COYOTE: Towards Many-to-Many Voice Communication in Bluetooth LE Audio

Theo Gasteiger*, Mahendra Tailor†, Markus Schuß*, Carlo Alberto Boano* and Kay Römer*
*Institute of Technical Informatics, Graz University of Technology, Austria
†Tailored Technology Limited, United Kingdom
Email: {gasteiger, markus.schuss, cboano, roemer}@tugraz.at, mahendra@tailoredwireless.com

***Abstract*—Bluetooth's Broadcast Isochronous Streams (BISes) allow unidirectional audio dissemination to many devices, while Connected Isochronous Streams (CISes) make one-to-one bidirectional audio exchange possible. Yet, neither is suitable for building communication systems in which many devices listening to the same broadcast audio exchange can also answer (i.e., speak back) at the same time. In fact, BISes provide no return path from receivers, and CISes scale only through pairwise connections. To address this gap, we present COYOTE, a mechanism that enables many-to-many communication in Bluetooth LE Audio by repurposing the BISes of a single broadcast isochronous group (BIG). COYOTE keeps the broadcaster fixed and organizes the BIG around a shared audio downlink and one or more receiver-side uplink opportunities for devices listening to the broadcast and willing to answer back. To reduce uplink collisions among simultaneous speakers, COYOTE introduces occupation-aware uplink selection based on feedback embedded in the downlink. We implement COYOTE on custom Bluetooth hardware and demonstrate low audio-path latency, few uplink collisions, and high uplink fairness under repeatable contention. With COYOTE, which we release as open-source, we thus show that Bluetooth's BISes can be repurposed beyond one-to-many dissemination and used as the basis for practical many-to-many voice communication over a shared downlink and multiple receiver-side uplink opportunities.**
***Index Terms*—Bluetooth Low Energy, Digital audio broadcasting.**

I. Introduction

In recent years, several updates to the Bluetooth Core Specification (BC) have significantly extended the capabilities of Bluetooth Low Energy (BLE), not only enhancing performance and communication range through additional physical modes, but also enabling new and previously unattainable applications through novel advertisement schemes such as extended and periodic advertising [1]. Among these updates, BC v5.2 is considered a milestone in Bluetooth history, as it introduced the so-called LE isochronous physical channel and added support for time-bounded data transmission and synchronized rendering across multiple receiver devices [1], [2]. In turn, this addition forms the basis for the new Bluetooth LE Audio standard [2], [3], introducing two distinct traffic patterns: broadcast (connection-less) and unicast (connection-oriented) [1], [2]. The latter is realized through Connected Isochronous Streams (CISes), providing bidirectional point-to-point communication between two devices and supporting interactive use cases such as voice calls [2]. The former is realized through Broadcast Isochronous Streams (BISes), enabling one broadcaster to disseminate audio to countless receiver devices in communication range, making BISes the natural basis for scalable one-to-many audio applications such as assisted hearing systems [2], [4].

Unfortunately, neither BISes nor CISes are well suited for scenarios in which multiple participants need to communicate over a common voice channel. Consider, for instance, production crews, live-event staff, factory-floor personnel, or incident-response teams. In such settings, communication is rarely a set of isolated conversations between pairs of users [5]. Instead, the whole team follows the same ongoing voice exchange, while individual participants speak as soon as the situation requires it. From a user perspective, this pattern is very simple. For Bluetooth LE Audio, however, it exposes a mismatch between the application and the available traffic patterns.

CISes provide bidirectional audio and therefore appear attractive for interactive speech scenarios. However, this bidirectionality is tied to point-to-point connections, so a CIS-based design would inherit the pairwise scaling limitations already examined in MANTIS [6]. BISes solve the opposite part of the problem: one broadcaster can reach many synchronized receivers. Yet, BISes remain unidirectional and provide no return path from receivers to the ongoing exchange. MANTIS addressed this mismatch by showing that BISes can be extended toward multiparty communication through coordinated role transitions between broadcaster and receiver [6]. However, only one device acts as the broadcaster at any given time. MANTIS therefore remains a single-speaker system and cannot support simultaneous speech contributions from multiple participants. Consequently, a different approach is necessary in order to support this communication pattern over Bluetooth LE Audio. Such an approach must preserve the scalability of BISes and avoid the pairwise structure of CISes, while still allowing multiple participants to contribute speech simultaneously as all devices continue to receive the same ongoing audio stream.

Contributions. In this paper, we present COYOTE (Central Orchestrator Yields One mixed Track for Everyone), a mechanism enabling many-to-many communication in Bluetooth LE Audio by repurposing the BISes carried within a Broadcast Isochronous Group (BIG). COYOTE organizes the BIG around a shared downlink audio path that disseminates the mixed audio to all participants, while one or more return paths allow speaking participants to contribute their audio to the mix. This allows several participants to speak at the same time, within the configured uplink capacity, while everyone receives the same shared audio stream. Additionally, we implement COYOTE on custom Bluetooth hardware and experimentally characterize its one-way audio-path latency and contention behavior in a controlled environment. Our results show that COYOTE can turn the inherently one-to-many nature of Bluetooth's BISes into a practical many-to-many communication pattern.

II. LE Audio Foundations

Bluetooth LE Audio introduces a framework for BLE-based audio transmission and sharing, enabling timely data transmission and synchronized rendering across multiple devices, thereby supporting a wide range of novel use cases [1], [2], [4], [7]. At its core lies the LE isochronous physical channel, introduced in BC v5.2, which forms the basis for Bluetooth LE Audio's isochronous streams, namely Broadcast Isochronous Streams (BISes) and Connected Isochronous Streams (CISes) [1]. BISes enable unidirectional, connectionless audio dissemination from an isochronous broadcaster to synchronized receiver devices, making them well suited for scalable one-to-many use cases such as gate announcements at airports or assistive hearing systems in public venues [1]–[4]. CISes, in contrast, enable reliable, bidirectional audio exchange between two connected devices, making them well suited for interactive point-to-point scenarios such as voice calls [1], [2].

A. Broadcast Isochronous Streams

In general, BISes can be characterized as a structured sequence of subevents with strict timing constraints. A Broadcast Isochronous Group (BIG) encapsulates one or more BISes and may additionally include an optional control subevent. As shown in Fig. 1, such a BIG may carry a stereo audio signal, with one BIS representing the left channel and one BIS representing the right channel. In this example, the subevents L_n and R_n constitute the corresponding BIS events. One or more BIS events make up a BIG event, while the optional control subevent (C_n) disseminates control information such as channel-map updates or stream termination information [1]. These BIG events repeat at predefined ISO intervals, thereby forming the recurring timing structure of the contained BISes [1], [2].

Since BISes are connection-less, synchronized receiver devices do not acknowledge individual packets. Instead, BISes rely on robustness mechanisms such as retransmissions, which improve reception probability but consume additional airtime and can, in practice, limit how many BISes fit into one BIG for a given ISO interval, payload size, and PHY configuration [1], [2].

