Concurrent Transmission Aware Routing in Wireless Networks

Shuai Wang, Song Min Kim, Linghe Kong, and Tian He, *Fellow, IEEE*

***Abstract*—Recent physical-layer designs capable of decoding multi-packet collisions demonstrate great potential for improving the network performance. Current network protocols, however, tailor the traditional physical layer to avoid collisions and thus cannot fully exploit the benefits of concurrent transmission techniques. In this paper, we propose an innovative generic concurrent transmission aware routing design called mShare. The mShare design exploits the benefits of these techniques by scheduling concurrent senders to utilize co-owned receiver(s) in parallel. This design significantly increases the available number of routing choices and thus improves the network performance. To illustrate the versatility of our design, we test mShare in three settings: unicast, opportunistic routing, and data collection (convergecast). The performance of mShare is evaluated with physical testbed experiments running on USRP and simulations. The experimental results show that compared to conventional designs, mShare: 1) improves 277% of the throughput in unicast; 2) saves 78% of transmissions in opportunistic routing; and 3) reduces 70% of the delivery delay in data collection.**

***Index Terms*—Concurrent transmissions, collision resolution, routing, wireless networks.**

I. Introduction

WITH previous upgrades in network hardware and developments in coding/decoding techniques, the physical-layer is able to decode concurrent transmissions from multiple senders. A series of advanced concurrent transmission techniques, such as successive interference cancellation [1], [2], constructive interference [3]–[6], and mZig [7], have been proposed that leverage physical-layer features to separate collisions into non-collided packets and successfully decode them at the same time. These concurrent transmission techniques have considerable potential for improving network performance by (i) reducing the number of retransmissions and delays, (ii) improving the throughput and delivery ratio, and (iii) saving energy consumption.

Manuscript received March 4, 2018; revised May 25, 2018; accepted July 8, 2018. Date of publication July 26, 2018; date of current version December 14, 2018. This work was supported in part by the National Science Foundation under Grant CNS-1525235, Grant CNS-1444021, Grant CNS-1717059, Grant CNS-1718456, and NSFC 61672349. The associate editor coordinating the review of this paper and approving it for publication was J. Luo. *(Corresponding author: Tian He.)*

S. Wang was with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA. He is now with the Department of Computer Science, Southeast University, Nanjing 211100, China (e-mail: wangshuai87410@gmail.com).

S. M. Kim is with the Department of Computer Science, George Mason University, Fairfax, VA 22030 USA.

L. Kong is with the Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China.

T. He is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: tianhe@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2018.2859927

Although existing network layer protocols partner well with traditional physical-layer designs, they cannot fully exploit the benefits of concurrent transmission techniques. Specifically, existing network layer designs prohibit multiple senders from utilizing a receiver in parallel to avoid collisions even though the receiver can decode concurrent transmissions with support from the physical layer.

This paper aims to enhance concurrent transmission techniques from a networking perspective by proposing a new generic concurrent transmission aware routing design capable of better exploiting the benefits of different physical-layer concurrent transmission techniques. The key idea of this design is to schedule a maximum number of concurrent transmissions by letting concurrent senders utilize co-owned receivers in parallel. This mechanism significantly increases the number of available receivers for network routing and thus can significantly improve network performance.

To achieve this in practice, we need to address two challenges. First, we need to make an efficient lightweight schedule to construct concurrent transmissions to maximize the use of all the potential forwarders. Second, the concurrent transmission techniques may fail to decode concurrent transmissions when too many packets arrive simultaneously. This failure should be avoided in our concurrent transmission aware routing design. In the following of the paper, we address these challenges and our contribution is summarized as follows:

- To the best of our knowledge, we propose the first generic concurrent transmission aware design mShare that exploits the benefits of different physical-layer concurrent transmission techniques. This design optimizes performance by different metrics, including but not limited to (i) minimizing the number of transmissions, length of delays, and energy consumptions, and (ii) maximizing the throughput and delivery ratio.
- To leverage the benefits of concurrent transmission techniques, we investigate the problem of scheduling concurrent transmissions among co-owned receivers to optimize the network performance. We prove that this problem is NP-hard and propose a lightweight schedule design with performance bound $1 - \frac{1}{e}$.
- To illustrate the versatility of this design, we test it in three different situations to optimize different performance metrics: (i) in unicast to maximize the throughput, (ii) in opportunistic routing to minimize the number of transmissions, and (iii) in data collection to minimize the delivery delay.
- The performance of our design is evaluated with testbed experiments using USRP and large-scale network simulations. Our results show that compared to conventional designs, our design achieves an impressive improvement in performance: (i) a 277% improvement in throughput in unicast, (ii) a 78%

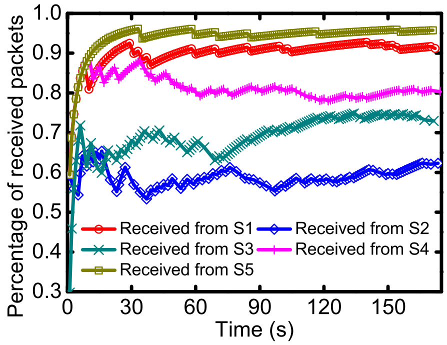


Fig. 1. Packet reception under collisions: the percentage of received packets from each sender.

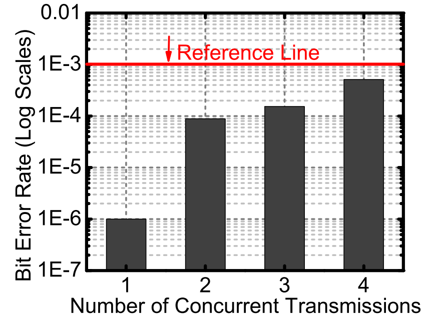


Fig. 2. The capability of decoding concurrent transmissions.

reduction in the number of transmissions in opportunistic routing, and (iii) a 70% reduction in latency in data collection in bursty networks.

In the rest of the paper, Section II provides some background information about the problem addressed in this study. Section III presents the motivation behind our generic design, which is presented in Section IV. Section V describes how we realize our design in existing protocols. Sections VI and VII evaluate the design with testbed experiments and simulations. Section VIII reviews related work. Finally, Section IX concludes the paper.

II. Preliminaries

In this section, we first discuss the collision issue in wireless communication and then demonstrate how concurrent transmission techniques resolve this issue as well as their concurrent decoding capability.

A. Collisions in Wireless Networks

With the increasing number of wireless devices and scarce spectrum resources, the collision issue is becoming more and more severe. A collision occurs when multiple data packets arrive at a receiver simultaneously. The risk of collisions is aggravated when traffic bursts increase the chance that multiple senders will transmit data packets to a receiver at the same time. Although contention-based CSMA protocols such as S-MAC [8] and B-MAC [9] typically use a random back-off mechanism to avoid retransmitting packets to a busy channel, in wireless networks with bursty traffic this mechanism may result in high latency and energy costs, since packets may collide at the receiver again when they are retransmitted [10].

We perform a simple experiment that demonstrates how the random back-off mechanism behaves in a bursty network, using the standard CSMA in 802.15.4 on a testbed in which five MICAz nodes send 50-byte data packets to a sink node. In the experiment, the transmission power is set at 0dBm. The senders broadcast one packet in every $200ms$. Figure 1 presents the percentage of received packets from each sender. Here, the percentage of received packets is defined as the number of received packets over the number of total transmitted packets. As the figure shows, the percentage of received packets for some nodes is very low. This is because many packets are lost or delayed by the random back-off strategy when the collision occurs.

