Disco: Improving Packet Delivery via Deliberate Synchronized Constructive Interference

Yin Wang, *Member, IEEE*, Yunhao Liu, *Senior Member, IEEE*, Yuan He, *Member, IEEE*,
Xiang-Yang Li, *Senior Member, IEEE*, and Dapeng Cheng, *Student Member, IEEE*

Abstract—Constructive interference (CI) enables concurrent transmissions to interfere non-destructively, so as to enhance network concurrency. In this paper, we propose deliberate synchronized constructive interference (Disco), which ensures concurrent transmissions of an identical packet to synchronize more precisely than traditional CI. Disco envisions concurrent transmissions to positively interfere at the receiver, and potentially allows orders of magnitude reductions in energy consumption and improvements in link quality. We also theoretically introduce a sufficient condition to construct Disco with IEEE 802.15.4 radio for *the first time*. Moreover, we propose Triggercast, a distributed middleware service, and show it is feasible to generate Disco on real sensor network platforms like TMote Sky. To synchronize transmissions of multiple senders at the chip level, Triggercast effectively compensates propagation and radio processing delays, and has $95th$ percentile synchronization errors of at most 250 ns. Triggercast also intelligently decides which co-senders to participate in simultaneous transmissions, and aligns their transmission time to maximize the overall link Packet Reception Ratio (PRR), under the condition of maximal system robustness. Extensive experiments in real testbeds demonstrate that Triggercast significantly improves PRR from 5 to 70 percent with seven concurrent senders. We also demonstrate that Triggercast provides $1.3\times$ PRR performance gains in average, when it is integrated with existing data forwarding protocols.

Index Terms—Constructive interference, concurrent transmissions, data forwarding, wireless sensor networks

✦

1 INTRODUCTION

IN wireless sensor networks (WSNs), it is widely believed that simultaneous transmissions will result in packet collisions. Recently works, e.g., Backcast [1] and Glossy [2], show that it is feasible for a common receiver to decode concurrent transmissions of an identical packet with high probability, if multiple transmissions are accurately synchronized. Their works enable simultaneous transmissions to interfere non-destructively, namely to generate constructive interference (CI), so as to enhance network concurrency. By leveraging CI, Glossy achieves near-optimal network flooding latency. Although CI requires different transmitters send the same packet with the same content, it can greatly increase network concurrency. Without CI, even when two nodes intend to send an identical packet to a common receiver, they need coordination to determine which node transfers the packet first. The coordination overhead is non-negligible, especially when the number of potential transmitters is large. That's why Glossy take a much shorter time and lower overhead to realize network flooding than existing

solutions (e.g., Flash [3]). Thus one question naturally arises: is traditional CI interfere constructively in the aspect of received signal strength indication (RSSI) and packet reception ratio (PRR)? Unfortunately, our extensive experiments disclose that traditional CI provides no guarantee of power gains and PRR improvements, compared with the single best link (Fig. 1a).

Deliberate synchronized constructive interference (Disco) advances the technique of constructive interference in WSNs. Disco is especially attractive for WSNs, because it potentially improves energy efficiency, and thus mitigates the issue caused by energy constraints. A set of N nodes can achieve an N^2-fold increase in the received power of *baseband* signals, compared to a single node transmitting individually. It indicates that, to achieve the same SNR, each node can reduce signal power with a factor of $\frac{1}{N^2}$, and the total power consumed by N nodes can be $\frac{1}{N}$ of the transmitting power required by a single sender. Moreover, simultaneously forwarding a packet can harness signal superposition gain, such that RSSI and PRR will be improved (Fig. 1b).

However, implementing Disco in WSNs is challenging due to the following reasons. First, simultaneous transmissions must be synchronized at the chip level, namely 0.5 μs for IEEE 802.15.4 radio. To generate CI, Glossy's synchronization is sufficient, since it compensates most factors, such as clock drifts, software routine uncertainties of OS as well as asynchronous clocks (e.g., transmitter's radio and receiver's radio, MCU and radio module). However, it is insufficient to construct Disco. Experiments reveal that propagation delays and radio processing delays significantly influence Disco generation. To make it even worse, estimating radio processing delays is a challenging task, as it varies from packet to packet, depends on the SNR, and is

---
- Y. Wang, Y. Liu, and Y. He, are with the School of Software and TNLIST, Tsinghua University, Beijing 100084, P.R. China.
  E-mail: wangyin00@gmail.com, {yunhao, he}@greenorbs.com.
- X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, and with the School of Software and TNLIST, Tsinghua University E-mail: xli@cs.iit.edu.
- D. Cheng is with the Department of Computer Science, Shandong Institute of Business and Technology, Yantai 264005, P.R. China.
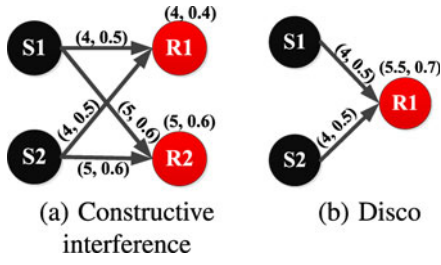 E-mail: chengdapeng@gmail.com.

Fig. 1. Compared with CI, Disco improves RSSI and PRR. The symbols (a, b) are used to describe a link, while a and b represent the RSSI and PRR, respectively.
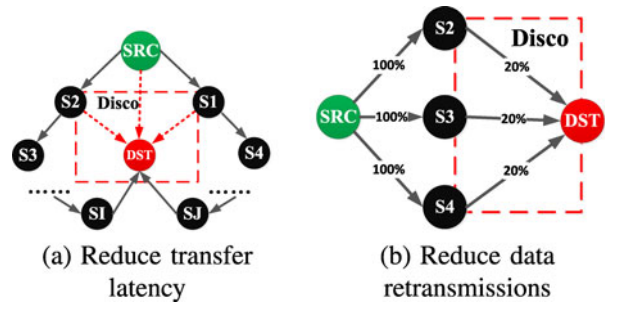

Fig. 3. (a) Disco generates a new link from three disconnected links and thus reduce data forwarding latency. (b) Disco makes use of signal superposition to improve PRR and hence reduce retransmission times.

affected by multi-path characteristics of the channel. Besides, in the absence of a central controller or a shared clock (e.g., GPS), they can only rely on their own radio signals as references.

Second, even if simultaneous transmissions are perfectly synchronized, i.e., no phase offset, they might not guarantee Disco. This is because of noises. Although signals are exactly aligned, noises superpose as well. Whether the SNR of the combined signal increases depends on SNRs and transmission (Tx) powers of individual signals.

Third, sensor nodes are usually battery-powered and have limited computational resources. It is difficult or even impossible to deploy complex signal processing algorithms in commercial off the shelf (COTS) sensor platforms.

