A classification of single-channel communication concurrency

Carole Al Mawla* and Eugen Dedu*
*Université Marie et Louis Pasteur, CNRS, institut FEMTO-ST, F-25200 Montbéliard, France
*{carole.almawla, eugen.dedu}@univ-fcomte.fr

***Abstract*—In computer networks, the word "concurrency" is invariably associated with flow-level concurrency, a characteristic of the well-known Internet stack (TCP and IP), where several flows (but not packets) are transferred *at the same time* through the network's links. However, this is not the only form of concurrency provided by networks, especially in wireless Internet of Things (IoT) context. Moreover, this term is vague and has different meanings in different articles. In this paper, we first review the mechanisms that provide concurrency. Afterwards, we identify and detail the various levels of communication concurrency — namely flow, packet, bit, and pulse levels — with examples of network technologies and protocols supporting each of them, and quantify the amount of this concurrency. We focus on concurrency at the same channel, where interference can occur, since using different channels does not create any interference. By clarifying the concept of concurrency, this paper increases the stock of knowledge on communication and provides valuable insights to protocol creators and users.**

***Index Terms*—Wireless network, Internet of Things, Communication concurrency, Communication channel**

I. Introduction

In various communication systems, the concept of concurrency, which enables multiple data transmissions to occur simultaneously, is of paramount importance, especially with the continuous increase of the number of users. Concurrent communication can take two distinct forms: concurrent communication on a single channel, or on several channels. A communication channel refers to a communication medium, such as a wire or a radio channel, through which information is transmitted between devices. In single-channel concurrency, multiple data streams coexist and can interfere within this channel, while in multi-channel data is transmitted simultaneously across multiple independent channels (e.g. different paths or sub-channels, or using Orthogonal Frequency Division Multiplexing, OFDM) and the different flows do not interfere with each other. This paper focuses on single-channel concurrency.

A. Motivations

The concept of *concurrent* data transmission, which describes multiple senders simultaneously transmitting packets, is wide and confusing sometimes. This concept, described by the word "concurrency" and its synonyms (such as "simultaneous" and "parallel") may have *different* meanings across different articles. For example, for a reader familiar with the concurrency term at flow-level, reading about PiP (which uses bit-level concurrency) might be confusing. Some random examples are:

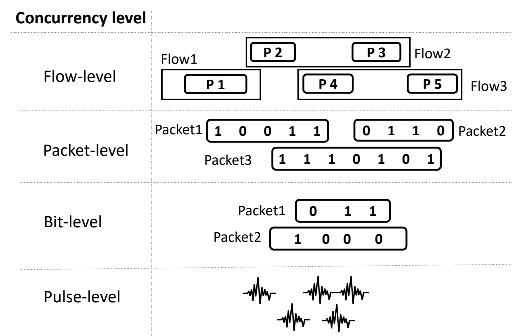

Fig. 1. The various concurrency levels.

- for several *bits* received at the same time: concurrent transmitters" [1], and "PiP exploits packet concatenation to achieve concurrent data collections from multiple neighboring nodes in a single transmission slot" [2];
- for several *packets*, but not bits, received concurrently: "TS-OOK enables robust and concurrent communication among nano-devices" [3];
- for several *pulses* received at the same time: "provide a scalable concurrent ranging solution that can be practically implemented on off-the-shelf UWB devices" [4];
- for several *channels* (or paths): "Multipath TCP provides the ability to simultaneously use multiple paths" [5].

B. Contributions

The goal of our paper is to gather the knowledge of the various meanings of the concept of concurrency, and provide a classification of them into different levels: flow, packet, bit, and pulse levels, as shown in Fig. 1, along with mechanisms enabling them and examples of their use. To the best of our knowledge, this is the only article that provides a classification of concurrency.

This classification considers *where* concurrency appears rather than its *purpose*. As an example, Glossy [6] aims to reconstruct packets, but the concurrency occurs at bit level, hence it will be classified as bit-level (and not packet-level) concurrency.

C. Scope of the article

As it was already stated, this article focuses on single-channel communications. However, it is important to understand that the word channel (or link) has several meanings in the literature. In this article, we use the word *channel* for a medium where, at some point in space and time, only one bit (or one pulse for pulse-based technologies) at a time can travel, and if a second one exists, then it interferes with the first one. The channel can be either dedicated (e.g. a wire) or allocated a frequency range in wireless.

