

MANTIS: Towards Multiparty Communication for Bluetooth LE Audio

Theo Gasteiger, Markus Schuß, Carlo Alberto Boano and Kay Römer
Institute of Technical Informatics, Graz University of Technology, Austria
Email: {gasteiger, markus.schuss, cboano, roemer}@tugraz.at

Abstract—Bluetooth’s Broadcast Isochronous Streams (BISes) enable scalable one-to-many audio dissemination, while Connected Isochronous Streams (CISes) support bidirectional communication between two parties via one-to-one connections. Yet neither is suitable for building ad-hoc communication systems in which a multitude of devices can communicate with each other bidirectionally. BISes are inherently unidirectional, and CISes do not scale. We fill this gap by presenting MANTIS, a mechanism that opens a path toward multiparty communication for Bluetooth LE Audio solely using BISes, despite the fact that this is neither supported nor anticipated by the current Bluetooth Core Specification. By introducing the concept of a token and a coordinated token-passing mechanism, MANTIS transforms inherently unidirectional streams into a flexible system where devices can seamlessly alternate between broadcaster and receiver roles, ensuring a single active broadcaster at any time without interrupting the ongoing, underlying isochronous data stream. Additionally, we implement MANTIS on real-world hardware and validate its operation. By bridging the scalability limitations of CISes and the unidirectionality of BISes, MANTIS introduces a new communication paradigm for Bluetooth, enabling ad-hoc, infrastructure-free audio scenarios previously considered impractical or unattainable.
Index Terms—Bluetooth Low Energy, Digital audio broadcasting.

I. INTRODUCTION

The Bluetooth Core Specification (BC) has undergone numerous updates in recent years, significantly enhancing its performance, reliability, and communication range. Among these, the introduction of additional physical layers and novel advertisement schemes, such as periodic advertising, represents a milestone in Bluetooth history [1]. BC v5.2 marks the introduction of a particularly groundbreaking development: the LE isochronous physical channel, often referred to simply as the isochronous channel. This channel enables a wide range of innovative use cases by supporting time-sensitive data transmission and synchronized data rendering across multiple receiver devices [2]. In fact, isochronous channels serve as the foundation for the development of the new LE Audio standard [2], [3]. Compared to Bluetooth Classic Audio, which has been the de facto standard, LE Audio not only improves upon the power efficiency and audio quality, but also introduces two distinct traffic patterns: broadcast (connection-less) and unicast (connection-oriented) [1], [2]. The latter, also known as a Connected Isochronous Stream (CIS), is used for applications in which data is transmitted in a bidirectional manner (e.g., voice data between a smartphone and a pair of earbuds during a phone call) [2]. In contrast, connection-less communication, also known as a Broadcast Isochronous Stream (BIS), is used to disseminate information to countless (potentially unlimited) de-

vices in communication range, enabling the creation of assisted hearing systems in public venues [2], [4]. Unfortunately, neither BISes nor CISes are well-suited for scenarios requiring multiparty communication. Consider, for instance, a live event crew distributed across stage, backstage, and production control. As productions scale, the crew encompasses multiple technical teams (e.g., audio, lighting, video, stage management) that must coordinate in real time, typically over a single shared voice channel on which any member can speak to ensure smooth operation. A similar need arises on the factory floor, where maintenance and incident-response teams require immediate voice coordination that complements existing communication infrastructure [5]. If a Bluetooth LE Audio system were created for such scenarios, neither BISes nor CISes would be suitable due to their inherent limitations. CISes would require dedicated connections between sender and receiver devices, which is not scalable as the group size increases, a limitation we examine in §III. BISes, though scalable, would require one BIS for each potential speaker. Given that BISes are initiated on a per-device basis, it is hardly possible to synchronize all crew members, especially in larger groups, as we demonstrate in §III. Consequently, a different approach is needed in order to support multiparty audio exchange over Bluetooth Low Energy (BLE).

Contributions. In this paper, we identify the limitations of Bluetooth LE Audio for ad-hoc group communication and introduce MANTIS (Multiparty Applicable Notion for Transition of Isochronous Streams), a BC-compliant mechanism that takes a step toward enabling multiparty communication based on Bluetooth’s BISes. MANTIS can be deployed on existing Bluetooth LE Audio hardware to dynamically transition isochronous broadcast roles while preserving the continuity of the underlying isochronous data stream. Based on a decentralized framework that uses the notion of a token, synchronized receiver devices can initiate a role change with an isochronous broadcaster and thereby create the illusion of bidirectional communication across multiple devices using connection-less communication. Real-world experiments show that MANTIS preserves this continuity even under dynamic device movement. As a result, MANTIS helps bridge the gap between the limited scalability of CISes and the lack of bidirectionality in BISes.

II. LE AUDIO FOUNDATIONS

Bluetooth’s LE Audio introduces a new framework for audio transmission and sharing over Bluetooth Low Energy, ultimately moving beyond the codec-centric, point-to-point model

of Bluetooth Classic Audio toward a more flexible, stream-oriented architecture centered on synchronized and scalable audio delivery [1], [2], [6]. At its core lies the LE isochronous physical channel, introduced in BC v5.2, which enables timely data transmission and synchronized rendering across multiple devices, supporting a wide range of novel use cases [2], [4].