B. Concurrent Transmission Techniques

To resolve the collision issue in concurrent transmissions, a series of such concurrent decoding techniques as successive interference cancellation [1], [2], constructive interference [3]–[6], and mZig [7] have been proposed. The interference cancellation technique resolves collisions by assigning distinct powers or pre-coded signatures. The constructive interference technique requires chip-level synchronization [4], [11] and is able to decode multiple synchronized transmissions of a same packet. Different from the constructive interference technique, mZig decomposes multi-packet collisions without synchronization. Although the decomposition modules in these designs are quite different, they all achieve the decoding capability of a certain number of packets' collision.

To illustrate the capability of decoding concurrent transmissions, we implement the most recent concurrent transmission technique – mZig [7] on USRP, as described in more detail in Section VI. Figure 2 shows the bit error rate (BER) of mZig in logarithmic scales with a varying number of concurrent transmissions. Generally, a packet can be successfully decoded when the BER is below 10^{-3}. From the figure, we can see that the BER of mZig is less than the reference line 10^{-3} when the number of concurrent transmissions equals four. This experiment demonstrates that the mZig technique can successfully decode four concurrent transmissions in this specific experimental environment.

III. Motivation

This section discusses the unexploited benefits of concurrent transmission techniques and their potential for improving performance with concurrent transmission aware design.

A. Unexploited Benefits

The concurrent transmission techniques in the physical layer efficiently resolve the collision issue when multiple senders transmit packets to a receiver simultaneously. Upper layer network designs, however, may not fully exploit the benefits of these concurrent transmission techniques. Current network designs are built upon the traditional physical layer and are

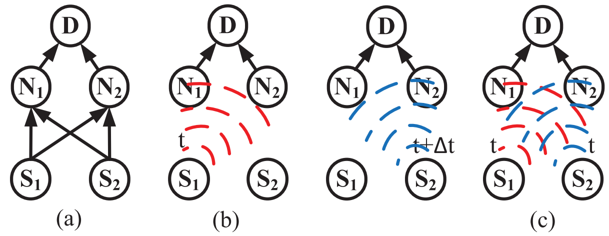


Fig. 3. Motivation examples.

unaware of concurrent transmission benefits. When senders deliver packets to their one-hop neighbors, they either transmit the packets to exclusive receivers (although lower layers can decode the packets when the packets are sent to co-owned one-hop receivers) or transmit them in series to avoid potential collisions. For example, in the network topology in Figure 3(a), the two senders S_1 and S_2 need to deliver packets through their one-hop neighbors N_1 and N_2 to the destination node D. In the traditional routing strategy shown in Figure 3(b), the senders S_1 and S_2 transmit packets to the receiver in time t and $t+\triangle t$ to avoid collisions. This example shows that existing network designs are unaware of the lower layers' concurrent decoding capability and do not exploit the benefits of concurrent transmission techniques.

B. Potential Improved Performance

With the support of the concurrent transmission technique, the senders may use a co-owned receiver in parallel when they know that receiver can successfully decode their transmissions. In the example shown in Figure 3(c), when upper layers are aware of the concurrent decoding capability from the physical layer, both senders can concurrently use their best receiver (no matter co-owned or not) for routing purposes. Furthermore, when network-layer routing design is aware that lower layers are able to decode concurrent transmissions, both sender S_1 and S_2 can maintain the co-owned receiver set $\{N_1, N_2\}$. If one of the receivers fails to deliver a packet, it is possible that the others can take over the routing task, which offers a great opportunity to improve network performance. In Figure 3(c), for example, if the transmission from S_1 to N_1 fails, it is possible that N_2 can successfully receive the packet. On the contrary, when network layer is unaware of low-layer's concurrent decoding capability, senders will select exclusive forwarder set to avoid collisions. Therefore, only one receiver is selected for each sender in its forwarder set in the example in Figure 3. The network performance is degraded when the size of forwarder set is reduced from two to one.

From the above two examples, we can see that (i) concurrent transmission techniques create a large number of potential available receivers (i.e., routing choices), and (ii) the awareness of concurrent transmission allows senders to utilize the co-owned receiver(s) in parallel in routing. In the following section, we provide a concurrent transmission aware design for best exploiting the benefits offered by concurrent transmission techniques.

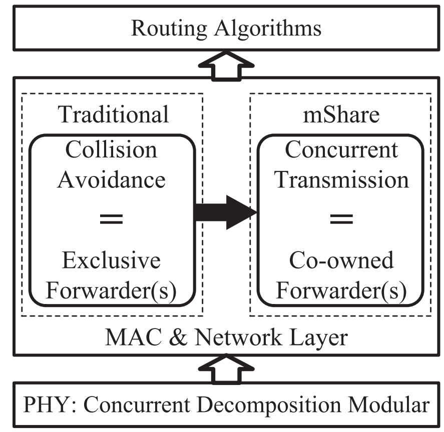


Fig. 4. The system overview of mShare.

IV. Main Design

This section introduces our design, which we have named mShare. mShare is a concurrent transmission aware routing design that aims to fully exploit the benefits of the concurrent transmission techniques. Figure 4 provides a system overview of mShare. From this figure, we can see that mShare uses the decomposing modular of the concurrent transmission technique in the physical layer for the purpose of decoding concurrent transmissions. With the support of the developed physical layer, this paper focuses on exploiting the benefit of the novel concurrent decoding techniques from the networking perspective. As shown in Figure 4, unlike the traditional network layer designs that assign exclusive forwarders/receivers in routing to avoid collisions, mShare schedules co-owned forwarders/receivers for concurrent senders. In the following, we first introduce the assumptions related to mShare in §IV-A. We then propose the concurrent transmission scheduling problem in §IV-B, followed by our proposed solutions in §IV-C.

A. Assumptions

Suppose the physical layer provides the capability of decoding concurrent transmissions. The following assumptions are made for our mShare design:

- The concurrent transmission techniques may fail when too many packets overlap [6], [7]. We thus assume that the concurrent decoding capability is limited by m, where m is decided by the type of concurrent transmission technique and the working environment.
- The MAC layer is modified to support concurrent transmissions. To avoid the concurrent transmissions exceeding the concurrent decoding capability – m and thus causing decoding failures, the MAC layer is responsible to schedule at most m senders in one transmission window [3], [7], [12]. For example, in duty-cycled networks, this is achieved by scheduling at most m senders to transmit data in the active slot of a receiver.

Note that m varies in dynamic environment. An online scheduler built-in the MAC layer is adopted to estimate m based on the bit error rate (BER) of decoded packets [7]. m is initialized with the theoretical value recommended by the concurrent transmission technique. The decoding capability m

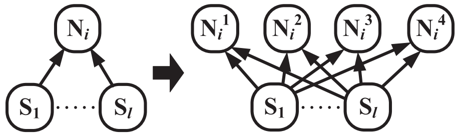


Fig. 5. An example of neighbor-expanded network.

is increased by one when BER $< 0.1\xi$. Here, $\xi = 10^{-3}$ which is a common setting in wireless communication to determine successful packet receptions. When BER $> \xi$, m is decreased by one, and m stays the same when $0.1\xi \leq$ BER $\leq \xi$.

B. Concurrent Transmissions Scheduling Problem

Based on the above lower layer settings, we introduce how to schedule concurrent transmissions to best exploit the benefits of the concurrent decomposition modular in physical layer.

1) Neighbor-Expanded Network Model: We first propose a neighbor-expanded network model which is tailored for lower layers that support m successful concurrent transmissions. The process represented by this model is as follows: (i) for each sender's one-hop neighbor N_i, we generate m virtual nodes $\{N_i^1, N_i^2, \ldots, N_i^m\}$, where m is the concurrent decoding capability of the concurrent transmission techniques, and (ii) the new generated nodes inherit all the information of the node N_i such as the neighbor relationship and link quality. Figure 5 provides an example of generating this neighbor-expanded network from the original network. The left part of Figure 5 shows the original network topology where l senders are connected with the neighbor representative N_i. The node N_i is expanded to four exact the same virtual nodes, i.e., N_i^1, N_i^2, N_i^3, and N_i^4, in the neighbor-expanded network. In this example, $m = 4$.

