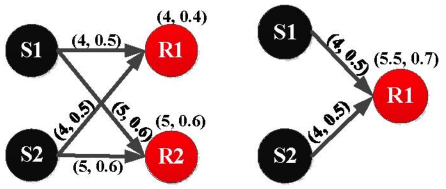
TriggerCast: Enabling Wireless Constructive Collisions

Yin Wang[†], Yuan He[†§], Dapeng Cheng[*], Yunhao Liu[†§], Xiang-yang Li[‡†]
[†]MOE Key Lab for Information System Security, School of Software, TNLIST, Tsinghua University
[§]Department of Computer Science and Engineering, HKUST
[*]Key Laboratory of Intelligent Information Processing in Universities of Shandong
(Shandong Institute of Business and Technology)
[‡]Department of Computer Science, Illinois Institute of Technology
{wangyin, he, yunhao}@greenorbs.com, chengdapeng@gmail.com, xli@cs.iit.edu

Abstract—**Constructive Interference (CI) proposed in the existing work (e.g., A-MAC [1], Glossy [2]) may degrade the packet reception performance in terms of Packet Reception Ratio (PRR) and Received Signal Strength Indication (RSSI). The packet reception performance of a set of nodes transmitting simultaneously might be no better than that of any single node transmitting individually. In this paper, we redefine CI and propose TriggerCast, a practical wireless architecture which ensures concurrent transmissions of an identical packet to interfere constructively rather than to interfere non-destructively. CI potentially allows orders of magnitude reductions in energy consumption and improvements in link quality. Moreover, we *for the first time* present a theoretical sufficient condition for generating CI with IEEE 802.15.4 radio: concurrent transmissions with an identical packet should be synchronized at chip level. Meanwhile, co-senders participating in concurrent transmissions should be carefully selected, and the starting instants for the concurrent transmissions should be aligned. Based on the sufficient condition, we propose practical techniques to effectively compensate propagation and radio processing delays. TriggerCast has 95^{th} percentile synchronization errors of at most 250ns. Extensive experiments in practical testbeds reveal that TriggerCast significantly improves PRR (from 5% to 70% with 7 concurrent senders, from 50% to 98.3% with 6 senders) and RSSI (about 6dB with 5 senders).**

I. INTRODUCTION

In wireless Sensor Networks (WSNs), it is widely accepted that simultaneous transmissions will result in packet collisions. Recently, A-MAC [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 basically operate on the passive side. In other words, they enable simultaneous transmissions to interfere non-destructively, namely to generate Non-Destructive Interference (NDI), in order to enhance network concurrency. Unfortunately, the packet reception performance of NDI might be no better than that of any single node transmitting individually (Fig. 1(a)), indicating NDI might degrade the performance of packet reception.



(a) Non-destructive interference (b) Constructive interference

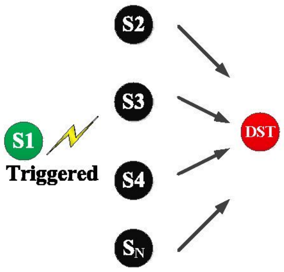
Fig. 1. Both NDI and CI enable concurrency. Only CI improves RSSI and PRR. Here, we use (a, b) to describe a link, while a and b represent the RSSI and PRR respectively.

Our work advances the technique by actively utilizing the capacity of Constructive Interference (CI) to potentially improve the received power and link quality (Fig. 1(b)). CI is especially attractive for WSNs, because it potentially improves energy efficiency, and thus mitigates the limited power supply issue. 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 Signal Noise Ratio (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 power required by a single sender. Moreover, simultaneously forwarding a packet can harness signal superposition gain, to improve Received Signal Strength Indication (RSSI) and Packet Reception Ratio (PRR).

However, implementing CI in WSNs is challenging due to the following reasons. First, simultaneous transmissions must be synchronized at the chip level, namely $0.5\mu s$ for IEEE 802.15.4 radio. To generate NDI, 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 not sufficient to construct CI. The Propagation delays and the radio processing delays significantly influence CI generation. Even worse, estimating the radio processing delays is an especially challenging task, as it varies from packet to packet, depends on the SNR, and is affected by 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 CI. The reason is because a radio signal has noise. Although signals are exactly aligned, noises also superpose. Whether SNR of the combined signal increases depends on SNRs and the Tx powers of individual signals.