Building on this channel, LE Audio transmits data over isochronous streams, which are periodically scheduled and grouped into either Connected Isochronous Groups (CIGs) or Broadcast Isochronous Groups (BIGs), enabling unicast and broadcast communication, respectively [1]. CIGs consist of one or more Connected Isochronous Streams (CISes) and support reliable, bidirectional communication between devices, making them ideal for interactive scenarios such as voice calls [1], [2].

BIGs, in contrast, comprise one or more Broadcast Isochronous Streams (BISes) and enable unidirectional, connection-less audio delivery. This design is particularly well suited for large-scale one-to-many use cases, such as gate announcements at airports or assistive hearing systems in public spaces [2]–[4].

Both CIGs and BIGs maintain a shared timing reference across their streams, enabling synchronized audio playback even when distributing multiple tracks or stereo channels in parallel. For example, true wireless earbuds typically use two CISes, one for the left channel and one for the right, grouped into a single CIG to keep both sides in sync. Similarly, assistive hearing setups may rely on several BISes within one BIG to deliver different languages of the same content simultaneously [2], [4].

A. Broadcast Isochronous Streams

BISes unlock a plethora of novel use cases, ranging from private audio sharing within small groups of devices to audio dissemination in public spaces [2], [7]. Especially real-world applications in public venues often benefit from the imbalance in power constraints, with transmitter nodes frequently accessing mains power, thus circumventing energy limitations, ultimately allowing for higher transmission power and extended communication range [7]. For example, BISes may be used to replace the aging telecoil standard, enabling turn-by-turn navigation for visually-impaired people or to create immersive audio guides for exhibitions or museums [2], [7], [8].

BIS architecture. BISes can be characterized as a structured sequence of subevents with strict timing constraints. Fig. 1 illustrates a typical stereo audio signal composed of left and right channels, represented as two BISes, each containing a single subevent (i.e., L_n and R_n), which also constitutes the BIS event. One or more BIS events make up the so-called BIG event, which may optionally include a control subevent (C) transmitted after the final BIS event. Due to the unidirectional nature of BISes, the control subevent is essential, enabling the transmission of critical metadata such as the channel map or stream termination information [1]. Each BIS event starts at a BIS anchor point, serving as the timing reference for subsequent BIS events. These repeat at predefined ISO intervals, collectively forming the isochronous data stream [1], [2].

To ensure reliable transmission in the absence of acknowledgments, BISes incorporate robustness mechanisms such as chan-

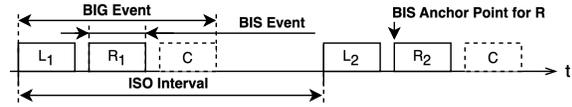


Fig. 1: **BIG containing two BISes representing the left and right channel of a stereo audio signal.** Every BIG event contains one or more BIS events (e.g., L_1 , R_1) as well as an optional control subevent C , used to disseminate control related information such as the termination of the ongoing stream.

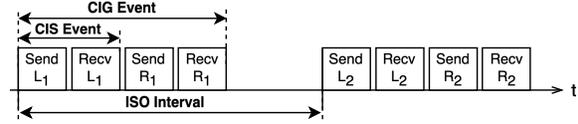


Fig. 2: **CIG containing two CISes representing left and right channel of a stereo audio signal.** Every CIG event, contains one or more CIS events, in turn comprised of one or more transmission and response slots (e.g., Send L_1 , Recv L_1).

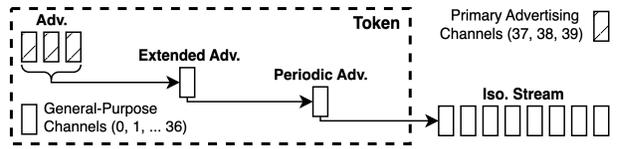


Fig. 3: **Isochronous stream establishment procedure.** Receiver devices synchronize to an ongoing periodic advertisement train using extended advertisements referenced in BLE advertisements. Each periodic packet includes the timing of the next BIG/CIG event, which comprises one or more BIS or CIS events and is used to synchronize to the isochronous stream.

nel blacklisting and retransmissions [1]. The former disables poor-performing channels via control subevents, while the latter increases delivery probability by transmitting the same packet multiple times. BISes may even employ pre-transmissions, allowing early transmission of subevents belonging to future BIS events to increase transmission scheme diversity and improve the overall robustness of the isochronous stream [2].

B. Connected Isochronous Streams

In contrast to BISes, CISes support point-to-point communication, making them a natural fit for use cases such as voice calls and real-time audio streaming between devices [2], [9]. While BISes rely on proactive retransmissions and pre-transmissions for robustness, CISes take advantage of their bidirectional nature (i.e., utilizing acknowledgments) to ensure reliable exchange, making them particularly effective in interactive low-power applications. Moreover, CISes also employ a so-called Asynchronous Connection-oriented Logical (ACL) channel, a dedicated BLE connection alongside the actual isochronous data stream, used to configure, control, and coordinate the CISes [1], [2]. For instance, the ACL channel may handle operations such as volume adjustment or playback control [2].

CIS architecture. Similar to BISes, CISes are described as periodic sequence of subevents. However, each CIS event includes both a transmission and a reception slot, enabling bidirectional exchange within every interval. Fig. 2 illustrates two CISes (i.e., L_n , R_n), each consisting of send and receive subevents (e.g.,

Send L_1 , Recv L_1). In turn, these CIS events act as timing anchors and repeat at a fixed interval, known as the ISO interval. To ensure reliable operation under strict timing constraints, CISes employ a so-called flush timeout [1]. This parameter defines the maximum number of ISO intervals a PDU may persist before being discarded [1], [2]. For example, even if a sender is allowed up to three retransmissions, it will stop immediately upon receiving an acknowledgment. Conversely, if no acknowledgment is received after the allowed attempts, the PDU is dropped to maintain the overall timing guarantees [1].

