This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

Age of Information in Physical-Layer Network Coding Enabled Two-Way Relay Networks

Haoyuan Pan, *Member, IEEE,* Tse-Tin Chan, *Member, IEEE,*
Victor C. M. Leung, *Life Fellow, IEEE,* Jianqiang Li, *Member, IEEE*

Abstract—This paper investigates the information freshness of two-way relay networks (TWRNs) operated with physical-layer network coding (PNC). Information freshness is quantified by age of information (AoI), defined as the time elapsed since the generation time of the latest received information update. PNC reduces the communication latency of TWRNs by turning superimposed electromagnetic waves into network-coded messages so that end users can send update packets to each other more frequently via the relay. While sending update packets more frequently has the potential to reduce AoI, how to handle packet corruption in TWRNs has not been investigated. Specifically, if an old packet is corrupted in any hop of a TWRN, one needs to decide whether to drop or to retransmit the old packet, e.g., a new packet has more recent information but may take more time to be delivered. Therefore, we study the average AoI with and without automatic repeat request (ARQ) in PNC-enabled TWRNs. Interestingly, our analysis shows that neither the non-ARQ scheme nor the pure ARQ scheme achieves a good average AoI. Hence, we put forth an uplink-lost-then-drop (ULTD) protocol that combines packet drop and ARQ. Experiments on software-defined radios indicate that ULTD significantly outperforms non-ARQ and pure ARQ schemes in terms of average AoI, especially when the two end users have imbalanced channel conditions. We believe the insight of ULTD on TWRNs generally applies to other two-hop networks: to achieve high information freshness, when packets are corrupted in the first hop, new packets should be generated and sent (i.e., old packets are discarded); when packets are corrupted in the second hop, old packets should be retransmitted until they are successfully received.

Index Terms—Age of information (AoI), automatic repeat request (ARQ), information freshness, physical-layer network coding (PNC)

1 Introduction

In recent years, age of information (AoI) has been recognized as a key performance metric for measuring information freshness in next-generation communication networks [1]–[3]. In many real-time monitoring and control systems, such as traffic monitoring in vehicular networks [4] and motion control in the industrial Internet of Things (IIoT) [5], timely delivery of regular and frequent information updates is crucial because correct decision making or precise control relies on real-time data. Prior studies have shown that conventional performance metrics, such as information rate and packet delay, are not appropriate to quantify information freshness in emerging timely information update systems [1], [2].

AoI is defined as the time elapsed since the generation time of the latest received information update at the destination [1]–[3]. Precisely, AoI captures both the delay and the generation timestamp of each information update. If at time t, the latest information update received by the receiver was an update packet generated by the source at time t', then the instantaneous AoI at time t is $t - t'$. Since AoI characterizes the effect of packet delay from the perspective of the destination, it is quite different from the conventional delay or latency metrics. For example, prior works have studied the "average AoI" in various networks, defined as the time average of the instantaneous AoI over a long period [2]. The analyses and optimizations of average AoI have shown that the optimality conditions for AoI often do not coincide with those for conventional metrics such as throughput and latency [1].

In a typical information update system setup, two end users need to send their latest update packets to each other over a wireless medium, in which they could be far away from each other (i.e., lacking a direct link). A dedicated relay is employed to extend the network coverage and help forward the update packets from both end users, as shown in Fig. 1. Such a network is known in the literature as a two-way relay network (TWRN) [6]. TWRN-type information update systems can be found in many scenarios. For example, in the emerging vehicle-to-everything (V2X) networks, two distant vehicles send periodic basic safety messages (BSM) to each other with the help of a road-side unit (RSU) as a relay [7]. The BSM generated by each vehicle usually contains immediate status information about the vehicle, such as velocity, direction, acceleration, and position. Since the periodic exchange of BSMs creates mutual awareness of the surroundings, it is crucial to receive fresh BSMs to reduce the risk of road accidents.

Although the relay helps to extend the network coverage and forms a TWRN, the additional hop increases the communication delay of two-way communication. The traditional store-and-forward relaying scheme requires a total of four time slots for the two end users to deliver a packet to each other (i.e., each user spends two time slots to send a packet to the other user) [6]. Physical-layer network coding (PNC) is a key technique to reduce

- H. Pan, Victor C. M. Leung, and J. Li are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China (e-mails: hypan@szu.edu.cn, vleung@ieee.org, lijq@szu.edu.cn).
- T.-T. Chan is with the Department of Computing, The Hang Seng University of Hong Kong, Hong Kong (e-mail: ttchan@hsu.edu.hk).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

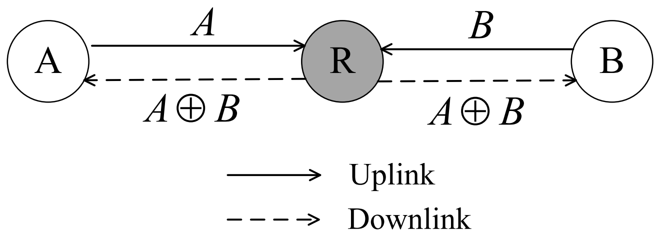


Fig. 1. A PNC-enabled two-way relay information update system, where two end users A and B want to send update packets to each other with the help of a relay.

communication latency and increase network throughput [6], [8]. PNC turns the mutual interference (superimposed signals) from simultaneously transmitting users into useful network-coded information. To see this, let us focus on a PNC-enabled TWRN shown in Fig. 1. Compared with traditional store-and-forward relaying, PNC reduces the total number of time slots for exchanging two packets from four to two. Specifically, the first time slot is an uplink phase in which the two end users simultaneously send packets to the relay. In the second time slot, the relay performs PNC decoding on the superimposed received signals and broadcasts back a network-coded packet to the end users in the downlink phase [6], [8]. We refer to the network-coded packet as a PNC packet, an eXclusive-OR (XOR) of the two source packets from the end users. Upon receiving the PNC packet, the two end users subtract their self-packet from the PNC packet to obtain the packet from the other user. As a result, PNC halves the transmission time and doubles the throughput of a TWRN [8].

In the context of information update systems, this paper considers the generate-at-will model, where the latest information about the observed phenomenon can be sampled whenever the source has an opportunity to send [2]. Since PNC reduces the time for one user to receive packets from the other user, end users can sample and send update packets to each other more frequently. When packet transmission is successful in every hop, more frequent updates result in higher information freshness (i.e., lower AoI). However, in practice, update packets are typically short. Information theory reveals that with finite block lengths, the packet error rates (PERs) cannot go to zero [9]. Although PNC is promising to achieve high information freshness, it has not been well investigated when update packets are corrupted in the uplink or downlink phase. In particular, how to deal with corrupted packets becomes a critical issue to achieve low AoI, especially in TWRNs with two hops. This paper is an attempt to fill this research gap.

Traditional wireless communication systems use automatic repeat request (ARQ) to ensure reliable transmission through packet retransmission [10]. That is, the receiver sends feedback to the transmitter. If a packet is corrupted, the same packet is retransmitted until it is successfully received. However, applying ARQ directly to information update systems may lead to high average AoI, since the number of retransmissions could be arbitrarily large. Prior studies on single-hop networks have indicated that a non-ARQ scheme, which sends a new packet immediately once the old packet got corrupted, outperforms the classical ARQ scheme significantly [11]. This is because new packets always have more recent information (and old packets become obsolete once new packets are generated). However, in two-hop TWRNs, new packets may require more time to be delivered, e.g., when a packet gets corrupted in the second hop, the transmission has to restart from the first hop. Therefore, a quantitative investigation is needed to study the relative merits between ARQ and packet drop.

In this paper, we first investigate the average AoI of a PNC-enabled TWRN under two protocols, namely, once-lost-then-drop (OLTD) and reliable packet transmission (RPT). In OLTD, once a packet gets corrupted in either the uplink or downlink phase, both end users generate and send new packets to the relay immediately, i.e., a non-ARQ scheme without packet retransmission. By contrast, in RPT, link-by-link ARQ is used in both the uplink and downlink transmissions to ensure that no packets are lost.

Interestingly, unlike single-hop networks, our theoretical analysis shows that OLTD improves little over RTP in error-prone environments. This indicates that neither the non-ARQ protocol (OLTD, in which old packets are always dropped) nor the pure ARQ protocol (RPT, in which old packets are always retransmitted) achieves a low average AoI in TWRNs. This observation is different from that in single-hop networks [11]. The problem with OLTD is that the PNC packet decoded at the relay is sent only once in the downlink, regardless of the decoding result. If the downlink packet is corrupted, retransmission once could have helped recover it (and hence have a successful update instantly), but OLTD blindly restarts from the uplink phase. Therefore, to reduce the average AoI, we put forth an uplink-lost-then-drop (ULTD) protocol that exploits the advantages of OLTD and RPT. More specifically, in the uplink phase, when the relay fails to decode the PNC packet, old packets are dropped, and the two end users send new packets to the relay afterward. In the downlink phase, ARQ is used to ensure reliable transmission. Hence, ULTD strategically leverages both packet drop and ARQ to enhance the information freshness of TWRNs.

For performance evaluation, we compare the average AoI of different protocols theoretically and experimentally. We first use Gallager's random coding bound (RCB) to estimate the PERs of short packets with different block lengths (i.e., the PERs of both uplink and downlink in TWRNs). We compare the average AoI of the three protocols under different block lengths and signal-to-noise ratios (SNRs). Our theoretical analysis shows that in error-prone wireless networks, the average AoI performance of ULTD is significantly better than that of RPT and OLTD. For concept proving in real-world environments, we conduct real experiments on software-defined radios (SDR). When the two users have the same channel condition with respect to the relay, our experimental results show that ULTD reduces the average AoI by 22% and 34% compared with OLTD and RPT, respectively, when the SNR is $7.5\ dB$ under practical medium access control (MAC) designs of the three protocols. Furthermore, when the two users have imbalanced channel conditions, the performance improvement of ULTD is more pronounced, indicating that ULTD is more robust against time-varying channel conditions in practice. Overall,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

thanks to ARQ and packet drop integration, ULTD is a preferable protocol for achieving high information freshness in TWRNs operated with PNC.

To sum up, we have three major contributions:

(1) We study PNC-enabled TWRNs with AoI requirements. Specifically, we are the first to investigate ARQ protocols in TWRNs to handle corrupted packets in uplink and downlink transmissions, aiming at high information freshness.

(2) We design an uplink-lost-then-drop (ULTD) protocol for PNC-enabled TWRNs. In particular, ULTD leverages both packet drop (dropping old packets and sending new ones in the uplink) and ARQ (retransmitting old packets until they are successfully received in the downlink), which is a key principle for achieving low AoI in error-prone TWRNs with two hops. We believe the insights from ULTD are generally applicable to other two-hop networks.

(3) We demonstrate the practical feasibility of ULTD in a practical SDR environment. Our experiments show that ULTD outperforms both the classical ARQ scheme and the non-ARQ scheme in terms of average AoI, especially when the channel conditions between the two end users and the relay are imbalanced. In other words, ULTD is more robust to time-varying channels in practical applications. Furthermore, our experimental investigation confirms that the optimal scheme for AoI often does not coincide with that for conventional metrics, such as throughput and latency, in a PNC-enabled TWRN setting.

