On-Time Communications Over Fading Channels

Yan Li, Yunquan Dong, Member, IEEE, Jian Wang, Member, IEEE, and Byonghyo Shim, Senior Member, IEEE

Abstract—We consider the on-time transmissions of a sequence of packets over a fading channel. Different from traditional in-time communications, we investigate how many packets can be received \( \delta \)-on-time, meaning that the packet is received around an expected slot with a deviation no larger than \( \delta \) slots. In this framework, we first derive the on-time reception rate of the random transmission scheme when no packet-controlling is used. Our analytical and simulation results show that the on-time reception rate of random transmissions decreases (to zero) with the sequence length. To improve the on-time reception rate, we further propose to schedule the packets by delaying, dropping, or repeating the transmissions. Specifically, we model the packet scheduling problem as a Markov decision process (MDP) and then obtain the optimal scheduling policy using an efficient iterative algorithm. By using the optimal packet scheduling, the on-time reception rate converges to a much larger constant, thus ensuring better on-timeliness. Moreover, we show that the on-time reception rate increases if the target reception interval and/or the deviation tolerance \( \delta \) is increased, or the randomness of the fading channel is reduced. By optimally scheduling the packets and choosing these parameters, therefore, wireless channels will be more suitable for the latency-sensitive applications.

Index Terms—Information freshness, On-time communications, on-time reception rate, packet scheduling.

I. INTRODUCTION

With the rapid development of the 5G communications and Internet-of-Things (IoT) technology, billions of smart devices will be connected to the internet to enable efficient interactions between the physical world and its digital counterpart. Surge of machine-type communications opens up the possibilities for various industry applications. Among these, applications like industrial sensing and controlling, remote surgery, and automatic driving require an extremely low latency (e.g., end-to-end delay being smaller than 10 ms) and a very small jitter (approximately several milliseconds) [1], [2], [3], [4]. For example, communications between the sensor, actuators, and controller of an industrial Internet should be completed on-time with a deterministic delay between 1 and 10 ms [5]; the braking/steering commands and advanced driver assistance systems (ADAS) type data need to be delivered to/from the actuators/sensors with a deterministic latency being less than 1 ms [3].

In a nutshell, deterministic information delivery has become one of the biggest challenges of modern wireline and wireless communications.

Thus far, successful implementations of deterministic networks include the latency-sensitive networking (TSN) and the 5G ultra reliable low latency communication (URLLC) [3], [4]. As a wireline network based on the Ethernet protocol, TSN reduces physical and link layer delays (and jitters) by IEEE 802.1AS clock synchronization, IEEE 802.1 Qcc flow reservation, and IEEE 802.1 QCh cyclic queuing [6]. By delivering the traffic with deterministic delays, TSN has become the basis of latency-sensitive applications like industrial automation and automotive driving. In an in-vehicle TSN network, for example, the communication between the vehicle control unit (VCU) and the cameras, radars, lidars, and the positioning module, is performed with a deterministic delay [3]. The deterministic inter-vehicle communications can be realized by 5G URLLC (see Fig. 1).

By adopting techniques such as flexible frame structure, advanced coding scheme, and timely scheduling mechanism, URLLC achieves an average delay as low as 0.16ms [4]. Although a proper combination of TSN and URLLC can be useful for the latency sensitive vehicular communications to some extent [7], we need to address the real-world issues more carefully in the practical system design. For example, the wireline TSN networks are expensive, less flexible and not quite scalable. 5G URLLC relies heavily on the computing power at the gNB.
(node B) and smart phones. In addition, the smart phones do not exchange information through a gNB directly even if they are close to each other, since 5G URLLC is designed for wide area networks other than local networks. To extend the applications of wireless local networks, therefore, we need to improve the transmission determinacy when wireless channels and less capable transmitters are used.

A. Motivations and Main Contributions

Due to the fading and time-varying property, the packet transmissions over wireless channels are random and rate-variable, i.e., with uncertainty. Many works have been proposed to compensate for the randomness of the channels and thus provide deterministic transmissions with smaller delay and delay jitter. However, the on-time reception of packets, i.e., delivering information with deterministic delays, has been rarely investigated.

If we can ensure that the packets is delivered on-time over wireless channels, corresponding delays and system status will be more predictable so that variety of latency-sensitive applications in 5G and 6G can be supported. For example, in a wireless implementation of the intra-vehicle network in Fig. 1, the VCU controls the vehicle by making a decision and then sending it to the powertrain system periodically. If the sensors can deliver the sensed data to VCU at the decision epoch, excessive buffering or waiting latency of the packets can be reduced. Another example is the communications in energy-limited wireless sensor networks. If the on-time transmissions among sensors are guaranteed, the sensors can stay in the sleeping mode except for some predefined short reception windows, improving energy efficiency considerably.

In this article, we improve the on-time performance of packets transmissions over wireless fading channels by using link-layer scheduling techniques. To be specific, we investigate whether and how the packets can be received on-time under the physical layer reliability constraint. Main focus of our work is to come up with analytical metrics to evaluate how many packets are received on-time and design packet scheduling schemes to enhance the chance of the packets being received on-time.

The contributions of the article are summarized as follows:

- We propose a method to evaluate the on-time performance of communications in terms of δ-on-time and on-time reception rate, from which we can investigate the deviation level and proportion of packets being received in the packet small deviation.
- We derive the on-time reception rate of the system when the packets are transmitted over the fading channel without any controlling. We show that the on-time reception rate decreases to zero as the sequence length goes to infinity. Specifically, we analyze the probability that each packet can be received δ-on-time and show that the average number of packets received δ-on-time is equivalent to the sum of these probabilities.
- We improve the on-time reception rate over fading channels by introducing the policies including delaying, dropping, and repeating packets. We also solve the optimal packet scheduling policy through an MDP formulation. We show that the optimal packet scheduling strategy improves the on-time reception rate of the system efficiently over the conventional random transmission strategy.

B. Related Works

The latency-sensitive communications and networks have received much attention in recent years, among which the wired TSN [6] and wireless 5G deterministic network [8] are the most popular ones. In TSN, IEEE 802.1 working group has developed a series of Ethernet protocols for the latency-sensitive applications, in which scheduling traffics with timed transmission gates, filtering traffics based on priorities, and forwarding traffics with repeating circles are some representative link-layer techniques. For example, a computational efficient solution to the fully deterministic 802.1Qbv scheduler was presented in [9]; a bandwidth-efficient TSN scheduler was investigated through a size-based queueing method in [10]; an asynchronous traffic scheduling algorithm achieving both low delay and low implementation complexity has been proposed in [11]; and an online scheduling approach was proposed to deal with the dynamic virtual machine migrations in multi-cast TSN networks in [12]. In [13], a simple hardware enhancement of switches has been proposed to increase the schedulability and throughput of time-triggered traffics. Routing is also an important part of TSN networks, for which an ILP-based scheduling and degree of conflict aware multi-path routing scheme was proposed in [14]. A joint routing-scheduling optimization for time-triggered Ethernet networks was investigated in [15]. Since 5G URLLC aims at transmitting packets with ultra low delay (2~12 ms) and ultra high reliability (99.999%), it is possible to support dedicated network referred to as the 5G deterministic networking (5GDN or 5G DeNet) by providing predictable and deterministic services [8]. As discussed in [16], 5GDN can be applied to many latency sensitive applications, such as real-time monitoring, remote controlling, material management, mass access, and product life-cycle management.

In these time-critical networks, the timeliness performance is often measured by metrics such as delay and delay jitter, which are defined from the transmitter perspective. In fact, delay measures the time that packet experiences after the beginning of transmission at the transmitter and delay jitter characterizes the deviation from its average. Depending on whether the delay bound (deadline) can be violated or not, the communication requirements can be further classified into soft real-time and hard real-time networks, which are popular in the computer society and automation society [17]. There have been some works analyzing the moments and bounding behavior of delay jitter of multimedia traffic flows [18], [19], [20]. However, delay jitter in the information transmission over wireless channels has been rarely investigated.

Contrary to the delay, a timeliness metric called age of information (AoI) has been proposed from a receiver perspective [21]. AoI is defined as the age of the latest available packet at the receiver and characterizes the in-timeliness of received packets.

In this framework, the arrivals, transmissions, and departures of packets can be modeled by a queueing system, based on which
the average AoIs and deadline violations can be computed [21], [22]. By providing an analytical method for information freshness, the AoI theory also enables further scheduling of packets and links. For example, the optimal link scheduling under some throughput and energy constraints has been studied in [23] and [24].

In this research area, however, the on-timeliness of wireless communications has been rarely investigated.

C. Organizations

This rest of the article is organized as follows. In Section II, we discuss the concepts of on-time reception, the channel model, and the source model. In Section III, we analyze the probability that each packet is received on time and the on-time reception rate of the random transmission scheme. In Section IV, we present three packet controlling strategies, viz., delaying, dropping, and repeating. In Section V, we formulate an MDP problem to solve the optimal packet scheduling policy. For a sequence of packet transmissions, we also derive the corresponding reward in theory in this section. In Section VI, we present the simulation and numerical results on the on-time reception rates over the fading channel, with both the random transmission scheme and the optimal packet scheduling policy. Finally, we conclude the article in Section VII.

II. SYSTEM MODEL

A. Definition of on Time

We consider the sequential transmissions of packets over a fading channel. Due to the fading property of the channel, the transmission time (e.g., the number of slots in 4G LTE and 5G NR) to successfully deliver a packet varies randomly. Suppose the destination node calculates and makes decisions periodically. To avoid packet buffering and reduce the idle time waiting for packets, these packets are expected to be received by the destination node at a sequence of preset slots (i.e., \( \{ T_{tg}, 2T_{tg}, 3T_{tg}, \cdots \} \)) with fixed intervals. The on-timeliness of the corresponding transmissions are defined as follows.