B. Connected Isochronous Streams

CISes are the connection-oriented counterpart to BISes. A Connected Isochronous Group (CIG) encapsulates one or more CISes, with a dedicated Asynchronous Connection-oriented Logical (ACL) connection alongside the isochronous data stream for configuration, control, and coordination [1], [2]. As shown in Fig. 2, a CIG may carry a stereo audio signal, with L_n and R_n representing the left and right channels. Unlike BISes, each CIS event contains scheduled transmission and response opportunities, such as Send L_1 and Recv L_1, enabling bidirectional exchange within the recurring ISO interval. These response opportunities even allow for acknowledgment-based retransmissions and thus are used to improve reliability [1].

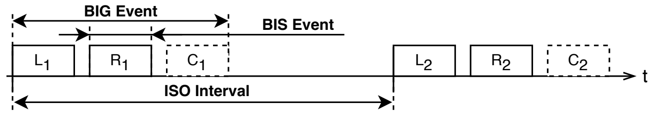

Fig. 1: **BIG of a stereo audio signal containing two BISes representing the left and right audio channels.** Every BIG event contains one or more BIS events (e.g., L_1, R_1) and an optional control subevent C_n, used to disseminate control information such as the stream termination signal.

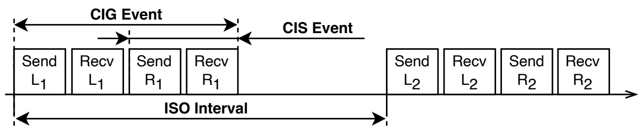

Fig. 2: **CIG of a stereo signal containing two CISes representing the left and right channels.** Every CIG event contains one or more CIS events, which, in turn, consist of one or more transmission and response slots (e.g., Send L_1, Recv L_1).

C. Audio Frames and the LC3 Codec

Since Bluetooth LE Audio carries audio data, sampled input, typically represented as Pulse-Code Modulation (PCM) samples, must be encoded before transmission over an isochronous stream [2], [7]. To this end, the Basic Audio Profile (BAP) [7] defines interoperable audio configurations based on the Low Complexity Communication Codec (LC3), Bluetooth LE Audio's mandatory codec for interoperability [2]. LC3 encodes audio into frames whose sampling rate, frame duration, bitrate, and payload size shape the trade-off between audio quality, latency, and bandwidth [2]. Consecutive LC3 frames are decoded as part of the same logical stream, while packet loss concealment (PLC) masks missing or corrupted frames by generating replacement audio for the affected part of the stream [2].

III. LE Audio and Party-Line Communication

Whether on a factory floor, construction site, film set, or search-and-rescue scene, mobile crews often rely on voice communication over a single shared audio path accessible to all crew members [8]–[10]. These systems, commonly known as party-line intercom systems [11], allow everyone to hear everything, and any participant can speak when needed, with turn taking managed by etiquette rather than explicit control [8], [11]. Typically, crew members initiate a talk by pressing a Push-to-Talk (PTT) button or by using Voice Operated eXchange (VOX), which allows them to speak hands-free by automatically initiating a transmission when speech is detected [11]. Since all communication takes place over the same shared audio path, these systems deliberately trade privacy and structured coordination for situational awareness and operational simplicity, allowing them to just work when it matters most [8].

Despite operating in very different environments, crews relying on this party-line model are remarkably consistent in size. Lean manufacturing work cells typically include 5–15 operators on a shared production line [5], [12], construction trade crews range from 5 to 12 workers [13], live-event intercom channels serve 8–25 crew members such as camera operators, stage managers, and directors [9], [11], and emergency response teams follow the NIMS Incident Command System [14] span-of-control principle of 3–7 subordinates per supervisor, yielding radio talkgroups of 4–8 members. Across these domains, typical group sizes therefore span 4 to 25 participants.

At first glance, Bluetooth LE Audio appears to provide all the necessary building blocks for party-line communication. CISes enable bidirectional point-to-point communication and are, therefore, well suited for voice communication between two connected devices. However, a party-line system solely based on CISes would require each participant to maintain an individual isochronous connection with every other member. Prior work has shown that, in a group of size n, this amounts to $n-1$ CIGs per device, each tied to its own ACL channel and subject to separate scheduling [1], [6]. Consequently, radio and timing demands increase rapidly with the group size and mobility further complicates matters, because every interrupted connection has to be recovered independently, which may temporarily fragment communication among dispersed crews [6].

BISes avoid these scalability limitations, but introduce the opposite problem. A single broadcaster can disseminate audio to an arbitrary number of receiver devices without maintaining per-receiver connections, yet the resulting stream remains unidirectional and provides no return path for receivers wishing to speak. Assigning each participant a dedicated BIG is also impractical, since every device would have to maintain its own BIG while synchronizing to the BIGs of all other members [6]. As shown by MANTIS [6], simultaneous synchronization to multiple BIGs quickly becomes infeasible and is therefore unsuitable for party-line communication. MANTIS addresses this limitation by coordinating role transitions over BISes, allowing participants to take over the broadcaster role while preserving the ongoing isochronous time base, i.e., essentially passing around a token to the device allowed to speak. This enables multiparty communication, but still restricts the system to one active broadcaster at a time and, therefore, cannot support simultaneous speech contributions from multiple participants.

Consequently, supporting party-line communication over Bluetooth LE Audio is not primarily a matter of reserving one transmission opportunity per crew member at all times. Instead, the system must accommodate the number of participants that may try to access the shared audio path within the same conversation window. While group size determines who may receive and participate, the number of concurrent speakers determines how many simultaneous transmission opportunities are needed.

This number can be bounded by looking at how people communicate in shared conversations. Dunbar et al. [15] observed that, once freely-forming groups exceed four individuals, the number of actively involved participants tends to level off at around three to four. This is in line with the conversation-size constraint, according to which everyday conversations become difficult to sustain with more than four speakers and often split once a fifth participant joins [16], [17]. Auditory perception adds another practical constraint, as Cherry's cocktail-party experiments show that following a target speech stream becomes difficult in the presence of competing speech, especially

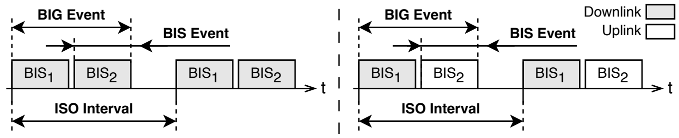

Fig. 3: **Operational overview of COYOTE.** Left: Conventional BIG consisting of two downlink BISes transmitted by the broadcaster. Right: COYOTE preserves the same BIG structure, but repurposes one BIS as an uplink toward the broadcaster.

when clear separation cues are missing [18]. This suggests that, although a party-line system may include many crew members, the number of participants trying to access the shared audio path within the same conversation window remains bounded.

Guided by these constraints, a practical party-line system does not need to scale the number of concurrent transmission opportunities with the total number of crew members. Instead, it should provide a bounded number of transmission opportunities that covers short overlap periods in which several participants start speaking at the same time before turn taking takes over.

