Aalborg Universitet

AALBORG UNIVERSITY

A Perspective on Time Toward Wireless 6G

Popovski, Petar; Chiariotti, Federico; Huang, Kaibin; Kalør, Anders E.; Kountouris, Marios; Pappas, Nikolaos; Soret, Beatriz

Link to publication from Aalborg University

A Perspective on Time Toward Wireless 6G

This article provides a systematic treatment of various timing measures in wireless communication, setting the basis for design and optimization for the next-generation real-time systems.

By Petar Popovski [ID], *Fellow IEEE*, Federico Chiariotti [ID], *Member IEEE*, Kaibin Huang [ID], *Fellow IEEE*, Anders E. Kalør [ID], *Graduate Student Member IEEE*, Marios Kountouris [ID], *Senior Member IEEE*, Nikolaos Pappas [ID], *Senior Member IEEE*, and Beatriz Soret [ID], *Member IEEE*

ABSTRACT | With the advent of 5G technology, the notion of *latency* got a prominent role in wireless connectivity, serving as a proxy term for addressing the requirements for real-time communication. As wireless systems evolve toward 6G, the ambition to immerse the digital into physical reality will increase. Besides making the real-time requirements more stringent, this immersion will bring the notions of time, simultaneity, presence, and causality to a new level of complexity. A growing body of research points out that latency is insufficient to parameterize all real-time requirements. Notably, one such requirement that received significant attention is information freshness, defined through the Age of Information (AoI) and its derivatives. In general, the metrics derived from a conventional black-box approach to communication network design are not representative of new distributed paradigms, such as sensing, learning, or distributed consensus. The objective of this article is to investigate the general notion of timing in wireless communication systems and networks, and its relation to effective information generation, processing, transmission, and reconstruction at the senders and receivers. We establish a general statistical framework of *timing* requirements in wireless communication systems, which subsumes both latency and AoI. The framework is made by associating a timing component with the two basic statistical operations: decision and estimation. We first use the framework to present a representative sample of the existing works that deal with timing in wireless communication. Next, it is shown how the framework can be used with different communication models of increasing complexity, starting from the basic Shannon one-way communication model and arriving at communication models for consensus, distributed learning, and inference. Overall, this article fills an important gap in the literature by providing a systematic treatment of various timing measures in wireless communication and sets the basis for design and optimization for the next-generation real-time systems.

KEYWORDS | 6G mobile communication; cellular technology; low latency communication; real-time systems.

Petar Popovski, **Federico Chiariotti**, and **Anders E. Kalør** are with the Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark (e-mail: petarp@es.aau.dk; fchi@es.aau.dk; aek@es.aau.dk).

Kaibin Huang is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, SAR, China (e-mail: huangkb@eee.hku.hk).

Marios Kountouris is with the Communication Systems Department, EURECOM, 06410 Biot, France (e-mail: marios.kountouris@eurecom.fr).

Nikolaos Pappas is with the Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden (e-mail: nikolaos.pappas@liu.se).

Beatriz Soret is with the Telecommunication Research Institute (TELMA), Universidad de Málaga, 29071 Málaga, Spain (e-mail: bsoret@ic.uma.es).

I. INTRODUCTION

How soon is *now*? When do two events occur simultaneously? These seemingly naïve questions have led to fundamental shifts in physics through the theory of relativity and irrevocably altered our notion of time. Besides the physical time, in a system with various interacting components, what matters is the *perception of time*. This is succinctly illustrated by the following excerpt from the novel "Recursion" by Crouch [1]:

> "Just what your brain does to interpret a simple stimulus like that is incredible. The visual and auditory information arrive at your eyes and ears at different speeds, and then are processed by

your brain at different speeds. Your brain waits for the slowest bit of stimulus to be processed, then reorders the neural inputs correctly, and lets you experience them together, as a simultaneous event—about half a second after what actually happened. We think we're perceiving the world directly and immediately, but everything we experience is this carefully edited, tape-delayed reconstruction."

The notions, the perception of time, simultaneity, synchronicity, and causality, get to a new level of complexity as wireless communication offers remote interaction among humans and machines at overextended distances. Indeed, wireless communication technology is radically transforming the very nature of human interactions, having a profound impact across our society and economy. Various names are used, such as tactile Internet or Internet of Senses [2], to denote the trend in which wireless connectivity augments human capabilities beyond their natural domain, enabling operation and interaction with objects and subjects placed within an extended space–time domain. We are at the dawn of the era of connected intelligence and autonomous automation, in which a myriad of interconnected sensor-empowered devices with computing, learning, and decision-making capabilities will underpin the global functioning of our societies, enabling formidable progress in the industrial, health, transportation, environmental, and educational sectors.

Naturally, these interactive applications need to perform a series of actions to work, all of which require some time: these include both the actual communication of the necessary data and the computation at both ends, e.g., to compress the raw sensor data into a more compact version at one end, then decode it, and present it to the user at the other end. These different components make up a latency budget [3], [4], which must satisfy strict requirements to maintain the real-time illusion. Just as humans collect and process stimuli in the brain, the processor of a device or robot gathers data from its sensors (including communication from other devices) and uses algorithms to create a unified estimate of the environment and translate data from the physical world into the digital domain [5]. In a more general sense, we can talk about interactivity and a "real-time illusion" not just for humans but also for machines: the perception of sensors and the granularity in time of control algorithms and actuators depend on the limitations of the hardware and software, and any timing difference shorter than this granularity will be unnoticeable to them. Naturally, while the latency budget in human communications has a hard floor given to the limits of perception, the latency budget for machines will depend on the specific device and its capabilities, as well as on the application for which it is used.

In order to keep the perception of time close to how "now" is commonly defined for all potential real-time applications, latency has been heralded as one of the main features of the widely publicized 5G wireless systems,
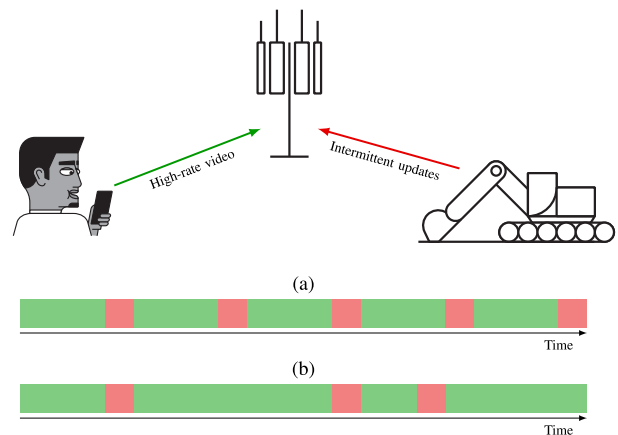

Fig. 1. *Simple scenario in which two users of different types transmit in the uplink to a common BS. One is a high rate user, and the other user sends intermittent critical updates. (a) Scheme with low-latency reservations for the intermittent user. (b) Scheme with pull-based updates from the intermittent user.*

as well as of the wireless systems beyond 5G. One of the three generic 5G services is ultrareliable low-latency communication (URLLC) [6], [7], where the ambition is to guarantee with very high reliability (e.g., >99.999%) that a given data packet will be delivered within a very short time frame (e.g., in the order of 1 ms). In a sense, URLLC aims to satisfy the least common denominator in terms of latency for all potential applications that exist or will emerge in the future. The upshot is that a wireless URLLC link cuts a small, predictable part of the latency budget in the overall digital service or application. Thus, if the end application has a more relaxed latency budget, then URLLC creates higher flexibility in designing the other system modules, such as compression or computation. Most of the research on low-latency communications in 6G is following the same pattern, putting even stricter constraints on latency and reliability, and foreseeing the use of new technologies, such as terahertz communications and intelligent surfaces, to meet them [8]. While 6G will be the first generation of mobile networks to natively rely on learning optimization at all layers [9] for resource allocation, parameter optimization, and network orchestration [10], most proposed solutions take the existing classes of traffic for granted [11], explicitly using the requirements of URLLC as their main objectives. The requirements for new use cases and applications, such as virtual reality (VR) or tactile communications [12], are also formulated in terms of the existing classes of traffic, with the same broad objectives as 5G.

Nevertheless, a growing body of emerging scenarios and research results points out that the maximalistic approach of URLLC to time-sensitive applications and services is limited. To illustrate this claim, consider the simple scenario in Fig. 1, where two different users share an uplink channel to a common base station (BS). One of the users requires a high rate, while the other sends intermittent

critical updates as a part of a networked control system. In a practical setting, the same BS may not be the communication endpoint for both users, so, here, we assume that the BS has an edge server for the low-latency user, while the high-rate user may have a different endpoint, and the timing aspects for this user are irrelevant in this discussion. Yet, the model is sufficient to show the interactions between the two services over the access resources. Fig. 1(a) shows the case that reflects a conservative URLLC approach, in which low-latency transmissions have reserved slots that guarantee maximal latency of at most four time slots. As mentioned above, this can make the system less efficient when the intermittent user has no updates in certain slots since those slots remain unused by the high rate user. Fig. 1(b) shows the same setting but with the assumption that the transmission of the updates from the intermittent user is based on a *pull-based communication model* [13]. To elaborate, the BS features a predictive controller that can estimate in advance when it will need the next update based on the freshness of the current update and the state of the system. Naturally, this prediction will also depend on the variability and the temporal evolution of the physical phenomenon observed by the monitoring control device. This simple example illustrates how *predictability* can improve system efficiency while satisfying the timing objectives without strict reservations defined by the minimal latency.

Based on this example, one can extrapolate more general conclusions about the insufficiency of the latency-only focused design of URLLC.

1) In a *quantitative* sense, aiming always for the least common denominator is inefficient due to the fact that low latency requires resource allocation that may lead to overprovisioning, thereby preventing other services from using the communication resources.

2) In a *qualitative* sense, there are multiple clear indications that the timing relations in a communication system cannot be condensed solely in the measure of communication latency. This is best illustrated by the emergence of alternative measures of timing, such as Age of Information (AoI) [14], [15], which aims to quantify the freshness of the data updates coming from, e.g., a Internet of Things (IoT) sensor. However, these are only instances of a general measure of timing in a distributed system with communication links. For instance, latency is usually measured with respect to a fixed point in time and space at which a data packet has been created, while AoI is measured with respect to the physical state of a certain cyber–physical system (CPS) or the occurrence of an event. In a distributed system of interconnected nodes, there could be other, more complex notions of timing or latency related to, for example, consensus or a distributed decision process.

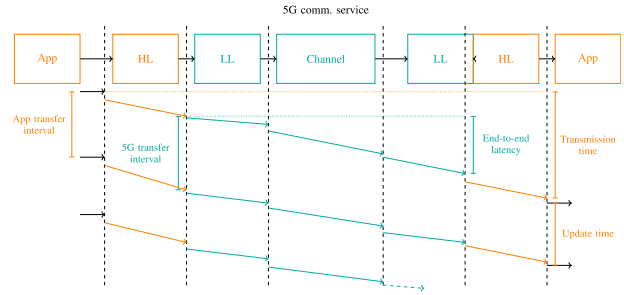Fig. 2 is adapted from a recent 3rd Generation Partnership Project (3GPP) technical document [16]. It shows that

Fig. 2. *General timing model from 3GPP [16].*

the standardization of mobile networks is also considering a wider approach to timing issues rather than simply setting extremely strict constraints on the wireless segment. Considering the timing of the application itself, as well as the higher communication layers, can lead to better resource efficiency, and considering different metrics, such as AoI, can increase the flexibility of the system with respect to the needs of different applications.

The objective of this article is to investigate the general notion of timing in wireless communication systems and networks, and its relation to effective information generation, processing, transmission, and reconstruction at the senders and receivers. We provide a systematic introduction to different timing measures through the way these measures interact with the layers in increasingly complex communication models. In this emerging heterogeneous, and often distributed, networking ecosystem, a general definition of the optimal communication system can be the one that chooses or generates the right piece of information that has to be efficiently transmitted at the right time instant, typically to achieve specific goals. Although the timing measures can be applied to general communication systems and networks, as also illustrated by treating the problems of consensus and distributed learning, the discussion in this article is biased toward the wireless access part. In practical systems, the latency of the core network represents a significant portion of the timing budget. However, the framework introduced in this article can accommodate the latency of the core network, as exemplified in the model with cascaded modules in Figs. 2 and 9.

We argue that the definition of *the right time instant* is not universal, and conventional approaches and metrics do not satisfy the requirements of many current and future applications and communication networks. From this perspective, the fundamental problem of communication becomes that of reconstructing the information generated at a source space–time point in a way that is sufficiently accurate for achieving a specific goal in a timely and effective manner at another, target space–time point. Furthermore, in specific scenarios, such as distributed learning, the communication system includes the postprocessing of the received information in order to achieve a certain goal.

A. Contributions and Article Organization

The main contributions of this article can be summarized in the following list.

1) Establish the context for defining timing measures in communication systems by considering the impact of various factors, such as the actors in the given communication scenario or the considered timing scales.

2) Provide a comprehensive view and systematization of the timing measures used in the research literature and in standardization.

3) Define a statistical framework for timing that is sufficiently general to encompass possibly all timing measures discussed in the literature.

4) Show how the statistical framework for timing can be applied in the context of different communication models and how a given timing measure can be optimized within a given communication model.

This article is organized as follows. First, we introduce the context for timing measures and describe the main use cases and communication actors in Section II. Our statistical framework, based on the concept of different timing references, is given in Section III. In Section IV, the framework is used to describe the current state of the art on timing, including latency, deadlines, and AoI. We then present in Section V the use of our framework in single-connection communication models, including Shannon's one-way communication model, two-way links, and connection through a cascade of system modules. The framework is extended to more general networking models for control, consensus, learning, and inference in Section VI. Finally, we conclude this article in Section VII.

II. CONTEXT FOR DEFINING TIMING MEASURES

In this section, we classify the notions of timing along multiple dimensions that depend on the context of the application or the communication service.

A. Time, Real-Time, and Simultaneity

In its everyday use, the term *real-time* is mostly associated with the sensation or perception of seeing things happen "instantly" or "simultaneously." For example, a sensor measuring our home temperature gives us the "feeling" of monitoring what is happening at home in real-time. The same applies to online chat applications, where we feel like talking in "real-time" with the other person. So what does *real-time* really mean, and more importantly, can we give a universal definition? The above description provides some sort of vague or general yet operational definition. However, any attempt to formalize it into a universal definition would, if possible, pass through a universal or absolute definition of simultaneity and time perception. Before attempting to give our definition of real-time, we first discuss several misconceptions associated with this term.

Real-time is often used interchangeably with the term "live." However, real-time and live are not the same. Think of an event (signal) transmitted live from Mars using an electromagnetic wave. On Earth, we are going to see that event with a minimum delay of around 4 min and a maximum delay of around 21 min, depending on the actual distance between Earth and Mars. Real-time is often associated with highly stringent latency requirements and extreme performance. Suppose that a robot is required to move 100 m under a "real-time" constraint of 50 s. This says nothing about the speed of the robot, as long as the travel duration remains below 50 s. This is because real-time is associated with a deadline, which is not necessarily stringent. In a refined definition of the term, real-time communication simply means that information or data (a message/packet or a set of messages) have to be transmitted and received on time, within a certain interval, not earlier, not later.

Timing is related to communication latency, whose operational definition is the time required for a packet to arrive from its sender to the destination. Measuring this time difference implies the use of a common reference and clock synchronization, which is often not available in practice. On top of that, real-time brings the notion of deadline and predetermined time instants into the picture. Therefore, a proper definition of timing requires an understanding of two important concepts: synchronicity and simultaneity, that is, the relation between two events is assumed to be happening at the same time in a given reference frame. The former is relatively well understood in communication systems and is often taken for granted. The latter is rather unexplored in wireless networking, and its relativity could bring new and interesting concepts and insights. Moreover, these two concepts bring up the theme of causality and space–time contiguity. Time at a particular location is defined by the measurement of a clock located in the immediate vicinity and is related to a certain reference frame. Every event that is spatially infinitely close to the clock can be assigned a time coordinate. Only the times of events occurring in the immediate vicinity of the clock can be ascertained directly by means of the clock. This means that, at this moment, one only has a notion of time the vicinity of the chosen clock, which is one of the main observations in the Special Theory of Relativity [17].

Fig. 3(a) illustrates the classic Einstein's example of the relativity of simultaneity. The static observer is at an equal distance from points A and B. Lightning strikes each of those points, and the static observer claims that the two strikes have occurred simultaneously. The moving observer sits on a fast train that moves toward B, and she claims that lighting has first struck B, and then, another bolt struck A. The tacit assumption made by A. Einstein is that difference in the observations is solely due to the physical propagation of the optical signals that carry information about the lightning bolts. This means that both observers have identical instruments for registering the lightning, and there is no difference in their observation due to,
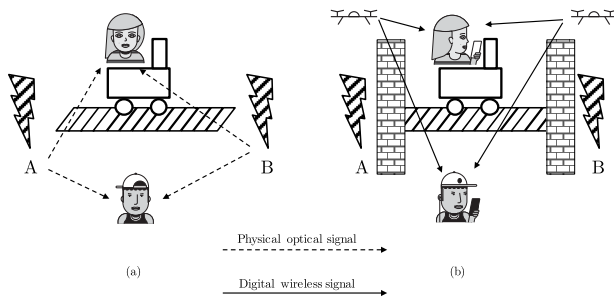
Fig. 3. *Relativity of simultaneity. (a) Communication through a propagation of the physical optical signals. (b) Communication through a wireless digital system.*

e.g., variations of the processing done in the measurement devices.

In Fig. 3(b), the setup is changed. The spatial points A and B are shielded by tall walls such that no visual information can arrive to the two observers. However, at each wall, there is a drone that captures a video of the respective lighting and transmits the video through wireless connections to both observers. The digital receiver of each observer uses a certain playout delay to make the events video screen seem as if they occurred simultaneously. Now, both observers agree that the two events have occurred simultaneously, which is a digital distortion of physical reality.

This parallel with the Special Theory of Relativity indicates that simultaneity and causality, as well as its bidirectional relation with time, are essential to defining timing. Drawing well-thought and operational analogies between timing in communication systems and time in physical systems (relativistic physics) and biological systems (horizon of simultaneity) could radically transform the notion of timing and synchronicity in future communication systems. This shift in thinking may lead to the development of a more general mathematical theory of timing in communications, one of the most difficult and important challenges remaining in communication theory.

B. Timing Scales and Requirements

Timing requirements, expressed as, e.g., latency or jitter, have traditionally been part of the set of Quality of Service (QoS) parameters defined for a given communication system, especially for applications tagged as real-time. However, as we discussed above, latency requirements and real-time constraints are highly dependent on the application, and different standards define different timing requirements. For example, the aim of 5G is to provide URLLC service for small data payloads (e.g., 32 B) with a maximum radio latency of 1 ms (i.e., the latency is measured at layer 2 or 3) and a reliability of higher than 99.999%. As wireless systems evolve beyond 5G toward a loosely defined set of technologies denoted as "6G," there

is a general tendency toward supporting lower latency and operating at shorter, ms or sub-ms timing scales [18].