**Definition 1:** The \( m \)-th packet is said to be received strictly on-time if the packet is received by the destination node exactly in the \( mT_{tg} \)-th slot.

Since the transmissions time of the packets are random, strictly on-time transmission over fading channels is quite difficult so we relax the definition and allow the packet reception to deviate from the target slot with a maximum tolerance of \( \delta \) slots. A slightly relaxed version of the on-timeliness is defined as follows.

**Definition 2:** The \( m \)-th packet is said to be received \( \delta \)-on-time if the packet is received by the destination node in any of the slots among \( \{ mT_{tg} - \delta, mT_{tg} - \delta + 1, \ldots, mT_{tg} + \delta \} \) (see Fig. 2). The period \( \{ mT_{tg} - \delta, mT_{tg} - \delta + 1, \ldots, mT_{tg} + \delta \} \) is referred to as the target reception range of the \( m \)-th packet.

It is clear that the \( \delta \)-on-time returns to the strictly on-time if we set the deviation tolerance to 0 (i.e., \( \delta = 0 \)).

**Remark 1:** Note that both strictly on-time and \( \delta \)-on-time are defined from the receiver perspective. Specifically, a packet is on-time only if it is received with the expectation of the receiver. On the contrary, the traditional delay metric measures the time from the transmission to the reception of a packet. Thus traditional delay is a metric defined from the transmitter perspective.

B. Channel and Source Models

In wireless systems, the transmission path of wireless signals changes from time to time due to the reflections and scatters of the ambient environment, as well as the relative movement between the transmitter and receiver. Specifically, the signal may arrive at the receiver through multiple paths and in different epochs, resulting in multi-path delay expansions and inter-symbol interfererences; the multi-path signals also arrive at the receiver with different angles, resulting in a stronger fading. In most practical scenarios, however, the multi-path components of the received signal is not so rich so that the maximum multi-path delay would be smaller than the length of a slot. In most low to medium-speed mobile (e.g., \( \pm 60 \text{ km/h} \)) communications, the doppler frequency offset is relatively small so that the channel gain do not change much within each slot of several channel uses. Therefore, the block fading channel model is widely used in the research of wireless communications [25].

In this article, we consider the packet transmissions over a block fading channel. In this channel model, the channel gain stays fixed within a slot and varies independently between slots. We also assume that the source node transmits information with a fixed data rate to enhance deterministic transmission. Thus, the efficiency of transmissions over the block fading channel can be characterized by a probability of outages, in which the channel state of the slot is poor so that the transmission of current packet falls.

We denote the power gain distribution of the channel as \( f_\gamma(x) \). Let \( d \) be the distance between the source and destination nodes and \( \alpha \) be the path loss exponent, and \( P_t \) be the transmit power of the source node. In the \( n \)-th slot, the power of the received signal at the destination node can then be expressed as \( P_{r,n} = \gamma_n P_t / d^{\alpha} \), in which \( \gamma_n \) is the random channel power gain in the \( n \)-th slot. The signal-to-noise ratio (SNR) at the destination node can be expressed as \( \rho_n = \gamma_n P_t / (d^{\alpha} \sigma^2) \), in which \( \sigma^2 \) is the power of Gaussian white noises.

Although acquiring channel state information (CSI) by the channel estimations and feedbacks have been widely used, transmitter-CSI based variable rate transmission increases the complexity of transceiver design and requires additional overhead. In this article, we adopt an outage-capacity formulation, in which the source node transmits at a constant rate [26]. In this context, we say the decoding is successful if the received SNR exceeds a predefined threshold \( V_T \). We assume that if the decoding is successful, the transmission of a packet is completed in one slot. If the destination node cannot decode the packet, the
source node will re-transmit the packet in the next slot until it receives an ACK from the destination node. In particular, the probability that the destination node successfully decodes a packet from the received signal can be expressed as

$$p = Pr \{ \rho_n > V_1 \} = \int_{\frac{-\infty}{\sqrt{N_0 + \sigma^2}}}^{\infty} f_\gamma(x)dx.$$  \hspace{1cm} (1)

Using $p$, we can model the physical layer reliability of the fading channel. The actual transmission time $S$ to successfully deliver a packet over the fading channel follows the geometric distribution with parameter $p$:

$$Pr \{ S = j \} = p(1 - p)^{j-1}, \quad j = 1, 2, \ldots \tag{2}$$

One can see that the on-time delivery of a packet is mainly determined by the transmission process, not the packet generation process. Generating a new packet and buffering it in the queue before the transmission does not increase the on-time reception probability. On the packet generations, therefore, we assume that the $(m + 1)$-st packet will be generated immediately after the completion of the $m$-th packet transmission. Irrespective of the packet generation time, the desired reception time of the $(m + 1)$-st packet is $(m + 1)\tau_{tgt}$. In case a packet is generated after its desired reception time, a reasonable scheduling scheme will drop the packet, as shown in Sections IV-B, V, and Fig. 11.

C. On-Time Reception Rate

The primary concern of this work is to analyze the number of packets that are received on time, i.e., within their respective target reception ranges. Among $M$ transmitted packets, let $\kappa_M$ be the number of packets received $\delta$-on-time. Then the on-time reception rate $\varrho_M$ is defined as

$$\varrho_M = \frac{\kappa_M}{M}. \tag{3}$$

We would like to mention that the on-time reception rate is closely related to the length of the packet sequence $M$. Specifically, the larger $M$ we have, the smaller the on-time reception rate would be. This is because when the number of transmit increases, there would be more unexpectedly large transmission times, which obstructs on-time reception of the subsequent packets. As will be discussed in Sections III and VI, the on-time reception rate of the random transmission scheme decreases monotonically with $M$. In this article, we will maximize the on-time reception rate of the system by scheduling the transmissions of packets.

III. ON-TIME RECEPTION RATE OF RANDOM TRANSMISSIONS

In this section, we consider the on-time reception rate of the transmissions over the fading channel in the absence of scheduling and controlling. By analyzing the probability that each packet is received $\delta$-on-time, we can estimate the average number of packets received with $\delta$-on-time and the corresponding on-time reception rate.

We denote the total number of packets to be transmitted as $M$, the transmission time of the $m$-th packet as $\tau_m$. We denote the event that the $m$-th packet is received $\delta$-on-time as $x_m$ and corresponding probability as $P(x_m)(m = 1, 2, \ldots, M)$.

First, the probability that the first $(m = 1)$ packet is received $\delta$-on time is

$$P(x_1) = Pr \{ T_{tgt} - \delta \leq \tau_1 \leq T_{tgt} + \delta \}.$$  \hspace{1cm} (4)

Note that its transmission time satisfies $\tau_m \geq 1$ and follows the geometric distribution with parameter $\rho$ (see (2)). In case $T_{tgt} \leq 1 + \delta$, we have $T_{tgt} - \delta \leq 1$ and (4) is equivalent to

$$P(x_1) = \sum_{i=1}^{T_{tgt} + \delta} Pr \{ \tau_1 = i \} = 1 - (1 - p)^{T_{tgt} + \delta}. \tag{5}$$

In case $T_{tgt} > 1 + \delta$, we have

$$P(x_1) = \sum_{i=1}^{T_{tgt} + \delta} Pr \{ \tau_1 = i \} = (1 - p)^{T_{tgt} - \delta - 1} - (1 - p)^{T_{tgt} + \delta}. \tag{6}$$

By combining (5) and (6), we can express the probability of the first packet being received $\delta$-on-time as

$$P(x_1) = \begin{cases} 1 - (1 - p)^{T_{tgt} + \delta}, & \text{if } T_{tgt} \leq 1 + \delta \\ (1 - p)^{T_{tgt} - \delta - 1} - (1 - p)^{T_{tgt} + \delta}, & \text{if } T_{tgt} > 1 + \delta. \end{cases} \tag{7}$$

For the $m$-th packet, which is intended to be received within $\{mT_{tgt} - \delta, mT_{tgt} - \delta + 1, \ldots, mT_{tgt} + \delta\}$, the total transmission time $\sum_{k=1}^{m} \tau_k$ follows the negative binomial distribution with parameter $p$. The following proposition describes the probability of a packet being received $\delta$-on-time.

**Lemma 1:** For a sequence of $M$ packet transmissions over the fading channel, the probability of the $m$-th packet being received $\delta$-on-time is given by

$$P(x_m) = \begin{cases} \sum_{k=m}^{mT_{tgt} + \delta} \binom{k-1}{m-1} p^m(1-p)^{k-m}, & \text{if } mT_{tgt} \leq m + \delta \\ \sum_{k=mT_{tgt} - \delta}^{mT_{tgt} + \delta} \binom{k-1}{m-1} p^m(1-p)^{k-m}, & \text{if } mT_{tgt} > m + \delta, \end{cases} \tag{8}$$

in which $p$ is the probability of successful reception in a slot, $\tau_{tgt}$ is the target reception interval, $\delta$ is the deviation tolerance.

**Proof:** See Appendix A.

In Fig. 3, we compare $P(x_m)$ obtained by analytical and simulation results, when $p = 0.2$ and $T_{tgt} = 5$. We observe that $P(x_m)$ decreases with packet index $m$. That is, the $P(x_m)$ of the $m$-th packet is no larger than that of previous packets. It is also seen that $P(x_m)$ increases with the deviation tolerance $\delta$.

Let $x_k^M$ be the event that a subset of $k$ (not necessarily successive) packets out of $M$ are received $\delta$-on-time and $P(x_k^M)$ be the corresponding probability. Then, among $M$ packets, the expected number of packets received $\delta$-on-time is

$$\kappa_M = \sum_{k=1}^{M} k P(x_k^M). \tag{9}$$
As shown in the following theorem, $\kappa_M$ can be further expressed in terms of $P(x_k)$.

