outperform Glossy across a wide spectrum of network configurations. Relative to Glossy, MF-Glossy can reduce latency by $9\times$ and boost goodput by $3\times$, while consuming less energy and providing the same high reliability.
- We demonstrate the first implementation of the compute-and-forwared approach on a small testbed of SDRs. As a proof-of-concept, our implementation uses general open-source software frameworks, which are not optimized for the specific coding operations at hand. Thus, we measure 2.4 ms and 5.3 ms for encoding and decoding one byte, respectively, on an Intel i5 processor. We discuss several directions to significantly improve on these figures.

Overall, we show that MF-Glossy promises significant performance gains and is indeed implementable. Our analytical performance upper bounds, while interesting in their own right, determine the processing overhead an implementation of MF-Glossy can afford to not outweigh the gains. Notwithstanding its limitations, our proof-of-concept implementation serves as a stepping stone to making MF-Glossy viable for state-of-the-art wireless embedded platforms (*e.g.*, leveraging 32-bit ARM Cortex-M microcontrollers). Based on our insights, we outline a concrete research agenda, which is part of our ongoing work.

II. Related Work

MF-Glossy is related to work on synchronous transmissions and physical-layer network coding, thus connecting research in the sensor network, wireless communications, and information theory communities.

Synchronous transmissions. Laneman et al. [12] theoretically demonstrate gains when spatially diverse wireless transmitters cooperate to relay information. Rahul et al. [13] are the first to demonstrate these gains in practice. Using an FPGA-based 802.11-like radio platform, they demonstrate reduced bit error rates and higher throughput when multiple nodes transmit the same packet at the same time on the same OFDM subcarrier. Dutta et al. [14] report on non-destructive interference effects when multiple sensor nodes simultaneously transmit identical, short acknowledgment frames generated in hardware. Ferrari et al. [7] show that this observation also applies to variable-size packets generated in software. As described in Section IV-A, their Glossy protocol provides both one-to-all communication and time synchronization in multi-hop networks. To facilitate data exchange among multiple nodes, LWB [8] allocates non-overlapping time slots to individual nodes for flooding single packets via Glossy, and Chaos [15] builds upon Glossy to compute aggregate functions. Notably, Chaos exploits synchronous transmissions of packets with different payloads, where nodes rely on the capture effect [16] to receive one of them.

MF-Glossy also exploits synchronous transmissions of different packets, yet it uses physical-layer techniques to receive and decode their superposition. When applied on the network scale, this enables the simultaneous flooding of multiple packets. Thus, MF-Glossy extends Glossy in a more fundamental way than prior work, and is the first effort applying physical-layer network coding to embedded wireless systems.

Physical-layer network coding. Some years ago, the way of thinking about interference changed with the introduction of network coding [17] and PLNC [9]. For wired networks, we can see a paradigm shift from packet-switched networks, where each data packet is routed individually through the network, to code-centric networks, where data packets are combined on their way through the network. This helps resolve bottlenecks on frequently used routes. But network coding is not limited to wired networks. The concept is also applicable in wireless networks, where interference is no longer

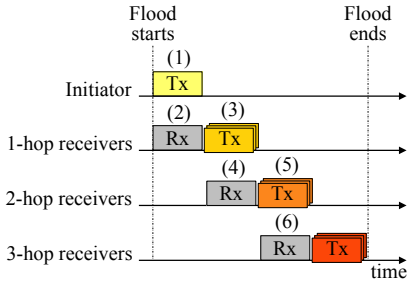


Figure 2: Series of synchronous receptions (Rx) and transmissions (Tx) as a Glossy flood propagates in a 3-hop network with re-transmission limit $N = 1$.

considered a disadvantage but instead exploited together with multipath data distribution. Since the introduction of PLNC, different schemes have been developed [18]–[21]. An approach that received a lot of attention in recent years is compute-and-forward [11]. Using lattice codes, it achieves the additive white Gaussian noise (AWGN) channel capacity with lattice decoding instead of maximum-likelihood decoding [22].

To make compute-and-forward practical, an efficient implementation of the corresponding coding schemes is essential. Although in a different context, Sheppard et al. [23] designed and implemented lattice coding schemes using digital signal processing techniques, and Wang et al. [24] integrate the leech lattice into IEEE 802.11a. Recent studies explore lattice codes for future 5G networks [25]. So, thus far, compute-and-forward based on lattice codes has been a theoretical topic, whereas we demonstrate the first implementation on a small SDR testbed.

III. Problem Statement and Scope

Motivated by emerging embedded wireless applications [1], [2], our goal is to design a communication scheme that allows K sources to distribute their packets to all nodes in a multi-hop network as fast as possible. Communication must be energy-efficient and reliable also when some or all nodes are mobile.

To achieve our goal, we build upon Glossy and PLNC. Section IV provides background on both these topics. In particular, our proposed MF-Glossy scheme exploits a specific flavor of PLNC known as compute-and-forward [11] for two main reasons: (*i*) unlike amplify-and-forward [10], nodes participating in the communication decode the received signals, which is a prerequisite if every node should obtain all packets, and (*ii*) it can achieve higher goodput than all other flavors of PLNC.

Using compute-and-forward means that we require a different physical layer than available on, for example, off-the-shelf ZigBee or Bluethooth Low Energy (BLE) devices. Specifically, conventional modulation and coding schemes are replaced by lattice codes (see Section IV). Nevertheless, all other layers in existing communication stack can remain unchanged.

IV. Background

This section briefly reviews the operation of state-of-the-art Glossy and then provides some background on PLNC.