2 Related Work

Age of information (AoI), as a new performance metric, has attracted considerable research interests in recent years [1], [2]. It was first proposed to characterize the timeliness of safety packets in vehicular networks [4]. Later, AoI was studied under different communication and network models. Prior AoI works focused more on the upper layers of the communication protocol stack (i.e., above the physical (PHY) and MAC layers). We refer the readers to the monograph [3] and the references therein for important research results. For example, early works focused on analyzing the AoI performance of different systems modeled by various abstract queueing models, in which information update packets arrive at the source node randomly following a memoryless Poisson process [3], [12]–[14]. Different scheduling policies [15]–[19] are then examined with the aim of minimizing different AoI metrics, such as average AoI [3], peak AoI [20], bounded AoI [21], etc. By contrast, our work does not study queueing models: there is no queue at the source nodes. We adopt a generate-at-will model, wherein the source node will make measurements and generate an update packet only when it has the opportunity to transmit [2], [22].

Moving down to the PHY and MAC layers, imperfect updating channels, such as noisy channels with transmission errors, were studied in various wireless information update scenarios [22]–[24]. Previous works optimized the average AoI by using packet retransmission to deal with the wireless impairments, including both ARQ and hybrid ARQ (HARQ) [11], [25]–[28]. In particular, recent studies have shown that packet preemption has a significant advantage in terms of average AoI, i.e., dropping and replacing old packets with new ones, which always contain more up-to-date information [29]. For example, [26] proposed a truncated ARQ scheme where the same old update packet is sent only a limited number of times until the maximum allowed number is reached. Moreover, advanced coding schemes were proposed to combine old packets and new packets into the same transmission as a form of ARQ or HARQ. When an old packet is retransmitted, a new packet is embedded in the old packet using different methods, e.g., joint packet coding is used in [27], [30]. In order to improve information freshness, dropping old packets properly was shown to be an effective way to balance the error correction of old packets and fast decoding of new packets in [27]. Inspired by these previous studies, this paper considers both packet drop and ARQ to lower the average AoI of TWRNs.

Despite the above efforts, only a few AoI studies have been dedicated to networks beyond one hop. In particular, the impacts of packet drop and ARQ schemes on AoI in each hop have not been well investigated. Ref. [31] abstracted a large multi-hop network as a directed graph and developed scheduling policies to achieve age-optimality or near age-optimality with a single information flow; [32] studied AoI and throughput optimization in routing-aware multi-hop networks. However, they did not focus on PHY or MAC layers. Refs. [33], [34] considered the age and energy tradeoff in a one-way multi-relay network with two hops. They modeled the PHY-layer PER using the short-packet theory and studied the relay selection strategy. Nevertheless, no packet drop or ARQ is studied in [33], [34]. In addition, unlike the single information flow in [31], [33], [34], the TWRN model considered in our work consists of bidirectional information flows, i.e., two end users exchange update packets with the help of a relay.

Very recently, [35], [36] investigated the potentials of applying PNC to information update systems. The first application of PNC to reduce the average AoI was studied in a non-orthogonal multiple access (NOMA) network presented in [35]. The authors showed that by combining PNC with multiuser decoding techniques, the decoded PNC packets significantly reduce the average AoI compared with conventional orthogonal multiple access (OMA). However, the results of [35] cannot be directly applied to the current work because [35] studied a single-hop network without relays. Ref. [36] evaluated the average AoI of TWRNs where the relay adopts an amplify-and-forward (AF) strategy. The received superimposed signals are amplified at the relay and forwarded to end users. The end user makes use of the amplified signals and its own update packet to decode the other user's update packet. By contrast, our work employs a decode-and-forward (DF) strategy where the relay tries to decode PNC packets explicitly. Hence, unlike DF, AF does not involve a link-by-link ARQ study due to its simplicity. For DF, we are the first to investigate how to deal with the corrupted packets to enhance the information freshness of TWRNs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

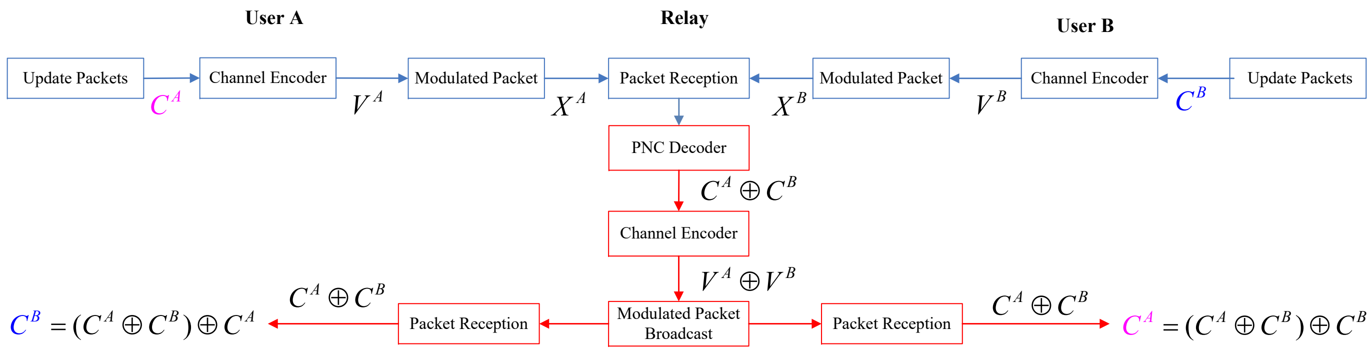


Fig. 2. The general architecture of information processing at the end nodes and the relay in a PNC-enabled TWRN.

3 Preliminaries

3.1 System Model

We consider a TWRN where two end users A and B want to send information update packets to each other with the help of a relay, as shown in Fig. 1. In an information update system, each end user (say, user A) wants to receive the update packets from the other end user (say, user B) as timely as possible. The timeliness is characterized by age of information (AoI) that will be formally defined in Section 3.2.

In a TWRN operated with PNC, only two time slots are required for the two end users to communicate with each other via a relay. As shown in Fig. 1, the first time slot is the uplink transmission phase from both end users to the relay, and the second time slot is a downlink transmission phase from the relay to the end users. In time slot 1, as illustrated in Fig. 2, the two users generate their respective packets C^A and C^B, which are then channel-coded into V^A and V^B, respectively. When the two users transmit to the relay simultaneously[1] , the relay receives superimposed signals. A PNC decoder attempts to decode a linear combination of the two packets C^A and C^B from the superimposed signals, i.e., a PNC packet $C^A \oplus C^B$ is the eXclusive-OR (XOR) of C^A and C^B. In time slot 2, the relay broadcasts $C^A \oplus C^B$ to both end users. An end user can recover the packet of the other user with the help of the received PNC packet $C^A \oplus C^B$ and its own packet. For example, user A can obtain packet C^B because $C^B = (C^A \oplus C^B) \oplus C^A$.

Throughout this paper, we adopt the XOR-channel decoding (XOR-CD) approach for PNC decoding in the uplink [8]. XOR-CD works and is easy to be implemented because it exploits the linearity of linear channel codes such as convolutional codes. Specifically, if we define $\Gamma(\cdot)$ as the convolutional encoding operation, we have

$$\Gamma\left(C^A \oplus C^B\right) = \Gamma\left(C^A\right) \oplus \Gamma\left(C^B\right) = V^A \oplus V^B. \quad (1)$$

As such, in the XOR-CD decoder, the received superimposed signals are first passed through a PNC demodulator to obtain bit-wise XOR information, i.e., XOR bits $V^A \oplus V^B$. Then, these XOR bits are fed to a standard Viterbi decoder (as used in conventional 802.11 WLAN systems) to decode the PNC packet $C^A \oplus C^B$. The XOR bits can be soft bits or hard bits. This paper considers soft bit-wise XOR information, and the detailed computation of the soft bits can be found in the Appendix of this paper and [8].

Compared with the traditional non-network-coded relaying scheme, PNC reduces the number of transmission time slots from four to two for the exchange of two packets between user A and user B [8]. Although PNC is well-known for reducing the communication latency of TWRNs, its merits have not been well investigated when the system performance metric is information freshness. Next, we introduce the AoI metric to evaluate information freshness in this paper.

3.2 Age of Information (AoI)

We consider the update packets sent by user A. In practice, to monitor physical quantities, a user may generate a time-stamped packet to report its status at the current time. At time instant t, suppose that the most recently received update packet from user A available at the destination (user B) was generated at time $G_A(t)$. Age of information (AoI) of user A, $\Delta_A(t)$, is a function of time t defined as

$$\Delta_A(t) = t - G_A(t) \quad (2)$$

which is measured at user B. That is, $\Delta_A(t)$ is the time difference between the current time t and the generation time $G_A(t)$ of the freshest received update. Hence, a small $\Delta_A(t)$ implies the existence of a fresh status sample at the destination.

Accordingly, the instantaneous AoI of user B, $\Delta_B(t)$, measured at user A, is $\Delta_B(t) = t - G_B(t)$. The instantaneous AoI $\Delta_j(t)$, $j \in \{A, B\}$, is a continuous-time continuous-value stochastic process [1]. A smaller instantaneous AoI means higher information freshness.

With the instantaneous AoI $\Delta_j(t)$, we can compute other AoI metrics. This paper uses a widely studied AoI metric, *average AoI* [1], to evaluate the information freshness of TWRNs. Average AoI is defined as the time average of the instantaneous AoI. Specifically, the average AoI of user j is given by

$$\bar{\Delta}_j = \lim_{T \to \infty} \frac{1}{T} \int_0^T \Delta_j(t) dt. \quad (3)$$

1. In a practical implementation, the relay can act as a controller and send a trigger frame to trigger the simultaneous transmission of the two users (see the MAC protocol in Section 6 for details).

A lower average AoI indicates that the update packets of user j are generally fresher over a long period.

3.3 Random Coding Bound (RCB)

As we will see in Sections 4 and 5, the average AoI of TWRNs is related to the PERs of both uplink and downlink transmissions. In practical information update systems with AoI requirements, the update packets are typically short. Information theory shows that with a finite block length, the PER cannot go to zero [9]. To theoretically charaterize the PERs of short packets, this paper uses the random coding bound (RCB) to estimate the PERs in TWRNs. Suppose that in the uplink, the relay can successfully decode the superimposed signals into a PNC packet with probability α. In the downlink, let β_j denote the probability that the other user (say, user j') recovers the update packet from user j. That is, the downlink PNC packet is decoded by user j' with a probability β_j such that user j's update packet is recovered. In general, for a coded packet, α and β_j increase with the coded block length.