For example, an Ethernet physical cable is a *channel* in the sense of this article, because two bits cannot exist in the cable at the same time without interference. Likewise, 802.11b (Wi-Fi) provides 14 channels, and each of them is a *channel* in the sense of this article (although this is not completely true, as two consecutive channels overlap each other). For LoRa, a *channel* means using the same frequency and the same spreading factor (SF 5..12) [7].

In this article, we are interested in one channel because this is the case where several communications disturb each other, cause interference, and as such lead to lost packets.

D. Related work

[8] provides a survey on multi-channel communication in wireless ad-hoc networks, allowing to alleviate interference and contention through parallel transmissions, with a special attention to channel overlaps. Instead, our work is on single-channel case, where interference does occur. To the best of our knowledge, there is no work classifying concurrency in single-channel case.

II. Classification of concurrent communications

This section presents in detail the classification of concurrency on the same channel and the mechanisms used for each class to achieve this concurrency, followed by a summary and information about the amount of concurrency.

Usually, information (data) is transmitted in a network as a flow, which is divided into packets, each of them consisting of several bits; additionally, for pulse-based wireless technologies, each bit is transmitted at physical layer as one or several pulses, or vice-versa. This can be seen in practice using a traffic analyser application (such as wireshark), which can capture the data from a network card (in this process pulses are replaced by bits) and group together bits in packets, and packets in flows, showing flow concurrency for example. Thus, we classify the concurrency into different levels, covering flow, packet, bit, and pulse-level concurrency.

A. Flow-level concurrency

The concept of *flow* is software-specific. At transport/routing layers, a flow denotes a single, meaningful end-to-end transmission between two peers.

Flow-level concurrency, the most popular form of concurrency, means that multiple flows are multiplexed on the same channel, otherwise said packets from one flow are transported *between* packets of a second flow, as shown at top of Fig. 1. It is essential to understand that while flows are processed *concurrently*, packets are processed *sequentially*, i.e. on any link packets are transmitted sequentially, one *after* the other.

Most of the current widely-used technologies do not support any concurrency by themselves, because, as they use signal carrier waves, two concurrent packets generate collisions. Thus, they only support software-based flow-level concurrency. Examples of such technologies are Ethernet (IEEE 802.3 standard [9]), widely employed in both local and wide-area networks, which has a collision detection mechanism, and Wi-Fi and Bluetooth [10] radio technologies.

However, other technologies and techniques do provide flow-level concurrency (and possibly packet-level concurrency too) natively. For example, Time-Division Multiple Access (TDMA), allows to share one channel among several nodes, however, each node has exclusive access at a given time, and its data *does not collide* with data of other nodes, hence the data flows are transmitted at the same time. Similarly, duty-cycling technologies, such as LoRa, allocate a limited transmission time for each device (e.g. 1% of the time); thus, if a device needs to send more data than it can send in one time slot, it has to cut the data in smaller packets and send each of them between packets from other devices, leading to flow-level concurrency.

B. Packet-level concurrency

Concurrency at packet level appears in a channel having multiple packets at the same time (as shown in Fig. 1), but with no concurrent bits (in the latter case it is classified as bit-level concurrency).

One *mechanism* to achieve this is, for example, through bit interleaving (assuming that there is some time between sending two consecutive bits), as shown in Fig. 1: the three packets are at the same time on the channel, but no two bits are simultaneously found in the channel.

Note that packet-level concurrency results in flow-level concurrency, i.e. a protocol supporting packet-level will naturally support flow-level concurrency too.

Ultra Wide Band (UWB) is a technology based on radio *pulses* (contrary to signal carriers like conventional wireless technologies), appropriate to IoT networks. Unlike traditional systems that modify sinusoidal wave properties (amplitude, frequency, and phase), UWB systems [11] transmit data by emitting radio energy at specific time intervals, on a wide bandwidth (exceeding 500 MHz).