A. Operation of State-of-the-art Glossy

Figure 2 shows the actions during a Glossy flood in a multi-hop network. An *initiator* starts the flood by transmitting the packet; all other nodes have their radio on (1). Since wireless is a broadcast medium, all nodes within transmission range of the initiator, the *1-hop receivers*, receive the packet at about the same time (2). After a minimal processing delay, the 1-hop receivers relay the *same* packet at about the same time (3). Even though these *synchronous transmissions* collide at the *2-hop receivers*, these nodes can successfully receive and decode the packet with high probability (4). Then, the 2-hop receivers again re-transmit the same packet synchronously, thereby propagating the flood deeper into the network (5). As a result, the flood spreads like a wave and reaches out to all nodes (6). Since each node transmits multiple times during the same flood up to a certain *re-transmission limit* N, there are multiple waves, boosting Glossy's reliability above 99.9 % [7].

To enable successful packet reception, Glossy aligns *identical* wireless signals from multiple concurrent senders within the 0.5 μs bound that allows them to interfere non-destructively using the IEEE 802.15.4 physical layer [7]. The synchronization of the concurrent senders is established on the fly and in a distributed fashion by using packet receptions during a flood as a reference point. For example, in Figure 2, the simultaneous reception of the packet from the 1-hop receivers in (4) serves to align the transmissions of the 2-hop receivers in (5). Glossy achieves this through a careful software design that makes the processing time between reception and transmission as short and deterministic as possible. Glossy also capitalizes on the *capture effect* that lets a receiver demodulate only the strongest of multiple overlapping signals [16]. In addition, Glossy can time-synchronize nodes with sub-microsecond accuracy [7].

B. Physical-layer Network Coding (PLNC)

The concept of linear network coding was first proposed for wired networks in [17]. The basic idea is to allow intermediate nodes in a network to combine the received data packets and forward a superposition. The intended receivers will be able to decode the original data packets if they receive enough linearly independent combinations of the original data packets.

In a wireless setup, it is possible to exploit the superposition property of the wireless channel. The transmitted signals are combined in the air and the receiver gets a superposition plus noise. Since the network coding occurs on the physical layer, we call this kind of network coding PLNC. If the receiver is an intermediate node that will relay a linear combination of the received signals, we call this node a *relay*. A relay can choose among different relaying strategies, including

- *decode-and-forward*, where the relay decodes the individual data and forwards a new linear combination;
- *amplify-and-forward*, where the relay amplifies the received superposition and forwards the signal;
- *compute-and-forward*, where the relay directly decodes a linear combination of the original data.

Using decode-and-forward strategy, a relay decodes just one signal and treats all others as noise. Treating interference as noise reduces the achievable rate. Amplify-and-forward does not decode the signals, but forwards an amplified version of the received signal. The downside is the noise amplification, which accumulates as the signal is forwarded through the network.

Compute-and-forward combines the advantages of both: The relay decodes a linear combination of the signal (interference is exploited and not treated as noise) and the noise is removed from the signal (no noise amplification and accumulation).

Channel codes such as Reed-Solomon provide forward error correction (FEC) for communication over noisy channels. The data to be sent is divided into messages from an index set $\{1, \dots, 2^{nR}\}$, where $R = \frac{q}{n}$ is the code rate, n is the code length, and q is the code dimension (*i.e.*, the message length). After encoding, the codewords get mapped to complex-valued samples using a modulation scheme. However, concentional modulation schemes, such as quadrature amplitude modulation (QAM), are not suitable for compute-and-forward, because the superposition of two modulation points is not necessarily a valid modulation point. For this reason, compute-and-forward uses *nested lattice codes*, which offer the required properties.

In the following, we define a few necessary terms related to nested lattice codes and the compute-and-forward framework. For a more detailed introduction, we refer the reader to [26]. Afterwards, we use an example to illustrate the communication using compute-and-forward based on lattice codes. Lets assume there are K source nodes and M relay nodes.

Definition 1 (Lattice)**.** A lattice Λ is a subgroup of $\mathbb{C}^n$ which is isomorphic to $\mathbb{Z}^n + j\,\mathbb{Z}^n$. If $s, t \in \Lambda$, then $s + t \in \Lambda$.

Definition 2 (Nearest Neighbor Quantizer)**.** The nearest neighbor quantizer Q is defined as

$$Q(x) \triangleq \arg\min_{\lambda \in \Lambda} ||x - \lambda||. \tag{1}$$

Definition 3 (Voronoi Region)**.** The fundamental Voronoi region $\mathcal{V}$ of a lattice Λ is the set of all points in $\mathbb{C}^n$ closest to the zero point, that is, $\mathcal{V} = \{x : Q(x) = 0\}$.

Definition 4 (Quantization Error)**.** The quantization error can be expressed by the modulo-Λ operation with respect to the lattice, which is defined as $x \bmod \Lambda = x - Q(x)$.

Definition 5 (Nested Lattice Code)**.** A nested lattice code $\mathcal{L}$ is the set of all points of a fine lattice Λ_F that are within the fundamental Voronoi region $\mathcal{V}_C$ of a coarse lattice $\Lambda_C \subset \Lambda_F$,

$$\mathcal{L} = \Lambda_F \cap \mathcal{V}_C = \{\lambda \bmod \Lambda_C,\, \lambda \in \Lambda_F\}. \tag{2}$$

Definition 6 (Lattice Equation)**.** A lattice equation v_m at relay m is an integral combination of lattice codewords x_k modulo the coarse lattice

$$v_m = \left[\sum_{k=1}^{K} a_{mk} x_k\right] \bmod \Lambda_C \tag{3}$$

with $a_{mk} \in \mathbb{Z} + j\,\mathbb{Z}$. We call $a_m = (a_{m1}, \dots, a_{mK})^T$ the lattice coefficient vector.