**Theorem 1:** When we transmit a sequence of $M$ packets over the fading channel, the probability that packets are received $\delta$-on time satisfies

$$
\sum_{k=1}^{M} k P(x_k^M) = \sum_{k=1}^{M} P(x_k),
$$

where $P(x_k)$ is the probability that the $k$-th packet is received $\delta$-on time (cf. Proposition 1) and $P(x_k^M)$ is the probability for $k$ out of the $M$ packets being received $\delta$-on time.

**Proof:** See Appendix B.

From Theorem 1, we observe that the average number of packets that can be received $\delta$-on time is equal to the sum of the probabilities that each packet is received $\delta$-on time. It should be noted that the result in Theorem 1 is different from the known equation $\mathbb{E}(X) = \sum \Pr(X > x)$, since $P(x_k)$ is physically defined as the probability for the $k$-th packet being received on-time and is different from the probability $\Pr(K > k) = \sum_{j=k+1}^{M} P(x_j^M)$. By combining Proposition 1 and Theorem 1, we obtain the on-time reception rate as

$$
\vartheta_M = \frac{\kappa_M}{M} = \frac{1}{M} \sum_{k=1}^{M} P(x_k).
$$

**Remark 2:** As the number of packets $M$ increases, the reception epochs of the packets become more random so that the ratio for the packets being received on-time decreases. In particular, on-time reception rate goes to zero as $M$ approaches infinity, as shown below.

In the case $T_{tgt} = 1$, (8), (9) and (10) yield

$$
\kappa = \sum_{m=1}^{M} \sum_{k=m}^{mT_{tgt}+\delta} C_{k-1}^{m-1} p^m (1-p)^{k-m}
$$

$$(a) \sum_{m=1}^{M} \sum_{k=m}^{mT_{tgt}+\delta} p C_{k-1}^{m-1} p^{m-1}(1-p)^{m-1}(1-p)^{(k-1)(1-p)}
$$

in which (a) is because that the binomial random variable $\xi_{k-1}$ with parameters $(k-1, p)$ has the maximum probability of outcome at $\xi_{k-1} = p(k - 1)$ (i.e., $\Pr(\xi_{k-1} = m - 1) \leq \Pr(\xi_{k-1} = p(k - 1))$), (b) follows the fact the maximum probability of outcome of a binomial distribution $B(k, p)$ decreases with $k$.

According to the De Moivre-Laplace Central Limit Theorem, there exists a sufficiently large $m_0$ that for any $m \geq m_0$, the binomial distribution is approximated by the normal distribution:

$$
C_{m_0}^{k} p^k (1-p)^{m-k} = \frac{1}{\sqrt{2\pi mp(1-p)}} e^{-\frac{(k-mp)^2}{2mp(1-p)}},
$$

By combining (12) and (13), we have

$$
\kappa < c_0 + p(\delta + 1) \sum_{m=m_0}^{M} \frac{1}{\sqrt{2\pi (m - 1) p(1-p)}}
$$

$$(a) < c_0 + p(\delta + 1) \int_{m_0 - 1}^{M} \frac{1}{\sqrt{2\pi x - 1}} dx
$$

$$
= c_0 + c_1(\sqrt{M - 1} - \sqrt{m_0 - 2}),
$$

in which $c_0 = \sum_{m=1}^{m_0-1} b(\delta + 1) C_{m-1}^{p(m-1)} p^{m-1}(1-p)^{m-1}$ $(\varpi = 1-p)$ is a finitely large partial sum, $c_1 = \frac{2p(\delta + 1)}{\sqrt{2\pi p(1-p)}}$, and (a) follows $\sum_{m=m_0}^{M} \frac{1}{\sqrt{x - 1}} \leq \sum_{m_0 - 1}^{M} \frac{1}{\sqrt{2\pi x - 1}} dx$.

By combining (11) and (12), the limitation of the on-time reception rate can be expressed as

$$
\vartheta_M = \lim_{M \to \infty} \frac{\kappa_M}{M} = \frac{c_0 + c_1(\sqrt{M - 1} - \sqrt{m_0 - 2})}{M} = 0.
$$

Likewise, it can be proved that $\lim_{M \to \infty} \vartheta_M = 0$ for the case $T_{tgt} > 1$.

**IV. ON-TIME RECEPTION RATE OF CONTROLLED TRANSMISSIONS**

In this section, we discuss strategies to improve the on-time reception rate of the system. We first discuss the low on-time reception rate of random transmissions and then present delaying, dropping, and repeating mechanism to improve the on-time reception rate.

**A. Drawbacks of Random Transmissions**

In the random transmission scheme, any packet that is not received on-time degrades system performance not only by itself, but also affects the transmissions of the subsequent packets. For example, if a packet is received before its target reception range, the probability that the next packet is received on-time will also be reduced since there will be more than enough time for its transmission so that the packet might be received earlier than the desired time. If the transmission time of a packet is large so the packet is received after the target reception range, the probability for the subsequent packet being received on-time
may also be reduced (even to zero), since the remaining time for its transmission is shortened. Therefore, the on-time reception rate of the random transmissions is often relatively small.

**B. Controlling With Delaying, Dropping or Repeating**

We propose three strategies to control (on demand) the transmission of packets and thus improve the on-time reception rate of the system.

1) **Delaying**: At the beginning of a packet transmission, the delaying strategy would delay the transmission for a period of \( n_d (n_d \geq 0) \) slots. This strategy is especially useful when the previous packet is received before its target reception range.

2) **Dropping**: In this strategy, the packets can be dropped on demand so that the next packet could be transmitted immediately. This strategy is useful if the transmission time of the previous packet is so large that the subsequent packet completely misses the chance to be received \( \delta \)-on time.

3) **Repeating**: If a packet is received before its target reception range, the repeat strategy allows the retransmission of the packet. This is especially useful when the destination node wakes up periodically only for a short period. We limit the retransmissions to a finite number of times and denote the maximum number of allowed retransmissions as \( n_r (n_r \geq 0) \).

**Remark 3**: In controlling schemes, the parameters \( n_d \) and \( n_r \) will be optimally chosen by maximizing the long-term average on-time reception rate of the system. In case \( n_d = 0 \) or \( n_r = 0 \), the delaying strategy and the repeating strategy return to random transmissions.

In the following, we investigate the performance of the repeating strategy for the single packet \( (M = 1) \) transmission over the fading channel. We denote the target reception interval as \( T_{tgt} \), the deviation tolerance as \( \delta \) and the maximum number of retransmissions as \( n_r (n_r \geq 0) \).

**Proposition 1**: Considering the transmission of a single packet \( (M = 1) \) with the maximum number \( n_r \) of retransmissions, the probability distribution function of the transmission time \( S_{n_r} \) is given by

\[
Pr \{ S_{n_r} - T_{tgt} > j \} = \begin{cases} 
1, & j < n_r - T_{tgt} \\
\sum_{m=0}^{n_r} C_{n_r}^m (1-p)^j T_{tgt}^m, & j \geq -1 - \delta \\
\sum_{m=0}^{n_r} C_{j+T_{tgt}}^m (1-p)^j T_{tgt}^m, & j < -1 - \delta 
\end{cases}
\] (16)

if \( T_{tgt} \geq 1 + n_r + \delta \). In the case \( T_{tgt} \leq 1 + \delta \), we have

\[
Pr \{ S_{n_r} - T_{tgt} > j \} = Pr \{ S_0 - T_{tgt} > j \} = \begin{cases} 
(1-p)^j T_{tgt}, & j \geq -T_{tgt} \\
1, & j < -T_{tgt} \end{cases}
\] (17)

In case \( 1 + \delta < T_{tgt} < 1 + n_r + \delta \), we have

\[
Pr \{ S_{n_r} - T_{tgt} > j \} = Pr \{ S_{T_{tgt} - 1 - \delta} - T_{tgt} > j \} = \begin{cases} 
(1-p)^j T_{tgt} + 1, & j \geq -1 - \delta \\
1, & j < -1 - \delta 
\end{cases}
\] (18)

**Proof**: The detailed proof of the proposition is provided in Appendix C in [28].

We see from Proposition 1 that the repeating strategy changes the distribution of the transmission time of each packet and improves the on-time performance of the transmissions. In fact, the probability that a packet is received \( \delta \)-on time increases substantially by repeating the transmission of the packet for some times. In Fig. 4, we present the complementary cumulative distribution functions (CCDF) of \( S_{n_r} - T_{tgt} \) for various \( n_r \) (the maximum numbers of retransmissions), in which the probability of successful transmission is set to \( p = 0.2 \), the target reception interval is set to \( T_{tgt} = 20 \), and the deviation tolerance is set to \( \delta = 1 \). We plot the target reception range, i.e., the area in interval \((-2, 1]\), by the shaded area. We denote the y-coordinates of the intersections of each CCDF curve and the boundary of the shaded area as \( q_1, n_r \) and \( q_2, n_r \). It is clear that \( q_2 - q_1 = Pr \{ S_{n_r} - T_{tgt} \geq -2 \} - Pr \{ S_{n_r} - T_{tgt} > 1 \} \) is the probability that the packet is received \( \delta \)-on time. From Fig. 4, we observe that when \( n_r \) increases, the probability that the packet is received \( \delta \)-on time will increase, while the probabilities of the packets being received before (i.e., \( 1 - q_2, n_r \)) and after (i.e., \( q_1, n_r \)) the target reception range decreases and increases, respectively. Due to the randomness of the fading channel, however, it is difficult to guarantee that all of the packets are received at their expected reception epochs exactly. For example, we observe that in case
\( n_t = T_{\text{tgt}} = 20 \), the probability that a packet is received after the right boundary of the target reception range is still quite large. Thus, the gain in the probability of \( \delta \)-on-time obtained by the repetition of the packets is also limited.