We propose Triggercast, a practical distributed middleware to generate Disco in WSNs. Triggercast enables a co-sender (sender-initiated Triggercast, Fig. 2a) or a receiver (receiver-initiated Triggercast, Fig. 2b) to trigger a radio signal, which acts as a common reference for all concurrent senders to implement synchronized transmissions. Such advantages of Triggercast can be exploited in general data gathering scenarios of WSNs to improve network performance. Triggercast can control the network topology without increasing Tx power or adding new nodes, which potentially reduces latency of data forwarding. In Fig. 3a, with Disco, the "DST" node might overhear the packet from the "SRC" node in two hops. Triggercast can also reduce packet retransmissions by improving PRR. For example, in Fig. 3b, the ETX (Expected Transmission Count) of traditional routing is $ETX_1 = 1/0.2 + 1 = 6$, while the ETX of Triggercast might be $1/0.32 + 1 \approx 4$ (the number 0.32 comes from real measurements).

There are two key modules in Triggercast, namely the chip level synchronization (CLS) algorithm and the link selection and alignment (LSA) algorithm. CLS enables

concurrent transmissions to be synchronized in 0.5 μs, by compensating propagation and radio processing delays. Our experiments demonstrate that CLS has 95th percentile synchronization errors of at most 250 ns. In our practice, the accuracy is bounded by the running frequency (4,194,304 Hz) of on-board MCU of TMote Sky sensor node. Triggercast's LSA algorithm intelligently decides which co-senders to participate in simultaneous transmissions, and aligns their transmission time to maximize the overall link PRR under the condition of maximal system robustness. The CLS and LSA algorithms together ensure Triggercast to generate Disco in practice. Extensive experiments show that Triggercast can improve PRR from 5 to 70 percent with seven senders, and from 50 to 98.3 percent with six senders. Experiments also demonstrate that Triggercast brings $1.3\times$ PRR performance gain in average with respect to data forwarding in realistic deployments.

The contributions of this paper are summarized as follows:

i)   We are the *first* to provide a *theoretical sufficient condition (SC)* for generating Disco in WSNs.
ii) We propose Triggercast, a practical middleware to ensure concurrent transmissions to interfere constructively. The CLS algorithm implemented in Triggercast effectively evaluates and compensates propagation and radio processing delays.
iii) We implement Triggercast in real testbeds. Extensive experiments show that Triggercast can construct Disco in TMote Sky platforms. We integrate Triggercast into data forwarding protocols and show its performance gains.

2   RELATED WORK

Exploiting concurrent transmissions while suppressing interference is a promising direction, for its ability to decode packets from collisions, increase network throughput [4], [5], [6], alleviate the broadcast storm problem of ackowledgments [1] and enhance packet transmission reliability. The existing works can be categorized as signal processing based and physical-layer phenomenon based.

Works based on signal processing include ANC [7] for network coding, SIC [8] and Zigzag [9] for interference cancelation, 802.11n+ [10] for interference alignment in MIMO, AutoMAC [11] for rateless coding, full-duplex wireless
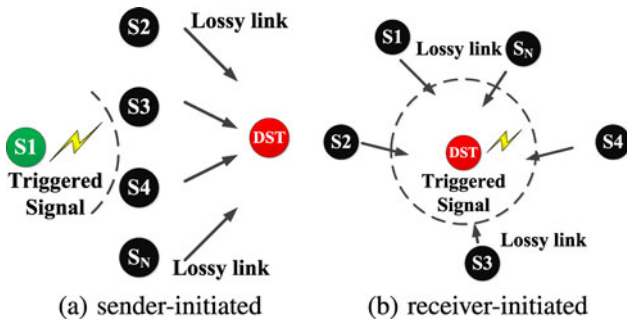

Fig. 2. Triggercast: a radio triggered concurrent transmission architecture.
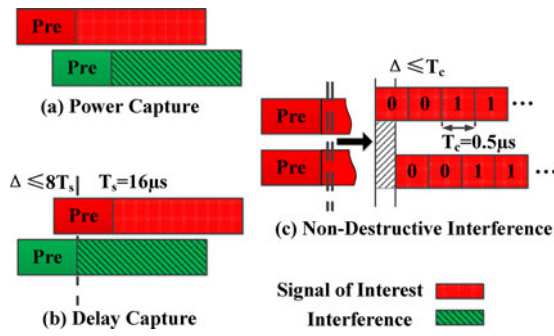
Fig. 4. Concurrent transmission techniques supported by COTS IEEE 802.15.4 transceivers.

TABLE 1
Symbols and Notations

Symbol	Definition
SNR	signal noise ratio of the received signals due to CI
PRR	packet reception ratio of the received signals due to CI
RSSI	received signal strength indication due to CI
ETX	expected transmission count of data forwarding
T_c	chip time of IEEE 802.15.4 signal
T_x	transmission power of a packet
τ_i	phase offset of the ith arriving signal

radios [12], and ZIMO for coexistence of ZigBee and WiFi networks [13]. Those works leverage powerful software-defined radio platforms (e.g., USRP), and cannot be directly applied in WSNs.

Physical-layer phenomenon based works mainly focus on exploring wireless radio properties of COTS transceivers. Such physical-layer phenomena mainly include capture effect [14] and message-in-message (MIM) [15] (in Fig. 4). Capture effect requires that the signal of interest is sufficiently stronger than the sum of interference. MIM needs special hardware support to continuously synchronize with the preamble of the stronger signal. Both of them can only decode the stronger signal at the cost of dropping the other signals.

Recently, Backcast [1] experimentally discovers that, concurrent transmissions of short acknowledgment packets automatically generated by the radio hardware can interfere non-destructively. This characteristic can be utilized to alleviate the ACK implosion problem [16]. Glossy [2] advances that work of CI by implementing designs such as interrupt compensation and precise timing controls. Though multiple senders transmitting the same packet many times means consuming more power and needs cooperation, CI is reasonable because it greatly reduces the time incurred by collision scheduling, and thus improves network throughput. By leveraging CI, Glossy achieves a magnitude of millisecond flooding latency in a network of 94 TelosB motes. The flooding latency of Glossy is almost 1 percent of that with traditional network flooding protocols such as Flash [3]. Splash [17] exploits CI and channel diversity to effectively create fast and parallel pipelines over multiple paths for data dissemination. However, all those works cannot ensure wireless transmissions interfere constructively.

Triggercast's radio-triggered synchronization mechanism is comparable with those in SourceSync [18] and Glossy [2]. SourceSync exploits the fundamental property of FFTs to evaluate the radio processing delay, which varies dynamically in multi-path channels. Unfortunately, single-carrier communication (e.g., IEEE 802.15.4) systems cannot benefit from the method introduced by SourceSync. Glossy is able to synchronize multi-hop packet transmissions at the magnitude of sub-microseconds, which is previously considered too challenging to implement on COTS sensor platforms. However, for signals to superpose constructively, we need more precise synchronization algorithms to compensate propagation delays and radio processing delays. Triggercast aims to address the above-mentioned problems.

3 PACKET TRANSMISSIONS OVER INTERFERENCES

3.1 Background

For ease of presentation, Table 1 lists the main symbols and notations used in this paper.

Concurrent transmissions. In Fig. 4, CI originates from the scenario that multiple spatially distributed transmitters send an identical packet to a common receiver simultaneously. Traditionally, one may believe that concurrent packet transmissions will collide and *prevent* the common receiver from successfully decoding the packet, if the receiving power of the common receiver for each transmission is the same (in this case, no capture effect happens). However, by employing CI, the receiver can decode the packet with high probability if the maximal temporal displacement of concurrent transmissions is within a chip time T_c, namely, 0.5 μs for the IEEE 802.15.4 radio. Indeed, CI is a kind of cooperative networks. CI requires the same packet and more accurate synchronization. CI differs from MIMO for CI doesn't require signal processing technique to decode collided signals and leverages the physical layer tolerance for multi-path signals. Therefore, CI is light-weighted and is especially attractive in many application scenarios of WSNs, such as network flooding, data dissemination, time synchronization, etc.