2) Problem Formulation: This neighbor-expanded network provides the maximum available receiver resources for the forwarding task in network routing. In the neighbor-expanded network, let the concurrent sender set who has the data delivery task be $S = \{S_1, S_2, \ldots, S_l\}$ and the one-hop neighbor set of sender S_i be N_{S_i}. To construct a generic model, we next formulate the network routing problem with a receiver set instead of with a single receiver. The nodes in the receiver set are responsible for receiving and forwarding a packet if needed. Let each sender S_i maintains a receiver set R_{S_i}, where $|R_{S_i}| \geq 1$. Then, the optimal goal $optimize \sum_{i=1}^{l} E(R_{S_i})$, i.e., optimizing the network performance of the concurrent senders, becomes:

$$
\begin{aligned}
&optimize \ \sum_{i=1}^{l} E(R_{S_i}) \\
&s.t. \ R_{S_i} \subseteq N_{S_i}, \quad \forall S_i \in S \\
&R_i^j \notin R_{S_i}, \quad \forall R_i^k \in R_{S_i}, \ j \neq k, \ \forall S_i \in S \\
&R_{S_i} \cap R_{S_j} = \varnothing, \quad i \neq j \\
&\cup_{i=1}^{l} R_{S_i} = \cup_{i=1}^{l} N_{S_i}, \quad |R_{S_i}| > 1
\end{aligned}
$$

Here the optimization target $E(R_{S_i})$ could be any network performance metrics, and the optimization goal includes (i) minimizing transmissions, delays, and energy consumption, and (ii) maximizing throughput and delivery ratio. The first constraint guarantees that the selected receivers are the sender's one-hop neighbors. In the neighbor-expanded network, it is meaningless for one sender to obtain multiple virtual nodes generated from the same physical node, a case that is avoided by the second constraint. The third constraint guarantees that no virtual nodes in the neighbor-expanded network are owned by more than two senders. Otherwise, more than m concurrent senders will be assigned to one physical node in the original network, which is not supported by the lower layers. This last constraint allows the sender set to utilize all the available nodes, thus achieving optimal network performance when the senders maintain multiple receivers as forwarding nodes. We find that solving this problem with the four constrains is NP-hard.

Lemma 1: Given a sender set $S = \{S_1, S_2, \ldots, S_l\}$ and their one-hop neighbor set $\cup_{i=1}^{l} N_{S_i}$, it is NP-hard to assign each sender S_i a receiver set R_{S_i} such that the sum of the network performance metric of each sender, i.e., $\sum_{i=1}^{l} E(R_{S_i})$ is optimized where the sender's receiver set satisfies $R_{S_i} \cap R_{S_j} = \varnothing, i \neq j$ and $\cup_{i=1}^{l} R_{S_i} = \cup_{i=1}^{l} N_{S_i}$.

The proof of Lemma 1 is via a reduction of this problem from the weighted exact cover problem. Since this problem is NP-hard, we propose a practical greedy algorithm with performance bound in the following section.

C. Schedule Algorithm

For the sake of clarity, we present the schedule algorithm in a centralized manner. However, our design is developed distributively with $O(n)$ neighbor information exchange.

In the following, we first examine a basic version of the concurrent transmissions scheduling problem in which the size of the receiver set equals one ($|R_{S_i}| = 1$) and then extend the design for a receiver set with an arbitrary size ($|R_{S_i}| > 1$).

1) Basic Case: When each sender is allowed to maintain one receiver to receive and forward its packet, the concurrent transmissions scheduling problem becomes selecting l receivers from the neighbor set $\cup_{i=1}^{l} N_{S_i}$ in the neighbor-expanded network for the concurrent sender set $S = \{S_1, S_2, \ldots, S_l\}$ to optimize network performance. This problem is a generalization of the linear assignment problem, which assigns a number of agents to a smaller number of tasks with one agent to each task, optimizing the benefit for the agents. This assignment problem has optimal solutions that can be achieved by the rectangular assignment algorithm [13] in polynomial time.

2) General Case: When a sender maintains a receiver set for routing purposes, the receivers have different priorities for serving the sender. The receiver with a lower priority will serve the sender only when all the receivers with higher priorities fail. This mechanism is achieved by setting a forwarding timer that indicates the time for starting to send a packet. The receiver with a higher priority will maintain a timer far enough ahead to send out a packet. In the routing design with multiple receivers, a better node (e.g., a node with better link quality) will serve the sender with a higher priority. The sender will use the marginal links only when the good links fail. Based on this observation, we pose the following proposition:

Proposition 2: Given a receiver set, the difference in the incremental network performance that a single receiver makes when added to a receiver set decreases as the size of the receiver set increases.

From the above proposition, we find that the property of the optimization function, i.e., network performance with the receiver set $\sum_{i=1}^{l} E(R_{S_i})$, fits the property of the submodular function. [1] Submodular functions have a natural diminishing returns property that makes them suitable for approximation algorithms. In the following, we propose the approximation algorithm used in our design. The key idea of the algorithm is to continuously select the best receiver from the remaining available neighbors for each sender to maximize incremental network performance, which is achieved by continuously executing the rectangular assignment algorithm with the remaining neighbor set.

Algorithm 1 Scheduling Concurrent Transmissions

Require: S, $N = \cup_{i=1}^{l} N_{S_i}$, and $E(R_i)$;

1: Initialize the assigned receiver set: $R_{S_i} = \varnothing$;
2: **while** $N \neq \varnothing$ **do**
3: Apply the rectangular assignment algorithm with S, N, and the weight $E(R_i)$, for each $S_i \in S$, get the new assigned receiver r_{S_i};
4: $R_{S_i} = R_{S_i} + r_{S_i}$
5: $N = N - \cup R_{S_i}$;
6: $\forall N_i \in N$, updates $E(R_i)$ given R_{S_i};
7: **end while**
8: **return** $\sum_{j=1}^{l} E(R_{S_i})$

The detailed pseudo-code is shown in Algorithm 1. Line 1 initializes the assigned receiver set R_{S_i} with an empty set for each sender S_i. Line 3 calls the rectangular assignment algorithm. Given the input of the sender set $S = \{S_1, S_2, \ldots, S_l\}$, the one-hop neighbor set $N = \cup_{i=1}^{l} N_{S_i}$ and the weight $E(R_i)$ (i.e., network performance metrics such as throughput, the number of transmissions and delays), the rectangular assignment algorithm makes an optimal schedule of optimizing the sum of all the senders' network performance metrics by selecting one receiver for each sender. Line 4 adds the selected receiver to each sender S_i's receiver set R_{S_i}. Line 5 updates the set of available receiver candidates by removing the selected receivers. Given the selected receiver set R_{S_i}, the rest of the available neighbors' performance metric to each sender is updated (Line 6). In this metric updating, to satisfy the first two constraints in the problem formulation, the incremental network performance of receiver candidate R_i to sender S_i, i.e., $E(R_i)$, is set to zero, if (i) R_i is not S_i's one-hop neighbor or (ii) one of the same physical neighbor's virtual nodes is already assigned to sender S_i. Finally, the algorithm ends when there are no available receiver candidates in the neighbor set N (Line 2&7).

Lemma 3: The performance bound of the proposed scheduling algorithm is $1 - \frac{1}{e}$.

The proof of Lemma 3 is based on the construction of the submodular function with the objective function $\sum_{i=1}^{l} E(R_{S_i})$. The details of the proof can be found in the appendix.

