IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 10 December 2025, accepted 7 January 2026, date of publication 14 January 2026, date of current version 21 January 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3654254

RESEARCH ARTICLE

Legacy-Compatible Multi-User Wi-Fi Aggregation for Enhanced Downlink Efficiency and Security

JOSÉ LUIS SALAZAR-RIAÑO, JULIÁN FERNÁNDEZ-NAVAJAS, ÁNGELA HERNÁNDEZ-SOLANA, JOSÉ RUIZ MAS, GUILLERMO AZUARA-GUILLÉN, AND RAMÓN CAJAL-PÉREZ
Aragón Institute for Engineering Research (I3A), University of Zaragoza, 50018 Zaragoza, Spain

Corresponding author: José Luis Salazar-Riaño (jsalazar@unizar.es)

This work was supported by Ministerio de Ciencia, Innovación y Universidades/Agencia Estatal de Investigación (MICIU/AEI)/10.13039/501100011033/Fondo Europeo de Desarrollo Regional (FEDER) EU, European Regional Development Fund/European Union (ERDF/EU), Gobierno de Aragón, under Grant T31_23R and Grant PID2022-136476OB-I00.

ABSTRACT This work introduces a novel Wi-Fi frame aggregation mechanism that enables efficient multi-user downlink transmissions in IEEE 802.11 networks (including earlier standards). We focus on exploring feasible and practical approaches to enable or approximate multi-user aggregation by leveraging existing features, while ensuring the compatibility with legacy standards. The proposed approach, implemented and evaluated on a real-world testbed, allows the Access Point (AP) to aggregate MAC Service Data Units (MSDUs) destined for multiple users/stations, using a scheme based on Multichannel Broadcast Encryption of Short Messages. It is particularly effective in scenarios involving a high volume of small data packets to multiple users/stations and where communication reliability is critical to maintaining high-quality service. This innovative solution not only extends and improves the efficiency of standard aggregation schemes in such scenarios but also enables efficient multi-user downlink transmissions and enhances security by strengthening the protection of transmitted data.

INDEX TERMS Frame aggregation, multichannel broadcast encryption, multi-user transmission downlink, small frame optimization, Wi-Fi downlink optimization, Wi-Fi networks IEEE 802.11.

I. INTRODUCTION

The evolution of Wi-Fi standards has been driven by the need to improve overall network performance by optimizing spectral efficiency and radio spectrum usage, increasing transmission rates, and reducing communication latency—key factors for ensuring high quality of service (QoS).

Among these advancements, frame aggregation techniques —first introduced in the 802.11n standard [1] and further refined in subsequent versions [2], including the most recent ones such as 802.11ax (Wi-Fi 6) [3] and the upcoming 802.11be (Wi-Fi 7)—have proven to be fundamental tools for enhancing overall network performance [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

Specifically, 802.11n introduced two aggregation methods: A-MSDU (Aggregated MAC Service Data Unit), which combines multiple MSDUs into a single MPDU (MAC Protocol Data Unit), and A-MPDU (Aggregated MPDU), which joints multiple MPDUs, into a single physical (PHY) frame—Physical Layer Protocol Data Unit (PPDU)— for transmission. Both methods reduce the overhead associated with transmitting individual PHY frames for each MSDU and improve overall channel utilization. However, A-MPDU offers greater robustness by allowing individual MPDUs within the aggregate to be acknowledged and retransmitted independently.

These mechanisms were further improved in later standards—most notably Wi-Fi 6—by increasing aggregation limits, enabling higher throughput and better performance in dense deployments. However, a key limitation of both A-MSDU and A-MPDU is that they only allow aggregating MSDUs destined for the same STA—same physical receiver address (RA). This limits efficiency in scenarios with numerous users running services that generate

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Yuan Chen.

small-sized packets, such as IoT applications, short-packet traffic (e.g., VoIP or gaming), or control processes like TCP acknowledgments [14].

Multiplexing of MPDUs with different DA/SA values provides flexibility only at the logical address level, not at the physical level. For instance, this can occur in scenarios involving group-addressed frames or proxy ARP, where the AP logically represents multiple entities. An AP may transmit an A-MPDU with multiple MPDUs, all physically addressed to the same STA (i.e., same RA), but each carrying a logical DA corresponding to a different virtual client or group. All of this leads us to conclude that the 802.11 standards do not support true aggregation of multiple users within a single A-MPDU.

By contrast, 802.11ac and 802.11ax support MU-MIMO (Multiple User – Multiple-Input Multiple-Output), enabling simultaneous transmission of separate spatial streams to multiple STAs—in 802.11ac limited to downlink and up to 4 STAs. However, MU-MIMO operates independently from frame aggregation, which remains limited to data packets destined for a single STA (per A-MPDU/A-MSDU). This separation restricts efficiency gains, as true multi-user aggregation within a transmission received in a spatial stream is not possible. 802.11ax introduces genuine multi-user transmission capabilities using MU-PPDUs (Multi-User – Physical Protocol Data Units), allowing separate A-MPDUs to be transmitted simultaneously to multiple STAs by dividing resources in the time and frequency domains using OFDMA (Orthogonal Frequency Division Multiple Access) [15], [16], [17]. Each A-MPDU is uniquely assigned to a specific STA and is transmitted over that STA's allocated Resource Unit (RU) when using OFDMA, or over a dedicated spatial stream in the case of MU-MIMO. This allows the AP to efficiently serve multiple devices in parallel within a single transmission opportunity, significantly improving spectral efficiency and reducing latency.

In any case, it should be noted that even in both 802.11ac and 802.11ax, these mechanisms—which, independently of frame aggregation, enable parallel multi-user transmissions—remain underutilized in current commercial implementations, primarily due to the complexities involved in effective scheduling. For this reason, we aim to explore feasible and implementable alternatives for both current (802.11ac, 802.11ax) and earlier standards that enable or approximate multi-user aggregation by reusing existing PHY/MAC mechanisms and protocol elements—such as logical addressing—while ensuring compatibility with legacy standards.

The proposed method focuses on downlink transmissions, where multiple small MSDUs, regardless of their destination STA, are grouped into a single aggregated frame with a unique RA MAC address: a multicast address. A key aspect of this approach is that, instead of relying on the WPA2 or WPA3 encryption models, each MSDU is individually encrypted using the key of its intended receiving STA. Any receiving STA can read the MPDU, but each MSDU can only be decrypted by the STA holding the corresponding key. This enables MSDUs destined for different STA to be aggregated into the same frame but requires a new security layer. The proposed cryptographic scheme is based on Multichannel Broadcast Encryption for Short Messages, we have first introduced in [18].

A key challenge of the proposed approach is the reliability of transmission of the aggregated multi-user frames, since multicast transmission will be used but they carry a grouping of unicast data whose information should not be readable except by its legitimate unicast destination, even if multicast addressing is employed. Furthermore, legacy Wi-Fi standards offer *limited multicast support, typically at low transmission rates and without any feedback mechanism to ensure quality of service* [19], [20]. Since multicast frames are transmitted without acknowledgment (ACK) frames from clients, reliable delivery cannot be guaranteed. Moreover, the absence of an ARQ mechanism forces multicast frames to be transmitted using the lowest available Modulation and Coding Scheme (MCS) [21]. To address these limitations and enhance both performance and reliability, *a multicast system capable of dynamically adapting the transmission rate is required* [19].

The objective of this work is to adopt a legacy-compatible mechanism for managing multicast transmission, thereby enabling multi-user frame aggregation in IEEE 802.11 networks compatible with PHY/MAC of existing standards. The selected application scenarios are those in which the proposed mechanism is expected to provide the greatest benefit. For instance, STAs involved in applications that generate very small data units and usually operate in relatively static scenarios. Examples include Wi-Fi environments where wireless devices—such as smartphones, tablets, and laptops—run interactive audio applications (e.g., WhatsApp), interactive video or video conferencing applications (e.g., Skype), or online gaming (e.g., First-Person Shooter games, FPS). These applications produce traffic with small packet sizes, typically less than 200 bytes for audio and gaming, and up to 600 bytes for Skype video [22], [23].

In summary, the goal of this work is to design and evaluate a secure multi-user frame aggregation method based on multiplexing techniques using a Multichannel Broadcast Encryption of Short Messages scheme [18]. The aggregation method will be integrated into a realistic testbed environment [24], [25]—a software-defined network (SDN) experimental prototype that also includes service-oriented slicing of resources [25]—ensuring proper operation with commercial network devices and validating its behavior under varying operational conditions. To this end, the efficiency of the proposed protocol will be assessed through a series of performance tests, allowing comparison with the aggregation techniques defined in the Wi-Fi standard. This analysis will identify the protocol's strengths, weaknesses, and areas for improvement, laying the groundwork for future enhancements.

The rest of the paper is organized as follows. Section II review the state of the art. Section III presents the full description of the proposed method. Section IV describes the testbed. Section V presents the performance analysis of the proposal and finally, section VI draws the conclusions.

II. RELATED WORK

The evolution of Wi-Fi standards has focused on improving network performance by optimizing spectral efficiency and reduced latency. Frame aggregation has been widely studied in terms of performance, reliability, and QoS adaptation. Over time, research has shifted toward optimizing aggregation for specific environments, with proposals that reduce latency by adjusting aggregate length or adapting to traffic load [26], [27], [28], [29], [30], extend device lifetime in high-load IoT or e-Health scenarios [31], [32], [33], or improve coexistence with other radio technologies [34]. In all these approaches, aggregated frames are always destined to a single STA and therefore cannot be used to transmit to multiple users simultaneously.

Certainly, multi-user PPDU was introduced later, in Wi-Fi 6. However, the complexity of effective scheduling and even the overhead of required control mean that these systems are currently underutilized, and viable and implementable solutions are being looked for both current and previous standards.

The proposal presented here is not intended to compete with IEEE 802.11ax, whose potential flexibility is evident, but rather to provide a practical and feasible solution for aggregating short MSDU addressed to multiple STAs using standards such as 802.11n, 802.11ac, or even 802.11ax. In the latter case, this is achieved without the need to implement all the additional signaling associated with full OFDMA multiplexing, thereby avoiding the overhead it entails.

The solution adopted to transmit content to more than one user has been to use a multicast RA address. Our proposal, designed for scenarios with low or no mobility, aims to be effective even when relying on the basic support provided by legacy 802.11a/b/g/n multicast transmissions. However, in order to improve both performance and reliability of multicast transmission, adaptive rate control and feedback mechanisms are required.

Recent amendments—notably IEEE 802.11aa (Group Addressed Service Enhancements) and IEEE 802.11ax—have introduced mechanisms such as Directed Multicast Service (DMS) and Groupcast with Retries (GCR), which provide adaptive rates, reliability, acknowledgments, and even QoS differentiation for multicast traffic. DMS operates by generating *n* copies of a multicast frame and assigning each copy as a unicast frame—allowing feedback and rate adaptation for each receiver. GCR, on the other hand, comprises three retransmission methods: traditional legacy multicast, which transmits frames without acknowledgments; Unsolicited Retries (UR), where a specified number of retry attempts are performed to improve reliability; and Block ACK (BACK), where the AP requests a Block ACK from receivers, enabling selective retransmissions and per-frame rate adaptation.

In this regard, solutions proposed in the literature focus on handling feedback information from the receivers, obtained for instance from data and control uplink frames coming from unicast transmissions or beacon responses—either from all nodes [35], a limited set of nodes or a designated leader (LBP-ACK [36], LBP-NACK [37], Pseudo Broadcast) [38])—to allow the transmitter to adjust its transmission rate and reliability, typically using ARQ-based mechanisms. In addition, in proposals as in Pseudo Broadcast, the multicast stream is actually sent as a unicast to a designated leader STA, while other STAs are required to operate in promiscuous (monitor) mode to receive the frames without sending ACKs, but represents an intrusion into the STA configuration. Going further, there are even attempts to recover A-MPDU aggregation using OFDM (Orthogonal Frequency-Division Multiplexing) to allow multiple clients to simultaneously transmit their feedback to the AP without collisions, thus minimizing feedback overhead [21].

Unfortunately, many of these existing mechanisms exhibit significant spectral efficiency degradation as a result of excessive control overhead (i.e., feedback overhead), depending on how that feedback is obtained. We must keep in mind that continuous collection of reports affects network performance [20]. Furthermore, certain approaches necessitate modifications to the IEEE 802.11 standard or to end-user device implementations, which hinders their practical deployment. This is why proposals such as AMuSe [20] emerge, based on accurate receiver feedback that incurs only in a small control overhead. AMuSe develops an algorithm for the dynamic selection of a subset of multicast receivers as feedback nodes, which periodically send information about channel quality to the multicast sender.