3.2 Does CI Really Interfere Constructively?

The interference due to concurrent packet transmissions is non-destructive if it *doesn't destroy* the normal packet reception. Note that CI requires different nodes transmit the same packet and thus may consume more power, more channel resources and require collaboration. One question naturally arises: is the aggregated effect of multiple concurrent transmissions better than arbitrary single packet transmission?

To test whether CI interfere constructively, we use three TMote Sky sensor nodes, each of which has CC2420 transceiver and MSP430F1611 micro-controller. One node is selected as the initiator (I), while the other two are selected as receivers ($R1$ and $R2$). We leverage Glossy source code, an open-source project running on Contiki OS, to generate CI and add a function for runtime parameter adaption. We do experiments in both outdoor environments and indoor environments. In each experiment, we fix the positions of I and $R1$, making them 1 meter away from each other. We move the position of $R2$, altering its distance to I from 1 meter to 100 meters (outdoor) or from 1 meter to 55 meters (indoor). We keep each individual link a perfect link (e.g., PRRs are larger than 99 percent), and measure the received PRR and RSSI of I when two receivers transmit packets simultaneously. The experimental results are illustrated in Figs. 5 and 6.
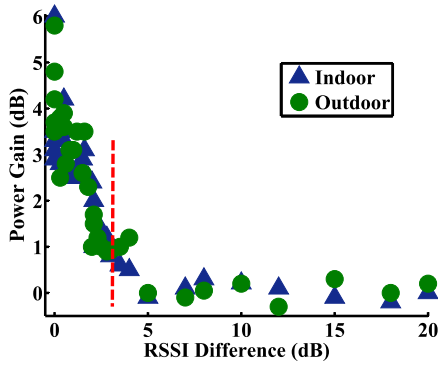
Fig. 5. No obvious power gain if RSSI differences of R1 and R2 exceed 3 dB.

To verify whether CI can bring power gain, we fix the positions of $R1$ and $R2$ at 1 meter distance away from I. We measure the received RSSI gain of concurrent transmissions compared with the single better link transmitting individually, drawn as Y-axis in Fig. 5. The X-axis denotes the received RSSI difference when two receivers transmit independently. It can be observed from Fig. 5 that, the best case of power gain due to two concurrent transmissions can be as high as 6 dB. However, if the received RSSI difference is larger than 3 dB, there is no noticeable power gain. It is possible that the *capture effect dominates the packet reception*. The results indicate that adding more senders *can not* help to increase the received RSSI under the condition of capture effect.

We also find that more senders may not lead to PRR improvement. Besides, more senders might degrade PRR significantly with CI. We change the distance between $R2$ and I, and record the PRR when $R1$ and $R2$ transmit simultaneously, as shown in Fig. 6. To mitigate the effect of capture effect, we make sure the RSSI values of successful packet receptions of each individual receiver are almost the same. Indeed, due to multi-path effect and external interferences, the received RSSI values are not always consistent. We accurately adjust parameters such as Tx powers, antenna directions and retransmission times, to make sure the RSSI values keep consistent in a short time interval. It can be seen from Fig. 6, when the distance between $R2$ and I increases, the PRR performance of node I drops apparently. The experiments indicate that *propagation delays* are
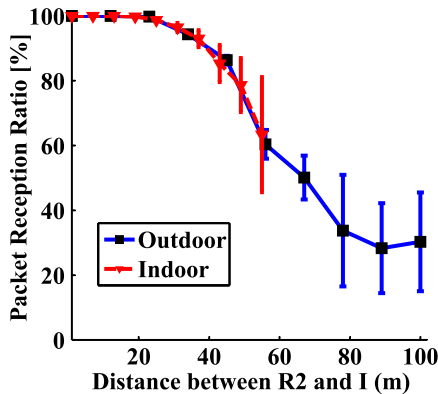

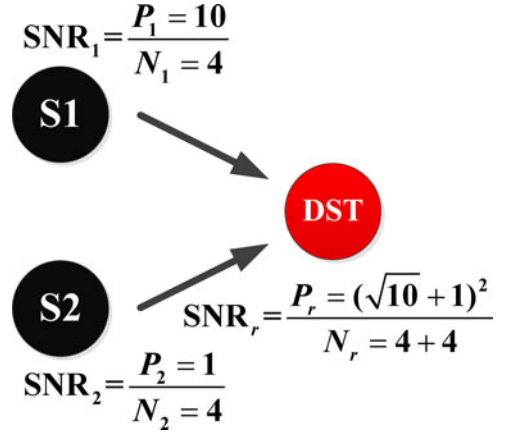Fig. 6. PRR drops quickly as differences of propagation delay increase.

Fig. 7. Two signals superpose destructively even if perfectly aligned.

also a key factor related to PRR, even if the differences of transmission distances are only about 40 meters. In order to ensure that concurrent transmissions interfere constructively, we must compensate propagation delays of spatially distributed transmitters.

4   A SUFFICIENT CONDITION FOR GENERATING DISCO

Could we ensure to construct Disco, if we compensate different delay uncertainties and perfectly align concurrent transmissions? To answer this question, we first take a simple case in Fig. 7, and then provide theoretical analysis to yield a sufficient condition for generating Disco in WSNs.

We assume sender S1 and S2's signals arriving at the antenna of the destination node have unified signal power 10 and 1, as well as noise power 4 and 4 respectively. Even if the signals of S1 and S2 exactly align at the common receiver, the effective power of the superposed signal is $(\sqrt{10} + 1)^2 \approx 17.3$, while the noise power equals 8. The SNR (2.17) of superposed signal degrades slightly compared with the single best signal (2.5). This simple case indicates that only chip level synchronization is insufficient for Disco to function. Exactly synchronized signals with different link qualities might also superpose destructively. In the following discussions, we will focus on *baseband* signals, and examine the role of link quality (e.g., PRR, SNR) through waveform analysis.