Time Spread On-Off Keying (TS-OOK) [3], [12] is a particular UWB modulation used in terahertz band. Bit 1 is transmitted as a femtosecond-long pulse, whereas bit 0 is “sent” as silence. The time between two consecutive bits sent is very high, much higher than the duration of a pulse (e.g. 1000 times [3]). This time is fixed and known by both sender and receiver, so the receiver can track all the bits of a packet. These high time gaps mean that the bits from several packets can be interleaved, and several packets be received concurrently, as shown in Fig. 2, leading to packet-

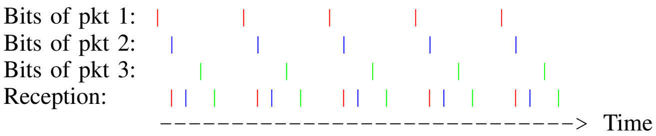

Fig. 2. Packet overlapping by bit interleaving in TS-OOK (each vertical bar is a bit).

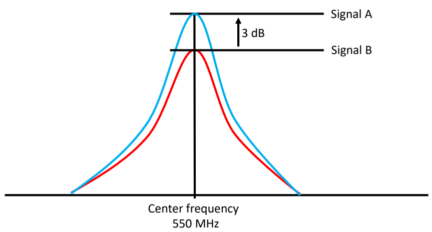

Fig. 3. Capture effect between two signals.

level concurrency (but not bit-level concurrency, as TS-OOK has no provision to decode concurrently-received bits).

The concurrency can be exploited to define the duty-cycling period [13] or the channel occupancy [14] in dense networks.

C. Bit / symbol-level concurrency

Bit-level concurrency appears when several bits are transmitted at the same time in a channel, as shown in Fig. 1. Naturally, this also leads to packet-level and flow-level concurrency.

Some *mechanisms* enabling bit-level concurrency are:

- *Capture effect*, as used by SIC and CT-LoRa for example, refers to a situation where, among multiple signals, only the strongest one can be correctly demodulated by a receiver, and the other signals are ignored. The stronger signal must be greater than the sum of all the other signals, for example by at least 3 dB as shown in Fig. 3.
- *Successive interference cancellation* (SIC) [15], used in SAR for example, is a physical capability allowing a wireless receiver to decode *several* bits that arrive concurrently. To do that, it applies for each bit the capture effect several times, each time extracting the strongest signal, cancelling it, and redoing the capture effect.
- *Constructive interference*, used in Glossy for example, occurs when multiple nodes transmit the *same* packet *simultaneously*, resulting in the constructive addition of their signals at the receiver. Interference is thus as an advantage rather than a problem. Receiver can use it when it can detect the superposition of the base-band signals generated by multiple transmitters of the same signal; if it cannot detect the two signals, *destructive interference* occurs.

Interference generates collisions. In conventional technologies, bit collisions occur (or signal overlap/collide) *in the channel*. However, this is not always the case, for example, in TS-OOK bits are transmitted as pulses (presented in the next section), and consequently, bit collision occurs at *receiver*: two bits sent by two different nodes do not collide in the air, and not at all receiving nodes, but only at nodes which receive them in the same capture window (the receiving time depends on the distance between sender and receiver) [3].

To prevent collisions, it is necessary to avoid multiple nodes transmitting data simultaneously, through techniques like carrier sensing, channel reservations, arbitrated medium access methods like TDMA, and random backoff mechanisms. Alternatively, CDMA (Code-Division Multiple Access) is a technique that uses different spreading codes (each sender uses a different power), so the receiver can extract the collided data (the wider signal can be extracted by order).

Concurrent bits can interfere destructively, leading to packet collisions, hence reducing communication efficiency. It can also interfere constructively, when the colliding bits correspond to the same data. Without constructive interference, even if the two nodes have to send a similar packet to a common receiver, they need coordination to determine which node sends the packet first. The capture effect can also be exploited if the time displacement of these two signals is less than the capture window. Otherwise, the bits are extracted based on constraints such as time displacement.

The above mechanisms might consider the time delta between two bits of packets sent concurrently (for constructive interference), the received signal strength RSS (for capture effect), the radio technology used (modulation and encoding), and the option to send an identical payload or not.

The following examples of protocols use bit-level concurrency.

Glossy [6] is a flooding scheme in WSN which achieves fast network flooding by making nodes transmit simultaneously the *same* packet. These transmissions create constructive interference, and the packets are successfully decoded if the temporal offset between concurrent transmitters is less than 0.5 µs. Nodes have their radios active only at specific scheduled times, during which they listen for communications and relay packets *immediately* after receiving them.

