Concurrent Low Power Listening: A New Design Paradigm for Duty-Cycling Communication

DAIBO LIU, Hunan University
ZHICHAO CAO, Michigan State University
HONGBO JIANG, Hunan University
SIWANG ZHOU, Hunan University
ZHU XIAO, Hunan University
FANZI ZENG, Hunan University

In this paper, we explore a new design paradigm of duty-cycling mechanism that supports low power devices to fully turn channel contention into transmission opportunities. To achieve this goal, we propose Concurrent Low Power Listening (CLPL) to enable contention-tolerant and concurrent media access control (MAC) for widely deployed low power devices. The fundamental principle behind CLPL is that frequency modulated receiver can reliably demodulate the strongest signal even if cochannel interference and noise exist. By using CLPL, a sender inserts a series of tailor-made signals (namely wake-up signal) between adjacent data frames to awaken appointed receiver, making it capable to receive the next data frame. According to system-defined maximum transmission power level, CLPL adopts an adaptive algorithm to adjust the transmission power of wake-up signals so that its signal strength is above receiver sensitivity and will not interfere the other data frames in transit. By exploiting the spatial-temporal correlation, we further develop a light-weight wake-up signal detection method to enable a waiting sender to accurately identify the current channel condition. Then, it schedules the sender’s data frame transmissions by overlapping with those wake-up signals, without conflicting with existing data frame transmissions. We have implemented the prototype of CLPL and conducted extensive experiments on a real testbed. In comparison with the state-of-the-art low power MAC schemes, such as ContikiMAC, A-MAC, BoX-MAC, and opportunistic scheme ORW, CLPL can improve the throughput by 2-6 times and halve the end-to-end transmission delay.

Additional Key Words and Phrases: Internet of thing (IoT); Duty-cycled networks; Concurrent low power listening; Media access control (MAC).

1 INTRODUCTION

Duty-cycling has emerged as the predominant method for optimizing power consumption of low-power radios in Internet of Things (IoT) [2]. Nowadays, duty-cycled radios like LoRa [24], Zigbee [35] and Bluetooth LE [36] have become the preferred choice for sensors that transmit sporadically in small bursts, for example, building sensors [19], air quality sensors [22], smart home security alarm sensors [29], industry monitoring sensing devices [16], and other IoT devices.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.
1550-4859/2022/3-ART $15.00
https://doi.org/10.1145/3517013

However, sporadic traffic for most of the network lifetime doesn’t mean that it has no need to consider the data delivery capacity of duty-cycling IoT networks. Considering an event-driven network [29][16][40], upon detecting the target event, a number of low power IoT devices (or called node for simplicity) have to report the event immediately via multi-hop relay, bringing about sudden bursts of traffic [19]. Note that this situation is common in widely applied monitoring and alert IoT systems. Moreover, in duty-cycling mode, low power nodes in IoT networks usually sleep for most of the time and are with asynchronous active-sleep schedule. For communication between duty-cycled nodes, a data transmission may go on for a long time, resulting in contenders having to postpone their data transmission for collision avoidance, as illustrated in Fig. 1(a). The sudden bursts of monitoring data and duty-cycling mode bring great difficulty for fast and reliable data transfer in duty-cycling IoT networks [21].

Existing methods are hopeless with a dilemma, that is, the conflict between the energy efficiency of duty-cycling mode and inefficient utilization of the sharing channel resource [12][5]. But with the widely applied IoT, it is urgent to explore a new design paradigm of duty-cycling channel access scheme that can guarantee both energy efficiency and fast data transfer for sudden bursts of traffic in reliable way. To achieve this goal, the central challenge that we face is how to deal with the interference problem between potential contenders when they use the shared channel at the same time. In other words, multiple contenders use the shared channel according to a heuristic mechanism with no need of pre-schedule and without interfering each other.

Note that there are considerable gaps between received signal strength of data packets and receiver sensitivity in different low power wireless techniques [33]. As demonstrated by capture effect [20], a receiver can reliably demodulate the strongest signal even if cochannel interference and noise exist. Given a system-defined maximum transmission power level, it is possible to construct an intentional signal with well-designed output power that can not only be above receiver sensitivity but will not interfere other data packets (referred to as data frame) in transit. In this situation, duty-cycled devices have great potential to turn channel contention to transmission opportunities without incurring additional data packet collision.

In this paper, we propose a new design paradigm of duty-cycling MAC mechanism, Concurrent Low Power Listening (referred to as CLPL), to enable contention-tolerant and concurrent media access for increasingly widely deployed low power IoT nodes, as illustrated in Fig. 1(b). The basic idea of CLPL is that a sender inserts a series of tailor-made signals (referred to as wake-up frame and denoted as WF) between adjacent data frames to awaken appointed receiver and make it capable to receive the next data frame (see Section 4). The series of wake-up frames are intentionally designed to rendezvous with receiver’s wake-up period. CLPL exploits a well studied physical model (PRR-SINR) [37] and proposes an online measurement scheme to determine the optimal transmission power of wake-up signals. Because the determined transmission power is limited to a predetermined maximum power level, we can guarantee that its signal strength is above receiver sensitivity and will not interfere potential data or acknowledgement (ACK) frames in transit (see Section 5). By exploiting the
spatial-temporal correlation, even though wake-up signals may not be decoded properly, CLPL can accurately identify the event of wake-up frame transmissions for accessing the shared channel resource (see Section 6). Then, it schedules data frame transmissions by overlapping with those wake-up signals, without conflicting with existing data frame transmissions.

Compared with existing media access schemes, CLPL can fully achieve concurrent and conflict-free data transmission for multiple senders without compensating with other performance indicators. CLPL can not only significantly reduce the data transmission delay caused by channel contention, but also improve network capacity by up to 6.4 times compared with the state-of-the-art media access mechanisms, e.g., BoX-MAC [23], A-MAC [8], ContikiMAC [7], and the opportunistic scheme ORW [15], which could provide outstanding service for time-critical applications.

The main contributions of this work are as follows:

- We propose CLPL to support concurrent and contention-tolerant data transmission for multiple senders. CLPL can fundamentally eliminate the interference caused by concurrent senders in duty-cycling communication.
- We adopt a novel online measurement scheme to adaptively generate physical interference model for computing the optimal output power of wake-up frames. The well-designed signals provide the fundamental support for concurrent media access for multiple contenders.
- By using the spatial-temporal correlation, we can effectively identify wake-up frame transmissions. This identification scheme is a tailor-made channel situation assessment for CLPL.
- We have implemented a prototype of CLPL and conducted extensive experiments on a real testbed. In comparison with the state-of-the-art low power communication approaches, CLPL can significantly improve the throughput by 2-6 times and halve the end-to-end transmission delay.

This paper is organized as follows. In the next section, we introduce background and motivation of CLPL. By giving an overview of CLPL in Section 3, the detailed design of CLPL is respectively presented in Section 4, 5, and 6. The evaluation results are presented in Section 7. Section 8 introduces the related work. Finally, we conclude this paper in Section 10.