The basic principle of 802.15.4 PHY layer is elaborated in [19]. Let $S_{msk}(t)$ be the transmitted signal after MSK modulation, $I(t)$ and $Q(t)$ denote the in-phase component and quadrature-phase component respectively. Let $\omega_c = \pi/2T_c$ represent the angular frequency of half-sine pulse shaping. The combined MSK signal can be calculated as

$$S_{msk}(t) = I(t)\sin\omega_c t - Q(t)\cos\omega_c t, \qquad (1)$$

$$\text{where}\begin{cases} l(t) = \sum_n (2C_{2n} - 1)\text{rect}\left(\frac{t}{2} - nT_c\right) \\ Q(t) = \sum_n (2C_{2n+1} - 1)\text{rect}\left(\frac{t}{2} - nT_c - \frac{T_c}{2}\right). \end{cases} \qquad (2)$$

Here, $C_n \in \{0, 1\}$ represents the nth chip, and rect() function is a rectangle window, defined as

$$\text{rect(t)} \triangleq \begin{cases} 1 & 0 \leq t \leq T_c \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

After Rayleigh multi-path channel, the received signal $S_R(t)$ is convolution of the original signal and the channel $H(t)$

$$S_R(t) = S_{msk}(t) * H(t) + N(t), \quad (4)$$

where $N(t)$ represents the channel noise. We suppose there are N transmitters $\{T_i, i = 1, 2, \ldots, N\}$ simultaneously sending an identical packet to a common receiver R. All the transmissions have already been synchronized at chip level relative to the strongest signal. In our experiments with the CC2420 chip, we find the receivers always synchronize with the strongest signal. The output signal from each transmitter T_i arriving at the antenna of the receiver R is denoted as $S_R^i(t)$. Let λ_i be SNR of the output signal $S_R^i(t)$, P_i denote average power of signal $S_R^i(t)$ and N_i represent power of noise $N_i(t)$. Obviously, we have $\lambda_i = \frac{P_i}{N_i}$. The SNR λ_i is mainly determined by the radio propagation environments (e.g., multi-path channels, interference) and Tx powers of the senders. The received superposed signal $\overline{S_R(t)}$ is the sum of the N output signals $S_R^i(t)$. Hence we can approach

$$\overline{S_R(t)} = \sum_{i=1}^{N} \left(A_i S_R^i(t - \tau_i) + N_i(t) \right), |\tau_i| \leq T_c, \quad (5)$$

where A_i and τ_i respectively depict the unified amplitude and phase offset of the ith arriving signal relative to the instant when the strongest signal reaching the receiver. Let $S_R^1(t)$ be the strongest signal. Correspondingly, we have $A_1 = 1$, $\tau_1 = 0$, $P_i = P_1 A_i^2$. According to [20], it can be derived that the effective power $\overline{P}$ of superposed signals after demodulation is

$$\overline{P} = P_1 \left(\sum_{i=1}^{N} A_i \cos \omega_c \tau_i \right)^2, \quad (6)$$

while the aggregated power of noise $\overline{S_R(t)}$ is

$$\overline{N} = \sum_{i=1}^{N} \frac{P_i}{\lambda_i}. \quad (7)$$

As a result, the SNR of the received superposed signal is

$$\frac{\overline{P}}{\overline{N}} = \frac{P_1 \left(\sum_{i=1}^{N} A_i \cos \omega_c \tau_i \right)^2}{\sum_{i=1}^{N} P_i/\lambda_i} \leq \frac{P_1 \sum_{i=1}^{N} A_i^2 \sum_{i=1}^{N} (\cos \omega_c \tau_i)^2}{P_1 \sum_{i=1}^{N} A_i^2/\lambda_i}. \quad (8)$$

The inequality (8) can be derived by Cauchy-Schwarz inequality and equality holds if the condition satisfies

$$\frac{A_i}{\cos \omega_c \tau_i} = \frac{A_j}{\cos \omega_c \tau_j}, (\forall i, j). \quad (9)$$

To guarantee the received SNR of the superposed signal is better than the SNR of any single signal in the worst case, namely to ensure simultaneous transmissions to interfere positively, it is required that the maximum value of the received SNR is no less than $\lambda_{\max}$

$$\left(\frac{\overline{P}}{\overline{N}} \right)_{max} > \lambda_{\min} \sum_{i=1}^{N} (\cos \omega_c \tau_i)^2 \geq \lambda_{\max}. \quad (10)$$

The existing proposal, Glossy [2], performs experiments to validate that multiple IEEE 802.15.4 signals can interfere *non-destructively* if the maximum temporal displacement is less than $T_c = 0.5 \, \mu s$. Since Disco requires more strict conditions than CI, consequently, we derive a theoretical *sufficient condition* for concurrent transmissions with IEEE 802.15.4 radio to interfere constructively.

i) Concurrent transmissions with an *identical* packet should be synchronized at *chip* level, namely less than $T_c = 0.5 \, \mu s$;

ii) The phase offset of the ith arriving signal should satisfy: $|\tau_i| \leq \cos^{-1} \frac{P_i}{P_1}/\omega_c$ (SC-I);

iii) The ratio of the minimum SNR $\lambda_{\min}$ and the maximum SNR $\lambda_{\max}$ of current transmissions should satisfy: $\frac{\lambda_{\min}}{\lambda_{\max}} \geq \frac{1}{\sum_{i=1}^{N} (\cos \omega_c \tau_i)^2}$ (SC-II).

5 TRIGGERCAST IMPLEMENTATION

5.1 Triggercast Overview

In this section, we introduce the implementation of Triggercast to generate Disco in WSNs. As illustrated in Fig. 2, Triggercast leverages the instant of a triggered signal as a common reference for all concurrent senders to implement synchronized packet transmissions. In the MAC layer design, the trigger node utilizes a standard CSMA/CA protocol to acquire the access to the the medium. Once the trigger node senses the channel is free, it first broadcasts a synchronization packet, and tells all the co-senders about the destination and when to start forwarding the data. After a promissory duration of time (e.g., tens of ms), all the co-senders begin to transmit simultaneously.

In the PHY layer design, we propose the chip level synchronization and link selection and alignment algorithms in Triggercast to ensure concurrently transmitted packets interfere constructively. For receiver-initiated Triggercast, the receiver first performs LSA to select which links will participate in concurrent transmissions. For sender-initiated Triggercast, each co-sender individually runs the LSA, to determine whether it joins in the concurrent transmission process. Then the selected senders will use CLS to evaluate propagation and radio processing delays. Finally, they insert a number of *no operations (NOPs)* (Eq. (15)) to compensate the evaluated delays and phase offsets obtained in LSA. Triggercast can leverage normal packet transmissions to obtain parameters for CLS and LSA, which can significantly mitigate the overhead.

It should be noticed that, we assume no node mobility and duty-cycle for Triggercast. We mainly focus on
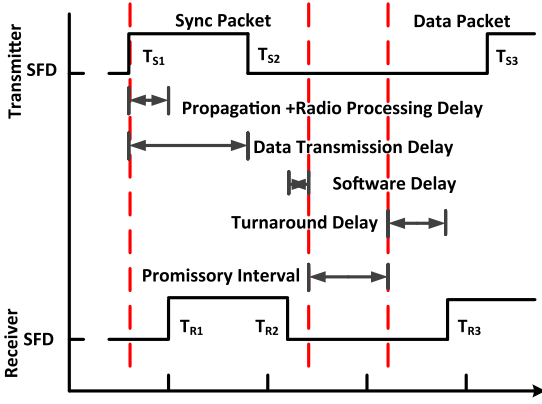
Fig. 8. The timing diagram of SFD signal for TMote Sky node with Triggercast (without Preamble).

improving the synchronization of concurrent transmissions and the benefits of Disco. Therefore, we assume all the nodes are always-on transceivers. As for the duty cycle, asynchronous nodes might not receive the triggered packet. Therefore, we can increase the header of triggered packet to guarantee all the neighbors can receive the triggered packet and generate Disco.