Our idea goes further and integrates this type of solution in an Software Defined Wireless Network (SDWN) environment [24], [25], in line with the solution proposed in SDN@Play [39]. SDWN facilitates the incorporation of a local monitoring system at each access point allowing to a central controller to collect information from all Wi-Fi APs and enables Software Defined Networks (SDN)-based network applications to make intelligent and coordinated decisions aimed at optimizing overall network performance. In [39], when up-to-date information on unicast transmissions to/from STAs is not available at SDN@Play platform, a monitoring mechanism of two phases is defined. In the first phase, if required, directed multicast (DMS) or unicast-based multicast is used to obtain the necessary measurements from the receivers involved via the rate control algorithm. In the second phase, the controller uses the collected information to select the MCSs with the highest probability of delivery in legacy multicast transmission. SDN@Play provides a practical and programmable multicast rate-adaptation solution, which is compatible with the 802.11 standard.

All this is done with scalable, efficient, and secure management of multicast communication. We aim to protect information while minimizing the management and variability of the groups of STAs involved in receiving multi-user aggregated frames, depending on the traffic pattern. In our case, it is not necessary to implement traditional multicast group association mechanisms and management; rather, each STA is simply linked by the AP, through management control, to a group for receiving a specific type of traffic pattern. The requirements imposed by the objectives we propose require us to provide security for an aggregated multicast frame, ensuring that each data block (MSDU or aggregated MSDUs) destined for each STA is protected separately.

The IEEE 802.11 standard allows the use of group keys for multicast/broadcast traffic, but this imposes limitations when trying to combine security with multi-user frame aggregation. A shared group key is only feasible if all recipients can decrypt the same plaintext (as in Unichannel Broadcast Encryption [40]). Furthermore, the IEEE 802.11 standard does not natively support having different encryption for data blocks aggregated within the same frame, not even in the case of A-MPDU. Each MSDU in an A-MPDU has its own MAC header, which theoretically could allow different encryption per MSDU. However in practice, each data block must still use the same temporal session key (TK) established between the AP and the corresponding STA, according to WPA2/WPA3 [41].

In our proposal, we aim to apply encryption individually to each aggregated data block while minimizing all associated overhead. This includes reducing or eliminating the need for separate MAC headers for each STA, allowing secure multi-user frame aggregation and, overcoming the limitations imposed by the standard. It also minimizes airtime consumption and processing overhead.

On this way, the objective is to meet these requirements by applying a secure cryptographic model/protocol. A cryptographic protocol well suited to these needs is Multichannel Broadcast Encryption [42], originally designed for pay television broadcasting but not widely accepted outside of these scenarios. It was in [42] that the concept was formally defined. It uses a combined key scheme: a special key or cryptographic information is generated for each authorized user, but the content is transmitted once in an aggregated (encrypted) transmission. This ensures that only authorized users can decrypt the message, while unauthorized users cannot access the information.

Authors in [42] define two security models to select were established depending on an adversary's capabilities to attack the system. Later, in [43], a third adversary model (adaptive) was defined that hardened the security conditions. However, MCBE solution is not practical due to the high complexity of the decryption, which is linear in function of the number of receivers. In addition, the message header contains cryptographic information required for multiple authorized receivers to decrypt it. In the original scheme, its size increases with the number of receivers, adding overhead. In [44], a modification of [42] is presented to achieve a constant header size regardless of the number of receivers. In [45], a double proposal for improving the MCBE protocol is made, adapting them to very specific user requirements. However, these approaches remain computationally expensive for decryption, as they all rely on bilinear maps as the underlying cryptographic primitive.

Departing from the traditional cryptographic primitive (bilinear maps), the key and novel approach proposed in this work, which forms the basis of our implementation, relies on using the Chinese Remainder Theorem (CRT) [46] for key distribution. The cryptographic scheme we propose extends the Multichannel Broadcast Encryption for Short Messages framework, we originally introduced in [18]. This approach allows MCBE to be adapted to Wi-Fi environments, reducing space requirements and overhead. For practical purposes, as will be explained in the detailed description of the proposal, the data stream is transmitted without containing specific information about the exact position of the data intended for each STA or the addresses of the STAs receiving the message. However, the proposal allows each STA—belonging to the group of STAs participating in this aggregation scheme—to recover only the information intended for it by applying the corresponding decryption keys.

The proposed security also controls scalability, as multicast communication is restricted by the AP through the appropriate selection of MSDUs from STAs when creating a multi-user aggregated frame. In this way, the proposal limits both the number of STA groups (pseudo-multicast groups of STAs monitoring a common multicast address) and the variability of these groups.

These multicast frames do not always have to contain information addressed to all members of the group, but can be a specific selection of members at any given time, optimizing transmission in terms of efficiency and reliability. The reception of a frame by group members that does not contain anything addressed to them does not involve any computational cost.

III. SECURE MULTI-USER FRAME AGGREGATION METHOD

This section presents the proposed secure multi-user frame aggregation scheme. The feasibility of the proposed approach requires compatibility with legacy IEEE 802.11 standards. To remain compatible with existing hardware and protocol specifications, each frame must still use a single Receiver Address (RA). Using a multicast RA enables simultaneous and aggregated delivery of data packets to multiple STAs while preserving the standard frame format and avoiding modifications to lower protocol layers.

The proposal is feasible without considering the need for ACKs and retransmissions at the MAC level, which is consistent with the legacy IEEE 802.11a/b/g/n multicast support. That is, reliable communication is achieved by selecting a robust MCS. Any lost data packets are detected and retransmitted at higher layers. However, the objective is to avoid transmitting multicast frames at the basic MCS/rate, which typically occurs due to the absence ACKs. To overcome this limitation, our proposed implementation can support rate adaptation in a scheme similar to the leader-based one. Realize that other proposals concerning rate adaptation and retransmission could be considered without affecting the core of the MU aggregation proposal. However, we have chosen a simple approach that ensures compatibility with legacy devices.

In our case, transmission efficiency is improved by selecting the optimal MCS for each multicast frame—specifically, the MCS with the highest probability of successful reception among all STAs included in the aggregated frame. This MCS corresponds to the STA with the weakest signal quality, considered to be the leader of the aggregation. Network performance can be further enhanced through channel-aware grouping. In this approach, data packet aggregation is selectively applied to STAs within a group that exhibit similar channel conditions, thereby enabling the use of higher transmission rates in many cases and ultimately improving overall efficiency.

The feasibility of the proposed scheme relies on maintaining a prioritized list of STAs selected to provide feedback, which can be dynamically adjusted based on channel quality measurements obtained from other unicast transmissions involving these STAs. This information must be kept up to date, including, if necessary, unicast transmission phases to collect new measurements [39]. Outdated MCS estimation could occur due to the delay between updating the MCS for a given STA (based on unicast measurements) and the moment the AP applies it. However, significant errors are not expected, as the MU aggregation proposal is intended for scenarios with little or no STA mobility—precisely the scenarios where the greatest performance gains are expected.

Finally, unlike traditional multicast transmissions, as already mentioned, each STA should only be able to recover the information contained in the aggregated frame intended for itself (data block). The separation between STAs and the guarantee of confidentiality are ensured by applying independent encryption to the data destined for each STA. Each station must possess the necessary keys to decrypt only its own data block and must not obtain any information about other users' data — not even the number of STAs receiving data. Key generation is an asynchronous task relative to communication. For optimal system efficiency, a set of pre-generated keys of varying lengths, denoted by λ, should be created in advance for each STA to match different message sizes. These keys, associated with each STA, are fully known by the AP and the STA.

In coordinated Wi-Fi networks, the formation and management of STA groups whose traffic can be aggregated is handled by the infrastructure, often delegated directly to the AP in simpler scenarios, and can be implemented with significantly lower complexity than standard multicast group management. Group formation and traffic management become simple in contexts where the infrastructure applies concepts such as slicing, since short-packet traffic may already be marked as belonging to a specific slice. In such cases, the multicast group address can be directly associated with that slice, streamlining the overall process. The AP monitors downlink traffic patterns in order to determine whether the proposed multiuser aggregation application is worthwhile for improving the overall network performance, or whether it is better to apply only unicast transmissions, with or without using a standard aggregation scheme. The details regarding traffic pattern identification and group formation are beyond the scope of this work.

This section details the proposed MAC frame aggregation mechanism, including the frame structure, multiplexing logic, and encryption procedures. Particular attention is given to the structural constraints imposed by the encryption process, which shape the organization of the aggregated frame.

A. FRAME AGGREGATION METHOD: CONSTRAINTS IMPOSED BY THE ENCRYPTION PROCESS

As mentioned, the system is designed to be compatible with the IEEE 802.11 specifications. The aim is to minimize additional overhead and reduce computational complexity. In this way, when a STA receives the multiplexed encrypted data, it cannot know in advance whether any information is intended for it, since the addressed STAs are not explicitly indicated in order to reduce overheads. Instead, it must apply its own keys to determine if the transmission contains data directed to it. This will influence the design.

The scheme aggregates multiple data blocks, each associated with a different STA. Each data block may encapsulate a single MSDU, a fragment of an MSDU, or a custom aggregation of multiple MSDUs belonging to the same STA, each payload preceded by a subheader indicating its length. Unlike the A-MSDU aggregation method defined in IEEE 802.11, our proposal omits the destination and source MAC addresses (DA/SA) from the individual data blocks within the multicast frame payload, thereby reducing overhead. Although this aggregation format is not specified by the standard, the approach is permitted under IEEE 802.11, as it allows the transmission of a standard multicast frame with an arbitrary payload, provided that the overall frame format is valid. LLC/SNAP (Logical Link Control/ SubNetwork Access Protocol) header explicitly indicates the use of a specific payload type, and both endpoints are designed to interpret the custom content. Notably, a destination address for each MSDU is not required, as each STA can recover its own data using its respective encryption key. As for the actual source, routing and addressing responsibilities are handled at the IP layer. Fig. 1 illustrates de structure for the proposed method compared with the A-MSDU aggregation method.

The proposed cryptographic scheme is based on Multichannel Broadcast Encryption for Short Messages, we have first introduced in [18]. This method is designed to securely and efficiently deliver multiple short messages to different recipients within a single physical frame. This approach relies fundamentally on the CRT [46], which