Figure 3 illustrates encoding, transmission, and decoding of two messages over an AWGN channel using the compute-and-forward approach. Each transmitter encodes its messages by a nested lattice code; that is, it maps a message $w_k \in \mathbb{F}_p^q$ to a lattice codeword $x_k \in \mathcal{L} \subset \mathbb{C}^n$. This encoding process is equivalent to modulation schemes in classical communication systems. Whereas a classical modulation scheme maps messages to one-dimensional complex-valued samples, a lattice

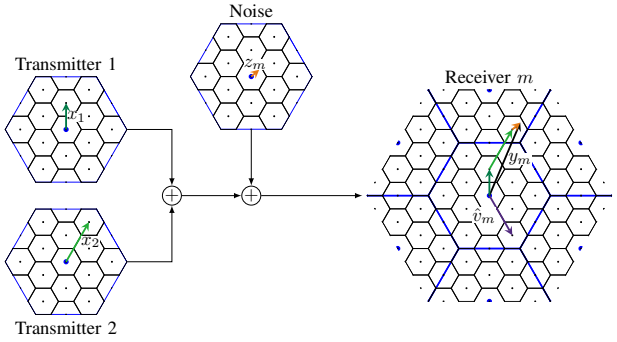


Figure 3: Illustration of a compute-and-forward transmission over an additive white Gaussian noise (AWGN) channel using nested lattice codes.

encoder maps messages to n-dimensional complex-valued samples, which are transmitted sequentially.

Receiver m in Figure 3 now aims to decode a lattice equation specified by the lattice coefficient vector a_m. It receives a noisy superposition of the transmitted codewords

$$y_m = \sum_{k=1}^{K} h_{mk} x_k + z_m, \tag{4}$$

where each transmitted codeword x_k is multiplied by a channel coefficient h_{mk} and summed up. Additionally, a noise vector z_m with $z_m \sim \mathcal{CN}(0, \sigma^2 I_n)$ gets added to the received signal. The channel coefficients are complex values, whereas each relay wants to decode a linear combination with coefficients from the set of Gaussian integers. Thus, a receiver must scale the channel output with a factor α_m, so $s_m = \alpha_m y_m$. The goal is to scale the received signal such that the channel values become integral values. However, there is a trade-off because scaling the received signal also scales the noise. The optimal scaling factor for a desired lattice equation with coefficients a_m is the minimum mean square error (MMSE) coefficient

$$\alpha_m = \frac{P h_m^H a_m}{\sigma^2 + P||h_m||^2}. \tag{5}$$

To get an estimate of the lattice equation v_m, vector s_m is quantized onto the fine lattice Λ_F modulo the coarse lattice Λ_C

$$\hat{v}_m = [Q(s_m)] \bmod \Lambda_C. \tag{6}$$

If the noise z_m is small enough such that $z_m \in \mathcal{V}_F$, the quantization step removes the noise and the linear combination can be reliably decoded.

Since each decoder aims for a superposition of codewords, we call the rate at which decoding is reliable *computation rate*. The achievable computation rate of the compute-and-forward framework is given by

$$\mathcal{R}(h_m, a_m) = \log_2^+\left(\left(||a_m||^2 - \frac{P|h_m^H a_m|^2}{\sigma^2 + P||h_m||^2}\right)^{-1}\right) \tag{7}$$

with $\log_2^+(x) \triangleq \max\{0, \log_2(x)\}$. This means all relays can simultaneously decode equations with coefficients a_m as long as the message rates are within the computation rate region

$$R_k < \min_{a_{mk}} \mathcal{R}(h_m, a_m). \tag{8}$$

Algorithm 1 Key operating steps of a node in MF-Glossy

1: **while** retransmission counter $\leq$ retransmission limit N **do**
2: **Wait for incoming signal.** Listen to the channel waiting for a transmitted signal to be detected.
3: **Receive signal.** When a signal is detected, receive it and verify successful reception by checking that there exists a coefficient vector a for which the achievable rate $\mathcal{R}(\tilde{h}, a)$ is greater than or equal to the source rate R, $\mathcal{R}(\tilde{h}, a) \geq R$, where $\mathcal{R}(\tilde{h}, a)$ is given by (7). If the rate condition is not satisfied, reception is unsuccessful and the node goes back to 2.
4: **Decode signal.** After successful signal reception, decode and store any linear combination of source packets with associated coefficient vector a for which $\mathcal{R}^*(\tilde{h}, a) \geq R$. The decoding maps each coefficient vector to the corresponding lattice point.
5: **Rank check.** Check if the rank of the coding coefficient matrix is at least K. If so, retrieve the K source packets.
6: **Encode signal.** Choose and encode one linear combination to be transmitted provided it is linearly independent on previously transmitted linear combinations. The encoding process maps each linear combination to the closest lattice point.
7: **Transmit signal.** Transmit the corresponding signal and increment the retransmission counter.
8: **end while**

V. Multi-flow Glossy (MF-Glossy)

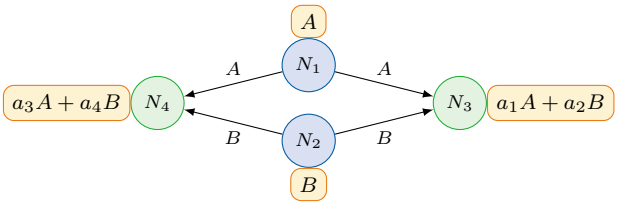
We now present MF-Glossy, a many-to-all communication scheme that exploits PLNC based on compute-and-forward to simultaneously flood $2 \leq K \leq V$ different packets (or *flows*) from K different sources to all V nodes in a multi-hop wireless network. The source nodes initiate the flood by synchronously transmitting their packets at the same source rate R. Afterward, each node follows the operating steps shown in Algorithm 1 until it reaches the re-transmission limit N. In essence, MF-Glossy inserts steps 2–6 of Algorithm 1 between the Rx and Tx phases of state-of-the-art Glossy as shown in Figure 2.

Figure 4 shows an example with $K = 2$ packets injected by source nodes N_1 and N_2; the re-transmission limit is $N = 2$. For reasons of exposition, we focus on nodes N_1–N_4, which may in fact be part of a larger wireless multi-hop network.