5.2 Chip Level Synchronization

5.2.1 Timing Diagram Analysis

Fig. 8 illustrate the detailed SFD pin activities during a packet transmission and reception. From Fig. 8, it can be noticed that the synchronization accuracy involves multiple factors, such as the propagation delay, the radio processing delay introduced by the radio at the beginning of a packet reception, the hardware turn around delay from the reception state to the transmission state, and the software delay. Note that the data transmission delay is a fixed value, determined by packet length.

(a) *Radio processing delay* describes the time between the arrival of the packet at the antenna and the instant when the radio circuits successfully decode the first sample. Estimating radio processing delay is a challenging task, as it varies from packet to packet, depends on the SNR, as well as the multi-path characteristics of the channel. Moreover, the asynchronous radio clocks between the transmitter and the receiver also cause a uniform distributed quantization error.

(b) *Propagation delay* is the signal's flight time between transmitter and receiver. The propagation delay is determined by the distance of a transmitter-receiver pair.

(c) *Software delay* is defined as the duration from the falling edge of the SFD interrupt to the end of a successful packet reception. The software delay uncertainty mainly depends on variable interrupt serving delays, and the unsynchronized clocks between the MCU and the radio module. The interrupt serving delays can be accurately evaluated and compensated with the method explained by Glossy [2]. The new generation chip CC2530 integrates MCU and radio module in one chip with synchronized clock frequency, indicating the software delay uncertainty can be perfectly eliminated.

(d) *Hardware turnaround delay* is the time required for a node to switch from packet reception phase to transmission phase. The hardware turnaround delay is constant, and determined by the speed of the radio front end.

5.2.2 Delay Measurement and Compensation

From Fig. 6, we conclude that propagation delays of spatially distributed transmitters must be compensated, in order to make concurrent transmissions positively superpose. The Tmote Sky node has an internal Digitally Controlled Oscillator (DCO) operating at frequency $f_p = 4{,}194{,}304$ Hz, which means the evaluated propagation delays can only have an accuracy of 0.238 μs (about 71.5 meters). To make it even worse, the frequency of the DCO can deviate up to $\pm 20\%$ from the nominal value, with the temperature and voltage drifts of $-0.38\%^\circ$C and 5%/V. Moreover, the pair-wise packet transmissions method can only measure the sum of the propagation delay and the radio processing delay. It is challenging to subtract the radio processing delay in realistic environment, as it varies from one packet to another, and is influenced by communication link quality. Fortunately, we manifest that the compensation of the *sum* of the propagation delay and the radio processing delay is sufficient for chip level synchronization. This task is still difficult due to many uncertain factors with regard to time, such as the quantization uncertainty, the software delay uncertainty due to asynchronous radio clocks, and clock drifts due to packet transmissions.

Methodology. According to the *law of large numbers*, the average of the results obtained from a large number of trails should be close to the expected value. Inspired by this, we select a transmitter-receiver pair which is 40 meters apart in an indoor environment, and let the transmitter periodically send a packet every 500 ms. Once the receiver successfully decodes a packet, it piggybacks a reply packet as soon as possible to the transmitter. As shown in Fig. 8, the time stamps T_{S1} and T_{S2} represent the phases, when the sender's radio starts transmitting a packet and ends a packet transmission, while the time-stamp T_{S3} denotes the phase when the radio begins a packet reception. The time-stamps T_{R1}, T_{R2} and T_{R3} characterize the phases when the receiver's radio starts a packet reception, ends a packet reception, and begins a packet transmission, respectively. The TMote Sky node can accurately capture the exact instants when MCU detects the rising edge and the falling edge of SFD interrupts, with MCU's timer capture functionality. The nth packet sent by the receiver includes time-stamps $T_{R1}(n)$, $T_{R2}(n)$ and $T_{R3}(n-1)$, which can be used by the transmitter, to evaluate the expected value of radio processing delay and propagation delay

$$\widehat{\Delta} = \frac{(\widehat{T_{S3}} - \widehat{T_{S1}}) - (\widehat{T_{R3}} - \widehat{T_{R1}})}{2}, \qquad (11)$$

where the symbol $\widehat{\lambda}$ defines the mean value of λ.

Experimental results of delay measurement using Eq. (11) are displayed in Fig. 9 as the 'raw' curve. Unfortunately, the result is pessimistic. The measured delay ranges from 0.596 to 5.01 μs, with average value 2.32 μs and variance 0.628 μs. The instability of measured delay
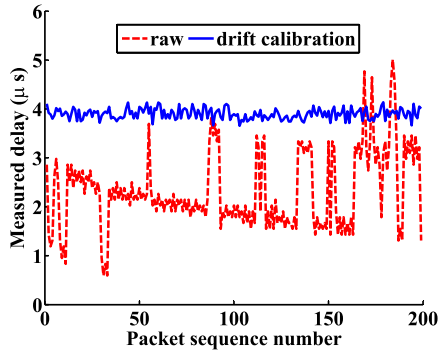
Fig. 9. Measured and calibrated delays of propagation and radio processing.

indicates that it is difficult to synchronize different transmitters at a magnitude of 0.5 μs, if we directly use the measured data for compensation. Fortunately, we find that the data transmission delay is the same for all nodes. And thus we have

$$T_{S2}(n) - T_{S1}(n) = T_{R2}(n) - T_{R1}(n). \qquad (12)$$

The data transmission delays of the transmitter and the receiver are drawn in Fig. 10.

We further find that the measured data transmission delays are not stable for the transmitter-receiver pairs. The instability is due to different factors, such as the jitters, clock drifts as well as hardware diversities of the nodes' DCOs. The drifts can be as high as 5,000 ppm according our measurement. We define $\chi(n) = (T_{S2}(n) - T_{S1}(n))/(T_{R2}(n) - T_{R1}(n))$ as the unified clock drift coefficient relative to the receiver. Consequently, we can calibrate Eq. (11) as

$$\widehat{\Delta}_{cal} = \frac{\text{mean}\left(\frac{T_{S3}(n) - T_{S1}(n)}{\chi(n)}\right) - (\widehat{T_{R3}} - \widehat{T_{R1}})}{2}. \qquad (13)$$

We obtain the expected radio processing and propagation delay represented by DCO Ticks after the calibration of Eq. (13). To translate them to time, we also utilize the *Virtual High-resolution Time (VHT)* [21], which calibrates the receiver's DCO with more stable external 32,768 Hz crystal as a reference. The measured propagation and radio precessing delay after clock drift calibration is shown as the 'drift calibration' curve in Fig. 9. The
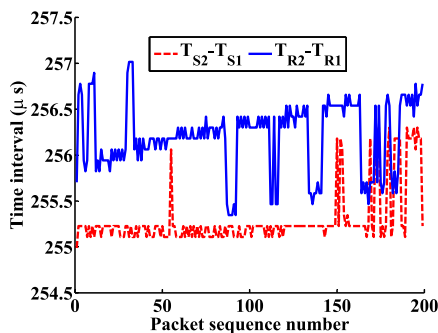

Fig. 10. Measured delays of data transmission for the transmitter and the receiver using DCO ticks.
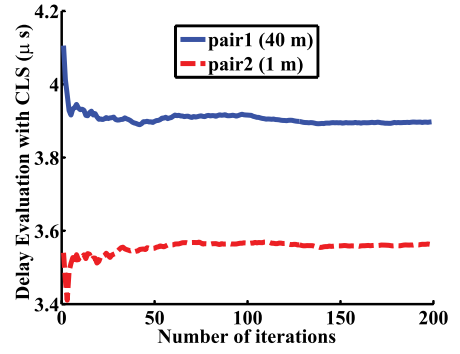

Fig. 11. Convergence time analysis of CLS. About 22 runs are enough for delay evaluation.

calibrated delay ranges from 3.66 to 4.12 μs, with average value 3.90 μs and variance 0.012 μs.