In order to define the relevant timing scale, we can follow the categorization used by the Open Radio Access Network Alliance (O-RAN) [19], which defines three time-scale categories (see Fig. 4): 1) real-time; 2) near real-time; and 3) non-real-time. A similar classification is provided by the 5G Alliance for Connected Industry and Automation (5G-ACIA) [20], where the three categories are: 1) hard real-time; 2) soft real-time; and 3) non-real-time. Here, we provide a slightly more general view of these timing categories.

1) *Real-time:* A universal definition of "real-time" is elusive, not to mention that it is often associated with speed and the notions of "live" and "interactive." Real-time does not necessarily mean that information can be exchanged instantly or with negligible latency. Although it may entail ultrafast response time or immediate actions, its foundational element is that of completion in a predetermined, guaranteed amount of time. As such, real-time means controlled rather than zero latency. Real-time comes along with latency "determinism" and behavior predictability, which enables guarantees of achieving specific deadlines, being more or less stringent. For example, in the context of O-RAN, real-time denotes the processes (MAC scheduler or power control) for which the latency/timing measure is below 10 ms, while, in the context of the 5G-ACIA requirements, hard real-time deals with timing on the order of ms or even μs.

2) *Near real-time:* This is also denoted as soft real-time, where the term "soft" denotes a relaxation in both the absolute timing horizon, allowing for longer latencies, and the level of determinism in the timing requirements, allowing for softened probabilistic guarantees. In terms of timing horizon, near real-time in O-RAN deals with timings between 10 ms and 1 s, while soft real-time in 5G-ACIA can allow latencies in the order of a second. For instance, in O-RAN, near real-time may involve mobility or interference management. The real-time versus near real-time dichotomy can be interpreted as an effect of the cost of delayed action: if delaying an action is costly, the system should provide stricter guarantees, leading to harder real-time requirements. As near real-time backs away from almost deterministic latency guarantees, it also encompasses applications that are sensitive to the freshness of the data and AoI.

3) *Non-real-time:* This refers to the case in which timing parameters are such that no latency or deadline guarantees can be provided. In both the O-RAN and the 5G-ACIA definition, non-real-time refers to timings longer than a second. Non-real-time is associated with applications and procedures that are not time-sensitive and are denoted as best-effort [21] or delay-tolerant [22].
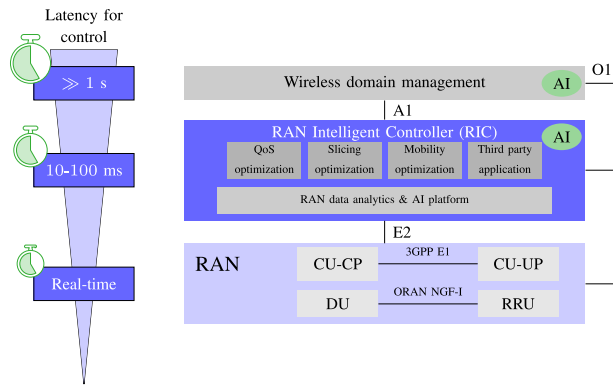
Fig. 4. *Timing scales of the RAN intelligent controller in O-RAN [19].*

Interestingly, the distinction between the above categories or the boundaries could be seen under the prism of effectiveness in achieving a specific goal. Timing requirements are usually imposed by services and may differ depending on the end user's perception or tolerance. Discrete automation and motion control may need end-to-end latencies of 1–5 ms, whereas process automation (remote control and monitoring) could operate with 50-ms latency. Specifically, according to the ITU Network 2030 [23], the upper bound on the end-to-end networking latency for haptic applications is on the order of 5 ms or less. This allows for round-trip control loops that allow feedback-based haptic applications to operate under 10 ms, even as low as 1 ms in some cases. Autonomous mission-critical infrastructure relies on similar latency objectives. Industrial and robotic automation requires not only "not-to-exceed" latency but an effectively "deterministic" latency, requiring predictability. This goes beyond in-time delivery; packets should be delivered "on time," i.e., not exceeding a certain latency but not arriving any sooner [24]. Industrial automation systems (Industry X.0) are based on real-time enabled CPSs, which will serve as platforms connecting people, objects, and systems to each other. Latency requirements for different applications range from several ms for mechanics to several ms down to 1 ms for machine-to-machine (M2M) and to 1 ms for electrics [25]. In vehicle-to-vehicle (V2V) networks, the time needed for collision avoidance in safety applications is below 10 ms [2]. In case a bidirectional data exchange for autonomous driving maneuvers is considered, latency on the order of 1 ms is most likely needed. In vehicle-to-everything (V2X), messages for situational awareness, e.g., cooperative awareness message (CAM) and basic safety message (BSM), are generated periodically (commonly every 100 ms), including vehicle state information, such as geolocation, velocity, heading, and other related information. In e-healthcare applications, an end-to-end latency of a few milliseconds, together with ultrahigh reliability in wireless link connection and data transmission, is required. In online gaming, latency around 20–100 ms could still

provide a satisfactory gameplay experience although lower latency is needed for maximum performance in games where timing is important. The latency requirement of holographic communications is on the order of 10 ms to allow instant viewer position adaptation at 60 frames per second (FPS). However, the latency requirement can be relaxed, becoming as low as conventional interactive video (on the order of 100 ms).

An example of timing-oriented networking design is time-sensitive networking (TSN), poised to connect and transform today's factories [26]. The TSN refers to a group of networking protocols and standards developed by the IEEE 802.1 TSN working group to provide accurate time synchronization, hard real-time constraints, and zero congestion loss in local area networks (LANs). The TSN handles three main functions: synchronizing all the clocks in the network, scheduling the most important traffic, and "shaping" the remaining traffic to achieve the desired traffic patterns. Taking TSN standards, which have been developed mainly assuming Ethernet as the underlying communication medium, the 3GPP has made significant progress in the last releases to complete the integration with 5G [27]. A limitation of TSN is that deterministic service is provided over a short distance. Moreover, TSN is geared toward constant bit rate (CBR) traffic, not variable bit rate (VBR) traffic.

C. Timing and Communication Actors

Through the description of the timing scales and requirements, it becomes apparent that communication actors represent an important factor that determines the perception of timing in a communication system. For example, real-time for machines that have sub-ms reaction times [28] has a different meaning than real-time for systems with a human in the loop, where latency longer than 50 ms could be tolerated. Then, is there a universal or optimal value for latency and reaction time? The answer depends on the context and the communication actors (humans or machines). Note that the term "machine" should be understood in a broader sense, beyond that of a simple man-made, electromechanical device. As such, a program or software application can also be treated as a machine in this context.

Depending on the actors and the communication parties involved, we can have the following first-order classification.

1) *Human–human:* In scenarios where humans communicate and interact with other humans, the timing and reaction time limits depend on the characteristics and the limitations of human senses, as well as on humans' capabilities in terms of sensory perception, cognition, and physical and neural transmission and processing times. For example, the neural processing time differs between the senses, and it is typically slower for visual stimuli than for auditory ones; approximately 50 and 10 ms, respectively. For touch,

the brain may have to take into account where the stimulation originated, e.g., toes and nose, as traveling time to the brain is not the same. Our brain can only process an image if our eye sees it for at least 13 ms [29], which corresponds to about 75 FPS, and receiving a stream of data faster than this will only underscore the limits of our perception. Accordingly, the definition of *real-time* for human communications has a hard limit given by perception: after video communications reach the perception threshold and achieve a reliable 10-ms latency at 75 FPS, any further improvement in the communication system will not provide any benefits to the user in terms of Quality of Experience (QoE). Providing exact values on this matter goes beyond the scope of this article and is an ongoing research topic. Nevertheless, an intriguing and surprising aspect is that, despite naturally occurring time lags and asynchronous arrivals of auditory and visual information, humans perceive intersensory synchronicity for most multisensory events in the external world and not only for those within the so-called "horizon of simultaneity," i.e., a distance of approximately 10–15 m from the observer [30].

2) *Human–machine:* This scenario entails communication and interaction between humans and machines. Machines are expected to be "faster" than humans, which will then define the timing requirements, as the human perceptual system is the bottleneck of the system. An interesting aspect here is how time is perceived by humans when they are interacting with a machine. Various studies on human–machine interaction, starting from R. B. Miller's seminal work in 1968 [31], have shown that the average human reaction time is on the order of 250 ms. Moreover, humans perceive a response time of 100 ms as instantaneous, whereas uninterrupted flow[1] is experienced with a 1-s response time.

3) *Machine–machine:* In this setting of increasing importance, machines are interacting with each other without the possibility of human intervention, and M2M traffic is becoming an important class in mobile networks. As such, the timing requirements will exclusively be dictated by the limits of the specific machines. The absolute performance limits of machines are not fully known or understood, but machines are, in general, subject to the theoretical limits described by computational complexity theory and the laws of physics.

Presently, there is a consensus that future communication networks will have to pass from human speeds to machine speeds; this will be even more emphasized as we are moving toward 6G communication systems [33]. The distinction between real-time and non-real-time

[1]The definition of the term "flow" corresponds to "a state of concentration so focused that it amounts to absolute absorption in an activity" [32]. When we experience flow, we lose track of time, and time feels accelerated.

optimization is also crucial for intelligent network optimization, as designing network elements that can cooperate distributedly and on different timescales is a complex task [34]. The Internet as we know it and current wireless networks have been designed for humans: humans browsing web pages, exchanging emails and messages, watching movies, and so on. Therein, we know that humans have limitations in terms of the visible spectrum (from 380 to 780 nm), the perceivable frame rate and resolution, and the audible frequency range (from about 20 Hz to 20 kHz). "This is why today's Internet—while fast enough for most humans—appears glacial when machines talk to machines" [33]. For example, an autonomous vehicle or a drone moving at 90 km/h will travel 100 m in 4 s. Avoiding collisions may require ultrafast decision-making: a delay of 100 ms could cause it to crash into something as far as 3 m away. However, what are the limitations of machines in the context of wireless communication systems? What does communicating and performing decisions and actions at machine speeds imply for the supported applications and services? We also note that the data generation process can vary significantly across communication actors. Some actors could generate "small and bursty" data, e.g., indicating a machine's status, whereas other actors or "things" (e.g., surveillance cameras) could generate very large amounts of data.

In addition to the involved communication actors, another classification considers who triggers the communication process such that there are event-triggered and time-triggered systems, respectively. In event-triggered (real-time) systems, a processing or a transmission activity is initiated as a consequence of the occurrence of a significant event. An example of an event-triggered system is an alarm system. In a time-triggered system, the activities are initiated periodically at predetermined points in time. An example of a time-triggered system is a production system with a preplanned production cycle or a traffic light system that follows a strict timing schedule. Event-triggered systems excel in flexibility, whereas time-triggered systems excel in temporal predictability. In event-triggered systems, the communication delay may be time-varying and quite susceptible to jitter. In time-triggered systems, it is essential to synchronize the actions of all participating nodes to a global time. Since the (off-line) scheduling predefines the time windows for all actions, the result is a time scheme with constant latencies and no jitter. If no synchronization is implemented, the latency and the jitter will most likely be of higher magnitude than for event-triggered systems.

III. STATISTICAL FRAMEWORK OF TIMING

An important element in defining a model for timing is the reference with respect to which timing is measured. In this section, we define the statistical framework for timing for the case of a single link, or even a multihop connection, between Node 1 and Node 2. In order to keep things simple at this stage, we also assume that the clocks of Node 1 and
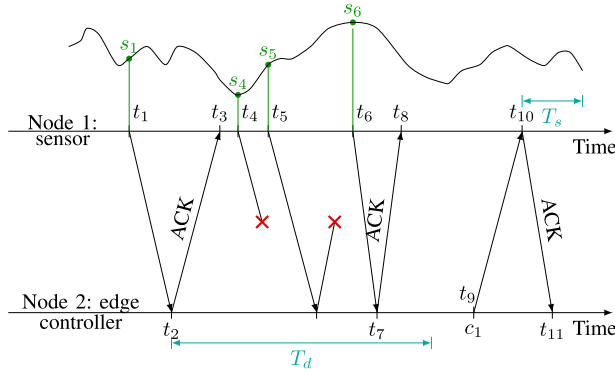
Fig. 5. *Illustration of the timing references for a link between a sensor (Node 1) and edge controller (Node 2).*

Node 2 are perfectly synchronized such that we can talk in terms of absolute time, as observed identically by both nodes.

A. Timing References and the Role of an Initiator

Consider the simple communication scenario in Fig. 5, in which Node 1 is a sensor that monitors a physical process and Node 2 is an edge controller. It is assumed that both nodes are synchronized and measure the time identically. Node 1 samples the physical process and sends updates to the edge controller. The sample s_1 is taken at time t_1, received by Node 2 at t_2 and acknowledged to Node 1 at t_3. Node 2 is interested in having an update on the state of the physical process that is as fresh as possible, i.e., to minimize the AoI with respect to the physical process observed by Node 1. When Node 2 receives s_1, its age is already $\Delta t = t_2 - t_1$. Hence, Node 1 measures the age with respect to a past timing reference t_1, associated with the value of the process state.

The system is programmed to work such that, if the controller does not receive any packet from the sensor within a time interval $\Delta t = T_d$, it initiates a safety shutdown of the system. For the example in Fig. 5, at time t_3, Node 1 learns that it must deliver at least one data sample to Node 2 before the deadline $t_2 + T_d$, or the system will shut down. Due to transmission errors, s_4 is not received by Node 2. The sample s_5 is received by Node 2, but its acknowledgment is not received by Node 1 such that, after t_5, Node 1 still considers the deadline to be $t_2 + T_d$ and invests extra communication resources to deliver the data sample s_6, whose reception at time t_7 is acknowledged at time t_8.

Finally, at time t_9, the edge controller sends the command c_1 to Node 1 to go into sleep mode for an amount of time $\Delta t = T_s$ after receiving the command. Node 1 sends an acknowledgment and goes into the sleep mode.

For all communication instances from Fig. 5, there is a certain time interval Δt during which communication takes place. We will refer to it as a *communication interval*, and an important aspect is the timing reference with

respect to which this interval is measured. The example illustrates three types of timing references.

1) *Past timing reference or, shortly, timing anchor:* This is the case when time is measured with respect to an instant that occurs in the past, such as the state of a monitored physical system. For example, AoI is defined with respect to the timing anchor, as, at the destination, the anchor is the generation time of the last received update.

2) *Future timing reference or, shortly, deadline:* In this case, the timing reference is at a point $t = T$ in the future, and it represents a certain deadline. The communication interval Δt is then measured backward, starting from the future moment. This reflects the fact that communication should start before time $T - \Delta t$ in order to meet the deadline.

3) *Relative timing reference:* In this case, one or more of the nodes participating in the communication process can choose the reference moment $t = 0$ and measure the interval Δt relative to that moment. This is the example from Fig. 5 with the sleep command.

For consistency, all these timing references are defined from a perspective of an external genie that can perfectly observe the system. In reality, the nodes can have a discrepancy in their timing references, and communication is used as a means to reconcile this discrepancy. For example, if Node 1 decides to denote a certain time as $t = 0$, then Node 2 does not know this until it receives a packet from Node 1. From the way they are defined, the past and the future timing anchors have a direct relation to a time instant in the physical world and are related to a sensing/actuating operation through which the digital system interacts with the physical world. Different from this, the relative timing reference is mostly related to the "digital time," as measured by the digital systems of the involved devices. For instance, the go-to-sleep command is not related to an event in the physical world, but it is initiated by a digital command conceived in the edge node.

Another important question is that of who plays the role of the *initiator* of the communication. For example, when Node 1 reports the status of a physical process, it is Node 1 that initiates the process. In a different case, if Node 2 sends a query that demands some information from Node 1, then the initiator is the receiver (Node 2). Depending on who has the role of the initiator, there are, in general, two types of communication:

1) *push-based communication*, where the initiator is the information sender;

2) *pull-based communication*, where the initiator is the information recipient.

At a first glance, push-based communication can be associated with a timing anchor or can be triggered by an event, while pull-based communication can be associated with a future deadline. However, this is not necessarily the case. For example, think of the case in which Node 1 is a controller that wants to put Node 2 in a certain state at a

future instant T. This is a push-based communication with a future deadline. As another example, Node 2 can send a query to ask for the most recent state of the system: this is a pull-based communication with a past anchor.

B. Statistical Characterization of Timing Measures

Let us assume that Node 1 observes the physical system at time $t = 0$, creates a packet of size D bits, and transmits it to Node 2. The communication interval Δt starts at $t = 0$, and it is convenient to describe the stochastic behavior of the connection by a *latency-reliability* function

$$F_D(\tau) = \Pr(\Delta t \le \tau \mid D) \quad (1)$$

which is a nondecreasing function that denotes the probability that the packet of size D is received and processed correctly at Node 2 by the time $t = \tau$. Intuitively, this function reflects the fact that, as time passes, Node 1 has more actions at its disposal to increase the probability that the packet is decoded by Node 2. If packets are never dropped and are always delivered without errors, i.e., $\lim_{\tau \to \infty} F_D(\tau) = 1$, the latency-reliability function is equivalent to the concept of a statistical delay bound, which is widely used in stochastic network calculus [35] to analyze delay violation probabilities in multihop queueing networks with stochastic arrival and service times.

The above stochastic model can be generalized by considering a more complex event in the communication system rather than the reception of a single packet. For example, in a multicast scenario, one can look at the time interval in which at least K nodes have received a certain data packet. Similarly, if there is a transmission of a batch of files, the event of interest can be the reception of at least L files from the batch. An interesting scenario is when reconstruction requires a specific ordered sequence of L packets carrying correlated information. Therein, timing measures have to be revisited; if packets do not arrive consecutively, timing (AoI) is measured as the difference between the current time and the generation time of the latest "entirely" received correlated sequence of packets. Further generalization can be made by considering a *prior context* $\mathcal{C}$ of the system instead of only a packet of size D. An example of a context is a prior knowledge that a node may have. Another example is the context in which Node 1 has the data file $\mathbf{D}_1$, and Node 2 has the data file $\mathbf{D}_2$, and we are looking at the occurrence of the event in which both nodes have both files. The event that we are looking at will be clear from the prior context such that we can write

$$F_{\mathcal{C}}(\tau) = \Pr(\Delta t \le \tau \mid \mathcal{C}) \quad (2)$$

which, like (1), is a nondecreasing function.