C. Finding Isochronous Streams

In general, the use of BISes as well as CISes necessitate their discovery by receiving a series of advertisement packets. Fig. 3 illustrates this mechanism by highlighting the chain of packets (i.e., advertising packets, followed by extended advertising and periodic advertising) a receiver device must follow in order to be able to synchronize to an ongoing BIS or CIS. Each layer of the chain adds more information, gradually allowing receiver devices to tune into the actual isochronous data stream [2].

III. LE AUDIO AND PARTY-LINE COMMUNICATION

Whether on a film set, festival stage, sports sideline, or search-and-rescue scene, mobile crews often rely on party-line [10] intercom systems [11]–[13]. Even on factory floors, where breakdown handling and maintenance demand immediate response, teams rely on radios for voice communication among them [5]. In steel manufacturing, workers likewise report that communication is needed for safety actions to be effective [14]. To support this style of coordination, party-line systems operate over a single shared audio path accessible to all crew members. Everyone hears everything, and any user can speak by pressing a Push-to-Talk (PTT) button, with turn taking managed by etiquette rather than explicit control [10], [11]. As an alternative, some systems even support VOX (Voice Operated eXchange), allowing users to speak hands-free by initiating a transmission automatically when speech is detected. Despite offering neither privacy nor structured coordination, party-line systems remain a staple in crew communication, valued for their operational simplicity and ability to just work when it matters most [11].

Although Bluetooth LE Audio appears, at first glance, to provide all the components necessary for such a communication system, its current traffic patterns do not match party-line practice. For example, a seemingly elegant solution based on LE Audio is to use CIS-based communication. As shown in Fig. 4, each crew member in a group of size n would establish a bidirectional link to every other member, requiring one CIG per crew pair. This results in a total of $n - 1$ CIGs per person, each maintained over a separate ACL channel. This may be feasible for very small groups and can offer strong interactivity through persistent pairwise links. However, party-line operation primarily relies on a shared audio path with rapid turn taking, rather than simultaneous full-duplex speech among all participants. As the group grows, the number of connections increases rapidly, and each must be scheduled individually. This places increasing demands on radio and timing resources, ultimately



Fig. 4: **Crew communication system using CISes.** All crew members can bidirectionally communicate with each other as long as everyone maintains a direct connection with all others.



Fig. 5: **Unidirectional crew communication system using BISes.** A single crew member, acting as the isochronous broadcaster, disseminates data to all devices in communication range. Individual members are unable to communicate with each other.



Fig. 6: **Bidirectional crew communication system using BISes.** All crew members create individual BISes (i.e., one BIG per crew member containing one or more BISes), while simultaneously synchronizing to the BISes of all other members.

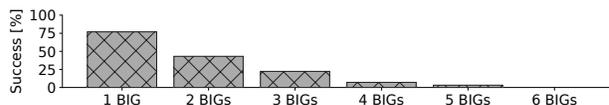


Fig. 7: **Feasibility of simultaneous BIG reception using the communication setup in Fig. 6.** Success rate declines rapidly as the number of BIGs to synchronize to increases.

limiting how many crew members can participate concurrently. Mobility further complicates matters. When a connection drops due to range or interference, links must be re-established one by one, often requiring proximity between affected members, which can be challenging in dispersed crews. This interrupts communication and can fragment it until all links are recovered. A unidirectional alternative solely based on BISes avoids these scaling issues by allowing one crew member to broadcast audio to all others. As illustrated in Fig. 5, this setup imposes minimal coordination overhead and works well for simple downlink communication. However, it lacks a return path, so no other member can respond, making interactive scenarios unfeasible. One could extend this unidirectional approach by letting a device wishing to speak negotiate with the current speaker to switch roles, so that the ongoing BIG is terminated and a new BIG is created by the device taking over. This restores bidirectionality, but requires all devices to re-initiate the isochronous stream establishment procedure (Fig. 3) at every speaker change before audio can be received again. As re-synchronization may not complete in time (e.g., if the new speaker starts transmitting immediately), parts of the audio may be lost, ultimately rendering this approach impractical.

Another alternative is to let each crew member maintain its own BIG, as illustrated in Fig. 6. In this setup, every member maintains its own BIG while synchronizing to the BIGs of all others. While this avoids repeated re-establishment when speakers change, it lacks centralized coordination and leads to stream collisions and inconsistent reception as the group grows, given that each device transmits on its own schedule. To highlight these limitations, we performed an experiment using the setup outlined in §V. In each run, each device maintains its own BIG and synchronizes to a given number of other BIGs. For each target number of simultaneously synchronized BIGs,

we performed 1000 independent runs and recorded the fraction in which all BIGs were successfully synchronized. As shown in Fig. 7, the success rate deteriorates rapidly beyond a single BIG, underscoring the impracticality of this design.

IV. MANTIS

In response to the real-world challenges identified in §III, we present MANTIS, a novel mechanism enabling bidirectional audio and voice communication across a theoretically unbounded number of devices using Bluetooth’s BISes. MANTIS introduces a decentralized token-passing protocol enabling seamless role transitions with minimal overhead, transforming inherently unidirectional isochronous streams into a flexible system supporting scalable, coordinated group communication.