We first consider the downlink phase. The downlink transmission can be regarded as two point-to-point channels. For point-to-point channels, Gallager's RCB on the average block error probability $1-\beta_j$ of random (L, K) codes has the form [9]

$$1-\beta_j \leq 2^{-LE_G(R)}, \tag{4}$$

where K is the number of source bits of a packet, L is the block length of coded bits, $R = K/L$ is the code rate, and $E_G(R)$ is the random coding error exponent [9]. Under perfect channel state information at the receiver, $E_G(R)$ is

$$E_G(R) = \max_{0\leq\rho\leq 1}[E_0(\rho) - \rho R], \tag{5}$$

where ρ is the auxiliary variable optimized to obtain the maximum value of the right-hand side of (5), and

$$E_0(\rho) := -\log_2 \mathbb{E}\left[\left(\frac{\mathbb{E}[p_{Y|C}(Y|C')^{\frac{1}{1+\rho}}|Y]}{p_{Y|C}(Y|C)^{\frac{1}{1+\rho}}}\right)^{\rho}\right], \tag{6}$$

where C and C' are the two independent and uniformly distributed binary random variables, i.e., the coded bits C and C' have values of 0 or 1. Y is the received signal in an additive white Gaussian noise (AWGN) channel. $\mathbb{E}[\cdot]$ denotes the expectation operator.

For the TWRN uplink transmission, [37] shows that the RCB (4) also holds for the ensemble of linear random codes and thus applies to PNC systems. In particular, (4) applies to PNC decoding that employs an XOR-CD decoder at the receiver. The PER performance of the XOR-CD decoder can be fully characterized by analyzing the transmission with a linear block code over a "virtual" memory-less point-to-point channel [37]. The difference of XOR-CD is that C and C' in (6) are now the two independent and uniformly distributed binary random variables associated with the XOR-coded bits (rather than the individual coded bits in point-to-point channels).

In the case of non-zero PERs, ARQ is commonly used to ensure reliable transmission in wireless systems. However, when the system performance metric is information freshness, this paper raises a question: *to achieve low average AoI, should the corrupted packets be retransmitted, especially in TWRN networks with two hops?* Since new packets always have the latest information, Section 4 studies the average AoI of a PNC-enabled TWRN without ARQ. Specifically, when old packets are corrupted in the uplink or downlink phase, new packets are generated and transmitted (i.e., old packets are dropped). We refer to this scheme as the once-lost-then-drop (OLTD) protocol. Later in Section 5, we evaluate the average AoI with ARQ. Notice that ARQ can be used in either the uplink or downlink phase, or even both. The average AoI under different strategies requires thorough investigation.

4 Average AoI in PNC-enabled TWRNs without ARQ

This section analyzes the average AoI in PNC-enabled TWRNs without ARQ, where the system adopts the once-lost-then-drop (OLTD) protocol. In OLTD, once old packets get corrupted either in the uplink phase (i.e., the relay fails to decode a PNC packet from the superimposed signals from the two end users) or in the downlink phase (i.e., an end user fails to decode the broadcast PNC packet), the two end users immediately generate and transmit new packets to the relay. In other words, old packets are always dropped, and no packets are retransmitted in OLTD.

We remark that in this section (and also in Section 5 below), to simplify the AoI analysis and highlight the key findings among different protocols (i.e., without or with ARQ), we assume a *time-slotted* system where each time slot is an uplink or a downlink transmission with the same duration. The uplink/downlink packet duration occupies one time slot. In other words, we assume delay-free feedback for the theoretical average AoI analysis. Practical MAC protocols involving feedback delays, as well as the AoI performance in real wireless environments on SDR, are discussed in Section 6.

4.1 Once-Lost-Then-Drop (OLTD) Protocol Description

We first describe the detailed operation of the OLTD protocol. Notice that in OLTD, uplink and downlink transmissions may not occur in alternative time slots due to packet corruption.

- **In the uplink phase**, the two users send their update packets to the relay simultaneously, and the relay tries to decode a PNC packet from the superimposed signals of the two uplink packets. If an error occurs, the relay gives feedback to the two end users and informs them to generate and transmit new packets in the next time slot. Otherwise, the decoded PNC packet is broadcast to the end users in the next time slot (i.e., entering the downlink phase).
- **In the downlink phase**, the two users attempt to decode the broadcast PNC packet. Whether the downlink PNC packet is decoded or not, the relay notifies both users to generate and send new update packets in the next time slot. In other words, the PNC packet is broadcast only once, regardless of the end user's decoding result.

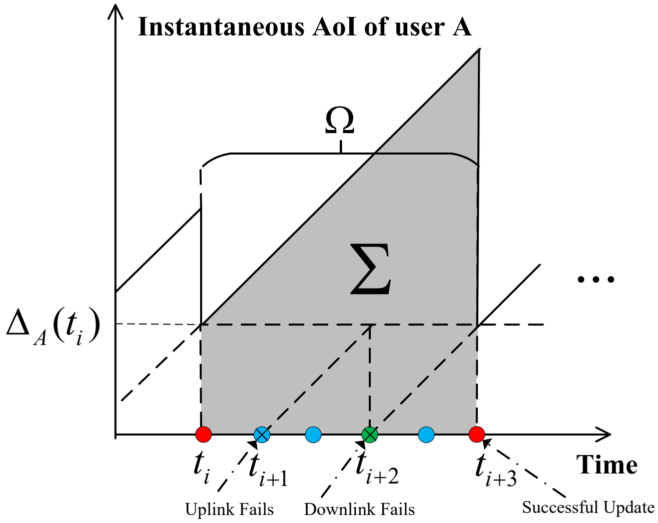


Fig. 3. An example of user A's instantaneous AoI, Δ_A, under the once-lost-then-drop (OLTD) protocol. In round i, users A and B send packets C_i^A and C_i^B, respectively, to the relay simultaneously. Since there is no ARQ, round i ends when the relay fails to decode the PNC packet $C_i^A \oplus C_i^B$ in the uplink. New packets C_{i+1}^A and C_{i+1}^B are sent in round $i+1$, but user A cannot receive the downlink PNC packet. An update is successful in round $i+2$, and Δ_A drops to Ω, which is the total duration from round i to round $i+2$.

Fig. 3 plots an example of user A's instantaneous AoI, Δ_A, measured at user B. In Fig. 3, the evolution of Δ_A between two consecutive successful updates is depicted. After a successful update, we use Ω to denote the time needed for the next update to be successfully received at user B. In other words, in OLTD, Ω is the time required for user B to receive a packet from user A, starting from the time instance of the last successful update.

We define a round as the duration of the time interval between two new update packets sent by the end user. In round i, denote by C_i^A and C_i^B the two update packets sent by user A and user B, respectively. As shown in Fig. 3, packets C_i^A and C_i^B are sent at time t_i. Suppose that the relay fails to decode the PNC packet $C_i^A \oplus C_i^B$ in round i (i.e., the uplink fails). Since there is no packet retransmission in OLTD, round i ends and new packets C_{i+1}^A and C_{i+1}^B are sent in round $i+1$. Now the relay successfully decodes the PNC packet $C_{i+1}^A \oplus C_{i+1}^B$, which is then broadcast to the end users. Here we assume that user B cannot receive $C_{i+1}^A \oplus C_{i+1}^B$ (i.e., the downlink fails, so user A's packet cannot be recovered). Again, there is no packet retransmission in the downlink, so round $i+2$ starts and new packets C_{i+2}^A and C_{i+2}^B are sent at t_{i+2}. Assuming that both uplink and downlink transmissions are successful in round $i+2$, the instantaneous AoI of user A, Δ_A, drops to two time slots at the end of round $i+2$. In addition, Ω is the total duration from round i to round $i+2$.

4.2 Average AoI in OLTD

We now analyze the average AoI of OLTD using the graphical decomposition method [1]. The average AoI can be computed from the area below the instantaneous AoI curve, which is decomposed into several trapezoids.[2] For simplicity, we normalize the duration of a time slot to one in the following analysis. The entire time interval $[0, T]$ contains a series of consecutive successful information updates. To compute the average AoI $\bar{\Delta}_j^{OLTD}$ of user j in OLTD, $j \in \{A, B\}$, let us consider the area Σ between the two consecutive successful updates at t_i and t_{i+3} in Fig. 3. The area of Σ is computed by

$$\Sigma = \Delta_j(t_i)\Omega + \frac{1}{2}\Omega^2 = 2\Omega + \frac{1}{2}\Omega^2, \tag{7}$$

where $\Delta_j(t_i) = 2$ since a successful update always drops the instantaneous AoI to two time slots in OLTD. The average AoI of user j, $\bar{\Delta}_j^{OLTD}$, is

$$\begin{aligned}\bar{\Delta}_j^{OLTD} &= \lim_{T\to\infty} \frac{1}{T}\int_0^T \Delta_j(t)dt = \lim_{W\to\infty} \frac{\sum_{w=1}^{W} \Sigma_{(w)}}{\sum_{w=1}^{W} \Omega_{(w)}} \\ &= \frac{\mathbb{E}\left[2\Omega + \frac{1}{2}\Omega^2\right]}{\mathbb{E}\left[\Omega\right]} = 2 + \frac{\mathbb{E}\left[\Omega^2\right]}{2\mathbb{E}\left[\Omega\right]}\end{aligned} \tag{8}$$

where $\Sigma_{(w)}$ and $\Omega_{(w)}$ denote the w-th Σ and Ω, respectively.

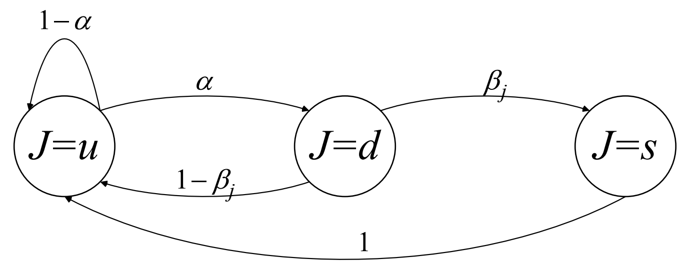


Fig. 4. The Markov model for the OLTD protocol. State $J = \{u, d, s\}$ represents the stage of the packet transmission in a TWRN: u stands for $u$plink; d stands for $d$ownlink; s stands for $s$uccessful reception at the destination.

To compute $\bar{\Delta}_j^{OLTD}$, we use a Markov model depicted in Fig. 4 to find $\mathbb{E}[\Omega]$ and $\mathbb{E}[\Omega^2]$. In the Markov model, state $J = \{u, d, s\}$ represents the stage of packet transmission in a TWRN. $J = u$ represents an uplink transmission from the end users to the relay; $J = d$ represents a downlink transmission from the relay to the end users; $J = s$ represents a successful update of user j (e.g., if $j =$ A, it means that user B successfully receives an update packet from user A). More specifically, at state $J = u$, the relay decodes the PNC packet successfully with probability α, and the system transits to $J = d$. The relay fails to decode the PNC packet (i.e., drops the old packets) with probability $1-\alpha$, and the state remains at $J = u$. At state $J = d$, the broadcast PNC packet is decoded, and the system transits to $J = s$ with probability β_j (i.e., user j's update packet is recovered by the other

2. Note that to compute the average AoI, both the continuous-time [11], [28] and discrete-time [25], [26] settings can be found in the literature. In particular, by using the graphical decomposition method, the area below the instantaneous AoI curve is decomposed into several trapezoids in the continuous-time setting, while it is decomposed into several rectangles in the discrete-time setting (whose areas are always smaller than the trapezoids). The different settings do not affect the AoI comparison among different protocols as long as they follow the same continuous-time or discrete-time setting. Our work adopts the continuous-time setting as in [11], [28] to compute the average AoI.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

user). Otherwise, with probability $1-\beta_j$, the system transits to uplink transmission (i.e., the next state is $J=u$, meaning that the old packets are discarded). Finally, at state $J=s$, the system will transit to $J=u$ with probability one for a new round of transmission.