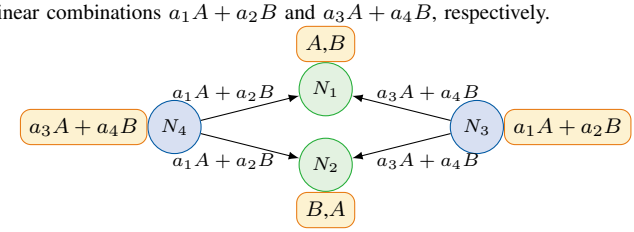
In the first time slot (see Figure 4a), nodes N_1 and N_2 simultaneously transmit two different packets A and B, which are encoded using lattice codes. Due to the broadcast nature of the wireless channel, nodes N_3 and N_4 receive a superposition of the transmitted signals and each decode a linear combination of the source packets (a_1A+a_2B and a_3A+a_4B, respectively) as explained in steps 3 and 4 of Algorithm 1.

In the second time slot (see Figure 4b), N_3 and N_4 transmit an encoded linear combination as explained in steps 6 and 7 of Algorithm 1, which are received and decoded by nodes N_1 and N_2. According to step 5 in Algorithm 1, because nodes N_1 and N_2 have acquired $K = 2$ independent linear combinations, they can already retrieve the two source packets A and B.

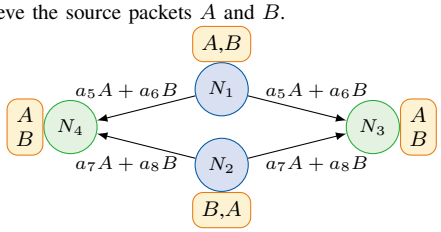
In the third time slot (see Figure 4c), N_1 and N_2 transmit again linear combinations $a_5A + a_6B$ and $a_7A + a_8B$ of packets A and B, since they have not yet reached the re-transmission limit $N = 2$. Nodes N_3 and N_4 receive and decode linear combinations of the transmitted linear combinations. Assuming that the newly received linear combinations are independent of the previously received linear combinations, they can now retrieve the source packets A and B.



(a) Time slot 1: Nodes N_1 and N_2 initiate the flood by simultaneously transmitting different source packets A and B. N_3 and N_4 receive and decode linear combinations $a_1A + a_2B$ and $a_3A + a_4B$, respectively.



(b) Time slot 2: Nodes N_3 and N_4 encode and transmit their respective linear combinations. N_1 and N_2 receive and decode linear combinations, allowing them to retrieve the source packets A and B.



(c) Time slot 3: Nodes N_1 and N_2 transmit again a linear combination of packets A and B, *i.e.*, $a_5A + a_6B$ and $a_7A + a_8B$, respectively. N_3 and N_4 receive and decode new linear combinations. Assuming these are linearly independent on their previously received linear combinations, they can now also retrieve the source packets A and B.

Figure 4: Example illustrating the operation of MF-Glossy with $K = 2$ source packets injected by nodes N_1 and N_2; the re-transmission limit is $N = 2$. Shows is only the communication among nodes N_1–N_4, which may in fact be part of a much larger multi-hop wireless network.

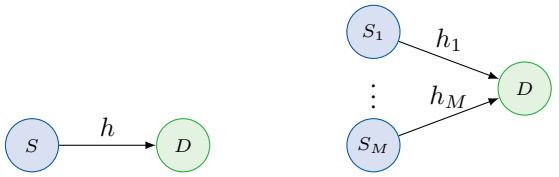
VI. Communication-theoretic Analysis

In this section, we derive bounds on the outage probability (*i.e.*, packet loss rate) of Glossy and MF-Glossy These bounds provide valuable insights on the theoretically possible goodput of Glossy and MF-Glossy, and thus allow us to quantitatively compare both schemes independent of the implementation.

A. Glossy

To derive the outage probability of Glossy, we decompose a full flood. At the start of a flood, there is one node sending to one or several nodes (see (1) and (2) in Figure 2), which can be modeled as point-to-point channels as depicted in Figure 5a. During the flood, instead, multiple nodes broadcast their data to their neighbors: one node receives from several transmitting nodes (see, for example, (3) and (4) in Figure 2). This can be modeled as multiple-access channels as depicted in Figure 5b.

Point-to-point channel. We start with the point-to-point channel shown in Figure 5a. Node S transmits a signal x at rate R and node D receives the signal $y = xh+z$, where z is AWGN with $z \sim \mathcal{CN}(0, \sigma^2)$ and $h \in \mathbb{C}$ is the channel gain. The



(a) Point-to-point channel. (b) Multiple-access channel.

Figure 5: The two basic cases in our communication-theoretic analysis.

channels for low-power wireless devices are typically slow-fading channels (*i.e.*, they are constant for some time). The maximum achievable rate over this channel is upper bounded by its capacity, which is given by [27, Chapter 2]

$$C_{\mathrm{P2P}} = \log_2\left(1 + \frac{|h|^2 P}{\sigma^2}\right), \tag{9}$$

where $P = \mathrm{E}[x^2]$ is the average transmit power of node S. If S sends at a rate higher than the channel capacity, the reception is not error-free and outage occurs. Hence, the outage probability (*i.e.*, packet loss rate) is given by $\Pr(C_{\mathrm{P2P}} < R)$ [28].

Multiple-access channel. We turn to the multiple-access case shown in Figure 5b. Each node S_m with $m \in \{1, 2, \ldots, M\}$ simultaneously transmits a signal x_m at rate R to node D.

$$y = \sum_{m=1}^{M} x_m h_m + z, \tag{10}$$

where $h_m \in \mathbb{C}$ is the channel gain between S_m and D. In Glossy, all nodes send the same signal, so $x_1 = \cdots = x_m$.