In order to expand the set of relevant statistical measures, recall that two basic problem categories in statistical modeling are *statistical decision* and *statistical estimation*, respectively. In the context of timing in communication systems, the above discussion is limited to discrete events and statistical decisions, and finding the probability that some event has taken place. A completely different set of problems is obtained when we put the statistical estimation in the context of timing.

To illustrate this, let us take a timing anchor. At time t, Node 1 measures a certain state, registers the value $x(t)$, and communicates the state to Node 2. The estimate that Node 2 has about the state $x(t)$ of the physical system after a communication interval of Δt is denoted by $\hat{x}_{\Delta t}(t)$. The quality of this estimate after the communication interval $\Delta t = \tau$ can be measured by a generic loss function $L(\hat{x}_{\tau}(t), x(t))$, which should increase over time, at least in the average sense. The mean square error (mse), i.e., $\|\hat{x}_{\tau}(t) - x(t)\|^2$, is a common way of measuring this estimation error, but other loss functions can be used.

To support this observation, one can think of a communication strategy that continuously sends refinements from Node 1 to Node 2 about the state observed at a past anchor t. Alternatively, consider the special case in which Node 1 creates a single packet to describe $x(t)$, and this packet is an atomic unit of communication. In this case, $L(\hat{x}_{\tau}(t), x(t))$ has a particular form: it has a positive value (e.g., based on a prior knowledge that Node 2 has about $x(t)$) until $\tau = \tau_0$ that corresponds to the time $t + \tau_0$ at which Node 2 receives successfully the packet from Node 1. For $\tau > \tau_0$, it is $L(\hat{x}_{\tau}(t), x(t)) = 0$ or, possibly, the quantization error for $x(t)$. As another example, in a setup with distributed learning, the true $x(t)$ is not known to any of the nodes, but the (empirical) loss decreases as learning progresses in time. In the opposite case, in which the state is high-dimensional (e.g., an image or depth map of the environment in a cooperative driving scenario) or the observation is distributed among different sensors, even the definition of the loss can become complex, and decisions need to be made based on which piece of information is more important at a given moment, i.e., which transmission results in the biggest reduction in the loss function, which never goes to zero. Finally, with respect to timing relativity and simultaneity, in remote actuation and distributed real-time systems, we need to minimize $L(\hat{x}_{\tau}(t), x(t))$ for small $\Delta t + \mathcal{T}$, where $\mathcal{T}$ could include time spent for information generation, processing, and reconstruction [36].

C. Summary of the Basic Framework

Our framework for describing the timing problems in communication systems will rely on the timing reference and the statistical operation (decision or estimation). In order to keep the discussion compact, we do not use the role of initiator to add a third dimension, but we will use it as supplementary information where relevant.

1) *Timing anchor:*

   a) *Statistical decision:* Node 1 sends updates to Node 2 about the state of a monitored physical process.

A relevant timing measure is AoI. This can be push-based such that Node 1 decides when to send an update and attempt to ensure that Node 2 always has the freshest update on the status of the process. Alternatively, it can be pull-based such that Node 2 sends queries to demand status updates.

b) *Statistical estimation.* Consider a case similar to the previous one, where Node 2 receives updates from Node 1 about the state of a certain physical system. However, the state at time t is a multidimensional variable and cannot be accommodated in a single packet transmission but rather sent gradually. Hence, the correctness of the estimate that Node 2 has about the state at time t will increase over time. In a push-based communication, Node 1 initiates the transmission and transmits either until receiving a stop feedback from Node 2 or until estimating that Node 2's estimate about the physical system is sufficiently correct. In the pull-based case, Node 2 initiates the communication, and as it receives data from Node 1, it judges the quality of the estimate and, if it is not satisfactory, sends further pull requests to require more data.

2) *Timing deadline:*

a) *Statistical decision:* This is the classic case of a latency constraint, where a data packet should be delivered within a given deadline. The timing requirements of URLLC are defined in this context, as the packet is considered to be ready for transmission and needs to be delivered within a deadline (e.g., 1 ms).

b) *Statistical estimation:* Here, the receiver wants to estimate a certain variable within a given deadline and with an error no larger than a certain ϵ. One example from satellite communication entails a satellite that is visible for a limited time period, as the estimation needs to have acceptable accuracy until the link becomes unavailable.

3) *Relative timing reference:*

a) *Statistical decision:* This is the case in which a group of nodes wants to reach a consensus on a decision, and the set of possible decisions is discrete. For instance, the decision could be related to the precedence among the autonomous vehicles at a traffic crossing or to which blockchain transaction is considered valid.

b) *Statistical estimation:* A use case that falls into that category is distributed learning. Therein, the model training among nodes should be completed within a given interval from the time the first node has initiated the process, where completion is declared based on a certain threshold on the measure of loss.

Another level of complexity is revealed when we start to ask other questions: what does one node know about the knowledge of another node? In the case with a past anchor, Node 1 observes the state of a physical system $x(t)$ and

sends it to Node 2, which, in turn, makes an estimate $\hat{x}_\tau(t)$. One related question is: what does Node 1 know about the value of $\hat{x}_\tau(t)$? In a simple case, if Node 2 receives the packet successfully from Node 1 after an interval τ_1 and sends an ACK that requires time τ_2, then Node 1 knows $\hat{x}_{\tau_1+\tau_2}(t)$ perfectly. This is important in, for example, a status monitoring application where Node 2 needs to take an action based on the current state of Node 1. Then, Node 1 may know what the status is only after time $t + \tau_1 + \tau_2$. If Node 2 cannot decode the message and sends instead a NACK, then Node 1 knows the last correctly received status $\hat{x}_{\tau_1+\tau_2}(t - \Delta)$, transmitted from Node 1 to Node 2 at $t - \Delta$. Note that, upon transmission failure, Node 1 has the option to resend the same data and thus potentially use some combining with the previously received version of the data. Alternatively, retransmissions of the same data are dropped, and upon failure, the status of the process monitored by Node 1 is sampled anew and transmitted. Two-way communication is further discussed in Section V-B.

IV. PUTTING THE PRIOR ART WITHIN THE STATISTICAL FRAMEWORK

Now that we have defined the basic framework of timing measures, we can look at the existing body of work on timing in communication networks, trying to frame the rich literature into the categories defined in Section III. Some recent standards, including those developed by 3GPP, are beginning to consider these factors and metrics in a more general way, as shown in the diagram in Fig. 2, which is adapted from 3GPP. The model includes some of the concepts that we will discuss in the later sections, such as the notion of timing at the higher and lower layers, as well as the importance of the intertransmission time, defined as "transfer interval" by 3GPP. In the following, we examine a few interesting cases, which are familiar to the networking community and include the notions used by 3GPP. Our framework can subsume all these metrics in the same perspective and allows us to think holistically about timing and the related metrics. The relevant references are summarized by topic in Table 1: as the table shows, past anchors are the most common method of measuring timing and are often used in standards and protocols, while the use of relative timing references, which consider complex networking scenarios, is still largely unexplored.

A. Latency

Latency, also known as delay, is perhaps the simplest and oldest metric used to measure timing in networks; it captures the time that a packet spends in the network. Latency is measured with respect to an event that happened in the past, and it characterizes packets. In our framework, it is an example of a past timing reference: the latency timer starts when the packet is transmitted by a given layer in the protocol stack and stops when the packet arrives at the same layer at the destination. A closely related metric

Table 1 Representative Works Grouped by Their Relation to the Instances of the Statistical Framework

Topic	Timing reference	Significant references
Latency in queuing systems	Past anchor (stat. decision)	[37]–[43]
Statistical latency guarantees	Past anchor (stat. decision)	[44]–[47]
End-to-end latency in realistic systems	Past anchor (stat. decision)	[35], [48]–[50]
Latency deadlines	Deadline (stat. decision)	[51]–[54]
Timely throughput	Deadline (stat. estimation)	[55]–[57]
Deadline-based optimization	Deadline (stat. decision)	[58]–[65]
End-to-end deadlines	Deadline (stat. decision)	[66]–[70]
AoI	Past anchor (stat. decision)	[15], [71]–[82]
Goal-oriented AoI extensions	Past anchor (stat. estimation)	[83]–[85]
Pull-based AoI	Past anchor (stat. estimation)	[13], [86]
3GPP service requirements	Deadline (stat. decision)	[16], [54], [87]–[89]
Time To First Fix (TTFF)	Past anchor (stat. estimation)	[90], [91]
Synchronization requirements	Past anchor (stat. estimation)	[20], [27], [92]–[95]
Distributed learning requirements	Deadline (stat. decision)	[96]–[99]
Distributed learning speed	Relative (stat. estimation)	[100], [101]

is the round-trip time (RTT), which represents the latency over both sides of the connection, from the moment the packet is sent to when the transmitter receives the related acknowledgment. Latency has been studied extensively in different system setups, both theoretical and practical. An exhaustive literature review on the topic is outside the scope of this work, but we will list some relevant works and refer the reader to existing surveys for a deeper examination [37], [39].

The main theoretical tool for analyzing latency in networks is queueing theory, which can go from simple $M/M/1$ systems [40] to complex access mechanisms with different arrival patterns [41]. In particular, random access mechanisms, such as ALOHA [42], have been extensively studied [43]. One of the first additions to plain communication latency was the observation that applications such as video are also sensitive to the jitter, defined as the variation in latency of the packet flow [102]. Another extension is to derive bounds on the tail of the latency distributions, which can provide statistical QoS guarantees, such as effective bandwidth/capacity and bounds in queue length and latency violation probability [44], [45]. The analyses get complex when the intricate correlations in the arrival and/or the channel processes are considered [46], [47].

It is also possible to study end-to-end latency, going beyond a single link and looking at the connection level. In this case, there are fewer theoretical works analyzing the latency under realistic access networks; they mostly consider the two-hop case [48] or Poisson traffic [49] due to the complexity of analyzing latency in other scenarios. Even small random access networks with bursty traffic become rather intractable due to the coupling among the queues [38], a scenario that still remains largely

unexplored. An alternative to address this complexity is using stochastic network calculus [35], which is a probabilistic extension of network calculus [103]–[105]. Network calculus builds upon dioid and $(\min, +)$ algebra and provides backlog and delay bounds to understand the statistical multiplexing and scheduling of nontrivial traffic sources. Its stochastic counterpart has been extensively employed to analyze wireless networks in various settings with time-varying random service rate [106]–[108].

Several learning-based mechanisms to reduce latency have been proposed for 6G networks, often including computation as well: the placement of computation tasks and network functions is critical for reducing latency in complex tasks [109], and the joint consideration of computational and communication aspects can lead to lower overall latency for different services [110]. Domains that have strict constraints and high throughput, such as VR transmission [111], or fast mobility, such as vehicular communications [112], pose specific challenges that need to be addressed individually. The use of digital twin models [113] of the network can also improve the effectiveness of learning-based optimization schemes, providing more training data and reducing the impact of the training on the real network. These models can be used with any kind of learning model, such as federated learning, and improve resource allocation for complex distributed applications, such as distributed ledger technology (DLT) [114].

The latency is minimized by reducing the time that a packet spends in the network, including the initial access delay, which can be large in wireless networks. In the queuing models mentioned above, this is equivalent to decreasing the total time spent in queues throughout in the network, which can be done by increasing the service

rate and decreasing the rate at which packets enter the network. However, reducing the latency comes at the cost of reducing the throughput of the network, and finding the optimal tradeoff is a central topic in many of the works above. At a more practical level, minimizing the end-to-end latency has also been one of the main goals of the recent research on transport protocols. Congestion control mechanisms are often too aggressive and overshoot the available capacity, causing a significant increase in the latency (see [50] and the references therein). The practical role of congestion control in terms of latency is in the shaping of the traffic, which, in turn, affects the state of the queue and, consequently, future decisions from all transmitters. This tight coupling makes the use of a metric as old and traditional as latency an interesting yet challenging research avenue.

B. Deadline-Constrained Traffic

The rise of real-time or near real-time and interactive applications has revealed the fact that minimizing latency is not sufficient for the smooth operation of such applications. Specifically, the network should operate with respect to *deadlines* [51]–[53]. Deadline-constrained traffic is an example of a future timing reference. When a packet is generated by the transmitter, the timer does not move forward but backward from the maximum allowed latency T. If a packet arrives within the deadline, i.e., before the timer reaches zero, the transmission is successful, and thus, the latency of the packet is irrelevant as long as it arrives within the deadline. The URLLC traffic class [54] in 5G and beyond systems is a classic example of deadline-constrained traffic, which is relevant in industrial scenarios.

A metric called timely throughput [52], [55] measures the amount of traffic that can be successfully delivered within the deadline, potentially including the effects of computation [56]. In general, there is often a tradeoff between the achievable throughput and the tightness of the deadline T, as setting a tighter deadline requires more resources for every single packet [57]. Naturally, the achievable deadline has a hard floor given by the minimum latency in perfect conditions: while 5G and beyond systems envisage achieving deadlines below 1 ms, the technical challenges may involve computational components and hardware limitations, as well as the effect of the medium access mechanism used.

As the timing reference is in the future, it is not necessarily ideal to handle the packets according to a first-come first-serve (FCFS) policy. As a result, optimizing the communication for deadline-constrained traffic is more involved than simply minimizing the latency. Most works that deal with deadlines aim at optimizing the medium access [64], resource provisioning [58], interference management [65], and packet scheduling [115] to reduce the deadline violation probability [59]. Another example is the optimization of error targets in deadline-constrained

hybrid automatic repeat request (HARQ) protocols for URLLC [116]. It is also possible to jointly optimize the scheduling with other transmission parameters, such as power control [60], [61]. More advanced schemes include the use of Markov decision processs (MDPs) [62] and randomization [63], combining the scheduled approach with adaptive techniques that alter scheduling decisions to fit traffic patterns.

It is also possible to impose deadlines on end-to-end traffic, providing probabilistic guarantees or adapting the sending rate to make sure that packets meet the deadline [66]–[68]. In this case, multiple connections are often used along with packet-level coding [69], considering the latency not in terms of a single packet but of an application block. If we go even higher on the protocol stack, an interesting case is given by HTTP adaptive streaming (HAS) [70], a video streaming protocol at the application layer. Therein, the deadline is not fixed, as it does not represent interactivity but depends on the state of the video playout buffer at the receiver: in order for the video to play smoothly, the transmission must be completed before the available video segments finish playing. This is an example of a relative timing reference, as the deadline for each block of data depends on the content of the packets themselves and on the state of the playout buffer.

C. Age of Information

Research on AoI has seen a remarkable development over the past decade [15]. In its original form, AoI refers to the timing metric that describes the *age* of the most recently received packet at the destination. Specifically, consider a sequence a packets generated by the source at times $\tau_1, \tau_2, \ldots$ and received by the destination at $\tau_1', \tau_2', \ldots$, respectively, and denote the generation time of the most recent packet by

$$\xi(t) = \max \left\{ \tau_n \mid \tau_n' \leq t \right\}. \tag{3}$$

The AoI at time t, usually denoted $\Delta(t)$, is then defined as

$$\Delta(t) = t - \xi(t), \quad t \geq \tau_1'. \tag{4}$$

Since packet generation times and transmission latency are usually random, $\Delta(t)$ is a random process with sample paths that increase linearly between packet receptions, leading to the characteristic sawtooth pattern often associated with AoI.

The AoI metric fits well into our framework by defining the timing anchor to be the instant at which the most recently received packet is generated, i.e., $\xi(t)$. Note that the timing anchor is updated every time a new packet is received. This is different from the traditional latency metric where the anchor is updated when the transmitter generates a new packet.

The fact that AoI measures the freshness of the information available at the receiver makes it a representative metric for remote monitoring and control-oriented tasks. The sawtooth pattern of linear AoI has led to many works deriving its average in different systems, such as carrier-sense multiple access (CSMA) [71], ALOHA [72], and slotted ALOHA [73] networks. Furthermore, the notion of AoI can be generalized to measure any nondecreasing function of the age [74], [75]. More specifically, as discussed in [74], nonlinear aging functions can implicitly capture the autocorrelation structure of the source signal. That work considers and analyzes three cases: the linear, the exponential, and the logarithmic as aging functions. When the autocorrelation is small, the exponential function can be a relevant choice since it penalizes the increase of system time between updates, which, in turn, will affect significantly the remote reconstruction of that process. If the autocorrelation is large, then the logarithmic function becomes more relevant, whereas, for intermediate values, the linear case can be a reasonable choice. This is also a step toward the effective age and the importance of information [117], discussed in detail in the following. The AoI has also generated several related metrics, which usually considers relative timing references. The most common example is Peak Age of Information (PAoI), which samples the AoI immediately prior to the reception of a new packet. Using the notation introduced above, the PAoI is the *discrete* random process $\Delta'(\tau_2'), \Delta'(\tau_3'), \ldots$, where $\Delta'(\tau) = \lim_{t \to \tau^-} \Delta(t)$. This is useful when measuring the worst case performance of a system and, particularly, due to its analytical tractability, when considering not just the mean, but higher moments of the age distribution [76], metrics related to its tail [77], or even the complete probability density function (pdf) [78]–[81]. The PAoI captures the key characteristics of the age process. Furthermore, as shown in [80] and [118], the average PAoI and the average AoI coincide in discrete-time systems under the generate-at-will policy of status updates.

Gossip is a mechanism to convey information, such as status updates, in distributed systems and networks. Thus, considering timeliness and freshness metrics in such setups becomes relevant and important. The work in [119] considers a setup where a source transmits updates that are distributed over a graph by a gossip network. An early attempt to study AoI in gossip networks is [120]. Yates [121] provides AoI analysis tools for gossip algorithms on network graphs, and the metric of version AoI is defined therein, extended in [122] and [123].

The Age of Synchronization (AoS) is defined in [124] to capture the freshness of a local cache. Moreover, the problem of how a local server allocates the refresh rate for each source to maintain overall data freshness given a constraint on the total refresh rate is also studied therein.

The challenges for AoI optimization are also particularly relevant in the integration of vehicular and nonterrestrial networks in 6G: in the former, large quantities of data need to be exchanged while maintaining a very low AoI in a highly dynamic environment, and distributed reinforcement learning (RL) solutions, along with the use of learning-based models at the edge, can control the tradeoff between the mobile network efficiency and the AoI of the sensor data [125]. In the latter, the choice of the access scheme for traffic offloaded to satellites or drones is complicated by the higher propagation delay and the fast-changing network topology [126]. Nonorthogonal schemes can be beneficial in some cases [127] although the benefits are strongly dependent on the sampling process [128]. In these cases, repetition can also provide some additional reliability and improve AoI: irregular repetition slotted ALOHA (IRSA) [129] is a nonorthogonal solution that can allow large numbers of nodes to maintain a low AoI without explicit coordination using packet replication to protect the transmission from collisions. More complex retransmission schemes can also be used if packets are acknowledged, in which case more aggressive retransmission policies can be beneficial for AoI [130].