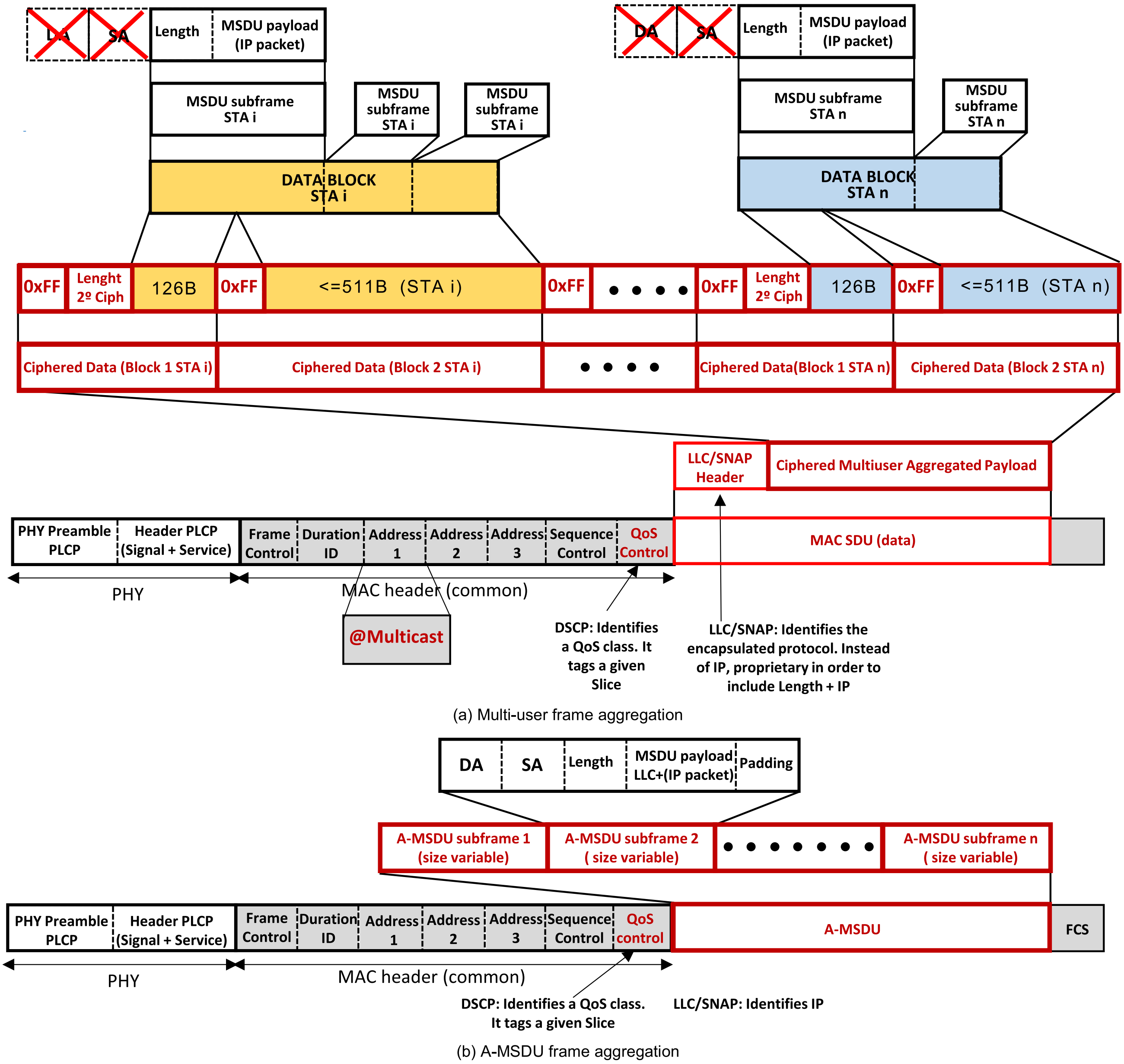

DA
SA
Length
MSDU payload (IP packet)
MSDU subframe STA i
MSDU subframe STA i
MSDU subframe STA i
DATA BLOCK STA i
MSDU subframe STA n
MSDU subframe STA n
DATA BLOCK STA n
0xFF
Lenght 2º Ciph
126B
0xFF
<=511B (STA i)
0xFF
• • • •
0xFF
Lenght 2º Ciph
126B
0xFF
<=511B (STA n)
Ciphered Data (Block 1 STA i)
Ciphered Data (Block 2 STA i)
• • • •
Ciphered Data(Block 1 STA n)
Ciphered Data (Block 2 STA n)
LLC/SNAP Header
Ciphered Multiuser Aggregated Payload
PHY Preamble PLCP
Header PLCP (Signal + Service)
Frame Control
Duration ID
Address 1
Address 2
Address 3
Sequence Control
QoS Control
MAC SDU (data)
PHY
MAC header (common)
@Multicast
DSCP: Identifies a QoS class. It tags a given Slice
LLC/SNAP: Identifies the encapsulated protocol. Instead of IP, proprietary in order to include Length + IP
(a) Multi-user frame aggregation
DA
SA
Length
MSDU payload LLC+(IP packet)
Padding
A-MSDU subframe 1 (size variable)
A-MSDU subframe 2 (size variable)
• • • • • • •
A-MSDU subframe n (size variable)
PHY Preamble PLCP
Header PLCP (Signal + Service)
Frame Control
Duration ID
Address 1
Address 2
Address 3
Sequence Control
QoS control
A-MSDU
FCS
PHY
MAC header (common)
DSCP: Identifies a QoS class. It tags a given Slice
LLC/SNAP: Identifies IP
(b) A-MSDU frame aggregation

FIGURE 1. **Proposed multi-user frame aggregation method compared with the A-MSDU aggregation method: (a) Multi-user frame aggregation (b) A-MSDU aggregation.**

provides the mathematical foundation. Using CRT, the sender combines all messages into a single value instead of sending each one separately, allowing each user to securely and individually recover their own message.

The proposed message combination must take into account the following restrictions:

1) Encryption is a mathematical operation that requires the data to always begin with a byte whose most significant bit is set to one (typically chosen as the byte value 0xFF). This convention is necessary to avoid ambiguity when the data starts with zero bytes, as numerically they would otherwise be indistinguishable.
2) The size of the data block to be encrypted— typically the MPDU plus the mentioned byte 0xFF— must not exceed the length of the encryption key. To optimize efficiency and minimize the need for padding, the data block should closely approximate the key size. This is because the resulting encrypted block will have a fixed size equal to that of the encryption key.
3) Key lengths are constrained to be integer multiples of 8 bytes to ensure alignment and compatibility with underlying algorithmic structures. The selection of an appropriate key size represents a critical trade-off between security and computational efficiency. Keys exceeding 512 bytes introduce

substantial computational overhead, which may negatively impact system performance, particularly in resource-constrained environments. Conversely, keys shorter than 128 bytes are generally considered insecure due to their susceptibility to brute-force attacks or cryptanalytic vulnerabilities. Consequently, the key size is bounded within the range of 128 to 512 bytes, establishing a practical compromise between security guarantees and computational feasibility.

4) The most efficient approach would be to have a distinct key size for each possible data block length between 128 and 512 bytes. However, to limit complexity, the number of possible key sizes will be constrained by incrementing in steps of 16 bytes. While this introduces some inefficiency due to padding, it significantly reduces the number of keys that must be generated and managed—24 different key sizes will suffice.

5) During the encryption process, key reuse between different data blocks corresponding to the same STA is strictly prohibited (and naturally, keys are also not shared among different STAs). Encrypting multiple blocks with an identical key leads to the ciphered data becoming algebraically entangled, thereby compromising message recoverability. This failure arises because the necessary conditions stipulated by the CRT are violated, undermining the mathematical foundation required for successful decryption.

As referred before, when STA receives the multiplexed encrypted data, it cannot know in advance whether any information is intended for it. A priori, it must attempt decryption using all of its assigned keys in order to identify any addressed information. This would impose a high computational burden on the STA.

To mitigate this problem, a scheme has been defined whereby, when traffic is directed to a given STA, two data blocks are created: one with the minimum allowed size (128 bytes) and another containing the remaining data (not exceeding the upper limit of 512 bytes). Each of these blocks must begin with the predefined byte value 0xFF. The first block includes an additional byte indicating the size of the second key to be used, represented as an index (since there are only 24 possible key sizes). If the second block is not required, this index is set to zero. Consequently, the actual payload capacity of the first block is limited to 126 bytes (128-0xFF and subsequent key block used), while the second block can carry up to 511 bytes. Together, these two blocks provide a total data capacity up to 637 bytes for a single STA.

B. MULTICHANNEL BROADCAST ENCRYPTION FOR SHORT MESSAGE. HOW IT WORKS

We will begin with a concise description of the mathematical foundation of CRT, followed by the phases of key generation, encryption, transmission, and decryption of the information, including the differential nuances introduced in the proposal—that condition MPDU structure described before—compared to a classical encryption scheme based on the CRT.

The basis of the CRT says that given a set of n integers $(p_1, p_2, \ldots, p_n)$ that are pairwise coprime—meaning that every pair of these numbers shares no common divisors other than 1 (i.e., their greatest common divisor, GCD, is 1)—and any set of residues $(a_1, a_2, \ldots, a_n)$, there exists a unique number x mod N (where $N = p_1 \cdot p_2 \cdots p_n$) such that satisfies (1):

$$x \equiv a_i \pmod{p_i} \text{ for each } \quad i = 1, \ldots, n \tag{1}$$

That is, the CRT allows you to combine multiple smaller values (the remainders a_i) into the single larger number x from which each original value can be independently recovered using its corresponding modulus.

1) KEY GENERATION PHASE

The key generation phase is extender in order to ensure multiple block transmission for each STA. That is, there are n users/STAs (with $u_1, u_2, \ldots, u_n$ as their identifiers), and each STA is intended to receive several blocks (messages). These messages may have different lengths, which affects the size of the prime moduli used for encryption, which are adjusted as closely as possible to the message sizes.

The key generation process traditionally starts by defining λ, a security parameter that specifies the bit-length of both the cryptographic keys and the primes employed in the encryption process. In our case, however, instead of a single λ, we define a vector of security parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_L)$ where each λ_k specifies the bit-length required for the k-th encryption length instance ($L = 24$ possible key sizes), reflecting varying message sizes or security levels. Based on this, the key generation operates as follows:

For every user or STA u_i, and for each message size index $j = 1, \ldots, L$ that STA is intended to receive, the AP selects a prime number $p_{i,j}$ of size λ_j bits (with λ_j taken from the index j of vector λ), such that all primes $p_{i,j}$ are pairwise coprime, as shown in (2):

$$\gcd(p_{i,j}, p_{n,m}) = 1 \quad \text{for all } (i, j) \neq (n, m) \tag{2}$$

This ensures compatibility with the CRT.

Then, for each $p_{i,j}$, a secret key $x_{i,j} \in \mathbb{Z}^*_{p_{i,j}}$ is chosen, ensuring (3):

$$\gcd(x_{i,j}, p_{i,j} - 1) = 1 \tag{3}$$

where $\mathbb{Z}^*_{p_{i,j}}$ represents the set (multiplicative group) of integers modulo $p_{i,j}$ that are coprime with $p_{i,j}$; that is, the set of numbers between 1 and $p_{i,j}$-1, since $p_{i,j}$ is a prime number. This ensures that exponentiation with $x_{i,j}$ is invertible modulo $p_{i,j}$.

Finally, the encryption key set (EK) is (4):

$$EK = \left\{ (p_{i,j}, x_{i,j}) \middle| \, 1 \leq i \leq n, 1 \leq j \leq L \right\} \tag{4}$$

Each user u_i privately receives and stores all its assigned secret key pairs $EK_i = \{(p_{i,1}, x_{i,1}), (p_{i,2}, x_{i,2}), \ldots, (p_{i,l}, x_{i,L})\}$, one for each message size it is allowed to decrypt. Optionally, multiple sets of L key pairs may be generated to account for the possibility of transmitting several data packets of identical size to the same STA.

Note that, key generation is asynchronous with respect to communication. As stated earlier, the key management and distribution process among destination STAs is out of the scope of this paper.

2) ENCRYPTION

Encryption first requires the sender (AP) to uniformly select a random integer Hdr from the interval $(1, 2^{1024})$.

Hdr serves as a cryptographic seed from which user-specific variants, denoted $Hdr_{i,j}$, are derived. Note that, specifically, each message $m_{i,j}$ of size type $j \in \{1, \ldots, 24\}$ sent to STA i requires a unique random $Hdr_{i,j}$ value for encryption with its corresponding key $x_{i,j}$. $Hdr_{i,j}$ is defined as the output of a pseudorandom function (e.g. any Hash-based Message Authentication Code—HMAC— function customizing the range set to be $\mathbb{Z}^*_{p_{i,j}}$) seeded with Hdr, tailored to each user and belonging to the ring $\mathbb{Z}^*_{p_{i,j}}$. However, it should be noted that, in fact, the first used values of $Hdr_{i,j}$ are derived from Hdr, and each subsequent used value is recursively computed from the preceding one as (5):

$$\begin{aligned} Hdr^{(1)}_{i,j} &= Derive(Hdr, p_{i,j}) \bmod p_{i,j},\ \ 1 < Hdr^{(1)}_{i,j} < p_{i,j} \\ Hdr^{(k)}_{i,j} &= Derive(Hdr^{(k-1)}_{i,j}, p_{i,j}) \bmod p_{i,j},\ \ 1 < Hdr^{(k)}_{i,j} < p_{i,j} \end{aligned} \tag{5}$$

Then, the ciphered message component, $c_{i,k}$, for the k-th message $m_{i,k}$ of user i is computed as (6):

$$c_{i,k} = (m_{i,k} + Hdr^{x_{i,j}}_{i,j})(\bmod p_{i,j}) \tag{6}$$

where $j = j(i, k) \in \{1, \ldots, 24\}$ denotes the message size category of $m_{i,k}$. Note that, it is important to ensure that the bit-length of the message $m_{i,k}$ is strictly less than the bit-length of the corresponding prime modulus $p_{i,j}$. Consequently, $p_{i,j}$ is randomly picked from the primes with just one bit more than the bit-length defined by the message size category j. By applying (6), it is ensured that the ciphered message is indistinguishable in reception from random without knowledge of the key and the seed $Hdr_{i,j}$ and key $x_{i,j}$. Note that is desirable to use a different encryption key pair ($p_{i,j}$, $x_{i,j}$) for each message, thus, we do not consider transmission of multiple messages of identical size unless multiple sets of key pairs are generated, as previously mentioned.