If the transmitters have no channel state information (CSI), the maximum achievable rate R_{MP} is the maximum rate at which node D can successfully receive all data transmitted by nodes $S_1, S_2, \ldots, S_m$. This rate is given by [29]

$$R_{\mathrm{MP}} = \log\left(1 + \frac{|\sum_{m=1}^{M} h_m \sqrt{P_m}|^2}{\sigma^2}\right), \tag{11}$$

where $P_m = \mathrm{E}[|x_m|^2]$ is the average transmit power of node S_m. The corresponding outage probability is $\Pr(R_{\mathrm{MP}} < R)$.

B. MF-Glossy

In MF-Glossy, the point-to-point channel model is valid as derived above. However, the multiple-access channel model changes, because node D decodes a superposition of the signals instead of the individual signals. We assume that there are K flows during a flood, that is, K nodes initiated a flood, whereas each node k with $k \in \{1, 2, \ldots, K\}$ sent a data packet d_k. Using the notation in Figure 5b, we assume that each node S_m transmits a linear combination $x_m = \sum_{k=1}^{K} a_{mk} d_k$ of the data packets d_k. Node D receives the signal

$$y = \sum_{m=1}^{M} h_m x_m + z = \sum_{m=1}^{M} h_m \left(\sum_{k=1}^{K} a_{mk} d_k\right) + z \tag{12a}$$

$$= \sum_{m=1}^{M}\sum_{k=1}^{K} h_m a_{mk} d_k + z = \sum_{k=1}^{K} d_k \sum_{m=1}^{M} h_m a_{mk} + z \tag{12b}$$

$$= \sum_{k=1}^{K} \tilde{h}_k d_k + z, \tag{12c}$$

where $\tilde{h}_k = \sum_{m=1}^{M} h_m a_{mk}$ is the effective channel from initiator k to node D. The maximum achievable rate for decoding a linear combination a_D of the data packets d_k follows from (7) and is given by $\mathcal{R}(\tilde{h}, a_D)$. The outage probability (*i.e.*, packet loss rate) is $\Pr(\mathcal{R}(\tilde{h}, a_D) < R)$.

All coefficient vectors a_D for which $\mathcal{R}(\tilde{h}, a_D)$ is greater than or equal to the transmit rate R are used for decoding and stored. If a node cannot find a coefficient vector that satisfies the rate condition, the reception is considered unsuccessful and the node goes back to the listening state. In order to maximize the benefits of each transmission, each node only transmits a linear combination with a coefficient vector if this vector is linearly independent of previously transmitted vectors. As soon as K linearly independent coefficient vectors are acquired by a node, the source packets are decoded at this specific node. The resulting outage probability at node D for the entire flood is the probability that node D acquires less than K linearly independent combinations of the original data packets d_k.

VII. Simulation Results

We use the models from Section VI to evaluate and compare the performance of MF-Glossy and Glossy in simulation.

A. Settings and Metrics

Network topology. We generate network topologies using a binomial point process with λ nodes randomly and independently placed in a square of side length L. We vary λ and L to study the impact of the network topology on performance; our results are averaged over 10,000 independent realizations for the same λ and L. Irrespective of the network topology, all nodes transmit with rate $R = 250$ kbps and a power of 0 dBm, which corresponds to the default and maximum settings as prescribed by the IEEE 802.15.4 standard, respectively.

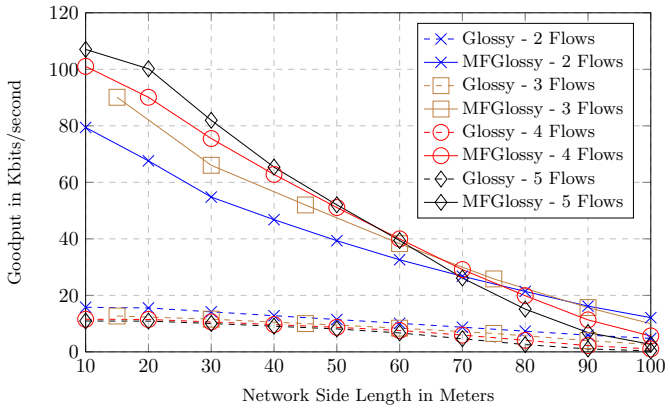
Channel model. We model the channel gains h_i as stationary independent according to $h_i = w_i g_i$, where small-scale fading w_i follows a complex normal distribution $\mathcal{CN}(0, 1)$, and large-scale fading g_i depends on the distance d between transmitter and receiver [30, E 5.3]

$$g_i = \begin{cases} 40.2 + 20\log(d) & , d \le 8\,\mathrm{m} \\ 58.5 + 33\log(d/8) & , d > 8\,\mathrm{m} \end{cases} \tag{13}$$

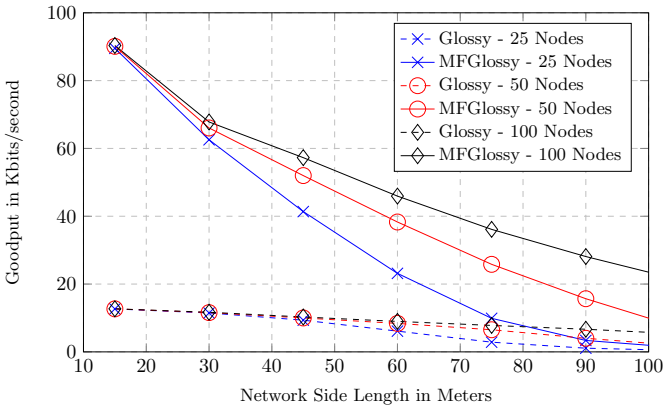
This corresponds to a path loss exponent of 2 for the first 8 m and a path loss exponent of 3.3 for distances larger than 8 m. To ensure $g_i > 0$ according to (13), we consider only topologies where the distance between any pair of nodes exceeds 0.1 m, which is reasonable in real deployments [31].