Convergence time. To measure CLS's convergence performance, we do experiments with two transmitter-receiver pairs. Pair 1 and pair 2 are 40 meters and 20 meters away respectively. We average the calibrated delay with Eq. (13), and test how well CLS works in terms of convergence time. Experimental results are shown in Fig. 11. The medium convergence time are 22 runs. Since each run can be done in at least 892 μs, the delay evaluation algorithm consumes about 20 ms and can be accomplished adaptively as the channel state dramatically changes. We disclose that, in our measurements, the delays don't change so much as thought before. The measurement delay are almost constant, unless the nodes move or the channel significantly changes.

5.3 Link Selection and Alignment

Note that SNR can be derived from PRR through theoretical models [20] or online measurements [22]. The relationships between SNR and PRR can be mapped as look-up tables and stored in the external flashes of sensor nodes for Triggercast to use. Even if all the concurrent transmissions are synchronized at the chip level with CLS, we have to choose the best links to satisfy the proposed sufficient condition in Section 4. The problem to make concurrent transmissions superpose constructively can be formalized as a Disco-generation problem.

Disco-generation problem. Let $\Phi = \{L_1, L_2, \dots, L_N, L_i = (P_i, \lambda_i)\}$ define a lossy link set, where P_i and λ_i denote the received signal's RSSI and SNR of transmitter T_i respectively. The problem is to find a lossy link subset Ω, in order to maximize the superposed signal's SNR on condition that the combined link is better than any lossy link in Φ and the phase offset τ_i is as large as possible.

We define a link pair (L_i, L_j) is *ordered* if $P_i \geq P_j$ indicates $\lambda_i \geq \lambda_j$, $1 \leq i, j \leq N$. According to the sufficient condition for CI, it can be proved that this problem is NP-hard if there exists any disordered link pair in Φ. For IEEE 802.15.4 signals, RSSI functions monotonically with PRR and thus with SNR and it is reasonable to assume all the link pairs in Φ are ordered.

The pseudocode of LSA is described in Algorithm 1. The for-loop can safely break if λ_i doesn't satisfy SC-II of the sufficient condition. For all $\lambda_j \leq \lambda_i$, we have $\tau_j \geq \tau_i$ and thus we can prove that they all don't satisfy SC-II:
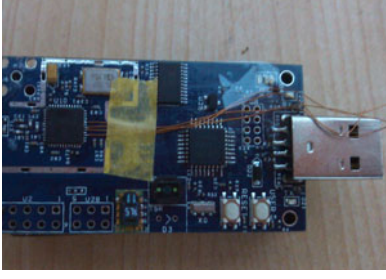
Fig. 12. We probe pin activities from CC2420 with very thin enameled wire.

Fig. 13. Controlled experiment with equal-power, equal-path delay.

$$\frac{\lambda_j}{\lambda_{\max}} \leq \frac{\lambda_i}{\lambda_{\max}} = \frac{1}{\sum_{k=1}^{i-1} (\cos \omega_c \tau_k)^2 + (\cos \omega_c \tau_i)^2}$$
$$\leq \frac{1}{\sum_{k=1}^{i-1} (\cos \omega_c \tau_k)^2 + (\cos \omega_c \tau_j)^2}. \tag{14}$$
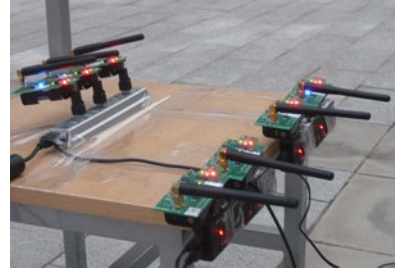
Fig. 14. Realistic experiment with equal-power, different-path delay.

Algorithm 1: Link Selection and Alignment

Input: Given an ordered lossy link set $\Phi < P_i, \lambda_i >$, where P_i and λ_i represent the received RSSI and SNR.

Output: A lossy link subset $\Omega < P_j, \lambda_j, \tau_j >$ to maximize the superposed signal's SNR, where τ_j is the maximal allowed phase offset.

1 Sort Φ, store the result as Φ' and get the best link $< P, \lambda >$ in Φ'
2 Insert link $< P, \lambda >$ and "0" (phase offset) in empty set Ω
3 **for** $i = 2 : N$ **do**
4   get the best link $< P_i, \lambda_i >$ in sorted set Φ', store as link $< P_t, \lambda_t >$;
5   calculate maximal allowed phase offset τ_i of link $< P_t, \lambda_t >$ using link $< P, \lambda >$;
6   use set Ω, link $< P_t, \lambda_t >$, phase offset τ_i to verify SC-II;
7   **if** *SNR of link $< P_t, \lambda_t >$ satisfies SC-II* **then**
8   insert link $< P, \lambda >$, phase offset τ_i to set Ω;
9   **else**
10   **break;**
11   **end**
12   **end**
13 **end**

Fig. 15. The illustration of ten links selected and aligned by LSA with max system robustness.

Time complexity. Clearly, the for-loop has $O(n)$ time complexity. Therefore, the time complexity of LSA algorithm is dominated by the sort function. Thus the time complexity of LSA algorithm is $O(n \log n)$.

Total compensation time. In Fig. 15, LSA selects the 10 best links in the overall link set, and aligns the orders and time intervals of concurrent transmissions using those links. We let τ_i as large as possible to obtain the system's maximal robustness for synchronization errors. Therefore, from (SC-I) in Section 4, we can derive $|\tau_i| = \cos^{-1} \frac{P_j}{P_i} / \omega_c$. Consequently, we select the value of τ_i, to minimize the total number N_{com} of NOPs for the co-sender T_i

$$N_{com} = [(T - \widehat{\Delta}_{cal} + \tau_i) f_p], \tag{15}$$

where $[\bullet]$ is the round function, and T is a predefined maximum delay calibration time.

6 PERFORMANCE

We have implemented a prototype Triggercast on TMote sky sensor nodes. The software is based on Contiki OS. During the overall Triggercast's duration, except for the promissory interval, all the relevant interrupts and hardware timers that are not essential to Triggercast's functioning are disabled. Since this interval is very short (several milliseconds), it is feasible that Triggercast doesn't influence the upper layer's functionality. A runtime parameter adjustment software is developed, to make sure we can online change the system running parameters. We test the performance of Triggercast through both controlled experiments (Fig. 13) and practical experiments (Fig. 14).