Third, sensor nodes are always battery-powered, and have limited computational resources. It is difficult or even impossible to deploy complex signal processing algorithms in


S2
S3
S1
Triggered
DST
S4
S$_N$

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 reaches the receiver, $T_c(=0.5\mu)$s is the duration of a chip in IEEE 802.15.4 radio. Let λ_i be the 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 $\lambda_i = \frac{P_i}{N_i}$. Let $S_R^1(t)$ be the strongest signal. Therefore, we have $A_1 = 1$, $\tau_1 = 0$, $P_i = P_1 {A_i}^2$. According to [14], it can be derived that the effective power $\overline{P}$ of superposed signals after demodulation is $\bar{P} = P_1(\sum_{i=1}^{N} A_i \cos(\omega_c \tau_i))^2$, while the aggregated power of noise $\overline{N}$ is $\sum_{i=1}^{N} \frac{P_i}{\lambda_i}$. As a result, the SNR of the received superposed signal is

$$\frac{\overline{P}}{\overline{N}} = \frac{P_1(\sum\limits_{i=1}^{N} A_i \cos(\omega_c \tau_i))^2}{\sum\limits_{i=1}^{N} P_i/\lambda_i} \leq \frac{P_1 \sum\limits_{i=1}^{N} A_i^2 \sum\limits_{i=1}^{N} (\cos(\omega_c \tau_i))^2}{P_1 \sum\limits_{i=1}^{N} A_i^2/\lambda_i}. \quad (3)$$

The inequality (3) can be derived by Cauchy-Schwarz inequality **and equality holds if the condition satisfies**

$$\frac{A_i}{\cos(\omega_c \tau_i)} = \frac{A_j}{\cos(\omega_c \tau_j)}, \quad (\forall i, j). \quad (4)$$

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 $\lambda_{\max}$

$$(\frac{\overline{P}}{\overline{N}})_{max} > \lambda_{\min} \sum_{i=1}^{N} (\cos(\omega_c \tau_i))^2 \geq \lambda_{\max}. \quad (5)$$

Consequently, we derive a theoretical ***sufficient condition (SC)*** for CI with IEEE 802.15.4 radio.

i) Concurrent transmissions with a same packet should be synchronized at chip level, namely less than T_c=0.5μs;

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

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

IV. TRIGGERCAST IMPLEMENTATION

A. Chip Level Synchronization (CLS)

Eliminating the propagation delays and the radio processing delays in realistic environment is very challenging. Those **delays vary from one packet to another, and are influenced** by communication link qualities, asynchronous radio clocks, clock drifts as well as quantization errors. Fortunately, according to the *law of large numbers*, we can obtain the expected propagation and radio processing delays by a large number of trials. We select one transmitter-receiver pair which is 40 meters away in a indoor environment, and let the transmitter periodically send a packet every 500ms. Once the receiver successfully decodes a packet, it piggybacks a reply as soon as possible to the previous transmitter. As shown in Fig. 4, 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 as well as begins a packet transmission respectively. The TMote Sky node can accurately capture the exact instants when MCU detects rising edge and 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

$$\widehat{\Delta} = \frac{(\widehat{T_{S3}} - \widehat{T_{S1}}) - (\widehat{T_{R3}} - \widehat{T_{R1}})}{2}, \quad (6)$$

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

Experimental results of delay measurement using Eq. (6) is displayed in Fig. 5 as the 'raw' curve. Unfortunately, **the result is not sufficiently accurate. The measured delay** ranges from 0.596μs to 5.01μs, with average value 2.32μs and variance 0.628μs. The instability of measured delay indicates **that it is difficult to synchronize different transmitters at a** magnitude of 0.5μs, if we straightly use the measured data for compensation. Fortunately, we disclose the data transmission delay is the same for all nodes. And thus we have $T_{S2}(n) - T_{S1}(n) = T_{R2}(n) - T_{R1}(n)$. The data transmission delays of the transmitter and the receiver are drawn in Fig. 6.

We also find that the measured data transmission delays are not stable for the transmitter-receiver pair. The reason for the instability is because of the jitters, clock drifts as well as hardware diversities of the nodes' DCOs. The drifts can be as high as 5000ppm in our measurement. We de**fine** $\chi(n) = (T_{S2}(n) - T_{S1}(n))/(T_{R2}(n) - T_{R1}(n))$ **as the unified clock drift coefficient relative to the receiver. Consequently,** we can calibrate Eq. (6) as

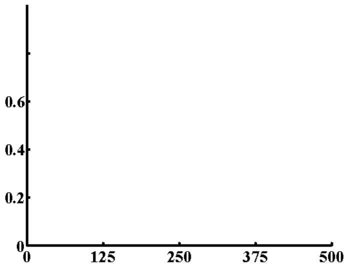
$$\widehat{\Delta}_{cal} = \frac{(\widehat{\frac{T_{S3}(n) - T_{S1}(n)}{\chi(n)}}) - (\widehat{T_{R3}} - \widehat{T_{R1}})}{2}. \quad (7)$$

We obtain the expected radio processing and propagation delay represented by DCO Ticks after the calibration of Eq. (7). To translate them to time, we also utilize the *Virtual High-resolution Time (VHT)* [17] approach, 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. 5. The calibrated delay ranges from 3.66μs to 4.12μs, with average value 3.90μs and variance 0.012μs. 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.**

B. Link Selection and Alignment (LSA)

Assuming all the concurrent transmissions are synchronized **at the chip level with CLS, according to the proposed sufficient** condition in Section III, the problem to make concurrent transmissions superpose constructively can be formalized as CI-generation problem.

Problem: Let $\Phi = \{L_1, L_2, ..., L_N, L_i = (P_i, \lambda_i)\}$ **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**


0.6
0.4
0.2
0
0
125
250
375
500