Compared schemes. We compare our proposed MF-Glossy against state-of-the-art Glossy. To do so in a fair manner for $K \ge 2$ flows, using Glossy, we consider K successive floods initiated by K different randomly selected nodes; our results for Glossy refer to these K successive floods as a whole. We set the re-transmission limit of Glossy to $N = 3$ based on experience from extensive real-world experiments [7], [8], and use $N = 6$ for MF-Glossy. We study the impact of N on the performance of both schemes in dedicated simulation runs.

Performance metrics. We consider the following key performance metrics of real-world applications [32]:



(a) Impact of the number of flows in a 50-node network.



(b) Impact of the number of nodes for 3 flows.

Figure 6: Goodput of MF-Glossy and Glossy against network side length.

- *Packet reception ratio (PRR)*: The number of nodes that correctly receive (or decode, in case of MF-Glossy) all packets divided by the total number of nodes λ.
- *Latency*: The time from the start of the flood until a node correctly receives (decodes) all packets, averaged over all nodes in the network. Given a certain packet size, the duration of a single slot during a Glossy flood is a network-wide constant [7]. We compute latency of Glossy and MF-Glossy for 8-byte packets, purposely ignoring the overhead of the de- and encoding in MF-Glossy to get an upper bound on the theoretically possible performance gains. As result, varying the packet size has no impact on the relative performance between MF-Glossy and Glossy in our simulations. We evaluate and discuss MF-Glossy's overhead in Section VIII using dedicated experiments.
- *Global energy consumption*: The total energy consumed by all λ nodes for communication between the start and the end of a MF-Glossy flood or K consecutive Glossy floods. To compute energy consumption, we take the current draws from the data sheet of the widely used CC2420 radio chip in states transmit, receive, listening, and idle, assuming batteries constantly supply 2000 mAh at 3 V. We consider a node to be idle only when its radio is turned off after reaching its re-transmission limit N.
- *Goodput*: The average amount of data a node receives (or decodes) successfully per time unit. Formally, we calculate goodput as $(PRR \times PayloadSize \times K)/Latency$. So, in a sense, goodput is a measure of the level of "service" provided, while energy measures the associated "cost."

B. Results

Goodput. Figure 6a shows goodput of MF-Glossy and Glossy against network side length for 50 nodes and different number of flows. We see that by serving multiple flows within the same flood MF-Glossy achieves several-fold improvements over Glossy across a wide range of network side lengths (*i.e.*, node densities). Using more flows K benefits MF-Glossy up to a network wide length of about 60 m, while Glossy generally suffers as K increases. We also see that the decrease in goodput for MF-Glossy becomes steeper for larger K till we reach crossing points that tell us exactly which value of K to use for a given network size. Figure 6b shows goodput against network side length for $K = 3$ flows and different number of nodes. Again, we see that MF-Glossy is particularly good at leveraging a higher node density to boost goodput, which confirms the trend we observed before in Figure 6a.

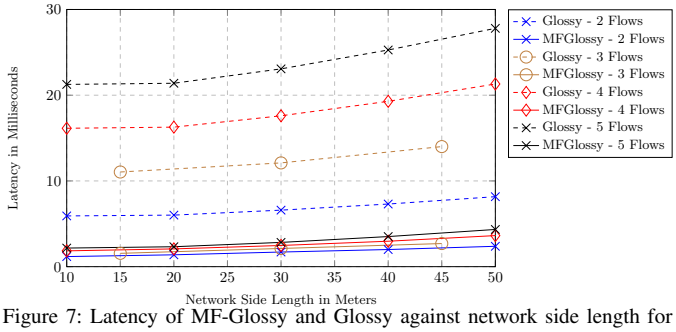


Figure 7: Latency of MF-Glossy and Glossy against network side length for 50 nodes and different number of flows.

Latency and PRR. To understand the goodput results, we look at latency and PRR as a function of network side length for 50 nodes and different number of flows. Looking at Figure 7, we see that the latency of Glossy increases linearly and significantly with the number of flows, because each flow is mapped onto a single independent flood. MF-Glossy, instead, accommodates multiple flows in the same flood, which comes only at a slight increase in latency per flow. As a result, for 5 flows, MF-Glossy reduces latency by about 9× compared with Glossy. Note that the comparison in Figure 7 is fair only when considering the same number of flows K.

Looking at Figure 8, we observe a faster decay in PRR as the number of flows increases. This is expected because, intuitively, delivering, say, 5 packets successfully is more difficult than delivering only 4 packets successfully. Nevertheless, by increasing the re-transmission limit N, it is possible to boost the PRR of MF-Glossy at the expense of a higher latency.

Goodput vs. energy. Having examined the factors impacting the level of "service" provided by MF-Glossy and Glossy, we now relate this to the associated "costs." To this end, we plot in Figure 9 goodput against global energy consumption for different number of flows K and re-transmission limits N, considering a 50-node network that is 60 m × 60 m in size.

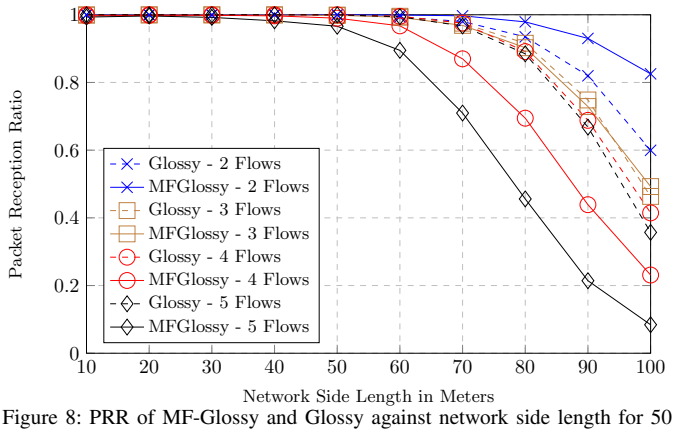


Figure 8: PRR of MF-Glossy and Glossy against network side length for 50 nodes and different number of flows.