IV. COYOTE

In response to the limitations of existing work in §III, we present COYOTE, a mechanism enabling many-to-many communication in Bluetooth LE Audio by repurposing the BISes within a single BIG. COYOTE keeps the isochronous broadcaster fixed and organizes the BIG around a shared downlink audio path and one or more return paths from receiver devices to the broadcaster. Receiver devices wishing to speak neither create their own BIG nor take over the broadcaster role. Instead, they contribute to the ongoing BIG while all devices continue to follow the same shared audio stream. As a result, COYOTE preserves the underlying isochronous time base while allowing receiver devices to contribute speech to the ongoing exchange.

A. Basic Structure

The most simplistic scenario illustrating COYOTE starts from the smallest BIG that can expose both directions of the communication pattern: a BIG with two BISes. In a conventional Bluetooth LE Audio broadcast, an isochronous broadcaster disseminates both BISes to all receiver devices in communication range, as shown in Fig. 3 (left). We refer to this direction, from the broadcaster to the receiver devices, as the downlink. Hence, both BISes in the conventional setup are downlink BISes.

COYOTE preserves this BIG structure, but uses one of the two BISes differently, as shown in Fig. 3 (right). One BIS remains the downlink. The second BIS is repurposed as an uplink, denoting the opposite direction from a receiver device toward the broadcaster. As a result, the broadcaster no longer transmits on both BISes, but transmits on one BIS and receives on the other within the same BIG. COYOTE thereby assigns one scheduled BIS opportunity to the receiver-to-broadcaster direction. Since the BC defines BISes in a BIG as broadcast streams from the isochronous broadcaster to synchronized receivers, this receiver-to-broadcaster assignment is not BC-compliant in its current form [1]. Consequently, devices contributing uplink audio require COYOTE-capable controller support, as the standard Host Controller Interface (HCI) does not expose per-BIS data path direction options within a single BIG. The same principle also extends to larger BIGs (i.e., BIGs with more than two BISes). In such cases, COYOTE can keep one or more BISes in the downlink direction and use the remaining BISes as receiver-side return paths. In the configuration used in this paper, the first BIS remains the shared downlink and all remaining BISes are uplinks. This choice reflects the evaluated configuration rather than a fundamental restriction of COYOTE, since a larger BIG may keep additional BISes in the downlink direction at the expense of reducing the number of receiver-side uplink opportunities. The remaining uplink BISes then define how many receiver-side contributions COYOTE can carry before contention occurs. This is a concurrency budget, not the group size, since additional receiver devices may follow the downlink and contend for an uplink BIS when speaking.

B. Uplink and Downlink Audio Path

Building on the structure introduced in §IV-A, all receiver devices continuously receive the downlink BIS of the ongoing BIG. When a receiver device wishes to speak, it keeps following this downlink and additionally transmits its encoded audio frames on one of the provisioned uplink BISes.

The broadcaster collects these uplink contributions and forms the shared downlink audio. Fig. 4 illustrates this audio path for an example configuration with two speaking receiver devices, two uplink BISes, and one shared downlink BIS. The broadcaster decodes the successfully received uplink frames into PCM samples, combines these samples into a single audio signal, and encodes the resulting mix for transmission on the downlink BIS. Receiver devices therefore do not need to receive, decode, or combine the uplink transmissions of other receiver devices. This keeps the receiver-side audio path simple, which is particularly relevant for resource-constrained devices such as earbuds or hearing aids. More importantly, it preserves one common audio view for the whole group. COYOTE resolves uplink reception at the broadcaster, so successfully received contributions enter the common downlink mix while lost or collided contributions remain absent for all receivers.

The broadcaster is therefore a fixed coordination role and may be implemented as a standalone coordinator without its own microphone or speaker. It may nevertheless include local audio input or output (in the mix) without changing the COYOTE audio path, since receiver devices contribute through uplink BISes and obtain the shared signal through the downlink.

C. Uplink BIS Selection and Occupancy Feedback

When a BIG contains more than one uplink BIS, a receiver device wishing to speak must decide which uplink BIS to use. This decision is non-trivial, since multiple receiver devices may select the same uplink BIS while others remain unused. In such a case, the broadcaster may capture only one colliding transmission (or none), causing other receiver contributions to be lost before they can be mixed into the shared downlink [19]. In principle, uplink selection can be organized at different granularities. An uplink BIS may be assigned to a receiver device ahead of time, selected dynamically when the receiver starts speaking, or changed from one ISO interval to the next. These approaches correspond to per-receiver assignment, per-talk-event selection, and per-ISO-interval selection, respectively.

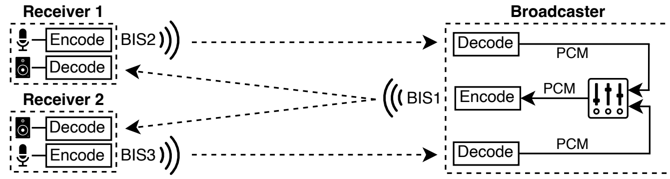

Fig. 4: **COYOTE audio path.** Example 3-BIS configuration with BIS1 as the shared downlink and BIS2 and BIS3 as receiver-side uplinks. The broadcaster decodes successfully received uplink contributions, combines the decoded PCM samples, and encodes the resulting mix for transmission on BIS1.

A per-receiver assignment is simple, but poorly matched to the party-line communication model discussed in §III. Since most participants listen most of the time, this would dimension the BIG for potential rather than active speakers. For larger groups, it either requires one uplink BIS per receiver device or restricts speaking to receivers with a provisioned uplink BIS. As a result, uplink BISes may remain unused while assigned receivers are silent, even though other receivers may wish to speak.

At the other extreme, per-ISO-interval selection would allow a receiver device to change the selected uplink BIS for every encoded audio frame. While this appears flexible, it is poorly matched to encoded audio streams. As discussed in §II, receiver devices transmit LC3 frames, whose packet loss concealment relies on stream continuity. If a receiver changed its uplink BIS within a talk event, consecutive frames of the same speech contribution could appear on different uplink BISes. Preserving codec state would then require the broadcaster to track receiver identity across BIS changes and maintain the corresponding decoder state. Without such tracking, each uplink BIS would be decoded independently, potentially causing audible artifacts whenever a speech contribution switches between uplink BISes.

COYOTE therefore uses per-talk-event selection. When a receiver device starts speaking, for example after a PTT or VOX trigger, it selects one uplink BIS and transmits on this BIS until the talk event ends. This matches the intended communication pattern, where participants contribute short speech bursts, avoids dimensioning the BIG for all potential speakers, and preserves the stream continuity required by the audio decoder.

However, per-talk-event selection still leaves one practical question. Which uplink BIS should a receiver choose when a talk event starts? The simplest COYOTE-compatible option is random selection, which requires no receiver identities, reservations, ownership tracking, or receiver-side monitoring of uplink BISes. Yet random selection can still cause avoidable collisions when two receiver devices select the same uplink BIS while another remains free. More coordinated alternatives, such as listen-before-talk or receiver-specific reservation, could reduce such collisions further, but would require receivers to monitor uplink activity, maintain reservation state, or be addressed individually by the broadcaster. COYOTE therefore focuses on a lightweight broadcast-feedback policy that reduces avoidable collisions while keeping the receiver-side decision simple.