However, there is a tacit assumption in both AoI and PAoI that the information should be fresh at any time. We may instead consider the case in which an application at the receiver accesses the information only at specific points in time, as introduced by the Age of Information at Query (QAoI) framework [13]: this metric is similar to PAoI, but, instead of sampling the AoI only prior to a new packet reception, it does so when the application requests the information (i.e., when the information has the highest value). This transforms the setting from a push-based system to a pull-based one, where the application is dictating the transmission process. If the application works over discrete time intervals, then this leads to a better characterization of the information freshness as perceived by the application, and using it leads to very different choices in terms of system optimization [86]. In the schematic from Fig. 6, this would be equivalent to integrating information about the timing step of the estimator, monitor, or controller, as well as the sampling process at the source.

The last example of composite measures considers a sense–compute–actuate cycle where the system has a requirement regarding the maximum time between an event and the corresponding action. In this kind of wireless network-controlled systems, Node 1 plays a dual role of sensor and actuator, and Node 2 is the remote controller. Node 1 sends the state of the system to Node 2, and Node 2 replies with a control command to Node 1, which acts accordingly. The Age of Loop (AoL) can then be defined as the AoI over the two-way connection, which includes both the transmission of the system state from Node 1 to Node 2 and the transmission of the command from Node 2 to Node 1. In most systems, the command is a very short packet with a deterministic latency, so it simply adds a constant value to the one-way AoI. If the command has a stochastic transmission latency or significant size, the overall AoL is equivalent to the AoI in a tandem system, which involves the two directions of the connection, significantly
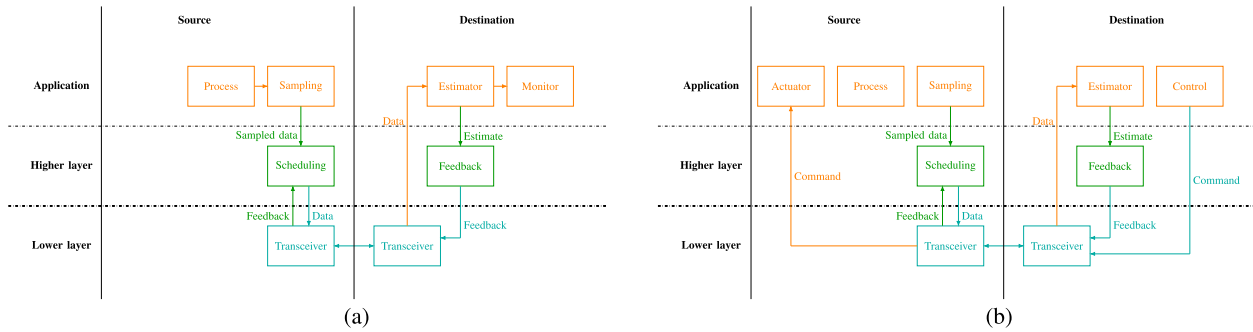
Fig. 6. *Transmission model in an AoI (or beyond AoI) optimization scheme. (a) Open-loop operation. (b) Closed-loop operation.*

affecting considerations for system optimization. A clear example of this difference is given by augmented reality (AR) systems, in which the visor transmits the image from a camera to a server. The server then renders virtual objects in the physical space and sends the content back to the visor: both the uplink and downlink flows can have a significant throughput [131], and the perceived delay between the user's movements and their effect on the virtual content depends on the AoL. In this case, Node 1 is at the same time the past anchor and the future timing reference [132].

The definitions of AoI and PAoI can readily be extended to the multisource case, as well as to the case where the sources are scheduled by the destination node. For a more complete overview of the literature on AoI, we refer the reader to [82].

D. Beyond Age of Information: Composite Measures and Data Quality

Coming back to our initial question of what is the right piece of information that should be transmitted at each time instant, a more complex set of timing measures arises when we aim at capturing both timing and other data quality attributes that define its significance for the system ultimate goal. Data quality has been broadly studied in the context of information systems with many different definitions and lists of attributes [133]. In any case, several of the desirable attributes for data quality are related to timing: freshness, currency, age, obsolescence, or staleness is often considered. Besides, data must be relevant, reliable, accurate, and complete.

Integrating the data quality in our network design can be done by moving from measures based on statistical decisions, i.e., on the average AoI, potentially measured at a specific point in time, or on the probability that its value is below a certain threshold, to measures based on statistical estimation, where we need to look more closely at the process that is being measured. Using this approach, several new composite metrics have been defined linking freshness and significance. The final objective is to design goal-oriented or *semantic* communications [36], [134], [135] for networked intelligent systems able

to optimize the use of resources. Fig. 6 shows a system model that can exploit this approach: while AoI is an end-to-end metric at the higher communication layers, and its optimization requires no knowledge of the blocks at the application level beyond the sampling statistics at the source, we can extend the framework to the statistical estimation by introducing knowledge about the process statistics at the source and the estimation, and the tracking process at the receiver. A classic example is the Kalman filter: if the receiver uses this filter model, as in [136], it is possible to maximize the accuracy by taking the uncertainty on the estimate, which is provided by the filter itself, into account. We can look at Fig. 6(a) for a fuller picture of how such an open-loop remote estimation system could work: unlike in simple AoI minimization, the feedback module should be provided with statistics on the current estimates so that it can make a better decision. Depending on the topology of the network and the capabilities of the nodes, we might consider the scheduling of updates to be entirely directed by the receiver. The receiver sends requests for new updates when it needs them, entirely directed by the source, which maintains an estimate of what the receiver knows and checks it against the actual state of the process, or distributed, with both nodes having partial knowledge and partial responsibility for the final decision on when and what to transmit.

The first example of these estimation-based metrics in the literature is the Age of Incorrect Information (AoII) [83], which extends the notion of fresh updates to that of fresh "informative" updates in statistical estimation, such that the age increases when the quality of the estimation of Node 2 about the process at Node 1 deteriorates. The Value of Information (VoI) [84] is a metric that looks not only at the freshness of the new information being transmitted, but at its content, as the relative timing reference is based on statistical estimation, and the objective is to minimize the difference between the actual measured process and the one estimated by the receiver through the updates.

We can now consider the system in Fig. 6(b): in this case, the feedback loop is closed not only at the higher transmission layer but also, at the application layer, as the receiver exerts some control over the remote process by

controlling an actuator. Teleoperation and remote control are standard examples of this kind of system, and in these cases, the main metric of interest is not the accuracy of the estimation but its effect on the control performance. The Urgency of Information (UoI) is a recent extension of VoI that considers this [85], making the objective not just the accuracy in measuring the process, but the stability of its control. If the controller needs to rely on the information sent over the wireless link, information that changes control decisions should be prioritized.

We would like to emphasize that all these AoI variants and other composite measures are metrics that can unify the past and future timing references. A generated packet starts its aging process immediately after its generation, so this is the past timing reference. In addition, systems can operate with AoI or VoI thresholds so that a new status update will be generated when the metric has reached a given value, which is a future time reference. When applied in different contexts, this metric can then provide a more holistic view of timing in communication systems.

E. Protocols and Standardization Efforts

Besides academic research, there is a vivid interest and ongoing work in the industry related to timing-aware designs for future communication networks.

One example is the 5G technology: over the last decade, the 3GPP made a great effort to understand the most relevant use cases and applications from the so-called *vertical domains*. This effort led to the initial classification into three generic services: enhanced Mobile BroadBand (eMBB), URLLC, and massive Machine-Type Communication (mMTC) [54], whose scenarios and requirements are set out in [87]. However, these categories are insufficient to capture the complexity and intricacies of the next generation of systems, including timing relations that go beyond the classical end-to-end latency and the reliability-latency couple. Therefore, 3GPP has continued the work to identify service requirements for new applications, such as the factories of the future, cyber–physical control applications, utility grid protection, medical monitoring, and autonomous driving [16], [88]. Three interesting timing metrics have been defined. The first one is the *survival time*, which is the time that an application consuming a communication service may continue without an anticipated message. We notice that this is dependent on the application and the allowed set of sequence of failures. Referring to our example from Fig. 5, one can understand the deadline T_d as a survival time. A related concept is the *watchdog timer*, used in control applications to automatically reset a device that hangs because of a software or hardware fault (or due to a delayed or lost packet when there is a communication network) [89]. The second metric is the *transfer interval*, which is, in principle, more relevant for periodic communication, but also applicable to scheduled aperiodic traffic. It is defined as the time elapsed between any two consecutive messages delivered by the automation application to the

ingress of the communication system. This measure is related to the relative timing reference. The third metric is the *time to first fix* (TTFF) [90], applicable to high-accuracy positioning and giving the time elapsed between the event triggering for the first time the determination of position-related data and the availability of position-related data at the positioning system interface [91]. This metric is related to a statistical decision with a past timing anchor.

Another interesting addition is the communication service reliability, which enlarges the URLLC reliability definition, and it refers to the ability to provide the communication service for a given time interval but under given conditions. These conditions would include aspects that affect reliability, such as mode of operation, stress levels, and environmental conditions. Reliability may be quantified using appropriate measures, such as the mean time between failures, or the probability of no failure within a specified period of time.

Requirements and definitions for the system synchronization are also observed, with multiple time domains: the global time domain, used to align operations and events chronologically, and the working clock domains, i.e., for a machine of the set of machines that physically collaborate. Different working clock domains may have different timescales, and different synchronization accuracies and precisions. Analogous to a latency budget, there is a synchronicity budget with the time error contribution between ingress and egress of the 5G system on the path of clock synchronization messages. Current solutions to achieving fast and continuous synchronization [92] will be certainly not sufficient to satisfy the demanding timing relations of the future use cases. For instance, industrial automation scenarios typically involve multiple timing domains. Despite the challenge of integrating 5G into a TSN synchronization network, the 3GPP has been working to make it feasible [137], [138]. Mechanisms for clock distribution are already included in Release 16, and the architectural solution has been consolidated in Release 17 [27]. The 5G-ACIA was established in 2018 and aims at bringing all industrial and networking stakeholders together to accelerate the adoption of 5G technology in the industrial domains. One of the objectives was to ensure that the requirements are adequately addressed in 5G standardization and regulation, and this includes many timing-related dependencies. For example, we already mentioned the report in [20], where the time-scale categories are defined, whereas [93] describes the requirements and functional capabilities needed to integrate 5G with TSN.

Finally, timeliness is identified by 3GPP as an attribute for timing accuracy useful to quantify the end-to-end latency [16]. A message is considered in time if it is received within the timeliness interval given by the target value, and the lower and upper bounds given by the allowed earliness and lateness. Besides, there can be a deviation or discrepancy between the actual time value and the target. This approach is useful for,
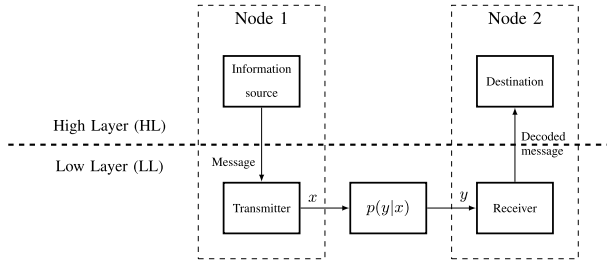
Fig. 7. *Shannon's communication model annotated by layering.*

e.g., applications where one of the nodes does not keep its own time but interprets the message arrival as a clock signal. Nevertheless, no specific metrics are defined.

Naturally, 3GPP is not the only entity working on timing requirements and standards, as several higher layer protocols whose specifications are published by the Internet Engineering Task Force (IETF) also consider timing. The well-known network time protocol (NTP) [94] is an older example that takes network timing requirements into account to achieve clock synchronization between different computers. In this case, the two-way latency is the critical parameter: as the two endpoints need to establish a common clock, they cannot rely on knowing the latency, and the common assumption is that the path is symmetrical, i.e., the one-way latency is half of the overall RTT.

Several other well-known protocols take time explicitly into account, using timing signals to trigger state changes and actions: a well-known example is the retransmission timeout in the transmission control protocol (TCP), which is triggered if a two-way deadline is not met, i.e., if the acknowledgment for a packet is delayed by more than a certain time [139]. The real-time transport protocol (RTP) [95] is another end-to-end protocol that takes timing into account, as it is designed for streaming media. The RTP packets are timestamped to compensate for jitter, playing each video frame or audio sample in the correct order and, at the correct time, and synchronizing events in a manner similar to the example from Fig. 3.

V. TIMING IN POINT-TO-POINT COMMUNICATION MODELS

We can now look at timing in point-to-point scenarios, i.e., when we have two endpoints communicating with each other. This is the simplest case of timing in networking as information is exchanged one to one.

A. Timing in Shannon's Communication Model

We start from the well-known Shannon communication model [140]. A variation on this model is depicted in Fig. 7, with two modifications from the original model in [140]: first, the noise source is absorbed within the conditional probability distribution $p(y|x)$ between the input x and the output y of the noisy communication channel, and second, the modules of the model are placed within the

context of a layered model of a communication protocol, consisting of only two layers: high layer (HL) and low layer (LL). It will be seen that these two layers are sufficient to introduce different notions of timing and their interaction through the layers.

The elementary unit in Shannon's model is the *channel use*, specified through the probabilistic relation $p(y|x)$ between input x and output y. The sequence of channel uses $x_1, x_2, \ldots, x_n$ are causally connected such that, for $i < j$, x_i takes place before x_j. Nevertheless, strictly speaking, timing is *absent* from Shannon's communication model. To elaborate, it should be noted that the model does not specify: 1) that the time interval between two adjacent channel uses should be constant and 2) what is the value of a time interval between any two adjacent channel uses.

Shannon's model is a mathematical model, and in order to connect it to a physical quantity, such as time, we need to specify additional parameters. We emphasize these points, as they go against the common view on communication systems, which assumes *a priori* that there is a constant time interval T between two adjacent channel uses, such that a data packet that requires n channel uses takes a time nT. Indeed, the norm in communication systems is that channel uses occur periodically, as a result of a certain sampling process. If periodic Nyquist sampling is assumed, the period between two samples is $(1/2B)$, where B is the used bandwidth. The time T_n that corresponds to n channel uses is then

$$T_n = \frac{n}{2B}. \tag{5}$$

Nevertheless, once a time duration is associated with T_n, for each $n > 0$, then latency of a given packet transmission is directly defined[2] as T_n. This is one common way in which timing enters the Shannon communication model.

The problem of transmission over a noisy channel, referred to as *channel coding problem*, deals with the design of the LL. In doing that, this problem is solved under a simple abstraction for the operation of the HL: the information source selects uniformly randomly one of the M possible messages, and this message should be recovered at the destination, which is also a HL module. This simple abstraction is a consequence of the separation between the channel and source coding, where the latter is assumed to deliver perfectly compressed messages to the transmitter. Furthermore, it is assumed *a priori* that there is data passed from the source to the transmitter such that the transmitter is always actively transmitting, and having an idle channel is not part of the model.[3]

---

[2]Strictly speaking, the duration of a packet with n channel uses is $(n - 1/2B)$, counting that the first channel use occurs at time 0, but, considering that $n \gg 1$, we will neglect this fact throughout the discussion.

[3]In fact, an idle channel can be seen as a specific type of transmitted symbol. Therefore, accounting for this requires changing the communication model.

Next, we discuss how Shannon's model can be used to represent the different options from the statistical framework from Section III. Note that, in this communication model, the message sent by the transmitter is already selected, encoded, and ready to be sent when the first channel use occurs. This involves at least the following tacit assumptions.

1) The receiver decodes the message instantly.
2) There is no additional latency due to protocol interactions.
3) Both the sender and the receiver are certain that the data selected by the sender are useful and timely for the receiver.
4) There is always data available.

We can conclude from this discussion that the only parameters that influence timing in Shannon's model under the assumption of Nyquist sampling are the number of channel uses and the bandwidth. Note that, for given $p(y|x)$, the channel capacity has an operational meaning by transmitting over asymptotically many channel uses. This may be interpreted as communication that takes an asymptotically long time or, alternatively, asymptotically large bandwidth. However, while increasing the bandwidth indeed leads to lower latency, the assumption of perfect compression means that the number of channel uses is already minimized and, thus, is generally not a free parameter unless we allow for distortion, as discussed later for the statistical estimation case. As such, the Shannon model has a deterministic latency and can be used for future timing references (e.g., it is easy to compute whether it will meet a deadline). However, this is not the case for timing metrics using a past anchor related to a physical process, such as AoI, since the process of sampling of the data has been abstracted from the model. Finally, the modeling of a relative time reference is trivial for the case when Node 1 determines the reference, but not possible in general, as the model has already predefined the role of a sender and recipient (e.g., in Fig. 7, Node 2 cannot be a transmitter).

In order to consider the statistical characterization of the timing measures, we first examine the statistical decision process. The latency-reliability function of a given channel can be determined by using the finite-blocklength theory [141]. Specifically, once bandwidth and T_n are fixed, then the number L of channel uses within the time interval Δt is also fixed. For a given size D of a data packet, the probability in (1) can be expressed in terms of the probability of success for a given data rate, expressed as bits per channel use, and this is precisely the type of result that can be obtained using the finite-blocklength theory.

We can now examine the statistical estimation case. Here, we need to consider the source encoding aspect of the model from Fig. 7, in order to show the quality of the estimated value at the receiver at the end of a given interval Δt. More precisely, the characterization of the quality of estimation that Node 2 can get after a certain
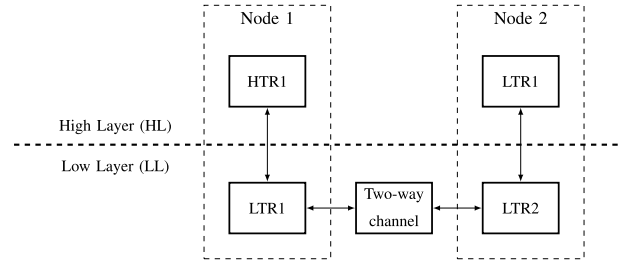

Fig. 8. *Two-way communication model with two layers. HTRi is the HL transceiver at node i; LTRj is the LL transceiver at node j.*

number of channel uses is subject to joint source-channel coding.

Some of the assumptions for the Shannon model, listed above, can easily be generalized. For example, instead of assuming that the receiver decodes the message instantly, we can assume that there is a certain decoding time T_d, potentially dependent on the actual received signal (e.g., resulting from a number of iterations of a belief propagation decoder due to noisy reception) and the content of the data (e.g., with the use of unequal error protection).

Assumptions 2 and 3 follow from the simplified functioning of the HL in Shannon's model. The transmitter at the LL obeys the command to transmit the data, not questioning its effectiveness or ultimate interpretation by the receiver. In principle, the model does offer the freedom to perform source coding and, thus, address the problem of statistical estimation. However, in order to characterize the source coding operation in terms of timing, one needs to enrich the model by introducing a time processing/computing related to the source coding operation.