A. Operational Overview

The most simplistic scenario showcasing a potential application of the MANTIS mechanism is a setup consisting of two devices (i.e., node A and node B), depicted in Fig. 8. Assuming node A (acting as a customary isochronous broadcaster using a BIS) initiates a unidirectional data stream by disseminating data to all devices in communication range, node B (being a traditional synchronized receiver) can then tune in to the BIS and start receiving the broadcasted data. So far, this scenario represents unidirectional communication (e.g., broadcast audio) using BISes. Given the unidirectional nature of BISes, node B is only able to listen to the data broadcasted by node A. Node B is neither able to stop node A from broadcasting nor is able to take over the ongoing BIS to communicate with node A.

MANTIS solves this problem by allowing nodes A and B to switch roles while adhering (i.e., staying synchronized) to the same underlying BIS. As illustrated in Fig. 9, after transitioning roles, node A is not able to broadcast data anymore, given that it will have switched into reception state (i.e., acting as the synchronized receiver), whereas node B will have switched from reception to transmission (i.e., taking over the isochronous broadcaster role), while preserving the continuity of the underlying, ongoing data stream, initially started by node A.

Additionally, MANTIS allows devices synchronized to the ongoing isochronous stream but outside the isochronous broadcaster’s communication range (e.g., due to a role transition, synchronized nodes may move out of the broadcaster’s range) to request the stream takeover and gain the ability to disseminate data. Fig. 10 exemplifies this using announcements in the form of advertisement packets disseminated by all devices currently in communication range of the isochronous broadcaster A (i.e., node B). These announcements then enable devices outside node A’s range (i.e., node C) to request a takeover (e.g., via scan requests), with node B acting as an intermediary that transitions the isochronous broadcaster and synchronized receiver roles.

B. Token-Based Role Coordination

The ability of MANTIS to seamlessly transition the isochronous broadcaster role, whether between neighboring devices or via an intermediary device, is enabled by token-based coordination, in which a token explicitly identifies the specific device authorized to act as the isochronous broadcaster.



Fig. 8: **Unidirectional communication based on BISes.** One device (i.e., A) broadcasts data while all other devices in communication range (i.e., B), are able to synchronize to it.

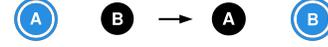


Fig. 9: **Role transition utilizing MANTIS.** MANTIS allows the seamless transition of the isochronous broadcaster role, i.e., switching the transmitter role while adhering to the stringent timing requirements imposed by the ongoing, underlying BIS.

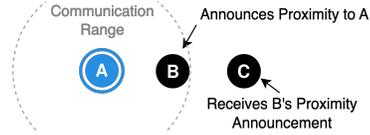


Fig. 10: **Extended role transition utilizing MANTIS.** MANTIS enables devices synchronized to the ongoing isochronous data stream but outside the isochronous broadcaster’s communication range (i.e., node C outside of node A’s vicinity) to request a role transition using scan requests and proximity announcements via an intermediary device (node B).

As outlined in §II, in order for ordinary synchronized receiver devices (i.e., devices not employing MANTIS) to receive BISes, they must first receive an advertisement packet, followed by an extended advertisement packet as well as a periodic advertisement packet. In the context of MANTIS, these three mandatory layers, illustrated in Fig. 3, collectively form the so-called token, which identifies a Broadcast Isochronous Group (BIG). Devices holding such a token are broadcast devices and therefore the only ones authorized to disseminate data via the BISes within that BIG. It is important to highlight that exactly one token (corresponding to a particular BIG) must exist within a two-hop neighborhood¹ to ensure coordinated access to the BISes and prevent reception conflicts. Accordingly, MANTIS coordinates access within two hops (i.e., among direct neighbors and their neighbors). Different teams can still operate in the same area by using different BIGs, since MANTIS only coordinates devices that refer to the same BIG. As the number of co-located independent teams increases, performance is primarily affected by additional on-air traffic, which may reduce synchronization reliability and increase the setup time, even though teams do not synchronize to each other’s BIGs.

Tokens may be dynamically transferred between devices synchronized to the same BIG. For example, the token, held by node A in Fig. 10 may be transferred to node B or node C with node B acting as an intermediary device. This transfer process ultimately enables the seamless transition of broadcasting role without interrupting the ongoing isochronous data stream.

Given that the transition of the broadcasting role induced by the transition of the token inherently also transitions the three advertisement layers, used for the BIS establishment procedure,

¹A hop refers to the theoretical link between nodes that can directly communicate with each other (i.e., without any intermediary nodes; also referred to as a one-hop neighborhood). A two-hop neighborhood extends the notion of a one-hop neighborhood by additionally including intermediary devices.

the node holding the token also acts as the entry point for new nodes wishing to synchronize to the ongoing BIS. This is crucial in order to preserve compatibility with the broadcast assistant role defined in the basic audio profile [6], which forms the basis for many LE Audio broadcast audio applications.

C. Token Management

Prior token passing schemes typically arbitrate packet transmissions on a shared medium [15], [16]. In MANTIS, the token is bound to the advertisements used to synchronize to an isochronous stream [2], thereby transferring synchronization context and enabling coordinated role transitions. Accordingly, token management is crucial for continuity and reliability.