6.1 Synchronization Accuracy

We first test the synchronization performance of multiple concurrent transmitters. We use three TMote sky nodes, one as a receiver and two as transmitters. We set the promissory interval parameter at 0 in receiver-initiated Triggercast. We connect the SFD pins of the receiver (R) and one of the transmitters (S1) to an Agilent MSO-X serial oscilloscope (Fig. 12). The other transmitter (S2) is
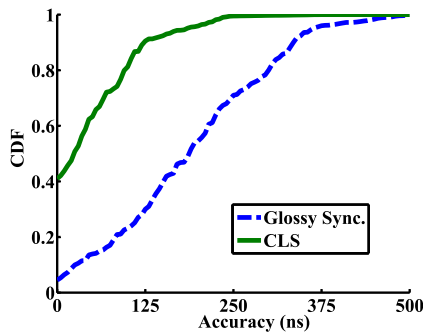
Fig. 16. Synchronization error of Triggercast less than 250 ns has more than 95 percent confidence.

Fig. 17. RSSI performance in controlled experiments.

30 meters away from the receiver in an indoor environment. However, it is difficult to measure the synchronization of S1 and S2 directly with the oscilloscope. Hence, we use R as a reference node. The synchronization between S1 and R can be monitored by the oscilloscope with a granularity of 5 ns. The durations between T_{R1} and T_{R3} (Fig. 8) of the receiver are accurately measured when S1 and S2 transmit independently. The differences of the durations can be used for synchronization accuracy measurement, since both S1 and S2 rely on the instant T_{R1} as a reference. The CDF of synchronization errors compared with the Glossy synchronization algorithm are illustrated in Fig. 16. Triggercast's CLS algorithm can synchronize multiple transmitters at a magnitude of 250 ns. The accuracy is limited by the operating frequency of the MCU of TMote Sky sensor nodes. The Glossy synchronization algorithm degrades as the distance differences between two transmitter-receiver pairs increase. CLS outperforms Glossy because CLS compensates the time due to propagation and radio processing delays.

6.2 Power Gains and PRR Improvements

The main purpose of this paper is to make wireless collisions interfere constructively. In words, multiple senders transmit an identical packet simultaneously can improve RSSI and PRR. We carry out experiments by implementing Triggercast in both controlled (Fig. 13) and realistic experiments (Fig. 14).

The carefully-controlled experiments aim to test CI performance without the influences of wireless channels. In Fig. 13, 7 2-way RF splitters are connected to function as a eight-way RF splitter. The sink node transmits a signal, which is used to trigger concurrent transmissions of eight receivers. Signals of concurrent transmissions from eight receivers have equal power and are combined in the eight-way RF splitter. We measure the received RSSI of the sink node versus the number of concurrent transmissions. The max, mean and min RSSI curves are illustrated in Fig. 17. It can be observed that the mean received RSSI increases almost monotonically with the number of simultaneous transmissions. It is not surprising because for every doubling of nodes, an additional 3 dB of power is injected into the channel. The power gain due to seven concurrent transmissions can be about 5 dB. It is slightly more surprising that RSSI decreases when two concurrent transmissions interfere. The possible reason behind that is
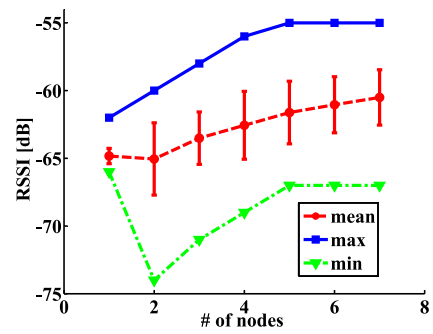
both positive (increasing RSSI) and negative (decreasing RSSI) effects of concurrent transmissions are at play. Our obtained results are very similar with the experiments of Backcast [1].

In practical experiments, up to eight senders with all three different kinds of links (disconnected (PRR < 5%), intermediate (5% < PRR < 90%), connected (PRR > 90%)), are executed to transmit packets at the same time. Due to the limit of physical space, we randomly insert NOPs to simulate different propagation and radio processing delays. We adjust the received RSSIs of each sender's individual packet transmission to almost the same, to eliminate the influence of capture effect. All the results are averages of more than 1,000 tests. Fig. 18 lists the power gains due to multiple senders of different link types (disconnected link: 1-5 dB, intermediate link: 2-6 dB, connected link: 2-6 dB). The maximum power gain can approach N^2 for N concurrent transmitters. However, the aggregated power does not monotonically increase when the number of nodes increases. The power gain due to seven concurrent transmitters is about 5 dB while that of eight concurrent transmitters reduces. As the number of concurrent transmitters increases, it is difficult to realize accurate synchronization due to clock drifts, and multi-path effects. Fig. 19 shows that PRR can be significantly improved by leveraging CI. For seven disconnected links, the PRR achieves almost 70 percent, which is better than our previous understandings of harnessing sender diversity gain $(1 - (1 - 0.05)^3 \approx 30.2\%)$. For seven disconnected links, to successfully send a packet, the ETX is 20 while for Disco, the ETX $= 1/0.7 \times 7 = 10$. It should be noticed that Disco needs multiple senders transmit the same packet several times. Therefore, Disco might transmit more packets than traditional data forwarding
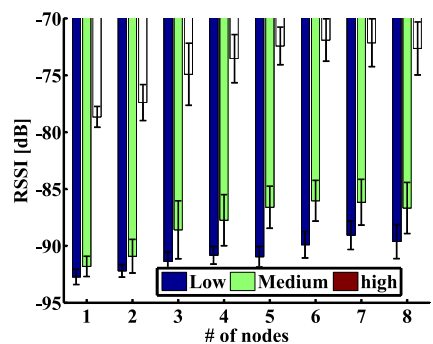

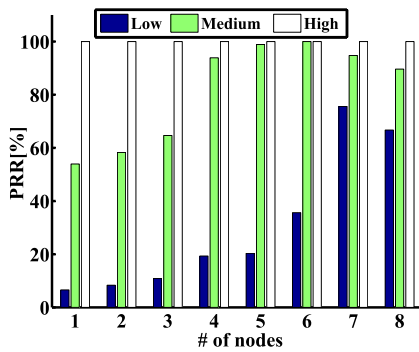Fig. 18. Triggercast increases RSSI.

Fig. 19. Triggercast improves PRR.

methods, and thus consume more power. Triggercast improves the PRR of intermediate links from 50 to almost 100 percent with six concurrent senders. Our experiments indicate that Triggercast can *control network topology* (e.g., increasing new communication links) *without changing the original network state* (adding new nodes, increasing nodes' power, etc.). This characteristic is attractive to improve routing performance (explained in Fig. 3). To the best of our knowledge, we are the *first* to report multiple concurrent transmitters can reach such PRR improvements in realistic WSNs.

6.3 Data Forwarding with Triggercast

We create a five node topology as in Fig. 3b. We select two nodes as the source node and the receiver node respectively. The other three nodes perform forwarding. All five nodes are placed in random positions in our office. One of the three relay nodes is deployed near the window, exposure to sunshine. We also use a hair dryer to heat the node to increase the DCO jitters and decrease its link quality. We first measure pairwise loss rates between the nodes to compute the ETX metric for each link. We also evaluate the propagation and radio processing delays with CLS algorithm. We fix Tx power of source node as 0 dBm, online adjust Tx power of relay nodes from -25 to 0 dBm, and record the PRR of the receiver. Experimental results are elaborated in Fig. 20.