Packet-in-Packet (PiP) [2] is a communication scheme in WSN which exploits both constructive interference and the capture effect. In PiP, a node initiates communication by broadcasting a "beacon" packet and, upon reception, neighboring nodes transmit their own packets concurrently. If synchronisation is accurate, packet contents are identical, and the receiver can correctly receive the packet due to constructive interference. Otherwise, packet contents differ, and the receiver can capture the packet with the highest energy using the capture effect. PiP also introduces the capability of packet concatenation, allowing multiple nodes to transmit simultaneously while dynamically adjusting the power amplifier. The goal is to ensure that the receiver can extract diverse packets from all senders during a single transmission round.

In Disco (deliberate synchronised constructive interference) [16], multiple senders in WSN coordinate to transmit their packets at the same time. Nodes include a chip-level synchronisation algorithm, which enables concurrent transmissions of less than 0.5 µs by compensating for propagation and radio processing delays, and a link selection and alignment algorithm, which determines which co-senders participate in simultaneous transmissions. One of the nodes broadcasts a synchronisation packet, which instructs all the co-senders about the destination and the time of the transmission. Thus, all the co-senders simultaneously transmit the packet, and the receivers use constructive interference to decode the packet.

Multi-hop LoRa networks benefit from the concurrent communication flooding multi-hop protocol (CT-LoRa), which combines the LoRa physical layer with concurrent transmission techniques for rapid dissemination of packets through a network [17]. After the initiator sends the initial packet, each node that receives it for the first time immediately retransmits it as another broadcast, aiming at simultaneous transmission.

Unlike other protocols, mZig protocol [18] enables a ZigBee receiver to decode/decompose multiple concurrent packets from one collision directly (from different transmitters). The smallest unit carrying information (a chip) consists of multiple samples. A chip 1 is represented by a positive half-sine shaping, and a 0 is represented by a negative half-sine. Every four bits are spread to their specified 32 chips. When packets are sent simultaneously, their base-band signals overlap if they have a chip-level time offset. If some collision-free samples (of packet 1), along with the sample shaping, are known, the transmitted chip of the first packet can be extracted; the extracted chip is subtracted from the collided packet to obtain the first collision-free chip of the second packet. Repeating the operations of subtraction and estimation decomposes the collided packet into two packets chip-by-chip. However, if there is no chip-level time offset, the collision-free samples method cannot be used. Therefore, mZig decomposes packets by leveraging the amplitude difference since the amplitude of all chips in one packet is equivalent.

SAR (SIC Aware Routing) protocol [1] uses SIC mechanism (differentiate several concurrent signals at link layer) for packet routing. Routing in SAR is done in three steps: source node initiates route discovery, intermediate nodes update path information while avoiding loops, and destination node selects the optimal path with minimum interference (to optimise spatial resource consumption) and answers with the selected path.

To conclude, several methods are used to achieve bit-level concurrency. Some protocols (like Glossy, PiP, and CT-LoRa) use constructive interference and capture effect, either of the same packet (for Glossy and CT-LoRa), or for different packets (like PiP), possibly with a power amplifier (like PiP). Other protocols (like mZig) use collision-free samples along with the sample shaping.

D. Pulse-level concurrency

Conventional wireless technologies, such as Wi-Fi and LoRa, are based on carrier waves, and information (bits) is encoded by varying the properties of the carrier signal, including amplitude, frequency, or phase. In contrast, in pulse-based communication systems, the fundamental units of information transmission are discrete pulses, which convey data, for example, through the presence or absence of these pulses at specific times. The modulation scheme represents information through analog radio pulses and defines whether a single or multiple pulses are used to represent a single or multiple bits.

Like bit-level, pulse-level concurrency implies packet and flow-level concurrency.

Some *mechanisms* enabling pulse-level communication concurrency are pulse injection (or pulse interleaving) and pulse shaping. In the following, we detail them through the use of Ultra-wideband (UWB), one example of pulse-based technology, where communication and ranging, its two main applications, use concurrent transmission and reception techniques.

1) Concurrency using pulse injection: Pulse injection transmits additional data by exploiting the time displacement of two pulses and injecting other data pulses or extra information in between them.

One method to implement pulse injection is time hopping. Time hopping (TH) allocates pseudo-random transmission times to pulses emitted by distinct users (Time Hopping Impulse Radio, TH-IP) [19], aiming to reduce collisions and thus facilitating a multitude of simultaneous communications. TH UWB system needs accurate timing for the generation of the transmitted sequence and subsequent reception.