V. OPTIMAL PACKET SCHEDULING

In practical transmissions of a sequence of packets, a packet may either be received before or after its desired target reception range. To increase the probability of being received \( \delta \)-on-time for the following packets, controlling strategies such as the delaying, dropping, and repeating should be considered.

In this section, we try to maximize the on-time reception rate of the system by modeling the optimal packet scheduling problem as an MDP problem. Using the MDP formulation, we can determine the optimal controlling strategy of a packet in an online manner. That is, we shall determine the controlling strategy and the corresponding parameter (e.g., \( n_t \) or \( n_d \)) of a packet based on the state of the system at the starting time of its transmission. In particular, we solve the optimal scheduling policy using the MDP-based iterative algorithm.

To be specific, the state set, available actions, transition probabilities, reward functions, and optimal packet scheduling policy of the MDP problem are elaborated in the following subsections 1 to 5, respectively.

A. States

Since the controlling strategies are selected in the beginning of packet transmissions, we only need to consider the states of the system when a packet starts its transmission.

To be specific, we define the state \( s_m \) of the system as the difference between the transmission starting time and the target reception time of the packet. For example, suppose the \((m - 1)\)-st packet is received in slot \( n - 1 \) and thus the \( m \)-th packet has a chance to be transmitted from slot \( n \). In slot \( n \), the state of the system would then be \( s_m = mT_{\text{tgt}} - n + 1 \), in which \( mT_{\text{tgt}} \) is the time when the packet is expected to be received. Since the previous packet \( m - 1 \) may be received even later than the target reception time \( mT_{\text{tgt}} \) of the \( m \)-th packet, the state \( s_m \) could also be negative. The state set, therefore, would be \( S = \mathbb{Z} \), i.e., the integer set. In case the state of a packet is negative, the packet would most probably be dropped. Suppose the current state is \( s_m = i \), the transmission time of the packet is \( S \), and the next state \( s_{m+1} = j \), we have

\[
s_{m+1} = s_m - S + T_{\text{tgt}}.
\] (19)

B. Actions

In the beginning of the transmission of a packet \( m \), we schedule the packet by either delaying it by \( n_d \geq 0 \) slots, dropping it, or repeating the transmission by \( n_r \geq 0 \) times, which are referred to as taking an action \( a \). The set of all possible actions is referred to as the action set \( \mathcal{A} \). Moreover, the parameters \( n_d \) and \( n_r \) vary with the states \( s_m \) of packets.

C. Transition Probabilities

For a given state \( s_m = i \) and the corresponding action \( a \in \mathcal{A} \), we denote the transition probability to state \( s_{m+1} = j \) as \( p_{ij}(a) = \Pr\{s_{m+1} = j|s_m = i\} \) and have \( \sum_{j \in S} p_{ij}(a) = 1 \).

When taking different actions, the transition probabilities will be different. First, we consider the transition probability of random transmissions. By combing (2) and (19), and the fact \( S \geq 1 \), we have (cf. Fig. 5(a))

\[
p_{ij} = \begin{cases} 0, & j \geq i + T_{\text{tgt}} \\ p(1 - p)^{i-j+T_{\text{tgt}}-1}, & j < i + T_{\text{tgt}}. \end{cases}
\] (20)

Second, we consider the transitions of delaying the transmission by \( n_d \) slots and denote the corresponding probability as \( p_{ij}(n_d) \), for \( n_d = 0, 1, 2, \ldots \). Likewise, the transition probability of the delaying action is (see Fig. 5(b))

\[
p_{ij}(n_d) = \begin{cases} 0, & j \geq i + T_{\text{tgt}} - n_d \\ p(1 - p)^{i-n_d-j+T_{\text{tgt}}-1}, & j < i + T_{\text{tgt}} - n_d. \end{cases}
\] (21)

Third, given that the current state is \( s_m = i \) and the packet \( m \) is dropped, its transmission time would be deterministic, i.e., \( S = 0 \). In addition, the packet \( m + 1 \) will be generated and then transmitted immediately. Thus, the time for the transmission of the \((m + 1)\)-st packet to be received strictly on-time (i.e., its state) is \( j = i + T_{\text{tgt}} \) (cf. Fig. 5(c)). That is, the transition probability \( p_{ij} \) from state \( i \) to state \( j \) is

\[
p_{ij} = \begin{cases} 1, & j = i + T_{\text{tgt}} \\ 0, & j \neq i + T_{\text{tgt}}. \end{cases}
\] (22)

Fourth, when the packet is allowed to be retransmitted with a maximum number \( n_r \) retransmissions, we denote the transition probability from state \( i \) to state \( j \) as \( p_{ij}(n_r) \).

**Proposition 2:** When a packet is allowed to be retransmitted for at most \( n_r \) times and if \( i \leq 1 + \delta \), we have

\[
p_{ij}(n_r) = \begin{cases} 0, & j \geq i + T_{\text{tgt}} \\ p(1 - p)^{i-j+T_{\text{tgt}}-1}, & j < i + T_{\text{tgt}}. \end{cases}
\] (23)

if \( 1 + \delta < i < 1 + \delta + n_r \), we have

\[
p_{ij}(n_r) = \begin{cases} 0, & j \geq 1 + \delta + T_{\text{tgt}} \\ p(1 - p)^{\delta-j+T_{\text{tgt}}}, & j < 1 + \delta + T_{\text{tgt}}. \end{cases}
\] (24)
if \( i \geq 1 + \delta + n_r \), we have

\[
p_{ij} (n_t) = \begin{cases} 
0, & y \leq n_t \\
C_{y-1}^{n_t} p_{1+m} (1 - p)^{y-1-n_t}, & n_t < y \leq i - 1 - \delta \\
\sum_{m=0}^{n_t} C_{i-1-n_t}^{m} p_{1+m} (1 - p)^{y-1-m}, & y > i - 1 - \delta,
\end{cases}
\]

in which \( y = i - j + T_{tg} \).

**Proof:** See Appendix D in [28]. As an intuitive explanation, \( y = i - j + T_{tg} \) is time reserved for the transmissions and the retransmissions of packet \( m \) (see Fig. 5(d)). As long as the packet is received at the destination before the target reception range \([mT_{tg} - \delta, mT_{tg} + \delta]\) and the number of retransmissions is less than \( n_t \), the retransmission of packet \( m \) continues. \( \blacksquare \)

### D. Reward Function

When the state of the system transits from \( s_m = i \) to \( s_{m+1} = j \), we define the reward function \( r_{ij} \) as the number of packet received on-time.

\[
r_{ij} = \begin{cases} 
1, & \text{if } T_{tg} - \delta \leq j \leq T_{tg} + \delta \\
0, & \text{else},
\end{cases}
\]

Using (26), we can evaluate the number of packets received \( \delta \)-on time with the total reward of the system. Note that this number will be increased by one if the \( m \)-th packet is received \( \delta \)-on time and remains unchanged otherwise.

From a state \( i \) and with an action \( a \), we define the expected reward function as \( r(i, a) \), which can be calculated by

\[
r(i, a) = \sum_{j \in S} p_{ij}(a) r_{ij},
\]

in which \( p_{ij}(a) \) (see (20)–(25)) is the state transition probability under action \( a \). In the following, we investigate the reward functions of different controlling strategies case by case.

First, we denote the reward function of the random transmissions as \( R(i) \). By combining (20), (26), and (27), we have

\[
R(i) = \sum_{j = T_{tg} - \delta}^{T_{tg} + \delta} p_{ij}
\]

\[
= \begin{cases} 
0, & i \leq -\delta \\
(1 - p)^{i-1-\delta} (1 - (1 - p)^{i+2\delta}), & i > \delta \quad (28)
\end{cases}
\]

Second, we denote the reward function of the dropping strategy as \( R_{dp}(i, n_d) \), where \( n_d \) is the delay time. By replacing \( i \) in (28) with \( i - n_d \), we have

\[
R_{dp}(i, n_d)
\]

\[
= \begin{cases} 
0, & i \leq n_d - \delta \\
(1 - p)^{i-n_d-\delta} (1 - (1 - p)^{i+2\delta}), & i > \delta + n_d
\end{cases}
\]

(29)

Third, we denote the reward function of the dropping strategy as \( R_{dp}(i) \). Since a dropped packet can never be received on time, we immediately have

\[
R_{dp}(i) = 0. \quad (30)
\]

Fourth, we denote the reward function of the repeating strategy as \( R_{rp}(i, n_r) \), in which \( n_r \) is the maximum allowed retransmissions. By combining (23)–(27), the reward function of the repeating strategy can be expressed as

\[
R_{rp}(i, n_r) = \sum_{j = T_{tg} - \delta}^{T_{tg} + \delta} p_{ij}(n_r)
\]

\[
= \begin{cases} 
0, & i \leq -\delta \\
1 - (1 - p)^{i+\delta}, & -\delta < i \leq 1 + \delta \\
1 - (1 - p)^{i+2\delta}, & 1 + \delta < i \leq 1 + \delta + n_r
\end{cases}
\]

(31)

in which \( D = \sum_{m=0}^{n_r} C_{i-1-n_r}^{m} p_{1+m} (1 - p)^{i-1-\delta-m} \).

### E. Optimal Packet Scheduling Policy

A packet scheduling policy \( \pi \) is a rule to choose actions (i.e., delaying, dropping, or repeating) for each packet, i.e., a mapping from the state space \( S \) to the action space \( A \). Specifically, for each state \( s = i \) (can be negative), the corresponding element of \( \pi \) specifies which action should be taken for the current packet. In case the packet is delayed, \( \pi \) also indicates how long it should be delayed, i.e., determining \( n_d \); in case the packet is repeated, \( \pi \) also indicates how many times it could be retransmitted, i.e., determining \( n_r \); in case the packet should be randomly transmitted or dropped, no other parameters are needed to be determined.