2 BACKGROUND AND MOTIVATION

**Low power listening**: Low power listening [26] (LPL) is a well-accepted MAC-layer concept for reducing energy consumption in duty-cycled wireless networks [27]. Because of high energy efficiency and small coordination overhead, the state-of-the-art MAC techniques, such as BoX-MAC-2 [23] and ContikiMAC [7], for low power communication are fundamentally built upon the design concept of LPL. With the LPL-based MAC mechanism as illustrated in Fig. 2, a node periodically wakes up to perform CCA (Clear Channel Assessment), i.e., sampling the energy of received signal. If energy is detected on the channel, the node extends its active phase to receive potentially upcoming data packet. Otherwise, according to fast sleeping mechanism, the node quickly goes back to
sleep as illustrated by Fig. 3(a). The minimal active period is referred to as $T_{idle\_wakeup}$. Without synchronization, a sender has to continuously transmit the same data frames until the receiver’s ACK is received or a pre-configured timer expires. When multiple senders access the media at the same time, except the one that has been occupying the shared channel, the others have to wait for a long time for collision avoidance, which significantly degrades channel utilization and causes serious transmission delay.

Motivation: Based on asynchronous LPL, when a duty-cycled node wakes up and detects a busy channel, it will extend its active phase to receive potentially upcoming data packets as illustrated by Fig. 3(b). For simplicity, we denote the period of the system-defined extension of active phase as $T_{EAP}$. This mechanism also indicates that as long as a sender can make its receiver to perceive the existence of a sender through any intentionally designed wake-up signal when it wakes up, the receiver will stay in active state for at least $T_{EAP}$ period. Then, if the sender can transmit a data frame during the following period of $T_{EAP}$ as illustrated by Fig. 4, it will not miss receiver’s wake-up phase.

Note that a sufficient condition for extending receiver’s active phase is that the captured strength of wake-up signal at receiver is larger than receiver’s sensitivity. Hence, as long as the intentionally designed signal can be detected by receiver and has no influence on the data/ACK frame decoding of neighboring nodes, it is rewarding to schedule multiple contenders to simultaneously access the shared channel resource. Moreover, in practical low power wireless networks, there are considerable gaps between received signal strength of data packets and receiver sensitivity [33]. Fig. 5 illustrates this gap between RSS and receiver sensitivity during short-range communication between a smartwatch and access point via Bluetooth by setting 0dBm output power. The capture effect further provides basic theory support to exploit well-designed wake-up signal to awaken receiver with weak output power, but contenders can completely treat them as background noise for media access. Based on
these considerations, it is possible to fully turn channel contention to transmission opportunities in duty-cycling communications.

**Difference to LPL.** Different to existing LPL-based media access schemes, this work explores a novel design paradigm of low power communication that fully supports conflict-free concurrent media access for duty-cycling IoT devices.

### 3 CLPL OVERVIEW

The main design concept of CLPL consists of five components: frame transmission schedule module, media access mechanism, wake-up frame generator and identification module, and energy optimization module.

#### Frame transmission schedule.
CLPL schedules data transmission by periodically alternating a data frame and a series of wake-up frames (Section 4.1). As shown in Fig. 6, \( S_1 \) and \( S_2 \) are two neighboring senders, and \( R_1 \) and \( R_2 \) are the corresponding receivers of them. To transmit a data packet, rather than continuously transmitting the same data frame as LPL-based MAC schemes [23][17][15] do illustrated in Fig. 2, a sender completes a data transmission by periodically alternating a data frame and a series of wake-up frames. The series of wake-up frames are designed to rendezvous with receiver’s waking moment. As illustrated in the figure, the boxes marked with “Data” denote data frames, and the “WF” represent wake-up frames.

#### Media access mechanism.
CLPL adopts a heuristic method to schedule multiple senders to concurrently access shared channel for data transmission (Section 4.2). Different to traditional channel access mechanism, CLPL allows a sender to access shared channel for data transmission in two situations: i) the current channel is free; or ii) through channel state assessment (see Section 6), sender can identify that during a period of time the channel was only used for wake-up frame transmissions, and this period can accommodate the sender’s data frame transmission. Because senders periodically transmit data frames with the same time interval, CLPL is fully convinced that no other neighbors’ data frame transmission will be scheduled to this period after a data frame transmission cycle. Hence, CLPL schedules the new sender to transmit in this period concurrent with neighbors’ WF transmissions.

#### Wake-up frame generator.
Given a pre-defined maximum transmission power level, the output power of wake-up frames must be carefully set. Wake-up frame generator guarantees that the signal strength of wake-up frame is above receiver sensitivity. As receiver, once it wakes up, it can certainly detect a busy channel by encountering/receiving either a wake-up frame transmission or a data frame transmission from the sender. Then, the receiver immediately extends its active phase by \( T_{EAP} \) ms, just like \( R_1 \) and \( R_2 \) do in Fig. 6. At the same time, the signal of wake-up frames must not interfere the data/ACK frames decoding in transit. CLPL adopts an online physical interference model (PRR-SINR) to optimize the output power of wake-up frame (Section 5).

#### Wake-up frame identification.
Besides to rendezvous with receiver’s wake-up period, wake-up frame is also used to provide identifiable features for neighbor senders to schedule concurrent transmissions. CLPL exploits the spatial-temporal correlation of wake-up frames to effectively identify their transmissions for assessing the feasibility of accessing shared busy channel (Section 6). In the following sections, we introduce the detailed design of these modules.

#### Energy optimization.
CLPL adopts a fast sleep mechanism to guarantee the targeting receiver will stay awake to receive a full data frame and transmit a link layer ACK. For the other neighboring nodes, by detecting the frame, could ignore it and quickly go to sleep again. Moreover, by hearing a wake-up frame, the receiver also uses fast acknowledgement mechanism to immediately inform sender to quickly transmit data frame (see Sec. 4.3).
Fig. 6. Workflow of CLPL. Senders complete a data transmission by periodically transmitting a data frame followed by a series of wake-up signals. In the figure, “Data” denotes data frame, and “WF” denotes wake-up signal.

4 CONCURRENT TRANSMISSION SCHEME

In this section, we first introduce the concurrent transmission scheme which consists of frame transmission scheduler and concurrent media access mechanism.

4.1 Frame Transmission Schedule

As illustrated in Fig. 6, to transmit a data packet, a sender periodically transmits data frame to appoint receiver, and each data frame is followed by a series of wake-up frames. The time interval between two consecutive frames (no matter data or wake-up frame) must be less than the minimum active duration ($T_{idle_{wakeup}}$) to invalidate receiver’s fast sleep (see Section 4.3) when it wakes up. Specifically, we refer to the interval between consecutive frames as $T_{frame_{interval}}$, which satisfies the conditions:

$$\begin{align*}
T_{frame_{interval}} & \geq T_{ack_{waiting}}, \\
T_{frame_{interval}} & < T_{idle_{wakeup}}.
\end{align*}$$

(1)

$T_{ack_{waiting}}$ denotes the mandatory idle-channel period for decoding potentially incoming ACK at sender side. Because both data frame and wake-up frame are carefully scheduled to rendezvous with receiver’s wake-up period, the receiver will certainly perceive the existence of a sender when it wakes up. Then, it immediately extends the active phase to receive potentially incoming data frames. Moreover, in order for the appointed receiver to decode at least one full data frame during the extended active period, the data frame cycle time, which is referred to as $T_{frame_{cycle}}$ and illustrated in Fig. 6, must satisfy

$$T_{frame_{cycle}} + T_{data_{onair}} \leq T_{EAP},$$

where $T_{data_{onair}}$ is transmission time of the scheduled data frame. In CLPL, we define $T_{frame_{cycle}}$ as

$$T_{frame_{cycle}} = T_{EAP} - T_{data_{onair}}(max).$$

(2)
where $\tau_{\text{data on air}}(\text{max})$ denotes the transmission time of the system-defined maximal data frame. As illustrated by Fig. 6, $T_{\text{frame cycle}}$ contains a data frame transmission and a series of wake-up frame transmissions. In this formula, $\tau_{\text{data on air}}(\text{max})$ denotes the possibly maximum waiting time for decoding a full data frame. When receiver wakes up right after the transmitting time of the start of frame delimiter (SFD) field of a transmitted data frame, receiver will fail to decode the data frame, even if the entire payload of the data frame is transmitted during the receiver’s active period. In this worst-case situation, to guarantee there is at least one opportunity to decode a full data frame, the receiver’s active period must be no less than the sum of sender’s $\tau_{\text{data on air}}$ and $T_{\text{frame cycle}}$.