Let Π denote the state transition matrix, which can be written as

$$\Pi = \begin{pmatrix} \pi_{uu} & \pi_{ud} & \pi_{us} \\ \pi_{du} & \pi_{dd} & \pi_{ds} \\ \pi_{su} & \pi_{sd} & \pi_{ss} \end{pmatrix} = \begin{pmatrix} 1-\alpha & \alpha & 0 \\ 1-\beta_j & 0 & \beta_j \\ 1 & 0 & 0 \end{pmatrix}, \quad (9)$$

where π_{mn} is the probability of transiting from state $J=m$ to state $J=n$, for $m,n \in \{u,d,s\}$. Since Ω is the time needed to successfully receive the next update, Ω equals the time to go through a series of states from state $J=u$ to state $J=s$ in the Markov model, i.e., the state transition goes through states $J_0=u, J_1, J_2, ..., J_\Omega = s$. Denote by τ_{mn} the expected time required to transverse from state $J=m$ to state $J=n$. Then, τ_{mn} can be expressed as

$$\tau_{mn} = \mathbb{E}[t_n | J_0 = m], \quad (10)$$

where t_n is a random variable that represents the time to reach state $J=n$ for the first time. By definition, $\mathbb{E}[\Omega]$ equals τ_{us}, and we compute τ_{us} by

$$\begin{aligned} \tau_{us} &= \mathbb{E}[t_s | J_0 = u] \\ &= 1 + \mathbb{E}[t_s | J_1 = u] \Pr(J_1 = u | J_0 = u) \\ &\qquad + \mathbb{E}[t_s | J_1 = d] \Pr(J_1 = d | J_0 = u) \\ &= 1 + \tau_{us}\pi_{uu} + \tau_{ds}\pi_{ud} \\ &= 1 + (1-\alpha)\tau_{us} + \alpha\tau_{ds}. \end{aligned} \quad (11)$$

In (11), τ_{ds} is computed by

$$\begin{aligned} \tau_{ds} &= \mathbb{E}[t_s | J_0 = d] \\ &= 1 + \mathbb{E}[t_s | J_1 = u] \Pr(J_1 = u | J_0 = d) \\ &\qquad + \mathbb{E}[t_s | J_1 = s] \Pr(J_1 = s | J_0 = d) \\ &= 1 + \tau_{us}\pi_{du} + \tau_{ss}\pi_{ds} \\ &= 1 + (1-\beta_j)\tau_{us}. \end{aligned} \quad (12)$$

Note that in (12), $\tau_{ss}=0$. Substituting (12) into (11), τ_{us} and $\mathbb{E}[\Omega]$ are computed and simplified as

$$\mathbb{E}[\Omega] = \tau_{us} = \frac{1+\alpha}{\alpha\beta_j}. \quad (13)$$

Similarly, to compute $\mathbb{E}[\Omega^2]$, we define λ_{mn} as the expectation of the second moment of the time required to transverse from state $J=m$ to state $J=n$ for the first time. Then, λ_{mn} can be expressed as

$$\lambda_{mn} = \mathbb{E}[(t_n)^2 | J_0 = m]. \quad (14)$$

By definition, $\mathbb{E}[\Omega^2]$ equals λ_{us}, which is computed by

$$\begin{aligned} \lambda_{us} &= \mathbb{E}[(t_s)^2 | J_0 = u] \\ &= \mathbb{E}[(1+t_s)^2 | J_0 = u] \Pr(J_1 = u | J_0 = u) \\ &\qquad + \mathbb{E}[(1+t_s)^2 | J_1 = d] \Pr(J_1 = d | J_0 = u) \\ &= 1 + 2\left(\tau_{us}\pi_{uu} + \tau_{ds}\pi_{ud}\right) + \left(\lambda_{us}\pi_{uu} + \lambda_{ds}\pi_{ud}\right) \end{aligned} \quad (15)$$

where λ_{ds} is

$$\begin{aligned} \lambda_{ds} &= \mathbb{E}[(t_s)^2 | J_0 = d] \\ &= \mathbb{E}[(1+t_s)^2 | J_1 = u] \Pr(J_1 = u | J_0 = d) \\ &\qquad + \mathbb{E}[(1+t_s)^2 | J_1 = s] \Pr(J_1 = s | J_0 = d) \\ &= 1 + 2\tau_{us}\pi_{du} + \lambda_{us}\pi_{du} \end{aligned} \quad (16)$$

Substituting (16) into (15), λ_{us} and $\mathbb{E}[\Omega^2]$ are now computed and simplified to

$$\mathbb{E}[\Omega^2] = \lambda_{us} = \frac{2 + 4\alpha + 2\alpha^2 - 3\alpha\beta_j - \alpha^2\beta_j}{\alpha^2\beta_j^2}. \quad (17)$$

Finally, with $\mathbb{E}[\Omega]$ from (13) and $\mathbb{E}[\Omega^2]$ from (17), we obtain the average AoI $\bar{\Delta}_j^{OLTD}$

$$\bar{\Delta}_j^{OLTD} = 2 + \frac{\mathbb{E}\left[\Omega^2\right]}{2\mathbb{E}\left[\Omega\right]} = 2 + \frac{2 + 4\alpha + 2\alpha^2 - 3\alpha\beta_j - \alpha^2\beta_j}{2(1+\alpha)\alpha\beta_j}. \quad (18)$$

In (18), the first term "2" refers to the instantaneous AoI once an update packet is received (i.e., two time slots are the lowest instantaneous AoI in a two-hop TWRN). When both α and β_j equal to one, $\bar{\Delta}_j^{OLTD}$ has the minimum average AoI of 3 time slots. However, when α and β_j are not equal to 1, we evaluate $\bar{\Delta}_j^{OLTD}$ by RCB theoretically and real experiments with SDR, as will be presented in Section 5.3 and Section 6, respectively.

5 Average AoI in PNC-enabled TWRNs with ARQ

This section presents the average AoI in TWRNs with ARQ. Specifically, we first consider reliable packet transmission where ARQ is used in both the uplink and downlink phases. After that, we investigate that ARQ is used only in the downlink phase and the old packets are dropped if they are lost in the uplink phase. A theoretical comparison among different schemes using RCB is presented in Section 5.3.

5.1 Reliable Packet Transmission (RPT)

We first study the average AoI of a TWRN employing reliable packet transmission (RPT). By "reliable", we mean that a link-by-link ARQ is used to issue retransmissions in case of uplink or downlink packet corruption (i.e., no packet loss in either the uplink or downlink phase). The detailed operations of the RPT protocol are as follows:

- **In the uplink phase**, the two users send their update packets to the relay simultaneously, and the relay tries to decode a PNC packet. If an error occurs, the relay gives the two end users feedback and informs them to retransmit the same packets. This procedure is repeated until the PNC packet is successfully decoded at the relay. When the PNC packet is decoded, it is broadcast to the end users in the downlink phase.
- **In the downlink phase**, the two users try to decode the downlink PNC packets. If a user successfully decodes the PNC packet, it then sends an acknowledgment (ACK) to the relay. If the relay does not receive ACKs from both users, the PNC packet is retransmitted. We remark that an end user may

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

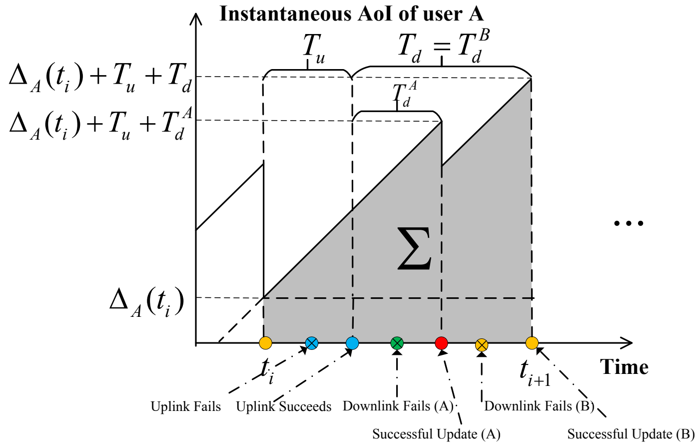


Fig. 5. An example of user A's instantaneous AoI, Δ_A, with reliable packet transmission (RPT) in round i. In the uplink transmission, the two users transmit the same packets C_i^A and C_i^B twice until the relay successfully decodes the PNC packet $C_i^A \oplus C_i^B$ at $t_i + T_u$. In the downlink transmission, user B decodes $C_i^A \oplus C_i^B$ and recovers user A's packet first (at time $t_i + T_u + T_d^A$), so Δ_A is reset to $T_u + T_d^A$. Δ_A continues to increase until the end of round i, when user B's packet is recovered at $t_{i+1} = T_u + T_d^B$.

successfully receive the PNC packet before the other end user. The relay has to continue sending the PNC packet until successful reception at both users. When both users receive the PNC packet, they are allowed to sample and send new update packets to the relay in the next time slot.

As in OLTD, we define a round as the duration between two new update packets sent by the user. In RPT, a round is the time between new packets first sent in the uplink and the PNC packet finally received by both users in the downlink. To see this, Fig. 5 plots an example of the evolution of user A's instantaneous AoI Δ_A with RPT. There is only one round between the two consecutive successful updates.

As shown in Fig. 5, suppose that round i starts at time t_i and ends at time t_{i+1} in this example. In the uplink transmission, the relay cannot decode the PNC packet when C_i^A and C_i^B are sent for the first time. Later, retransmission of C_i^A and C_i^B is issued, and the relay successfully decodes the PNC packet $C_i^A \oplus C_i^B$ at time $t_i + T_u$, where T_u is the total duration of the uplink phase. In the downlink, the relay broadcasts $C_i^A \oplus C_i^B$ four times until both users receive the PNC packet. User B decodes $C_i^A \oplus C_i^B$ when it is sent for the second time. Using the received $C_i^A \oplus C_i^B$, C_i^A is recovered at user B and the instantaneous AoI Δ_A drops to $T_u + T_d^A$, where T_d^A is the time required for C_i^A to be recovered in the downlink phase. We also use T_d^B to denote the time required for C_i^B to be recovered at user A. In this example, user A recovers C_i^B when $C_i^A \oplus C_i^B$ is sent for the fourth time. Since C_i^B is recovered later than C_i^A, after Δ_A is reset to $T_u + T_d^A$, Δ_A continues to increase linearly. We use T_d to denote the total duration of the downlink phase. It is easy to see that $T_d = \max\{T_d^A, T_d^B\} = T_d^B$ in Fig. 5, since $T_d^A < T_d^B$.