As expected, exploring CLS and LSA together results $1.3\times$ PRR gains on average over traditional single-path routing. The gain of using CLS alone is not obvious. The reason is because the dirty link (the heated node/receiver pair) influences the overall performance. Glossy works even worse than single-path routing. The reason is CI in
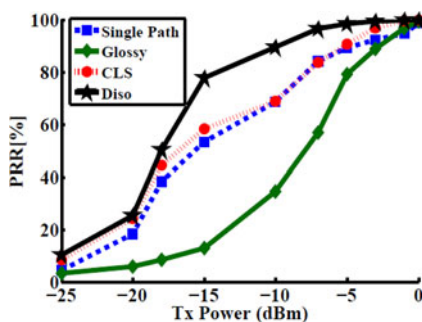
Glossy is non-destructive, and it doesn't compensate propagation and radio processing delays, as well as make link selections.

7   CONCLUSIONS

Concurrency helps to improve network performance. Following that direction, we in this paper propose Triggercast, the *first* work to implement Disco rather than CI in WSNs. Triggercast compensates propagation and radio processing delays, and makes link selection as well as transmission alignment, in order to construct CI. We implement Triggercast in real testbeds, and experimentally demonstrate that Triggercast produces significant performance gains in data forwarding protocols. We also provide a theoretical sufficient condition on how to ensure concurrent transmissions interfere constructively.

REFERENCES

[1]   P. Dutta, S. Dawson-Haggerty, Y. Chen, C. 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. Sens. Syst.*, 2010, pp. 1–14.

[2]   F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with Glossy," in *Proc. ACM/IEEE 10th Int. Conf. Inf. Process. Sens. Netw.*, 2011, pp. 73–84.

[3]   J. Lu and K. Whitehouse, "Flash flooding: exploiting the capture effect for rapid flooding in wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2009, pp. 2491–2499.

[4]   X. Wang, L. Fu, and C. Hu, "Multicast performance with hierarchical cooperation," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 917–930, Jun. 2012.

[5]   T. Jing, X. Chen, Y. Huo, and X. Cheng, "Achievable transmission capacity of cognitive mesh networks with different media access control," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 1764–1772.

[6]   S. Chu and X. Wang, "Opportunistic and cooperative spatial multiplexing in MIMO ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 5, pp. 1610–1623, Oct. 2010.

[7]   S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *Proc. ACM Conf. Appl., Technol., Architectures, Protocols Comput.*, 2007, pp. 397–408.

[8]   D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *Proc. ACM 14th Int. Conf. Mobile Comput. Netw.*, 2008, pp. 339–350.

[9]   S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2008, pp. 159–170.

[10] K. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous MIMO networks," in *Proc. ACM SIGCOMM Conf.*, 2011, pp. 146–157.

Fig. 20. Triggercast vs. traditional single-path routing.

[11] A. Gudipati, S. Perreira, and S. Katti, "AutoMAC: Rateless wireless concurrent medium access," in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 5–16.

[12] M. Jain, J. Choi, T. Kim, D. Bharadia, S. Seth, K.Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, real-time, full duplex wireless," in *Proc. ACM 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 301–312.

[13] Y. Yan, P. Yang, X. Li, Y. Tao, L. Zhang, and L. You, "ZIMO: Building cross-technology MIMO to harmonize ZigBee smog with WiFi flash without intervention," in *Proc. ACM 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 465–476.

[14] K. Leentvaar and J. Flint, "The capture effect in FM receivers," *IEEE Trans. Commun.*, vol. 24, no. 5, pp. 531–539, May 1976.

[15] N. Santhapuri, J. Manweiler, S. Sen, R. Choudhury, S. Nelakuditi, and K. Munagala, "Message in message (MIM): A case for reordering transmissions in wireless networks," in *Proc. ACM 7th Workshop Hot Topics in Netw.*, 2008.

[16] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. ACM 5th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, 1999, pp. 151–162.

[17] M. Doddavenkatappa, M. C. Chan, and B. Leong, "Splash: Fast data dissemination with constructive interference in wireless sensor networks," in *Proc. 10th USENIX Conf. Networked Syst. Des. Implementation*, 2013, pp. 269–282.

[18] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: a distributed wireless architecture for exploiting sender diversity," in *Proc. ACM SIGCOMM Conf.*, 2010, pp. 171–182.

[19] N. Oh and S. Lee, "Building a 2.4-Ghz radio transceiver using IEEE 802.15.4," *IEEE Circuits Devices Mag.*, vol. 21, no. 6, pp. 43–51, Jan./Feb. 2006.

[20] Y. Wang, Y. He, X. Mao, Y. Liu, Z. Huang, and X. Li, "Exploiting constructive interference for scalable flooding in wireless networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2104–2112.

[21] T. Schmid, P. Dutta, and M. B. Srivastava, "High-resolution, low-power time synchronization an oxymoron no more," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, 2010, pp. 151–161.

[22] S. Liu, G. Xing, H. Zhang, J. Wang, J. Huang, M. Sha, and L. Huang, "Passive interference measurement in wireless sensor networks," in *Proc. IEEE 18th Int. Conf. Netw. Protocols*, 2010, pp. 52–61.

Yin Wang (M'08) received the BS and MS degrees both from the Electronic Engineering Department of Tsinghua University, in 2004 and 2007, respectively. He is currently working toward the PhD degree in the Computer Science Department at Tsinghua University, Beijing, China. His research interests include wireless sensor networks, internet of things, mobile computing, and distributed simulation. He is a member of Tsinghua National Lab for Information Science and Technology. He is a member of the IEEE.

Yunhao Liu (SM'06) received the BS degree in automation from Tsinghua University, China, in 1995, and the MS and PhD degrees in computer science and engineering both from Michigan State University, in 2003 and 2004, respectively. He is currently the Cheung Kong professor at Tsinghua University. His research interests include RFID and sensor network, the Internet, and pervasive computing. He is a senior member of the IEEE.

Yuan He (M'07) received the BS degree from the University of Science and Technology of China, the MS degree from the Institute of Software, Chinese Academy of Sciences, and the PhD degree from the Hong Kong University of Science and Technology. He is currently an assistant professor in the School of Software, Tsinghua University and the vice director of the IOT-Tech Center, TNLIST. His research interests include sensor networks, peer-to-peer computing, and pervasive computing. He is a member of the IEEE.

Xiang-Yang Li (SM'09) received the BS degree from the Department of Computer Science and a bachelor's degree from the Department of Business Management, Tsinghua University, China, both in 1995, and the MS and PhD degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, in 2000 and 2001, respectively. He is a professor at the Illinois Institute of Technology. His research interests include the mobile computing, cyber physical systems, wireless networks, security and privacy, and algorithms. He is a senior member of the IEEE.

Dapeng Cheng received the BS degree from the Shandong University of Technology and the MS degree from the Dalian University of Technology. He is currently working toward the PhD degree in the Shandong University of Science and Technology, Qingdao, China. He is a member of Key Laboratory of Intelligent Information Processing in Universities of Shandong (Shandong Institute of Business and Technology). His research interests include wireless sensor networks, internet of things, mobile computing, and distributed simulation. He is a student member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.