This organization of frame transmissions in CLPL can provide sufficient channel access opportunities for potential senders to access the shared channel for data transmissions. For each specific pair of sender and receiver, they access the shared channel as almost the same they would monopolize the channel resource. The frame transmission organization could significantly reduce the idle-waiting time of data transmission.

4.2 Media Access Mechanism

If the current channel state is not free, as new sender, it must take three stages to schedule its data frame transmission. As illustrated by Fig. 7, the three stages can be summed up as: i) Extract channel features to identify wake-up frame transmission; ii) Detect the periodically appeared successive wake-up frame transmissions to verify the capacity of scheduling its own data frame transmission in this period; and iii) Determine data frame transmitting time and immediately schedule wake-up frame transmissions to avoid missing receiver’s waking moment.

As above-mentioned, in CLPL, a sender can transmit its data frame when the shared wireless channel is free or the assessed busy channel is caused by senders’ wake-up frame transmissions. Hence, in order to confirm the feasibility of the new sender’s data frame transmission, CLPL should first extract the features of shared channel states. If the features indicate the shared channel is free and the assessed idle channel period, referred it to as $T_{\text{free span}}$, satisfies

$$
T_{\text{free span}} \geq \tau_{\text{data on air}} + 2 \times \tau_{\text{ack waiting}} + \tau_{\text{max backoff}}^{\text{system-defined}}
$$

(3)

the sender can immediately schedule its data frame transmission. $\tau_{\text{data on air}}$ is transmission time of scheduled data frame and $\tau_{\text{max backoff}}^{\text{system-defined}}$ is a system-defined maximum backoff time for collision avoidance in CLPL. The intention of two $\tau_{\text{ack waiting}}$ is to guarantee that there are enough time gap before and after the scheduled data frame transmission. We use $t_0$ to denote the ending time of $T_{\text{free span}}$. Because there is a possibility that a neighboring
node has completed a data frame transmission at the starting of $T_{\text{free span}}$, hence, to guarantee that there are enough time gap before and after the scheduled data frame transmission for ACK receiving, we set two $T_{\text{ack waiting}}$ here. In addition, in consideration of the robustness of data transmission schedule, we require an extra duration of $T_{\text{max backoff}}$ to tolerate the backoff time once detecting potential interference.

In addition, according to the continuously sampled channel states, if the frame identification module (see Section 6) indicates the current channel is just used for wake-up frame transmission\footnote{The wake-up frames can be transmitted by one or multiple senders.}, it continues monitoring the shared channel and calculates the duration of successive wake-up frame transmissions, which is referred to as $T_{\text{wf span}}$. If $T_{\text{wf span}}$ satisfies

$$T_{\text{wf span}} \geq T_{\text{data on air}} + 2 \cdot T_{\text{ack waiting}} + T_{\text{max backoff}},\quad (4)$$

the sender is also permitted to schedule its data frame transmission. As above, we use $t_0$ to denote the ending time of the $T_{\text{wf span}}$.

On that basis, the new sender immediately starts its frame transmission at $t_0$ beginning with successive wake-up frame transmissions. At the same time, CLPL further computes the transmitting time, $t_{\text{data}}$, of the next data frame. Because the time length of data transmission cycle is a constant $T_{\text{frame cycle}}$. Hence, CLPL can compute the transmitting time of the next data frame according to

$$t_{\text{data}} = (t_0 + T_{\text{frame cycle}}) - (T_{\text{on air}} + T_{\text{ack waiting}}) - T_{\text{random backoff}},\quad (5)$$

where $T_{\text{random backoff}}$ is a random backoff time between $[0, T_{\text{max backoff}}]$ to avoid data collision caused by concurrent senders with exactly the same frame transmission schedule. Before the $t_{\text{data}}$ comes, CLPL schedules successive wake-up frame transmissions with fixed frame interval $T_{\text{frame cycle}}$ to avoid missing receiver’s wake-up period. When $t_{\text{data}}$ arrives, CLPL immediately transmits the data frame. If an ACK is not received, CLPL continues to transmit wake-up frames until the next data transmitting time, $t_{\text{next}}$, comes, where

$$t_{\text{next}} = t_{\text{data}} + T_{\text{frame cycle}}.$$

By this mean, CLPL can heuristically schedule multiple senders to simultaneously access the shared channel for data frame transmissions in conflict-free way. Essentially, by applying CLPL, to concurrently transmit with a neighboring node, the new sender has to synchronize with the ongoing transmissions and then schedule its frame transmissions. However, different to existing works, the synchronizing process of CLPL is heuristic and does not require any interaction to neighboring nodes. Hence, the maintenance overhead of CLPL is negligible and without requiring any initialization phase.

The main process of concurrent channel access mechanism is explained by Algorithm 1 and summarized as follows: Like LPL, if the sampled channel states indicate the shared channel is free, i.e. Eqn. (3) is satisfied, a new sender is authorized to transmit. Rather than immediately transmitting the data frame like LPL, CLPL first schedules a series of wake-up frame transmissions until the computed data transmitting time ($t_{\text{data}}$) comes. Different to traditional LPL, by perceiving a busy channel in CLPL, a sender should further figure out whether it is caused by wake-up frame transmission (Line 9), rather than just giving up the chance of data transmission. If the assessed $T_{\text{wf span}}$ satisfies Eqn. (4), CLPL immediately schedules the new sender to transmit wake-up frames and data frame. On the other hand, if the busy channel is caused by senders’ data frame transmissions or external interference, i.e. Eqn. (4) is not satisfied (Line 15), CLPL continues to sample channel state until the channel feature satisfies the above requirements. In this way, CLPL can heuristically achieve concurrent channel access for multiple senders.
ALGORITHM 1: Concurrent media access mechanism

Input: Successively sampled channel states;
Output: Frame transmission schedule.

1 while Sampling channel state do
2    if Free channel then
3        Update $T_{free\_span}$;
4        if Eqn. (3) is satisfied then
5            Record $t_0$;
6            Compute $t_{data}$ according to Eqn. 5;
7            Return Schedule frame transmission;
8    else if Busy channel then
9        if Identify wake-up frame transmission then
10           Update $T_{wf\_span}$;
11           if Eqn. 4 is satisfied then
12              Record $t_0$;
13              Compute $t_{data}$ according to Eqn. 5;
14              Return Schedule frame transmission;
15        else if Identify other frame transmission or interference then
16            $T_{wf\_span} = 0$;
17        end
18    end
19 end

4.3 Energy Optimization at Receiver

4.3.1 Fast sleep.
If a neighbor is transmitting a data/wake-up frame to the receiver, the receiver should stay awake to receive the full data frame and transmit a link layer ACK. Other nodes, which detect the frame, could quickly go to sleep again. Potential receivers cannot go to sleep quickly, however, as they must be able to receive the full data frame. The naive way to determine how long to be awake when a CCA has detected radio activity is to stay awake for $T_{EAP}$. This ensures that the full data frame will be received by the receiver.

![Fast acknowledgement mechanism](image)

Fig. 8. Fast acknowledgement mechanism for informing sender to quickly transmit data frame.