COYOTE shifts the observation point to the broadcaster. Since it observes all provisioned uplink BISes, it can track recent uplink use and report it through the downlink. COYOTE therefore appends a compact 4-byte occupancy descriptor to each downlink payload. The descriptor is a 32-bit bitmap over the provisioned uplink BISes. Bit i corresponds to the i-th uplink BIS and indicates whether the broadcaster successfully received a packet on that BIS during the previous ISO interval. If overlapping uplink transmissions prevent decoding on an uplink BIS, the corresponding bit remains unset. The fixed 4-byte size covers the architectural upper bound of 31 BISes in one BIG.

COYOTE-aware receiver devices remove this descriptor before passing the audio payload to the decoder. The descriptor is appended directly after the encoded LC3 frame bytes and is not part of the BAP-described audio payload. Since the downlink therefore carries COYOTE-specific framing, transparent reception by unmodified Bluetooth LE Audio receivers is implementation-dependent and not assumed.

Receiver devices use the most recent available descriptor to restrict their per-talk-event selection to uplink BISes reported as free. If several uplink BISes are free, a receiver chooses one of them at random. If none are free, the receiver defers its transmission and re-evaluates the next descriptor. As a result, COYOTE reduces avoidable collisions without static BIS assignment or receiver-side monitoring of all uplink BISes.

The descriptor is deliberately advisory. It reports recent occupancy rather than reservations or receiver identities and cannot eliminate contention entirely. Two receiver devices may still start speaking before their transmissions are reflected in the next descriptor and select the same apparently free uplink BIS. In such a case, the uplink BIS does not carry two separate audio streams. The broadcaster can only use the successfully received transmission, while the other contribution is lost before entering the downlink mix. If capture alternates between colliding receivers, the affected uplink stream may lose the continuity expected by the audio decoder and contain audible artifacts.

COYOTE deliberately implements the descriptor as broadcast occupancy feedback rather than as a mechanism that assigns uplink BISes to individual receivers. This choice reflects our goal of reducing avoidable collisions while preserving a simple receiver-side decision that does not require receiver identities, reservations, or per-receiver feedback. More coordinated policies could introduce receiver-specific state, static assignments, or reservations, but would turn advisory feedback into receiver-specific coordination. The broadcaster would then have to track membership, bind uplink observations to specific receivers, handle stale assignments when talk attempts end or packets are lost, and provide receiver-specific feedback so devices know whether they are currently assigned an uplink BIS. Such a design would also require explicit rules for who may reserve uplink resources and under which conditions. COYOTE therefore keeps the descriptor advisory. Receivers can avoid occupied BISes, while residual contention is handled as loss on the affected uplink BIS rather than through per-receiver control.

V. Experimental Evaluation

We evaluate COYOTE on real hardware in a controlled setting with up to seven devices. The goal is to characterize its operating behavior under a bounded number of devices and uplink resources. In particular, we focus on two properties that are essential for the feasibility of COYOTE. First, what one-way audio-path latency is introduced by the uplink, broadcaster-side mixing, and downlink? Second, how much does lightweight occupancy feedback improve access to the provisioned uplink BISes compared with descriptor-free random selection?

A. Experimental Setup

All experiments are conducted on the Bluetooth LE Audio Playground hardware platform v2.1[1] and use nRF Connect v3.1.0. The evaluated COYOTE proof-of-concept implementation is released as open-source[2] and supports this platform as well as other development hardware. Unless stated otherwise, the setup consists of one broadcaster and six receiver devices (covering the four provisioned uplinks and two overload cases), all placed within a 50 cm radius on a desk in a typical office environment. The broadcaster acts as coordinator, mixer, and downlink source, but does not inject local speech into the mix. Hence, all speaking participants in the evaluation are receiver devices and must access the shared audio path through one of the provisioned uplink BISes. This makes the number of uplink BISes (M) the configured upper bound on concurrent receiver-side speakers. The number of receiver devices trying to access the shared audio path (N) is varied per experiment. The relevant factor for these experiments is therefore the number of contending receivers, not the number of passive listeners.

The transmitted audio data adheres to the BAP 16_2_1 set [7], i.e., the LC3 codec at a sampling rate of 16 kHz with a 10 ms frame duration and a bitrate of 32 kbit/s, without configuring an additional presentation delay. With this configuration, each LC3 frame produces a 40 byte encoded audio payload.

The BIG is configured with an ISO interval of 10 ms, matching the LC3 frame boundary, and consists of one shared downlink BIS and four provisioned uplink BISes. We choose four uplink BISes to match the upper end of the expected active-speaker range discussed in §III, while keeping the BIG configuration manageable for the proof-of-concept implementation.

Each experiment is repeated 10 times to account for variability across runs. Each uplink and downlink BIS uses the standard BIS retransmission scheme determined by the BAP 16_2_1 set, i.e., RTN = 2. This distinction matters because receivers select uplink BISes based on the most recent occupancy descriptor they received. If descriptor updates are missed, receivers may act on stale occupancy information, which can increase collision probability and possibly degrade fairness. Under our controlled setup with RTN = 2, such stale-descriptor effects were not visible in the reported experiments, but this observation may not hold under different RF conditions. LC3 and PCM timings

[1]Open-hardware development platform based on the Nordic nRF5340 and Maxim MAX9867 codec: {https://iti.tugraz.at/le-audio-playground}

[2]https://iti.tugraz.at/coyote

are measured by wrapping the corresponding operations with GPIO toggles and recording them with a logic analyzer.

B. One-Way Audio-Path Latency

To quantify the one-way audio-path latency introduced by COYOTE, we use a two-device setup in which the speaking receiver also reproduces the downlink containing its own returned contribution. One device acts as the speaking receiver, provides the audio input, and simultaneously follows the downlink. The second device acts as the broadcaster and performs mixing as well as rebroadcasting. This setup therefore captures the complete implemented audio path from signal injection at the speaking receiver, through uplink transport and broadcaster-side mixing, back to audio output at the same receiver.

We record this path with a black-box setup inspired by the general principle used by NIST for mouth-to-ear latency measurements, where a known source signal is injected at the sender side and the delayed output is recorded at the receiver side [20]. As source signal, we use the English single-talk speech material from ITU-T P.501 Clause C [21] and keep the signal set fixed across all measurements. One experimental run consists of the concatenation of the four audio signals `P501_C_english_{f1,f2,m1,m2}_SWB_48k.wav`, reflecting the speech-oriented nature of COYOTE while remaining reproducible across runs. This concatenated signal is simultaneously captured on the left channel of a Zoom H1n audio recorder as the reference and injected into the line input of the speaking LE Audio Playground receiver. After traversing COYOTE, the downlink signal reproduced by the same receiver is recorded on the right channel. This captures both channels with a common sampling clock and avoids the need for synchronization between separate recordings.

We derive the audio-path delay offline from the two recorded channels using framewise cross-correlation of smoothed amplitude envelopes, following the signal-conditioning principle used by NIST [20]. Using envelopes instead of raw waveforms makes the estimate more robust to codec-induced waveform changes. For each run, we capture the delay across all accepted frames of the signal set. The resulting value is a reproducible estimate of the implemented COYOTE audio path, but not a full mouth-to-ear measurement. It excludes mouth-to-microphone and loudspeaker-to-ear paths, push-to-talk or voice-activity access delay, and standards-conformant measurement equipment such as artificial mouths, ears, or a head-and-torso simulator.