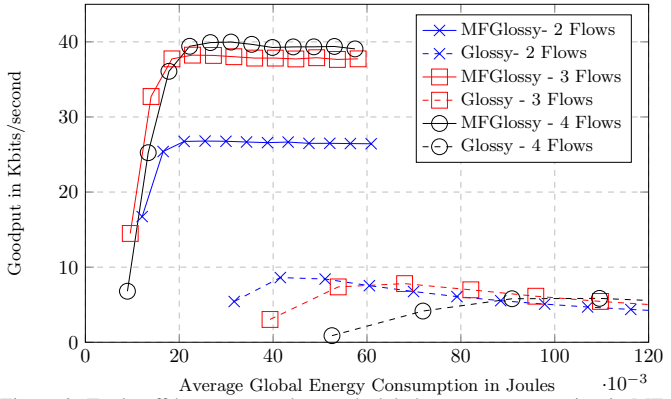


Figure 9: Trade-off between goodput and global energy consumption in MF-Glossy and Glossy for different number of flows and re-transmission limits, in a network that consists of 50 nodes and is 60 m × 60 m in size.

We see that MF-Glossy provides a significantly better trade-off than Glossy: MF-Glossy always provides higher goodput or reduced energy consumption without impairing the other metric in comparison to Glossy. Moreover, we find that for a given number of flows K there exists a distinct setting for the re-transmission limit N that maximizes goodput. Increasing N beyond this point helps PRR, yet this improvement cannot counter the effect of increased latency, thus ultimately resulting in lower goodput and higher energy consumption.

We also see that using more flows K increases the goodput of MF-Glossy, but the relative improvements become smaller and smaller for higher K. Based on the network characteristics and application requirements at hand, users can decide on the best parameter setting (*e.g.*, N and K), thereby trading higher (lower) goodput for higher (lower) energy consumption.

VIII. Proof-of-concept Implementation

MF-Glossy uses the compute-and-forward framework based on lattice codes. We report here on the first implementation of this framework on real hardware, thus demonstrating that the key concept underlying MF-Glossy is indeed implementable.

Hardware setup. We use three USRP N210 SDRs. Two act as transmitters and one acts as a receiver, as shown in Figure 10. This setup corresponds to that happens during the first time slot of a MF-Glossy flood with $K = 2$ sources. The SDRs are time-synchronized by both a 10 MHz and a pulse per second

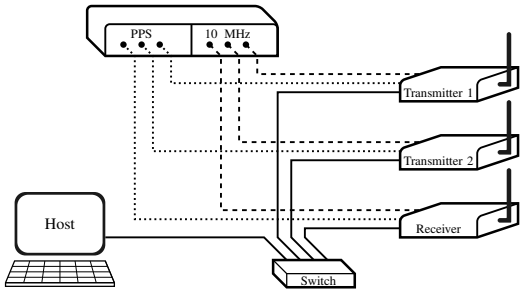


Figure 10: Hardware setup for proof-of-concept implementation.

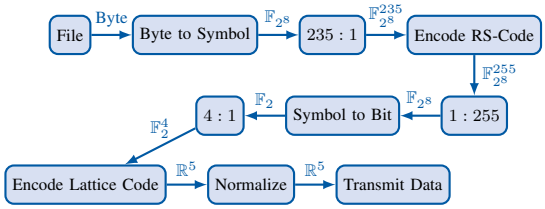


Figure 11: Lattice encoding steps in proof-of-concept implementation.

(PPS) signal. A host PC, which features an Intel i5 processor, interacts with the three SDRs via a switch over Ethernet.

Software components. We use GNU radio [33] to implement the interaction between the host PC and the three SDRs, and SageMath [34] to implement the encoding and decoding on the host. Encoding proceeds according to the steps shown in Figure 11; decoding proceeds in the reverse direction. In addition to the lattice code, we use a Reed-Solomon code for forward error correction. GNU radio as well as SageMath are general open-source frameworks, and hence do not provide optimized algorithms for our peculiar lattice encoding and decoding operations. Nevertheless, our implementation provides a prototype that allows us to evaluate the performance of the compute-and-forward framework in a practical setting.

Method. To this end, we let the two transmitters simultaneously send two different 19 kB files. After encoding both files on the host, this triggers 80 simultaneous transmissions of 235-byte data frames by the two transmitters. The receiver forwards the received signals to the host, which performs the decoding.

Results. All data frames were correctly decoded. On the host, we measure 2.4 ms and 5.3 ms for encoding and decoding one byte, respectively. Including these processing overheads in our simulations, we get, for example, a latency of 241 ms for 5 flows with MF-Glossy in a 50-node network with a side length of 45 m (see Figure 7). Thus, about 10× faster en- and decoding are required for MF-Glossy to be faster than Glossy, which has a latency of 28 ms for this network configuration.

Challenges and solutions. We see that encoding and decoding take a significant amount of time. Nevertheless, a very large part is spent in methods that just provide data models for finite fields, vector and matrix arithmetic, etc. Significant speed-ups are possible by using optimized libraries for these operations (*e.g.*, [35]). Further, it is possible to optimize the algorithms for the modulo operation of a certain lattice. Since the modulo operation is costly and frequently used, optimizing it will bring a significant performance boost. In standardized protocols, there is typically a finite set of modulation and coding schemes

from which the nodes choose depending on certain system parameters. The same holds for lattice codes. One can design a set of good lattices with optimized algorithms from which the nodes choose one. Also, efficient algorithms exist to compute the coding coefficients [36] and to map the receive vector to a valid lattice point [37], which we exploited only partially. Thus, we believe an implementation on embedded devices is indeed possible, especially in light of the trend towards more powerful, yet highly energy-efficient 32-bit microcontrollers with rich instructions sets (*e.g.*, ARM Cortex-M series).