Average AoI in RPT: We now analyze the average AoI in RPT $\bar{\Delta}_j^{RPT}$, $j \in \{A, B\}$. Similar to OLTD, we consider the area Σ shown in Fig. 5. The area of Σ is computed by (19), where $\Delta_j(t_i) = {T_u}' + {T_d}'$ equals to the duration of the last round. Then the average AoI $\bar{\Delta}_j^{RPT}$ is given by (20). In the following, we compute each component in (20).

Computation of $\mathbb{E}[T_u]$ and $\mathbb{E}[(T_u)^2]$: T_u is the duration of the uplink phase, and the relay successfully decodes the PNC packet with probability α. Thus, T_u is a geometric random variable with parameter α. We have

$$\mathbb{E}[T_u] = \frac{1}{\alpha}, \tag{21}$$

$$\mathbb{E}[(T_u)^2] = \frac{2-\alpha}{\alpha^2}. \tag{22}$$

Computation of $\mathbb{E}[T_d^j]$: Since $\mathbb{E}[T_d^j]$ is the time for packet $C_i^j, j \in \{A, B\}$ to be recovered in the downlink phase (i.e., the PNC packet $C_i^A \oplus C_i^B$ is decoded), T_d^j is also a geometric random variable with parameter β_j. Thus, we have

$$\mathbb{E}[T_d^j] = \frac{1}{\beta_j}. \tag{23}$$

Computation of $\mathbb{E}[T_d]$ and $\mathbb{E}[(T_d)^2]$: T_d is the total duration of the downlink phase, and by definition, $T_d = \max\{T_d^A, T_d^B\}$. The cumulative distribution function (CDF) of $T_d^j, j \in \{A, B\}$, is

$$F_d^j(t) = \Pr(T_d^j \le t) = 1 - (1-\beta_j)^t. \tag{24}$$

Therefore, the CDF of T_d is

$$F_d(t) = \Pr(T_d \le t) = \Big(1-(1-\beta_A)^t\Big)\cdot\Big(1-(1-\beta_B)^t\Big). \tag{25}$$

With the CDF of T_d, we compute $\mathbb{E}[T_d]$ and $\mathbb{E}[(T_d)^2]$, respectively, where

$$\begin{aligned}\mathbb{E}[T_d] &= \sum\nolimits_{t=1}^{\infty}\left(1-\Pr(T_d \le t-1)\right)\\ &= \sum\nolimits_{t=1}^{\infty}\left(1-F_d(t-1)\right),\end{aligned} \tag{26}$$

$$\begin{aligned}\mathbb{E}[(T_d)^2] &= \sum\nolimits_{t=1}^{\infty}(2t-1)\left(1-\Pr(T_d \le t-1)\right)\\ &= \sum\nolimits_{t=1}^{\infty}(2t-1)\left(1-F_d(t-1)\right).\end{aligned} \tag{27}$$

Finally, we substitute all the components into (20) to obtain the average AoI $\bar{\Delta}_j^{RPT}$ as in (28). From (28), we see that the average AoI $\bar{\Delta}_j^{RPT}$ is much more complicated than $\bar{\Delta}_j^{OLTD}$ in (18). However, when both α and β_j equal to one, it is easy to verify that $\bar{\Delta}_j^{RPT} = \bar{\Delta}_j^{OLTD} = 3$.

5.2 Uplink-Lost-Then-Drop (ULTD) Protocol

Although RPT ensures packet reliability, as will be seen, it leads to high average AoI due to the retransmission of old packets in the uplink phase. Therefore, we put forth an uplink-lost-then-drop (ULTD) protocol that exploits both ARQ and packet drop. To be specific, when the relay fails to decode the PNC packet in the uplink phase, old packets are dropped. The two end users generate and transmit two new packets to the relay in the next time slot. In the downlink phase, ARQ is used to ensure reliable transmission until both users receive the downlink PNC packet. In other words, the uplink transmission of ULTD is the same as that of OLTD, while the downlink transmission of ULTD is the same as that of RPT.

We compute the average AoI of ULTD, $\bar{\Delta}_j^{ULTD}, j \in \{A, B\}$, in the same way as RPT. The area of Σ is computed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

$$\begin{aligned}\Sigma &= \Delta_j(t_i)(T_u + T_d^j) + \frac{1}{2}(T_u + T_d^j)^2 + (T_u + T_d^j)(T_d - T_d^j) + \frac{1}{2}(T_d - T_d^j)^2 \\ &= (T_u{}' + T_d{}')(T_u + T_d^j) + (T_u + T_d^j)(T_d - T_d^j) + \frac{1}{2}(T_u + T_d^j)^2 + \frac{1}{2}(T_d - T_d^j)^2.\end{aligned} \tag{19}$$

$$\begin{aligned}\bar{\Delta}_j^{RPT} &= \lim_{W\to\infty} \frac{\sum_{w=1}^{W} \Sigma_{(w)}}{\sum_{w=1}^{W} T_{u(w)} + T_{d(w)}} = \frac{\mathbb{E}\left[(T_u{}' + T_d{}')(T_u + T_d^j) + (T_u + T_d^j)(T_d - T_d^j) + \frac{1}{2}(T_u + T_d^j)^2 + \frac{1}{2}(T_d - T_d^j)^2\right]}{\mathbb{E}\left[T_u + T_d\right]} \\ &= \frac{(\mathbb{E}[T_u])^2 + \mathbb{E}[T_u]\mathbb{E}[T_d^j] + 2\mathbb{E}[T_u]\mathbb{E}[T_d] + \mathbb{E}[T_d^j]\mathbb{E}[T_d] + \frac{1}{2}\mathbb{E}[(T_u)^2] + \frac{1}{2}\mathbb{E}[(T_d)^2]}{\mathbb{E}[T_u] + \mathbb{E}[T_d]}.\end{aligned} \tag{20}$$

$$\bar{\Delta}_j^{RPT} = \frac{\left(\ 2\alpha + 4\beta_j - \alpha\beta_j + \left(4\alpha\beta_j + 2\alpha^2\right)\sum_{t=1}^{\infty}\left(1 - F_d(t-1)\right) + \alpha^2\beta_j \sum_{t=1}^{\infty}\left(2t-1\right)\left(1 - F_d(t-1)\right)\ \right)}{2\alpha\beta_j\left(1+\alpha\right)\sum_{t=1}^{\infty}\left(1 - F_d(t-1)\right)}. \tag{28}$$

$$\begin{aligned}\bar{\Delta}_j^{ULTD} &= \frac{\mathbb{E}[T_u] + \mathbb{E}[T_d^j] + 2\mathbb{E}[T_u]\mathbb{E}[T_d] + \mathbb{E}[T_d^j]\mathbb{E}[T_d] + \frac{1}{2}\mathbb{E}[(T_u)^2] + \frac{1}{2}\mathbb{E}[(T_d)^2]}{\mathbb{E}[T_u] + \mathbb{E}[T_d]} \\ &= \frac{\left(\ \alpha\beta_j + 2\alpha^2 + 2\beta_j + \left(4\alpha\beta_j + 2\alpha^2\right)\sum_{t=1}^{\infty}\left(1 - F_d(t-1)\right) + \alpha^2\beta_j \sum_{t=1}^{\infty}\left(2t-1\right)\left(1 - F_d(t-1)\right)\ \right)}{2\alpha\beta_j\left(1+\alpha\right)\sum_{t=1}^{\infty}\left(1 - F_d(t-1)\right)}.\end{aligned} \tag{29}$$

---

the same as (19), except that the instantaneous AoI upon a successful update becomes $\Delta_j(t_i) = 1 + T_d{}'$. Here, "1" refers to the time required for the uplink transmission of each successful update, which is always one time slot since old uplink packets are dropped in ULTD. Thus, the average AoI $\bar{\Delta}_j^{ULTD}$ is computed by (29).

It is easy to verify that, given the same α and β_j, $\bar{\Delta}_j^{ULTD}$ in (29) is smaller than $\bar{\Delta}_j^{RPT}$ in (28), because $0 < \alpha, \beta_j < 1$. This means that if packet reliability is not a concern (e.g., only information freshness is concerned in many monitoring systems), new packets should be generated and sent immediately when the relay fails to decode the PNC packet in the uplink so as to achieve high information freshness.

We notice that one can also have a downlink-lost-then-drop (DLTD) protocol. In the uplink, old packets are retransmitted until the PNC packet is successfully decoded at the relay (i.e., ARQ is used in the uplink). In the downlink, the PNC packet is broadcast only once. Whether the downlink PNC packet is decoded or not, the two end users will turn to transmit new packets in the next time slot. In other words, for DLTD, the uplink transmission is the same as that in RPT, while the downlink transmission is the same as that in OLTD. However, it is easy to figure out that the average AoI in DLTD is high. Even if the relay can finally decode a PNC packet in the uplink after several retransmissions (e.g., it takes $E[T_u]$ time slots on average), old packets will be dropped once the subsequent downlink transmission fails. In addition, ULTD indicates that new packets should be sent when the relay fails to decode the PNC packet in the uplink. Therefore, we omit DLTD in this paper. In the following subsection, we evaluate the theoretical average AoI of OLTD, RPT, and ULTD using RCB.

5.3 Average AoI Comparison Using RCB

In this subsection, we use RCB to estimate the PERs and provide a theoretical average AoI comparison among the three protocols. We assume that the two end users have the same channel conditions, i.e., they have the same received SNR at the relay in the uplink and the same decoding probability $\beta_A = \beta_B$ in the downlink. To estimate α and β_j by RCB, we assume that the number of source bits per update packet is $K = 400$ and vary the block length of a coded packet. The case of imbalanced channel conditions will be investigated experimentally by SDR in Section 6.

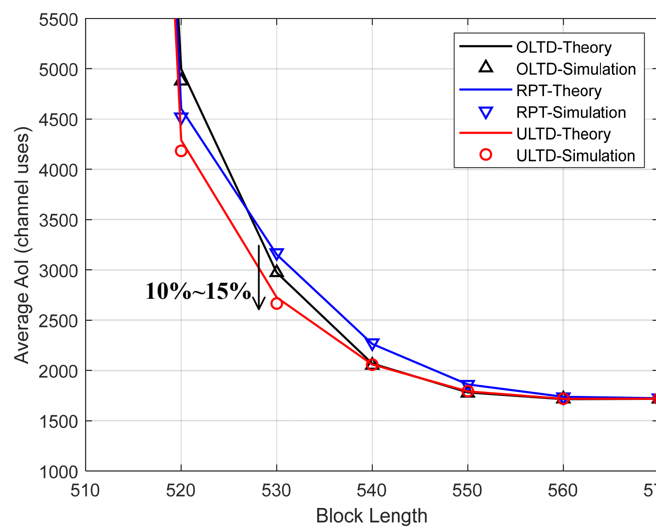


Fig. 6. The average AoI versus the block length of a coded packet when the two users have SNR $= 1\ dB$ in both uplink and downlink transmissions. The successful packet decoding rates α and β_j are estimated by RCB.