The fast sleep optimization lets potential receivers go to sleep earlier if the CCA woke up due to spurious radio noise. The fast sleep optimization leverages knowledge of the specific pattern of CLPL frame transmissions.

as follows. First, if the CCA detects radio activity, but the radio activity has a duration that is longer than the maximum frame length $f_{\text{data}_{\text{on-air}}(\text{max})}$ and not caused by wake-up frame transmissions, the CCA has detected noise and can go back to sleep. Second, if the radio activity is followed by a silence period that is longer than the system-defined intervals, i.e., $T_{\text{frame, interval}}$, the receiver can go back to sleep. Third, if the activity period is caused by one sender’s wake-up frame transmissions which are transmitted to another receiver$,^2$ by overhearing one of the wake-ups frames, the receiver can go back to sleep.

4.3.2 Fast acknowledgement. Contrary to the third case of fast sleep mechanism, if a node overhears a wake-up frame and it happens to be the appointed receiver, as illustrated in Fig. 8, CLPL immediately replies an ACK to inform sender to quickly transmit data frame. By receiving the fast ACK, the sender can immediately reschedule its data frame transmission if no concurrent neighboring sender has been detected. Otherwise, the sender transmits its data frame according to its original schedule.

5 WAKE-UP FRAME GENERATOR

In this section, we first introduce the well-studied measurement of PRR-SINR model, and then use the online model to optimize the output power of wake-up frame.

5.1 Online PRR-SINR Model

In CLPL, the determination of wake-up frame’s output power is based on the widely-studied physical model, i.e., a packet from the sender is lost at the receiver only if the signal-to-interference-plus-noise-ratio (SINR) falls below a given threshold $[37]$, referred to as PRR-SINR model. CLPL adopts tailor-made power test beacon to capture the received signal strength (RSS) of each beacon and the corresponding interference-plus-noise strength for online measurement of PRR-SINR model.

Power beacon consists of normal power beacon (NPB) and low power beacon (LPB). Each NPB records the output power of the next LPB. LPBs are set up to different power levels.

Fig. 9. Schedule of power test beacon (PB) transmission. PBs are continuously transmitted by setting a normal output power (NPB) and a relatively low output power (LPB) alternately. $L_1$, $L_2$, and $L_n$ denote the transmission output power of LPBs.

Power Test Beacon. In order to make all neighbor nodes accurately assess the RSS of wake-up frames by setting different transmission output power, CLPL uses an additional type of beacon, referred to as power test beacon or denoted as PB for short, as illustrated in Fig. 9. Different to the schedule of data frame transmission, all types of $^2$Receiver information is specified in wake-up frame.
beacons in CLPL are continuously transmitted like Fig. 2. Hence, when a node transmits a network beacon, all neighbor nodes are not allowed to access the shared channel.

Power test beacons are continuously transmitted by setting a normal output power and a relatively low output power alternately. The normal output power is the same as data frame’s transmission output power, and we express the kind of beacons transmitted by normal output power as NPB. The low output power is set between the minimal output power and the normal output power in ascending order. The beacons transmitted by low output power is expressed as LPB, and the power level $i$ is marked as $L_i$. The time interval between two adjacent NPBs, referred to as $T_{\text{NPB \_interval}}$, should be less than the minimal active period $T_{\text{idle \_wakeup}}$. Therefore, when a neighbor wakes up, it will invalidate the fast sleep mechanism to hear more power test beacons. For each NPB, it records the low output that is set for the following LPB. For example, $\mathbf{Tx}$’s first NPB in Fig. 9 records the power level $L_1$ in its payload, and $L_1$ is the transmission power of the following LPB. Moreover, each PB (NPB and LPB) also records its own transmission output power. Hence, by either extracting the recorded output power and the embedded RSSI (measured during receiving process) from a decoded LPB or extracting the attached output power from a NPB and assessing the RSSI of the following LPB that is not correctly received, the neighbor $\mathbf{Rx}$ in Fig. 9 can exactly know the RSSI of a PB corresponding to a specific transmission power $L_i$, expressed as $<\mathbf{Tx}, L_i, RSSI_{L_i}>$. In our experiment of CLPL, the power test beacon size is set to 15 Bytes, corresponding to 480\(\mu s\) on-air time. The interval between two consecutive power test beacon is set to 128\(\mu s\).

**Online Measurement of PRR-SINR.** Beyond the RSSI of PBs, CLPL also samples the channel states during the time span between consecutive PBs. The detected energy can be used to express the strength of interference and noise in the shared channel, which is referred to as $\text{RSSI}_{IN}$. Hence, for a power test beacon transmitted by $\mathbf{Tx}$, a neighboring node $\mathbf{Rx}$ can exactly know whether it has been successfully received, the RSSI of the PB (referred to as $\text{RSSI}_{PB}$), and the corresponding $\text{RSSI}_{IN}$. For each $\text{RSSI}_{PB}$ and the corresponding $\text{RSSI}_{IN}$, CLPL can compute the corresponding SINR according to the formula

$$\text{SINR}(\text{Tx}, \text{Rx}) = \frac{\text{RSSI}_{PB}}{\text{RSSI}_{IN}}.$$  

By collecting enough of this kind of information, CLPL can quickly build the relationship between PRR and SINR. For a specific SINR value (e.g. idB), CLPL records and maintains all the PB reception results (successfully received or lost). According to these results, CLPL can compute the averaged PRR (i.e., $\text{PRR(idB)}$) of each SINR. Then, CLPL builds a PRR-SINR model and can update it through online measurement.

**5.2 Determination of Wake-up Frame Output Power**

To compute the optimal RF output power for wake-up frames, each node should accurately analyze the influence of wake-up frame transmission on all neighbors’ data and ACK frame receiving. For each node $\mathbf{Tx}$, all its neighbors within one hop communication range can exactly know the received signal strength indicator (RSSI) of wake-up frame transmitted by setting different output power. Note that CLPL is built upon the capacity of clear channel assessment (CCA) component. Once the state of CCA Pin indicates the channel is busy, CLPL periodically reads RSSI value by accessing the corresponding register.

We mark all neighbors of node $\mathbf{Tx}$ as $N_{\mathbf{Tx}} = \{n_1, n_2, \ldots, n_X\}$, where $n_i$ denotes a neighbor of node $\mathbf{Tx}$. When the output power of wake-up frame is set to $L_j$, the signal strength at neighbor $n_i$ is denoted as $\text{RSSI}_{\mathbf{Tx}}^{n_i}(L_j)$. In large-scale IoT deployment, each node is in charge of generating, receiving, and forwarding data/ACK frames for part of its neighbors, namely children/parent nodes. Hence, as neighbor of $\mathbf{Tx}$, $n_i$ could receive data/ACK from its children-parent set denoted as $\text{CP}_{n_i} = \{c_{p_i}^{n_i}, c_{p_i}^{n_i}, \ldots, c_{p_i}^{n_i}\}$ and the received signal strength of data/ACK frames from children/parent nodes is denoted as $\text{RSSI}_{n_i}(\text{CP}_{n_i}) = \{\text{RSSI}_{n_i}^{p_1}, \text{RSSI}_{n_i}^{p_2}, \ldots, \text{RSSI}_{n_i}^{p_m}\}$. For each power level $L_j$, the influence of $\mathbf{Tx}$’s wake-up frame transmission on $n_i$’s data/ACK frame decoding can be expressed as

Average Power Ratio (PAPR) is a common measure of the fluctuation of signal power. Peak to Average Power Ratio (PAPR) is a common measure of the fluctuation of signal power. As shown in Table ??, ZigBee adopts Direct Sequence Spread Spectrum (DSSS) which utilizes the entire frequency range to transmit data

6 WAKE-UP FRAME IDENTIFICATION

The use of wake-up frame is also to provide identifiable features for neighbor nodes to assess the feasibility of accessing shared busy channel. In this module, by continuously sampling channel states, CLPL exploits the spatial-temporal features of wake-up frame transmissions to identify them. The features contain specific on-air time, frame interval, strict periodicity, and the change of received signal strength.

