ALIGNER: Make the Utmost of Transmission Concurrency for Low Power Wireless Networks

Daibo Liu*, Zhichao Cao†, Mengshu Hou*
*School of Computer Science and Engineering, University of Electronic Science and Technology of China
†School of Software, Tsinghua University
{dbliu.sky, mshou}@uestc.edu.cn, caozc@mail.tsinghua.edu.cn

Abstract

Concurrent transmission (CT) is widely used to increase the throughput of various data transmissions in wireless networks, such as bulk data dissemination and high-rate data collection. In CT, besides the possible data frame collision at receivers, we observe that acknowledgement (ACK) collision at senders can significantly diminish concurrency opportunities.

In this paper, to avoid the potential ACK collision in CT, we propose ALIGNER which develops a new transmission pattern to coordinate concurrent senders in distributed manner. To avoid the collision between ACKs and data frames, we align the end of the data frames concurrently transmitted by several senders. ALIGNER adopts a tailor-made metrics to analyze the throughput benefit of concurrent transmission for data collection protocols. We have implemented ALIGNER in TinyOS and conducted extensive experiments on a real testbed. Experimental results show that ALIGNER can significantly increase the concurrency opportunities compared with the state-of-the-art CT methods.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Experiment, Measurement, Performance

Keywords

Internet of Things, Duty cycled networks, Transmission concurrency

1 Introduction

The fast development of Internet-of-Things (IoT) [1] boosts the explosive growth of energy-constrained wireless devices. Low power wireless networks [33] becomes a major approach to connect the numerous devices into the Internet. Lora [39] and SigFox [38] can provide long-range wireless communication with infrastructure mode. A client device connect to Internet through a gateway. The data rate of a client is usually very low to ensure energy efficiency and extend the lifetime. Meanwhile, ad-hoc wireless networks (e.g., wireless sensor/mesh networks) is an alternative way to achieve large scale and low power IoT deployment. The inherent data rate is relatively larger than LoRa. However, it is difficult to achieve high network throughput due to the long back-off time incurred by severe signal interference.

Concurrent transmission (CT) is a promising apporach to increase network throughput of various data transmissions, such as bulk data dissemination [43] [44] [41][36] and high-rate data collection [25][26][11]. CT measures data delivery performance of each link under mutual interference[1] , then exploits potential opportunities to concurrently transmit data packets in the presence of "exposed terminal". The effectiveness of CT depends on the success of both data frame delivery and acknowledgement (ACK) reception. Existing methods [36][24][25][41][37][26][11][23][3] capture the successful CT opportunities by either measuring the bidirectional link quality [25][11] or adopting delayed/windowed response [6][36] to avoid ACK loss which cannot completely avoid the ACK collision. However, the ACK collision problem can significantly diminish concurrency opportunities.

This problem gets worse in low power wireless networks where nodes work in duty-cycled mode and each sender occupies the channel to transmit each data for a long time. Take a widely used power management protocol, Low Power Listening (LPL) [4] as illustrated in Figure 1, as an example. In low power wireless networks, to transmit a packet, a sender needs to continuously transmit the same packets (called *data frame*) until the receiver's ACK is received or a pre-configured timer at the sender expires. A fixed time interval between successive data frames is set to wait for potentially oncoming ACK. The insistent channel access of LPL increases the probability of mutual interference between concurrent senders, bringing about severe negative effect on ACK decoding. To assess the feasibility of concurrent transmission, existing techniques have to suffer the disappointing ACK reception so that miss potential opportunities that the

[1] Mutual interference denotes the transmission signal of concurrent senders interferes with packet decoding at receivers and ACK decoding at the senders.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2019
25–27 February, Beijing, China

ISBN: 978-0-9949886-3-8

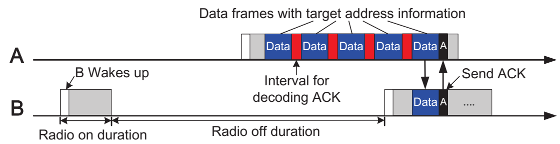


Figure 1. The basic mechanism of packet transmission in low power wireless sensor networks.

concurrently transmitted packets can be successfully delivered to receivers.

In this paper, we explore a new transmission pattern to eliminate the effect of ACK collision problem in concurrent transmission. The fundamental insight is that as long as concurrent senders can simultaneously complete their data frame transmissions, the shared channel will be completely clear during the subsequent short period. In that period, senders can decode the arriving ACK(s) without suffering from the mutual interference because all concurrent senders are waiting for their receivers' ACKs. To achieve this goal, we propose ALIGNER to coordinate the data transmissions of concurrent senders in distributed manner. ALIGNER can accurately measure the time of each transmission phase (such as transmitting preparation time, data frame on-air time, and the processing time of each received data frame) between two successive data frames. Once capturing a frame receiving event, ALIGNER immediately calculates the next data frame's transmitting time of the ongoing sender and aligns its transmitting time with the ongoing one. As a result, two senders will simultaneously complete their data frame transmissions. Moreover, we employ a tailor-made metric which was proposed in COF [25] to analyze the feasibility of concurrent transmission. We implemented ALIGNER in TinyOS-2.1.1 [22] and evaluated its performance through extensive indoor testbed experiments.

Overall, this paper makes the following contributions:

- We propose a new concurrent transmission pattern, ALIGNER, to avoid the potential ACK collision in CT. As far as we know, this is the first practical and effective method for fully avoiding ACK collision in concurrent transmission (Section 3).
- By eliminating the effect of mutual interference on ACK decoding, we develop enhanced metrics to analyze the feasibility of concurrent transmission (Section 4).
- We have implemented ALIGNER in TinyOS-2.1.1 [22], and integrated it with data forwarding protocols (CTP [13] and LPL [33]). The extensive indoor testbed evaluation results demonstrate that ALIGNER can significantly improve transmission opportunities (Section 6).

This paper is organized as follows. In Section 2 we present the empirical studies to justify the necessity of ALIGNER. The detailed design of ALIGNER is presented in Section 3. We further employ tailor-made metrics to analyze the feasibility of concurrent transmission in Section 4. The implementation problems and evaluation results are presented in Section 5 and 6, respectively. Section 7 discusses the related work. We conclude this paper in Section 8.

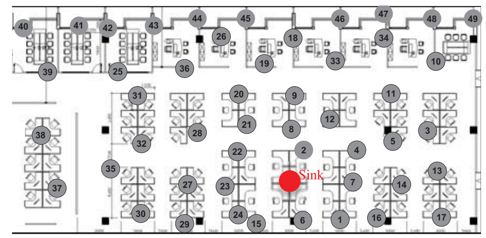


Figure 2. Indoor testbed with 50 Telosb nodes deployed on our 40×70m^2 office. The red node denotes sink node.

2 Empirical Study

In this section, we conduct empirical studies to demonstrate that:

- ACK collision can seriously diminish transmission opportunities in CT.
- The simultaneous completion of senders' data frame transmissions can fully eliminate the effect of mutual interference on ACK decoding.

2.1 Low Power Listening

As illustrated by Figure 1, low power listening (LPL) is a common MAC-layer technique for reducing energy consumption in low power wireless ad-hoc networks. With LPL, nodes asynchronously and periodically wake up to sample wireless channel to detect activity. In a low power ad-hoc network with extremely low traffic, LPL can make sensor nodes keep in silent state as much as possible for saving energy. While in burst traffic scenario and dynamic environment, LPL can provide much reliable data delivery service compared with Low Power Probing (LPP) [9] mechanism. Furthermore, LPL can be seamlessly adopted by various types of data forwarding protocols, such as deterministic [13] and opportunistic [21] forwarding protocols. Up to now, LPL has been widely adopted in practically deployed wireless sensor networks [30][10][29]. However, by adopting LPL to transmit a packet, since the wakeup schedule is usually asynchronous, a sender will keep occupying wireless channel until its packet is successfully delivered. If a neighboring node also has packet to transmit at this time, it should keep radio on to repeat the same carrier sense-defer process.