Fig. 6 plots the average AoI versus the block length when the SNR is $1\ dB$. Similar observations can be found when other SNRs are considered (so we omit these results here). The unit of the average AoI is the number of channel uses.[3] Fig. 6 plots both the theoretical and simulation results. For the theoretical results, we substitute the successful packet decoding rates α and β_j into the average AoI formulas

3. The average AoI in the unit of channel use is the multiplication of the block length and the average AoI in the unit of time slot.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

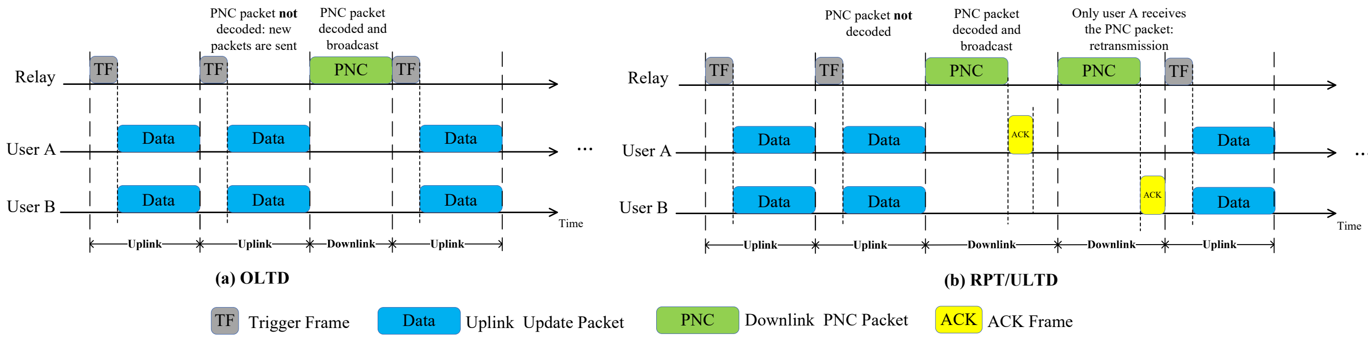


Fig. 7. MAC protocols used in our experiments: (a) OLTD, (b) RPT and ULTD. The duration of an update packet is 0.16ms and the duration of a trigger frame/ACK frame is 0.04ms in our experiments.

of the three protocols (i.e., (18), (28), and (29)) to calculate the average AoI. For the simulations results, we perform simulations over a large number of time slots and collect the packet decoding outcomes in each time slot, from which we compute the instantaneous AoI followed by the average AoI. We see from Fig. 6 that the simulation results are consistent with the theoretical results, thus proving the correctness of the average AoI formulas.

As depicted in Fig. 6, for all the three protocols, the average AoI decreases as the block length increases. When the block length is small, the successful packet decoding rates α and β_j are small, leading to unsuccessful updates most of the time. As the block length increases, the average AoI decreases as more update packets can be successfully received. When the block length is large enough that α and β_j approach one, e.g., when the block length is around 550, the three protocols tend to have the same average AoI (so there is no need to use ARQ). We also see that the average AoI slightly increases as the block length increases beyond 560 (i.e., further enlarging the block length does not increase α and β_j any further, but takes more time). In general, the optimal block length decreases as the SNR increases, since a higher SNR can help reduce coding redundancy to achieve the same error rate performance.[4]

In time-varying wireless environments, optimizing block lengths to accommodate changing SNRs may not be possible or practical. When the block length is not large enough such that α and β_j are less than one, Fig. 6 shows that the average AoI performance of ULTD outperforms those of RPT and OLTD greatly, e.g., when the block length is around 530 and β_j is around 0.7. This indicates that in error-prone TWRNs, neither a pure packet drop protocol (e.g., OLTD, where new packets always have higher priority) nor a pure packet retransmission protocol (e.g., RPT, where old packets always have higher priority) is suitable for achieving good information freshness. Instead, exploiting the ideas from both sides leads to a better average AoI. Such an observation is different from many previous point-to-point information update systems, where dropping old packets leads to better average AoI performance under the generate-at-will model [11].

The numerical results using RCB are theoretical in nature and only serve to highlight certain points, but do not reflect what actually happens in real wireless communication systems. For example, due to channel fading, the wireless channel is not an AWGN channel in practice, and the two users may have different channel conditions with respect to the relay. More importantly, practical considerations such as MAC protocols to support ULTD, especially the feedback delay due to ARQ, need to be taken into account. Hence, we need to validate the advantages of ULTD in a real wireless system, as will be presented in the next section.

4. In addition, the optimal block lengths of the three protocols do not differ much, because the optimal average AoI is achieved when the PER is low enough and ARQ is not often needed. In other words, the optimal average AoI is almost the same for all three protocols. However, in error-prone environments (when the PER is not low enough), ULTD outperforms OLTD and RPT, as shown in Fig. 6.

6 Experimental Evaluations

This section presents the experimental evaluation on AoI of different transmission protocols in TWRNs, namely OLTD, RPT, and ULTD. In particular, we examine the average AoI performance by employing trace-driven simulations using PHY-layer decoding results. We first obtain the PHY-layer decoding outcomes on software-defined radios (SDR), i.e., collecting the PNC decoding results at the relay in the uplink and the broadcast PNC packet decoding results at the end users in the downlink. Then, we generate traces based on the PHY-layer decoding outcomes to drive our AoI simulations under practical MAC protocols of different schemes.

Hence, in Section 6.1, we first present the MAC protocols we used in the trace-driven simulations. Subsequently, Section 6.2 describes the experimental setup on how we collect the PHY-layer decoding outcomes using SDR. Section 6.3 details the average AoI performance of different schemes. Furthermore, we discuss the differences between AoI and conventional system metrics, such as throughput and delay, using the SDR experimental results.

6.1 MAC Protocols of Different Schemes

Recall that in our theoretical analysis, we simplify the system model and consider a time-slotted system, where an uplink or downlink transmission occupies a time slot with equal duration. However, control frames such as ACK frames do consume airtime, which should be considered when comparing the average AoI performance in real experiments.

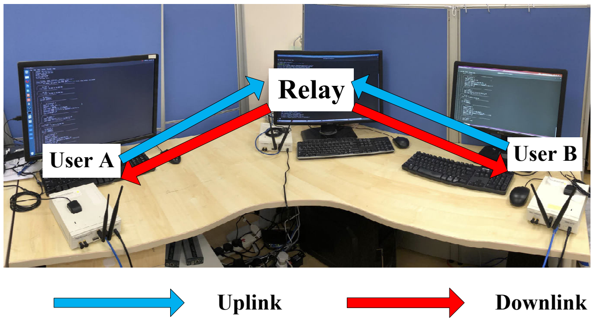


Fig. 8. An indoor information update TWRN system built upon an OFDM-based real-time PNC system [8].

Fig. 7 depicts the MAC protocols used in our trace-driven experiments for (a) OLTD and (b) RPT/ULTD. As shown in Fig. 7 (a), in OLTD, the relay sends a trigger frame (TF) to trigger the simultaneous transmission of the two users in the uplink. If the relay cannot decode the PNC packet, it sends another TF to inform both users to send two new packets (i.e., no ARQ is used in the uplink). When the relay decodes the PNC packet, the downlink phase is entered, and the decoded PNC packet is broadcast only once (i.e., again, no ARQ is used in the downlink). Regardless of the decoding results of the users, a new TF is sent to trigger a new round of uplink transmission.

For RPT and ULTD, in the uplink, the relay also sends a TF to trigger a simultaneous transmission of the two users. A new TF is sent when the relay cannot decode the PNC packet. Note that the only difference between RPT and ULTD is that when the uplink transmission fails, old packets are retransmitted in RPT, while new packets are generated and sent in ULTD. The PNC packet is broadcast after being decoded in the uplink. When a user decodes the downlink PNC packet, it sends an ACK to the relay and waits for the next TF for a new uplink transmission. The two users are designed to send their ACKs at different times to avoid collision (see Fig. 7 (b)). In both RPT and ULTD, the PNC packet is retransmitted until the relay receives ACKs from both users. Upon receiving ACKs from both users, the relay sends a new TF.

In our SDR experiments below, we find that the successful reception rate of TFs and ACKs exceeds 99.9%. Hence, in our trace-driven simulations, we assume that the control frames are reliable. In practice, a timeout mechanism as in the 802.11 stop-and-wait protocol can be adopted when control frames are lost. The transmitter resends a packet when it does not receive any feedback after the timeout period [38]. For example, when the relay sends a TF in the uplink phase and does not receive uplink packets from the users, the relay can send a new TF after a timeout.

6.2 Experimental Setup

For the experiment with SDR, our system uses the USRP hardware (USRP N210 with SBX daughterboards) and the GNU Radio software with the UHD hardware driver. We consider an indoor information update TWRN system, and the system prototype is shown in Fig. 8. Our system is built upon an orthogonal frequency division multiplexing

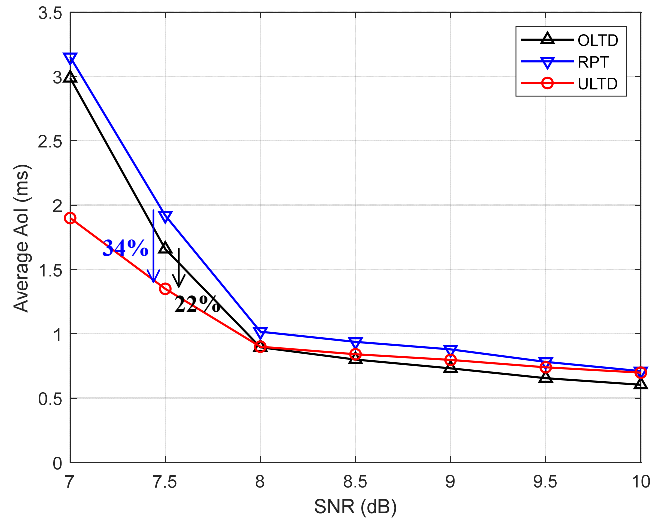


Fig. 9. SDR experimental results of the average AoI under different schemes versus the SNR in the SNR-balanced scenario.

(OFDM)-based real-time PNC system, and we refer the readers to [8], [39] for the implementation details of the algorithms used in this system, such as packet detection, synchronization, etc.

To gather the PHY-layer decoding results, our experiments are carried out at 2.185 GHz center frequency with 10 MHz bandwidth. We control the transmit power of the two end users so that the received SNR is varied from 7.5 dB to 10 dB in the uplink. Note that the two users may have different SNRs. For each SNR pair of the two users, the relay sends a trigger frame to trigger the simultaneous transmission of a series of update packets (e.g., 2000 update packets in our experiments) from the two users in the uplink phase.[5] Also, notice that the received SNR of a user's update packets could be slightly different due to channel fading. The SNR here is the average SNR of the 2000 update packets. We collect the PNC decoding outcomes of the 2000 superimposed packets at the relay. Similarly, in the downlink phase, we adjust the users' position so that the relay has a SNR ranging from 7.5 dB to 10 dB at the users in the downlink. The relay sends 2000 packets to the users at each SNR, and then both users decode the downlink packets to gather the decoding results.