Next, all individual ciphered messages are combined into a single encrypted value ET modulo N using CRT, as shown in (7):

$$\begin{aligned} ET = \Bigg[\sum_{i\in U}\sum_{k=1}^{k_i} c_{i,k} \cdot \left(\left(\frac{N}{p_{i,j(i,k)}}\right)^{-1} \bmod p_{i,j(i,k)}\right) \\ \cdot \left(\frac{N}{p_{i,j(i,k)}}\right)\Bigg] \bmod N \end{aligned} \tag{7}$$

where U is the set of users or STAs receiving data in a given transmission and each user $i \in U$ has k_i messages, $m_{i,k}$ ($k = 1,\ldots, k_i$), each one associated with its best size type $j = j(u, k) \in \{1,\ldots,24\}$ which determines the encryption key pair ($p_{i,j}$, $x_{i,j}$) and, N is defined as shown in (8):

$$N = \prod_{i\in U}\prod_{k=1}^{k_i} p_{i,k} \tag{8}$$

Finally, in the traditional CRT-based scheme, the transmitted message explicitly includes (Hdr, ET). In contrast, it is important to highlight that our proposal differs from the traditional CRT-based scheme in some key respects:

- To reduce overhead, Hdr is completely excluded from the transmitted encrypted message. Similar to the encryption key set (EK), Hdr is generated asynchronously with respect to communications and distributed independently. Consequently, the values of $Hdr_{i,j}$ are derived locally at the AP and each recipient(STA) using Hdr as a global seed, without the need to transmit the seed itself. This ensures that each recipient can independently generate the required per-message random values while minimizing communication overhead. That means that, contrary to the traditional CRT-based scheme—where the values Hdr or $Hdr_{i,j}$ were generated for each encrypted message and targeted the specific set of users that were the intended recipients of the MU-aggregated frame—our approach does not rely on per-message header generation. Instead, the Hdr seed is common to all potential recipient users of this type of MU aggregation within the cell and remains valid for deriving $Hdr_{i,j}$ for all possible message lengths.
- The Hdr seed is updated only after a certain number of transmissions to enhance system robustness; its update is not time-critical and may not occur during the entire residence time of a STA in the cell. In contrast, the $Hdr_{i,j}$ values are updated locally for each encrypted message according to the recursive scheme defined in (5). It is true that reusing the same $Hdr_{i,j}$ across multiple transmissions may, in general, increase the risk of ciphertext correlation, message pattern leakage, and potential recipient inference—thereby potentially undermining semantic security and unlink ability. However, the formal analysis in [18] demonstrates that a reduced update frequency for seed Hdr maintains robust security. This is because, in fact, each previously used $Hdr_{i,j}$ value is replaced with a new $Hdr_{i,j}$ for every transmission. This provides strong protection against both passive and active adversaries.
- According with the proposal, message detection necessarily operates in a blind manner. Consequently, each STA within the set of STAs that are allowed to receive this type of multi-user frame aggregation may be required to attempt decryption using all of its assigned keys in order to determine whether any information is addressed to it. Nevertheless, as outlined in the frame aggregation structure, the proposed scheme introduces mechanisms to significantly reduce the number

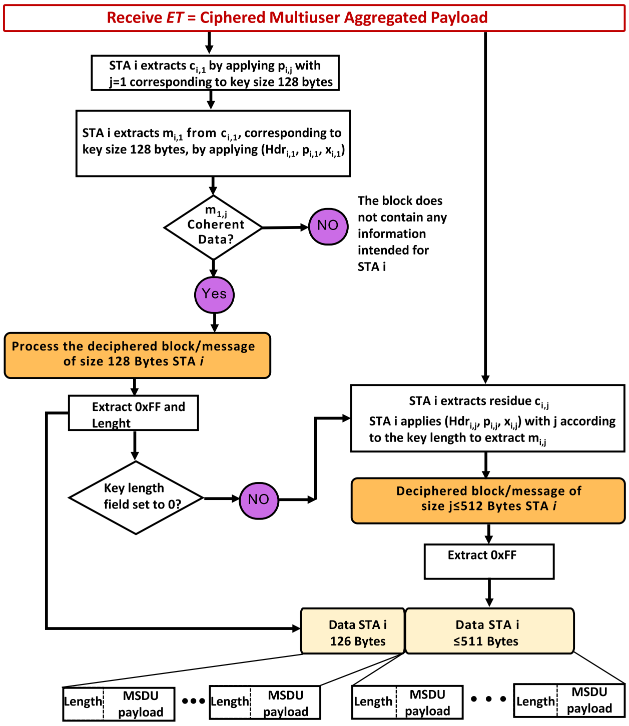

Receive ET = Ciphered Multiuser Aggregated Payload
STA i extracts c_{i,1} by applying p_{i,j} with j=1 corresponding to key size 128 bytes
STA i extracts m_{i,1} from c_{i,1}, corresponding to key size 128 bytes, by applying (Hdr_{i,1}, p_{i,1}, x_{i,1})
m_{1,j} Coherent Data?
NO
The block does not contain any information intended for STA i
Yes
Process the deciphered block/message of size 128 Bytes STA i
Extract 0xFF and Lenght
Key length field set to 0?
NO
STA i extracts residue c_{i,j}
STA i applies (Hdr_{i,j}, p_{i,j}, x_{i,j}) with j according to the key length to extract m_{i,j}
Deciphered block/message of size j≤512 Bytes STA i
Extract 0xFF
Data STA i 126 Bytes
Data STA i ≤511 Bytes
Length MSDU payload
Length MSDU payload
Length MSDU payload
Length MSDU payload

FIGURE 2. Illustration of the detailed decryption process.

of required decryption attempts. Each STA is restricted to receiving at most two messages: one with the minimum allowed size (128 bytes), and another carrying the remaining data, which does not exceed the upper bound of 512 bytes.

3) TRANSMISSION

AP transmits the single frame containing *ET*.

4) DECRYPTION

Upon receiving *ET*, each user or STA i attempts to obtain a residue corresponding to a message (first with a length of 128) according with (9):

$$c_{i,j} = ET \bmod \quad p_{i,j} \tag{9}$$

If residue is obtained, then, STA i recovers the original message according with (10):

$$m_{i,j} = c_{i,j} - Hdr_{i,j}^{x_{i,j}} \quad \bmod p_{i,j} \tag{10}$$

Note that, since *Hdr* is sent asynchronously with respect to the transmission, each user has already computed all $Hdr_{i.j}$ values from *Hdr*.

Moreover, upon detecting a 128-bit message, recall that—according to the protocol design—this first block contains an additional byte indicating the size of the second key to be used. As a result, subsequent detections become computationally more efficient, since the user no longer needs to iterate over all possible key lengths.

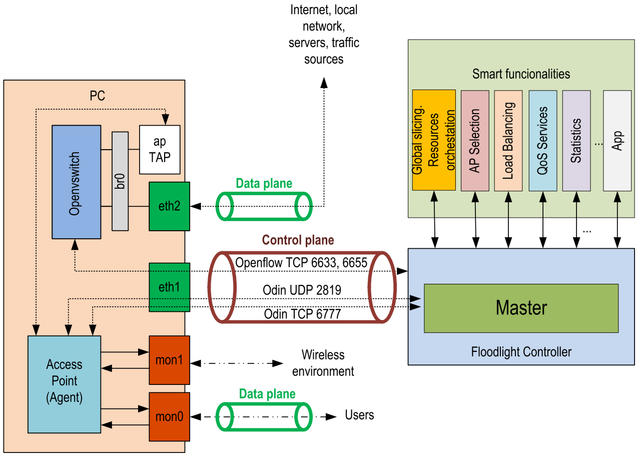


FIGURE 3. High level architecture of the solution.

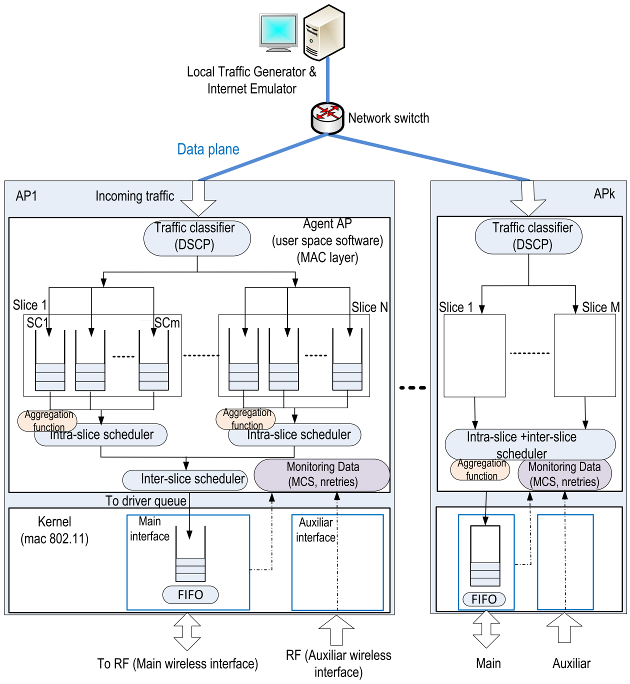

Local Traffic Generator & Internet Emulator
Network switch
Data plane
AP1
Incoming traffic
Traffic classifier (DSCP)
Agent AP (user space software) (MAC layer)
Slice 1
SC1
SCm
Slice N
Aggregation function
Intra-slice scheduler
Aggregation function
Intra-slice scheduler
Inter-slice scheduler
Monitoring Data (MCS, nretries)
To driver queue
Kernel (mac 802.11)
Main interface
FIFO
Auxiliar interface
To RF (Main wireless interface)
RF (Auxiliar wireless interface)
APk
Traffic classifier (DSCP)
Slice 1
Slice M
Intra-slice +inter-slice scheduler
Aggregation function
Monitoring Data (MCS, nretries)
FIFO
Main
Auxiliar

FIGURE 4. Local implementation in the APs (data plane).

The decryption procedure carried out by each user/STA to extract their corresponding messages from the aggregated ciphertext ET is depicted in Fig. 2.

IV. ARCHITECTURE FRAMEWORK AND ISSUES CONCERNING PRACTICAL IMPLEMENTATION

A. ARCHITECTURE FRAMEWORK

The work emphasizes the practical implementation and performance evaluation of the proposal. The experimental prototype, which integrates the proposed frame aggregation mechanism, is built upon the open-source framework described in [24] and derived from [47]. The high-level architecture of this framework is shown in Fig. 3. The framework consists of a central controller and multiple distributed agents (the access points-APs).

The central controller collects information from all Wi-Fi access points and enables SDN-based network applications to make intelligent and coordinated decisions aimed at optimizing overall network performance. These mechanisms address multiple aspects of network operation, including client-to-AP association, load balancing through proactive and reactive handoffs, and the coordination and management of resources across APs, including support for features as network slicing and, more broadly, QoS provisioning. These applications operate reactively and proactively using measurements collected by the APs.

Then, in a distributed manner, the local agents—implemented in user space on their respective APs—handle non-time-critical Wi-Fi MAC functionalities. Operating under the configuration and policies set by a centralized controller, the agents enable localized optimization of QoS and resource utilization at the AP level. In contrast, time-critical operations of the Wi-Fi MAC protocol—such as the Distributed Coordination Function (DCF), timing, and acknowledgments—are handled directly by the mac80211 kernel, independently of the underlying driver.

Local agents are linked to the mac80211 subsystem and are compatible with any Linux-supported Wi-Fi chipset that allows frame injection in monitor mode. Frame injection according to Radiotap [48] allows the active transmission of crafted or modified frames into the wireless medium, making this configuration the most suitable option for experimental and testing scenarios in Wi-Fi network environments.

Our testbed deliberately excludes the use or modification of driver-specific features such as rate control or the aggregation due to the difficulty of managing them jointly with frame injection. Instead, such functionalities—including A-MSDU and the proposed aggregation method, in addition to local slicing or scheduling features—are entirely implemented within the software AP agent. Thus, the AP agent is vendor-agnostic and compatible with a wide range of Wi-Fi module drivers. Consequently, a central contribution of this work is the development of a frame aggregation mechanism that does not depend on any specific 802.11 driver implementation.

In this context, Fig. 4 illustrates the basic functionality of the user plane as depicted in Fig. 3, which serves as the basis for implementing and testing the proposed multi-user frame aggregation method based on Multichannel Broadcast Encryption of Short Messages.

Since this aggregation occurs in user-space software (within the agent), it is important to clarify all aspects of the Wi-Fi MAC layer involved in this process. As shown in Fig. 4, at each AP, incoming data IP packets are temporarily stored in the socket buffer and classified into queues based on predefined traffic rules. APs gather key performance metrics—such as throughput, delay, MCS in use, retries, RSSI, and per-STA airtime consumption—supporting not only local functionalities (rate control, scheduling and frame aggregation) but also coordinated decision-making at the controller level, as well as comprehensive performance evaluation and analysis. Each AP in the coordinated Wi-Fi network is responsible for injecting frames, whether aggregated or not, under the conditions defined at any given time.

B. FRAME AGGREGATION, TRAFFIC SCHEDULING AND RAN SLICING

An important aspect to highlight is the evaluation of the proposed solution within a context that also incorporates the concept of network slicing. Network slicing enables the creation of multiple virtual end-to-end networks on top of a shared physical infrastructure, or alternatively, allows partitioning of network resources into multiple logical segments tailored to different applications or use cases [49], [50], [51].