B. Timing in a Two-Way Communication Protocol

In order to introduce more complicated protocol interactions, the one-way communication model from Fig. 7 is generalized with the two-way model from Fig. 8. In this model, each module is a *transceiver*, as it can both transmit and receive. The model loses the simplicity and the mathematical elegance of Shannon's one-way model, but it is suitable to introduce protocol interactions, without the ambition to have a consistent model that can mathematically describe the ultimate efficiency bounds, as it is the case with the notion of channel capacity. We assume a time-division duplex (TDD) system in which the time division is followed ideally by both nodes, i.e., when Node $i \in \{1, 2\}$ is in the transmit state, then Node $j \neq i$ is always in a receive state. Note that, in this model, we are not talking about individual channel uses but rather about a duration of transmission; in that sense, the duration can also accommodate the idle time that occurs due to the switching in a TDD system. This model can be generalized to other duplexing scenarios but is not essential for the discussion on timing.

In this model, Node 1 is the originator of the useful data (shortened to "data" when there is no danger of confusion). More specifically, each message is assumed to originate at the HL transceiver HTR1 and is passed on to the LL transceiver LTR1 to be sent over the noisy channel. To make timing part of the channel definition, let us consider some simple protocols that can take place in this model and start with a protocol for acknowledged transmission. Assume that Node 1 is a sensor that can measure some time-variant quantity in a CPS. HTR1 samples the system at time $t = 0$, processes the data, and passes the data of size D_1 on to LTR1.[4] These data are immediately sent to LTR2. Upon decoding the data, LTR2 passes it on to HTR2, and simultaneously sends and acknowledges size D_a to LTR1. Upon receiving the acknowledgment successfully, LTR1 immediately informs HTR1. Provided that the transmission of both data and acknowledgment is successful, the acknowledgment is received after one RTT, t_{RTT}, which can be computed using the same tools as in the one-way channel. This leads us to the first characterization of latency in a two-way communication protocol as a statistical decision process, where the latency is defined as the time from data transmission until reception of the acknowledgment. However, while it was sufficient to fix the bandwidth and the symbol period T_n in the one-way channel, the two-way channel also requires us to specify how the allocated symbols are split between data transmission and acknowledgment. Let us denote the fraction of symbols used for the data transmission by k so that, when there are a total of L channel uses within the time interval Δt, the data are transmitted with rate $R_1 = (D_1/kL)$, and the acknowledgment has rate $R_a = (D_a/(1-k)L)$. The latency-reliability function can then be obtained as the convolution of the one-way latency-reliability functions of the data and acknowledgment transmissions computed for R_1 and R_a, respectively.

Similar to the Shannon model, the protocol for acknowledged transmission is a *push-based protocol*, where the originator of the data (Node 1) knows exactly which data and when it is requested by the destination (Node 2). The model can be also used for *pull-based protocols*, where Node 2 initially sends a request of size D_r to Node 1, and in turn, it receives the data it requested. In this case, the timing quantities of interest are measured with respect to the moment when the request is sent by Node 2, not with respect to when the data are sent by Node 1. The model can also be extended to include retransmissions at LTR1, where the data are retransmitted until an acknowledgment is received. In this case, Δt represents a random multiple of t_{RTT} representing the number of transmissions required to get successful data followed by a successful acknowledgment transmission, taking into account the potential use of retransmissions and HARQ.

To characterize the statistical estimation, consider the case where Node 2 estimates the state at Node 1. Compared to the one-way model, the main difference is that the feedback can be used by Node 1 to decide what to transmit next or how to encode the data in the next transmission. This may, for instance, be the case in the training of a machine learning model, where Node 2 keeps requesting data until the test accuracy has converged or is sufficiently low. These benefits are also pronounced in pull-based systems, where Node 2 can use side information, such as confidence intervals, to decide which information to request in order to minimize the estimation error.

So far, we assumed that Node 1 had the data that were of interest to Node 2. In some cases, a more reasonable model is to assume that Node 1 and Node 2 are interested in exchanging data and coordinate decisions. This could, for instance, be the case in a factory automation scenario where two robots need to coordinate in order to solve a task. We defer the discussion of this case to Section VI-B where we consider the more general problem of distributed consensus.

A practical example of this kind of system is given by NTP synchronization: in this case, the data that Node 1 transmits are itself a timing reference. We note that, in a general case that does not involve a Shannon channel with deterministic packet transmission times, such as most wireless systems, the jitter in the network prevents the two nodes from knowing the one-way latency, unless they already have a common timing reference. The NTP assumes that the latency is symmetrical, dividing the RTT in two. As such, its precision is limited by the network latency. Naturally, the same considerations hold for timestamped data if the two nodes do not have a precise common reference: by determining the RTT, the two nodes can agree on a shared timeline of events, but the order of events that occur at the two nodes within one RTT of each other is impossible to determine in the absence of any information on the one-way latency.

C. Timing and Freshness of Updates

The AoI, which is one of the simplest timing metrics, extends latency in that it accounts for the characteristics of the source (information generator) as an entity regarding freshness.

The baseline one-way communication model consists of Node 1, which observes a certain process that evolves[5] over time and Node 2, which is interested in having the latest status of that process. For that, Node 1 needs to measure (sample) the process, generate an update at HTR1 that describes the status of the process, and forward it to LTR1 to be sent to the LTR2 and, ultimately, HTR2. The requirement for having the latest possible update at Node 2 has an impact on both the data generation process

---

[4]We can add a processing delay in HTR1, but we keep it simple for the moment.

[5]A generalization is to consider that the process changes over space and time, such as the temperature in a smart factory depending on the operations of machines that can create heat.

at Node 1 (i.e., when the sampling/measuring process occurs) and the data transmission process at LTR1. Assume that HTR1 forwards an update D_1 to LTR1 at time t_1. At a time $t_2 > t_1$, HTR1 generates a new update D_2, but, when it communicates (internally within Node 1) with LTR1, it finds out that the data D_1 have not yet been transmitted. Then, one action is to request LTR1 to purge D_1 and replace its transmission with D_2 such that communication resources are utilized for the latest update. However, this heavily depends on the purpose of communicating the status updates. In case one is interested in tracking and the past of the process, we may still need to transmit the freshest status update. Instead of purging the older ones though, LTR1 stores them in a queue and transmits them when no more urgent transmission is required. Since these updates are timestamped, they can be reordered upon reception at LTR2 to provide the required knowledge about the past. Furthermore, if there is a nonnegligible communication delay between LTR2 and LTR2, we may consider that LTR2 will forward the freshest received update to HTR2, and then, the rest will follow. This can be crucial in cases when there is a need for actuation at Node 2.

The concept of AoI and information freshness can be extended to two-way communication: Node 1 cannot know how fresh is the information at Node 2 unless Node 2 provides feedback, which means that there is two-way communication in place, and AoL accounts for the feedback loop as well. Another interesting two-way scenario is the one in which Node 1 is a sensor/actuator, while Node 2 is an edge controller. Node 1 sends a state to Node 2 and expects back a command that needs to be actuated. In this case, the correct system operation and stability depend on the AoL, defined as the age of the two-way loop and measured from the moment the state is sent until the command is received [132]. In general, one can define various composite measures of freshness taking into account the states and the data reception at both nodes, as we described in Section IV-D.

D. Timing in a Cascade of Modules

In the definition of information freshness, we have involved a cascade of modules from two layers: the higher layer doing the sampling and using the sample, and the lower layer responsible for transport of the sampled data. In general, end-do-end information processing is a sequential process that involves different operations, such as computation, compression, encoding, and baseband processing. This can be conveniently represented through a cascade of modules, exemplified in Fig. 9. It is easy to note how this model corresponds to the general system depicted in Fig. 6 and can be applied in the scenarios that we discussed with respect to the new timing metrics in the literature. It should also be noted that the communication module itself can have multiple subcomponents, such as a wireless access link and a core network connection.

Let us take, at first, the case in Fig. 9(a), in which the processing time of a given module depends only on the
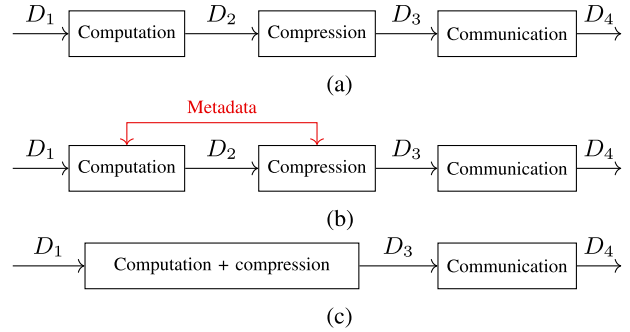

Fig. 9. *Example of information processing through a cascade of modules. (a) Processing time of each module depends only on the data. (b) Processing times of computation and compression are coupled through control metadata. (c) Computation and compression are combined in a single module, and their processing times are inseparable.*

input received from the previous module. Recalling the latency-reliability function $\Pr(\Delta t \leq \tau \mid D)$ from (1), we can now interpret $D = D_1$ as the input data to the first processing module (computation). The total timing budget is given by the interval

$$\Delta t = \sum_{i=1}^{M} \Delta t_i \qquad (6)$$

where Δt_i is the contribution of the ith module. For the model in Fig. 9(a), the processing time Δt_i depends only on the input data D_i, that is, given D_i, then Δt_i is conditionally independent of the other Δt_j, where $j \neq i$. This model is the basis for optimizing the latency budget: for example, one can increase the processing time of the compression module in order to get a better compression, which would decrease the size of D_2 and lead to a potentially shorter Δt_3.

Fig. 9(b) depicts a two-way exchange of metadata/control information that coordinates the processing of modules for computation and compression.[6] The cumulative processing time is again given by (6), but, now, Δt_j for $j = 1, 2$ is not only dependent on D_j: the joint distribution of Δt_1 and Δt_2 also depends on the exchanged metadata. For example, the computation module may signal to the compression module that current data have a higher priority, which will change the processing in the compression module and lead to a lower Δt_2. In principle, the timing performance attained for coupled modules in Fig. 9(b) should never be worse than the one for Fig. 9(a). Nevertheless, when working with very short time scales, one needs to take into account the timing performance of the metadata/control exchange as well.

---

[6]In this example, we are illustrating a possible coupling between computation and compression. In general, compression can be coupled with communication, or all three operations can be joined into a single module.

Finally, in Fig. 9(c), the total timing budget is given by

$$\Delta t = \Delta t_{12} + \Delta t_3 \qquad (7)$$

where Δt_{12} is the time consumed by the joint computation/compression. For a well-defined scenario, Fig. 9(c) can be optimized to achieve the best timing performance, i.e., it can be ensured that $\Delta t_{12} < \Delta t_1 + \Delta t_2$. This is because any operation regime that can be attained in Fig. 9(b) can be attained in Fig. 9(c), but not vice versa.

The contrast between Figs. 9(b) and (c) reflects the ever-present tradeoff between architecture and performance. Fig. 9(b) reflects an architecture that can scale and proliferate, such as the Internet, where the interaction among black boxes takes place through well-defined interfaces. This is also the approach of O-RAN [142], which allows the wireless system to be built based on components with open interfaces. What is uncertain in terms of performance is whether the specification of the interfaces between the black boxes, along with the timing performance of their interaction, is capable to offer superior timing performance. However, given the interfaces, there is a broad base of competitors that can offer new processing algorithms and smart interactions through those interfaces. One important feature to achieve this goal is synchronization: the system should operate under a global time domain and distribute the working clock accordingly, as explained in Section IV-E. On the opposite side is a solution fully implemented by a single vendor, which can optimize the interactions and the timings of different operations beyond the limitations of the open interfaces. However, it is uncertain at the time of design whether the optimization is versatile enough to support all future timing requirements.

VI. TIMING IN NETWORKING MODELS

The features that we discussed above were all related to a point-to-point model, which is conceptually simple, as the two actors' roles are clear. If we extend the framework to a *network* of actors, we have three principal models.

1) *One-to-many transmission:* In this case, a single transmitter needs to relay information to multiple receivers. In more classical terms, broadcast or multicast applications follow this model.

2) *Many-to-one transmission:* In this case, an aggregator node receives updates from multiple sources. This can represent, for example, a remote estimation or control process, in which a central monitor gathers data related to a complex process from multiple sensors;

3) *Many-to-many transmission:* This is the most general case, in which there are multiple transmitters and receivers. Examples within this category are updating of a DLT, a "swarm" scenario with coordinated robots or drones, or distributed learning networks.

In the following, we elaborate on representative examples of the many-to-one category (see Section VI-B) and the many-to-many paradigm (see Sections VI-B–VI-D).

A. Model for Networked Control Beyond Point-to-Point

Networked control systems refer to systems where multiple devices exchange information with the aim to coordinate some action that requires precise control and frequent feedback. Control systems can be centralized, as is often the case in industrial manufacturing scenarios, or decentralized, e.g., a power distribution system comprising distributed energy sources and loads that need to be controlled to maintain the stability of the overall system [143]. Note that, although sensors, actuators, and controllers may be distributed, we limit the discussion in this section to the case where the control actions are given from a central entity and dedicate Section VI-B to the problem of distributed consensus that for instance arises in multiagent systems.

In the centralized case, the systems usually fall into the category of real-time systems and follow a star or ring topology with a single controller that receives samples from sensors and sends directions for actuators [144]. The timing requirements are dictated by the controller, which is typically executed periodically in the sense–compute–actuate cycles, during which the controller receives feedback from sensors, computes new actions for actuators, and sends the actions to the actuators. This way, the communication link provides the means to close the feedback loop and synchronize the components. The tight integration of communication and computation enforces a deterministic time schedule of the communication by reducing the need for queueing, dynamic scheduling, and so on.

To put this into the statistical framework, let us consider the timing from the point of view of the controller, which dictates the control process. When the controller is about to compute the control actions, the most critical timing references are the times at which the current sensor readings must be recent to allow the controller to pick an accurate control action. In particular, if the readings are dated, the controller needs to predict the sensor state based on what it already knows (i.e., operate in the open-loop mode), which is likely to be associated with a high amount of uncertainty. This timing perspective can be described using the AoL and its variants sampled at the discrete instants where the controller is about to compute its next control actions, underlining the observation mentioned earlier that the information needs to be delivered on time as opposed to in-time.

So far, we have assumed that the sensors and actuators are directly connected to the controller in a star or ring topology. However, the topology could also be a completely virtual overlay network that is implemented on top of general infrastructure, such as the Internet. This situation arises, for instance, in power distribution systems with separate power and communication infrastructure, teleoperation, and the Tactile Internet [2], where there is a physical distance between the sensors and actuators
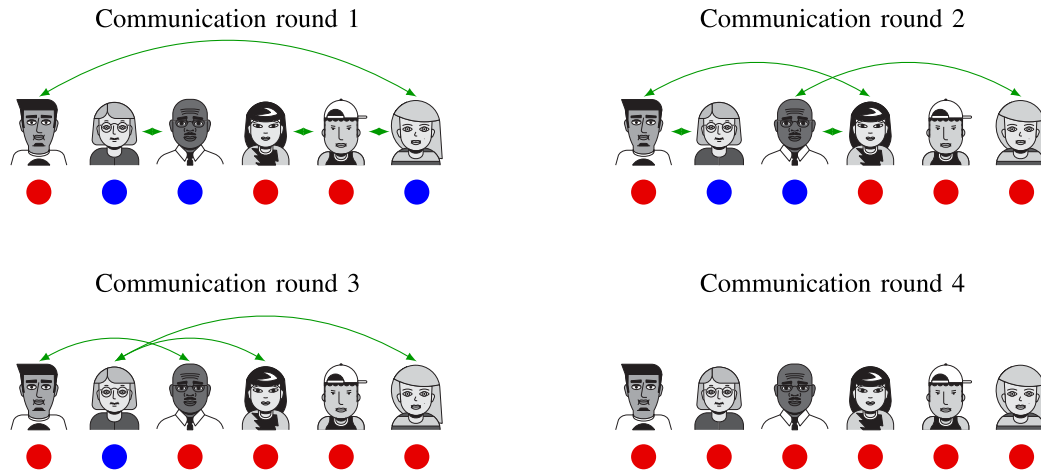
Communication round 1

Communication round 2

Communication round 3

Communication round 4

Fig. 10. *Simple example of a distributed consensus.*

on the one side and the controller and the operator on the other side. The increased distance between the components, as well as the fact that the infrastructure is shared with other (unknown) users, makes precise control over the timing more challenging. One way to compensate for this is to reduce the frequency of the cycle, provided that the controlled process allows for it, or to adopt predictive strategies [145] that attempt to estimate the state of the remote system and revise control commands accordingly.

However, a more common strategy is to delegate part of the control to controllers close to the actuators so that the system forms a hierarchical control system in which the primary (high-level) controller is responsible only for supervising the secondary controllers [146]. In this case, the timing perspective becomes more complicated, as it is not sufficient to consider timing from the perspective of the primary controller but also of the secondary controllers, which should adapt to the state of the primary controller. Consequently, it is useful to adapt the same hierarchical structure when considering the timing anchors: the primary controller's reference is both the originator of its overall control strategy, e.g., an operator, and the state of the secondary controllers. Similarly, the references of the secondary controllers are both the state of the primary controller and the states of the sensors and actuators that it directly interacts with.

The hierarchical control structure can be generalized further to a fully decentralized control system where sensors, actuators, and controllers are interconnected as a mesh network. From the perspective of a single device, the timing references are given by the states of all other devices. However, the overall objective carried out by the devices may have a deadline, as is the case in self-driving cars that are coordinating to prevent an accident. When the devices are acting independently toward reaching the overall objective, the problem is characterized as a distributed consensus problem as discussed next.

1) *Past timing reference:* In this case, the AoI and AoL are the most relevant metrics. Having data from multiple sources, or even having multiple controllers coordinate, complicates the definition of a single metric, but common strategies are either to look at the average or to consider the last stragglers, adopting a more careful approach.

2) *Future timing reference:* If we know that the controllers have compensation systems for latency, as we discussed above, there is a limit to the maximum latency that they can compensate without significantly degrading the system's performance. We can then set a deadline from the moment the data are generated to the moment they arrive at the controller or controllers or from the moment the data are generated to the execution of the subsequent control command.

3) *Relative timing reference:* We can also consider the control system itself in the timing computation, setting objectives related to the control performance. In this case, the AoI of measurements is substituted by the UoI or AoII, as we have explained in Section IV-D.