Both end users adopt the BPSK modulation and the standard rate-1/2 $[133, 171]_8$ convolutional code used in the 802.11 standard [38]. Since the received SNR is an average value, we choose the fixed-rate convolutional code in our experiments. Optimizing the block length based on instantaneous SNR to minimize the average AoI may not be possible or practical. In our experiments, the payload of an update packet is 48 bytes (384 bits), since packets containing update information from IoT devices are typically short (e.g., tens of bytes) in practice. The TF and the ACK start with an 8-bit device ID field, followed by a 16-bit control field for

5. This achieves synchronization for PNC in our experiments. Since the end users use OFDM, as long as the arrival time difference between users is within the cyclic prefix of OFDM symbols (i.e., within 1.6 μs in our implementation), the OFDM frequency-domain symbol misalignment between users can be eliminated, thereby enabling PNC decoding [8].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

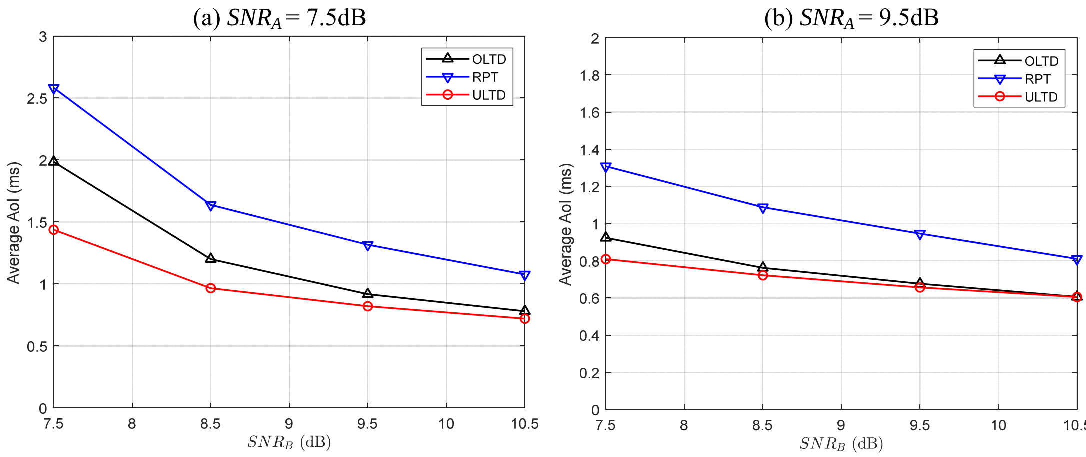


Fig. 10. SDR experimental results of the average AoI under different schemes versus the SNR in the SNR-imbalanced scenario.

packet triggering or packet acknowledgment. Each update packet or control frame is preceded by a preamble before the payload. The preamble assists the receivers with packet synchronization and channel estimation [8].

As a result, the duration of an update packet (for both uplink and downlink packets) is

$$T^{pk} = \left(320 + \frac{384 \times 2}{48} \times 80\right) / 10^7 \ s = 0.16 \ ms, \qquad (30)$$

where ms stands for millisecond. In (30), 320 is the number of samples in the preamble. 80 is the total number of samples in one OFDM symbol, consisting of a 64-FFT OFDM symbol and a 16-sample cyclic prefix. Although an OFDM symbol has 64 subcarriers, only 48 subcarriers are used for actual data transmission. $384 \times 2/48$ gives the number of OFDM symbols in an update packet with 384 source bits and a channel coding rate $1/2$. 10^7 means the bandwidth is $10 \ MHz$, i.e., the duration of one sample is $10^{-7}s$. Similarly, the duration of a TF or an ACK, T^c, is given by (where c denotes the control frames)

$$T^{c} = \left(320 + \frac{24 \times 2}{48} \times 80\right) / 10^7 \ s = 0.04 \ ms. \qquad (31)$$

6.3 Experiment Results

We first consider the same scenario as in Section 5.3, where the two end users have balanced channel conditions, as shown in Fig. 9. Notice that we compute the average AoI with the unit of ms here. We see from Fig. 9 that when the SNR is less than $8 \ dB$, the experimental results corroborate with the theoretical results. That is, ULTD outperforms both RPT and OLTD significantly, e.g., compared with OLTD and RPT, ULTD reduces the average AoI by 22% and 34%, respectively, at the SNR of $7.5 \ dB$. Our experiments indicate that in the low SNR regime, the ULTD protocol is preferred for achieving low average AoI in TWRNs operated with PNC. More specifically, when old packets get corrupted, dropping the old packets should be adopted in the first hop (i.e., in the uplink). Since the newly sent packets contain fresher information, the instantaneous AoI of ULTD is always smaller than that of RRT once the packet is successfully received at the destination. This is the reason why ULTD is superior to RPT. Although OLTD always sends new packets, once the downlink phase fails, OLTD generates new packets blindly. The relay may take time to decode a PNC packet, but if there is an error in the downlink, the PNC packet is discarded. When the SNR is low, the time interval between the reception of two consecutive update packets could be very large in OLTD (i.e., the instantaneous AoI is high). As a result, the average AoI of OLTD is much larger than that of ULTD in the low SNR regime, indicating that packet retransmission should be used in the second hop.

As the SNR increases, the average AoI performance of OLTD is gradually better than that of ULTD. This is because when the SNR is high, the packet decoding rates (for both uplink and downlink) also increase, and ARQ is not often needed. ULTD now suffers from the larger overhead of control frames and therefore has a higher average AoI. The lower average AoI of OLTD indicates that OLTD is a practical and simpler choice in the high SNR regime. This is the main difference between the theoretical results in Section 5.3 and the experimental results here, where realistic MAC implementations should be considered in practice.

However, in time-varying wireless environments, the two users are likely to have different channel conditions with respect to the relay. We now compare the average AoI performance when the two users have different SNRs. We fix the SNR between user A and the relay (i.e., from user A to the relay in the uplink and from the relay to user A in the downlink, denoted by SNR_A) to $7.5 \ dB$ and $9.5 \ dB$ and vary the SNR between user B and the relay (i.e., from user B to the relay in the uplink and from the relay to user B in the downlink, denoted by SNR_B) from $7.5 \ dB$ to $10.5 \ dB$. Fig. 10 plots the average AoI of the TWRN versus SNR_B, when SNR_A is (a) $7.5 \ dB$ and (b) $9.5 \ dB$, respectively. We see in both plots that ULTD gives the lowest average AoI among the three protocols. In addition, the performance improvement of ULTD over its counterparts is larger when SNR_A is $7.5 \ dB$. This is because in the SNR-imbalanced scenario, for one user to deliver an update packet to the other user, the update packet has to go through a link with a lower SNR, either in the uplink phase or in the downlink phase. In particular, in the uplink transmission (the first hop), a user must have a low SNR (e.g., $7.5 \ dB$). It is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

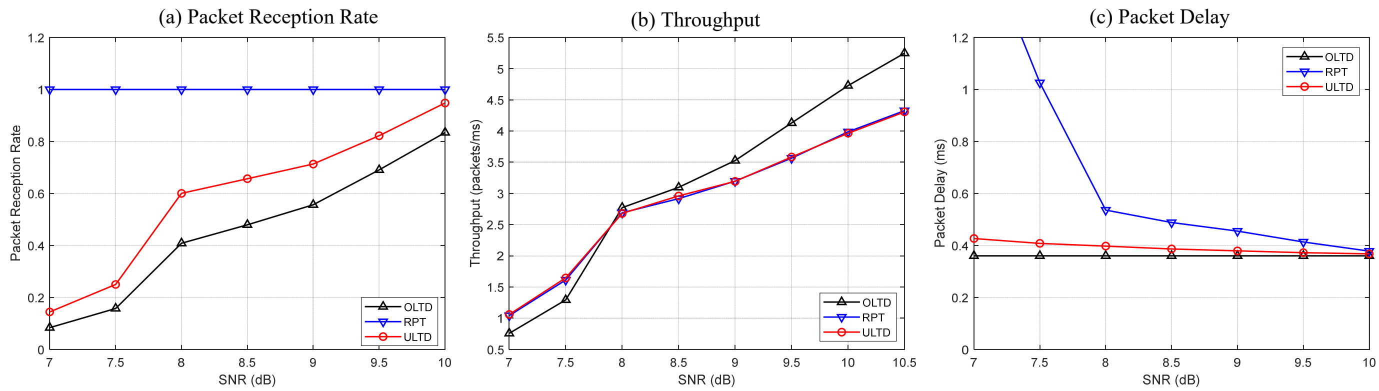


Fig. 11. SDR experimental results for (a) packet reception rate, (b) throughput, and (c) packet delay versus the SNR under the three protocols in the SNR-balanced scenario.

more difficult for the relay to decode a PNC packet when one user has a low SNR. When the downlink transmission (the second hop) fails, blindly restarting the communication procedure from the first hop as in OLTD leads to a high average AoI, because a large portion of the time is spent on the uplink. By contrast, in ULTD, retransmitting old packets by ARQ in the downlink helps to improve the average AoI performance significantly. This is similar to the SNR-balanced scenario, in which both users have a low SNR with respect to the relay (i.e., the relay has a low probability of successfully decoding the PNC packets). Overall, OLTD does not perform well when there is at least one poor link in the two-hop TWRN, because it drops all the old packets once they are corrupted. In practical wireless environments with varying SNRs, ULTD is a viable solution to provide both stable and low average AoI.

Discussion: Differences between AoI and Conventional Metrics $-$ We now use the balanced SNR scenario to discuss the differences between the new AoI metric and the conventional metrics. Let us first consider the packet reception rate as the system performance metric in Fig. 11(a). The packet reception rate is defined as the number of packets received by the destination divided by the total number of packets sent by the sources under the generate-at-will model. In general, RPT receives all the update packets by ARQ, while OLTD has the lowest packet reception rate because it drops old packets once the uplink or downlink transmission fails. The packet reception rate of ULTD falls in between. Since ULTD achieves a moderate packet reception rate, it is a promising solution for information update systems with packet reception rate requirements. For example, in vehicular networks, vehicles exchange their basic safety message (BSM) with each other via a road-side unit (RSU) as a relay. The vehicles send BSM update packets containing their latest status information, such as position. On the one hand, a vehicle requires high information freshness to acquire the current positions of other vehicles; on the other hand, it also needs a certain number of received packets to infer the trajectories of other vehicles to better help improve road safety. In other words, it is critical to receive BSM update packets with high information freshness and high packet reception rate, and ULTD can satisfy both requirements.