6.1 Features of Wake-up Frames

The intentionally designed wake-up frames have four key features for distinguishing them from all types of data frames and interference signals.

On-air time: On-air time denotes the transmission period of an individual frame. Due to the different data rate and maximum frame size of different techniques, their on-air time is usually different. In CLPL, wake-up frames are intentionally generated by setting a specific frame size to distinguish them from all types of data frames. Take Zigbee as an example. The data frame size in CLPL is restricted to [30, 133] bytes, and the corresponding on-air time is [960, 4256] μs. Different to data frame, CLPL sets the size of wake-up frame to 20 bytes with 640 μs on-air time. The repeatedly transmitted wake-up frames carry the same payload, hence a series of wake-up frames are set to access media in concurrent way. If this condition is not satisfied, CLPL can effectively schedule multiple senders to access media in concurrent way. If this condition is not satisfied, TX is banned from accessing shared channel when it is busy.
Data frame features of common 2.4GHz technologies [39].

<table>
<thead>
<tr>
<th>Wireless technology</th>
<th>On-air time</th>
<th>PAPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZigBee</td>
<td>[576, 4256]µs</td>
<td>≤1.3</td>
</tr>
<tr>
<td>WiFi</td>
<td>[192, 542]µs</td>
<td>≥1.9</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>366µs</td>
<td>≤1.3</td>
</tr>
<tr>
<td>MWO</td>
<td>10ms</td>
<td>≥2.9</td>
</tr>
</tbody>
</table>

Fig. 10. Strict periodicity feature of wake-up frame transmission with different number of contenders.

so that its PAPR is relatively stable. PAPR can act as a supporting characteristics to identify external interference and enhance the wake-up frame identification.

**Strict periodicity:** Because the on-air time of wake-up frame and the interval between successive wake-up frames are completely fixed in CLPL, if the current channel is accessed by neighbor nodes just for wake-up frame transmissions for a period of time, it is certain that these states at this stage will occur with strict periodicity. The cycle length is the sum of the on-air time of wake-up frame and the time interval between wake-up frames, i.e. $T_{WF_{on\_air}} + T_{frame\_interval}$. For convenience, we refer it as $T_{off\_cycle}$.

### 6.2 Identification Procedure

To determine whether a sender is permitted to access a busy channel for data frame transmission, CLPL must continuously sample the RSSI of shared channel to figure out the current channel status. If the busy channel was just used for transmission of wake-up frames and this period ($T_{WF_{span}}$) can accommodate the sender’s data frame transmission by satisfying Eqn. (4), the sender can access the busy channel.

Note that if a busy channel is only caused by wake-up frame transmissions, the successively sampled channel RSSIs are with strict periodicity, with a period of $T_{off\_cycle}$ as mentioned above. For each cycle, CLPL uses a sampling window $W$ to record the continuously sampled RSSIs. The time span of $W$ is equal to $T_{off\_cycle}$. $W$ can be denoted as: $W = \{R_0, R_1, R_2, \cdots, R_n\}$, where $R_i$ denotes the $i^{th}$ sampled RSSI value. For two successive cycles of RSSI sequences, they can be expressed as two uninterrupted sampling windows $\{W^i, W^{i+1}\} = \{\{R^i_0, R^i_1, R^i_2, \cdots, R^i_n\},$ $\{R^{i+1}_0, R^{i+1}_1, R^{i+1}_2, \cdots, R^{i+1}_n\}\}$. Because of the strict periodicity feature, the successive sampling windows are with very similar changes of RSSI value. As illustrated in Fig. 10, regardless of how many senders are transmitting their wake-up frames, the sampled RSSIs periodically have the same varying tendency. On the contrary, if successive sampling windows are with different varying tendency of RSSI values, it is likely that the channel situation has changed caused by external interference, the arrival or departure of a neighboring sender.

Based on this characteristic, by first using the characteristics of on-air time and PAPR to distinguish data frame transmission [39], CLPL adopts Pearson correlation coefficient (PCC) [25] to compute the correlation between
two successive sampling windows. According to the property of PCC, if the successive two sampling windows are with very high correlation, it can indicate the two sampling windows are with the same varying tendency of RSSI values. Considering the strict periodicity of wake-up frame transmission, it is reliably to deem the two sampling windows are used for wake-up transmissions.

PCC is the covariance of the two variables \((W_i, W_{i+1})\) divided by the product of their standard deviations. It is represented by \(\rho\). The formula for \(\rho\) is:

\[
\rho(W_i, W_{i+1}) = \frac{\text{cov}(W_i, W_{i+1})}{\sigma_{W_i} \sigma_{W_{i+1}}},
\]

where \(\text{cov}(W_i, W_{i+1})\) is the covariance of two variables \(W_i\) and \(W_{i+1}\), and \(\sigma_{W_i}\) is the standard deviation of \(W_i\). The formula for \(\rho\) can be expressed in terms of mean and expectation,

\[
\rho(W_i, W_{i+1}) = \frac{E[(W_i - \mu_{W_i})(W_{i+1} - \mu_{W_{i+1}})]}{\sigma_{W_i} \sigma_{W_{i+1}}},
\]

where \(E\) is the expectation, and \(\mu_{W_i}\) is the mean of \(W_i\).

![Identification accuracy](https://example.com/identification_accuracy.png)

(a) Different values of \(\bar{\rho}\)

![Probability](https://example.com/probability.png)

(b) Identification accuracy: \(\bar{\rho}=0.7\)

Fig. 11. Determination of the value of \(\bar{\rho}\) and the identification accuracy.

The correlation coefficient ranges from -1 to 1. A value of 1 implies that a linear equation describes the relationship between \(W_i\) and \(W_{i+1}\) perfectly, with all data points lying on a line for which \(W_{i+1}\) increases as \(W_i\) increases. A value of 0 implies that there is no linear correlation between the variables. If \(W_i\) and \(W_{i+1}\) are two successive sampling windows and \(\rho(W_i, W_{i+1})\) is above a threshold \(\bar{\rho}\) which is larger than 0, we can credibly deem that the sampling windows \(W_i\) and \(W_{i+1}\) are just used for wake-up frame transmission. In the indoor testbed we conduct experiments to evaluate the relation between \(\bar{\rho}\) and the accuracy of identification. For each value of \(\bar{\rho}\), we schedule network nodes to transmit data frame, wake-up frame, and network beacon with different manner. All network nodes work in the 19\(^{th}\) channel which suffers from the coexisting WiFi interference. Then, we adopts PCC to identify the wake-up frame transmission according to the strict periodicity and compute the accuracy. Experimental results plotted in Fig. 11(a) show that by setting \(\bar{\rho} = 0.7\), the accuracy of identification reaches up to 96.5%. In our implementation of CLPL, \(\bar{\rho}\) is set to 0.7. Moreover, we also compute the probability of correct identification, false positive rate (denoted as FP) and false negative rate (FN), respectively. The false positive denotes that channel activity caused by other signals is falsely identified as wake-up frame transmission, and the false negative denotes that CLPL fails to identify wake-up frame transmission. As illustrated by Fig. 11(b), CLPL can accurately identify more than 92.8% wake-up frame transmissions no matter testbed networks suffer from (92.8%) or are free of the coexisting WiFi interference (95.6%). The results indicate that CLPL can effectively harness potential concurrency opportunities especially when networks are free of external interference. The false
positive rate is lower than 0.5% in this experiment, hence the probability that CLPL makes an incorrect decision for concurrent channel access is extremely low. Because of complex network state, the false negative of CLPL is respectively 6.7% (with WiFi interference) and 4% (without WiFi interference).