For a sufficiently long sequence of packet transmissions, we seek a policy \( \pi^* \) that maximizes the average reward of the system with any initial state \( s_1 = i \). That is,

\[
\pi^* = \arg \max_{\pi} \mathbb{E} \left( \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} R(s_m, a_m) | s_1 = i \right), \quad (32)
\]

in which \( s_m (m = 1, 2, \ldots, M, s_m \in S) \) and \( a_m (m = 1, 2, \ldots, M, a_m \in A) \), are, respectively, the state of the \( m \)-th packet and the action assigned for the \( m \)-th packet, \( R(s_m, a_m) \), the reward function of the \( m \)-th packet when the state is \( s_m \) and the action is \( a_m \). From (27), we know that the average reward function \( \mathbb{E}(R(s_m, a_m)) \) is the probability that the packet is received \( \delta \)-on time after taking action \( a_m \) in state \( s_m \) so that we define the corresponding cost as

\[
C(s_m, a_m) = 1 - R(s_m, a_m), \quad (33)
\]

which is non-negative. Therefore, we can also find out the optimal packet scheduling policy \( \pi^* \) by minimizing the following average cost.

\[
V_\pi(i) = \min_{\pi} \mathbb{E} \left( \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} C(s_m, a_m) | s_1 = i \right) \quad (34)
\]
As shown in [27, Ch. 6.7, Theorem 6.17], (34) can be solved by the following functional equation,

\[ g + h(i) = \min_{a \in A} \left\{ C(s, a) + \sum_{j \in S} p_{ij}(a) h(j) \right\}, \quad (35) \]

in which \( g \) is a constant, \( h(i) \) is a bounded function, \( p_{ij}(a) \) is the state transition probability of the packet from state \( i \) to state \( j \) when action \( a \) is taken.

It is noted that, however, (35) is not a contraction mapping [27, Ch. 6.4, Theorem 6.10]. Thus, the searching process with (35) may not converge or converge very slowly. This motivates us to consider an alternative expected total \( \alpha \)-discounted cost given by

\[ V_{\pi}^\alpha(i) = \min_{\pi} \mathbb{E} \left( \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} \alpha^m C(s_m, a_m) | s_1 = i \right), \quad (36) \]

for all \( i \in S \), in which \( 0 < \alpha < 1 \) is a discounting factor. Moreover, the \( \alpha \)-optimal policy \( \pi^\alpha \) and the \( \alpha \)-optimal cost function \( V_{\pi}^\alpha(i) \) satisfies [27, Ch. 6.7, (24)],

\[ V_{\pi}^\alpha(i) = \min_{a \in A} \left\{ C(i, a) + \alpha \sum_{j \in S} p_{ij}(a) V_{\pi}^\alpha(j) \right\}. \quad (37) \]

Particularly, the following theorem shows that as \( \alpha \) approaches unity, \( \pi^\alpha \) would converge to \( \pi^* \).

**Theorem 2:** For some sequence \( \alpha_n \to 1 \), we have \( h(s) = \lim_{n \to \infty} V_{\pi^\alpha_n}(s) - V_{\pi^\alpha}(s_1) \), \( g = \lim_{\alpha \to 1} (1 - \alpha) V_{\pi^\alpha}(s_1) \), for any fixed reference state \( s_1 \). In particular, (34) and (36) share the same optimal policy.

**Proof:** Based on the state transition probabilities in (21), (22), (23), (24), and (25), it can be seen that each state can reach all other states directly or indirectly through some intermediate states, which means that the Markov chain is irreducible. According to [27, Ch. 6.8, Corollary 6.20], \( V_{\pi^\alpha_n}(s) - V_{\pi^\alpha}(s_1) \) would be uniformly bounded, and hence the conditions of [27, Ch. 6.7, Theorem 6.17] are satisfied, which yield the results in Theorem 2 immediately.

Theorem 2 shows that (32), (34), and (36) have the same optimal scheduling policy. Thus, the optimal scheduling policy of the system can be calculated by the matrix iteration method. To be specific, for each state \( i \), we shall calculate the expected costs \( C(i, a) + \alpha \sum_{j \in S} p_{ij}(a) V_{\pi^\alpha}(j) \) for each action \( a \in A \), including the random transmission, dropping the packet, delaying the packet for some slots \( 0 < \alpha < 1 \), or retransmit the packet for some times \( n_t = 1, 2, \ldots, n_t^\text{max} \). With the obtained expected costs, we can determine the best action and update the cost \( V_{\pi^\alpha}(i) \) with (37). In particular, it was proved in [27, Ch. 6.8] that the mapping given in (37) is a contract mapping. Thus, Algorithm 1 is guaranteed to converge quickly. We denote the vector of all the states as \( s \) and the cost vector as \( V_{\alpha} \). By applying (37) to \( s \) (which is done element by element) iteratively, the cost vector \( V_{\alpha} \) converges to the optimal cost vector while the obtained actions are all optimal for corresponding states, as shown in Algorithm 1.

### Algorithm 1: Solving the optimal packet scheduling policy.

1. **Input:**
   - cost matrix \( C_{DL} \) and transition probability matrix \( P_{DL} \) of the delaying strategy;
   - cost vector \( C_{DP} \) and transition probability matrix \( P_{DP} \) of the dropping strategy;
   - cost matrix \( C_{RP} \) and transition probability matrix \( P_{RP} \) of the repeating strategy;

2. **Initialization:**
   - Set iteration error to \( \Delta v = +\infty, \varepsilon = 10^{-3} \);
   - Initialize the cost function vector \( V_{\alpha} = \text{zeros}(|\alpha_{\text{max}} - \alpha_{\text{min}}| + 1, 1) \);
   - Initialize the policy vector \( \pi_{\alpha}^* = \text{zeros}(|\alpha_{\text{max}} - \alpha_{\text{min}}| + 1, 1) \);

3. **Iteration:**
   - while \( \Delta v > \varepsilon \), do
     - \( f_{DP} = C_{DP} + \alpha P_{DP} V_{\alpha} \);
     - for \( n_d = 0 \) to \( n_d^\text{max} \) do
       - \( F_{DL}(::, n_d + 1) = \text{zeros}(|\alpha_{\text{max}} - \alpha_{\text{min}}| + 1) \);
       - \( V_{\alpha} \);
     - end for
     - for \( n_r = 0 \) to \( n_r^\text{max} \) do
       - \( F_{RP}(::, n_r + 1) = \text{zeros}(|\alpha_{\text{max}} - \alpha_{\text{min}}| + 1) \);
       - \( V_{\alpha} \);
     - end for
     - \( V^\text{old} = V_{\alpha} \);
     - \( S = [f_{DP}, F_{DL}, F_{RP}] \);
     - \( [V_{\alpha}, \pi_{\alpha}^*] = \text{min}(S, 2) \), %find the minimum over the 2-nd dimension
   - \( \Delta v = \max(|V_{\alpha} - V^\text{old}|) \); 
   - end while

4. **Output:** \( V_{\alpha}, \pi_{\alpha}^* \).

Since the state of a packet is the difference between its transmission starting time and its target reception time, the state space \( S \) is often infinitely large. The probability for the state of packets to be very large or small, however, is very small and can be neglected. Thus, we shall limit the state space to the set of integers within the finite range \([\alpha_{\text{min}}, \alpha_{\text{max}}]\), so that we can solve the problem more efficiently. In this case, the number of desirable states is \( \alpha_{\text{max}} - \alpha_{\text{min}} + 1 \).

From (21), (22), (23), (24), and (25) we can explicitly express the transition matrices of delaying, dropping and repeating, which are denoted, respectively, as \( P_{DL}, P_{DP}, \) and \( P_{RP}. \) In particular, \( P_{DL} \) and \( P_{RP} \) are three-dimensional matrices. In \( P_{DL} \), the first and the second dimensions represent the states before and after the transition, while the third dimension represents the number of slots the packets are delayed, i.e., \( n_d \). Likewise, the third dimension of \( P_{RP} \) represents the maximum allowed number of retransmission, i.e., \( n_r \). From (29), (30), (31), and (33), we can also obtain the cost functions of the three strategies, i.e., \( C_{DL}, C_{DP}, \) and \( C_{RP}. \) Matrices \( C_{DL} \) and \( C_{RP} \) are two-dimensional matrices defining the costs for each state and each \( n_d \) and \( n_r \).

The complexity of Algorithm 1 is mainly determined by the input matrix \( P_{DL} \) of the delaying strategy and the matrix \( F_{RP} \).
of the repeating strategy in the initialization phase, as well as the computation of matrices \(\text{FDL} \) and \(\text{FDP} \) and \(\text{FRP} \) in the iteration phase. With the Big-O representation, the computational time complexity of matrices \(\text{P}_{\text{DL}}, \text{P}_{\text{DP}}, \text{P}_{\text{DL}} \) and \(\text{F}_{\text{RP}} \) can be expressed as \(O(n_{\text{d}}^{\max} + 1)(t_{\max} - t_{\min} + 1)^2) \), \(O(n_{\text{d}}^{\max} + 1)(t_{\max} - t_{\min} + 1)^2) \), \(O(f(\alpha)(n_{\text{d}}^{\max} + 1)) \) and \(O(f(\alpha)(n_{\text{d}}^{\max} + 1)) \), respectively, in which \(f(\alpha) \) is the number of computations required for the convergence of the cost vector \(V_{\alpha} \) and is decreasing with the increase of the discount factor \(\alpha \). Thus, the time complexity of Algorithm 1 can be expressed according to the rule of Big-O operations as \(O((n_{\text{d}}^{\max} + n_{\text{r}}^{\max}))[f(\alpha) + (t_{\max} - t_{\min})^2]) \).