TH can be implemented by changing the preamble code in front of each transmitted packet, which defines the pseudo-random sequence used for time-hopping during the transmission of that packet [20]. Preamble codes are thus envisaged as a mechanism to enable multiple non-interfering accesses to the wireless medium.

TH can also be implemented by dividing the time into frames of fixed duration with precisely one pulse transmitted per frame [21]. However, the choice of the frame of each pulse is randomized to avoid catastrophic collisions using a pseudo-random time-hopping sequence (THS).

In practice, the IEEE 802.15.4-compliant DW1000 communication chip [22] achieves non-interfering communication through a different pulse repetition frequency (PRF) with different preamble codes. The PRF signifies the number of pulses sent per second, including the pulse and the pause time. The preamble is part of the packet preceding the actual data, indicating that someone is about to transmit data, and its purpose is synchronization.

Another method to implement pulse injection is provided by Chorus [23]. It is a UWB ranging (localization) scheme in which several localization anchors at known positions send a pulse *at the same time*. The target device, given that it is at different distances from anchors, receives them at *different* times, and computes the distance using the time difference of their arrival time (like in GPS).

2) Concurrency using pulse shape changing: Modifying pulse shape enables the receiver to reliably detect signal peaks associated with different responders, even in case of overlapping responses. For instance, DW1000 radio chip can create several shape widths by widening up the transmitted pulses [4], and receivers can differentiate them.

To sum up, pulse-level concurrency can be achieved either by exploiting the time displacement between two pulses (sent at the same time or at different times by several senders), or by changing the shape of the transmitted pulses.

E. Summary

Table I provides a summary of the communication concurrency levels on the same channel across the technologies and the protocols presented in this section.

We recall that pulse-level concurrency can be achieved by pulse injection or by pulse shape changing; bit-level concurrency refers to two bits that exist on the channel at the same time (combined); packet-level refers to having multiple packets simultaneously in the channel (without bit-level concurrency); and flow-level refers to having multiple flows (but not multiple packets) simultaneously in the same channel.

Pulse-level concurrency exists only in pulse-based technologies (like UWB), and not in carrier-based technologies (like Wi-Fi and Ethernet, where concurrent packets create destructive collisions). Flow-level concurrency is software-based and as such is supported by all the technologies and protocols.

We recall that a concurrency level support automatically enables support for the "right-side" levels too (in the table), i.e. bit-level concurrency implies packet and flow-level concurrency as well, as confirmed in the table.

Note that this table, while being representative of the concurrency levels, is not exhaustive, some network protocols exist which support for example only packet and flow-level concurrency, as shown in the last row of the table (Others).

Different methods can be used to allow concurrency. For instance, signal-based technologies may exploit the capture effect and constructive interference, while pulse-based technologies may use pulse injection or pulse shaping. The capture effect ensures the demodulation of the strongest signal, effectively filtering out interference, while SIC systematically extracts weaker signals from overlapping transmissions. Constructive interference enhances reception when identical signals align (exploit their combined strength). Bit interleaving and pulse injection organize data bits/pulses and inject additional information within signal intervals.

Finally, a fundamental question is whether the *concurrency level* is a characteristic of the *network technology* (hardware, physical layer), or of the *protocol* (software). As shown in the table, the level of concurrency supported depends either on network technology (based on carriers or on pulses) in pulse-level concurrency, or on protocol in pulse, bit, and packet-level concurrency.

F. Amount of concurrency

The ability to achieve concurrent communication is influenced by several parameters of network technologies and protocols, and also by node capability [27]. They also give the *amount* of concurrency, analysed in the following.

Node capability is given by hardware factors, including device transmission, reception, and processing capabilities, and sometimes also by its size, which might prevent complex operations in hardware. For example, nodes in nanonetworks are tiny and thus do not have enough memory to store numerous concurrently received packets: "the receiver can simultaneously track a fixed number of incoming packets, K" [28].