2.2 Quantification of Exposed Terminals

In this section, we first quantify the proportion of suppressed opportunities of concurrent transmission in exposed terminals. On this basis, we further show the effect of ACK collision on turning exposed terminal into transmission opportunities.

We conduct experiments to test the performance of data forwarding (denoted as DF) protocol in an indoor testbed with 50 Telosb nodes as illustrated by Figure 2. These nodes are controlled by a central computer through wireline connections. The performance of DF is evaluated by using the standard open source code of CTP [13]. CTP is built upon LPL.

For each experiment, we randomly select two nodes as senders by restricting that the two are within the carrier sense range of each other. The selected senders will immediately generate a new packet to transmit once the previous one has

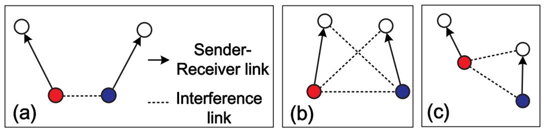
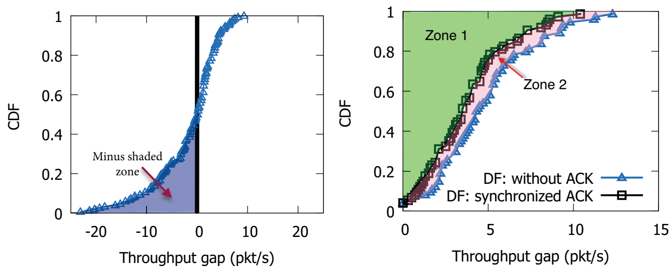


Figure 3. Different scenarios of data forwarding (a, b, and c).



(a) Quantified exposed terminals (b) Effect of ACK collision

Figure 4. The quantified exposed terminals in deterministic data forwarding (a) and the effect of ACK collision on harnessing exposed terminal (b).

been successfully delivered. Each of the rest of network nodes generates a packet every 10 minutes. Except that sink node's radio is set to always on, all other nodes are set to wake up every 512ms. The selected senders and their corresponding receivers form different structures as shown in Figure 3. Figure 3(a) denotes each sender's receiver is not interfered by the other sender, Figure 3(b) presents the opposite case, and Figure 3(c) denotes only one sender's receiver is interfered by the other sender. For each experiment, we repeat it by setting the two selected senders' network parameter configuration to each of the following four cases:

• Enabled CSMA and ACK required mechanism (marked as *csma_ack*): Contention-based forwarding;

• Disabled CSMA and ACK required mechanism (*nocsma_ack*): In concurrent transmission, a sender will not generate a new packet until the previous one has been acknowledged;

• Disabled CSMA and no ACK required mechanism (*nocsma_noack*): In concurrent transmission, each data frame is assigned with a different data sequence number (DSN) and receivers are not required to reply ACK;

• Disabled CSMA and synchronized data frame transmission (*nocsma_syn_ack*): In concurrent transmission, senders are synchronized to simultaneously complete their data frame transmissions and receivers are required to reply ACK. Central controller synchronizes senders' data frame transmissions with negligible error.

Except for the two selected senders, the other nodes' CSMA mechanism is enabled always. We repeat 150 experiments by selecting different senders each time. By setting the senders' network parameter to each one of the above four cases, each receiver removes duplicate packets and reports the number of received packets to central computer through wireline connection. According to the duration of each experiment, we compute the average throughput (packets per second that is marked as pkt/s) of the four cases and mark them as T_{csma_ack}, T_{nocsma_ack}, T_{nocsma_noack}, and $T_{nocsma\_syn\_ack}$, respectively. Then we further calculate the throughput gap (T^{ET}_{gap}) between *csma_ack* and *nocsma_noack* according to

$$T^{ET}_{gap} = T_{csma\_ack} - T_{nocsma\_noack}. \tag{1}$$

The gap denotes throughput difference when CSMA is enabled and concurrent transmission is fully adopted in the same experimental scenario. For the latter of Eqn. 1, the effect of ACK collision on data delivery is completely eliminated because each sender immediately transmits the next packet without waiting for the previous packet's ACK. We plot the CDF of the 150 T^{ET}_{gap}s in Figure 4(a). As shown in the figure, to transmit packet by disabling the CSMA mechanism and completely eliminating the effect of ACK collision on data delivery, a great portion of transmissions can achieve higher throughput (see the minus shaded zones of Figure 4(a)) than the case with CSMA-enabled. It means there are many data transmissions (i.e. 49% in this experiment) suffering from exposed terminal problem.

The experimental results indicate there are still great potential opportunities for concurrent transmission in low power ad-hoc networks. Existing concurrent transmission approaches [25][36][11] only harness a very limited part of exposed terminals because they require both packet delivery and ACK reception are likely to succeed. However, if concurrent senders interfere with each other, existing approaches fail to exploit these opportunities. As a matter of fact, successful packet delivery is the ultimate aim of forwarding protocol, and an ACK is just a sign of successful data delivery. Retransmission caused by ACK loss directly hurts performance and will pay a big price to recover it. Next, we quantify the effect of ACK collision on performance, and discuss the feasibility of ACK collision avoidance in concurrent transmission for low power ad-hoc networks.

2.3 Effect of ACK Collision on Harnessing Exposed Terminals

For each experimental scenario corresponding to the minus shaded zones of Figure 4(a), we use two additional experimental results (*nocsma_ack* and *nocsma_syn_ack*) to further calculate the throughput gap, T^{noACK}_{gap}, between *nocsma_noack* and *nocsma_ack*, and the throughput gap, T^{synACK}_{gap}, between *nocsma_syn_ack* and *nocsma_ack* according to the following two formulas,

$$T^{noACK}_{gap} = T_{nocsma\_noack} - T_{nocsma\_ack},$$

$$T^{synACK}_{gap} = T_{nocsma\_syn\_ack} - T_{nocsma\_ack},$$

where T^{noACK}_{gap} denotes the throughput improvement by completely eliminating the effect of ACK collision on data delivery in concurrent transmission, and T^{synACK}_{gap} denotes the throughput improvement by eliminating the effect of mutual interference on ACK decoding in concurrent transmission. We plot the CDF of computed T^{noACK}_{gap}s (marked as *without ACK*) and T^{synACK}_{gap}s (marked as *synchronized ACK*) in Figure 4(b).

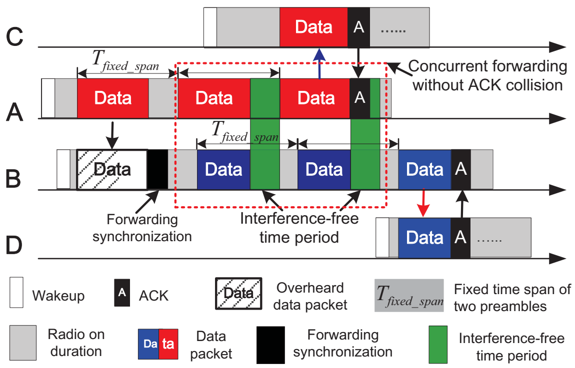


Figure 5. Overview design of ALIGNER. ***A*** **and** ***B*** **are within the carrier sense of each other.** ***C*** **and** ***D*** **are their potential receivers respectively. By synchronizing,** ***B*** **and** ***A*** **simultaneously complete their data frame transmission.**

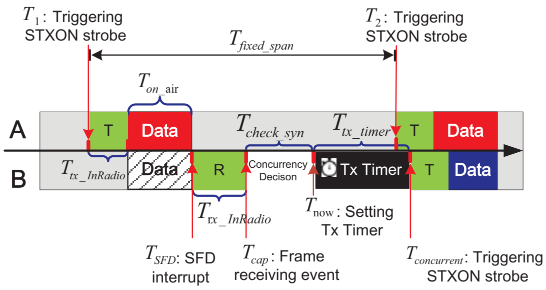


Figure 6. Zooming in to the events timeline of synchronization process in ALIGNER. By capturing the SFD interruption at T_{SFD} and frame receiving event at T_{cap}, B calculates T_2 that A will transmit the next data frame.