Token announcement. In order for synchronized devices (i.e., devices tuned into the ongoing BIS) outside of the isochronous broadcaster's (i.e., token holder's) communication range to request a stream takeover, devices in vicinity of the token holder (i.e., intermediary devices part of the one-hop neighborhood) must announce the presence of a token in the form of, for example, BLE advertisements. This procedure ensures that devices part of the two-hop neighborhood can request the token in case they wish to take over, utilizing the intermediary devices, the ongoing isochronous stream. Moreover, it enables devices which temporarily moved out of the token holders' one-hop neighborhood and, therefore, possibly lost track of the isochronous time base, to detect the ongoing stream again.

Token creation. Any device wishing to start the dissemination of data (i.e., not the token holder) shall be able to create a token, ultimately representing the isochronous broadcaster. In order to do so, devices must follow the token creation procedure.

Token creation procedure. Devices wishing to create a new token periodically (e.g., periods of scan interval length) check (e.g., by scanning for corresponding BLE advertisement packets) if other devices in vicinity already advertise a token. If no token is detected after a certain, developer-configurable, time delay has passed, a random number $r \in [0, n_{rand}]$ is generated, where n_{rand} is developer-configurable and defines the random back-off range for token creation. If the generated number is zero, the device shall create a new token. If $r \neq 0$, the device is not allowed to create a token and must wait for the next period in order to re-calculate r . This procedure (i.e., the random back-off) is illustrated in Fig. 11 and is necessary to minimize the likelihood that multiple devices create tokens simultaneously, which would lead to uncoordinated access to concurrent BIGs.

Lost tokens. In case that devices holding tokens disconnect, other devices in the MANTIS communication must re-create the lost token. Therefore, they invoke the token creation procedure once a timeout, in the form of a future packet sequence number, is reached. Because token re-creation is timeout-based, a device may assume the token is lost even though the token holder is only delayed. This can result in multiple tokens. One such case is when multiple devices re-create a lost token simultaneously, in which case the device with the lower `BD_ADDR` is considered the token holder and all other devices shall discard the illegitimate token. This reflects a fundamental

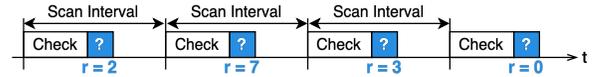


Fig. 11: **Token creation.** Nodes utilize a periodic time interval (e.g., scan interval) to check for tokens in vicinity. If no tokens are in communication range after developer-configurable, time delay, a random number r is generated (represented as a question mark). If r is zero, a node is allowed to create a new token.



Fig. 12: **Cluster separation.** Left: Nodes A, B, C, and D form a cluster with A as the isochronous broadcaster and B as the intermediary announcing A's token. Right: Nodes C and D move out of B's range and are separated from the cluster.

limitation of timeout-based coordination in mobile networks, where delay and disconnection are not always distinguishable. MANTIS therefore permits transient duplicates but resolves them deterministically once devices regain mutual visibility.

Token passing. Devices in the vicinity of the token holder, or intermediary devices disseminating token announcements, may at any point in time initiate a request for the transition of a token by executing the token passing procedure.

Token passing procedure. In case of a direct token passing with a token holder, a device shall request a token transition using scan requests. If the token holder accepts the incoming request (i.e., it replies with the corresponding scan response), the token is seamlessly transferred from the token holder to the requesting device. Consequently, the token holder transitions from the isochronous broadcaster role to the synchronized receiver role, while the requesting device transitions from the synchronized receiver role to the isochronous broadcaster role. In case multiple devices request a token transition at the same time, only one request succeeds (i.e., is acknowledged) and the others may retry, without application-level prioritization.

In the case of a token passing procedure with an intermediary device, the latter device shall first request the token from the current holder (i.e., take over the isochronous broadcaster role) and then accept the pending token passing procedure issued by the requesting node (the second transfer step thus increases the overall transition latency). For example, in the case of Fig. 10, node C may request the token held by node A by executing the token passing procedure with node B. Hence node B is considered the intermediary device and itself executes the token passing procedure to take over the ongoing isochronous data stream and, therefore, token. After node B took over the token, it accepts the pending request and transfers the token to node C.

D. Cluster Management

While token management ensures orderly role transitions within a two-hop neighborhood, devices holding or requesting tokens inevitably move. This necessitates a mechanism to manage device behavior and dynamically form clusters. Fig. 12 illustrates a scenario in which a cluster (left side) of four nodes (A, B, C, and D) becomes separated as nodes C and D move

away (right side). This separation causes nodes C and D to invoke the token creation procedure, since the token holder is perceived as disconnected and the token is considered lost. If nodes C and D do not return within range of the token holder before a set timeout (e.g., a future packet sequence number) is reached, they create a new token on the same ongoing BIS instance. As shown in Fig. 13, this allows nodes C and D to re-establish communication, even though the original token was lost when they moved outside the initial two-hop neighborhood.

Cluster merging. In order to merge clusters (Fig. 13) that belong to the same underlying BIS (e.g., caused by a cluster separation), the cluster merging procedure shall be invoked.

Cluster merging procedure. Similar to the token creation procedure, devices periodically (e.g., periods of scan interval length) check (e.g., by scanning for corresponding BLE advertisement packets) if other devices in the vicinity already advertise a token. If a synchronized device detects a token with the same underlying BIS, the device shall terminate synchronization to the node with the higher `BD_ADDR`. In case a synchronized device or token holder detects another (e.g., second) token with a higher `BD_ADDR`, it shall instruct the other token holder (or intermediary device acting on behalf of the instructing device) to immediately abandon its token.