Specifically, the architecture developed in this work implements a hierarchical slicing approach, defined in [25], that flexibly integrates two complementary levels of control. For instance, it can be used to define a first level following an ISS (Infrastructure Slice Selection) approach, which segments network resources into multiple slices, each tailored to specific categories of users. Embedded within each ISS slice, the second level—based on the QoSS (Quality of Service Slicing) approach—further differentiates services and applications according to their specific QoS requirements.

Nevertheless, in the context of this work, slicing is used exclusively to separate traffic generated by applications with short-message patterns—such as messaging services (e.g., WhatsApp) or online gaming—from other types of services. Specifically, we configure only two slices, where traffic from different STAs belonging to the same service class (slice) is aggregated into a shared queue. The identification of slices and the corresponding queue is based on the Differentiated Services Code Point (DSCP) field, present in the IP headers.

This slice partition helps constrain the network resources consumed by such short-message traffic types, while ensuring that other services, classified into different slices, receive the appropriate level of resource allocation and quality of service. Nevertheless, the essential element for effective execution of the Radio Access Network (RAN) slicing solution is the local slicing component, built at the local agent (on the top of IEEE 802.11 MAC layer of the wireless transceivers). It is responsible for distributing physical resources—in the case of Wi-Fi, the airtime—among active flows sharing the same physical radio, i.e., the flows generated by Wi-Fi clients associated with a given AP. The allocation depends on the slice to which each flow is assigned. The most widely adopted approach to enabling network slicing in Wi-Fi is airtime allocation, defined as the fraction of time that an Access Point (AP) transmission to a Station (STA) occupies the shared wireless medium. The slice policies implemented in this work build upon a revised version of the well-known Airtime Deficit Weighted Round Robin (ADWRR) scheduling principles [52].

According to the ADWRR scheduling, each service flow contending for a link has a corresponding queue with an associated count called Deficit Counter (DC), which in our case indicates the amount of airtime (time in μs) the flow can use in the round. Queues are visited in a round robin fashion. Upon each visit, the DC is firstly increased by a fixed quantity: the called quantum (Q). Then, packets in the queue are selected for transmission as long as the airtime required by each packet is smaller than the DC's current value, knowing that DC is decreased by the airtime of each selected packet after each one is sent. Otherwise, if DC is smaller than the required airtime, the queue is skipped, and the process continues with the next queue. When queues become empty, DC is set to zero.

The practical scheduling implementations ensure isolation between slices and predictable service behavior in heterogeneous applications as proved in [25].

The airtime share assigned to a slice s depends on its quantum $Q[s]$ relative to the sum of the quanta of all active slices. Specifically, when a slice s has pending traffic in its queues, it is guaranteed to receive at least a fraction of the AP's airtime, as expressed in (11):

$$Q[s]/\sum_{i\in S} Q[i] \tag{11}$$

where S is the set of slices instantiated at the AP. A slice may also utilize unused airtime from other slices, distributed proportionally to the quanta of slices requesting additional resources. This approach ensures fair service across active slices while maximizing the efficient use of available airtime.

When a packet is scheduled and selected for transmission, the airtime estimation is primary computed according with (12) as the time required to deliver a unicast frame over the air interface and to receive the corresponding acknowledgment.

$$Airtime = \overline{T_{Backoff}} + T_{DIFS} + T_{DATA} + T_{SIFS} + T_{ACK} \tag{12}$$

The per-frame transmission overhead due to the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol is estimated as the sum the average back-off time before transmission ($T_{Backoff}$), the Distributed Inter-Frame Space (T_{DIFS}), the Short Inter-Frame Space (T_{SIFS}) and the transmission time of the acknowledgment (ACK) frame control (T_{ACK}). Note that when multi-user frames are transmitted using a multicast address, the standard operation is that the access point (AP) does not expect an ACK. Since no single station is responsible for acknowledging the frame, the ACK time and the SIFS before the ACK are not needed in the airtime estimation in (12).

Concerning the data frame transmission time (T_{DATA}), it is computed according to the used standard and antenna configuration. Equation (13) shows the expression for OFDM based standards 802.11n and 802.11ac:

$$\begin{aligned} T_{DATA} &= T_{PHY_{OH}} + T_{MPDU} \\ &= T_{PHY_{OH}} + T_{symbol}\left\lceil \frac{8\cdot(MAC_{OH}+L)}{T_{symbol}\cdot R(MCS)}\right\rceil \end{aligned} \tag{13}$$

where T_{PHYOH} (μs) involves the physical overheads and the time associated to MPDU transmission, T_{MPDU}, depends on how many OFDM symbols are needed to transmit the total bits of MPDU at the given physical rate. It depends on the MAC overheads, MAC_{OH}, the length of the data (L) received from the LLC level and the physical rate (depends of the MCS). Then, if a packet is retransmitted, the additional airtime required can be approximated as the number of retries monitored by the auxiliary interface ($n_{retries}$) multiplied by the airtime obtained with (12). This correction is applied during the next scheduling decision associated with the same slice, as described in [25].

Finally, when aggregation is enabled, the maximum length of the MPDU payload that can be transmitted is determined based on the available deficit, regardless of whether the standard A-MSDU aggregation mode or the proposed method is used. Using this value—subject to the maximum allowed aggregation size and the constraints imposed by the defined structures—as many packets as possible are aggregated.

Different policies can be implemented for selecting packets to aggregate. In this work, a basic policy has been adopted. For standard A-MSDU, aggregated MSDUs belong to the same STA as the packet at the head of the queue. Since a queue may store packets from multiple STAs, it is permitted to search within the buffer for packets from the same STA to complete the aggregation up to the allowed size.

When the multi-user frame aggregation method is applied, the same principles are followed to construct the data blocks. Aggregation begins with the STA corresponding to the head-of-queue packet and proceeds with subsequent STAs in FIFO order until the aggregation reaches the allowed size. Finally, the airtime of the aggregated frame is computed.

V. EXPERIMENTAL TESTBED & RESULTS

In this phase of the experimental analysis, we evaluate the effectiveness of the proposed multi-user frame aggregation method (hereafter referred to as MU-Cyph_A-MSDU) and compare its performance with the currently standardized Wi-Fi aggregation scheme, A-MSDU (hereafter referred to as A-MSDU).

Note that, concerning the practical implementation, the encryption based on the CRT is a technique that guarantees confidentiality but involves a computational cost that must be taken into account, since it requires intensive mathematical operations. This makes the choice of hardware hosting the AP agent critical, as it must meet minimum requirements that needs to be combined with an efficient programming and integration within the software on the agent side. The computational cost depends on the size of the encrypted block, and it is not linear. That is, encrypting a 512-byte block is more expensive than encrypting four 128-byte blocks. Modular computation is among the most computationally expensive operations in cryptographic protocols, primarily due to the repeated modular reductions involved. In our case, modular exponentiation (6) and the modular inverse computation (7) constitute the dominant contributors to the overall computational cost. However, when the bases used in the modular exponentiation—such as $Hdr_{i,j}$ in (6)—are known beforehand, both the exponentiation (6) and the modular inverse of the associated product of integers (7) can be partially precomputed. In our case, it should also be noted that, starting from the *Hdr* seed, a number of recursively derived $Hdr_{i,j}$ values can be made available at both ends (AP and STA). Thus, precomputations are viable, allowing the ciphering process to be significantly accelerated and resulting in a substantial reduction of overall execution time. At the AP, a parallel module maintains the repository of $Hdr_{i,j}$ values and precomputations, updating them only as they are consumed. In the usage scenarios where MU aggregation is expected to be beneficial, the number of STAs involved in this type of communication per AP is unlikely to be high (up to 20–30), and mobility among cells very low. Consequently, the rate at which precomputations need to be incorporated for a new STA is minimal. Moreover, since the *Hdr* communicated to a new STA can be the same as that of existing STAs (i.e., the *Hdr* does not need to be updated), there is no need to rebuild the precomputation bases for all STAs. At the STA, which lacks such capabilities, the worst-case scenario is still manageable: once an $Hdr_{i,j}$ value is used, there is at least an interval equal to the average inter-packet time (for messages of the same size) available to perform precomputations. For the target traffic patterns, this interval is on the order of several milliseconds.

On the other hand, the size of the aggregation needs to be considered. In this work, the maximum aggregated block size has been restricted to 1468 bytes (close to 1500 bytes) for both A-MSDU aggregation and the multicast aggregation payload. It should be noted that these aggregation sizes are significantly smaller than those contemplated by the standard A-MSDU or A-MPDU mechanisms defined in IEEE 802.11n and subsequent amendments. Nevertheless, this choice is consistent with the specific usage context targeted by the proposed multiuser aggregation scheme. In particular, the focus is on aggregating very short packets (approximately 100–200 bytes), corresponding to applications such as online gaming or interactive audio applications (e.g., WhatsApp), where inter-packet arrival times are typically in the range of 20–40 ms. In other words, we are dealing with applications generating traffic on the order of only a few tens of Kbps. Even under scenarios with high STA density, it would not be reasonable to assume very large aggregation sizes. This context serves as a suitable basis for evaluating encryption delays thresholds, taking, for instance, as a reference the encryption of three blocks of (128 + 512) bytes and an upper bound on the expected aggregated frames per second.

We will first describe our testbed, the metrics and evaluation conditions. Then, results are presented and discussed.

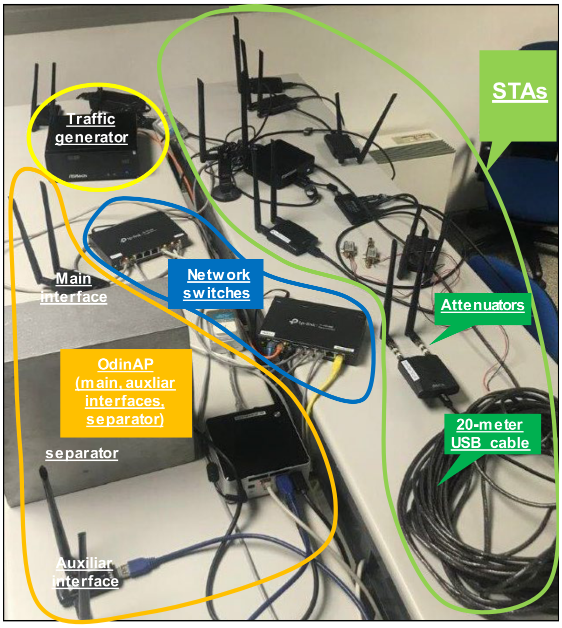


FIGURE 5. Schematic and real equipment used for the tests.

A. TESTBED SETUP

Fig. 5 shows the testbed including the different elements involved in Fig.3: APs as software entities running in user space on a mini PC, Ethernet switches to build the networks corresponding to the data plane and the management plane, a mini PC acting as a traffic generator and IP flow classifier, and a set of STAs consisting of several USB wireless network cards connected to a mini PC. All the elements are connected to the management network, where the traffic can be monitored.

The AP run on a miniPC—in our case, an Asus NUC RNUC13ANKI5, i5-1340P with Ubuntu 22 kernel 5.19.9. Each AP uses two USB wireless cards, which support frame injection in monitor mode in the 5 GHz band (i.e. Alfa AWUS036ACH and/or Alfa AWUS036ACM with RTL8812au and MT7612u chipsets). Both network cards (and their antennas) are separated by a reflector as an effective strategy to reduce channel correlation in MIMO systems or configurations with closely located antennas.

The STAs are also a set of USB wireless network cards (i.e. Alfa AWUS036ACH and/or Alfa AWUS036ACM) connected to a mini PC (also an Asus NUC RNUC13ANKI5, i5-1340P, with Ubuntu 22 and kernel 5.19.9). These wireless cards operate in active monitor mode, meaning they are capable of replying with ACKs to received data frames.

The traffic generator and IP flow classifier runs on mini PC (also an Asus NUC RNUC13ANKI5, i5-1340P with Ubuntu 22 kernel 5.19.9) running Ubuntu (kernel 5.19.9). Traffic generation is handled using the iPerf tool and a specific C program to generate exponential traffic, while the ARP cache is manually manipulated to include the MAC addresses of the

STAs. Additionally, the iptables utility is employed as a traffic classifier by tagging outgoing datagrams with slice identifiers in the DSCP field of the outgoing datagrams.

B. METRICS AND EVALUACION CONDITIONS

The evaluation was conducted on the testbed shown in Fig. 4, which corresponds to the data plane depicted in Fig. 3, using real equipment operating in the 5 GHz band with IEEE 802.11n and 802.11ac standards.

The evaluation is conducted in a scenario that incorporates the concept of slicing, where applications generating short-packet traffic, which can potentially be aggregated, are classified and managed within a dedicated slice. The configuration of the slicing guarantees this slice is allocated a percentage P% of the airtime resources, coexisting with another slice that handles the remaining traffic and also has a minimum guarantee of (100-P)%. The test scenario is configured so that the traffic demand of the competing slice exceeds its guaranteed resources and the combined demand of both slices surpasses the total AP airtime capacity. Consequently, as inherent to the AWDRR-based slicing method employed, any unused resources from the short-packet traffic slice can be dynamically utilized by the competing slice.