As shown in the figures, the vast majority of potential concurrent transmission opportunities suffer from ACK collision resulting from mutual interference. By synchronizing concurrent senders' data frame transmission to eliminate the effect of mutual interference on ACK decoding, compared with existing concurrent transmission strategies (with the same network parameter of *nocsma_ack*), there are more than 95% exposed terminals which can be fully turned into transmission opportunities. By synchronizing concurrent transmission (*nocsma_syn_ack*), the effectiveness of harnessing exposed terminals is significantly improved, which is visually expressed as the green dash area (*Zone 1*).

Note that except for the mutual interference, there is another factor affecting ACK decoding at concurrent senders: ACK loss due to the collision of multiple incoming ACKs or the burstness of wireless link. Hence, when concurrent transmission protocol is not required to acknowledge each received packet (*nocsma_noack*), it can achieve the optimal throughput compared with both *nocsma_ack* and *nocsma_syn_ack*. The gap between *nocsma_noack* and *nocsma_syn_ack* denotes the effect of the second factor on ACK decoding (see *Zone 2* in Figure 4(b)). *Zone 2* is smaller than *Zone 1* because the probability that multiple ACKs simultaneously arrive at the senders in low power ad-hoc networks is extremely small.

The empirical studies shed light on the existence of great potential transmission opportunities in the presence of exposed terminals and the serious effect of ACK collision on turning exposed terminals into transmission opportunities. Furthermore, the experimental results provide key insights we use to address this problem: synchronized concurrent transmission for simultaneously completing senders' data frame transmissions can effectively eliminate the effect of mutual interference on ACK decoding. Additionally, the results prompt us to further resolve ACK collision when multiple ACKs arrive at senders simultaneously. In the next section, we introduce the detailed design of ALIGNER to harness exposed terminals for data forwarding protocols.

3 Design of ALIGNER

In this section, we first give an overview of ALIGNER. Then, we introduce the detailed design of ALIGNER.

3.1 Design Overview

The overview of ALIGNER is illustrated in Figure 5. Node *A* and *B* are within the carrier sense range of each other. They are transmitting packet to *C* and *D*, respectively. When *B* overhears a data frame from *A*, it first confirms whether concurrent transmission is benefit according to an *enhanced forwarding metric* introduced in Section 4. If concurrent transmission is benefit for network performance, by checking the data frame size of *A* and itself, *B synchronizes* the schedule of its own data frame transmission to *A*, namely they simultaneously complete their data frame transmissions.

When the schedule of data frame transmission is determined, *B* periodically transmits the same data frame until *D*'s acknowledgement (ACK) is received. The cycle of each data frame transmission is set to a fixed time span (T_{fixed_span}) in ALIGNER. By repeatedly transmitting the same data frame, *B* disables CSMA mechanism to guarantee invariable time span between two successive data frame transmissions.

When two concurrent senders simultaneously complete their data frame transmissions, in the following time (a few milliseconds), the senders will not interfere with each other because both of them are "silent" to wait for the potentially oncoming ACK(s). During the interference-free period as shown in Figure 5, *A* successfully decodes an ACK replied by *C*. When *A* completes its data forwarding, *B* continues to forward its data frame according to the pre-set data frame span and disables CSMA mechanism until *D* receives its data frame and replies an ACK to *B*.

By adopting ALIGNER, a new sender should overhear a full data frame from ongoing sender to make a correct transmission decision. Compared with traditional data forwarding mechanism, if the shared channel is not free, rather than keeping radio on and repeating the same carrier sense-defer process to obtain access to the channel, a new sender can decide whether it can quickly access the busy channel once it overhears an ongoing data frame. Compared with the energy consumption for idle waiting, the slight energy consumption for overhearing an ongoing data frame is cost-effective. In the following section, we present the design of ALIGNER in detail.

3.2 Synchronization Principle

The detailed events timeline of synchronization process is given in Figure 6. A is the ongoing sender. T_{fixed_span} equals the time span between two successive STXON strobes corresponding to T_1 and T_2, respectively. STXON strobe is hardware instruction to start data transmission. In this period, A first triggers STXON at T_1 to start data frame transmission. After a fixed period of time ($T_{tx\_InRadio}$), the signals of data frame are emitted by antenna. During the period of $T_{tx\_InRadio}$, bit stream of the data frame is continuously encoded and modulated. The data frame transmission will last for a certain time, which is called data frame on-air time and denoted as T_{on_air}.

In the meantime, B demodulates and decodes the arriving signals. As shown by Figure 7, in receive mode, SFD pin of radio chip [17] goes high after the start of frame delimiter (SFD) field has been completely received. SFD pin goes low again only after the last byte of data frame has been received. Hence, in Figure 6, when A finishes the transmission of the last byte, B (almost) simultaneously generates SFD interrupt at T_{SFD}. By successfully decoding the data frame and conducting the CRC verification, after almost a fixed period of time (marked as $T_{rx\_InRadio}$ in the figure), network system signals a data frame reception event at time point T_{cap}. By capturing the event, ALIGNER uses the collected information to analyze the feasibility of concurrent transmission according to the metric presented in Section 4. The analytical time is marked as T_{check_syn} . If network performance cannot get benefit from concurrent transmission, B defers the transmission of pending packet and repeats the carrier sense-defer process. Otherwise, ALIGNER grabs the current time (marked as T_{now}) and calculates the time of the next data frame transmission (marked as $T_{concurrent}$), which is introduced in the next section in detail. Then, it immediately activates a *Tx Timer* (as shown in Figure 6). The timer will fire at $T_{concurrent}$. We mark the timer interval from T_{now} to $T_{concurrent}$ as T_{tx_timer}. When *Tx Timer* expires, B immediately triggers STXON strobe to concurrently transmit its data frame with A. Because ALIGNER disables CSMA mechanism in repeatedly transmitting the same data frame, both A and B will simultaneously complete their data frame transmissions, even though their data frames may be in different sizes. Hence, the timeline can be divided into five successive parts: $T_{tx\_InRadio}$, T_{on_air}, $T_{rx\_InRadio}$, T_{check_syn}, and T_{tx_timer}. Because capturing SFD signal is generic approach for various wireless communication, the strategy mentioned above for accurate timing is easy to accomplish and applicable to various application scenarios.

3.3 Data Frame Synchronization

As illustrated in Figure 6, by capturing SFD interruption at T_{SFD} and subsequently obtaining the successful packet reception event at T_{cap}, as a deferred sender, B knows the ID of the ongoing sender and the size (marked as m Bytes) of the corresponding packet. Taking T_{SFD} as the reference time, ALIGNER firsts calculates the relative time of T_1. Then, according to the fixed cycle of data frame transmission, ALIGNER further calculates the relative time of the next STXON strobe triggering time T_2. Finally, by considering the difference of packet size of A and B, ALIGNER

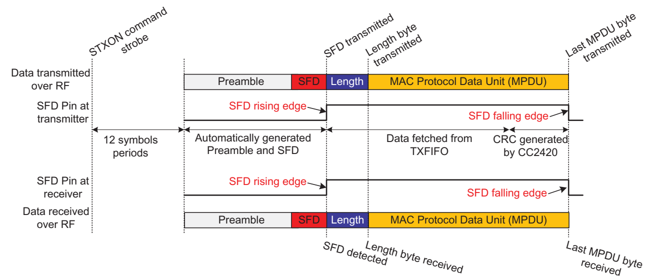


Figure 7. SFD pin activity examples during transmit and receive in CC2420.

schedules B's STXON strobe triggering time in order to let them simultaneously complete data frame transmissions.