Fig. 14 illustrates an example scenario involving two clusters: A–B and C–D. In step ①, both clusters are physically separated. If the clusters move closer together (i.e., nodes B and C come within communication range; shown in step ②), nodes B and C will receive each other’s token announcements. Since node B is synchronized to A and node C to D, this situation remains valid because A and D (the token holders) are separated by more than two hops. If node B wishes to participate in the communication, it initiates the token passing procedure (step ③). Once the token is successfully passed from node A to B (step ④), node B becomes the new token holder. However, this results in node C being within range of two tokens (B and D). Node C must follow the cluster merging procedure and terminate synchronization to the node with the higher `BD_ADDR`. Assuming node B has the higher `BD_ADDR`, node C also instructs B to abandon its token (step ⑤). If node B still wishes to participate in the conversation, it executes the token passing procedure (step ⑥), temporarily making node C the token holder (step ⑦). Node C then passes the token to B (step ⑧) to fulfill B’s token request. Ultimately, this process merges the two initially separate clusters into a single cluster.

V. MANTIS IN ACTION

Given that MANTIS has the potential to extend Bluetooth LE Audio beyond its capabilities, we implement it on real-world hardware to demonstrate and validate its intended operation.

A. Experimental Setup

Our prototype is based on Nordic nRF54L15 development kits running the Zephyr real-time operating system [17] (i.e., Nordic nRF Connect SDK v3.0.1). For Bluetooth support, we use Zephyr’s open-source controller in all experiments presented in



Fig. 13: **Separated nodes create another cluster.** After the separation in Fig. 12, nodes C and D invoke the token creation procedure and form another (new) cluster sharing the same BIS.

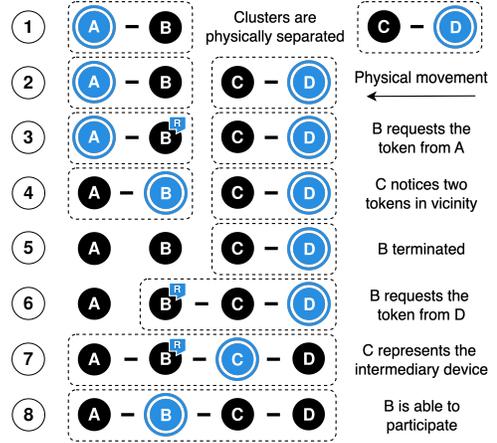


Fig. 14: **Cluster merging example.**

§V, and use the SoftDevice Controller for §III [18]. We build on Zephyr’s isochronous broadcaster and receiver samples with default parameters. To support seamless role transitions without interrupting the ongoing isochronous stream, we extend the Bluetooth controller with two vendor-specific HCI commands, thereby enabling MANTIS while preserving BC-compliance.

Token and cluster management are implemented at the application level using scan requests and scan responses for signaling between devices. The underlying isochronous transmission remains unmodified, ensuring compatibility with standard LE Audio devices. Instead of using actual audio data, which would require additional hardware for audio input and output, we generate random PCM samples and use them as input for LC3 encoding. Since the Basic Audio Profile [6] mandates the `16_2_1` broadcast audio stream configuration set (LC3 at 16 kHz with 10 ms frames, 32 kbps codec bitrate), we employ this configuration across all experiments to ensure consistency.

We configure the prototype to reflect realistic LE Audio operation. The selected parameters are representative of typical LE Audio settings and allow token transitions to complete with latencies suitable for interactive audio, though different choices would influence the observed latencies [19]. The isochronous broadcaster uses an advertising interval of 20–40 ms and transmits periodic advertisements every 40–80 ms. Synchronized receivers scan for advertisements every 10 ms with a 10 ms scan window. Token announcements and advertisements for token abandonment use a 20 ms interval, with receivers scanning at 10 ms intervals and 10 ms windows. Token creation follows the procedure in §IV, with an observation window of 50 ms to check for existing tokens before generating a random integer $r \in [0, 6]$ every 200 ms to determine eligibility. Additionally, synchronized receivers periodically scan every 200 ms for 50 ms to detect clusters (i.e., advertisements of the same BIS). To account for interference, the isochronous broadcaster role

is terminated six ISO intervals after receiving a scan request, following the BC convention, which frequently adopts this margin for critical transmissions without acknowledgments. During this period, the broadcaster continues to accept scan requests, allowing receivers to retry if a scan response is lost. The experiment on simultaneous BIG reception described in §III is carried out with seven Nordic devices placed in close proximity (i.e., line-of-sight on a desk at approximately 20 cm center-to-center spacing for all experiments in §V, unless stated otherwise), while for the evaluation of MANTIS itself we deploy four devices with the same close-proximity placement. All experiments are conducted in the same office environment with typical indoor interference and a transmission power of -40 dBm to limit range. Communication logs are collected via UART to track token transitions and clustering behavior.

B. Stream Continuity and Token Transition Latency

A core objective of MANTIS is to ensure that seamless transitions of the isochronous broadcaster role do not disturb the underlying isochronous time base. To validate this, we conduct an experiment consisting of 1000 token transitions, involving four devices participating in the ongoing isochronous stream. All devices are placed in close proximity (as defined in §V-A), and each transition is initiated by a button press on one randomly selected device. Log messages are captured on all devices to verify token passing occurs as intended and the isochronous time base remains continuous and stable during role transitions. Timestamped log entries allow us to determine the token transition latency, defined as the time from issuing the token request to the moment the requesting device assumes the isochronous broadcaster role. As illustrated in Fig. 15, the distribution of token transition latencies shows that most transitions complete quickly, with only a small fraction reaching higher values. Across 1000 transitions, the average latency is 69 ms with a standard deviation of 54 ms, and even at the 99th percentile it remains modest at 266 ms. Most importantly, the isochronous time base is preserved, with no shifts during or after transitions. To put these results in context, benchmarks such as the acceptable one-way latency in VoIP systems [19] consider latencies below 300 ms tolerable for interactive communication, and ITU-T G.114 [20] recommends that one-way delay should not exceed 400 ms. The measured token transition latency remains well within this range for the vast majority of cases, suggesting that MANTIS remains within commonly cited bounds for voice communication and is well-suited for multiparty audio.