The objective is not to conduct an explicit evaluation in terms of throughput although it could be derived. Rather, under varying and increasing IP traffic demand conditions of short-packet slice, the goal is to assess the potential advantage of the proposed method in terms of- airtime usage. The less airtime consumed by the aggregated short-packet traffic, the more resources become available for the traffic of other applications. Note that, under all conditions in which the airtime consumption of a slice is below its guaranteed percentage, the traffic demand is fully satisfied. In other words, the traffic demand (in Mbps) matches the achieved throughput. Evaluation also includes performance in terms of delay (from the moment an IP packet enters the AP via the ethernet interface until it is transmitted over the Wi-Fi interface). For a clearer interpretation of the results, the number of frames transmitted and the IP data size carried in each of these frames—whether aggregated or not—are also reported.

Since the results may be influenced by the slicing configuration parameters, their impact will be also evaluated. The same percentage of resource allocation can be achieved with different quantum (Q) values. Quantum values determine the time slice and may range from tens of microseconds (μs) to tens of milliseconds (ms). When aggregation is allowed, the quantum value must be sufficiently large to accommodate the airtime of aggregated frames. A larger quantum allows more time for packet aggregation, resulting in larger frames, lower airtime consumption but higher delay. There is a trade-off between efficiency and latency. Therefore, the choice of the quantum is not trivial and must be adjusted according to the type of delay-sensitive services, such as voice or online gaming. High latencies cannot be tolerated, even if channel efficiency is improved.

Concerning the impact of channel conditions at the intended receiver STAs, which affect both reliable delivery and airtime efficiency, we introduce some simplifying assumptions to better assess the advantages of the proposed multi-user aggregation method. First, as illustrated in Fig. 5, most STAs are located within a 5-meter radius of the AP, ensuring favorable RSSI conditions. To emulate different propagation scenarios—resulting in different transmission rates under rate control—we selected different but fixed MCS values for all STAs. As noted in the introduction, the intended application scenarios involve multiple STAs running applications that inherently require low terminal mobility. Therefore, MCS variations associated with rate control mechanisms applied to STAs are expected to be minimal. Consequently, the tests were carried out without considering an explicit implementation of feedback collection and the associated rate-adaptation mechanisms at the STAs. Instead, we opted to emulate their effects by always considering the use of the MCS with the highest probability of successful reception among all STAs included in the aggregated frame. That is, the lowest MCS of receiving STAs. Specifically, performance is evaluated under three basic conditions illustrative conditions: (i) aggregating STA traffic following a FIFO discipline and applying the lowest MCS required by any STA in the frame; (ii) aggregating STA traffic with a FIFO discipline but selecting only packets from STAs with similar channel conditions, and thus requiring the same or a very close MCS; and (iii) to assess the extreme effect of STA channel-based selection, aggregating packets from all STAs but all requiring the same MCS, thereby isolating the impact of the chosen MCS.

Finally, regarding multicast transmission, the multi-user aggregated frame is implemented in the testbed as a unicast transmission addressed to a designated leader STA. All STAs operate in promiscuous (active monitor) mode, allowing them to receive and process the transmitted frames without sending ACKs, except for the leader STA (the one with the lowest MCS, as determined by the scheduling selection). This setup provides feedback to the AP regarding reception by the leader STA, enabling, for example, the discarding of test realizations affected by external interferences. That is, control the evaluation of MU-Cyph_A-MSDU and A_MSDU are conducted in similar conditions. However, this approach affects airtime estimation, which must account for SIFS and ACK. This assumption does not significantly affect the comparison with the A-MSDU scheme. In fact, in a true multicast transmission, the airtime consumed would be even lower under the proposed scheme, further highlighting its benefits. Moreover, the results are applicable to multicast-supporting schemes that require STA adaptations, such as the Pseudo-Broadcast approach [38], which also involve ACKs.

C. RESULTS

This section evaluates the effectiveness of the proposed multi-user frame aggregation method. The system operates in 802.11n/HT (5 GHz) supporting MCS 0 to 15.

Downstream UDP traffic, consisting of short-packet flows assigned to slice 0, is generated from the miniPC towards nine STAs. This traffic emulates the behavior of applications characterized by frequent small-packet transmissions, such as interactive audio applications (e.g., WhatsApp) or online gaming.

Although the number of STAs considered in the test is relatively limited, it is sufficient to yield results that are both representative and generalizable. Thus, the focus lies not on the absolute values obtained, but rather on the qualitative comparison between the proposed method (MU-Cyph_A-MSDU) and the standard aggregation mechanisms (e.g., A-MSDU) in terms of spectrum efficiency and delay.

Traffic classified into slice 0 is modeled as generic, with an IP packet size distribution averaging 125 bytes and uniformly ranging from 46 to 204 bytes, while the global packet generation process is modeled according to a Poisson process with rate λ IP packets/s, with an equally probable distribution across the nine STAs. The specific C program generates IP packets according to the required exponential interarrival time, but software limitations imposes a minimum interarrival interval of $t_{\min} \approx 80\ \mu s$. To achieve the desired mean target rate, the exponential rate is adjusted, resulting in a truncated distribution with a mean given by $1/\lambda = t_{\min} + 1/\lambda_{\text{set}}$, being λ_{set} the introduced parameter on the generation process. For IP rates requiring shorter inter-arrival times, parallel software generation processes are activated.

This short-packet traffic competes with iPerf traffic assigned to slice 1, directed to a single STA (which does not reduce the generality of the results), using MCS7, and generated at a constant rate of 10 Mbps with packets of 250 bytes. No aggregation is applied in slice 1.

The quantum (Q) assigned to each slice are equal, ensuring that 50% of the airtime is allocated to each slice. Three different quanta (2, 5, and 10 ms) have been considered in order to evaluate the impact of this parameter on performance. As previously mentioned, the traffic demand of slice 1 far exceeds its guaranteed airtime while the global short-IP packet traffic rate varies from 1 to 14 Mbps. Specifically, the estimated airtime for the iPerf traffic is 221.5 μs, which implies that the number of packets that can be transmitted is below 4514 p/s, assuming the AP uses all of its available airtime. Consequently, the demand from slice 1 not only exceeds its guaranteed 50% but will also utilize any resources left unused by slice 0.

The results are presented in Fig. 6, which depicts twelve graphs, four for each of the referred quantum configurations (columns corresponds with Q = 2ms, 5ms and 10ms, respectively). Each row of the Fig. 6 corresponds to one of the evaluated metrics: airtime, delay in ms, mean data size per aggregated frame (mean number IP bytes) and number of aggregated frames/s.

Each graph further compares the results of MU-Cyph_A-MSDU and A-MSDU under three different scenarios: all STAs requiring MCS1 (denoted as MCS1), all requiring MCS7 (denoted as MCS7), and the nine STAs requiring MCS 1, 3, and 7 in equal proportions (mixed MCS scenario). In the mixed MCS scenario, two sub-cases are considered for MU-Cyph_A-MSDU:

> *3a Lowest MCS: transmissions are performed using the lowest required MCS among the aggregated STAs*
> 3b *Channel-aware grouping*: packets are aggregated only for STAs with similar channel conditions

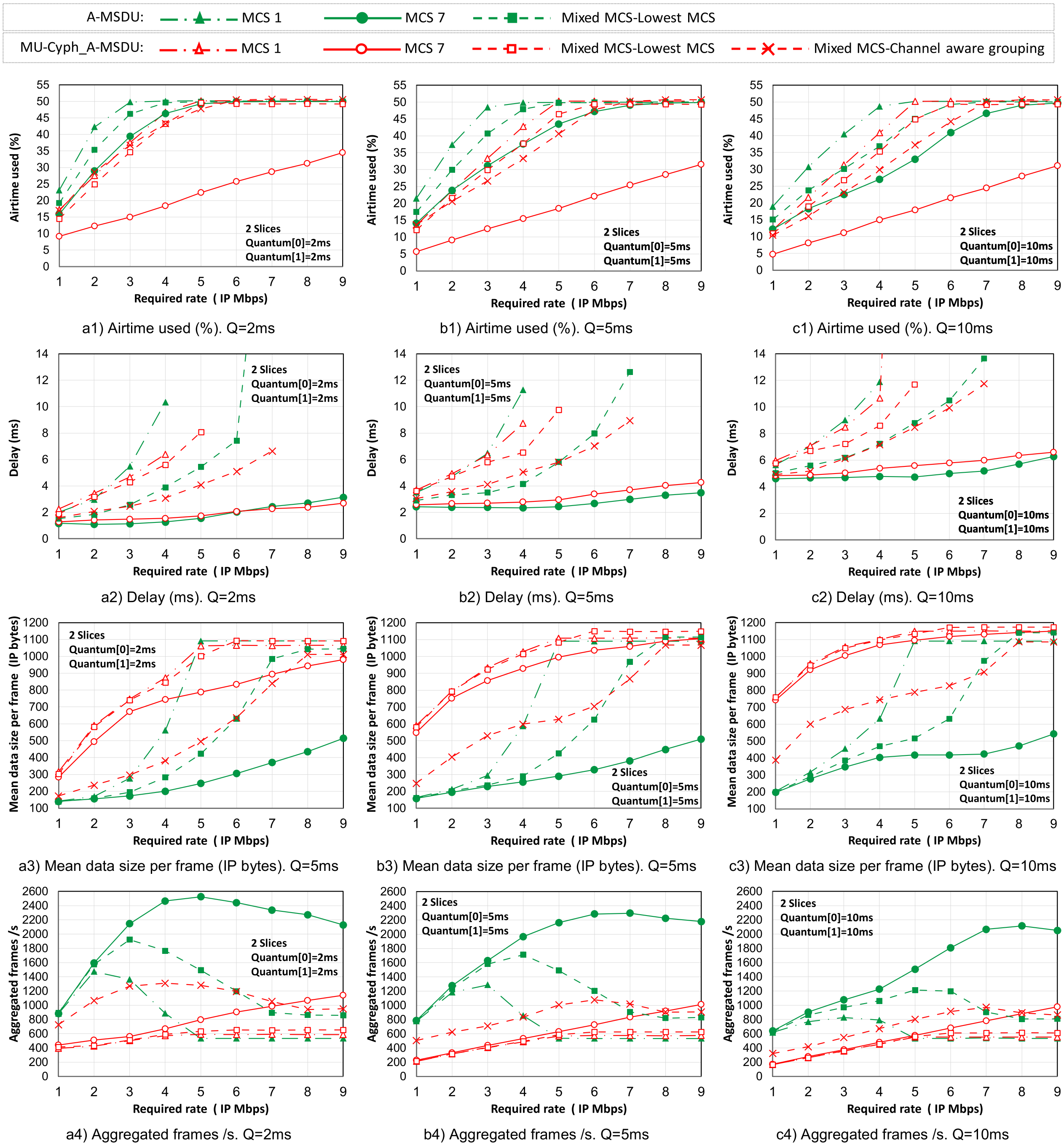
Fig. 6 shows data only up to 9 Mbps to improve the visualization of most options within the range in which they can operate without saturation. Fig. 6 is complemented by Fig. 7, which extends the range of required rates for the scenario where all STAs operate with MCS7, while simultaneously illustrating the impact of the quantum value within the same graph.

A general observation across all cases (A-MSDU or MU-Cyph_A-MSDU) is that as long as airtime consumption remains below 50%, 100% of the required traffic is served. When the full guaranteed airtime must be used (50%), the representative parameter becomes delay. Delay remains moderate as long as the entire demanded traffic can be transmitted. Note that when delay is not shown in the graph, it is because the demand from slice 0 exceeds capacity, causing the slice to enter saturation (delays spike or, alternatively packet drops spike in user-space buffers). In all the cases, as the load increases, the number of frames per second initially rises, but then, as packets accumulate, delays increase while promoting greater packet aggregation, as reflected in the mean IP data frame size. This subsequently causes a drop in the number of transmitted frames per second due to airtime consume limitations. Finally, both the number of aggregated frames per second and the size of the aggregated frames stabilize in the saturation operating range (a range in which the system could only operate for very short and sporadic intervals). This behavior is more pronounced in A-MSDU than in MU-Cyph_A-MSDU. In the latter case, aggregation starts earlier and the increase in the number of frames per second remains almost linear up to saturation, except for the channel-aware grouping option. This performance is because the slicing method ensures isolation between slices. Thus, slice 0 never exceeds its guaranteed airtime as long as the demand from slice 1 requires all of its resources.

Regarding the impact of the quantum parameter selection on airtime consumption, the airtime curve shows lower values as the quantum increases. This is because the quantum, by distributing airtime among slices, inherently introduces a delay that promotes greater frame aggregation, with the effect being qualitatively similar for both aggregation methods.