A second challenge is time and frequency synchronization of the distributed nodes. The compute-and-forward framework assumes perfect synchronization, yet we demonstrate its feasibility in practice on SDRs. It must be seen how large the delays and offsets among nodes (without dedicated synchronization hardware) can be without sacrificing performance.

IX. Conclusions

We have explored the opportunities and challenges of using PLNC to enable the simultaneous flooding of different packets from different sources to all nodes in the network. We thus introduced MF-Glossy and performed a communication-theoretic analysis of Glossy and MF-Glossy to determine upper bounds on their performance. Simulation results show that MF-Glossy has the potential to achieve several-fold improvements in goodput and latency at lower energy costs and comparable packet reception rates. A proof-of-concept implementation on SDRs demonstrates that our design is indeed implementable. Our work thus represents the first step towards utilizing PLNC in embedded wireless networks, and shows its potential—a potential that, if harvested efficiently, can result in huge performance gains and resource savings.

Acknowledgments. This work was supported by the German Research Foundation (DFG) within the Cluster of Excellence "Center for Advancing Electronics Dresden (cfaed)" and through Priority Programs 1914, and by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie project 641985 (ETN-5Gwireless).

References

[1] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: Challenges and directions," *IEEE Trans. on Industrial Informatics*, vol. PP, no. 99, 2016.

[2] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, 2016.

[3] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proc. of IEEE INDIN*, 2011.

[4] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[5] BBC News, "US military tests swarm of mini-drones launched from jets," http://www.bbc.com/news/technology-38569027.

[6] RCR Wireless News, "5 connectivity challenges for drones," http://www.rcrwireless.com/20160802/europe/five-challenges-drones-tag28.

[7] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with Glossy," in *Proc. of the ACM/IEEE IPSN*, 2011.

[8] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *Proc. of the 10th ACM SenSys*, 2012.

[9] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. of the 12th annual international conference on Mobile computing and networking*, vol. 1, no. Node 1, 2006, pp. 358–365.

[10] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *SIGCOMM*, Oct. 2007, pp. 397–408.

[11] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. on Information Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[12] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. on Information Theory*, vol. 50, no. 12, 2004.

[13] H. Rahul, H. Hassanieh, and D. Katabi, "Sourcesync: A distributed wireless architecture for exploiting sender diversity," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, Aug. 2010.

[14] P. Dutta, R. Musăloiu-E., I. Stoica, and A. Terzis, "Wireless ACK collisions not considered harmful," in *ACM HotNets*, 2008.

[15] O. Landsiedel, F. Ferrari, and M. Zimmerling, "Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale," in *Proceedings of the 11th ACM SenSys*, NY, USA, 2013, pp. 1:1–1:14.

[16] K. Leentvaar and J. Flint, "The capture effect in FM receivers," *IEEE Trans. Commun.*, vol. 24, no. 5, 1976.

[17] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[18] S. Zhang and S. C. Liew, "Applying physical-layer network coding in wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 1–12, 2010.

[19] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proceedings of the IEEE*, vol. 99, no. 99, pp. 438–460, 2011.

[20] C. Feng, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Trans. on Information Theory*, vol. 59, no. 11, pp. 7576–7596, Nov. 2013.

[21] Z. Chen, B. Xia, Z. Hu, and H. Liu, "Design and analysis of multi-level physical-layer network coding for gaussian two-way relay channels," *IEEE Trans. on Communications*, vol. 62, no. 6, pp. 1803–1817, 2014.

[22] U. Erez and R. Zamir, "Achieving 1/2 log (1+ SNR) on the AWGN channel with lattice encoding and decoding," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2293–2314, 2004.

[23] J. A. Sheppard and A. G. Burr, "The design and implementation of lattice codes using digital signal processing techniques," in *Sixth Int. Conf. on Digital Processing of Signals in Communications*, 1991.

[24] N. Wang and J. D. Gibson, "Leech lattice coding and modulation for ieee 802.11a WLAN," in *IEEE Emerging Technologies Symposium on BroadBand Communications for the Internet Era*, 2001.

[25] L.-J. Hu, C. Duan, D.-F. Zhao, and X. Liao, "Study status and prospect of lattice codes in wireless communication," in *Fifth IMCCC*, Sep. 2015.

[26] R. Zamir, *Lattice Coding for Signals and Networks*. Cambridge University Press, 2014.

[27] A. Gamal and Y. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[28] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.

[29] P. Marsch and G. P. Fettweis, *Coordinated Multi-Point in Mobile Communications: From Theory to Practice*, 1st ed. New York, NY, USA: Cambridge University Press, 2011.

[30] "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless l (MAC) and (PHY) specifications for low rate (WPANs)," *IEEE Std 802.15.4-2006*.

[31] M. Ceriotti *et al.*, "Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment," in *ACM/IEEE IPSN*, 2009.

[32] ——, "Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels," in *ACM/IEEE IPSN*, 2011.

[33] The GNU Radio Foundation, Inc. (2016) GNU Radio Website. [Online]. Available: http://www.gnuradio.org

[34] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 7.4)*, 2016. [Online]. Available: http://www.sagemath.org

[35] Steinwurf, "fifi – optimized finite fields," 2017. [Online]. Available: http://steinwurf.com/_products/fifi.html

[36] J. Richter, C. Scheunert, and E. A. Jorswieck, "An efficient branch-and-bound algorithm for compute-and-forward," in *IEEE PIMRC*, 2012.

[37] A. Sakzad, J. Harshan, and E. Viterbo, "Integer-forcing MIMO linear receivers based on lattice reduction," *IEEE Trans. on Wireless Communications*, vol. 12, no. 10, pp. 4905–4915, October 2013.