B. Distributed Consensus and DLT

Consensus is a well-known problem in distributed systems, in which multiple nodes need to arrive at the same conclusion over a measurement or a future action by exchanging messages over a constrained communication system as a means to jointly achieve a global objective. The problem is exacerbated if a subset of the nodes is faulty or actively malicious, as the system needs to protect itself from incorrect or harmful information. A simple example of consensus is depicted in Fig. 10, in which six people (voters) must decide on casting a red or a blue vote. In each round, a maximum of two other voters can be contacted using unicast communication. Looking at the first round, voter 6 receives information that voters 1 and 5

will choose red, so being in minority she decides to change her vote to red. Instead, voters 2 and 3 exchange messages that reinforce their intention to vote for blue. In the figure, three rounds of communication allow the voters to agree on voting for red, but the solution and the time until it converges depend on the availability of communication resources and the initial preferences of each person.

A canonical example of consensus in the presence of untrustworthy nodes is the Byzantine generals problem [147], where several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The commanding general must decide on a plan of action and communicate it to the other generals to be carried out in unison. However, there might be one or several traitors (including the commanding general itself) that disseminate false information or are otherwise unreliable.

There are many practical examples of consensus in distributed computing [148] and multiagent systems. A prime example is for implementing distributed management of road intersections and for increased efficiency and safety. The idea is that the cars communicate with each other to coordinate their actions (speed and direction). The consensus is achieved through an iterative process where the location-aware vehicles select conflict-free trajectories that minimize their travel time. Besides the nonidealities of the communication channel, another limitation to take into account is the deviation of the measurements from the true location. A second example is represented by DLT. Although the rise of blockchain and DLT has been driven by cryptocurrency and supported by wired networks, there are more and more envisioned use cases in existing or future wireless networks [149]. Generally speaking, the DLT enables consensus among a group of geographically distributed nodes without a central controller. Examples in the wireless realm include *ad hoc* networks and clusters of small cells. As all transactions need to be confirmed by a qualified majority of the nodes (weighted either through proof of work or other mechanisms, such as proof of stake) before they are inserted in the ledger, timeliness is crucial. Furthermore, DLT is designed to be resistant to malicious nodes, as long as they do not represent a majority, thus placing the problem in the "Byzantine generals" category.

Consensus in these scenarios is communication-heavy, and defining timing reference(s) in this context is not trivial. The simplest way to measure consensus is by taking a *bird's eye* view of the network, i.e., considering consensus to be achieved at the instant when the last node required for the majority gets the necessary information. However, this view is often unrealistic in real networks, as it requires ideal communication links and full synchronization among nodes. Therefore, a full consensus can be defined when all nodes have been informed, taking the point of view of the last straggler node to confirm that it received and accepted the transaction: in game-theoretic terms, all agents must have complete information on the state of the system, i.e., not only must the consensus state be common knowledge,

the fact that it is common knowledge must also be common knowledge. A relaxation to this definition would be to declare consensus once the node that started the update is informed that consensus is achieved.

Gossip networks, in which nodes propagate information generated by any of the others [121], are a useful model for the scenarios that we outlined above. A real application of gossip networks is the distributed ledger, in which a majority of nodes (often weighted by their computing power, or their stake in the transaction itself) must agree on a transaction before it can be inserted into the ledger. The meaning of consensus depends on the mechanism used for the ledger: classical blockchains use a mining process, such as the well-known and resource-intensive proof-of-work [150], while HyperLedger uses a lighter method that is based on simulating transactions against the local ledger [151]. In general, the hard consistency requirements before a transaction can be confirmed make distributed ledgers a very complex scenario in terms of timing but clarify the timing anchor: the delay in a transaction is the time difference between the instant a transaction is initiated and the instant in which the originator node is updated with the block confirming that the transaction is registered in the distributed ledger.

1) *Past timing reference:* In this case, we consider the consensus latency in a network, using the initial message as an anchor and computing the time until consensus is achieved. As we discussed above, the definition of consensus can be tricky, and different metrics can be devised depending on the precise objective of the latency.

2) *Future timing reference:* In the same way, the design of the system can be oriented at guaranteeing a maximum latency, using deadlines, and allocating communication resources to the nodes, so as to meet the deadline.

3) *Relative timing reference:* In a distributed ledger, the consensus latency includes both the communication latency and the duration of the consensus mechanism, which can be significant, e.g., in proof-of-work systems, and the confirmation instant can have different definitions. This means that a model of the consensus mechanism must be included in the computation of latency or age, making the timing reference tied to this process.

C. Timing in Distributed Machine Learning

While the rise of cloud computing caused computing to become more centralized in the 2010s, recent years have witnessed the opposite trend of pushing computing to the edge of a local network (i.e., devices and powerful computers near BSs), called mobile edge computing (MEC). Several factors that drive the paradigm shift include the availability of powerful processors for both devices and servers, the emergence of latency-critical applications, the issue of network traffic congestion, and the concerns over
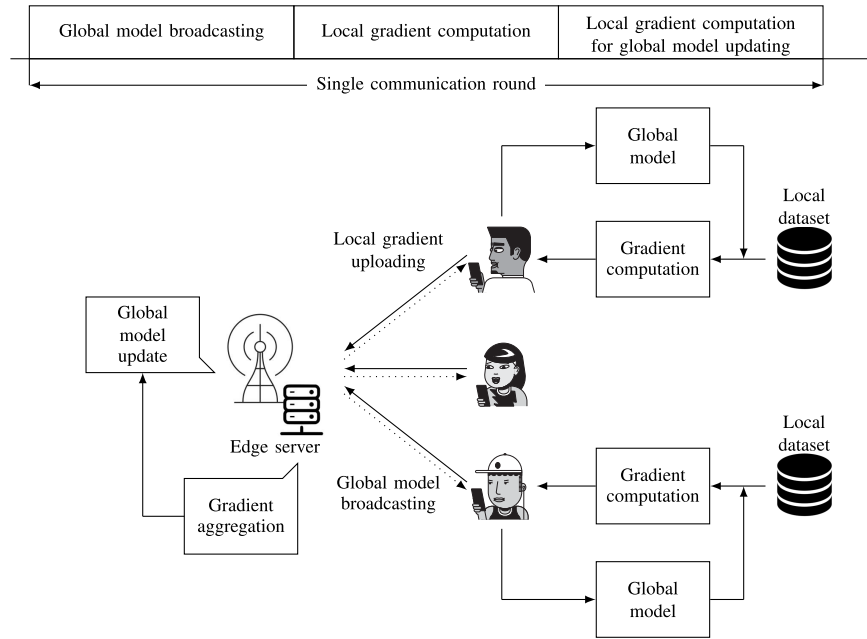
Fig. 11. *Federated learning system and its operations.*

data privacy. Among many others, the training of artificial intelligence (AI) algorithms is an important application of MEC. In this section, we will discuss distributed machine learning (also called edge learning), while its use, called edge inference, is to be discussed in the next subsection.

1) Principle of Distributed Learning: Distributed machine learning refers to the distribution of a learning task over multiple edge devices to leverage either their data or computational resources or both. A learning algorithm usually attempts to minimize (or maximize) some function $L(\mathbf{w})$ of the model-parameter set $\boldsymbol{w}$, referred to as loss function. Thus, a learning task involves finding the optimal model $\boldsymbol{w}^\star$ that solves the optimization problem: $\boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} L(\mathbf{w})$.

There exist various approaches to do this, designed for different purposes. Perhaps, the most popular approach is federated learning (FL) that solves the mentioned optimization problem by implementing the stochastic gradient descent (SGD) algorithm in a distributed manner. Its key feature is the avoidance of direct data uploading, allowing the exploitation of users' data while preserving their privacy. The FL system and operations are illustrated in Fig. 11. The FL iterative algorithms comprise multiple communication rounds. At the beginning of each round, say round n, a server broadcasts the global model to all devices for distributed model/gradient estimation, described as follows. The gradient corresponding to gradient descent on the function is called a *ground-truth gradient*. Each device estimates the ground-truth gradient, $\nabla L(\boldsymbol{w}^{(n)})$, using its local dataset. The result is a *local gradient* that is a noisy version of the ground truth. Upon the completion of local computation, each device uploads its local gradient to the

server. Alternatively, each device updates the downloaded global model by performing multi-round gradient descent locally and then uploads the resulting *local model* to the server. To suppress the estimation noise, the server aggregates (i.e., averages) the local gradients (or local models) and applies the aggregation result to update the global model, completing the round. Let $g(n, \boldsymbol{w})$ be the global (aggregated) gradient in the nth round, which is then applied to updating the global model based on gradient descent

$$\boldsymbol{w}^{(n+1)} = \boldsymbol{w}^{(n)} - \mu g\left(\boldsymbol{w}^{(n)}\right). \tag{8}$$

where μ denotes the step size. In the case in which the distributed data are independent identically distributed (i.i.d.), $\mathsf{E}[g(n, \boldsymbol{w}^{(n)})] = \nabla L(\boldsymbol{w}^{(n)})$, and its variance from the ground truth is inversely proportional to the number of devices K as a result of aggregation. The rounds are repeated till the model converges. The commonly used convergence criteria require that the global gradient is sufficiently small, and the loss function is evaluated to be below a given threshold, corresponding to reaching a target learning accuracy.

Another well-known framework is called parameter server training (PST), which does not aim at leveraging mobile data but instead attempts to harness computation resources distributed at many low-complexity devices for training a large-scale model. To this end, a server partitions the model into parts, called parameter blocks, and allocates each device one block for updating. The PST is based on the classic block coordinate descent algorithm, which is similar to SGD. The PST algorithm is similar to FL, as described earlier, except for two differences. First, the parameter server also downloads a training dataset from

the cloud to devices in the first round. Second, each device is required to update only an assigned parameter block instead of the whole model. This reduces the computation complexity and the communication overhead of individual devices.

2) Statistical Characterization of Timing: It is assumed that the server synchronizes the clocks of devices, and a time-reference point is defined by the server that initializes the learning process. The time spent on learning, called *learning latency*, is measured from the instant when a server initiates the learning process to the instant when the global model converges. The computation capacity of the edge cloud is much larger than that of every single device, and the server's broadcasting latency is much shorter than the uploading latency of individual devices. Therefore, the latency of each (communication) round is limited by the computation-plus-communication latency of devices. The information processing at a device is an example of the modularized architecture in Fig. 9, which cascades the modules of local gradient/model computation, source encoding, and communication. Their efficiencies and latency performance can be improved via joint design. Consider the joint design of computation and source encoding. For instance, the sparsity of a local gradient/model can be exploited for achieving a large compression ratio (e.g., tens to hundreds of times) without significant degradation of learning performance [152]; given its geometry, a stochastic gradient is more suitably compressed using a Grassmannian quantizer instead of one using the mse as the distortion measure [153]. Moreover, the rate information can be fed back from the communication module to control the gradient/model compression ratio. Unlike a point-to-point system, the effect of gradient/model distortion due to a high compression ratio at a particular device can be alleviated by update aggregation over many devices. The total latency of the cascaded on-device modules and those at the server in each communication round gives the per-round latency.

Let K denote the number of devices, $\Delta t_k^{(n)}$ the per-round latency of device k in the nth round, and N the total number of rounds in the learning process. The aggregation operation in the FL and PST learning algorithms introduces an update synchronization constraint: in each round, the server needs to wait for all devices to finish their upload before the global model can be updated. Consequently, the learning latency can be written as

$$\Delta t = \sum_{n=1}^{N} \max \left(\Delta t_1^{(n)}, \Delta t_2^{(n)}, \dots, \Delta t_K^{(n)} \right). \qquad (9)$$

Using the FL system in Fig. 11 as an example, the three types of timing in distributed learning can be characterized as follows.

1) *Past timing reference:* In this case, considering a particular device, the usefulness of a local-model update uploaded by the device depends on how much the current global model differs from the original one downloaded by the device (or, equivalently, its local model), where the downloading instant defines the time anchor. This can be translated into the number of updates to the global model performed by the server in the duration in which the device computes and transmits its local update. The case arises when the system comprises *stragglers*. They refer to those devices that are slowest in computation or communication or both, which results in a latency bottleneck in the learning process. One technique for coping with stragglers, termed *lazy updating,* is for the server to only ask devices with large gradient norms, which indicate significant updates on the model, to upload their local gradients [154]. Another technique, called *synchronous updating,* is to reduce the stragglers' upload frequencies (i.e., uploading only once over multiple rounds) while requiring other devices to perform upload in each round [155]. Such techniques give rise to a tradeoff between per-round latency and the required number of rounds. The lazier the straggler updates are, the lower the per-round latency is. On the other hand, this increases the staleness of the stragglers' updates and, thereby, causes the required number of rounds of the learning process to grow. Therefore, it is necessary to evaluate the staleness of stragglers' updates before they are applied to updating the global model. Moreover, to minimize the learning latency, it is desirable to control the updating frequencies of individual stragglers depending on their computation and channel capacities.

2) *Future timing reference:* It is often necessary to impose a deadline on distributed learning in a mobile network where the connections and donated computation resources of mobile devices, which are the data sources, are transient. For mission-critical applications (e.g., disasters or robotic exploration of an unknown environment), there is usually an urgency in acquiring a trained model. In addition, finite allocated radio resources may also limit the duration of channel use. In such cases, the needed duration for the learning task needs to be estimated. By placing a constraint on per-round latency Δt, the duration is measured by a planned number of rounds, N, which is chosen to ensure the completion of the task with a high probability. To this end, we define the *round-reliability* function as the probability that the model converges as

$$F_C(N) = \Pr(g(N, \boldsymbol{w}) \le \delta \mid W, \mathcal{D}). \qquad (10)$$

Then, given a target probability p_0, the required number of rounds is $N^\star = F_C(p_0)$. While the devices are assumed fixed in our exposition, it should be emphasized that, in practice, they can vary from round to round due to mobility or scheduling. Though
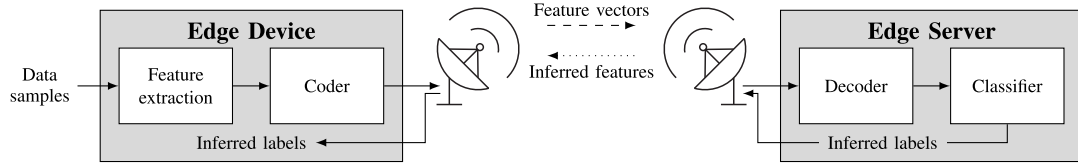
Fig. 12. *Edge inference system and its operations.*

such randomness may incur some learning bias or data loss, there exist rich techniques to alleviate the effects, e.g., probabilistic or data-importance-aware scheduling. There exist several approaches for reducing the learning duration. One is to shorten the per-round communication latency by allocating more radio resources (power and bandwidth) or spatial degrees of freedom at the BS to support space-division multiple access (SDMA) of more devices and enhance the spatial multiplexing of data. Researchers also design new communication techniques targeting federated learning, such as over-the-air aggregation [96], scheduling [97], [156], and radio resource management [157]. Alternatively, devices' computation speeds can be boosted by increasing their processors' clock frequencies at the cost of higher power consumption [98]. It is also possible to reduce the required number of rounds by selecting those devices with important data, i.e., data that are more informative for the model, to participate in model training [99].

3) *Relative timing reference:* Recall that the reference point $t = 0$ refers to the instant the edge server initiates the learning process. The learning latency, denoted as Δt in (9), measures the duration from $t = 0$ until the instant when the convergence criterion is met. In terms of this criterion, extensive research has been conducted on quantifying the convergence speed, measured by the expectation of the averaged global gradient (or loss function) over rounds. The typical form of the speed is given as follows [100], [101]:

$$\mathsf{E}\left[\frac{1}{N}\sum_{n=1}^{N} g(\boldsymbol{w}^{(n)})\right] \leqslant \frac{c_1\left\{L(\boldsymbol{w}_0) - L(\boldsymbol{w}^\star) + \mathsf{E}\left[\frac{c_2}{K}\right]\right\}}{\sqrt{N}}$$

(11)

where K is the number of connected devices, $\boldsymbol{w}_0$ is the initial model, $\boldsymbol{w}^\star$ is the optimal model, the descent step size is chosen as $\mu = (c_3/(N)^{1/2})$, and $\{c_1, c_2, c_3\}$ are constants. Note that K is a random variable due to fading in wireless links, and the constants $\{c_1, c_2\}$ depend on wireless parameters, such as signal-to-noise ratios and outage probabilities [101]. The convergence result allows the estimation of the required number of rounds. The mentioned techniques on reducing per-round latency are also applicable in this case.

D. Timing in Edge Inference

The preceding Section VI-C focuses on the training of machine learning algorithms. The theme of this subsection is their application, called edge inference. Specifically, edge inference refers to making intelligent predictions and decisions at an edge server based on data generated by IoT devices, which finds practically unlimited applications, ranging from smart cities to autonomous driving to smart wearables. Compared with on-device inference, edge inference has the advantages of operating a large-scale AI model (e.g., the Google-Cloud classifier that can discern 700 image classes), enabling centralized decision and control in an IoT scenario with many sensors, and continuous AI model improvements using aggregated data or distributed learning.

To make the discussion concrete, consider the edge learning system in Fig. 12. The information processing in the system is another example of the modularized architecture in Fig. 9, which cascades the on-device modules, including feature extraction (computation), source encoder, and transmitter and server modules, including the receiver, source decoder, and classifier (computation). Their operations, timing, and tradeoffs are discussed as follows. Usually, only a small fraction of information embedded in a high-dimensional sample of a raw-data distribution, denoted as a $N \times 1$ vector **d**, is useful for inference. To reduce the communication overhead, the sample is compressed by projection onto a low-dimensional space that contains the most important information of the data distribution, called a *feature space*. The operation is called *feature extraction*, and the result is called a *feature vector*, denoted as a $M \times 1$ vector **x** with $M \ll N$. A classic technique for feature extraction, called principal component analysis (PCA), computes the feature space as a *linear subspace* [158]. The modern approach applies a neural network, called *encoder*, to identify a *nonlinear* feature space, which yields better inference performance than PCA, at the cost of higher complexity [159]. The feature vector (or part of it) is then transmitted reliably to the server as regular data (i.e., by digital modulation/demodulation and coding/decoding) and fed into another neural network, called decoder, for generating the predicted value. Alternatively, the wireless channel can be treated as a part of the encoder and trained jointly with the decoder to achieve satisfactory inference performance in the presence of channel distortion [160]. Due to imperfect sensing conditions, there is always uncertainty in prediction, which can be

measured using different metrics, such as entropy. Let $U(\mathbf{y})$ denote the uncertainty of prediction on the received feature vector, $\mathbf{y}$, containing a subset of the features in $\mathbf{x}$. $U(\mathbf{y})$ is a monotone decreasing function of the feature subset. As low uncertainty can be usually translated into high prediction accuracy, from the inference perspective, it is desirable for the device to transmit as many features as possible. However, the transmitted features may have to be limited due to a constraint on radio resources or latency. Given this tradeoff, we can discuss timing in edge inference as follows.