Finally, \((t_{\max} - t_{\min} + 1) \times 1 \) vector of optimal packet scheduling policy \(\pi_{\alpha}^{*} \) can be obtained by Algorithm 1. As shown in Theorem 2, we have \(\pi_{\alpha}^{*} = \pi^{*} \), which specifies the actions for all the states. With the obtained optimal scheduling policy \(\pi_{\alpha}^{*} \), which can be expressed by a state-action mapping table, one can find out the optimal action (i.e., transmit it without control, delay it, drop it, or repeat it) of each packet based on its current state.

1) **Calculation of Expected Total Rewards:** In this section, we calculate the total expected reward (equal to the number of packets received on-time) of the system using a greedy algorithm. Note that the distribution of the final of the system is characterized by its initial state, the controlling strategy taken for each packet, and the probability transition matrices \(\text{P}_{\text{DL}}, \text{P}_{\text{DP}}, \text{P}_{\text{RP}} \) (see Algorithm 1 and (21) to (25)). Thus, we can calculate the expected reward of each possible transition path and find out the maximum among them. In particular, we can reduce the computational complexity by removing those paths that are not locally optimal. As shown in Section VI, the obtained total expected reward matches well with the MDP based Algorithm 1. Note also that the total expected reward of the system represents the number of packets received \(\delta\)-on-time.

a) **System with random transmissions:** For the system using the random transmission strategy, the expected reward of a single transition from state \(i \) can be expressed as

\[
R(i) = \sum_{j \in S} p_{ij} r_{ij}, \quad i \in S,
\]

in which \(p_{ij} \) (cf. (20)) and \(r_{ij} \) (cf. (26)) are, respectively, the probability and the reward of the transition from \(i \) to \(j \). We denote the vector of expected transition rewards of all the states as \(R = [R(t_{\min}), R(t_{\min} + 1), \ldots, R(t_{\max})]^T \), which is a constant vector.

We denote the expected total reward of a sequence of \(m \) transitions from state \(i \) as \(v'_{m}(i) \), which can be calculated based on \(v'_{m-1}(j) \) and \(r_{ij} \) as

\[
v'_{m}(i) = \sum_{j \in S} p_{ij} [r_{ij} + v'_{m-1}(j)]
\]

\[
= R(i) + \sum_{j \in S} p_{ij} v'_{m-1}(j), \quad i \in S.
\]

We denote \(V'_{m} = [v'_{m}(t_{\min}), v'_{m}(t_{\min} + 1), \ldots, v'_{m}(t_{\max})]^T \) as the vector of the expected \(m\)-transition rewards of all the states. It is clear that

\[
V'_{1} = R,
\]

\[
V'_{m} = R + PV'_{m-1}, \quad m = 2, 3, \ldots, M,
\]

in which \(P \) is the state transition probability matrix of the random transmission strategy (cf. (20)). Starting from (40), we use (41) repeatedly to obtain the expected total reward vector \(V'_{M} \) of a sequence of \(M - 1 \) state transitions (Algorithm 2). Moreover, \(V'_{M} \) is also the expected total reward of the system for transmitting \(M \) packets to the destination node.

**Remark 4:** Note that the transmission of the first packet starts from the first slot and the corresponding initial state is \(s_1 = T_{1\text{gt}} \).

When the transmission of all of \(M \) packets is completed, the expected total reward of the system would be \(v'_{M}(T_{1\text{gt}}) \), which can be obtained by Algorithm 2. Note also that the expected total reward of a system using random transmissions equals to the number \(\kappa_{M} \) (cf. (9)) of packets received \(\delta\)-on-time, which can be obtained through classical probability methods, as shown (9) and (10), in Section III. In particular, it can be verified through simulations that the \(v'_{M}(T_{1\text{gt}}) \) obtained by Algorithm 2 equals to \(\kappa_{M} \) exactly.

b) **System with scheduling:** We calculate the expected total reward of the system with packet scheduling iteratively, as shown in Algorithm 2.

We note that the transition probability matrices \(\text{P}_{\text{DL}}(:, \ldots, n_{d}) = [p_{ij}(n_{d})], \text{P}_{\text{DP}} = [p_{ij}], \text{P}_{\text{RP}}(:, \ldots, n_{r}) = [p_{ij}(n_{r})] \) of the delaying strategy, the dropping strategy, and the repeating strategy are given, respectively, by (21), (22), and (23) to (25) in Algorithm 2. In particular, it can be seen that the time complexity of the algorithm is mainly determined by that of matrices \(\text{P}_{\text{DL}}, \text{P}_{\text{DP}}, \text{P}_{\text{DL}} \) and \(\text{P}_{\text{RP}} \), which can be expressed as \(O((n_{d}^{\max} + 1)(t_{\max} - t_{\min} + 1)^2) \), \(O(n_{r}^{\max} + 1)(t_{\max} - t_{\min} + 1)^2) \), \(O(M(n_{d}^{\max} + 1)) \) and \(O(M(n_{r}^{\max} + 1)) \), respectively. Thus, the time complexity of Algorithm 2 can be obtained as \(O((n_{d}^{\max} + n_{r}^{\max})[M + (t_{\max} - t_{\min})^2]) \) according to the Big-O operation rule. For each state \(i \in S \) and each chosen strategy, therefore, the expected reward of the next transition \(R(i) \) can be calculated by (38), in which \(p_{ij} \) is replaced by \(p_{ij}(n_{d}), p_{ij}, \) and \(p_{ij}(n_{r}) \), respectively.

We denote the vector of expected \(m\)-transition rewards of the system with scheduling as \(V_{m} = [v_{m}(t_{\min}), v_{m}(t_{\min} + 1), \ldots, v_{m}(t_{\max})]^T \). Given the expected \(m\)-transition reward vector \(V_{m} \), we shall first estimate the expected \((m+1)\)-transition rewards for all the cases when the delaying (for each \(n_{d} = 0, 1, \ldots, n_{d}^{\max} \)), dropping, and the repeating (for each \(n_{r} = 0, 1, \ldots, n_{r}^{\max} \)) strategy are used. Specifically, we have

\[
\text{FDL}(:, n_{d} + 1) = \text{RD}_{\text{L}}(:, n_{d} + 1) + \text{R}_{\text{DL}}(:, n_{d} + 1) V_{m}
\]

\[
f_{\text{DP}} = R_{\text{DP}} + P_{\text{DP}} V_{m}
\]

\[
\text{FRP}(:, n_{r} + 1) = \text{R}_{\text{RP}}(:, n_{r} + 1) + \text{P}_{\text{RP}}(:, n_{r} + 1) V_{m},
\]
Algorithm 2: Total expected reward.

1: Input:
reward vector \( R \) and transition probability matrix \( P \) of random transmission;
reward vector \( R_{DP} \) and transition probability matrix \( P_{DP} \) of drop strategy;
reward matrix \( R_{DL} \) and transition probability matrix \( P_{DL} \) of delay strategy;
reward matrix \( R_{RP} \) and transition probability matrix \( P_{RP} \) of repeat strategy;
2: Initialization:
Initialize optimal scheduling policy vector \( V_0 = \text{zeros}(t_{\text{max}} - t_{\text{min}} + 1, 1) \);
Initialize random transmission vector \( V'_0 = \text{zeros}(t_{\text{max}} - t_{\text{min}} + 1, 1) \);
3: Iteration:
for \( m = 1 \) to \( M \) do
\[
V'_m = R + PV'_{m-1};
\]
\[
f_{DP} = R_{DP} + P_{DP}V_{m-1};
\]
for \( n_d = 0 \) to \( n_d^{\text{max}} \) do
\[
F_{DL}(i; n_d + 1) = R_{DL}(i; n_d + 1) + P_{DL}(i; ; n_d + 1)
\]
\[
V_{m-1};
\]
end for
for \( n_r = 0 \) to \( n_r^{\text{max}} \) do
\[
F_{RP}(i; n_r + 1) = R_{RP}(i; n_r + 1) + P_{RP}(i; ; n_r + 1)
\]
\[
V_{m-1};
\]
end for
\[
S = [f_{DP}, F_{DL}, F_{RP}];
\]
\[
V_m = \max(S, 2);
\]
end for
4: Output: \( V_M, V'_M \).

for \( n_d = 0, 1, \ldots, n_d^{\text{max}} \) and \( n_r = 0, 1, \ldots, n_r^{\text{max}} \). For each state \( i \), therefore, we have obtained the expected total reward for all controlling strategies (i.e., delaying, dropping, and repeating) and parameters (i.e., \( n_d \) and \( n_r \)). By searching the maximum reward among \( \{F_{DL}(i, 1), \ldots, F_{DL}(i, n_d^{\text{max}} + 1), f_{DP}(i), F_{RP}(i, 1), \ldots, F_{RP}(i, n_r^{\text{max}} + 1)\} \), we can then determine the optimal controlling action and parameter. With the obtained controlling strategy and parameter, we can further update the expected \((m + 1)\)-transition rewards \( V_{m+1} \) of system. As shown in Algorithm 2, this process continues until the controlling strategies of all the packets have been determined and the expected total reward of the system with scheduling is \( v_M(T_{\text{tgt}}) \).

VI. SIMULATION RESULTS

In this section, we investigate the on-time reception rate of a sequence of \( M \) packet transmissions over the Rayleigh fading channel. In particular, we transmit a sequence of \( M \) packets over the channel and schedule each packet with the optimal scheduling policy obtained by Algorithm 1. In Figs. 6–11, we then calculate the corresponding on-time reception rate (which is referred to as the simulation result) by counting the packets received \( \delta \)-on-time. We also calculate the on-time reception rate theoretically using Algorithm 2, which is referred to as the theoretical results.