7 EVALUATION RESULTS

In this section, we conduct experiments to evaluate the performance of CLPL. We first introduce the experimental testbed and performance indicators (Section 7.1). We demonstrate the feasibility of online PRR-SINR model in Section 7.2. Then, we conduct experiments to qualitatively demonstrate the improvement of network capacity under different experimental scenarios (Section 7.3). After that, we respectively evaluate the data collection performance of CLPL from different aspects and compare it with the state-of-the-art LPL-based protocols, low power probe-based protocols [8], and opportunistic data forwarding protocol [15] (Section 7.4). The main parameter settings of CLPL are listed in Table ??.

7.1 Experimental Testbed and Performance Indicator

**Experimental testbed.** Our experiments are conducted in indoor testbeds with 40 Telosb nodes which is deployed on our 40 \times 70m^2 office as illustrated in Fig. 12. By setting different transmission power levels (RF output power) to testbed networks, nodes automatically form multi-hop networks with different densities. All experiments are conducted in the 19th Zigbee wireless channel which is overlapped with part of WiFi operating frequency used by the office APs. Unless mentioned otherwise, the packet size transmitted by all network nodes is randomly set between [40, 80] Bytes. Network nodes work in duty-cycled mode except sink node and the wake-up interval is set to 512ms.

**Baseline protocols.** In the evaluation of network capacity and the performance of data collection, we conducted extensive experiments on the testbed by comparing with the state-of-the-art low power MAC schemes, such as ContikiMAC, A-MAC, BoX-MAC, and opportunistic scheme ORW. (1) BoX-MAC is an enhanced CSMA-based protocol for duty-cycled networks, which means nodes in the interference range of current transmitters defer to send packets. The performance of BoX-MAC is regarded as a baseline in this work to measure how much improvement can be taken from concurrent channel access. (2) ORW is a traditional opportunistic forwarding
<table>
<thead>
<tr>
<th>System parameters</th>
<th>Value</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{EAP}$</td>
<td>23ms</td>
<td>Extended active period;</td>
</tr>
<tr>
<td>$T_{frame_interval}$</td>
<td>0.4ms</td>
<td>Inter-frame interval;</td>
</tr>
<tr>
<td>$T_{idle_wakeup}$</td>
<td>0.8ms</td>
<td>Minimal active period;</td>
</tr>
<tr>
<td>$T_{ack_waiting}$</td>
<td>0.4ms</td>
<td>ACK waiting time;</td>
</tr>
<tr>
<td>$T_{frame_cycle}$</td>
<td>18ms</td>
<td>Data frame cycle time;</td>
</tr>
<tr>
<td>$T_{WF_on_air}$</td>
<td>0.64ms</td>
<td>Wake-up frame transmission time;</td>
</tr>
<tr>
<td>$T_{PB_on_air}$</td>
<td>0.64ms</td>
<td>Power test beacon transmission time;</td>
</tr>
<tr>
<td>$T_{wf_cycle}$</td>
<td>1.04ms</td>
<td>Wake-up cycle time;</td>
</tr>
<tr>
<td>$\bar{\rho}$</td>
<td>0.7</td>
<td>Threshold of PCC;</td>
</tr>
<tr>
<td>$T_{NPB_interval}$</td>
<td>0.74ms</td>
<td>Interval of normal power test beacon;</td>
</tr>
<tr>
<td>$T_{max_backoff}$</td>
<td>0.3ms</td>
<td>Maximum backoff time.</td>
</tr>
</tbody>
</table>

protocol by exploiting candidate forwarders to forward data packets. ORW is built upon BoX-MAC and it ignores all potential concurrency opportunities. (3) A-MAC is the state-of-the-art receiver-initiated link layer protocol for low-power wireless networks. A-MAC can provide the most energy efficiency compared with other receiver-initiated link layer protocols, because A-MAC adopts phase-lock mechanism for energy saving. (4) ContikiMAC was proposed to provide extremely outstanding energy efficiency by adopting a fast sleep and phase lock mechanism. Compared with the current implementation of BoX-MAC, ContikiMAC can provide better energy efficiency performance.

**Metrics.** In the following sections, we use packet delivery ratio as the indicator of network reliability. The energy consumption is measured as duty cycle, the portion of radio-on time, as a platform-independent metric for energy efficiency. Besides, we use single-hop transmission time to approximate delivery latency, and use single-hop transmission count to indicate data collision caused by concurrent transmission because aggressively exploiting concurrency opportunities could bring about more serious network interference. 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 sender receives an ACK.

### 7.2 Performance of Online PRR-SINR Model

In this section, we conduct experiments to demonstrate the accuracy of online PRR-SINR model.

We first run CLPL in our indoor testbed for two hours, which is a long enough period for generating a stable online PRR-SINR model by exploiting power test beacons. Then, we randomly select two pairs of sender-receiver from the testbed network by guaranteeing each receiver is in the communication range of the two sender. Hence, each receiver can accurately measure the RSSI of both its sender and the other sender. If the two sender transmit in concurrent way, it is easy for each receiver to compute the actual SINR, which is marked as $\overline{\text{SINR}}$ to distinguish its online SINR. Then, through wire cables, we reconfigure the selected senders to work in CSMA-disabled mode which means senders can access shared channel without any limitation, and reconfigure the receivers’s radio state to be always on. The senders will continuously generate data packets and transmit them to corresponding receivers. Receiver will report the sequence number of received packet to central computer. Then we can compute the actual PRR (marked as $\overline{\text{PRR}}$) of each pair of sender-receiver under the influence of the other sender. In addition,
according to online PRR-SINR model, we can get the theoretical PRR corresponding to the given $\overline{\text{SINR}}$. We form them as $<\overline{\text{SINR}}, \overline{\text{PRR}}, \overline{\text{PRR}}>$ triple. By adjusting senders’ transmitting power level and selecting different pairs of sender-receiver to repeat this experiment, we get a total of 538 $<\overline{\text{SINR}}, \overline{\text{PRR}}, \overline{\text{PRR}}>$ triples. We further compute the gap between the actual $\overline{\text{PRR}}$ and the theoretical PRR according to

$$\text{Gap}(\overline{\text{SINR}}) = |\overline{\text{PRR}} - \overline{\text{PRR}}|.$$ 

For each SINR ranging from 0 to 10dB, the minimum, 25th percentile, median, 75th percentile, and maximum of gaps are plotted in Fig. 13(a). We also plot the relationship between an actual $\overline{\text{SINR}}$ and the averaged value of corresponding $\overline{\text{PRR}}$s in these triples in the figure. As illustrated by the figure, when SINR is larger than 4dB, the averaged PRR gap stabilizes around 2.4%, which is comprehensively demonstrated by the cumulative distribution of all PRR gaps plotted in Fig. 13(b). The experimental results demonstrate that the actual relationship between SINR and PRR closely approximates the online PRR-SINR model. CLPL can use the online PRR-SINR to accurately compute an optimal power level for wake-up frame transmission.