Relative time of T_1. The time span between T_1 and T_{SFD} can be divided into two parts: $T_{tx\_InRadio}$ and T_{on_air}. $T_{tx\_InRadio}$ is the time span from triggering a transmitting command strobe to the time point when the first bit is begin to be emitted by antenna. $T_{tx\_InRadio}$ is a constant for each sensor node. In the widely applied radio chip CC2420 [17], $T_{tx\_InRadio}$ is equivalent to 12 symbols period as shown in Figure 7. T_{on_air} is determined by packet size (m Bytes) and radio data rate π (kbps). As shown in Figure 7, the packet contains the MAC protocol data unit, preamble header, SFD byte and length byte. Then, the on-air time of A's data frame can be expressed as

$$T_{on\_air} = \frac{8m}{\pi} ms. \tag{2}$$

Note that the precision of existing timer in sensor node can be less than millisecond. For example, the precision of Telosb [40] node is jiffy which is equal to one thirty-second ms. Hence, T_{gap_frame} can be accurately expressed as

$$T_{on\_air} = \frac{256m}{\pi} jiffy.$$

Then, we compute the relative time of T_1 corresponding to the reference time T_{SFD} by

$$T_1 = T_{SFD} - T_{on\_air} - T_{tx\_InRadio}. \tag{3}$$

Relative time of T_2. Based on the relative time of the previous STXON strobe triggered by A, ALIGNER further calculates the relative time for triggering the next STXON strobe. Because the time interval between two successive STXON strobes is constant in ALIGNER, T_2 can be directly expressed as

$$\begin{aligned} T_2 &= T_1 + T_{fixed\_span} \\ &= T_{SFD} - T_{on\_air} - T_{tx\_InRadio} + T_{fixed\_span}. \end{aligned} \tag{4}$$

Scheduled transmitting time. By analyzing the feasibility of concurrent transmission and computing the time point of A's next STXON strobe time, B should first calculate the on-air time gap, T_{gap_frame}, between on-air time of both A's and B's data frames. For simplicity, we mark the packet size of B as n Bytes. Then, T_{gap_frame} is calculated by

$$T_{gap\_frame} = \frac{(n-m)\times 8}{\pi} ms, \tag{5}$$

that equals $\frac{(n-m)\times 256}{\pi}$ jiffies. If T_{gap_frame} is positive, which means B's data frame size is larger than that of A, B should trigger its transmitting strobe ahead of A by T_{gap_frame} ms. Otherwise, later than A by T_{gap_frame} ms. Hence, B should trigger its packet transmitting strobe at the time point $T_{concurrent}$ that can be expressed as

$$T_{concurrent} = T_2 - T_{gap\_frame}. \tag{6}$$

By getting to know the time of triggering the STXON strobe of itself's data frame, B should activate a transmission timer (marked as *Tx Timer* in Figure 6) to expire at $T_{concurrent}$. By grabbing the current time T_{now}, ALIGNER calculates the timer interval T_{tx_timer} according to

$$T_{tx\_timer} = T_{concurrent} - T_{now}. \tag{7}$$

Once the timer expires, B should immediately trigger STXON strobe to transmit its data frame. Then, node A and B will concurrently transmit their data frames and simultaneously complete their transmissions. Note that ALIGNER exploits the captured SFD signal to achieve synchronization between neighboring senders. By getting the time of SFD signals of neighboring sender's data frame, ALIGNER can synchronize them immediately. Hence, the clocks drift between neighboring nodes can not affect the accuracy of temporary synchronization.

4 Concurrent Transmission Metrics

As mentioned above, if a sender has packet to transmit while the channel is busy, once it hears a data frame transmitted by the ongoing sender and captures the data frame reception event, it should analyze the feasibility of concurrent transmission according the adopted metric. Based on the inherent ACK collision avoidance ability of ALIGNER, in this section, we introduce an enhanced concurrent transmission metrics for data forwarding protocols.

Intuitively, concurrent transmission metric is used to quantify the benefit of concurrent transmission than forwarding in exclusive mode. In this paper, we define the benefit as the instantaneous throughput improvement which was proposed by COF [25]. To quantify the instantaneous throughput of concurrent transmission, it is necessary to measure each link's success probability of packet delivery. Under the condition of concurrent transmission, it is denoted as conditional packet delivery ratio (*cpdr*) by COF. In this paper, we will not introduce the mechanism of computing *cpdr* in detail because ALIGNER works above the link estimation layer and can use different link estimation modules.

Here, we define the *cpdr* as the probability, $P^k_{i,j}$, that receiver j can receive the packet transmitted by sender i when neighbor k is concurrently transmitting. $P^{\o}_{i,j}$ denotes the link quality from sender i to receiver j when there is no other ongoing sender. We assume each node has successfully measured the *cpdr* of each link. The introduction on how to compute cpdr is explained in Appendix A in detail.

In data forwarding, A and B are within the carrier sense range of each other. C is A's parent node (receiver) and D is B's parent node. Then A can compute conditional bidirectional link quality (CLQ) between A and C when B is concurrently transmitting according to

$$CLQ^B_{A,C} = P^B_{A,C} \times P^{\o}_{C,A}. \tag{8}$$

In the same way, B computes $CLQ^A_{B,D}$ according to

$$CLQ^A_{B,D} = P^A_{B,D} \times P^{\o}_{D,B}.$$

The computation of CLQ has considered the influence of concurrent transmission on receivers' data frame decoding, but the ACK decoding at each concurrent senders is considered to be free from mutual interference. Because ALIGNER can effectively eliminate the impact of mutual interference on ACK decoding, it is sensible and reasonable to compute CLQ with $P^{\o}_{C,A}$ (or $P^{\o}_{D,B}$).

Then, the benefit, $T_{benefit}$, of concurrent transmission for node B can be computed according to

$$T_{benefit}(B,A) = (CLQ^B_{A,C} + CLQ^A_{B,D}) - CLQ^{\o}_{A,C}, \tag{9}$$

where the former is the instantaneous throughput of both A and B by concurrently transmitting, and the latter denotes the instantaneous throughput of A when there is no concurrent transmission. If $T_{benefit}(B,A)$ satisfies the condition of

$$T_{benefit}(B,A) >= \omega, \tag{10}$$

B deems that it is better to concurrently transmit than to do carrier sense and defer. ω is a compensation value for the extra consumption (e.g., energy or extra transmission) of B's joining in. We will further discuss the impact of ω on performance in Section 5.

5 Implementation of ALIGNER

We have implemented ALIGNER based on LPL [4] in TinyOS-2.1.1 [22]. As we mentioned in Section 3 and 4, several implementation details are further discussed in this section, which include the determination of ALIGNER's parameters for synchronization of data frame transmission, determination of the value of ω, and the effect of synchronization error on performance. To begin with, we give the summary of several system parameters of ALIGNER in Table 1.

Table 1. Several useful parameters of ALIGNER.

Time span	value
$T_{tx\_InRadio}$	192 μs
T_{on_air}	[576, 4256]μs
T_{fixed_span}	8ms
$T_{turnaround}$	192μs
π	250kbps

5.1 Synchronization Parameters

As mentioned in Section 3, for ease of understanding, the synchronization process has been divided into five consecutive time spans: $T_{tx\_InRadio}$, T_{on_air}, $T_{rx\_InRadio}$, T_{check_syn}, and T_{tx_timer}. While in fact, only $T_{tx\_InRadio}$ and T_{on_air} have been used to compute the next data frame transmitting time. Note that ALIGNER can accurately calculate T_{on_air} by Eqn. 2 once ALIGNER successfully decodes the data frame of ongoing sender. Moreover, T_{tx_timer} is a constant value. For example, in CC2420 [17], the T_{tx_timer} equals 12 symbols period as shown in Figure 7. In other radio chips [16] [18], the value of T_{tx_timer} may be different due to the use of different coded

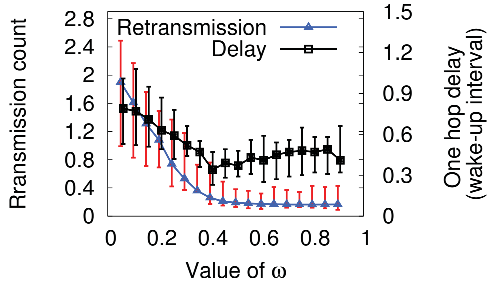


Figure 8. Effect of ω on performance.

modulation technology, but the invariance property fits to all radio chips.

In addition, although $T_{rx\_InRadio}$ is not directly used to compute the data frame transmitting time, it is necessary to measure its value for determining the accurate time stamp of SFD falling edge as shown in Figure 7. The time interval between T_{SFD} and T_{cap} is almost invariant.