In the following, we develop the above solution to an efficient and scalable distributed design that relies on local information and operations. Each sender exchanges its one-hop neighbor information with its nearby senders, e.g., one-hop senders, which is used as the input of the distributed algorithm. The distributed algorithm can obtain better performance when it has more senders' information, but this also causes more communication overhead. When each sender executes the scheduling algorithm locally, it cannot guarantee that one virtual node is assigned to only one sender. When one virtual node is assigned to multiple senders, the algorithm compares the incremental network performance of the virtual node to the senders and assigns the virtual node to the sender with the maximum incremental network performance.

With the scheduling algorithm, every receiver is assigned to at most m senders. To support concurrent transmissions, each receiver transmits a 'mShare start' message to its assigned senders in turn after a fixed time window. The senders who receive the 'mShare start' message disable their CSMA/CA and transmit concurrently in the notified time window. CSMA/CA is re-enabled in these senders after the window time is expired. The receiver stops to receive packets and sends an ACK to its assigned senders when the time window is expired.

•**Algorithm Complexity and Communication Overhead:** The scheduling algorithm (i.e., Algorithm 1) runs $\lceil \frac{k}{l} \rceil$ rounds of the rectangular assignment algorithm, where k is the size of the neighbor set and l is the size of the sender set. In our implementation, we adopt the SKAP algorithm [13], which provides an exact solution for the rectangular assignment problem. The time complexity of the SKAP algorithm is $O(lk \log k)$. The time complexity of Algorithm 1 is thus $O(k^2 \log k)$. The time complexity is further reduced when approximation algorithms for the rectangular assignment problem are adopted. The communication overhead of the distributed algorithm is low since it uses only one-hop senders' information.

•**Multi-hop Performance Guarantee:** The above design provides an efficient schedule of multiple concurrent senders with their neighbors to optimize the *one-hop* network performance. To optimize the end-to-end network performance in multi-hop networks, we adopt the existing schedule algorithms tailored for multiple source-destination communication sessions with forwarder sets [15]–[17] and replace their one-hop metrics (e.g., delays or the expected number of transmissions) to our optimized one-hop network performance metrics.

V. Design Realizations

To illustrate the versatility of the proposed generic design, we describe how we realize this design in existing routing

[1] In mathematics, a submodular function (also known as a submodular set function) is a function whose value, informally, has the property that the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases [14].

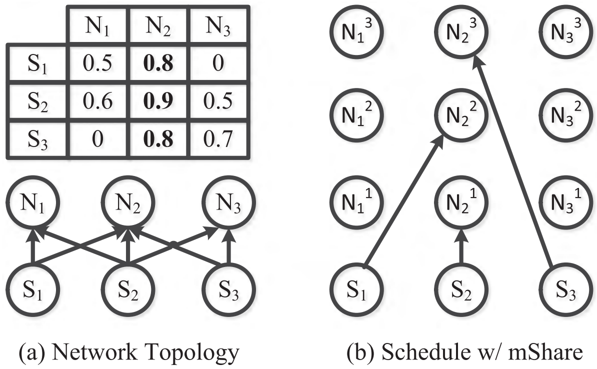


Fig. 6. An example in unicast realization.

protocols, including unicast, opportunistic routing, and data collection.

A. Realizations in Unicast

•**Benefit of mShare in Unicast:** Take the network topology in Figure 6(a) as an example. The table above the topology shows the link quality from S_i to N_j. Traditional unicast protocols have two strategies. The first strategy is that each sender selects its best receiver, i.e., N_2, and transmits the packets in series. The other strategy is that senders S_1 and S_3 transmit packets to their exclusive receivers N_1 and N_3 in parallel while the sender S_2 transmits in different time windows. With mShare, as shown in Figure 6(b), all three senders are able to transmit packets to the best receiver N_2 in parallel when the concurrent decoding capability is equal to or greater than three.

•**Realize mShare with the Throughput Metric:** In the mShare realization in unicast, we first generate the neighbor-expanded network from the original network topology (as shown in Figure 6(b)) and then assign receivers to concurrent senders with the rectangular assignment algorithm since the size of the receiver set equals one. The input of the rectangular assignment algorithm is S, $N = \cup_{i=1}^{l} N_{S_i}$, and $E(R_i)$. The input S and N is straightforward, which is the set of senders and the union of senders' one-hop neighbors. This information can be obtained through the traditional neighbor discovery process. The input information of $E(R_i)$, which is the throughput metric with the receiver R_{S_i}, needs a bit more explanation. To obtain the throughput metric, we need to measure the total received packet size within a time duration. This process is combined with existing link quality measurement designs that periodically send out probe packets to estimate the link status. After obtaining all the input information, mShare is able to assign each sender a receiver and allows the senders to transmit packets to the receivers in parallel to optimize their throughput.

B. Realizations in Opportunistic Routing

In this subsection, we introduce how to integrate our design into opportunistic routing. In opportunistic routing [18]–[20], a sender maintains a set of receivers. When one receiver fails to receive a packet, the rest of the receivers may receive the packet and take over the forwarding task.

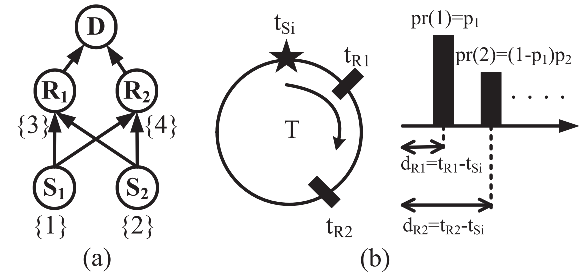


Fig. 7. Illustration examples: (a) A data collection example in low-duty-cycle networks; (b) Computation of expected delay.

•**Benefit of mShare in Opportunistic Routing:** Opportunistic routing, however, may lead to performance degradation when there are multiple concurrent senders. Since the sender in opportunistic routing occupies multiple receivers, it is likely that the concurrent senders can only maintain small exclusive receiver sets (e.g., one receiver), or even worse, can not maintain exclusive receiver sets at all and have to transmit packets to their receiver sets in series. Our mShare design brings massive potential receiver resources which makes it possible for concurrent senders to deliver the packets to their receiver sets in parallel.

•**Realize mShare with the Transmission Metric:** To make mShare consistent with the classic opportunistic routing design ExOR [18], we use the metric – the expected number of transmissions in our integration. Given an arbitrary sender $S_i \in S$, let its receiver set be $R_{S_i} = \{R_1, R_2, R_3, \ldots, R_M\}$. We assume that the link quality from S_i to R_i be p_i. Then we set the expected number of transmissions for the sender S_i to successfully deliver a packet to the receiver set R_{S_i} at

$$E(R_{S_i}) = \frac{1}{1 - \prod_{i=1}^{M}(1 - p_i)}.$$

After obtaining the transmission metric with an arbitrary receiver set, we have the incremental network performance, i.e., the reduced number of transmissions, for an arbitrary receiver R_i given the selected receiver set R_{S_i}: $E(R_i) = E(R_{S_i}) - E(R_{S_i} \cup R_i)$. We thus obtain the input, i.e., $E(R_i)$, for Algorithm 1. The schedule algorithm returns co-owned receivers for opportunistic routing to minimize the total number of transmissions of the concurrent senders.

C. Realizations in Data Collection

In this realization, we integrate our design into the data collection protocols in low-duty-cycle networks. In low-duty-cycle networks, a node has two possible states: active and dormant. In the active state, the node can sense, transmit, and receive packets. In the dormant state, the node turns off all function modules except a timer to wake itself up. In Figure 7(a), the node S_1, S_2, R_1, and R_2's time slots in the active state are $\{1\}$, $\{2\}$, $\{3\}$, and $\{4\}$ separately.