![Fig. 13. Accuracy of online PRR-SINR model.](image)

### 7.3 Network Capacity

In this section, we conduct experiments to quantify the expected benefit of concurrent transmission in contention-tolerant way by using CLPL. The performance of CLPL is evaluated by two ways: i) the instantaneous throughput of multiple senders locating within one-hop communication range; ii) network-wide capacity that denotes the instantaneous throughput of all network nodes.

Before this experiment, we use CTP [10] to repeatedly construct tree-based topology in the indoor testbed in which all nodes send data to sink node. Note that testbed nodes are all controlled by a central computer through wire cables. We can control each node and collect logging information through central computer. Then, according to the statistical tree-based topologies, we adopt a fixed routing strategy by setting an “optimum” (most frequently used) receiver to each node. The packet size is randomly set between [40, 80]Bytes.

#### 7.3.1 One-hop Instantaneous Throughput

We construct single hop networks from the testbed by selecting multiple senders to transmit data packets to their receivers. The wake-up interval is set to 512ms, and the inter-packet interval (IPI) of each sender is also set to 512ms to ensure the occurrence of burst traffic load. The number of senders is selected from 3 to 10 in the testbed. Some selected senders may share the same receiver. By respectively adopting BoX-MAC, A-MAC, ContikiMAC, ORW and CLPL, we record the number of data packets that are successfully transmitted by the selected senders. Then, we quantify the instantaneous throughputs of these senders located within one-hop communication range by counting the total packets received by their
receivers during a period of 5 seconds windows. We have eliminated all duplicate packets before computing the instantaneous throughput. The theoretical maximum throughput of each window ranges between \([30, 100]\) packets (corresponding to \([3, 10]\) senders) in this experiment. Each evaluation is run for 10 minutes and repeated no less than 10 times. We respectively compute the instantaneous throughput of BoX-MAC, A-MAC, ContikiMAC, ORW and CLPL, and further compute the ratio of the instantaneous throughput of CLPL to that of BoX-MAC, A-MAC, ContikiMAC, and ORW. We refer to the ratio as increase rate. We plot the distribution of increase rate in Fig. 14. The instantaneous throughput of CLPL is averagely 3-5 times of BoX-MAC, 1.8-3.9 times of ORW, 2-6.4 times of ContikiMAC, and 2.2-6.1 times of A-MAC under one-hop scenarios. Due to the data frame transmission cycle is limited to 18ms, the maximum increase rate is limited to 7.

7.3.2 Network-wide Capacity. Besides the one-hop instantaneous throughput, we also evaluate the network-wide capacity of all network nodes. In this experiments, all network nodes are selected as both sender and receiver. The parameters on data transmission are the same as above experiment. According to the collected information through wire cable, we further compute the number of packets successfully transmitted by all network nodes and received by the corresponding receivers during a period of 5 seconds windows. In addition, we also compute the average retransmission count, and single-hop delay of each packet. The experimental results are listed in
Fig. 15. Data forwarding performance of CLPL and BoX-MAC in tested networks with different traffic load. Each node has about 6 neighbors on average.

Table ??: The network capacity of CLPL is almost 3 times of BoX-MAC, A-MAC, and ContikiMAC, 1.8 times of opportunistic forwarding protocol ORW, and the average transmission delay can be significantly reduced.

7.4 Performance of Data Collection Networks

In this section, we further conduct experiments to evaluate the data forwarding performance of CLPL in terms of delivery ratio, network retransmission efficiency, latency, and energy efficiency in testbed networks with different traffic load. Besides, we also compare CLPL with BoX-MAC, A-MAC, ContikiMAC, and ORW by setting the same network configuration.

To evaluate the effects of network load on performance, we construct testbed networks with CTP+BoX-MAC, CTP+CLPL, CTP+A-MAC, CTP+ContikiMAC and CTP+CLPL by setting different traffic patterns. For each protocol, we respectively set all network nodes’ IPI to 16 seconds (s), 32s, 64s, 128s, and 256s. Besides, we also construct burst traffic load by selecting 10 senders in testbed network to continuously produce data packets lasting for 5 minutes. In this case, the IPI of selected 10 nodes is respectively set to 2s, 4s, and 8s. When a new data packet is generated and the previous data packet hasn’t been successfully delivered, the new packet is put into sending buffer. In this experiments, nodes’ wake-up interval is set to 512ms and RF output power is set to level 5 [11]. Then, the average number of neighbors of each network node is about 9. For each traffic load, we respectively run experiments with the two suits of protocols, and compute the average performance of the 10 times of running as result.
**Delivery ratio.** Fig. 15(a) compares the delivery ratio of the two protocols with different traffic load. Note that the values 2, 4, and 8 on horizontal axis denote testbed networks with burst traffic pattern. We can see that in all scenarios with burst traffic load, the delivery ratio of CLPL significantly outperforms BoX-MAC, A-MAC, ContikiMAC. By increasing traffic load (the IPI of selected neighboring nodes decreases from 8s to 2s), the delivery ratio of BoX-MAC decreases from 98% to 73%, and similar improvement for other protocols. In contrast, the delivery ratio of CLPL has always been kept over 99.4%. The outperformed delivery ratio of CLPL is mainly attributed to the supporting of concurrent and contention-tolerant transmission mode. The exploiting of concurrency opportunities can reduce the blind waiting time when neighboring nodes are transmitting in asynchronous duty-cycled networks. The significant decreasing of delivery ratio of BoX-MAC, A-MAC, ContikiMAC, and ORW in high traffic load is due to the inefficient channel utilization. Compared with them, the advantage of supporting concurrent channel access in CLPL can significantly improve channel utilization.

**Delivery latency and transmission count.** With the increasing of pending data packets in the network, it brings about more serious intra-network interference and directly results in more data collisions and retransmissions. Fig. 15(b) compares the average one-hop transmission count of each packet. We can see that in the scenarios with bursty traffic load (i.e., IPI less than 16s), CLPL has transmission count of 8.7–20.4 percent lower than BoX-MAC, A-MAC, ContikiMAC, and ORW, which indicates that CLPL can efficiently empty the in-network data packet compared with other baseline protocols. However, with the decreasing of traffic load, due to the inherent intra-network interference caused by wakeup frame transmission, the average transmission count of CLPL is slightly higher than the ORW and BoX-MAC.

The averaged delivery latency of CLPL and other protocols under the scenarios with different traffic loads are illustrated in Fig. 15(c). From this figure, we can see that the average one-hop delivery latency of CLPL is always smaller than BoX-MAC, A-MAC, ContikiMAC, and ORW when the network is with burst traffic load (IPI less than 16s). BoX-MAC and CLPL have decreasing latency as traffic load become high because it may increase the probability that a sender’s receiver could be awake during the sender’s transmission phase. Compared with the increasing of receiver’s active period, the increasing of channel contention resulting from burst traffic has a greater impact on delivery latency. By adopting lock-phase mechanism, although A-MAC and ContikiMAC can achieve extreme energy efficiency, they can’t harness the potential opportunities of extended active period at receivers. Hence, with the increasing of traffic load, the increasing channel contention makes the one-hop transmission latency significantly increasing.

On the other hand, with the decreasing of traffic load (the IPI is no less than 16 seconds), the probability that the busy traffic makes potential receivers in active period declines gradually. Even though, the harnessing of concurrency also makes the delivery latency of CLPL lower than the other protocols (i.e., BoX-MAC, A-MAC, ContikiMAC) without harnessing the opportunistic forwarders. The delivery latency of ORW is far superior to all other protocols because it can fully exploit the potential forwarders to reduce transmission latency.