5.2 Determination of Compensation Value

In Eqn. 10, the weight ω is a compensation value for the expected benefit of concurrent transmission. Generally, a larger ω could reduce the opportunities for concurrent transmission, but it also can reduce the retransmission rate because the concurrent senders can deliver their packets more reliably. On the other hand, assigning a relatively small value to ω could increase the transmission opportunities and increase retransmission rate caused by data collision, may leading to high transmission delay. Thus, assigning an appropriate value to ω is important for achieving high network performance, such as one-hop delay and retransmission rate. To get the optimal value of ω for data forwarding protocols, we conduct evaluations on the indoor testbed illustrated in Figure 2. We plot the average retransmission count and average one hop delay corresponding to different setting of ω in Figure 8. The experimental results plotted in the figures indicate that 0.4 is the optimal value for data forwarding protocol.

5.3 Effect of Synchronization Error on Performance

Note that the time stamp of SFD interruption (falling edge at Figure 7) at a deferred sender (e.g., node B in Figure 6) can accurately represent the specific time point when the ongoing sender (e.g., node A in Figure 6) will finish its data frame transmission. In the implementation of ALIGNER, the time interval of triggering data frame transmitting strobe ($T_{tx\_InRadio}$) is predefined by system manager and is invariant for all network nodes. Hence, the precision of system timer is the only influence on the accuracy of synchronization of ALIGNER. Taking our experimental platform as an example, the highest achievable accuracy of system timer is jiffy that is one thirty-second *millisecond* in the widely applied CC2420 radio chip. In this section, we conduct experiments to measure the distribution of synchronization error in ALIGNER by using Telosb mote that equipped with CC2420.

Table 2. CDF of synchronization error.

Synchronization error	⩽32μs	⩽64μs	⩽96μs
CDF	79%	97%	100%

As shown in Table 2, the synchronization error is distributed between 0 and 64μs. Note that TXRX turnaround time period ($T_{turnaround}$) is a necessary period before replying an ACK by receiver. The TXRX turnaround time period is significantly larger than the synchronization error even though auto-ack mechanism is adopted by which the TXRX turnaround time period is 192 μs. What's more, ALIGNER disables the auto-ack mechanism to achieve random back-off of ACK, and the turnaround period can up to 2 *ms*. In our experiments, the ACK loss caused by mutual interference due to the synchronization error has never happened. We are confident that this case will never happen if the turnaround period is far larger than synchronization error.

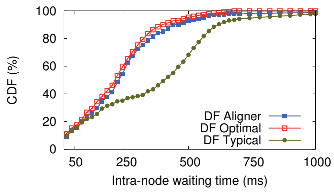
6 Evaluation

In this section, we conduct extensive evaluations to test the performance of ALIGNER from two aspects. First, we demonstrate the efficiency of ALIGNER in Section 2.2. Then, we evaluate the performance improvement of data forwarding protocols by using ALIGNER and comparing with the state-of-the-art concurrent transmission protocols in Section 6.3.

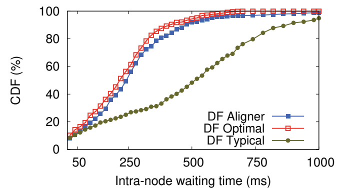
6.1 Experimental Setting and Performance Indicator

Our experiments are conducted in indoor testbeds with 50 Telosb nodes which are deployed on our 40×70m^2 office as shown in Fig. 2. By setting different transmission power levels (RF output power) to testbed networks, nodes automatically form multi-hop networks with different densities and topologies. All experiments are conducted in the 19th Zigbee wireless channel which is overlapped with part of WiFi operating frequency used by the office APs. All senders transmit 80-byte data packets in the experiments. The wake-up interval is set to 512ms. When a node wakes up, it has to perform a Clear Channel Assessment (CCA) to assess channel condition. This period is a constant about 11 ms. During this period, if sensor node detects a busy channel condition, it extends its radio-on period to receive potential incoming packets. The extended active period in TinyOS-2.1.1 is defaulted to 30 ms. Except 512ms, we have also evaluated the effect of different wake-up intervals on performance. The experimental results show the same conclusion as 512ms. All network nodes work in duty-cycled mode except sink node. When a node wakes up, if the shared channel is clear and the node has no data packet to transmit, it will keep in active state for 11ms, and then it turns off radio and returns to sleeping state. However, if it has data packet to transmit, it will keep in listening state to occupy shared channel. After the completion of data transmission, it returns to sleeping state by turning off radio. Note that external interference could cause node's active state to be extended even if the disturbed node has no data packet to transmit.

In the following sections, we use packet delivery ratio as the indicator of network reliability. The energy consump-



(a) Free from external interference



(b) Suffering from external interference

Figure 9. CDF of intra-node waiting time in experimental scenario where nodes are free from external interference (a) and suffer from the coexisting wireless interference (b).

tion is measured by duty cycle, the portion of radio-on time, as a platform-independent metric for energy efficiency. This metric is a good proxy for power, because typical sensor platforms have their power profile dominated by the radio chip and transmitting and listening operations commonly have a similar current draw. Besides, we use end-to-end delay to express delivery latency. The delivery latency is defined as the time duration from the time when a packet is put into the sender's transmission buffer to the time when the sink node receives the packet.

6.2 Efficiency of ALIGNER

To demonstrate the efficiency of turning exposed terminals into transmission opportunities by ALIGNER, we conduct experiments to reveal the performance gap between a widely approved concurrent transmission approach for DF protocol (denoted as DF-Typical) and DF-ALIGNER, and the performance gap between DF-ALIGNER and the best concurrent transmission approach (denoted as DF-Optimal) which is with the highest harnessing ratio of exposed terminals.

DF-Typical considers the ACK loss caused by mutual interference when it harnesses exposed terminals, and it has no synchronization mechanism and random back-off ACK mechanism. On the contrary, DF-Optimal completely ignores ACK loss during concurrent transmission (see below). For each of these experiments, we design an intuitive and controllable experimental scenario as shown in one of the Figure 3(*a*-*c*). In each specific scenario, we test DF-Typical, DF-ALIGNER, and DF-Optimal respectively. The wakeup interval is set to 512ms. Each sender will randomly generate a packet in its next wakeup interval once the previous packet is successfully delivered. The selected four nodes are all connected to a central computer to report each packet's generating time at two senders and report the successful data delivery time, e.g, the ACK receipt time at senders for DF-Typical and the data receipt time at receivers for DF-Optimal. Furthermore, because DF-Optimal disables the ACK mechanism, once a receiver receives a packet, the central computer should immediately notify the corresponding sender so that it can generate the next packet. The concurrent transmission metric presented in Section 4 is used for them to construct concurrent decision, but DF-Typical should further consider the ACK loss caused by mutual interference in Equation 8.

Table 3. The harnessing ratio of exposed terminals.

Protocols	DF-ALIGNER	DF-Typical
Harnessing ratio	71%	34%

The network overhead for constructing concurrent decision in the initial stage is filtered out from the experimental results.