The node switches from the dormant state to the active state under the following two scenarios. In the first, this node is scheduled to actively receive packets. For example, the node S_1 switches from the dormant state to the active state in time slots $\{1\}$ to receive a packet. In the second, this node needs to send out some packets, which happens when its neighbor nodes are in the active state and ready to receive its packets. For example, the node S_1 and S_2 switch from the dormant state to the active state in time slots $\{3\}$ and $\{4\}$ in which the receiver R_1 and R_2 are in the active state.

•**Benefit of mShare in Data Collection:** Our mShare design has the ability to significantly reduce data collection delays. Take the network topology in Figure 7(a) for example. In traditional data collection design, senders S_1 and S_2 exclusively use the receivers R_1 and R_2 to avoid collisions. If the link from the sender to the receiver fails, the sender needs to wait a whole working period for the receiver to wake up again, which leads to a huge delay. With the help of mShare, the senders S_1 and S_2 now can use co-owned receivers, i.e., R_1 and R_2, simultaneously. If one of the receivers fails, the sender can use the next wake-up receiver, which significantly reduces delivery latency.

•**Realize mShare with the Delay Metric:** To realize our mShare design in data collection protocols, we use a delay metric to shorten the delivery latency. Given a sender S_i, we assume that its receiver set with size M is $\{R_1, R_2, R_3, \ldots, R_M\}$. Let the wake-up time slots of the nodes in the receiver set be $\{t_{R_1}, t_{R_2}, t_{R_3}, \ldots, t_{R_M}\}$. Without loss of generality, we assume that $t_{R_1} < t_{R_2} < \ldots < t_{R_M}$. In the forwarding process with a receiver set, the receivers deliver a data packet subsequently based on the wake-up time slots $\{t_{R_1}, t_{R_2}, t_{R_3}, \ldots, t_{R_M}\}$ until the packet is successfully delivered. In other words, if receiver R_i fails to receive a data packet at time slot t_{R_i}, the sender switches to receiver R_{i+1} which wakes up at time slot $t_{R_{i+1}}$ for the packet delivery.

We use Figure 7(b) to illustrate the computation of expected delays. If sender S_i receives the data packet at its wake-up time slot t_{S_i}, the delay for the data packet to be successfully delivered at receiver R_i is $d_{R_i} = t_{R_i} - t_{S_i}$. In data collection with a receiver set, sender S_i delivers packets to receiver R_i when all the previous wake-up receivers fail. The probability that the packet delivery fails at the first $n-1$ times while is successful at the n-th attempt is $Pr(n) = \prod_{i=1}^{n-1}(1-p_i)p_n$, where p_i is the link quality from S_i to R_i. Then, the expected delay of a packet from sender S_i to the receiver set $R_s = \{R_1, R_2, R_3, \ldots, R_M\}$ is given by

$$E(R_{S_i}) = \sum_{i=1}^{n} d_{R_i} Pr(n).$$

Up to now, we have deduced $E(R_{S_i})$ – the expected delay for one sender to successfully deliver one packet to an arbitrary receiver set R_{S_i}. We then apply Algorithm 1 to help concurrent senders obtain their co-owned receivers to minimize the total delay in their delivery tasks.

VI. Testbed Implementation

To understand the performance of mShare in practical settings, we conduct experiments on our USRP testbed (Figure 8) to evaluate the performance of mShare in unicast, opportunistic routing, and data collection. We compare mShare to ETX [21] in the unicast realization and to ExOR [18] in the opportunistic routing realization. In the data collection realization, we compare mShare to a data collection protocol that also utilizes multiple receivers for packet delivery, DSF [15], which is one of the most cited routing designs using a receiver set in wireless sensor networks. In all the three realizations, we also compare our concurrent transmission aware design with ETX, ExOR and DSF under the support of physical-layer concurrent transmission technique mZig [7], i.e., ETX+mZig, ExOR+mZig and DSF+mZig.

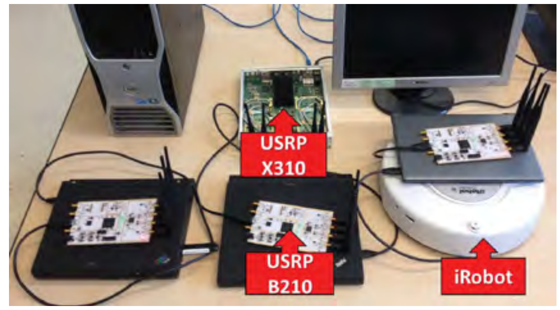


Fig. 8. USRP testbed.

A. Experiment Settings

This experiment is built upon two USRP X310s and four USRP B210s in a 7.5m $\times$ 6.8m office. The hardware devices are shown in Figure 8. For ETX+mZig, ExOR+mZig, DSF+mZig, and mShare, the USRP X310s are developed with the concurrent decoding module – mZig [7] and used as the receivers. The senders are developed in USRP B210s, which require no change in physical layer. The CSMA/CA is disabled in senders. For ETX, ExOR, and DSF, we adopt the traditional decoding module and make no change in either senders or receivers.

In this experiment, the USRP B210s and USRP X310s are linked to laptops. Four iRobots carry the laptops and USRP B210s and randomly move in the office with a speed less than 0.4m/s. We set the sampling rate as 32Mbps, which is 16x the chip rate. We set the transmission gain at 70 in Gnu-Radio, so the transmission power is 0dB, which is the default transmission power in 802.15.4. With such a power setting, all the receivers are within the transmission range of the senders. To avoid cross technology interference, we use channel 26 in the experiment, which does not overlap the Wi-Fi channel. Under such a setting, the four USRP B210s transmit 1000 packets to the two USRP X310s. The payload length of the packet is 1000 bits, the data rate is 250kbps, and the duty cycle is 1% with 8ms unit time. We then run ETX, ETX+mZig, ExOR, ExOR+mZig, DSF, DSF+mZig and mShare using the packet reception records.

B. Evaluation Results

This section first reports the successful rate of packet reception at receivers under the concurrent transmission scenario. Then we show the performance of mShare in unicast, opportunistic routing, and data collection.

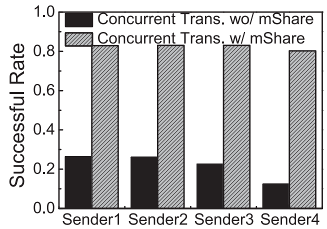


Fig. 9. Reception successful rate.

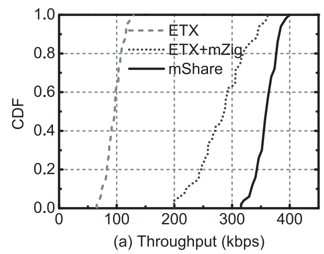


Fig. 10. Performance in unicast.

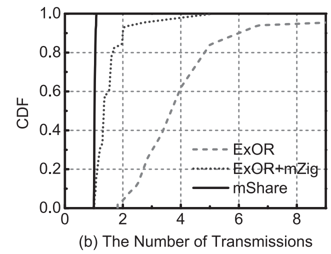


Fig. 11. mShare performance in opportunistic routing.

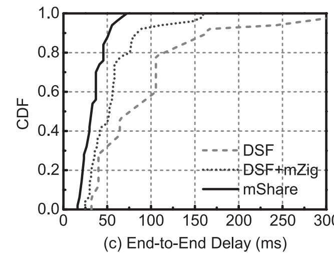


Fig. 12. mShare performance in data collection.

1) Reception Successful Rate: Figure 9 shows the successful rate of packet receptions at receivers (i.e., USRP X310s) when four senders (i.e., USRP B210s) transmit packets concurrently. From the figure, we can see that the successful rate with mShare is about 82.3% on average due to its capability of successfully decode concurrent transmissions. The reception successful rate drops to 21.8% when mShare is not adopted. It's because that collisions cause severe reception failures.