Focusing on the performance comparison between the proposed method and A-MSDU both in Fig. 6 and 7, the following observations can be made:

When examining the MU-Cyph_A-MSDU curves, we can see that airtime use values are lower than those of A-MSDU in all the cases, until throughput approaches the saturation levels, at which point demand exceeds transmission capacity. This is obviously, because more packets can be aggregated



a1) Airtime used (%). Q=2ms

b1) Airtime used (%). Q=5ms

c1) Airtime used (%). Q=10ms

a2) Delay (ms). Q=2ms

b2) Delay (ms). Q=5ms

c2) Delay (ms). Q=10ms

a3) Mean data size per frame (IP bytes). Q=5ms

b3) Mean data size per frame (IP bytes). Q=5ms

c3) Mean data size per frame (IP bytes). Q=10ms

a4) Aggregated frames /s. Q=2ms

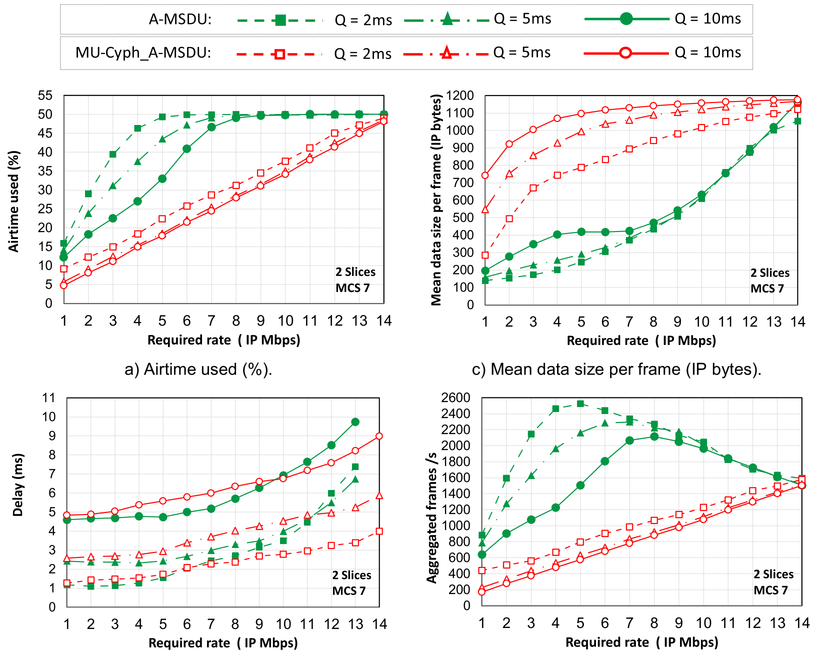
b4) Aggregated frames /s. Q=5ms

c4) Aggregated frames /s. Q=10ms

FIGURE 6. Performance results. Comparison between proposed multi-user aggregation method (MU-cyph-A-MSDU) and standard A-AMSDU in terms of airtime used (%), delay(ms), mean data size per aggregated frame (IP bytes) and aggregated frames per second.

without the limitation that they must go to the same STA as occurs in A-MSDU or A-MPDU. This trend is further confirmed by the graphs of mean frame data size, where MU-Cyph_A-MSDU exhibits significantly larger initial values and a more linear growth. In contrast, A-MSDU exhibits a more exponential growth, starting from much lower values and not exceeding 1000 bytes until the end of the range, under non-congested conditions. This highlights that, even at high transmission rates, MU-Cyph_A-MSDU takes better advantage of each transmission opportunity by aggregating more data into fewer transmissions. Note that, in both schemes, the maximum aggregation size is maintained in the saturation levels.

The advantages of MU-Cyph_A-MSDU over A-MSDU in terms of airtime are greater as long as the receiver STAs have better channel conditions (high-quality links / high rate) and the MU-Cyph_A-MSDU can use higher MCS. Comparing the test scenario where all STAs require MCS1 with the one



a) Airtime used (%).

c) Mean data size per frame (IP bytes).

b) Delay (ms).

d) Aggregated frames /s.

FIGURE 7. Performance results. Comparison between proposed multi-user aggregation method (MU-Cyph-A-MSDU) and standard A-AMSDU in terms of airtime used (%), delay(ms), mean data size per aggregated frame (IP bytes) and aggregated frames per second, for scenario MCS 7 and depends of quantum set(Q = [2ms,5ms, 10ms]).

where MCS7 is required, and considering them as extreme examples of behavior, it can be observed that in scenarios where high-quality transmissions are possible (MCS 7), the airtime savings are significant, providing a significant quantity of additional resources for other services. Indeed, Fig. 7 shows that it is possible to reach rates of up to 14 Mbps with the proposed MU-Cyph_A-MSDU scheme, which is not achievable in the case of A-MSDU (up to 13Mbps). As can be seen in Fig. 7, the absolute gain values also depend on the chosen quantum configuration. Fig. 7 shows that the differences are more noticeable for smaller quantum values. This is because the delay caused by the distribution of airtime among slices allows more packets to be aggregated in both cases as the quantum increases. The Q = 2 ms configuration provides a short-term fair-slicing environment, with only 2 ms available to aggregate packets before starting transmission in slice 0 following the slice 1 round. As a result, the average frame size is even more limited, particularly for A-MSDU. For instance, in Fig. 7 we see how for a required rate of 4Mbps and Q = 2ms only 18.42% of airtime is used with the MU-Cyph_A-MSDU scheme, whereas 43.3% of airtime is required by the standard A-MSDU aggregation. In Fig. 6, when MCS 1 is considered, the highest benefits are obtained for lower required rate. For instance, for a required rate of 2Mbps and Q = 2ms only 28.6% of airtime is used with the MU-Cyph_A-MSDU scheme compared to 42.3% when A-MSDU is considered. Results for MCS values between MCS1 and MCS7 would fall at intermediate levels of gain.

Obviously, comparing the two schemes requires a scenario in which the MCS requirements of the receiving STAs are heterogeneous, and the MCS is adapted to ensure the feasibility of reception for the MU-Cyph_A-MSDU transmission. Fig. 6 illustrates also the benefits when Lowest MCS is used in the aggregated multi-user frame. Note, however, that in this case, the advantages when parametrization of Q = 10ms is considered are limited. This is because the configuration allows a higher degree of aggregation even for A-MSDU. For A-MSDU, transmissions correspond to an approximately equal use of MCS 1, 3, and 7, whereas MU-Cyph_A-MSDU aggregation uses the lowest MCS, resulting in a distribution with a higher probability of MCS 1, followed by MCS 3 and MCS 7. As a result, although MU-Cyph_A-MSDU achieves longer aggregated frames, the increase in the length is not fully compensated due to the additional airtime required by the lower average MCS. It is clear that channel-aware grouping could provide benefits in these cases. As shown in Fig. 6, the option of aggregating packets associated with STAs in similar channel conditions offers higher advantages in terms of airtime usage compared to A-MSDU, particularly for the Q = 5 ms and Q = 10 ms configurations. For instance, with Q = 10 the previously mentioned limitations related to lowest MCS are overcome, and for a demanded rate of 4 Mbps the airtime consumption decreases from 37% to 29.7%. In the case of Q = 5, airtime consumption drops from 47.9% with A-MSDU to 33.3% with the channel-aware approach. This occurs even though the set of STAs whose traffic can be aggregated is very small—only 3. However, in the experimental results presented, it does not provide improvements in the case of Q = 2 ms compared to simply aggregating any packets. The reasons are straightforward: in the chosen test configuration, only 3 STAs can be aggregated in the channel-aware grouping case, compared to nine STAs for the lowest MCS approach. Since packet accumulation is smaller for Q = 2 ms, the increase in aggregated frame size does not compensate for the use of a lower average MCS distribution. In any case, it should be noted that the proposal is intended for scenarios where the system can benefit from aggregating traffic associated with a significant number of STAs. In cases where it is not possible to substantially increase the aggregation size, the system can fall back to standard aggregation.

Performance in terms of latency depends on the configured quantum. In both MU-Cyph_A-MSDU and A-MSDU it remains bounded within a few milliseconds with not significant differences. In all the cases, the minimum delay is bounded by the A-MSDU scheme with MCS7. The minimum delays in the experimentations are 1.2ms (Q = 2ms), 2.42ms (Q = 5ms) and 4.6ms (Q = 10ms). These values are consistent with the test scenarios. It should be noted that the first packets aggregated in slice 0 group packets generated during the previous Q ms, which implies a minimum average queuing delay of (Q/2) ms. However, starting from lower flow rate demands, throughout the airtime window assigned to slice 0, packets are aggregated with shorter delays. As a result, the average delay can be lower than (Q/2) ms, particularly for higher Q values.

Comparing now the MU-Cyph_A-MSDU and A-MSDU delay performances, the same pattern is consistently observed, regardless of the quantum Q. The delay increases more rapidly as the selected MCS for transmission decreases. A shown in Fig. 6 and 7, for all STA requiring MCS7 delays are similar, improving the performance of MU-Cyph_A-MSDU compared to A-MSDU as the rate demand increases. Delay of MU-Cyph_A-MSDU always improve results obtained with A-MSDU for the test with all STA requiring MCS1. However, despite the higher airtime consumption, A-MSDU improves performance in terms of delay when the lowest MCS is used in the scenario where receiver STA require different MCS. The difference is due to the fact that the average airtime consumed by each aggregated A-MSDU frame compared to MU-Cyph_A-MSDU is significantly higher. From the airtime perspective, the balance between aggregation size and the number of aggregated frames results in lower airtime consumption for MU-Cyph_A-MSDU. However, from the latency perspective, the aggregated packets experience higher transmission and queuing delays in the user buffer. On the contrary, when the Channel-aware grouping scheme is applied, this effect is mitigated, improving performance also in terms of delay.

In summary, the results show that MU-Cyph_A-MSDU provides an improvement over A-MSDU even when aggregation is not based on channel conditions. When aggregation is performed more intelligently, it can deliver appreciable benefits even without grouping many STAs. Moreover, the results for MCS 1 and MCS 7 suggest that if the number of STAs with similar channel conditions were higher, the advantages would be very significant. Additionally, it is important to explicitly consider the most appropriate quantum configuration. From the tests carried out, a quantum value of Q = 5 ms provides a very good balance between airtime savings and minimal delay.

Ultimately, although due to practical limitations and scope, the tests do not cover all possible combinations, the results show that the multi-user aggregation method offers a clear advantage over standard aggregation methods as A-MSDU in terms of spectral efficiency and aggregation size.

VI. CONCLUSION

The results of this work demonstrate that the proposed multi-user aggregation mechanism is a novel contribution beyond the current state of the art, providing clear improvements over conventional aggregation schemes such as A-MSDU and, by extension, A-MPDU, particularly in scenarios with high volumes of services involving small packets transmitted to multiple STAs (a few tens) with little or no mobility and very low inter-cell mobility. We have limited our work to a dense but very stable network. This assumption has a direct impact on the feasibility of rate adaptation with a high degree of reliability while maintaining acceptable levels of key management scalability and computational overhead.

The solution is compatible with existing IEEE 802.11 standards and has been validated in a realistic testbed, confirming its practical applicability in commercial network environments. The study shows that even without aggregating packets for a large number of stations, the proposed method delivers tangible benefits, while scenarios with a greater number of users under similar channel conditions could yield even more significant performance gains. In scenarios where the proposal is integrated with advanced features such as RAN slicing, the configuration of system parameters plays a critical role in balancing airtime efficiency and latency. In addition to improving efficiency, the scheme also develops the necessary data protection through the use of multi-channel transmission encryption techniques.

REFERENCES

[1] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 5: Enhancements for Higher Throughput*, IEEE Standard Std 802.11nTM, 2009.

[2] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between SystemsLocal and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications-Amendment 4: Enhancements for Very High Throughput for Operation in Bands Below 6 GHz.*, IEEE Standard Std 802.11ac-2013, Dec. 2013, pp. 1–425.

[3] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN*, IEEE Standard Std 802.11ax-2021, 2021.

[4] D. Skordoulis, Q. Ni, H.-H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour, "IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs," *IEEE Wireless Commun.*, vol. 15, no. 1, pp. 40–47, Feb. 2008.

[5] B. Ginzburg and A. Kesselman, "Performance analysis of A-MPDU and A-MSDU aggregation in IEEE 802.11n," in *Proc. IEEE Sarnoff Symp.*, Princeton, NJ, USA, Apr. 2007, pp. 1–5.

[6] Y. Lin and V. W. S. Wong, "WSN01–1: Frame aggregation and optimal frame size adaptation for IEEE 802.11n WLANs," in *Proc. IEEE Globecom*, San Francisco, CA, USA, Nov. 2006, pp. 1–6.