6.2.1 *Intra-node Waiting Time*

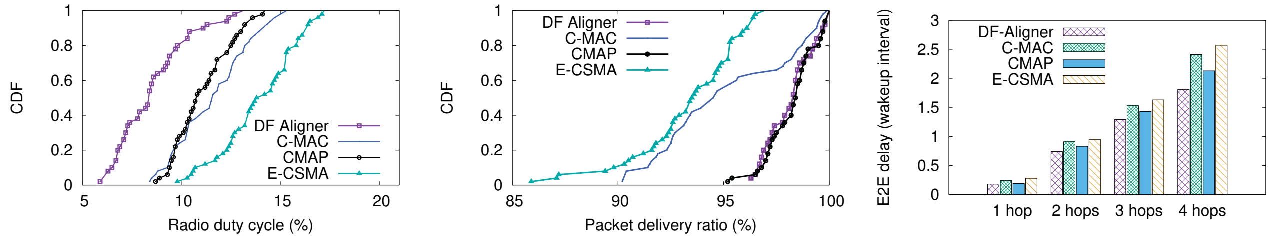
For each scenario of Figure 3, we repeat the experiments by changing the position of each node to construct various possible cases in practical networks. By collecting the reported intra-node waiting time of each packet, we plot the CDF of them in Figure 9. As shown by Figure 9(a), the distribution of intra-node waiting time of DF-ALIGNER is extremely close to that of DF-Optimal. Note that DF-Optimal can completely turn exposed terminals into transmission opportunities without suffering ACK loss or collision. The very small performance gap indicates that ACK loss/collision in DF-ALIGNER is rare. However, because DF-Typical should further consider ACK collision resulted from mutual interference when it constructs the concurrent transmission decision, the harnessing ratio of exposed terminal is significantly decreased compared with DF-Optimal and DF-ALIGNER. Furthermore, in concurrent transmission, DF-Typical cannot avoid the mutual interference on ACK collision. As shown by the figure, the intra-node waiting time of DF-Typical is $1.9\times$ of DF-ALIGNER.

6.2.2 *Impact of Network Dynamics*

The experiments mentioned above are conducted in interference-free environment. Moreover, we also conduct these prototypes in indoor scenario using the 19^{th} wireless channel that is overlapped with WIFI. In this condition, we conduct the same experiments mentioned above and collect the recorded intra-node waiting time. By plotting the distribution of waiting time in Figure 9(b). As shown by the figures, by suffering the external interference, the distribution of intra-node waiting time of ALIGNER varies little. It is because although external interference has impact on data frame decoding at receiver and ACK decoding at senders, it does not deprive the transmission opportunities. In contrast, DF-Typical cannot harness most of exposed terminals. If an ongoing sender can not quickly deliver its packet due to data collision caused by external interference, the deferred sender should keep waiting so that the intra-node waiting time significantly increases.

6.2.3 *Harnessing Exposed Terminals*

During the above experiments, each node also records: when a new packet is generated, whether the wireless channel is occupied by another ongoing sender. If so, whether it can concurrently transmit with the ongoing sender. According to the statistical data, we compute the exposed terminal harnessing ratio and list them in Table 3. Because DF-Typical should further consider the influence of mutual interference on ACK decoding, most of the exposed terminals cannot be turn into transmission opportunities. The statistical results indicate that by eliminating the mutual interference on ACK decoding, ALIGNER can double (increased by $1.05\times$) the harnessing ratio of exposed terminals. The



(a) Radio duty cycle (b) Packet delivery ratio (c) End to end delay

Figure 10. Performance of DF-ALIGNER, C-MAC, CMAP, and E-CSMA in a busy network where each node generates a packet every 30 seconds.

Table 4. ACK collision ratio in concurrent transmission by using ALIGNER and in non-concurrent transmission by using BoX-MAC.

Experimental setup	ACK collision ratio		
	Mean	Minimum	Maximum
Concurrent ALIGNER	0.92%	0.33%	1.63%
Non-concurrent BoX-MAC	0.75%	0.26%	1.33%

experimental results demonstrate that the effect of mutual interference on ACK decoding has a significant impact on harnessing exposed terminals.

6.2.4 ACK Collision Ratio

In this section, we conduct experiments to evaluate the ACK collision ratio by adopting ALIGNER for concurrent transmission. We randomly select two neighboring senders to concurrently transmit data packets to their receivers. Both the senders and receivers are all connected to the central control via wireline. Receivers report the receipt time of each data packet and the corresponding time of replying an ACK to sender. By receiving an ACK, senders report the receipt time of ACK to central control. For each pair of sender and receiver, at least 500 data packets have been transmitted. We also conduct experiment to test ACK collision ratio when senders adopt BoX-MAC [31] (an enhanced LPL-based MAC protocol) to transmit data packet in non-concurrent mode. If a receiver replies an ACK while the corresponding sender can not receive it, we consider that an ACK collision occurred. By selecting different senders and receivers to repeat this experiment, we compute the ACK collision ratio of each pair of sender and receiver by using ALIGNER to achieve concurrent transmission.

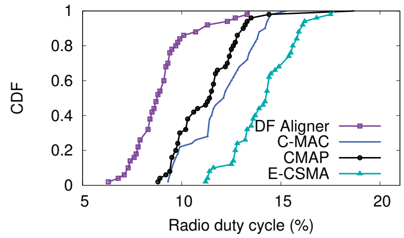
As shown in Table 4, the mean (minimum and maximum) ACK collision ratio of ALIGNER is 0.92% (0.33% and 1.63%), which is slightly higher than the mean ACK collision ratio (0.75%) of BoX-MAC which transmits data in non-concurrent mode. Compared with BoX-MAC, he additional ACK collision ratio of ALIGNER, which is about 0.17%, is mainly caused by the collision of ACKs which are simultaneously replied by different receivers. Compared with the significantly improved harnessing ratio of exposed terminals, the slightly increased ACK collision has little impact on the performance of data forwarding.

6.3 ALIGNER for Data Forwarding

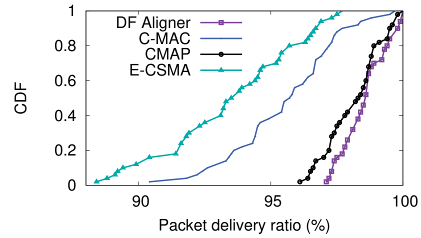
To further evaluate the performance of ALIGNER in low power ad-hoc networks, we further perform experiments in the indoor testbed with 50 nodes. We have reimplemented and used several related methods that are implemented based on TinyOS. We first use the ready-made CTP and BoX-MAC to test the performance of non-concurrent data transmission (denoted as DF-BoX-MAC). The performance of BoX-MAC is regarded as a baseline to see how much improvement can be taken from concurrent transmission. In addition, we also integrated ALIGNER into CTP and LPL (marked as DF-ALIGNER), implemented the channel probability based E-CSMA [11] that is proposed to address exposed/hidden terminals, implemented C-MAC [36] that is physical interference model based concurrent transmission protocol, and implemented the conflict graph [25] based approach for harnessing exposed terminals (marked as CMAP). Due to the different constructing process of concurrent transmission decision, for fairness, there's plenty of time (2 minutes in our experiments) to construct the concurrent transmission decision for each experiment and we filter out the network overhead of the constructing process when we analyze the experimental results.

We construct a busy network by setting each node's inter-packet interval to 30 seconds and simulate an event-driven network where 10 selected neighboring nodes will continuously generate packets. Each event will last for 30 seconds and the event normally occurs every 5 minutes. Each node's radio power level is set to 5 to form multiple hops networks. The wakeup intervals is 512ms. All network nodes are synchronized to an external control node by overhearing synchronization message. We perform DF-BoX-MAC, DF-ALIGNER, C-MAC, CMAP, and E-CSMA in the testbed respectively. We analyze the end-to-end delay of each packet, and the energy consumption and packet delivery ratio of each node. Then we plot the experimental results in Figure 10 and 11, respectively.

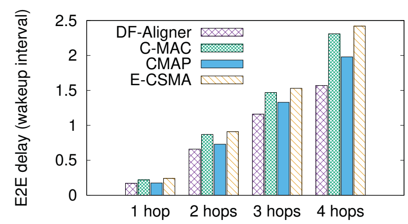
For no matter the busy network or event-driven network, DF-ALIGNER achieves the best performance compared with C-MAC, CMAP, and E-CSMA. By fully harnessing exposed terminals, DF-ALIGNER significantly reduces the intra-node waiting time, so as to reduce the radio duty cycle and the end-to-end delay of each packet. As



(a) Radio duty cycle



(b) Packet delivery ratio



(c) End to end delay

Figure 11. Performance of DF-ALIGNER, C-MAC, CMAP, and E-CSMA in an event-driven network.

Table 5. Performance difference between DF-ALIGNER and DF-BoX-MAC in busy network.

Protocols	DF-ALIGNER	DF-Box-MAC
Avg. PDR	99.83%	99.58%
Avg. Duty cycle	7.35%	9.29%
Avg. One-hop Delay	125 ms	221 ms

Table 6. Performance difference between DF-ALIGNER and DF-BoX-MAC in event-driven network.