The distribution of the channel power gain of the Rayleigh fading channel is given by

\[
f_\gamma(x) = \lambda e^{-\lambda x}.
\]  
(45)

We set the channel parameter as \( \lambda = 2 \), the transmit power of the source node as \( P_s = 1 \) W, the distance between the source and destination nodes as \( d = 100 \) m, the path loss exponent as \( \alpha = 2 \), and the channel noise as \( \sigma^2 = 10^{-4} \) W. For a given SNR threshold \( V_T \), the probability that the transmitted packet can be successfully decoded by the destination node would be \( p = \exp(-\lambda V_T d^\alpha \sigma^2) = \exp(-2V_T) \) (see (1)). Thus, we can adjust the probability of successful transmissions by changing the threshold \( V_T \), as shown in Fig. 6. For example, we have \( p = 0.2 \) if \( V_T = 0.8047 \). Without loss of generality, we consider a finite number of states and set the maximum and the minimum state as \( t_{\text{max}} = 500 \) and \( t_{\text{min}} = -500 \), respectively, i.e., \( s \in \{-500, -499, \ldots, 500\} \). In the simulation, we also limit the delay time and the number of retransmissions by \( n_d^{\text{max}} = 20 \) and \( n_r^{\text{max}} = 20 \). In the implementation of the MDP algorithm, we set the discount factor as \( \alpha = 0.999 \). In our Monte Carlo simulation of the optimal scheduling policy, we start a sequence transmissions from initial state \( s_1 = T_{\text{tgt}} \). The transmission time \( S \) of each packet is randomly generated according to a geometric distribution with probability \( p \). Based on the current state and the optimal transmission scheme obtained by Algorithm 1, the strategy adopted in the current state \( i \) can be determined (see Algorithm 1 Section V). After taking this action (strategy), the state of the system would transfer to the next state \( j \), also according to the optimal scheduling policy \( \pi^*_s \) obtained by Algorithm 1 (see Fig. 5). Specifically, if the random transmission is adopted, we have \( j = i - S + T_{\text{tgt}} \); if the delay strategy is adopted, then \( j = i - S - n_d + T_{\text{tgt}} \), in which \( n_d \) is the number of delayed slots; if the drop strategy is adopted, then
$S = 0$ and $j = i + T_{tgt}$; if the repeat strategy is taken, then the current state of packet is determined by whether it meets the condition of retransmission (i.e., whether it is received before the target reception range) and we have $j = i - y + T_{tgt}$ ($y$ is the total transmission time of the packet with the number of retransmission $n_i$). After checking the packet is received on-time according to (26), we can transmit the next packet. Finally, the number of packets received on-time is counted and the on-time reception rate of the system is obtained from (3). For the Monte Carlo simulation of the system with the random transmission, each packet will be transmitted without any controlling. Thus, the on-time reception rate might be lower than that of the system with optimal scheduling policy. In Fig. 7, we investigate the behavior of the on-time reception rate $\varrho_M$ (see (11)) as a function of the deviation tolerance $\delta$. The probability of successful transmission is set to $p = 0.2$, the number of packets is set to $M = 10,000$, and the target reception interval is set to $T_{tgt} = 5$ slots. We observe that under the optimal packet scheduling policy obtained by Algorithm 1, the on-time reception rates are much larger than that of the random transmission scheme. This shows that the proposed MDP-based packet scheduling is very effective. In case $\delta = 0$, the $\delta$-on-time requirement reduces to the strictly on-time. One can also observe that the corresponding on-time reception rates are relatively small, even though the optimal packet scheduling is used. In fact, due to the fading of the wireless channel, it is very difficult to alleviate the randomness of transmissions. Nevertheless, by using the optimal scheduling policy, the strictly on-time reception rate can be increased about 13%, which is much larger than that of random transmissions. Moreover, we see that our simulation results and theoretical results match well.

Fig. 8 presents how the on-time reception rate $\varrho_M$ changes with the target reception interval $T_{tgt}$. When $T_{tgt}$ increases, we have more freedom of scheduling, so that the on-time reception rate increases with $T_{tgt}$. For the random transmissions, we see that $\varrho_M$ changes differently and reaches its maximum at $T_{tgt} = 5$, which is exactly the expected value of the transmission time, i.e., $E(S) = 1/p = 5$ slots. This is in consistent with our intuitions that most of the transmission times fall into a finite range around their common expectation.

In Fig. 9, we plot the on-time reception rate $\varrho_M$ as a function of the successful reception probability $p$ (i.e., the reliability of the fading channel), where we set $T_{tgt} = 4$, $M = 10,000$, and $\delta = 2$. With the optimal scheduling policy, we see that $\varrho_M$ increases with $p$ and almost approaches the unity as $p$ reaches 0.5. Under the random transmission scheme, however, $\varrho_M$ does not change much as $p$ increases. This is because if $p$ increases, the expected reception time of a packet deviates the target reception time.

In Fig. 10, we present how the on-time reception rate (11), (15)) changes when the length $M$ of the packet sequence increases. For the random transmission strategy, we observe that the on-time reception rate decreases with $M$ and is expected to approach zero as $M$ goes to infinity. This is because the channel gains are random and difficult to predict while the accumulated deviation from the target times increases with $M$. For the transmission with optimal packet scheduling, we observe that the on-time reception rate (theoretical results obtained by Algorithm 2) is much larger and converges to a constant as $M$ goes to infinity. By optimally scheduling the packets, however, the gain in the on-time reception rate is also limited, since the randomness of the channel cannot be removed completely.
We denote the transmission time of the $pT$ packets are $\tau$. From (3), $\tau = \tau_0 + mT_{tgt}$, where $mT_{tgt}$ is relatively large. From $0.2, \leq \tau \leq mT_{tgt}$, we have $\tau_k$, which is because $\tau_0 = 0.2, mT_{tgt} = 33$ is large. From (2) and the total transmission time $\tau_0 = mT_{tgt}$ follows the negative binomial distribution with parameter $p$, we have

$$\Pr\{\tau_k = j\} = C_m^{m-1} p^m (1-p)^{j-m}, j = m, m+1, \ldots$$

(47)

To calculate $P(x_m)$, we consider the following two cases.

1) $mT_{tgt} \leq m + \delta$: Since $\tau_0 \geq 1$, we have $\sum_{k=1}^m \tau_k \geq m$, and $mT_{tgt} - \delta \leq m \leq \sum_{k=1}^m \tau_k$. Thus,

$$P(x_m) = \Pr\left\{\sum_{k=1}^m \tau_k \leq mT_{tgt} + \delta\right\}$$

$$= p^m + \cdots + C_m^{m-1} p^m (1-p)^{mT_{tgt}+\delta-m}$$

$$= \sum_{k=33}^m C_m^{m-1} p^m (1-p)^{k-m}.$$  

(48)

2) $mT_{tgt} > m + \delta$: In this case, we have

$$P(x_m) = \Pr\left\{\sum_{k=1}^m \tau_k \leq mT_{tgt} + \delta\right\}$$

$$- \Pr\left\{\sum_{k=1}^m \tau_k \leq mT_{tgt} - \delta - 1\right\}$$

$$= \sum_{k=mT_{tgt}} C_m^{m-1} p^m (1-p)^{k-m}.$$  

(49)

This completes the proof of Proposition 1.

**APPENDIX**

**A. Proof of Proposition 1**

Proof: We denote the transmission time of the $m$-th packet as $\tau_m$ and the probability that the $m$-th packet is received $\delta$-on time as $P(x_m)$. For the $m$-th packet, we have

$$P(x_m) = \Pr\left\{mT_{tgt} - \delta \leq \sum_{k=1}^m \tau_k \leq mT_{tgt} + \delta\right\}. \quad (46)$$

Since transmission time $\tau_k$ follows the geometric distribution (cf. (2)) and the total transmission time $\sum_{k=1}^m \tau_k$ follows the negative binomial distribution with parameter $p$, we have

$$\Pr\{\sum_{k=1}^m \tau_k = j\} = C_m^{m-1} p^m (1-p)^{j-m}, j = m, m+1, \ldots$$

(47)

To calculate $P(x_m)$, we consider the following two cases.

1) $mT_{tgt} \leq m + \delta$: Since $\tau_0 \geq 1$, we have $\sum_{k=1}^m \tau_k \geq m$, and $mT_{tgt} - \delta \leq m \leq \sum_{k=1}^m \tau_k$. Thus,

$$P(x_m) = \Pr\left\{\sum_{k=1}^m \tau_k \leq mT_{tgt} + \delta\right\}$$

$$= p^m + \cdots + C_m^{m-1} p^m (1-p)^{mT_{tgt}+\delta-m}$$

$$= \sum_{k=33}^m C_m^{m-1} p^m (1-p)^{k-m}.$$  

(48)

2) $mT_{tgt} > m + \delta$: In this case, we have

$$P(x_m) = \Pr\left\{\sum_{k=1}^m \tau_k \leq mT_{tgt} + \delta\right\}$$

$$- \Pr\left\{\sum_{k=1}^m \tau_k \leq mT_{tgt} - \delta - 1\right\}$$

$$= \sum_{k=mT_{tgt}} C_m^{m-1} p^m (1-p)^{k-m}.$$  

(49)

This completes the proof of Proposition 1.