Fig. 11(b) plots the system throughput of the three protocols, which is defined as the average number of received packets of the two users per ms. Interestingly, although OLTD has the lowest packet reception rate among the three schemes, it has the highest system throughput when the SNR is higher than $8\ dB$. This is due to the less frequent need for ARQ at high SNRs and fewer control frames in OLTD that save airtime (see Fig. 7 for the MAC protocol of OLTD). Furthermore, Fig. 11(b) shows that the average system throughput is the same for RPT and ULTD. Theoretically, if we ignore the duration of control frames, the system throughput of both protocols is $2/\left(E[T_u] + E[T_d]\right)$ packets per time slot. However, both theoretical results (Fig. 6) and experimental results (Fig. 9) show that ULTD provides a much better average AoI performance than RPT does. This shows that, although having the same throughput, dropping old packets can increase information freshness as ULTD does.

As far as delay is concerned, Fig. 11(c) compares the packet delay of the successfully delivered packets for the three protocols. The packet delay is defined as the time from one user first sending a packet to the other user finally receiving it. Unlike average AoI, OLTD has the smallest packet delay among the three protocols. Theoretically, the delay of all packets delivered in OLTD is always two time slots (i.e., the meaning of "2" in the average AoI (18)), which is the lowest among the three protocols. This also means that if an information update system wants to minimize the instantaneous AoI upon a successful update, OLTD should be adopted, i.e., for each successful update, the received packet always contains the most up-to-date information in OLTD.[6] In addition, the theoretical average packet delays of RPT and ULTD are $E[T_u]+E[T_d^j]$ and $1+E[T_d^j]$, respectively, i.e., the packet delay of ULTD is shorter than that of RPT, which is also validated by our experimental results as shown in Fig. 11(c). Furthermore, the delay difference between ULTD and OLTD is small, indicating that ULTD is a promising solution to achieve low packet delay and low average AoI in the low SNR regime (see the average AoI result in Fig. 9).

6. In addition, comparing the MAC protocols of ULTD and OLTD (see Fig. 7), it is easy to see that the MAC protocol of OLTD is simpler (e.g., there are fewer control signals in OLTD due to the absence of ARQ). Hence, for information update systems aiming at implementation simplicity and having loose AoI requirements, OLTD can be adopted.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

7 Conclusions

In this paper, we have studied the average AoI of PNC-enabled TWRNs with and without ARQ. In particular, we put forth an uplink-lost-then-drop (ULTD) protocol that combines packet drop and ARQ to achieve high information freshness.

PNC is a key technique that turns the superimposed wireless signals into network-coded messages, thereby reducing the communication delay of TWRNs. With a shorter delay, end users can send update packets to each other more frequently, thus enhancing the information freshness. However, when an update packet is corrupted in any hop of a TWRN, one needs to decide to discard or retransmit the packet. A new packet always contains more recent information, but it may take more time to be delivered. Handling corrupted packets to achieve a low average AoI of TWRNs is the main focus of this paper.

Our theoretical analysis indicates that, in error-prone TWRNs, neither a non-ARQ scheme nor a pure ARQ scheme can achieve good average AoI performance. Unlike single-hop networks, a non-ARQ scheme where old packets are dropped when they get corrupted (i.e., OLTD) shows slight improvement in the average AoI performance over a classical ARQ scheme with no packet lost (i.e., RPT). Therefore, we strategically exploit both packet drop and ARQ: in ULTD, corrupted packets in the uplink are dropped, but corrupted packets in the downlink are retransmitted until successfully received. We believe the insights from ULTD are generally applicable to other two-hop networks. Experiments on software-defined radios indicate that when channel conditions for two users are unbalanced, the average AoI of ULTD is significantly lower than that of OLTD and RPT, indicating that ULTD is robust against time-varying channel conditions in practice. Moreover, our experimental results identify the key differences between AoI and conventional performance metrics, such as throughput and delay, thus providing insights into ARQ designs with heterogeneous quality-of-service requirements.

References

[1] Y. Sun, I. Kadota, R. Talak, E. Modiano, and R. Srikant, *Age of Information: A New Metric for Information Freshness*. Morgan & Claypool, 2019.

[2] M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "On the role of age of information in the internet of things," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 72–77, Dec. 2019.

[3] A. Kosta, N. Pappas, and V. Angelakis, "Age of information: A new concept, metric, and tool," *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017.

[4] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. IEEE SECON*, 2011, pp. 350–358.

[5] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[6] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proc. ACM MOBICOM*, 2006, pp. 358–365.

[7] M. Zhang, G. G. M. N. Ali, P. H. J. Chong, B.-C. Seet, and A. Kumar, "A novel hybrid mac protocol for basic safety message broadcasting in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4269–4282, Oct. 2020.

[8] S. C. Liew, L. Lu, and S. Zhang, *A Primer on Physical-Layer Network Coding*. Morgan & Claypool Publishers Synthesis Lectures on Communication Networks, 2015.

[9] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: John Wiley & Sons, Inc., 1968.

[10] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge university press, 2005.

[11] M. Xie, Q. Wang, J. Gong, and X. Ma, "Age and energy analysis for LDPC coded status update with and without ARQ," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 388–10 400, Oct. 2020.

[12] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "A general formula for the stationary distribution of the age of information and its application to single-server queues," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 8305–8324, Dec. 2019.

[13] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 2731–2735.

[14] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Minimizing the age of information through queues," *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 5215–5232, Aug 2019.

[15] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637–2650, Dec 2018.

[16] C. Joo and A. Eryilmaz, "Wireless scheduling for information freshness and synchrony: Drift-based design and heavy-traffic analysis," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2556–2568, Dec 2018.

[17] I. Kadota, A. Sinha, and E. Modiano, "Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1359–1372, Aug 2019.

[18] N. Lu, B. Ji, and B. Li, "Age-based scheduling: Improving data freshness for wireless real-time traffic," in *Proc. ACM MOBICOM*, New York, NY, USA, 2018, pp. 191–200.

[19] K. Chen and L. Huang, "Age-of-information in the presence of error," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 2579–2583.

[20] J. P. Champati, R. R. Avula, T. J. Oechtering, and J. Gross, "Minimum achievable peak age of information under service preemptions and request delay," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1365–1379, May 2021.

[21] H. Pan and S. C. Liew, "Information update: TDMA or FDMA?" *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 856–860, Jun. 2020.

[22] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7468–7482, Jul. 2019.

[23] S. Feng and J. Yang, "Age of information minimization for an energy harvesting source with updating erasures: Without and with feedback," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5091–5105, May 2021.

[24] B. Zhou and W. Saad, "Performance analysis of age of information in ultra-dense internet of things (IoT) systems with noisy channels," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[25] E. T. Ceran, D. Gündüz, and A. György, "Average age of information with hybrid ARQ under a resource constraint," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1900–1913, Mar. 2019.

[26] Y. Gu, H. Chen, Y. Zhou, Y. Li, and B. Vucetic, "Timely status update in Internet of things monitoring systems: An age-energy tradeoff," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5324–5335, Jun. 2019.

[27] H. Pan, S. C. Liew, J. Liang, V. C. M. Leung, and J. Li, "Coding of multi-source information streams with age of information requirements," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1427–1440, May 2021.

[28] M. Xie, J. Gong, X. Jia, and X. Ma, "Age and energy tradeoff for multicast networks with short packet transmissions," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 6106–6119, Sep. 2021.

[29] V. Kavitha, E. Altman, and I. Saha, "Controlling packet drops to improve freshness of information," *Technical report*, available at: https://arxiv.org/abs/1807.09325.

[30] S. Feng and J. Yang, "Adaptive coding for information freshness in a two-user broadcast erasure channel," in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3166155, IEEE Transactions on Mobile Computing

[31] A. M. Bedewy, Y. Sun, and N. B. Shroff, "The age of information in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1248–1257, Jun. 2019.

[32] J. Lou, X. Yuan, S. Kompella, and N.-F. Tzeng, "Boosting or hindering: AoI and throughput interrelation in routing-aware multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1008–1021, Jun. 2021.

[33] M. Xie, J. Gong, and X. Ma, "Age and energy tradeoff for short packet based two-hop decode-and-forward relaying networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[34] ——, "Age-energy tradeoff in dual-hop status update systems with the m-th best relay selection," in *IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5.

[35] H. Pan, J. Liang, S. C. Liew, V. C. M. Leung, and J. Li, "Timely information update with nonorthogonal multiple access," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4096–4106, Jun. 2021.

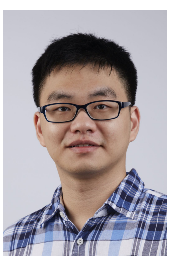
[36] B. Li, H. Chen, N. Pappas, and Y. Li, "Optimizing information freshness in two-way relay networks," in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2020, pp. 893–898.

[37] S. Salamat Ullah, S. C. Liew, G. Liva, and T. Wang, "Short-packet physical-layer network coding," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 737–751, Feb. 2020.

[38] IEEE802.11ac-2013, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 4: Enhancements for very high throughput for operation in bands below 6 GHz."

[39] L. You, S. C. Liew, and L. Lu, "Reliable physical-layer network coding supporting real applications," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2334–2350, Aug. 2017.

Victor C. M. Leung (S'75-M'89-SM'97-F'03) is a Distinguished Professor of Computer Science and Software Engineering at Shenzhen University, China. He is also an Emeritus Professor of Electrical and Computer Engineering and Director of the Laboratory for Wireless Networks and Mobile Systems at the University of British Columbia (UBC), Canada. His research is in the broad areas of wireless networks and mobile systems, and he has published widely in these areas. Dr. Leung is serving on the editorial boards of the IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Access, IEEE Network, and several other journals. He received the 1977 APEBC Gold Medal, 1977-1981 NSERC Postgraduate Scholarships, IEEE Vancouver Section Centennial Award, 2011 UBC Killam Research Prize, 2017 Canadian Award for Telecommunications Research, 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and 2018 ACM MSWiM Reginald Fessenden Award. He co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, 2017 IEEE Systems Journal Best Paper Award, 2018 IEEE CSIM Best Journal Paper Award, and 2019 IEEE TCGCC Best Journal Paper Award. He is a Fellow of IEEE, the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of "Highly Cited Researchers".



Haoyuan Pan (S'16-M'19) received the B.E. and Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2014 and 2018, respectively. He was a Post-Doctoral Fellow with the Department of Information Engineering, CUHK, from 2018 to 2020. He is currently an assistant professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include wireless communications and networks, physical-layer network coding (PNC), Internet of Things (IoT), and age of information (AoI).

Jianqiang Li received his B.S and Ph.D. degree from South China University of Technology in 2003 and 2008. He is a professor at the College of Computer and Software Engineering of Shenzhen University. He led three projects of the National Natural Science Foundation, and three projects of the Natural Science Foundation of Guangdong province, China. His major research interests include robotic, hybrid systems, internet of things and embedded systems.

Tse-Tin Chan (S'16-M'20) received the B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2014 and 2020, respectively. He is currently an Assistant Professor with the Department of Computing, The Hang Seng University of Hong Kong (HSUHK), Hong Kong. His research interests include wireless communications and networking, and machine learning. He is currently working on research about age of information (AoI), non-orthogonal multiple access (NOMA), and physical-layer network coding (PNC).