Protocols	DF-ALIGNER	DF-Box-MAC
Avg. PDR	98.51%	72.57%
Avg. Duty cycle	9.23%	19.29%
Avg. One-hop Delay	131 ms	383 ms

shown by Figure 10(a) and 11(a), the radio duty cycle of DF-ALIGNER is smaller than CMAP by 34%, smaller than C-MAC by 38%, and smaller than E-CSMA by up to 45%. Moreover, by exploiting the potential transmission opportunities, DF-ALIGNER quickly guides network packets to the sink node. The relatively small number of network packets result in less interference. As shown in Figure 10(b) and 11(b), DF-ALIGNER can provide the highest packet delivery ratio. CMAP is inferior to DF-ALIGNER. Because CMAP should consider the effect of mutual interference on ACK decoding, a great portion of exposed terminals cannot be harnessed by CMAP. Similarly, E-CSMA also consider the impact of mutual interference on ACK decoding. Due to the complex mapping relationship between channel signal strength and forwarding decision, the channel probability based E-CSMA has the worst performance. Different from CMAP and E-CSMA, the physical interference model based C-MAC is proposed to transmit large data block. However, in the vast majority of practical networks, C-MAC cannot exert its advantage. According to the experimental results, we can conclude that DF-ALIGNER significantly outperforms CMAP, C-MAC, and E-CSMA.

As a baseline, we can see that in all scenarios with busy network traffic or event-driven burst traffic, the delivery ratio, radio duty cycle, and transmission latency of DF-ALIGNER outperforms DF-BoX-MAC. As shown in Table 5 and 6, for networks with busy traffic load, by harnessing exposed terminals and fully avoiding ACK collision, ALIGNER can save energy consumption by 26.4%, and reduce one-hop transmission latency by 76.8% compared with the CSMA-based BoX-MAC. Moreover, in event-driven networks, the packet delivery ratio of DF-ALIGNER is 98.51%, which significantly outperforms the 72.57% of DF-BoX-MAC. For this kind of event-driven application scenarios, when networks experience sudden surge of traffic, exposed terminals can significantly impact network data transmissions. On that basis, the averaged radio duty cycle of DF-BoX-MAC increases sharply up to 19.29%, and the averaged one-hop data transmission latency increases to 383ms. Compared with DF-BoX-MAC, the averaged radio duty cycle of DF-ALIGENR is only 9.23%, and the one-hop transmission delay is 131ms.

7 Related Work

Concurrent transmission is a well-known concept in wireless networks. MACA [19] observes that carrier sense cannot fully utilize the capacity of wireless channel. Many works are proposed to address this problem. By removing the potential ACK collisions, ALIGNER mainly targets on improve the throughput of concurrent transmission in duty cycle wireless networks. Next, we divide the most of existing mechanisms into three categories.

Capture Effect: In wireless communication, capture effect [35] [42] indicates that a receiver can successfully decode the strongest signal when the strongest signal fulfill the spacial and temporal constrains. Some mechanisms are proposed to construct capture effect and enable concurrent transmission. [15] and [20] theoretically investigate the modulation and coding schemes of capture effect in ALOHA and 802.11 networks, respectively. Moreover, by adjusting the transmission timing of concurrent frames to fulfill the constraints of capture effect, [12] [28] [5] improve the network throughput in sensor networks. ALIGNER is parallel with these mechanisms. Combining with ALIGNER, the network throughput can be further improved.

Parameter Adjustment: Some works enable concurrent transmission by adjusting protocol parameters (e.g., transmission power, carrier sense threshold) in wireless ad-hoc networks. On one hand, several works [23] [36] are designed to mitigate data collision by adjusting transmission power. On the other hand, adjusting the carrier sense threshold of CSMA is exploited in both 802.11 [3] and sensor [2] networks. Parameter adjustment can optimize the network performance, but it can never completely solve the problem.

Different to parameter adjustment strategies, ALIGNER can improve the network throughput by fully turning exposed terminals into transmission opportunities.

Conflict Graph: Conflict graph is a widely used tool to schedule concurrent transmission and avoid collisions. According to different link models of conflict graph, existing works can be divided into two categories. The first category is based on physical model [14] [34], in which transmissions over two links are either collision or collision-free. Compared with ALIGNER, the physical model is more conservative so that prohibits part of concurrent opportunities. The second category [45] uses signal strength of other links to illustrate the condition of interference. The per-link signal strength is either actively [27] [41] or passively [26] [37] measured. Beside the signal strength, the timing constraints of concurrent transmission is equally important in duty cycle networks. ALIGNER carefully designs both passively measured concurrent transmission metric and timing alignment to fully explore the capacity of concurrent transmission.

Although different to traditional concurrent transmission techniques that adopt constructive interference or capture effect on physical layer, we still use the term 'concurrent transmission' to emphasize that ALIGNER is proposed to optimize the concurrency opportunities in low power ad-hoc networks. Different to the transmission protocols based on constructive interference or capture effect, the carry out of ALIGNER depends on a concurrent metric, such as cpdr or signal-to-interference-plus-noise-ratio model, which tells ALIGNER whether it is feasible to perform concurrent transmission. On that basis, ALIGNER schedules neighboring senders to concurrently access shared wireless channel for data transmission. Hence, ALIGNER still falls into the category of concurrent transmission technique.

ALIGENR is also related to some other works that mainly focus on energy efficiency or collision resolution, such as ContikiMAC [8] and Strawman [32]. ContikiMAC [8] was proposed to provide extremely outstanding energy efficiency by adopting fast sleep and phase lock mechanism. Compared with the current implementation of ALIGNER, ContikiMAC can provide better energy efficiency performance, Even so, ALIGNER can provide more transmission opportunities for neighboring senders with the ubiquitous exposed terminals. Actually, ContikiMAC and ALIGNER are respectively proposed to address different problems in wireless networks. We look forward to combining ALIGNER technique with ContikiMAC for both harnessing exposed terminals and providing excellent energy efficiency performance in the future work. Strawman [32] is a contention resolution mechanism that applies to the scenarios that multiple senders transmit data packets to the same receiver. By adopting Strawman, receiver lets multiple senders to contend for the next data transmission time by simultaneously sending contention packets with different lengths after detecting a data collision. The sender with the longest contention packet wins and then completes its transmission in the next time slot. Different to Strawman, ALIGNER is designed for ACK collision avoidance in harnessing exposed terminals, which probably involve multiple senders and different receivers.

8 Conclusion

In this paper, we propose ALIGNER, which can completely prevent ACK corruption during concurrent transmission. The basic concept is to align the end of all concurrently transmitted packets. Thus during the common ACK waiting period, all ACKs do not suffer from the interference of the exposed terminal. In addition, for ALIGNER, we present a tailor-made metrics to analyze the feasibility of concurrent transmission. Experimental results demonstrate that ALIGNER can significantly increase concurrency opportunities.

9 Acknowledgments

The authors would like to thank the shepherd, Dr. Luca Mottola, for his insightful feedback. Thanks also to anonymous reviewers for reading this paper and giving valuable comments.

This study is supported in part by NSFC No. 61472067, Leadership Funding support for Academic and Technical Leaders of Sichuan Province, No. W03002360100203, NSFC No. 61472217, and NSFC No. 61202444.

10 References

[1] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, October 2010.

[2] C. A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Romer, and M. A. Zuniga. Making sensornet mac protocols robust against interference. In *Proceedings of EWSN*, pages 272–288. Springer Berlin Heidelberg, April 2010.

[3] M. Z. Brodsky and R. T. Morris. In defense of wireless carrier sense. In *Proceedings of SIGCOMM*, pages 147–158. ACM, August 2009.

[4] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of SenSys*, pages 307–320. ACM, October 2006.

[5] Z. Cao, D. Liu, J. Wang, and X. Zheng. Chase: taming concurrent broadcast for flooding in asynchronous duty cycle networks. *IEEE/ACM Transactions on Networking*, 25(5):2872–2885, June 2017.