1) *Past timing reference:* The basic operation of an IoT system is to aggregate data from a large number of sensors, making inferences and decisions, and then transmit commands to actuators for execution. Many IoT applications, such as vehicle-accident avoidance, crime detection and prevention, and smart manufacturing, are latency-sensitive. Thus, the value of sensing data decreases with their age. On the other hand, the heterogeneity in computation capabilities, locations, link reliability, and bandwidth can cause the data (or features) transmitted by sensors to arrive at servers of different ages. Ensuring the accuracy of inference and quality of decisions requires the server to select uploaded data as inference inputs by considering their staleness. This gives rise to a tradeoff between the age of inference output and data diversity, affecting inference accuracy. For instance, the prediction of a traffic accident is less accurate given sensing data from fewer nearby vehicles, and the recognition of an object/human being is more accurate with multiple camera observations from different perspectives. Such tradeoffs can be optimized by scheduling, radio resource management, and evaluation of the importance of sensing data (e.g., a rare event or a sample in a minority class of imbalanced data).

2) *Future timing reference:* Many timing-sensitive applications, such as autonomous driving and VR, require a device to receive the inference result within a fixed time duration (usually ranging from tens to hundreds of milliseconds) from the instant of sensing. The required fast responses are essential for an auto-pilot to prevent accidents or a VR device to avoid causing dizziness to the user. This limits the cumulative timing for sensing, round-trip communication, and overall computation. Given a deadline or under a constraint on inference accuracy, the latency can be reduced by using a simple technique for on-device feature extraction (e.g., PCA) that is compensated for by deploying a complex high-performance deep neural network for classification at the server, transmitting only the minimum number of features and allocating sufficient radio resources for transmission.

3) *Relative timing reference:* Mobile devices are usually constrained in computation and communication

resources. Given a target inference accuracy, it is desirable to extract and transmit only the minimum number of features. However, the number cannot be estimated in advance due to the geographic separation of data and AI model. The problem can be solved using a progressive feature transmission protocol. The essential idea is to transmit the features block by block until the server confirms that the desired accuracy is met. The communication latency for edge inference on a sample is stochastic, as it depends on the sample and channel realizations and the data distribution, among other factors. Designing the protocol requires the use of an accuracy measure, such as the uncertainty function U discussed earlier, as the uncertainty computation at the server uses a deep neural network model and its feedback to the device. For data containing objects with weak differentiability, reaching the target accuracy may require the transmission of a large number of features or may fail even when all features are transmitted. In such cases, to avoid excessive communication overhead, the device needs to predict and balance the communication cost and uncertainty reduction from additional feature transmission and decides on when to stop transmission. The policy design can be formulated as an optimal stopping problem [161], [162].

VII. CONCLUSION AND FUTURE VISION

Wireless connectivity is the cornerstone of digital technologies that bridge the gap between the physical and digital worlds. Wireless connections offer remote interaction between humans and machines at overextended distances. This calls for careful system optimization to conform to the measurement and perception of time in the physical and digital realm.

5G is a first step toward providing the ultimate connectivity, i.e., a communication system that can flexibly meet any timing requirement. We can think of the three connectivity types (eMBB, mMTC, and URLLC) defined in it as different axes in a wider *service space*, which includes multiple kinds of performance. Services can then be seen as combinations of traffic flows belonging to one of the three services: for example, we can imagine a manufacturing scenario in which sensors and IoT devices gather information about the environment (mMTC), while robots and drones send a video or depth map feed to navigate in the environment (eMBB) and give a central controller information about future actions, which are implemented by reliably transmitting real-time commands (URLLC) to the same robots and drones. In this sense, the approach of 5G technology to satisfy certain end-to-end real-time requirements has been maximalistic: service requirements are defined only in the RAN, with very strict timing requirements (e.g., URLLC allocates resources so that the wireless transmission consumes a very small, predictable part of

the overall timing budget), in order to allow the maximum possible flexibility to other components of the system.

However, as wireless systems evolve toward 6G, the ambition to immerse the digital into the physical reality will increase, and the real-time requirements posed to wireless connectivity will become even more stringent. Recent research has brought a number of other timing measures, such as the AoI or AoL, which are more suitable to characterize the overall real-time operation compared to a mere optimization of a latency parameter. A more general framework of the operations that involve the network is then necessary, as well as a wider view of timing: in this article, we have provided a panoramic view on the field of timing measures and defined a general statistical framework that offers their systematic characterization.

In our vision, the interaction between the digital and physical world is a key component of future networks: as the constraints on timing are often inherently analog, i.e., depend on decisions and events in the physical world, the digital part of the system, consisting of communication, computation, memory, and control, needs to be considered

as a whole, and timing optimization needs to become more flexible than traditional latency minimization. The development of metrics such as AoI and VoI, as well as the integration of applications, such as learning and consensus protocols into the timing characterization, is a step toward a more holistic view of communication, in which the most important data are transmitted at the right time to enable the applications, associating *semantics* to measurements and signaling [36], [134], [135]. This transformation is critical for providing a new type of connectivity, which can go beyond the limited, predefined classes of 5G.

In this work, we have presented the immediate and future usefulness of the proposed framework in different communication models, identified the basic tradeoffs, and established the relation between different types of characterization of timing. The objective of this article is to provide a tutorial view on this emerging area and offer the general statistical framework for timing as a tool for defining and solving problems in real-time wireless communication systems. ∎

REFERENCES

[1] B. Crouch, *Recursion: A Novel*. New York, NY, USA: Macmillan, 2019.

[2] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Feb. 2016.

[3] Y. Alfadhli *et al.*, "Latency performance analysis of low layers function split for URLLC applications in 5G networks," *Comput. Netw.*, vol. 162, Oct. 2019, Art. no. 106865.

[4] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar. 2018.

[5] H. Viswanathan and P. E. Mogensen, "Communications in the 6G era," *IEEE Access*, vol. 8, pp. 57063–57074, 2020.

[6] P. Popovski *et al.*, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Netw.*, vol. 32, no. 2, pp. 16–23, Mar./Apr. 2018.

[7] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[8] B. Hassan, S. Baig, and M. Asif, "Key technologies for ultra-reliable and low-latency communication in 6G," *IEEE Commun. Standards Mag.*, vol. 5, no. 2, pp. 106–113, Jun. 2021.

[9] C. She *et al.*, "A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning," *Proc. IEEE*, vol. 109, no. 3, pp. 204–246, Mar. 2021.

[10] Y. Wei, M. Peng, and Y. Liu, "Intent-based networks for 6G: Insights and challenges," *Digit. Commun. Netw.*, vol. 6, no. 3, pp. 270–280, Aug. 2020.

[11] A. Salh *et al.*, "A survey on deep learning for ultra-reliable and low-latency communications challenges on 6G wireless systems," *IEEE Access*, vol. 9, pp. 55098–55131, 2021.

[12] M. Giordani *et al.*, "Toward 6G networks: Use cases and technologies," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 55–61, Dec. 2020.

[13] F. Chiariotti *et al.*, "Query age of information: Freshness in pull-based communication," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1606–1622, Jan. 2022.

[14] S. Kaul, R. Yates, and M. Gruteser, "Real-time

status: How often should one update?" in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2731–2735.

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

[16] *Service Requirements for Cyber-Physical Control Applications in Vertical Domains*, document TR TS 22.104 V18.3.0, 3GPP, Dec. 2021.

[17] E. Gourgoulhon, *Special Relativity in General Frames: From Particles to Astrophysics*. Berlin, Germany: Springer-Verlag, 2013.

[18] R. Adeogun, G. Berardinelli, P. E. Mogensen, I. Rodriguez, and M. Razzaghpour, "Towards 6G in-X subnetworks with sub-millisecond communication cycles and extreme reliability," *IEEE Access*, vol. 8, pp. 110172–110188, 2020.

[19] C.-L. I and S. Katti, "O-RAN: Towards an open and smart RANs," O-RAN Alliance, Alfter, Germany, White Paper, Oct. 2018.

[20] *Key 5G Use Cases and Requirements*, 5G-ACIA, Frankfurt am Main, Germany, Feb. 2019.

[21] R. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. D. Fiore, P. Thubert, and N. Montavont, "From best effort to deterministic packet delivery for wireless industrial IoT networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4468–4480, Oct. 2018.

[22] A. Roy, T. Acharya, and S. DasBit, "Quality of service in delay tolerant networks: A survey," *Comput. Netw.*, vol. 130, pp. 121–133, Jan. 2018.

[23] *New Services and Capabilities for Network 2030: Description, Technical Gap and Performance Target Analysis*, document TR FG-NET2030-2019-SUB Report, ITU-T Focus Group Technologies, Nov. 2019.

[24] P. Bartolomeu, M. Alam, J. Ferreira, and J. A. Fonseca, "Supporting deterministic wireless communications in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4045–4054, Sep. 2018.

[25] G. Nova *et al.*, "Tactile internet enabled by pervasive networks," Accenture Consulting, Dublin, Republic of Ireland, Res. Rep., Oct. 2018. [Online]. Available: https://www.accenture.com/us-en/insights/communications-media/tactile-internet-pervasive-networks

[26] A. Larranaga, M. C. Lucas-Estañ, I. Martinez,

I. Val, and J. Gozalvez, "Analysis of 5G-TSN integration to support Industry 4.0," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1111–1114.

[27] *System Architecture for the 5G System (5GS)*, document Tech. Rep. TS 23.501 V17.0.0, 3GPP, Mar. 2021.

[28] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, Oct. 2011.

[29] A. Boccolini, A. Fedrizzi, and D. Faccio, "Ghost imaging with the human eye," *Opt. Exp.*, vol. 27, no. 6, pp. 9258–9265, Mar. 2019.

[30] E. Poppel, *Mindworks: Time and Conscious Experience*. New York, NY, USA: Harcourt Brace Jovanovich, 1988.

[31] R. B. Miller, "Response time in man-computer conversational transactions," in *Proc. AFIPS Fall Joint Comput. Conf.*, 1968, pp. 267–277.

[32] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper & Row, 1990.

[33] *State of the Edge 2020 Report*, Linux Foundation, San Francisco, CA, USA, Apr. 2020.

[34] A. Banchs, M. Fiore, A. Garcia-Saavedra, and M. Gramaglia, "Network intelligence in 6G: Challenges and opportunities," in *Proc. 16th ACM Workshop Mobility Evolving Internet Archit.*, Oct. 2021, pp. 7–12.

[35] M. Fidler and A. Rizk, "A guide to the stochastic network calculus," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 92–105, Jul. 2015.

[36] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 96–102, Jun. 2021.

[37] B. Briscoe *et al.*, "Reducing internet latency: A survey of techniques and their merits," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2149–2196, Nov. 2016.

[38] I. Dimitriou and N. Pappas, "Stable throughput and delay analysis of a random access network with queue-aware transmission," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3170–3184, May 2018.

[39] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G:

RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 4th Quart., 2018.

[40] P. Nain, "Analysis of a two node ALOHA network with infinite capacity buffers," in *Proc. IFIP Int. Seminar Comput. Netw. Perform. Eval.*, Sep. 1985, pp. 2.2.1–2.2.16.

[41] A. B. Behroozi-Toosi and R. R. Rao, "Delay upper bounds for a finite user random-access system with bursty arrivals," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 591–596, Mar. 1992.

[42] V. Naware, G. Mergen, and L. Tong, "Stability and delay of finite-user slotted ALOHA with multipacket reception," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2636–2656, Jul. 2005.

[43] L. Georgiadis, L. Merakos, and P. Papantoni-Kazakos, "A method for the delay analysis of random multiple-access algorithms whose delay process is regenerative," *IEEE J. Sel. Areas Commun.*, vol. JSAC-5, no. 6, pp. 1051–1062, Jul. 1987.

[44] C.-S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Trans. Autom. Control*, vol. 39, no. 5, pp. 913–931, May 1994.

[45] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, Jul. 2003.

[46] B. Soret, M. C. Aguayo-Torres, and J. T. Entrambasaguas, "Capacity with explicit delay guarantees for generic sources over correlated Rayleigh channel," *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 1901–1911, Jun. 2010.

[47] B. Soret, M. C. Aguayo-Torres, and J. T. Entrambasaguas, "Capacity with probabilistic delay constraint for voice traffic in a Rayleigh channel," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2009.

[48] N. Pappas, M. Kountouris, A. Ephremides, and A. Traganitis, "Relay-assisted multiple access with full-duplex multi-packet reception," *IEEE Trans. Wireless Commun.*, vol. 14, no. 7, pp. 3544–3558, Jul. 2015.

[49] K. Stamatiou and M. Haenggi, "Delay characterization of multihop transmission in a Poisson field of interference," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1794–1807, Dec. 2014.

[50] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3584–3608, Aug. 2019.

[51] P. P. Bhattacharya and A. Ephremides, "Optimal scheduling with strict deadlines," *IEEE Trans. Autom. Control*, vol. 34, no. 7, pp. 721–728, Jul. 1989.

[52] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in *Proc. 28th Conf. Comput. Commun.*, Apr. 2009, pp. 486–494.

[53] I. H. Hou and P. R. Kumar, "Packets with deadlines: A framework for real-time wireless networks," in *Synthesis Lectures on Communication Networks*. San Rafael, CA, USA: Morgan & Claypool, May 2013.

[54] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55765–55779, 2018.

[55] S. Lashgari and A. S. Avestimehr, "Timely throughput of heterogeneous wireless networks: Fundamental limits and algorithms," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8414–8433, Dec. 2013.

[56] C.-S. Yang, R. Pedarsani, and A. S. Avestimehr, "Timely-throughput optimal coded computing over cloud networks," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jul. 2019, pp. 301–310.

[57] B. Soret, P. Mogensen, K. I. Pedersen, and M. C. Aguayo-Torres, "Fundamental tradeoffs among reliability, latency and throughput in cellular networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2014, pp. 1391–1396.

[58] A. Anand and G. de Veciana, "Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2411–2421, Nov. 2018.

[59] A. Fu, E. Modiano, and J. N. Tsitsiklis, "Optimal transmission scheduling over a fading channel with energy and deadline constraints," *IEEE Trans. Wireless Commun.*, vol. 5, no. 3, pp. 630–641, Mar. 2006.

[60] A. E. Ewaisha and C. Tepedelenlioğlu, "Optimal power control and scheduling for real-time and non-real-time data," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2727–2740, Mar. 2018.

[61] E. Fountoulakis, N. Pappas, Q. Liao, A. Ephremides, and V. Angelakis, "Dynamic power control for packets with deadlines," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018.

[62] M. J. Neely and S. Supittayapornpong, "Dynamic Markov decision policies for delay constrained wireless scheduling," *IEEE Trans. Autom. Control*, vol. 58, no. 8, pp. 1948–1961, Aug. 2013.

[63] C. Tsanikidis and J. Ghaderi, "On the power of randomization for scheduling real-time traffic in wireless networks," in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 59–68.

[64] G. Pocovi *et al.*, "MAC layer enhancements for ultra-reliable low-latency communications in cellular networks," in *Proc. IEEE ICC Workshops*, May 2017, pp. 1005–1010.

[65] B. Soret, A. D. Domenico, S. Bazzi, N. H. Mahmood, and K. I. Pedersen, "Interference coordination for 5G new radio," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 131–137, Jun. 2018.

[66] E. Fountoulakis, N. Pappas, Q. Liao, V. Suryaprakash, and D. Yuan, "An examination of the benefits of scalable TTI for heterogeneous traffic management in 5G networks," in *Proc. 15th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2017.

[67] F. Chiariotti, S. Kucera, A. Zanella, and H. Claussen, "Analysis and design of a latency control protocol for multi-path data delivery with pre-defined QoS guarantees," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1165–1178, Jun. 2019.

[68] F. Chiariotti, A. Zanella, S. Kucera, K. Fahmi, and H. Claussen, "The HOP protocol: Reliable latency-bounded end-to-end multipath communication," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2281–2295, Oct. 2021.

[69] Y. E. Guo, A. Nikravesh, Z. M. Mao, F. Qian, and S. Sen, "Demo: DEMS: DEcoupled multipath scheduler for accelerating multipath transport," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 477–479.

[70] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over HTTP," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 562–585, Aug. 2019.

[71] A. Maatouk, M. Assaad, and A. Ephremides, "On the age of information in a CSMA environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 818–831, Apr. 2020.

[72] R. D. Yates and S. K. Kaul, "Age of information in uncoordinated unslotted updating," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1759–1764.

[73] X. Chen, K. Gatsis, H. Hassani, and S. Bidokhti, "Age of information in random access channels," in *Proc. IEEE ISIT*, Jun. 2020, pp. 1770–1775.

[74] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "The cost of delay in status updates and their value: Non-linear ageing," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4905–4918, Apr. 2020.

[75] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *J. Commun. Netw.*, vol. 21, no. 3, pp. 204–219, Jun. 2019.

[76] R. D. Yates, "The age of information in networks: Moments, distributions, and sampling," *IEEE Trans. Inf. Theory*, vol. 66, no. 9, pp. 5712–5728, Sep. 2020.

[77] B. Zhou, W. Saad, M. Bennis, and P. Popovski, "Risk-aware optimization of age of information in the Internet of Things," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020.

[78] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone, and E. Uysal, "Reliable transmission of short packets through queues and noisy channels under latency and peak-age violation guarantees," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 721–734, Apr. 2019.

[79] F. Chiariotti, O. Vikhrova, B. Soret, and P. Popovski, "Peak age of information distribution for edge computing with wireless links," *IEEE Trans. Commun.*, vol. 69, no. 5, pp. 3176–3191, May 2021.

[80] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "The age of information in a discrete time queue: Stationary distribution and non-linear age mean analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1352–1364, May 2021.

[81] E. Fountoulakis, T. Charalambous, N. Nomikos, A. Ephremides, and N. Pappas, "Information freshness and packet drop rate interplay in a two-user multi-access channel," *J. Commun. Netw.*, vol. 24, no. 3, pp. 357–364, May 2022.

[82] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.

[83] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2215–2228, Oct. 2020.

[84] O. Ayan *et al.*, "Age-of-information vs. value-of-information scheduling for cellular networked control systems," in *Proc. 10th ACM/IEEE Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2019, pp. 109–117.

[85] X. Zheng, S. Zhou, and Z. Niu, "Urgency of information for context-aware timely status updates in remote control systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7237–7250, Nov. 2020.

[86] F. Li, Y. Sang, Z. Liu, B. Li, H. Wu, and B. Ji, "Waiting but not aging: Optimizing information freshness under the pull model," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 465–477, Feb. 2021.

[87] *Study on Scenarios and Requirements for Next Generation Access Technologies*, document TR 38.913 V17.0.0, 3GPP, Mar. 2022.