The latency experiment uses the BAP 16_2_1 set introduced in §V-A as the standards-aligned baseline, with one downlink BIS and four provisioned uplink BISes. To further explore COYOTE's low-latency potential, we additionally evaluate a 5 ms variant. This configuration keeps the sampling rate and bitrate unchanged, but replaces standard LC3 with Zephyr's liblc3 implementation [22] in LC3plus mode and configures a matching 5 ms ISO interval. Due to limitations of our COYOTE implementation, this 5 ms variant supports only up to three BISes in one BIG. We therefore configure one downlink BIS and two provisioned uplink BISes for the 5 ms configuration. All remaining parameters follow the setup in §V-A. This allows

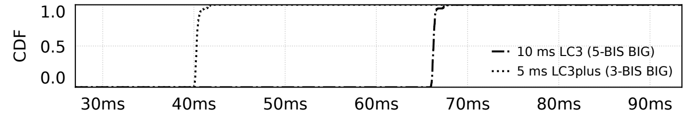

Fig. 5: **One-way audio-path latency.** The 10 ms LC3 baseline has a median of 66.3 ms and a P95 of 66.6 ms, while the 5 ms LC3plus variant has a median of 40.3 ms and a P95 of 41.0 ms.

us to evaluate how much latency can be saved by shortening the audio and transport intervals while leaving the rest unchanged. Fig. 5 illustrates the measured audio-path latency for both configurations. The BAP 16_2_1 baseline achieves a median latency of approximately 66 ms, while 5 ms codec frames with a matching ISO interval lower this value to approximately 40 ms. This reduction is expected, since shorter audio and transport intervals reduce the time spent in the COYOTE signal chain. The improvement, however, comes at a cost. Each 5 ms frame carries only 20 bytes of audio, so the fixed 4-byte occupancy descriptor increases the relative downlink overhead to 20%, compared to 10% in the 10 ms baseline. The 5 ms variant therefore highlights COYOTE's latency potential while exposing the payload-efficiency trade-off introduced by shorter intervals.

To put these values into perspective, we relate them to three latency references as context rather than as direct mouth-to-ear comparisons. 3GPP TS 22.179 specifies that MCPTT systems should deliver 95% of voice bursts with a mouth-to-ear latency below 300 ms [23], while ITU-T G.114 considers one-way delays below 150 ms acceptable for most user applications [24]. A stricter perceptual reference is the delayed-self-voice case, where the downlink mix contains the speaker's own voice and delays around 50 ms are already detectable, but still barely noticeable [25], [26]. Our measurement, however, captures the implemented COYOTE audio path from signal injection to receiver output and excludes acoustic capture, acoustic rendering, and PTT or VOX access delay. The measured 66 ms and 40 ms therefore do not constitute mouth-to-ear latency, but isolate uplink and downlink transport, LC3 codec execution, and broadcaster-side PCM handling. We report operation timings as medians, with P95 values in parentheses. In the 10 ms baseline, LC3 encode, LC3 decode, and broadcaster-side PCM handling take 1365 μs (1458 μs), 990 μs (1089 μs), and 144 μs (145 μs), respectively. In the 5 ms variant, LC3plus encode, LC3plus decode, and PCM handling take 702 μs (721 μs), 365 μs (398 μs), and 61 μs (85 μs), respectively. Each additional uplink mainly adds one broadcaster-side decode operation.

C. Uplink Contention

Contention on COYOTE's uplink path occurs when several receiver devices start talk attempts within a short period of time. Since COYOTE uses per-talk-event selection, each speaking receiver selects one uplink BIS at the beginning of its attempt and keeps it until the attempt ends. COYOTE can therefore influence contention only at initial access. Once two receivers have selected the same uplink BIS, the broadcaster may capture at most one of them [19], while the affected talk attempts continue on the selected BIS. We therefore evaluate initial access to the provisioned uplink BISes, i.e., the moment at which occupation-aware selection can prevent duplicate BIS choices before they persist for the remainder of the talk attempt. Initial-access contention can take two forms. A collision occurs when two or more receivers select the same uplink BIS. Such a collision is avoidable if at least one other uplink BIS is still unused when the receiver starts its talk attempt. An overload occurs when all M uplink BISes are already occupied, leaving at least one receiver without a free uplink opportunity. This distinction matters because COYOTE's occupancy descriptor can reduce avoidable duplicate choices, but cannot create additional uplink capacity once the configured uplink budget is exhausted. To evaluate COYOTE, we compare occupation-aware uplink BIS selection against random selection as the simplest descriptor-free baseline for stateless per-talk-event selection. We fix the uplink configuration at $M = 4$, as motivated in §III, and vary the number of contending receivers $N \in 2, 3, 4, 5, 6$. Values $N \leq M$ remain within the provisioned uplink budget, while $N > M$ exceed this design point and serve as overload stress cases. The remaining configuration follows §V-A.

To create repeatable initial-access contention, we use a Seeed Studio XIAO RP2040 trigger node that drives a wired GPIO pulse to all receiver devices every 5 s. Upon each pulse, every receiver waits for a random offset uniformly drawn from 0 to 300 ms and then starts a talk attempt. This burst model represents situations in which multiple participants react to the same event and initiate PTT or VOX transmissions within a short time interval. It is designed to isolate initial-access contention rather than reproduce complete human conversation dynamics. We choose the 0 to 300 ms window as a simplified contention model grounded in reported turn-taking timings, not as a model of human response behavior. In Stivers et al.'s ten-language sample, language-specific response-offset distributions are unimodal, with most frequent offsets between 0 and 200 ms [27]. Meyer summarizes everyday turn-taking as fast, with median conversational latencies often below 300 ms [28].

A talk attempt represents the period in which a receiver has audio to contribute, independent of whether this audio is eventually delivered. We set this period to 3.4 s, matching the median duration of conversational turns summarized by Meyer [28]. Since this exceeds the 0 to 300 ms start window, uplink BISes selected by earlier receivers remain occupied when later receivers attempt initial access within the same trigger burst. In principle, a receiver that does not obtain an uplink immediately may transmit later once a descriptor reports a free uplink BIS. In this experiment, however, we evaluate only initial access. If all uplink BISes are occupied when the talk attempt starts, we count this as overload and do not model immediate retries.

For each N, we perform 10 independent runs with 100 trigger bursts each. Each run is repeated with occupation-aware and random selection using the same trigger schedule. The random baseline does not use the occupancy descriptor and selects uniformly among the M uplink BISes at talk-attempt start. Occupation-aware receivers use the most recent descriptor and select uniformly among the uplink BISes reported as free. If no uplink BIS is reported as free, the receiver defers transmission.