[6] H. Chen, Z. Zhang, L. Cui, and C. Huang. Nopsm: a concurrent mac protocol over low-data-rate low-power wireless channel without prr-sinr model. *IEEE Transactions on Mobile Computing*, 16(2):435–452, March 2017.

[7] Contiki. Contiki: The open source operating system for the internet of things, [online] http://www.contiki-os.org/.

[8] A. Dunkels. The contikimac radio duty cycling protocol. Technical report, May 2012.

[9] 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 *Proceedings of SenSys*, pages 1–14. ACM, November 2010.

[10] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pasztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef. Evolution and sustainability of a wildlife monitoring sensor network. In *Proceedings of SenSys*, pages 127–140. ACM, November 2010.

[11] S. Eisenman and A. Campbell. E-csma: Supporting enhanced csma performance in experimental sensor networks using per-neighbor transmission probability thresholds. In *Proceedings of INFOCOM*, pages 1208–1216. IEEE, May 2007.

[12] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *Proceedings of IPSN*, pages 73–84. ACM/IEEE, April 2011.

[13] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of SenSys*, pages 1–14. ACM, November 2009.

[14] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–40, March 2000.

[15] I. Habbab, M. Kavehrad, and C. Sundberg. Aloha with capture over slow and fast fading radio channels with coding and diversity. *IEEE Journal on Selected Areas in Communications*, 40(3):79–88, January 1989.
[16] T. Instruments. Cc1100 data sheet, [online] https://www.ti.com/lit/ds/symlink/cc1100.pdf.
[17] T. Instruments. Cc2420 data sheet, [online] https://www.ti.com/lit/ds/symlink/cc2420.pdf.
[18] T. Instruments. Cc2520 data sheet, [online] [online] https://www.ti.com/lit/ds/symlink/cc2520.pdf.
[19] P. Karn. Maca : A new channel access method for packet radio. *Proc. ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990.
[20] J. H. Kim and J. K. Lee. Capture effects of wireless csma/ca protocols in rayleigh and shadow fading channels. *IEEE Transactions on Vehicular Technology*, 4(4):1277–1286, July 1999.
[21] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson. Low power, low delay: opportunistic routing meets duty cycling. In *Proceedings of IPSN*, pages 185–196. ACM, April 2012.
[22] P. Levis and D. Gay. *TinyOS Programming*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
[23] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. Atpc: Adaptive transmission power control for wireless sensor networks. In *Proceedings of SenSys*, pages 223–236. USENIX Association, October 2006.
[24] T. Lin and J. Hou. Interplay of spatial reuse and sinr-determined data rates in csma/ca-based, multi-hop, multi-rate wireless networks. In *Proceedings of INFOCOM*, pages 803–811. IEEE, May 2007.
[25] D. Liu, M. Hou, Z. Cao, Y. He, X. Ji, and X. Zheng. Cof: Exploiting concurrency for low power opportunistic forwarding. In *Proceedings of ICNP*, pages 32–42. IEEE, November 2015.
[26] S. Liu, G. Xing, H. Zhang, J. Wang, J. Huang, M. Sha, and L. Huang. Passive interference measurement in wireless sensor networks. In *Proceedings of ICNP*, pages 52–61. IEEE, October 2010.
[27] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste. Dirc: Increasing indoor wireless capacity using directional antennas. In *Proceedings of SIGCOMM*, pages 171–182. ACM, August 2009.
[28] J. Lu and K. Whitehouse. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *Proceedings of INFOCOM*, pages 2491–2499. IEEE, April 2009.
[29] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu. Citysee: Urban co_2 monitoring with sensors. In *Proceedings of INFOCOM*, pages 1611–1619. IEEE, March 2012.
[30] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai. Canopy closure estimates with greenorbs: Sustainable sensing in the forest. In *Proceedings of SenSys*, pages 99–112. ACM, November 2009.
[31] D. Moss and P. Levis. Box-macs: Exploiting physical and link layer boundaries in lowpower networking. Technical report, 2008.
[32] F. Osterlind, L. Mottola, T. Voigt, N. Tsiftes, and A. Dunkels. Strawman: Resolving collisions in bursty low-power wireless networks. In *Proceedings of IPSN*, pages 161–171. ACM/IEEE, April 2012.
[33] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of SenSys*, pages 95–107. ACM, November 2004.
[34] I. Rhee, A. Warrier, J. Min, and L. Xu. Drand: Distributed randomized tdma scheduling for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 8(10):1384–1396, March 2009.
[35] L. G. Roberts. Aloha packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, April 1975.
[36] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang. C-mac: Model-driven concurrent medium access control for wireless sensor networks. In *Proceedings of INFOCOM*, pages 1845–1853. IEEE, April 2009.
[37] V. Shrivastava, S. Rayanchu, S. Banerjee, and K. Papagiannaki. Pie in the sky: Online passive interference estimation for enterprise wlans. In *Proceedings of NSDI*, pages 337–350. USENIX Association, April 2011.
[38] Sigfox. Lora alliance standard specification, [online] http://www.sigfox.com/.
[39] N. Sornin, M. Luis, T. Eirich, and T. Kramp. Lora alliance standard specification, [online] https://www.lora-alliance.org.
[40] C. Technology. Telosb mote platform, [online] http://www.xbow.com/products/product_pdf_files/wireless_pdf/telosb_datasheet.pdf.
[41] M. Vutukuru, K. Jamieson, and H. Balakrishnan. Harnessing exposed terminals in wireless networks. In *Proceedings of NSDI*, pages 59–72. USENIX Association, April 2008.
[42] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the capture effect for collision detection and recovery. In *Proceedings of EmNets*, pages 45–52. IEEE, May 2005.
[43] Z. Zhao, W. Dong, J. Bu, T. Gu, and G. Min. Accurate and generic sender selection for bulk data dissemination in low-power wireless networks. *IEEE/ACM Trans. Netw.*, 25(2):948–959, April 2017.
[44] X. Zheng, J. Wang, W. Dong, Y. He, and Y. Liu. Bulk data dissemination in wireless sensor networks: analysis. *IEEE Transactions on Computers*, 65(5):1428–1439, June 2015.
[45] X. Zhou, Z. Zhang, G. Wang, X. Yu, B. Y. Zhao, and H. Zheng. Practical conflict graphs for dynamic spectrum distribution. In *Proceedings of SIGMETRICS*, pages 5–16. ACM, June 2013.

A Computation of Cpdr

The computation of cpdr is not complex. To compute cpdr, several statuses of data transmission are indispensable. For each data transmission, the sender needs to know which receiver successfully received the data packet. Meanwhile, sender needs to know what caused a failed transmission: i) data collision at receivers receiving end; or ii) ACK collision at its own transmitting end.

As sender, for each data transmission assigned with a unique DSN (data sequence number), it records whether the data transmission was acknowledged or not, and whether there was a neighboring node that transmitted concurrently with the data transmission. As receiver, for each data packet transmitted by its sender, it records the number of received data packets which are assigned with the same DSN. In data transmission, once a receiver receives a data packet transmitted by its sender, it immediately replies an ACK. Hence, the received multiple (k) data packets assigning with the same DSN indicate at least the previous k-1 ACKs were collided and lost at the sender.

By feeding back the information to the sender, combining with the above-mentioned transmitting status maintained by the sender, it can exactly know the transmission result of each data transmission and the number of lost ACKs under the influence of a specific neighboring node. Then we can compute the unidirectional cpdr from the sender to the receiver under the influence a specific neighboring node, and compute the unidirectional cpdr from the receiver to the sender under the same influence of the neighboring node. Considering both the unidirectional cpdr from sender to receiver and the unidirectional cpdr from receiver to the sender, it is easy to compute the bidirectional cpdr between the sender and receiver under the influence of a specific neighboring node.

The memory consumption for computing cpdr is mainly dependent on network density, i.e, the number of neighboring nodes. In previous research work [25], we have conducted experiment to demonstrate the relation between memory overhead and network density. If readers want to know more about it, please refer the experimental results presented in the paper [25].