2) Results in Unicast Realization: The CDF of throughput with ETX, ETX+mZig and mShare is shown in Figure 10. As we can see, the received throughput from the four senders (USRP B210s) with ETX ranges from 65kbps to 129kbps while that with mShare ranges from 316kbps to 397kbps. Compared to ETX, mShare improves throughput by 277% since mShare fully exploits the benefits of concurrent transmission techniques and the mShare receiver (USRP X310) can successfully receive the packets from the four concurrent senders. The average throughput from the four senders with ETX+mZig and mShare is 284kbps and 361kbps. mShare improves the throughput of ETX+mZig by 21%. That is because that (i) mShare selects a forwarder with the best link quality although this forwarder is co-owned by other concurrent senders as long as the forwarder is within the scope of concurrent decoding capability, and (ii) ETX+mZig only selects a non-occupied forwarder with the best link quality because of the unawareness of lower layers' concurrent decoding capability.

3) Results in Opportunistic Routing Realization: The CDF of the number of transmissions with ExOR, ExOR+mZig and mShare is shown in Figure 11. As we can see, *almost* all the number of transmissions with mShare is one. The reason of such an observation is as follows. First, the mShare receiver set (i.e., the two USRP X310s) can successfully decode the packets from the four concurrent senders. Second, the senders will not retransmit a packet when any node in the receiver set receives the packet. mShare reduces ExOR about 78% of the number of transmissions. That is because that under the scenario with bursty traffic, it is difficult for ExOR to obtain enough exclusive receivers for packet delivery and the collisions among transmissions (from the four concurrent senders) greatly increase ExOR's number of transmissions. From Figure 11, we also find that ExOR+mZig reduces ExOR about 67% of transmissions. That is because that mZig resolves the collisions among the four concurrent senders and avoids retransmissions caused by collisions. Compared with mShare, ExOR+mZig has 35% more transmissions because that the unawareness of concurrent decoding capability leads to a smaller size of forwarder set.

4) Results in Data Collection Realization: The CDF of the end-to-end delay with DSF, DSF+mZig and mShare is shown in Figure 12. As we can see, the average delay with DSF, DSF+mZig and mShare is 104ms, 57ms and 35ms respectively. mShare reduces DSF about 70% of the delay. There are two reasons for this improvement. First, mShare creates multiple receivers and helps the dynamic switching get rid of degrading to the traditional one-to-one forwarding scheme when there are not enough receivers for concurrent senders during data collection. Second, mShare resolves the collision issue and thus reduces packet loss and random back-off delay.

VII. Large-Scale Network Simulation

We then simulates the performance of mShare in large-scale multi-hop networks under different network settings that are difficult to control in testbed. In our simulation of the data collection realization, excepted the delay metric used in the testbed experiment, we compare mShare with DSF and DSF+mZig in terms of their (i) delivery ratio (i.e., the number of packets received by the sink over the number of packets sent by the senders) and (ii) the number of transmissions (i.e., the average number of transmissions of one packet from

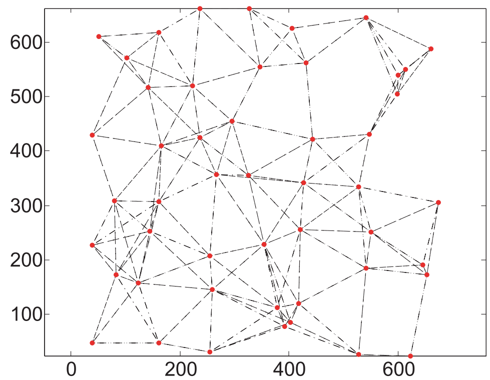


Fig. 13. The default network topology ($700m \times 700m$) in simulation.

the sender to the sink over the sender's delivery ratio). In our simulation of the opportunistic routing realization, we compare mShare with ExOR and ExOR+mZig under different network sizes and link qualities.

A. Simulation Setup

We randomly generate network topologies ranging in network size from 25 to 100 nodes using MATLAB. The default network size is 49 in a 700m $\times$ 700m area whose network topology is shown in Figure 13. The default communication range is 160m. We generate the packet reception bitmaps for each node's neighbors with the latest *link correlation model* [22]. In the simulation, the source nodes deliver 1000 packets to the destination(s). The maximum number of retransmissions in each hop is six, and the concurrent decoding capability of mShare is set to four. The default average link quality is set at 0.6. In the data collection experiment, the default duty cycle is 1%. We set the working cycle length at $T = 0.5s$ and the active slot length at 5ms, which completes a round-trip of the data and ACK transmissions. The simulation results have been averaged over 10 rounds, and the related standard deviations are provided as error bars.

B. Results in Data Collection Realization

1) Different Duty Cycles: We evaluate the performance of data collection designs – DSF, DSF+mZig and mShare under different duty cycles, which has a direct impact on the number of wake-up slots (i.e., available receivers). Figure 14(a) shows the end-to-end delays of DSF, DSF+mZig and mShare when the duty cycle varies from 1% to 4%. From the figure, we see that the delay with mShare is 71% lower than with DSF when the duty cycle is 1%. The delays of DSF, DSF+mZig and mShare are reduced when the duty cycle is increased. This is because (i) the increased active slots can be seen as increased available receivers, which improve the performance of dynamic forwarding, and (ii) the time interval between consecutive active slots decreases when the duty cycle increases.

Figure 14(b)shows the average number of transmissions from the senders to the sink node with DSF, DSF+mZig and mShare. With DSF, the number of transmissions is about 100% higher than that with mShare. In DSF, there are a large amount of retransmissions because of collisions, although its CSMA mechanism partially alleviates the collision issue.

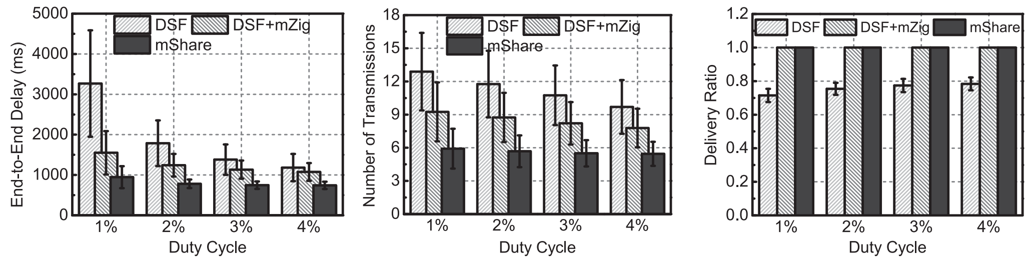
Figure 14(c) plots the delivery ratio of DSF, DSF+mZig and mShare when the duty cycle increases. On average, DSF achieves a delivery ratio of about 75.7%, while the delivery ratios of DSF+mZig and mShare are almost always 100% when the maximum number of retransmissions is six. The major reason for the packet loss in DSF is the collision in the data collection scenario where multiple senders transmit packets to the same receivers in the same active slot. Our design guarantees that there are no collisions among the concurrent transmissions. Besides, mShare provides multiple times of receivers for each sender, which almost always achieves a 100% delivery ratio.

2) Comparison With TDMA: We also compare the performance of our mShare design with DSF using the conventional TDMA mechanism, named DSF w/ TDMA. In DSF w/ TDMA, the available receivers (slots) are equally assigned to the senders, and the senders are not allowed to switch to the receivers assigned to other senders. Figure 15 reports the cumulative distribution of the end-to-end delay of DSF w/ CSMA, DSF w/ TDMA, and mShare. As it shows, the end-to-end delay of DSF with TDMA is 22.4% less than that of DSF with CSMA, which it accomplishes by restricting the sharing of receivers among senders and thus providing collision-free scheduling in the scenario of high traffic volume data collection. mShare provides efficient scheduling of the concurrent transmissions and thus reduces the delay in DSF w/ TDMA by 61.3% because mShare creates concurrent transmissions by scheduling multiple senders using the same receivers simultaneously. The concurrent decoding ability of mShare helps it multiply receiver resources more than the TDMA method does.