C. Mobility and Cluster Behavior

To evaluate how well MANTIS responds to node mobility and dynamic changes in node topology, we conduct an experiment that emulates cluster separation and merging in a controlled environment. Initially, four devices are placed in close proximity (as defined in §V-A), forming a single cluster with one active token. In each of the 10 experimental runs, two devices are manually moved away from the initial cluster to physically separate them, closely resembling the scenario in Fig. 12. As the moved devices lose contact with the original token holder,

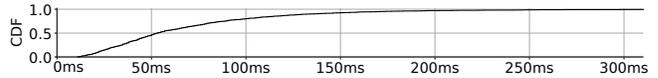


Fig. 15: CDF of the token transition latency.

they detect a token loss after a predefined timeout, and independently initiate the token creation procedure. This process includes a randomized back-off to prevent multiple devices from creating a token at the same time. Once the procedure completes, a new token is created within the separated group. We quantify token creation time as the interval between the log entry indicating token loss and the entry confirming creation. Across all experimental runs, the token creation time averages 548 ms, with a standard deviation of 274 ms. After the new cluster is established, the devices are manually moved back into communication range of the initial cluster. The merging process is automatically triggered: one token is abandoned, and the clusters converge. Merge time is quantified as the interval from the first log entry indicating detection of a token announcement from the other cluster to the point at which all four devices synchronize to the same token. Across all runs, the merge time averages 828 ms, with a standard deviation of 141 ms depending on advertisement timing and device state. While literature on clustering in mobile ad-hoc networks often focuses on control overhead or convergence criteria, it rarely reports timing for operations like token creation or merging. In contrast, MANTIS offers concrete, measured latencies. While brief desynchronization may occur, devices automatically re-synchronize to the correct stream without user intervention. These results confirm that MANTIS detects cluster separation, recovers through token creation, and resolves multiple tokens upon merging, all while preserving the stream continuity.

VI. DISCUSSION AND FUTURE WORK

Experimental limitations. While our evaluation confirms that MANTIS achieves seamless token transitions and cluster management, several aspects remain unexplored. Our experiments involved only a few devices in close proximity to validate token coordination and cluster dynamics. We used randomly generated PCM samples rather than real audio and did not quantify communication range, as it depends strongly on transmit power and deployment environment. Moreover, we did not evaluate reliability under interference or BIS pre-transmissions, though these may affect token transition timing and stream continuity. Further experiments should assess scalability and reliability in larger deployments, LC3 behavior during speaker changes, and MANTIS under interference and robustness configurations.

Energy consumption. Our implementation relies on frequent scanning and short advertising intervals and was not optimized for low-power operation. In our evaluation, we stream random PCM samples and omit audio capture and playback, and thus do not reflect a complete audio path. Under these conditions, reporting absolute energy consumption would largely reflect the chosen latency-oriented duty cycles and the incomplete audio path, and would not be representative for the considered use cases nor comparable to an end-to-end implementation. Future work should provide a dedicated energy evaluation.

Security. MANTIS inherits Bluetooth encryption and authentication, but decentralized token coordination adds risks. A malicious device may withhold the token, block transmissions, or inject false token announcements to disrupt coordination. These risks resemble ad-hoc system challenges. Future work should explore secure token passing and revocation, denial-of-service mitigation, and duplicate-token resolution after partitions.

Platform diversity. MANTIS was implemented on Nordic nRF54L15 development kits with Zephyr’s open-source Bluetooth controller. Future work should port it to other hardware and Bluetooth stacks to validate interoperability across devices.

VII. RELATED WORK

Various technologies have explored scalable wireless group communication, especially Bluetooth Mesh [21] has been proposed as a scalable framework for Bluetooth-based many-to-many communication. However, it is unsuitable for audio or voice applications due to its reliance on asynchronous operations, lack of timing guarantees, and need for pre-provisioned device roles [21], [22]. While effective for control signals in smart lighting or sensor networks, Bluetooth Mesh fails to meet the requirements of ad-hoc conversational audio [22].

Traditional radio systems such as Digital Mobile Radio (DMR) [23], General Mobile Radio Service (GMRS) [24], Family Radio Service (FRS) [25], Private Mobile Radio (PMR446) [26], Terrestrial Trunked Radio (TETRA) [27], Push-to-Talk over Cellular (PoC) [28], and Citizen Band (CB) radio [29] have long supported group voice communication. However, these systems typically depend on dedicated infrastructure or regulatory licenses, and are poorly suited for modern, short-range voice interactions in mobile settings [30]. They lack support for spontaneous, multiparty communication, as well as key features like seamless device discovery, synchronized audio, and low-power operation, capabilities increasingly critical in collaborative and ad-hoc environments [31].