**Energy consumption.** Fig. 15(d) illustrates the duty cycle of BoX-MAC, A-MAC, ContikiMAC, ORW and CLPL. We can see that CLPL significantly outperforms them in energy efficiency when unexpected surge of traffic occurs (i.e., IPI of selected nodes is less than 16s). This mainly results from full utilization of shared channel. Both of them can significantly reduce data delivery latency and then reduce the radio on time. With the decrease of traffic load (the IPI of network nodes increases from 16s to 256s), ContikiMAC and A-MAC is much more energy-efficient because they adopt the phase-lock mechanism to avoid unnecessary active period and quickly go to sleep for extremely energy saving. They work well in low data traffic networks. However, in burst traffic network, the failure of the phase-lock mechanism and the using of fast sleeping make the performance of ContikiMAC significantly lower than that of CLPL. Even though, CLPL is very close to ContikiMAC and A-MAC on energy efficiency with low traffic load.
In short, CLPL has an advantage over the well-applied BoX-MAC, A-MAC, ContikiMAC, and ORW in networks with different traffic loads. With increase of traffic load, CLPL significantly outperforms it by supporting concurrent and contention-tolerant transmission mode.

8 RELATED WORKS

Concurrent transmission is a well-known concept employed in wireless communications to enhance channel utilization by improving spatial reuse. It is crucial to performance of data forwarding protocols, concerning both traditional unicast protocols or opportunistic protocols. Researchers have proposed a lot of methods and mechanisms to exploit concurrency in wireless networks. Here we have a brief discussion on the existing methods in three aspects respectively.

Collision tolerance: In wireless communication community, capture effect has been a well known phenomenon for collision tolerance and various capture models have been proposed. Different from collision avoidance, the idea of capture effect allows collisions. Glossy [9], Chase [3], Chase [3], and Pando [6] were subsequently proposed for efficient data transmission exploiting capture effect in flooding scenarios. They detect and recover packets from collisions taking advantage of capture effect, whereby a packet with the strongest signal strength can be received in spite of a collision. These techniques using capture effect are heavily dependent on high precise time synchronization. In resource-restricted and duty-cycled wireless networks, considering network dynamics, it is not cost-effective to achieve this level time synchronization by paying considerable energy consumption.

PRR-SINR model: Physical interference model is another effective way for improving channel utilization in wireless networks. Son et al. [37] and Sha et al. [34] studied the PRR-SINR model in sensor networks and showed the modeling accuracies and impacts on link scheduling performance. In particular, it is shown that adopting the PRR-SINR model can lead to significant link throughput improvement. Reis et al. [30] presented interference and packet delivery models that can be instantiated by packet transmission traces. Qiu et al. [28] proposed a general interference model to characterize the interference among arbitrary number of 802.11 senders and predict the resultant throughput. In [14], a measurement-based approach is proposed to model the interference and link capacity in 802.11 networks. Aguayo et al. [1] experimentally studied the effect of SINR on the causes of packet loss in a 802.11 mesh network (Roofnet). Different to existing techniques, CLPL proposes tailor-made beacon to generate PRR-SINR model through online measurement in resource-restricted ad-hoc networks.

Conflict relationship: Conflict graph has been used to model wireless interference between neighboring nodes. Conflict graph provides a simplified description of the interference status, which greatly eases the design of channel assignment/spectrum allocation algorithms, and consequently gives birth to a series of highly efficient wireless network optimization algorithms [13][17].

Existing works can be divided into two categories based on the type of conflict graphs they use. The first category uses per-link signal measurements to capture interference conditions among individual links, using either active measurements [18][38] or passive measurements [38][4]. These link-based conflict graphs are for immobile networks where transmission links are known a priori. The second category of works builds coverage-based conflict graphs based on propagation models [31][41]. Zhou et al. [41] used real-world measurements to evaluate the conflict graph accuracy of coverage-based conflict graph. In resource-restricted ad-hoc networks, CMAP [38], NoPSM [4], and COF [17] use conflict relationship to exploit concurrency opportunities for improving network performance.
9 DISCUSSIONS

Impact of node diversity. The difference of node models may bring about the difference of receiving sensitivity and hence lead to asymmetry of bidirectional links. Even so, the influence of node diversity is not serious, because this physical characteristics will act on the quality of wireless links that connect different nodes. By dynamically adjusting the link quality, the network layer protocol (e.g., CTP) for maintaining network topology will update the connection relationship between each pair of nodes. Through this cross-layer interactions and updates, the negative effects of node diversity can be eliminated. Furthermore, the self-adjustment capability of CLPL can also directly resolve the influence of node diversity. For instance, if a node’s receiving sensitivity is extremely low, it may may not be aware of its neighboring nodes’ wake-up frame transmissions and even data transmissions. As a result, it will immediately schedule the transmission of data frame and wake-up frame, which could interfere the neighbor’s transmissions. In this case, the neighbor can detect the chaotic transmission patterns and suspend its current transmission, and then it reattempts to transmit by sensing the channel situation.

Setting of network parameters. CLPL is built upon TinyOS, an open source, BSD-licensed operating system designed for low-power wireless devices. The parameters listed in Table ?? are a suite of achievable parameters for Telosb nodes to run CLPL. The setting of these parameters depends on the flying time of data frame (variable), ACK turnaround time, backoff time, and the maximum possible number of senders allowed to run concurrently, etc. Acutally, we haven’t proved that it is the optimal. To avoid misunderstanding, we explicitly state that this set of parameters are not the only one that works, but it really can make CLPL workable.

Side effect. CLPL is proposed to fully turn exposed terminals into transmission opportunities. Although CLPL can address data collision problem very well between concurrent senders. However, the significantly increased transmission opportunities in whole network could also increase hidden terminal problems, which could further impact both data forwarding and ACK reception. Hidden terminal is an inherent problem in wireless networks, which is left for future work.

10 CONCLUSION

In this paper, we propose a generic concurrent low power listening mechanism CLPL supporting concurrent and contention-tolerant data transmission service for multiple senders. The basic idea of CLPL is that rather than continuously and repeatedly transmitting the same data frame, a sender completes a data transmission by periodically alternating a data frame and a series of intentionally constructed signals. The series of intentional signals are designed to rendezvous with receiver’s waking moment and transmitted by setting a well-designed low output power so that they won’t interfere neighbors’ data frame transmission. On that basis, as new sender, if the captured channel state indicates the channel is free or only used for the transmission of intentional signals, it immediately schedules data frame transmission over the free channel or overlapping with the identified intentional signals, rather than repeating the same carrier sense-defer process. In this way, CLPL can easily realize concurrent data transmission for multiple neighboring senders without compensating with any other performance indicators.

Extensive testbed experiments demonstrate that CLPL can significantly improve network capacity by supporting the concurrent channel access mechanism. Compared with the state-of-the-art LPL-based protocols, CLPL can improve the one-hop data delivery performance by 2-6 times.

ACKNOWLEDGMENTS

We sincerely thank the editor and anonymous reviewers for insightful comments to improve our work. This work was partially supported by NSFC with Nos. 61902122, U20A20181, U19A2067, 61732017, 62172153, 62072167, 62072167.
Hunan Provincial Natural Science Foundation of China (Nos. 2020J5089 and 2020J5082), Key Research and Development Project of Hunan Province of China (Nos. 2022GK2020 and 2021WK2001), Changsha Municipal Natural Science Foundation (No. KQ2014057), and Zhejiang lab (No. 2021LC0AB05).

REFERENCES