3) Different Network Sizes and Link Qualities: We evaluate the performance of our design in networks of different sizes, ranging from 25 to 100. In Figure 16(a), the end-to-end delay increases as the network's size increases, as expected. The mShare design works well in large-scale networks. Compared with DSF, mShare reduces delivery delays by about 70.5% on average. The end-to-end delays with DSF, DSF+mZig and mShare under different link qualities in Figure 16(b) show that (i) the delays with mShare range from 1346ms to 852ms, (ii) the delays with DSF+mZig range from 1346ms to 852ms, and (iii) the delays with DSF range from 2137ms to 1018ms when the average link quality varies from 0.4 to 0.8. Compared with DSF and DSF+mZig, the percentage of mShare's performance improvement ranges from 72.6% to 61.5% and from 37.1% to 16.2% respectively when the link quality changes from 0.4 to 0.8. From these results, we see that mShare significantly reduces delay, especially for those scenarios with poor link statuses, because of the dynamic switching among the multiplied available receivers.

C. Results in Opportunistic Routing Realization

We evaluate the performance of ExOR, ExOR+mZig and mShare under different network sizes and link qualities. Figure 17(a) shows that the mShare design works well in



(a) End-to-End Delay (b) The Number of Transmissions (c) Delivery Ratio

Fig. 14. The performance comparison in different duty cycles.

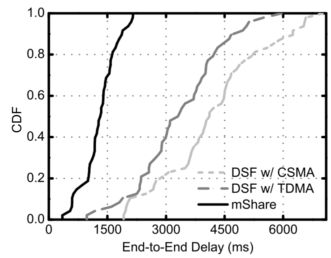


Fig. 15. Comparison with TDMA.

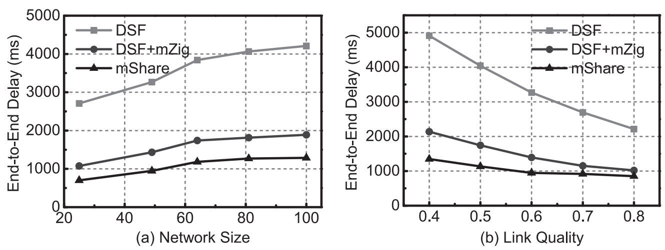


Fig. 16. The performance comparison under different (a) network sizes and (b) link qualities in data collection.

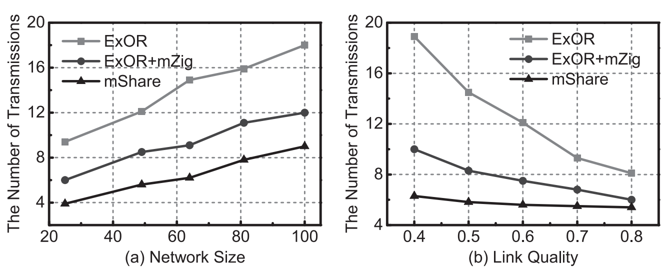


Fig. 17. The performance comparison under different (a) network sizes and (b) link qualities in opportunistic routing.

large-scale networks. Compared to ExOR and ExOR+mZig, mShare reduces the number of transmissions from sources to destinations by about 54% and 31% on average. The number of transmissions with ExOR, ExOR+mZig and mShare under different link qualities are shown in Figure 17(b), where we can see that the number of transmissions with mShare ranges from 6.3 to 5.4, while the number of transmissions with ExOR ranges from 18.8 to 8.2 when the average link quality ranges from 0.4 to 0.8. The percentage of performance improvement ranges from 66% to 34% when the link quality changes from 0.4 to 0.8. From these results, we see that mShare significantly reduces the number of transmissions, especially for those scenarios with poor link statuses, because of its utilization of multiple co-owned receivers.

VIII. Related Work

In this section, we review the works on collision solutions that are most related to our work, including research on collision avoidance and on collision resolution.

A. Collision Avoidance

The conventional ZigBee adopts CSMA to avoid collisions by setting a random back-off timer. This random back-off mechanism [8], [9] may cause high delays in low-duty-cycle networks, especially under traffic bursts, given the high chance that the retransmitted packets will also collide. Besides, CSMA may suffer the hidden terminal problem, which is alleviated by the RTS/CTS handshake mechanism [23] but incurs high overhead. TDMA, on the other hand, solves the hidden terminal problem with low overhead. Several researchers have proposed scheduling algorithms based on TDMA to avoid the collision issue. For example, Chipara *et al*. [24] propose a dynamic collision-free query scheduling technique for high data-rate sensor network applications, and Yu *et al*. [25] propose a collision-free scheduling algorithm for the data collection application in WSNs. While the TDMA mechanism is collision-free, it is not trivial to find an efficient time scheduling in a scalable fashion. Researchers have thus proposed hybrid schemes [26], [27] to obtain the strengths of CSMA and TDMA while eliminating their weaknesses. In addition, researchers also propose multi-channel techniques to increase transmission parallelism and avoid collisions [28].

B. Collision Resolution

Recently, researchers have advocated solving the collision issue through collision resolution and thus realizing multi-packet reception. Halperin *et al*. [1] and Sen *et al*. [2] propose using interference cancellation to resolve collisions by assigning distinct powers or pre-coded signatures. Researchers have also utilized constructive interference [3]–[6] to receive multiple synchronized transmissions of the same packet. The full duplex technique [29] resolves collisions by subtracting the known packets from the collided one. Recently, Kong and Liu [7] propose mZig, which decodes a m-packet collision with only one collided packet, but the mZig technique cannot successfully decode the collided packet when there are too many concurrent transmissions. While these concurrent transmission techniques provide concurrent decoding modular in physical layer, this paper proposes a generic network layer design to better utilize the benefits of these techniques.

IX. Conclusion

In this work, we propose mShare, a generic concurrent transmission aware design that currently best exploits the benefits of physical-layer concurrent transmission techniques. We realize the mShare design in unicast, opportunistic routing, and data collection protocols. The performance of mShare is evaluated with large-scale network simulations and physical testbed experiments running on USRP. The evaluation results show that, compared to conventional designs, mShare improves throughput by 277% in unicast, lowers the number of transmissions in opportunistic routing by 78%, and reduces latency in the data collection realization by 70%.

Appendix
Proof of Lemma 3:

For an arbitrary sender S_i, let its receiver set be $R_{S_i} = \{R_1, R_2, R_3, \ldots\}$. We define a function $f(X) = |C - E(R_{S_i})|$, where C is the initial value of performance metric $E(R_{S_i})$ and is a constant number. Then $f(X)$ represents the improved network performance with the receiver set R_{S_i}. We apply Proposition 2 to obtain the following property of $f(X)$. For every receiver set $X, Y \subseteq R_{S_i}$ with $X \subseteq Y$ and every receiver $r \in R_{S_i} \setminus Y$, we have that $f(X) \le f(Y)$ and $f(X \cup \{r\}) - f(X) \ge f(Y \cup \{r\}) - f(Y)$. Since the class of submodular functions is closed under non-negative linear combinations, the function $g(R_{S_i}) = |C - \sum_{i=1}^{l} E(R_{S_i})|$ is a monotone submodular function.