Another parameter impacting the concurrency is the *time*. In TS-OOK, this amount is quantified by a parameter known as β. We recall that in TS-OOK, pulses are sent with a large space between them, and β is the ratio between two consecutive pulses and the length of a pulse. Other pulses can be transmitted in between. The higher β, the higher the number of possible concurrent packets. The value of 1000 is found in the literature ("The ratio between the time between pulses and the pulse duration is kept constant and equal to β=1000" [3]), which means that up to 1000 packets can be transmitted concurrently. Likewise, in UWB, concurrency is achieved through time displacement, and the extent (amount) of concurrency achievable is limited by the temporal spacing between pulses.

Another parameter is the *signal strength*. In SIC [15], the amount of concurrency (number of concurrent bits/packets that can be decoded) is given by the differences in signal strength of the received signals. If one signal is stronger than all the other signals together (the residual), then the strongest signal can be extracted through the capture effect [1], and recursively the same procedure can be applied to the residual, thus extracting the concurrent bits one by one. To conclude, the number of bits that can be decoded concurrently (the amount of concurrency) is given by the differences in signal strengths.

Another parameter is *pulse shape* (which applies to pulse-based technologies only). To achieve concurrency, pulses with different widths can be used. For example, four senders can simultaneously transmit four pulses with sufficiently different widths, and the receiver can differentiate and extract all of them. The number of unique pulse shapes available determines the maximum amount of concurrency.

To conclude, the amount of concurrency of a transmission is influenced by various parameters.

III. Conclusions

This paper provides a comprehensive understanding of concurrency in data communication. It classifies concurrency in different levels, namely pulse, bit, packet, and flow levels. It introduces the mechanisms used to enable concurrency, and gives numerous examples of network technologies and protocols in each level. Carrier-based network technologies support only flow-level, whereas pulse-based technologies support all the levels of concurrency through pulse injection, and pulse shaping (pulse, bit, packet, and flow levels). Network protocols support various levels, using mechanisms like capture effect, constructive interference, pulse injection, and pulse shaping.

Perspectives include analysing the benefits and drawbacks of concurrent communication (e.g. at packet level) compared to sequential communication, and the feasibility of nodes in supporting concurrency based on their characteristics.

TABLE I
CONCURRENCY LEVELS SUPPORTED BY VARIOUS TECHNOLOGIES AND PROTOCOLS.

	Concurrency level			
	Pulse	Bit	Packet	Flow
Network technologies:				
Carrier-based (Ethernet [9], Wi-Fi [24] etc.)	X	X	X	✓
Pulse-based (UWB [25], TS-OOK [3])	✓	✓	✓	✓
Network protocols:				
Glossy [6], PiP [2], CT-LoRa [17], mZig [18], SAR [1], [26]	X	✓	✓	✓
Pulse injection [19], [20], [21], [23] and pulse shape changing [4]	✓	✓	✓	✓
Others	✓/ X	✓/ X	✓/ X	✓

ACKNOWLEDGMENTS

Carole Al Mawla had a grant from Islamic Center Association for Guidance and Higher Education (Lebanon). This work is also funded by Pays de Montbéliard Agglomération (France).

The authors thank Dominique Dhoutaut and Masoud Asghari for their comments on this article.

The authors have not used any AI tool for the work and for the manuscript preparation process.

REFERENCES

[1] R. Liu, M. Sheng, K.-S. Lui, and Y. Shi, "SIC aware high-throughput routing in multihop wireless networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2013, pp. 3229–3233.

[2] P. Zhang, X. Ma, O. Theel, and J. Wei, "Concurrent transmission-based packet concatenation in wireless sensor networks," in *43rd Annual IEEE Conference on Local Computer Networks (LCN)*, Chicago, USA, Oct. 2018, pp. 421–424.

[3] J. M. Jornet and I. F. Akyildiz, "Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1742–1753, May 2014.

[4] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer, "Concurrent ranging with ultra-wideband radios: From experimental evidence to a practical solution," in *38th IEEE International Conference on Distributed Computing Systems (ICDCS)*. Vienna, Austria: IEEE, July 2018, pp. 1460–1467.

[5] C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, "TCP extensions for multipath operation with multiple addresses," IETF standard, Mar. 2020, RFC 8684.

[6] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with Glossy," in *10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. Chicago, IL, USA: IEEE, Apr 2011, pp. 73–84.

[7] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, p. 1466, Sep 2016.

[8] O. Durmaz Incel, "A survey on multi-channel communication in wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 3081–3099, Sep. 2011.