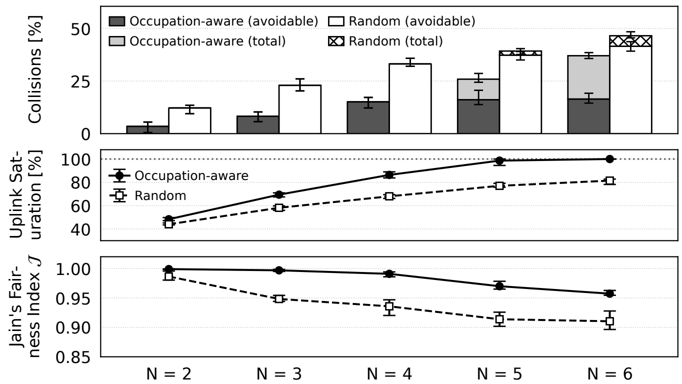

Fig. 6: **Uplink contention** $(\mathbf{M = 4})$**.** Top: how often receiver-side uplink contributions collide, split into total and avoidable collisions. Middle: how much of the uplink budget is used, with the dotted line marking the point at which no unused uplink BIS remains. Bottom: fairness across the received uplinks. Bar heights and markers show the mean across runs, and vertical error bars span the observed minimum and maximum.

Receivers transmit modified uplink payloads in which the encoded LC3 payload is replaced by a receiver identifier, selected uplink BIS, and sequence number, allowing the broadcaster to identify the received sender on each uplink BIS. The modified uplink packets retain the configured 40-byte payload length and fill the remaining bytes with padding. Receiver-side UART logs record the selected uplink BISes and talk-attempts, allowing us to reconstruct initial access and derive the contention plots.

We first measure uplink collisions at talk-attempt start. For each trigger burst, we process receiver talk attempts in the order given by their random offsets. An uplink BIS is considered occupied once an earlier receiver in the same burst has selected it and its talk attempt is still active. We report collisions in two forms. First, we count all collisions, i.e., all cases in which a receiver selects an occupied uplink BIS. Second, we count avoidable collisions, i.e., the subset in which at least one other uplink BIS remains available when the talk attempt starts. This distinction separates collisions that better uplink selection could have prevented from collisions that occur only after the configured uplink budget is exhausted. Since our experiment focuses on initial access, we count duplicate selections independently of whether the broadcaster later decodes one of the colliding packets. This decoding outcome still affects subsequent feedback. If one packet is captured, the next descriptor reports the corresponding uplink BIS as occupied. If no packet is decoded, the descriptor bit remains unset, so later receivers may again treat this BIS as free. This can contribute to residual collisions. However, capture probability depends on RF conditions and relative signal strengths, while our experiment targets initial-access selection under a controlled setup. We therefore do not isolate capture probability as a separate physical-layer metric.

We further measure uplink saturation, which captures how much of the configured uplink budget is occupied during a trigger burst. For each burst, we process receiver talk attempts in offset order and count how many distinct uplink BISes are selected by at least one receiver. Duplicate selections do not increase this count, since they are already captured as avoidable collisions. Deferred talk attempts also do not increase the count, since they do not occupy an uplink BIS. We normalize the result by the configured uplink budget M. Thus, 100% means all configured uplink BISes are occupied, while lower values indicate unused uplink budget. Saturation therefore complements the plot by showing whether collisions occur while unused uplink BISes exist, or only after the configured uplink budget is filled.

As shown in the top of Fig. 6, occupation-aware selection substantially reduces avoidable collisions compared with random selection. Random selection causes avoidable collisions even when $N \leq M$, because two receivers may select the same uplink BIS while another remains unused. Occupation-aware selection reduces these cases by steering later receivers away from BISes reported as occupied. Residual collisions remain possible because the descriptor is delayed broadcast feedback rather than a reservation. Several receivers may act on the same descriptor and select the same apparently free uplink BIS before their transmissions appear in the next descriptor. The gap between all and avoidable collisions appears mainly in the overload stress cases. Once all four uplink BISes are occupied, later receivers may still collide or defer, but such collisions are no longer avoidable because no unused uplink BIS remains.

Finally, we quantify whether uplink contention affects all receivers equally. For this, we count how many uplink packets from each receiver are actually received by the broadcaster during the receiver's talk attempts. Packets that a receiver attempts to transmit but that are not received, for example due to a collision or overload deferral, do not contribute to this count. We then compute Jain's fairness index [29] over the per-receiver received-packet counts. A value of one indicates that all receivers contribute equally to the uplink, whereas lower values indicate that the contributions are unevenly distributed.

As shown in the bottom of Fig. 6, occupation-aware selection maintains $J > 0.95$ across all tested values of N, including the overload stress cases. Random selection, in contrast, degrades to $J \approx 0.91$ at $N = 6$. This higher fairness is explained by fewer duplicate selections. When several receivers collide on the same uplink BIS, the broadcaster can include at most one of them in the received uplink. If capture occurs, only the captured receiver adds to its received-packet count while the other colliding receivers continue their talk attempt without being included. If no packet is decoded, none of the colliding receivers contributes during the affected uplink opportunity. Repeated collisions, especially if capture favors one receiver, therefore make received contributions uneven across receivers.

VI. Discussion and Future Work

Although §V demonstrates COYOTE's feasibility on real hardware, the results are obtained under controlled conditions and therefore do not capture all aspects of real-world deployments.

Controller support and deployability. COYOTE requires vendor-specific Bluetooth controller support to transmit and receive audio within the same BIG, a capability neither exposed by standard HCI interfaces nor supported by off-the-shelf controllers. Our prototype therefore relies on modified controller firmware, but requires no hardware modifications. It is a proof of concept rather than a production-ready implementation. Unmodified Bluetooth LE Audio receivers may decode the downlink depending on how they handle COYOTE-specific framing, but transparent reception is implementation-specific and not assumed. Future listen-only variants could embed the occupancy descriptor in the audio signal, for example through audio watermarking, thereby preserving the BAP-described audio payload size. A production-ready controller implementation and HCI documentation for standardization remain future work.

Uplink BIS dimensioning. In our evaluation, we use four uplink BISes. This exposes non-trivial contention while keeping the BIG configuration manageable, but it does not determine the optimal number of uplinks. Since the evaluated COYOTE configuration uses one BIS as the shared downlink, the BC limit of 31 BISes per BIG leaves at most 30 uplink BISes [1]. This is an architectural bound rather than an implementation guarantee, since the usable number and resulting latency depend on controller resources, on-air timing, PHY, payload size, and retransmissions. How the uplink count may scale with group size and expected simultaneous speakers remains future work.

Descriptor freshness and residual collisions. Occupation-aware uplink BIS selection depends on receivers acting on a sufficiently recent occupancy descriptor. This assumption holds in our controlled experiments, but missed descriptors may make receivers act on stale occupancy information and reduce the benefit over random selection. Since the descriptor is carried in the downlink payload, descriptor loss is coupled to downlink audio loss. Similarly, overlapping uplink transmissions that prevent decoding may cause the next descriptor to report the affected uplink BIS as free. Under repeated loss, occupation-aware selection may approach random selection while the shared audio stream is already impaired. Even with fresh descriptors, collisions cannot be avoided completely, since the descriptor reports only the previous occupancy state and cannot predict that two or more receivers choose the same free BIS at the same time. The descriptor also only tells receivers which BISes appear occupied, not which device is currently received on them. This avoids per-receiver feedback and preserves COYOTE as a broadcast-oriented mechanism, but also means that a receiver cannot know whether its own uplink is actually included in the downlink mix. Consequently, COYOTE does not resolve such collisions during an ongoing talk attempt. Explicit reception feedback could allow receivers to stop, switch to another uplink BIS, or retry after a random delay, but this would require additional coordination and is left to future work.