In Algorithm 1, we initialize the solution with the the empty set. The algorithm treats the senders equally and takes them as a super sender. In each round, we add a super receiver r, i.e., one receiver for each sender, to the receiver set R_i with the rule $r = \arg\min_r E(R_{i-1} \cup r)$, which is the equivalent of $r = \arg\max_r g(R_{i-1} \cup r)$. Let the optimal solution be $T = r_1, \ldots, r_k$ and $\delta_i = g(R_i) - g(R_{i-1})$, we have

$$\begin{aligned} g(T) &\le g(R_i \cup T) = g(R_i \cup T) - g(R_i) + g(R_i) \\ &\le \sum_{j=1}^{k} [\delta_{i+1}] + g(R_i) = g(R_i) + k\delta_{i+1} \end{aligned} \tag{1}$$

Based on Eq.(1), we have

$$\begin{aligned} g(R_i) &= g(R_{i-1}) + \delta_i \ge g(R_{i-1}) + \frac{1}{k}[g(T) - g(R_{i-1})] \\ &= (1 - \frac{1}{k})g(R_{i-1}) + \frac{1}{k}g(T) = [1 - (1 - \frac{1}{k})^i]g(T) \\ &\ge [1 - \frac{1}{e}]g(T). \end{aligned} \tag{2}$$

The performance bound of Algorithm 1 is thus $1 - \frac{1}{e}$.

References

[1] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *Proc. ACM MobiCom*, 2008, pp. 339–350.

[2] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive interference cancellation: Carving out MAC layer opportunities," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 346–357, Feb. 2013.

[3] P. Dutta, S. Dawson-Haggerty, Y. Chen, C. J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst.*, 2010, pp. 1–14.

[4] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proc. IEEE/ACM IPSN*, Apr. 2011, pp. 73–84.

[5] M. Doddavenkatappa, M. C. Chan, and B. Leong, "Splash: Fast data dissemination with constructive interference in wireless sensor networks," in *Proc. NSDI*, 2013, pp. 269–282.

[6] Y. Wang, Y. He, D. Cheng, Y. Liu, and X.-Y. Li, "TriggerCas: Enabling wireless consrucive collisions," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 480–484.

[7] L. Kong and X. Liu, "mZig: Enabling multi-packet reception in ZigBee," in *Proc. ACM MobiCom*, 2015, pp. 552–565.

[8] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, Jun. 2002, pp. 1567–1576.

[9] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys*, 2004, pp. 95–107.

[10] F. Österlind, L. Mottola, T. Voigt, N. Tsiftes, and A. Dunkels, "Strawman: Resolving collisions in bursty low-power wireless networks," in *Proc. IEEE/ACM IPSN*, Apr. 2012, pp. 161–171.

[11] T. Schmid, P. Dutta, and M. B. Srivastava, "High-resolution, low-power time synchronization an oxymoron no more," in *Proc. IEEE/ACM IPSN*, 2010, pp. 151–161.

[12] G. R. Woo, P. Kheradpour, D. Shen, and D. Katabi, "Beyond the bits: Cooperative packet recovery using physical layer information," in *Proc. ACM MobiCom*, 2007, pp. 147–158.

[13] M. Dell'Amico, A. Lodi, and S. Martello, "Efficient algorithms and codes for K-cardinality assignment problems," *Discrete Appl. Math.*, vol. 110, no. 1, pp. 25–40, Jun. 2001.

[14] *Submodular Set Function*. Accessed: Feb. 1, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Submodular_set_function

[15] Y. Gu and T. He, "Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links," in *Proc. ACM SenSys*, 2007, pp. 321–334.

[16] S. Gobriel, D. Mosse, and R. Cleric, "TDMA-ASAP: Sensor network TDMA scheduling with adaptive slot-stealing and parallelism," in *Proc. IEEE ICDCS*, Jun. 2009, pp. 458–465.

[17] L. Cheng *et al.*, "Taming collisions for delay reduction in low-duty-cycle wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[18] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proc. ACM SIGCOMM*, 2005, pp. 133–144.

[19] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: Opportunistic routing meets duty cycling," in *Proc. IEEE/ACM IPSN*, Apr. 2012, pp. 185–196.

[20] Z. Cao, Y. He, and Y. Liu, "L2: Lazy forwarding in low-duty-cycle wireless sensor network," in *Proc. IEEE INFOCOM*, 2012, pp. 922–930.

[21] D. Couto, D. S. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. ACM MobiCom*, 2003, pp. 419–434.

[22] K. Srinivasan *et al.*, "The κ-factor: Inferring protocol performance using inter-link reception correlation," in *Proc. ACM MobiCom*, 2010, pp. 317–328.

[23] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Ad Hoc Netw.*, vol. 1, no. 1, pp. 107–123, Jul. 2003.

[24] O. Chipara, C. Lu, and J. Stankovic, "Dynamic conflict-free query scheduling for wireless sensor networks," in *Proc. IEEE ICNP*, Nov. 2006, pp. 321–331.

[25] B. Yu, J. Li, and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2159–2167.

[26] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A hybrid MAC for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[27] G. S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks," in *Proc. ACM SenSys*, 2006, pp. 293–306.

[28] F. Li, J. Luo, G. Shi, and Y. He, "ART: Adaptive frequency-temporal co-existing of ZigBee and WiFi," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 662–674, Mar. 2017.

[29] M. Jain *et al.*, "Practical, real-time, full duplex wireless," in *Proc. ACM MobiCom*, 2011, pp. 301–312.

Shuai Wang received the B.S. and M.S. degrees from the Huazhong University of Science and Technology, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA, in 2017.

He is currently a Researcher Professor with the Department of Computer Science, Southeast University. His research interests include Internet of Things, cyber-physical systems, data science, and wireless networks and sensors.

Song Min Kim received the B.E. and M.E. degrees from the Department of Electrical and Computer Engineering, Korea University, Seoul, South Korea, in 2007 and 2009, respectively, and the Ph.D. degree from the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA, in 2016.

He is currently an Assistant Professor with the Department of Computer Science, George Mason University, Fairfax, VA, USA. His research is supported by the National Science Foundation and the Commonwealth of Virginia. His research interests include wireless and low-power embedded networks, mobile computing, Internet of Things, and cyber-physical systems.

Prof. Kim was a recipient of the Best Paper Award from the IEEE ICDCS in 2018.

Linghe Kong received the B.E. degree from Xidian University, Xi'an, China, in 2005, the master's degree from TELECOM SudParis, Paris, France, in 2007, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2012.

He was a Post-Doctoral Fellow with Columbia University, New York City, NY, USA, and McGill University. He is currently a Research Professor with Shanghai Jiao Tong University. His research interests include wireless communications, Internet of Things, and mobile computing.

Tian He (F'18) received the B.E. degree from the Nanjing University of Science and Technology, China, in 1996, the M.E. degree from the Chinese Academia of Science, China, in 2000, and the M.S. and Ph.D. degrees from the University of Virginia, Charlottesville, VA, USA, in 2002 and 2004, respectively. He is currently a Full Professor with the Department of Computer Science and Engineering, University of Minnesota-Twin Cities, Minneapolis, MN, USA. He has authored or co-authored over 250 papers in premier network journals and conferences with over 22 000 citations (H-index 64). His research includes wireless networks, networked sensing systems, cyber-physical systems, real-time embedded systems, and distributed systems.

Prof. He was a recipient of the National Science Foundation (NSF) CAREER Award in 2009, the McKnight Land-Grant Chaired Professorship in 2011, the George W. Taylor Distinguished Research Award in 2015, the China NSF Outstanding Overseas Young Researcher I and II in 2012 and 2016, respectively, and eight best paper awards in international conferences including MobiCom, SenSys, and ICDCS. He served as the general/program chair for international conferences and for many program committees. He served as an editorial board member for six international journals, including the *ACM Transactions on Sensor Networks*, the IEEE TRANSACTIONS ON COMPUTERS, and the IEEE/ACM TRANSACTIONS ON NETWORKING.