[9] D. Law, D. Dove, J. D'Ambrosia, M. Hajduczenia, M. Laubach, and S. Carlson, "Evolution of ethernet standards in the IEEE 802.3 working group," *IEEE Communications Magazine*, vol. 51, no. 8, pp. 88–96, Aug 2013.

[10] C. Bisdikian, "An overview of the Bluetooth wireless technology," *IEEE Communications Magazine*, vol. 39, no. 12, pp. 86–94, Dec 2001.

[11] M. Ghavami, L. Michael, and R. Kohno, *Ultra wideband signals and systems in communication engineering*. John Wiley & Sons, 2007.

[12] H. Mabed, "Enhanced spread in time on-off keying technique for dense terahertz nanonetworks," in *IEEE Symposium on Computers and Communications (ISCC)*, Heraklion, Greece, Jul. 2017, pp. 710–716.

[13] A. Medlej, E. Dedu, K. Beydoun, and D. Dhoutaut, "Self-configuring asynchronous sleeping in heterogeneous networks," *ITU Journal on Future and Evolving Technologies*, vol. 2, no. 7, pp. 51–62, Oct. 2021.

[14] T. Arrabal, D. Dhoutaut, and E. Dedu, "Efficient multi-hop broadcasting in dense nanonetworks," in *17th IEEE International Symposium on Network Computing and Applications (NCA)*. Cambridge, MA, USA: IEEE, Nov. 2018, pp. 385–393.

[15] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive interference cancellation: Carving out MAC layer opportunities," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 346–357, Feb 2013.

[16] Y. Wang, Y. Liu, Y. He, X.-Y. Li, and D. Cheng, "Disco: Improving packet delivery via deliberate synchronized constructive interference," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 713–723, Mar 2015.

[17] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-hop LoRa networks enabled by concurrent transmission," *IEEE Access*, vol. 5, pp. 21 430–21 446, Sept 2017.

[18] L. Kong and X. Liu, "mZig: Enabling multi-packet reception in ZigBee," in *21st Annual International Conference on mobile computing and networking (ACM MobiCom)*, Paris, France, July 2015, pp. 552–565.

[19] A. F. Molisch, "Ultra-wideband communications: An overview," *URSI Radio Science Bulletin*, vol. 2009, no. 329, pp. 31–42, Jun. 2009.

[20] D. Vecchia, P. Corbalán, T. Istomin, and G. P. Picco, "Playing with fire: Exploring concurrent transmissions in ultra-wideband radios," in *16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. Boston, MA, USA: IEEE, June 2019, pp. 1–9.

[21] J. Colli-Vignarelli, J. Vernez, R. Merz, C. Dehollain, S. Robert, and J.-Y. Le Boudec, "Concurrent transmissions in IR-UWB networks: an experimental validation," in *IEEE International Conference on Ultra-Wideband (ICUWB)*, Vancouver, Canada, Sept 2009, pp. 337–342.

[22] DecaWave Ltd, "DW1000 user manual. v2.18," 2016.

[23] P. Corbalán, G. P. Picco, and S. Palipana, "Chorus: UWB concurrent transmissions for GPS-like passive localization of countless targets," in *18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Montreal, Canada, Apr. 2019, pp. 133–144.

[24] X. Yang, D. Akhmetov, and H.-Y. Liu, "RF concurrency performance study for integrated Wi-Fi radio," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Miami, FL, USA, Dec. 2010, pp. 1–5.

[25] D. Minoli and B. Occhiogrosso, "Ultrawideband (UWB) technology for smart cities IoT applications," in *IEEE International Smart Cities Conference (ISC2)*, Kansas City, MO, USA, Sep. 2018, pp. 1–8.

[26] D. Yuan and M. Hollick, "Let's talk together: Understanding concurrent transmission in wireless sensor networks," in *38th Annual IEEE Conference on Local Computer Networks (LCN)*. Sydney, Australia: IEEE, Oct 2013, pp. 219–227.

[27] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides, "A survey on sensor networks from a multiagent perspective," *The Computer Journal*, vol. 54, no. 3, pp. 455–470, Mar 2011.

[28] J. M. Jornet, J. C. Pujol, and J. S. Pareta, "PHLAME: A Physical Layer Aware MAC protocol for Electromagnetic nanonetworks in the Terahertz band," *Nano Communication Networks*, vol. 3, no. 1, pp. 74–81, Mar. 2012.