**B. Proof of Theorem 1**

Proof: Let $P(x_k)$ and $P(\overline{x_k})$ be the probability for the $k$-th packet to be and to be not received $\delta$-on time, respectively. We denote the probability that $k$ packets out of the $M$ packets are received $\delta$-on time as $P(x_k^M)$. Under this setting, We prove the theorem by mathematical induction.
We start from $M = 1$ and readily see that $P(x_1^M) = P(x_1)$. We assume that Theorem 1 holds for $M = n$, i.e.,

$$
\sum_{k=1}^{M} kP(x_k^M) = \sum_{k=1}^{M} P(x_k).
$$

(50)

That is,

$$
nP(x_1, x_2, \ldots, x_n)
+ (n-1)[P(x_1 x_2 \ldots x_n) + \ldots + P(x_1 x_2 \ldots x_n x_{n-1})] +
+ (n-2)[P(x_1 x_2 x_3 \ldots x_n) + \ldots + P(x_1 x_2 x_3 \ldots x_n x_{n-2})] +
\ldots + (n-k)[P(x_1 x_2 \ldots x_n) + \ldots + P(x_1 x_2 \ldots x_n x_{n-1})] = P(x_1) + P(x_2) + \ldots + P(x_n).
$$

(51)

For $M = n + 1$, we then have

$$
\sum_{k=1}^{n+1} kP(x_k^{n+1}) = (n+1)P(x_1 x_2 \ldots x_n x_{n+1})
+ n[P(x_1 x_2 \ldots x_n x_{n+1}) + \ldots + P(x_1 x_2 \ldots x_n x_{n-1})]
+ (n-1)[P(x_1 x_2 x_3 \ldots x_n x_{n+1}) + \ldots + P(x_1 x_2 x_3 \ldots x_n x_{n-2})]
\ldots + (n-k)P(x_1 x_2 \ldots x_n x_{n+1}) + \ldots + P(x_1 x_2 \ldots x_n x_{n-1})

(52)

in which (c) follows from (51) and (b) is obtained by reorganizing the equation (a). For example, the first term of (52) can be calculated by

$$
(n+1)P(x_1 \ldots x_n x_{n+1}) + nP(x_1 \ldots x_n x_{n-1})
= nP(x_1 x_2 \ldots x_n) + P(x_1 x_2 \ldots x_n x_{n+1}).
$$

(53)

Thus, (10) holds true for $M = n + 1$, and thus holds for all $M \geq 1$ and the proof of Theorem 1 is completed.

**REFERENCES**

[1] E. Sisinni, A. Safiullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial Internet of Things: Challenges, opportunities, and directions,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[2] M. Wollschläger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0,” *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[3] X. Ge, “Ultra-reliable low-latency communications in autonomous vehicular networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5050–5016, May 2019.

[4] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, “Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects,” *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

[5] T. Huang, S. Wang, Y. Huang, Y. Zheng, J. Liu, and Y. Liu, “Survey of the deterministic network,” *J. Commun.*, vol. 40, no. 6, pp. 160–176, Jun. 2019.

[6] N. Finn, “Introduction to time-sensitive networking,” *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 22–28, Jun. 2018.

[7] M. Khashoeivisian, V. Joseph, P. Gupta, F. Meshkati, R. Prakash, and P. Tinnakornsrisuphap, “5G industrial networks with CoMP for URLLC and time sensitive network architecture,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 947–959, Apr. 2019.

[8] F. Zhao, A. Liu, and H. Zhou, “Applications and transmission technology of 5G deterministic networks,” *ZTE Technol. J.*, vol. 25, no. 5, pp. 62–67, Oct. 2019.

[9] S. S. Craciun et al., “Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks,” in *Proc. IEEE 24th Int. Conf. Real-Time Netw. Syst.*, 2016, pp. 183–192.

[10] M. K. Al-Hares, P. Assimakopoulos, D. Muench, and N. J. Gomes, “Traditional queuing regimes and time-aware shaping performance comparison in an Ethernet fronthaul network,” in *Proc. IEEE Int. Conf. Transp. Opt. Netw.*, 2017, pp. 1–4.

[11] J. Specht and S. Samii, “Emergency-based scheduler for time-sensitive switched Ethernet networks,” in *Proc. IEEE 28th Euromicro Conf. Real-Time Syst.*, 2016, vol. 1, pp. 75–85.

[12] Q. Yu, H. Wan, X. Zhao, Y. Gao, and M. Gu, “Online scheduling for dynamic VM migration in multicast time-sensitive networks,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3778–3788, Jun. 2020.

[13] M. Vik, Z. Hanzalek, K. Brejchova, S. Tang, S. Bhattacharjee, and S. Fu, “Enhancing schedulability and throughput of time-triggered traffic in IEEE 802.1Qbv time-sensitive networks,” *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 7023–7038, Nov. 2020.

[14] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, “Routing and scheduling of time-triggered traffic in time-sensitive networks,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4525–4534, Jul. 2020.

[15] E. Schweissguth et al., “ILP-based joint routing and scheduling for time-triggered networks,” in *Proc. 25th Int. Conf. Real-Time Netw. Syst.*, 2017, pp. 8–17.

[16] W. Huang et al., “The convergence of 5G deterministic networking and industrial Internet white paper,” 2020. [Online]. Available: https://www-file.huawei.com/-/media/corporate/pdf/news/5gdn-based-industrial-internet-white-paper.pdf

[17] D. Cavalcanti, J. P. Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, “Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems,” *Proc. IEEE*, vol. 107, no. 6, pp. 1132–1152, Jun. 2019.

[18] J. C. R. Bennett, K. Benson, A. Charny, W. F. Courtney, and J.-Y. Le Boudec, “Delay jitter bounds and packet scale rate guarantee for expedited forwarding,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 529–540, Aug. 2002.

[19] K. Hamadnam, A. Moubuyad, A. Shami, and S. Primak, “Analytical approximation of packet delay jitter in simple queues,” *IEEE Wireless Commun. Lett.*, vol. 5, no. 6, pp. 564–567, Dec. 2006.

[20] C. A. Fulton and S.-Q. Li, “Delay jitter first-order and second-order statistical functions of general traffic on high-speed multimedia networks,” *IEEE/ACM Trans. Netw.*, vol. 6, no. 2, pp. 150–163, Apr. 1998.

[21] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in wireless networks with throughput constraints,” *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6419–6428, Sep. 2018.

[22] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, “On the age of information with packet deadlines,” *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6419–6428, Sep. 2018.

[23] I. Kadota, A. Sinha, and E. Modiano, “Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints,” *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1359–1372, Aug. 2019.

[24] H. Tang, J. Wang, L. Song, and J. Song, “Minimizing age of information with power constraints: Multi-user opportunistic scheduling in multi-state time-sharing channels,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 854–868, May 2020.

[25] L. H. Ozawar, S. Shamai, and A. D. Wyner, “Information theoretic considerations for cellular mobile radio,” *IEEE Trans. Veh. Technol.*, vol. 43, no. 2, pp. 359–378, May 1994.
Yan Li received the B.S. degree in electronic information science and technology from Linyi University, Linyi, China, in 2020, and the M.S. degree in electronic information from the Nanjing University of Information Science and Technology, Nanjing, China, in 2023. His research interests include on-time communications for wireless networks and deterministic networking techniques.

Yunquan Dong (Member, IEEE) received the B.S. degree in electronic and information engineering from Qingdao University, Qingdao, China, in 2005, the M.S. degree in communication and information systems from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008, and the Ph.D. degree in communication and information engineering from Tsinghua University, Beijing, in 2014. From 2015 to 2016, he was a BK Assistant Professor with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea. He is currently a Professor with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include performance evaluations and performance optimizations of wireless networks, with recent focus on the age of information and ubiquitous sensing.

Jian Wang (Member, IEEE) received the B.S. and M.S. degrees from the Harbin Institute of Technology, Harbin, China, in 2006 and 2009, respectively, and the Ph.D. degree from Korea University, Seoul, South Korea, in 2013. After the graduation, he was a Postdoc and Assistant Research Professor with Rutgers University, New Brunswick--Piscataway, NJ, USA, Duke University, Durham, NC, USA, and Seoul National University, Seoul, South Korea. Since 2017, he has been with the School of Data Science, Fudan University, Shanghai, China, where he is currently an Associate Professor. His research interests include sparse and low-rank recovery, phase retrieval, oblique projection, lattice, signal processing in wireless communications, and statistical learning. He was nominated for the 2016 Young Author Best Paper Award in the IEEE Signal Processing Society.

Byonghyo Shim (Senior Member, IEEE) received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University (SNU), Seoul, South Korea, in 1995 and 1997, respectively, and the second M.S. degree in mathematics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana Champaign, Champaign, IL, USA, in 2004 and 2005, respectively. From 1997 and 2000, he was an Officer (First Lieutenant) and Academic Full-time Instructor with the Department of Electronics Engineering, Korean Air Force Academy. From 2005 to 2007, he was a Staff Engineer with Qualcomm Inc., San Diego, CA, USA. From 2007 to 2014, he was an Associate Professor with the School of Information and Communication, Korea University, Seoul, South Korea. Since 2014, he has been with the SNU, where he is currently a Professor with the Department of Electrical and Computer Engineering. His research interests include wireless communications, statistical signal processing for Big Data and machine learning, compressed sensing and matrix completion, and information theory. He was the recipient of the M. E. Van Valkenburg Research Award from the ECE Department, University of Illinois in 2005, Hadong Young Engineer Award from IEIE in 2010, Irwin Jacobs Award from Qualcomm and KICS in 2016, Shinyang Research Award from the Engineering College of SNU in 2017, Okawa Foundation Research Award in 2020, and IEEE COMSOC AP Outstanding Paper Award in 2021. He was a Technical Committee Member of Signal Processing for Communications and Networking (SPCOM), and is currently an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE WIRELESS COMMUNICATIONS LETTERS, Journal of Communications and Networks, and the Guest Editor of IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS (location awareness for radios and networks).