3M’s Natural Interaction Behavior (NIB) system [32] targets high-noise environments and enables short-range, connectionless voice communication among participants. It supports up to four simultaneous transmitters and over sixty receivers within 10 m [32], forming a low latency single-hop network with VOX or PTT interaction. While NIB supports dynamic speaker transitions, it requires proprietary hardware, lacks encryption, and uses dedicated frequencies subject to regional compliance [32]. From a coordination perspective, token passing is an established mechanism to coordinate who may transmit in a network, for example, standardized in IEEE 802.5 Token Ring [33], and has also been considered for mobile ad-hoc networks [15]. Previous work typically uses tokens to coordinate packet transmissions on a shared medium [16]. MANTIS follows the same principle, but binds the notion of a token to the advertising layers needed to synchronize to an isochronous stream [2].

VIII. CONCLUSION

In this paper, we present MANTIS, a novel, BC-compliant mechanism that takes a step toward multiparty communication over Bluetooth’s BISes. By introducing a token and coordinated

token-passing, MANTIS transforms unidirectional BIS communication into a system supporting seamless role transitions without interrupting the isochronous stream. We implement MANTIS on real hardware and validate its intended operation. Our evaluation shows that MANTIS preserves the isochronous time base during role transitions and operates consistently under mobility and dynamic clustering. While MANTIS does not replace existing paradigms, it broadens the design space for multiparty communication based on Bluetooth LE Audio.

REFERENCES

- [1] Bluetooth SIG. Bluetooth Core Specification, v6.0. Aug 2024.
- [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: Sep 2025.
- [4] Andrew Zignani. LE Audio, Auracast Broadcast Audio, and the Future of Bluetooth Audio. <https://bit.ly/3ISxIvg>. Last access: Sep 2025.
- [5] Sipsas Kostantinos et al. Collaborative Maintenance in flow-line Manufacturing Environments. *Procedia CIRP*, 2016.
- [6] Bluetooth SIG. Basic Audio Profile, v1.0.1. Jun 2022.
- [7] Bluetooth SIG. Key Use Cases in Public Locations. <https://bit.ly/44DxTOM>. Last access: Sep 2025.
- [8] Georger Kimathi. Auracast Broadcast Audio: The New Way To Share Audio. <https://bit.ly/4bXF6ea>. Last access: Sep 2025.
- [9] Mohammad Afaneh. The Ultimate Guide to What’s New in Bluetooth version 5.2. <https://bit.ly/4jF1Kv5>. Last access: Sep 2025.
- [10] Telex Communications, Inc. *Handbook of Intercom Systems Engineering*. Bosch Communications Systems, 2007.
- [11] Susan Williams. Production Communication. <https://bit.ly/412GJj5>. Last access: Sep 2025.
- [12] Clear-Com LLC. Live Performance & Events. <https://bit.ly/3U2F256>. Last access: Sep 2025.
- [13] Savox Communications. Confined space communications. <https://bit.ly/4fbZXfy>. Last access: Sep 2025.
- [14] Nordlöf Hasse et al. Safety culture and reasons for risk-taking at a large steel-manufacturing company. *Safety Science*, 2015.
- [15] Navneet Malpani et al. Distributed token circulation in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2005.
- [16] Daniela Maniezzo et al. T-MAH: A Token Passing MAC protocol for Ad Hoc Networks. In *Proc. of the 1st IFIP Med Hoc Net*, 2002.
- [17] Zephyr Project. About the Zephyr Project. <https://zephyrproject.org/learn-about>. Last access: Sep 2025.
- [18] Nordic Semiconductor. Bluetooth stack architecture. <https://bit.ly/46hNNio>. Last access: Sep 2025.
- [19] Maulik Shah. What is VoIP Latency and Why Does It Matter and How to Fix It. <https://bit.ly/4iXUjhn>. Last access: Sep 2025.
- [20] International Telecommunication Union. ITU-T G.114. May. 2003.
- [21] Bluetooth SIG. Mesh Protocol, v1.1. Sep 2023.
- [22] Mohammad Afaneh. Bluetooth Mesh Networking: The Ultimate Guide. <https://bit.ly/3RQUcJA>. Last access: Sep 2025.
- [23] ETSI. Digital Mobile Radio (DMR), TR 102 398 V1.5.1. Nov 2023.
- [24] Federal Communications Commission. General Mobile Radio Service (GMRS). <https://bit.ly/4k3gxzE>. Last access: Sep 2025.
- [25] Federal Communications Commission. Family Radio Service (FRS). <https://bit.ly/4m8tt9p>. Last access: Sep 2025.
- [26] Ofcom. Analogue and Digital PMR446 Information Sheet. Feb 2018.
- [27] 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.
- [28] Hytera. Push-to-Talk Over Cellular (POC). <https://bit.ly/3Yr9hoX>. Last access: Sep 2025.
- [29] Federal Communications Commission. Citizens Band Radio Service (CBRS). <https://bit.ly/4d2u1th>. Last access: Sep 2025.
- [30] Christopher Bayliss. Scalability and Connectivity Challenges for the Future of Digital Radio Communication. Univ. of Nebraska-Li. 2024.
- [31] National Research Council; Computer Science and Telecommunications Board. *The Evolution of Untethered Communications*. 1997.
- [32] Minnesota Mining and Manufacturing Company. 3M™ PELTOR™ Natural Interaction Behavior (NIB) Overview. Jun 2022.
- [33] IEEE Standards Association. Token Ring Bus Access Method, IEEE/ANSI 802.5-1985. Apr. 1985.