Latency and access delay. The latency experiment starts at signal injection at the speaking receiver and follows the implemented path through the broadcaster back to the same receiver. It therefore measures COYOTE's returned audio-path latency, but not full mouth-to-ear latency or subjective echo tolerance in a deployed device. Acoustic capture and rendering, sidetone or echo control, and PTT or VOX triggering remain implementation-dependent and may affect perceived usability.

Access delay can further occur when all uplink BISes are already occupied, so a receiver may have to wait or retry before its voice first appears in the shared mix. Quantifying this delay under overload and larger group sizes remains future work.

Codec configuration and audio quality. We evaluate LC3 with 10 ms frames and LC3plus with 5 ms frames, both at 16 kHz and 32 kbit/s. The 5 ms variant uses the reduced three-BIS setup described in §V-B. This captures two latency-oriented configurations, but not the full trade-off between latency, robustness, and audio quality, since additional uplink BISes may further affect latency through increased LC3 and audio operations. Although the 5 ms variant uses shorter frames, it does not shorten the usable audio. At the same 32 kbit/s bitrate, it still carries continuous audio in 20-byte payloads every 5 ms instead of 40-byte every 10 ms. The fixed 4-byte descriptor therefore doubles descriptor traffic and reduces payload efficiency, leaving less room for higher-bitrate audio or additional retransmissions. Perceived audio quality further depends on how simultaneous contributions are mixed and how missing contributions affect the shared audio. The broadcaster currently sums decoded PCM samples with gain normalization. This validates COYOTE and quantifies contention, but does not assess perceived speech quality under overlap or contention.

Traffic model. For contention, we use an RP2040-triggered burst model with fixed 3.4 s talk attempts and random start offsets within a 0 to 300 ms window. This creates repeatable near-simultaneous access attempts and allows both selection strategies to be compared under the same load. However, the model is synthetic and does not capture real group-talk dynamics, where speaking turns, pauses, and overlaps are shaped by user behavior. Replaying real audio traces and varying the trigger window and talk duration remain future work.

Scalability, coexistence, and field validation. Our evaluation uses one broadcaster and up to six contending receivers. This is sufficient to evaluate COYOTE's uplink behavior and overload cases, but does not validate larger deployments with passive listeners, coexisting groups, or mobility, including receiver devices or the fixed broadcaster leaving communication range. Such settings may expose synchronization, descriptor, or coexistence effects not visible in our setup. Evaluating COYOTE at scale and in representative environments remains future work.

Security and talk policy. The prototype assumes cooperative participants and unencrypted BIG operation. It does not enforce who may speak, authenticate speakers, prevent receivers from deliberately occupying uplink BISes, or implement talk policies such as priority or emergency override. In encrypted deployments, the occupancy descriptor would be part of the encrypted payload and removed by COYOTE-aware receivers after decryption. These aspects are outside the evaluated COYOTE mechanism, but would be required for real deployments.

VII. Related Work

At first glance, Bluetooth Mesh appears ideal for party-line communication, as it enables scalable many-to-many communication among provisioned BLE devices [30]. Unfortunately, it is designed for managed networks and asynchronous control traffic, such as smart lighting, building automation, and sensor networks [31]. Hence, it lacks the isochronous timing needed for continuous audio transport and, in turn, conversational interaction, rendering it unsuitable for party-line audio scenarios. TSCH organizes low-power wireless communication through a time-slotted channel-hopping schedule [32]. 6TiSCH [33] integrates TSCH into IPv6-based low-power and lossy networks, and SmarTiSCH [34] adapts TSCH schedules in response to interference. These mechanisms determine how packet transmissions are scheduled or adapted in low-power wireless networks. However, COYOTE operates within Bluetooth LE Audio, where the timing and BIS positions are fixed by the configured BIG rather than dynamically allocated. It repurposes this structure so that one BIS carries the common downlink mix and the remaining BISes provide uplink opportunities for LC3 encoded audio from receiver devices. Occupation-aware selection is limited to choosing one of these uplink BISes when a receiver starts speaking, thereby reducing avoidable duplicate choices while preserving per-talk-event audio-stream continuity and keeping receiver-side behavior simple.

Voice over IP (VoIP) systems enable group voice communication on top of existing network infrastructure. For example, legacy deployments often use the Session Initiation Protocol (SIP) [35] for session signaling and the Real-time Transport Protocol (RTP) [36] for transporting real-time audio streams. Modern systems rely on Web Real-Time Communication (WebRTC) [37], which enables interactive audio scenarios and likewise uses RTP for media transport [38]. Building on these mechanisms, many-to-many audio communication can be realized by conferencing entities such as Multipoint Control Units (MCUs), which mix participant streams into a shared audio signal, or Selective Forwarding Units (SFUs), which forward selected RTP streams to receivers [39]. Push-to-Talk over Cellular follows a similar network-based model, but typically arbitrates floor access rather than supporting simultaneous talkers [40]. In all cases, communication is implemented on top of existing infrastructure rather than in the transmission mechanism itself.

Traditional radio and intercom systems have long supported group voice communication. Commercial party-line intercom systems are widely used by stage crews, broadcast teams, and production staff, where all participants listen to the same ongoing audio exchange [8], [9], [11]. Similarly, Digital Mobile Radio (DMR) [41], General Mobile Radio Service (GMRS) [42], Family Radio Service (FRS) [43], Private Mobile Radio 446 (PMR446) [44], Terrestrial Trunked Radio (TETRA) [45], and Citizen Band (CB) radio [46] provide practical group communication in many operational settings. However, depending on the system and region, they may rely on licenses, dedicated RF spectrum, repeaters or trunked infrastructure, and specialized radios [47]. Hence, these systems demonstrate the maturity of group voice communication as an interaction model, but typically realize it through floor-controlled access rather than by mixing simultaneous contributions from multiple speakers.

3M's Natural Interaction Behavior (NIB) [48] system targets local group voice communication in high-noise environments and enables short-range, connection-less voice communication among nearby participants. NIB supports up to four simultaneous speakers and more than 60 receivers within an approximately 10 m radius, using a low-latency single-hop network with VOX or PTT interaction. However, NIB is not based on Bluetooth LE Audio. It relies on proprietary hardware, operates on region-dependent frequencies, does not encrypt transmissions, and remains tied to dedicated equipment [48].