[88] *Service Requirements for the 5G System*, document TR TS 22.261 V18.2.0, 3GPP, Mar. 2021.

[89] G. Miremadi and J. Torin, "Evaluating processor-behavior and three error-detection mechanisms using physical fault-injection," *IEEE Trans. Rel.*, vol. 44, no. 3, pp. 441–454, Sep. 1995.

[90] L. Kruczynski and S. Teasley, "Time to first fix: A definition and recommended test method," in *Proc. 8th Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GPS)*, Sep. 1995, pp. 1665–1672.

[91] M. Paonni, M. Anghileri, S. Wallner, J.-A. Avila-Rodriguez, and B. Eissfeller, "Performance assessment of GNSS signals in terms of time to first fix for cold, warm and hot start," in *Proc. Int. Tech. Meeting Satell. Division Inst. Navigat. (ION)*, Jan. 2010, pp. 1051–1066.

[92] M. Lévesque and D. Tipper, "A survey of clock synchronization over packet-switched networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2926–2947, 4th Quart., 2016.

[93] *Integration of 5G With Time-Sensitive Networking for Industrial Communications*, 5G-ACIA, Frankfurt am Main, Germany, Feb. 2021.

[94] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network time protocol version 4: Protocol and

algorithms specification," IETF, Fremont, CA, USA, Tech. Rep. RFC 5905, Jun. 2010.

[95] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF, Fremont, CA, USA, Tech. Rep. RFC 3550, Jul. 2003.

[96] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[97] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[98] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, Jun. 2021.

[99] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[100] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proc. 35th Int. Conf. Mach. Learn.*, Jul. 2018, pp. 560–569.

[101] G. Zhu, Y. Du, D. Gunduz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2020.

[102] A. Balakrishnan, "On the problem of time jitter in sampling," *IRE Trans. Inf. Theory*, vol. 8, no. 3, pp. 226–236, Apr. 1962.

[103] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet*. Cham, Switzerland: Springer, 2001.

[104] R. L. Cruz, "A calculus for network delay. I. Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[105] R. L. Cruz, "A calculus for network delay. II. Network analysis," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 132–141, Jan. 1991.

[106] H. Al-Zubaidy, J. Liebeherr, and A. Burchard, "A (min, ×) network calculus for multi-hop fading channels," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2013, pp. 1833–1841.

[107] N. Pappas and M. Kountouris, "Delay violation probability and age of information interplay in the two-user multiple access channel," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019.

[108] M. Kountouris, N. Pappas, and A. Avranas, "QoS provisioning in large wireless networks," in *Proc. 16th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2018.

[109] K. Kaur, F. Guillemin, V. Q. Rodriguez, and F. Sailhan, "Latency and network aware placement for cloud-native 5G/6G services," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2022, pp. 114–119.

[110] Z. Liao *et al.*, "Distributed probabilistic offloading in edge computing for 6G-enabled massive Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5298–5308, Apr. 2020.

[111] R. Fantacci and B. Picano, "End-to-end delay bound for wireless uVR services over 6G terahertz communications," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17090–17099, Dec. 2021.

[112] F. Zhou, L. Feng, P. Yu, W. Li, X. Que, and L. Meng, "DRL-based low-latency content delivery for 6G massive vehicular IoT," *IEEE Internet Things J.*, early access, Mar. 9, 2021, doi: 10.1109/JIOT.2021.3064874.

[113] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69,

no. 10, pp. 12240–12251, Oct. 2020.

[114] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5098–5107, Jul. 2020.

[115] A. Destounis, G. S. Paschos, J. Arnau, and M. Kountouris, "Scheduling URLLC users with reliable latency guarantees," in *Proc. 16th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2018.

[116] R. Kotaba, C. N. Manchon, N. M. K. Pratas, T. Balercia, and P. Popovski, "Improving spectral efficiency in URLLC via NOMA-based retransmissions," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019.

[117] P. Agheli, N. Pappas, and M. Kountouris, "Semantics-aware source coding in status update systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2022.

[118] Z. Chen, N. Pappas, E. Bjornson, and E. G. Larsson, "Optimizing information freshness in a multiple access channel with heterogeneous devices," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 456–470, 2021.

[119] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and scalable distribution of content updates over a mobile social network," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr. 2009, pp. 1422–1430.

[120] J. Selen, Y. Nazarathy, L. L. Andrew, and H. L. Vu, "The age of information in gossip networks," in *Proc. Int. Conf. Anal. Stochastic Modeling Techn. Appl.*, Jul. 2013, pp. 364–379.

[121] R. D. Yates, "The age of gossip in networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 2984–2989.

[122] R. D. Yates, "Timely gossip," in *Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2021, pp. 331–335.

[123] B. Buyukates, M. Bastopcu, and S. Ulukus, "Version age of information in clustered gossip networks," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 1, pp. 85–97, Mar. 2022.

[124] J. Zhong, R. D. Yates, and E. Soljanin, "Two freshness metrics for local cache refresh," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1924–1928.

[125] B. Sliwa, R. Adam, and C. Wietfeld, "Client-based intelligence for resource efficient vehicular big data transfer in future 6G networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5332–5346, Jun. 2021.

[126] X. Zhang, J. Wang, and H. V. Poor, "AoI-driven statistical delay and error-rate bounded QoS provisioning for mURLLC over UAV-multimedia 6G mobile networks using FBC," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3425–3443, Nov. 2021.

[127] Z. Gao, A. Liu, C. Han, and X. Liang, "Non-orthogonal multiple access based average age of information minimization in LEO satellite-terrestrial integrated networks," *IEEE Trans. Green Commun. Netw.*, early access, Mar. 15, 2022, doi: 10.1109/TGCN.2022.3159559.

[128] F. Chiariotti, I. Leyva-Mayorga, C. Stefanović, A. E. Kalor, and P. Popovski, "RAN slicing performance tradeoffs: Timing versus throughput requirements," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 622–640, 2022.

[129] A. Munari, "Modern random access: An age of information perspective on irregular repetition slotted Aloha," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3572–3585, Jun. 2021.

[130] B. Zhou and W. Saad, "Performance analysis of age of information in ultra-dense Internet of Things (IoT) systems with noisy channels," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3493–3507, May 2022.

[131] M. Jia and W. Liang, "Delay-sensitive multiplayer

augmented reality game planning in mobile edge computing," in *Proc. 21st ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Oct. 2018, pp. 147–154.

[132] P. M. de Sant Ana, N. Marchenko, P. Popovski, and B. Soret, "Age of loop for wireless networked control systems optimization," in *Proc. IEEE 32nd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2021.

[133] C. Cichy and S. Rass, "An overview of data quality frameworks," *IEEE Access*, vol. 7, pp. 24634–24648, 2019.

[134] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, "Semantic-effectiveness filtering and control for post-5G wireless connectivity," *J. Indian Inst. Sci.*, vol. 100, no. 2, pp. 435–443, Apr. 2020.

[135] E. Uysal *et al.*, "Semantic communications in networked systems: A data significance perspective," 2021, *arXiv:2103.05391*.

[136] F. Mason, M. Giordani, F. Chiariotti, A. Zanella, and M. Zorzi, "An adaptive broadcasting strategy for efficient dynamic mapping in vehicular networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5605–5620, Aug. 2020.

[137] I. Godor *et al.*, "A look inside 5G standards to support time synchronization for smart manufacturing," *IEEE Commun. Standards Mag.*, vol. 4, no. 3, pp. 14–21, Sep. 2020.

[138] A. Nasrallah *et al.*, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, Sep. 2019.

[139] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Computing TCP's retransmission timer," IETF, Fremont, CA, USA, Tech. Rep. RFC 6298, Jun. 2011.

[140] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jan. 1948.

[141] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[142] L. Gavrilovska, V. Rakovic, and D. Denkovski, "From cloud RAN to open RAN," *Wireless Pers. Commun.*, vol. 113, no. 3, pp. 1–17, Mar. 2020.

[143] J. M. Guerrero, M. Chandorkar, T.-L. Lee, and P. C. Loh, "Advanced control architectures for intelligent microgrids—Part I: Decentralized and hierarchical control," *IEEE Trans. Ind. Electron.*, vol. 60, no. 4, pp. 1254–1262, Apr. 2013.

[144] D. Chowdhury and H. K. Khalil, "Fast consensus in multi-agent systems with star topology using high gain observers," *IEEE Control Syst. Lett.*, vol. 1, no. 1, pp. 188–193, Jul. 2017.

[145] P. Farajiparvar, H. Ying, and A. Pandya, "A brief survey of telerobotic time delay mitigation," *Frontiers Robot. AI*, vol. 7, Dec. 2020, Art. no. 578805.

[146] Q. Shafiee, J. M. Guerrero, and J. C. Vasquez, "Distributed secondary control for islanded microgrids—A novel approach," *IEEE Trans. Power Electron.*, vol. 29, no. 2, pp. 1018–1031, Feb. 2014.

[147] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[148] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4972–4983, Jun. 2016.

[149] P.-H. Kuo, A. Mourad, and J. Ahn, "Potential applicability of distributed ledger to wireless networking technologies," *IEEE Wireless Commun.*, vol. 25, no. 4, pp. 4–6, Aug. 2018.

[150] A. Rovira-Sugranes and A. Razi, "Optimizing the age of information for blockchain technology with applications to IoT sensors," *IEEE Commun. Lett.*, vol. 24, no. 1, pp. 183–187, Jan. 2019.

[151] M. Kim, S. Lee, C. Park, and J. Lee, "Age of
information analysis in hyperledger fabric
blockchain-enabled monitoring networks,"
in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021.

[152] Y. Lin, S. H. Han, H. Mao, Y. Wang, and W. J. Dally,
"Deep gradient compression: Reducing the
communication bandwidth for distributed
training," in *Proc. Int. Conf. Learn. Represent.
(ICLR)*, Feb. 2018, Paper 33.

[153] Y. Du, S. Yang, and K. Huang, "High-dimensional
stochastic gradient quantization for
communication-efficient edge learning," *IEEE
Trans. Signal Process.*, vol. 68, pp. 2128–2142,
2020.

[154] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG:
Lazily aggregated gradient for

communication-efficient distributed learning," in
*Proc. 32nd ACM Int. Conf. Neural Inf. Process. Syst.
(NeurIPS)*, Jan. 2018, Paper 31.

[155] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous
federated optimization," in *Proc. 12th Annu.
Workshop Optim. Mach. Learn. (OPT)*, Jan. 2020,
Paper 28.

[156] J. Song and M. Kountouris, "Wireless distributed
edge learning: How many edge devices do we
need?" *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7,
pp. 2120–2134, Jul. 2020.

[157] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and
S. Cui, "A joint learning and communications
framework for federated learning over wireless
networks," *IEEE Trans. Wireless Commun.*, vol. 20,
no. 1, pp. 269–283, Oct. 2021.

[158] H. Abdi and L. J. Williams, "Principal component
analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*,
vol. 2, no. 4, pp. 433–459, 2010.

[159] J. Shao and J. Zhang, "Communication-
computation trade-off in resource-constrained
edge inference," *IEEE Commun. Mag.*, vol. 58,
no. 12, pp. 20–26, Dec. 2020.

[160] M. Jankowski, D. Gunduz, and K. Mikolajczyk,
"Wireless image retrieval at the edge," *IEEE J. Sel.
Areas Commun.*, vol. 39, no. 1, pp. 89–100,
Jan. 2021.

[161] Y. S. Chow, H. Robbins, and D. Siegmund, *Great
Expectations: Theory of Optimal Stopping*.
Boston, MA, USA: Houghton Mifflin, 1971.

[162] A. N. Shiryaev, *Optimal Stopping Rules*. Cham,
Switzerland: Springer-Verlag, 1978.

ABOUT THE AUTHORS

Petar Popovski (Fellow, IEEE) received the Dipl.Ing. and M.Sc. degrees in communication engineering from the University of Ss. Cyril and Methodius, Skopje, North Macedonia, in 1997 and 2000, respectively, and the Ph.D. degree from Aalborg University, Aalborg, Denmark, in 2005.

He is currently a Professor with Aalborg University, where he heads the Section on Connectivity. He is also a Visiting Excellence Chair with the University of Bremen, Bremen, Germany. He authored the book *Wireless Connectivity: An Intuitive and Fundamental Guide* (Wiley, 2020). His research interests are in the areas of wireless communication and communication theory.

Dr. Popovski is also a member at large of the Board of Governors of the IEEE Communication Society. He received the ERC Consolidator Grant in 2015, the Danish Elite Researcher Award in 2016, the IEEE Fred W. Ellersick Prize in 2016, the IEEE Stephen O. Rice Prize in 2018, the Technical Achievement Award from the IEEE Technical Committee on Smart Grid Communications in 2019, the Danish Telecommunication Prize in 2020, and the Villum Investigator Grant in 2021. He was the General Chair of IEEE SmartGridComm 2018 and the IEEE Communication Theory Workshop 2019. He is also the Vice-Chair of the IEEE Communication Theory Technical Committee and IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING. He is also an Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.

Kaibin Huang (Fellow, IEEE) received the B.Eng. and M.Eng. degrees from the National University of Singapore, Singapore, in 1998 and 2000, respectively, and the Ph.D. degree from The University of Texas at Austin, Austin, TX, USA, in 2008, all in electrical engineering.

He is currently a Professor and an Associate Head of the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, SAR, China.

Dr. Huang received the IEEE Communication Society's 2021 Best Survey Paper, the 2019 Best Tutorial Paper, the 2019 Asia–Pacific Outstanding Paper, the 2015 Asia–Pacific Best Paper Award, and the Best Paper Awards at IEEE GLOBECOM 2006 and IEEE/CIC ICCC 2018. He received the Outstanding Teaching Award from Yonsei University, South Korea, in 2011. He was named a Highly Cited Researcher by Clarivate Analytics from 2019 to 2021 and awarded the Research Fellowship by the Hong Kong Research Grants Council. He has served as the Lead Chair of the Wireless Communications Symposium of IEEE GLOBECOM 2017 and the Communication Theory Symposium of IEEE GLOBECOM 2014 and the TPC Co-Chair for IEEE PIMRC 2017 and IEEE CTW 2023 and 2013. He has served on the Editorial Board of IEEE WIRELESS COMMUNICATIONS LETTERS. He has guest-edited special issues of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, and *IEEE Communications Magazine*. He is also an Executive Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Associate Editor of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and an Area Editor of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING. He is also a Distinguished Lecturer at the IEEE Communications Society and the IEEE Vehicular Technology Society.

Federico Chiariotti (Member, IEEE) received the Ph.D. degree in information engineering from the University of Padua, Padua, Italy, in 2019.

He is currently an Assistant Professor with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark. He has authored over 50 published papers on wireless networks and the use of artificial intelligence techniques to improve their performance. His current research interests include network applications of machine learning, transport layer protocols, age of information, bike-sharing systems, virtual reality, and semantic communications.

Dr. Chiariotti was a recipient of the Best Paper Award at several conferences, including the IEEE INFOCOM 2020 WCNEE Workshop.

Anders E. Kalør (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and the Ph.D. degree in wireless communications from Aalborg University, Aalborg, Denmark, in 2015, 2017, and 2022, respectively.

He was a Visiting Student with Robert Bosch, Stuttgart, Germany, in 2017, and King's College London, London, U.K., in 2020. He is currently a Postdoctoral Researcher with the Connectivity, Aalborg University. His research interests include communication theory, MAC layer design for wireless systems, and networking.

Dr. Kalør was a recipient of an International Postdoc Grant from the Independent Research Fond Denmark in 2021.
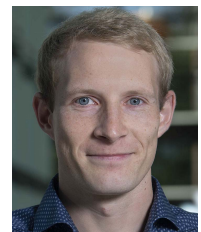
Marios Kountouris (Senior Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2002, and the M.S. and Ph.D. degrees in electrical engineering from Télécom Paris, Paris, France, in 2004 and 2008, respectively.

He has held positions at CentraleSupélec, Gif-sur-Yvette, France; The University of Texas at Austin, Austin, TX, USA; the Huawei Paris Research Center, Boulogne-Billancourt, France; and Yonsei University, Seoul, South Korea. He is currently a Professor and the Chair in advanced wireless systems with the Communication Systems Department, EURECOM, Biot, France.

Dr. Kountouris received several awards and distinctions, including the 2020 IEEE ComSoc Young Author Best Paper Award, the 2016 IEEE ComSoc CTTC Early Achievement Award, the 2013 IEEE ComSoc Outstanding Young Researcher Award for the EMEA Region, the 2012 IEEE SPS Signal Processing Magazine Award, the IEEE SPAWC 2013 Best Paper Award, and the IEEE GLOBECOM 2009 Communication Theory Best Paper Award. He was awarded a Consolidator Grant by the European Research Council (ERC) in 2020 on goal-oriented semantic communication. He has served as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE WIRELESS COMMUNICATIONS LETTERS.

Postdoctoral Researcher with the Department of Telecommunications, CentraleSupélec, Gif-sur-Yvette, France. He is currently an Associate Professor with the Department of Computer and Information Science, Linköping University, Linköping, Sweden. His main research interests include the field of wireless communication networks with an emphasis on semantics-aware communications, energy harvesting networks, network-level cooperation, age of information, and stochastic geometry.

Dr. Pappas has served as the Symposium Co-Chair of the IEEE International Conference on Communications in 2022 and the IEEE Wireless Communications and Networking Conference in 2022. From 2013 to 2018, he was an Editor of the IEEE COMMUNICATIONS LETTERS. He was a Guest Editor of the IEEE INTERNET OF THINGS JOURNAL on "Age of Information and Data Semantics for Sensing, Communication and Control Co-Design in IoT." He is also an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE/KICS JOURNAL OF COMMUNICATIONS AND NETWORKS, and the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY and an Expert Editor for invited papers of the IEEE COMMUNICATIONS LETTERS.

Nikolaos Pappas (Senior Member, IEEE) received the B.Sc. degree in computer science, the B.Sc. degree in mathematics, the M.Sc. degree in computer science, and the Ph.D. degree in computer science from the University of Crete, Rethymno, Greece, in 2005, 2012, 2007, and 2012, respectively.

From 2005 to 2012, he was a Graduate Research Assistant with the Telecommunications and Networks Laboratory, Institute of Computer Science, Foundation for Research and Technology—Hellas, Heraklion, Greece, and a Visiting Scholar with the Institute of Systems Research, University of Maryland at College Park, College Park, MD, USA. From 2012 to 2014, he was a

Beatriz Soret (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications from the University of Málaga, Málaga, Spain, in 2002 and 2010, respectively.

She is currently an Associate Professor with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark, and a Senior Research Fellow with the Communications Engineering Department, University of Málaga. Her research interests include low Earth orbit (LEO) satellite communications, Age of Information (AoI) and semantic communications, and 5G and 6G systems.