VIII. Conclusion

In this paper, we present COYOTE, a mechanism that enables many-to-many communication in Bluetooth LE Audio by repurposing the BISes of a single BIG. COYOTE keeps the isochronous broadcaster fixed and organizes the BIG around a shared downlink audio path and one or more receiver-side uplink opportunities for speaking receiver devices. The broadcaster receives these uplink contributions, combines the successfully received ones into a shared audio signal, and rebroadcasts this signal to all participants. To reduce avoidable contention when multiple receivers start speaking at the same time, COYOTE introduces occupation-aware uplink BIS selection using feedback embedded in the downlink. We implement COYOTE on real hardware and show median audio-path latencies of 66.3 ms and 40.3 ms for the evaluated 10 ms LC3 and 5 ms LC3plus configurations, respectively. Our evaluation further shows that occupation-aware uplink BIS selection reduces avoidable collisions within the provisioned uplink budget while maintaining high fairness across the evaluated contention scenarios. While COYOTE requires controller support and cannot eliminate residual contention, it shows that BISes can be repurposed as a basis for many-to-many voice communication in Bluetooth LE Audio.

Acknowledgments

We thank Vinayak Kariappa Chettimada for his contributions to the development of the open-source COYOTE implementation and Nick Hunn for sharing his valuable input on this work.

References

[1] Bluetooth SIG. Bluetooth Core Specification, v6.3. May 2026.
[2] Nick Hunn. *Introducing Bluetooth LE Audio*. 2022.
[3] Kai Ren. 10 Frequently Asked Questions on LE Isochronous Channels. https://bit.ly/40kDDbz. Last access: June 2026.
[4] Andrew Zignani. LE Audio, Auracast Broadcast Audio, and the Future of Bluetooth Audio. https://bit.ly/3lSxIvg. Last access: June 2026.
[5] Sipsas Kostantinos et al. Collaborative Maintenance in flow-line Manufacturing Environments. *Procedia CIRP*, 2016.
[6] Theo Gasteiger et al. MANTIS: Towards Multiparty Communication for Bluetooth LE Audio. In *Proc. of the 22nd WFCS Conf.*, 2026.
[7] Bluetooth SIG. Basic Audio Profile, v1.0.1. June 2022.
[8] Susan Williams. Production Communication. https://bit.ly/412GIj5. Last access: June 2026.
[9] Clear-Com LLC. Live Performance & Events. https://bit.ly/3U2F256. Last access: June 2026.
[10] Savox Communications. Confined space communications. https://bit.ly/4fbZXfy. Last access: June 2026.
[11] Telex Communications, Inc. *Handbook of Intercom Systems Engineering*. Bosch Communications Systems, 2007.
[12] Rachna Shah and Peter T. Ward. Lean manufacturing: context, practice bundles, and performance. *Journal of Operations Management*, 2003.
[13] H. Randolph Thomas et al. Reducing Variability to Improve Performance as a Lean Construction Principle. *Journal of Construction Engineering and Management*, 2002.
[14] Federal Emergency Management Agency. Introduction to the Incident Command System. https://bit.ly/3OWz0Ej. Last access: June 2026.
[15] R. I. M. Dunbar et al. Size and Structure of Freely Forming Conversational Groups. *Human Nature*, 1995.
[16] Jaimie A. Krems et al. Something to talk about: Are conversation sizes constrained by mental modeling abilities? *Evolution and Human Behavior*, 2016.
[17] Jaimie A. Krems and Jason Wilkes. Why are conversations limited to about four people? a theoretical exploration of the conversation size constraint. *Evolution and Human Behavior*, 2019.
[18] E. Colin Cherry. Some Experiments on the Recognition of Speech, with One and with Two Ears. *The Journal of the Acoustical Society of America*, 1953.
[19] Krijn Leentvaar and Jan H. Flint. The capture effect in fm receivers. *IEEE Transactions on Communications*, 1976.
[20] Andrew Catellier and Stephen Voran. NIST IR 8206: Mission Critical Voice Quality of Experience Access Time Measurement Methods, 2018.
[21] International Telecommunication Union. ITU-T P.501. April 2025.
[22] Zephyr Project. liblc3: Low Complexity Communication Codec (LC3). https://github.com/zephyrproject-rtos/liblc3. Last access: June 2026.
[23] ETSI. Mission Critical Push to Talk (MCPTT); Stage 1, TS 122 179 V17.1.0. Apr 2022.
[24] International Telecommunication Union. ITU-T G.114. May. 2003.
[25] Hideki Takaso et al. The Effect of Delayed Auditory Feedback on Activity in the Temporal Lobe While Speaking: A Positron Emission Tomography Study. *Journal of Speech, Language, and Hearing Research*, 2009.
[26] Zarinah K. Agnew et al. Group and individual variability in speech production networks during delayed auditory feedback. *The Journal of the Acoustical Society of America*, 2018.
[27] Tanya Stivers et al. Universals and Cultural Variation in Turn-Taking in Conversation. *Proceedings of the National Academy of Sciences*, 2009.
[28] Antje S. Meyer. Timing in Conversation. *Journal of Cognition*, 2023.
[29] Raj Jain et al. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. Digital Equipment Corporation, 1984.
[30] Bluetooth SIG. Mesh Protocol, v1.1. Sep 2023.
[31] Mohammad Afaneh. Bluetooth Mesh Networking: The Ultimate Guide. https://bit.ly/3RQUcJA. Last access: June 2026.
[32] IEEE Standards Association. IEEE Standard for Low-Rate Wireless Networks, IEEE 802.15.4-2020. July 2020.
[33] Thomas Watteyne et al. Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT). May 2015.
[34] Zihao Yu et al. SmarTiSCH: An Interference-Aware Engine for IEEE 802.15.4e-based Networks. In *Proc. of the 21st IPSN Conf.*, 2022.
[35] Jonathan Rosenberg et al. SIP: Session Initiation Protocol, RFC 3261. June 2002.
[36] Henning Schulzrinne et al. RTP: A Transport Protocol for Real-Time Applications, RFC 3550. July 2003.
[37] World Wide Web Consortium. WebRTC: Real-Time Communication in Browsers. https://www.w3.org/TR/webrtc. Last access: June 2026.
[38] Colin Perkins et al. Media Transport and Use of RTP in WebRTC, RFC 8834. Jan 2021.
[39] Magnus Westerlund and Colin Perkins. RTP Topologies, RFC 7667. Nov 2015.
[40] Hytera. Push-to-Talk Over Cellular (POC). https://bit.ly/3Yr9hoX. Last access: June 2026.
[41] ETSI. Digital Mobile Radio (DMR), TR 102 398 V1.5.1. Nov 2023.
[42] Federal Communications Commission. General Mobile Radio Service (GMRS). https://bit.ly/4k3gxzE. Last access: June 2026.
[43] Federal Communications Commission. Family Radio Service (FRS). https://bit.ly/4m8tt9p. Last access: June 2026.
[44] Ofcom. Analogue and Digital PMR446 Information Sheet. Feb 2018.
[45] ETSI. Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 1: General network design, ETSI EN 300 392-1 V1.4.1. Jan 2009.
[46] Federal Communications Commission. Citizens Band Radio Service (CBRS). https://bit.ly/4d2u1th. Last access: June 2026.
[47] Christopher Bayliss. Scalability and Connectivity Challenges for the Future of Digital Radio Communication. Univ. of Nebraska-Li. 2024.
[48] Minnesota Mining and Manufacturing Company. 3M™ PELTOR™ Natural Interaction Behavior (NIB) Overview. June 2022.