[7] B. S. Kim, H. Y. Hwang, and D. K. Sung, "Effect of frame aggregation on the throughput performance of IEEE 802.11n," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Las Vegas, NV, USA, Mar. 2008, pp. 1740–1744.

[8] A. Saif, M. Othman, S. Subramaniam, and N. A. W. A. Hamid, "An enhanced A-MSDU frame aggregation scheme for 802.11n wireless networks," *Wireless Pers. Commun.*, vol. 66, no. 4, pp. 683–706, Oct. 2012.

[9] X. Zhou and A. Boukerche, "AFLAS: An adaptive frame length aggregation scheme for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 855–867, Jan. 2017.

[10] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "Impact of IEEE 802.11n/AC PHY/MAC high throughput enhancements on transport and application protocols—A survey," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2050–2091, 4th Quart., 2017.

[11] S. Seytnazarov and Y.-T. Kim, "QoS-aware adaptive A-MPDU aggregation scheduler for voice traffic in aggregation-enabled high throughput WLANs," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2862–2875, Oct. 2017.

[12] I. Jabri, K. Mansour, I. Al-Oqily, and T. Ezzedine, "Enhanced characterization and modeling of A-MPDU aggregation for IEEE 802.11n WLANs," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 1, Jan. 2022, Art. no. e4384.

[13] K. Suzuki and S. Yamazaki, "Throughput maximization based on optimized frame-aggregation levels for IEEE 802.11 WLANs," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1725–1728, May 2021.

[14] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, D. Wing, M. A. M. Perumal, M. Ramalho, G. Camarillo, F. Pascual, D. R. Lopez, M. Nuñez, D. Flórez, J. A. Castell, T. de Cola, and M. Berioli, "Emerging real-time services: Optimizing traffic by smart cooperation in the network," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 127–136, Nov. 2013.

[15] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs," *IEEE Commun. Surv. Tut.*, vol. 21, no. 1, pp. 197–216, 1st Quart., 2019.

[16] D.-J. Deng, Y.-P. Lin, X. Yang, J. Zhu, Y.-B. Li, J. Luo, and K.-C. Chen, "IEEE 802.11ax: Highly efficient WLANs for intelligent information infrastructure," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 52–59, Dec. 2017.

[17] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "IEEE 802.11ax: Challenges and requirements for future high efficiency WiFi," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 130–137, Jun. 2017.

[18] J. L. H. Salazar, J. Saldaña, J. Fernández-Navajas, J. Ruíz-Mas, and G. Azuara, "Short message multichannel broadcast encryption," *Adv. Intell. Syst. Comput.*, vol. 1267, pp. 122–131, May 2020.

[19] M. McBride and C. Perkins. (Sep. 26, 2015). *Multicast WiFi Problem Statement*. Working Draft, IETF Internet-Draf. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-mcbride-mboned-wifi-mcast-problem-statement-00

[20] V. Gupta, C. Gutterman, Y. Bejerano, and G. Zussman, "Experimental evaluation of large scale WiFi multicast rate control," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2319–2332, Apr. 2018.

[21] Y. D. Park, S. Jeon, J.-P. Jeong, and Y.-J. Suh, "FlexVi: PHY aided flexible multicast for video streaming over IEEE 802.11 WLANs," *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2299–2315, Oct. 2020.

[22] S. Sengupta, H. Gupta, P. De, B. Mitra, S. Chakraborty, and N. Ganguly, "Understanding data traffic behaviour for smartphone video and audio apps," in *Proc. 8th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, Jan. 2016, pp. 1–2.

[23] X. Che and B. Ip, "Packet-level traffic analysis of online games from the genre characteristics perspective," *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 240–252, 2011.

[24] J. Saldana, J. Ruiz-Mas, J. Fernández-Navajas, J. L. S. Riaño, J.-P. Javaudin, J.-M. Bonnamy, and M. Le Dizes, "Attention to Wi-Fi diversity: Resource management in WLANs with heterogeneous APs," *IEEE Access*, vol. 9, pp. 6961–6980, 2021.

[25] A. Hernández-Solana, J. Ruiz-Mas, M. Canales, J. Fernández-Navajas, J. R. Gállego, and S. Ibánez-Alloza, "Practical challenges of implementing a hybrid ISS-QoSS RAN slicing in SDN WLAN networks," *Comput. Netw.*, early access, 2025.

[26] J. Saldana, J. Ruiz-Mas, and J. Almodóvar, "Frame aggregation in central controlled 802.11 WLANs: The latency versus throughput tradeoff," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2500–2503, Nov. 2017.

[27] J. Saldana, O. Topal, J. Ruiz-Mas, and J. Fernández-Navajas, "Finding the sweet spot for frame aggregation in 802.11 WLANs," *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1368–1372, Apr. 2021.

[28] Y. Zhu and M. Xu, "Enhancing network throughput via the equal interval frame aggregation scheme for IEEE 802.11ax WLANs," *Chin. J. Electron.*, vol. 32, no. 4, pp. 747–759, Jul. 2023.

[29] S. Seytnazarov, D. G. Jeong, and W. S. Jeon, "Performance analysis of aggregation-enabled IEEE 802.11 WLANs with variable aggregation size," *IEEE Access*, vol. 11, pp. 119373–119387, 2023.

[30] H. Tamura, D. Nobayashi, K. Kawahara, and K. Tsukamoto, "Dynamic A-MPDU adaptation method for airtime fairness in channel bonding-ready WLANs," *IEEE Access*, vol. 12, pp. 87954–87966, 2024.

[31] J. Wang, J. Wang, J. Chen, L. Bai, and J. Choi, "An efficient frame aggregation scheme for relay-aided Internet of Things networks with age of information constraints," *IEEE Trans. Mobile Comput.*, vol. 24, no. 9, pp. 9141–9152, Sep. 2025.

[32] A. Behara and T. G. Venkatesh, "Performance analysis and energy efficiency of MU-(OFDMA MIMO) based hybrid MAC protocol of IEEE 802.11ax WLANs," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6474–6490, May 2023.

[33] Y. Lv, M. P. I. Dias, L. Ruan, E. Wong, Y. Feng, N. Jiang, and K. Qiu, "Request-based polling access: Investigation of novel wireless LAN MAC scheme for low-latency e-health applications," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 896–899, May 2019.

[34] D. Candal-Ventureira, F. J. González-Castaño, F. Gil-Castiñeira, and P. Fondo-Ferreiro, "Coordinated allocation of radio resources to Wi-Fi and cellular technologies in shared unlicensed frequencies," *IEEE Access*, vol. 9, pp. 134435–134456, 2021.

[35] X. Wang, L. Wang, Y. Wang, and D. Gu, "Reliable multicast mechanism in WLAN with extended implicit MAC acknowledgment," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, May 2008, pp. 2695–2699.

[36] Z. Feng, G. Wen, C. Yin, and H. Liu, "Video stream groupcast optimization in WLAN," in *Proc. Int. Conf. Internet Technol. Appl.*, Wuhan, China, Aug. 2010, pp. 1–4.

[37] W.-S. Lim, D.-W. Kim, and Y.-J. Suh, "Design of efficient multicast protocol for IEEE 802.11n WLANs and cross-layer optimization for scalable video streaming," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 780–792, May 2012.

[38] Y. Park, C. Jo, S. Yun, and H. Kim, "Multi-room IPTV delivery through pseudo-broadcast over IEEE 802.11 links," in *Proc. IEEE 71st Veh. Technol. Conf.*, Taipei, Taiwan, May 2010, pp. 1–5.

[39] E. Coronado, R. Riggio, J. Villalon, and A. Garrido, "Joint mobility management and multicast rate adaptation in software-defined enterprise WLANs," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 2, pp. 625–637, Jun. 2018.

[40] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology—CRYPTO'93*, vol. 773, D. R. Stinson, Ed., Heidelberg, Germany: Springer, 1994, pp. 480–491.

[41] *WPA3 Specification, Version 3.4*, Wi Fi Alliance, Austin, TX, USA, 2024.

[42] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *En Advances in Cryptology—CRYPTO 2005*, vol. 3621. Heidelberg, Germany: Springer, Jul. 2005, pp. 258–275.

[43] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," in *En Advances in Cryptology—EUROCRYPT 2009*, vol. 5479, A. Joux, Ed., Heidelberg, Germany: Springer, 2009, pp. 171–188.

[44] S. Canard, D. H. Phan, D. Pointcheval, and V. C. Trinh, "A new technique for compacting ciphertext in multi-channel broadcast encryption and attribute-based encryption," *Theor. Comput. Sci.*, vol. 723, pp. 51–72, May 2018.

[45] K. Acharya, "Secure and efficient public key multi-channel broadcast encryption schemes," *J. Inf. Secur. Appl.*, vol. 51, Apr. 2020, Art. no. 102436.

[46] C. Ding, T.-I. Pei, and A. Salomaa, *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography*. Singapore: World Scientific, 1996.

[47] J. Saldana, R. Munilla, S. Eryigit, O. Topal, J. Ruiz-Mas, J. Fernández-Navajas, and L. Sequeira, "Unsticking the Wi-Fi client: Smarter decisions using a software defined wireless solution," *IEEE Access*, vol. 6, pp. 30917–30931, 2018.

[48] Radiotap. *Introduction*. Accessed: Jan. 14, 2026. [Online]. Available: https://www.radiotap.org/

[49] E. Coronado, R. Riggio, J. Villaón, and A. Garrido, "Lasagna: Programming abstractions for end-to-end slicing in software-defined WLANs," in *Proc. IEEE 19th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Chania, Greece, Jun. 2018, pp. 14–15.

[50] P. H. Isolani, N. Cardona, C. Donato, J. Marquez-Barja, L. Z. Granville, and S. Latré, "SDN-based slice orchestration and MAC management for QoS delivery in IEEE 802.11 networks," in *Proc. 6th Int. Conf. Softw. Defined Syst. (SDS)*, Rome, Italy, Jun. 2019, pp. 260–265.

[51] M. Richart, J. Baliosian, J. Serrat, J.-L. Gorricho, and R. Agüero, "Slicing in WiFi networks through airtime-based resource allocation," *J. Netw. Syst. Manage.*, vol. 27, no. 3, pp. 784–814, Jul. 2019.

[52] K. Gomez, R. Riggio, T. Rasheed, and I. Chlamtac, "On efficient airtime-based fair link scheduling in IEEE 802.11-based wireless networks," in *Proc. IEEE 22nd Int. Symp. Pers., Indoor Mobile Radio Commun.*, Toronto, ON, Canada, Sep. 2011, pp. 930–934.

JOSÉ LUIS SALAZAR-RIAÑO received the B.S. and Ph.D. degrees in mathematics from the University of Zaragoza (UZ), Spain, in 1993 and 1999, respectively. Currently, he is a Lecturer with the Department of Communications and Electronic Engineering, UZ. His research interests include modern cryptography, theory and technology of information security, electronic voting, and IP-security. He is also a member of the Aragón Institute of Engineering Research (I3A).

JULIÁN FERNÁNDEZ-NAVAJAS received the M.S. degree in telecommunications engineering from the Polytechnic University of Valencia (UPV), Spain, in 1993, and the Ph.D. degree in telecommunications engineering from the University of Zaragoza (UZ), Spain, in 2001. In 1994, he joined the Department of Electronics Engineering and Communication, UZ, where he is currently an Associate Professor and a member of I3A. His research interests include communication networks, with an emphasis on wireless networks and distributed multimedia systems.

ÁNGELA HERNÁNDEZ-SOLANA received the M.S. and Ph.D. degrees in telecommunications engineering from the Polytechnic University of Catalonia (UPC), Spain, in 1997 and 2005, respectively. She joined the Department of Electronics Engineering and Communications, University of Zaragoza (UZ), in 1999, where she is currently an Associate Professor and a member of I3A. Her current research interests include wireless communications, with an emphasis on radio resource management and wireless sensor networks.

JOSÉ RUIZ MAS received the M.S. degree in telecommunications engineering from the Polytechnic University of Catalonia (UPC), Spain, in 1991, and the Ph.D. degree in telecommunications engineering from the University of Zaragoza (UZ), Spain, in 2001. He joined the Department of Electronics Engineering and Communications, UZ, in 1994, where he is currently an Associate Professor and a member of I3A. His research interests include communication networks, with an emphasis on wireless networks and distributed multimedia systems.

GUILLERMO AZUARA-GUILLÉN received the Ph.D. degree in information and communication technologies and in mobile networks from the University of Zaragoza (UZ), Spain, in 2013, through the doctoral program. He joined the Department of Electronics Engineering and Communications, UZ, in 2000, where he is currently a Lecturer and a member of I3A. His research interests include communication networks and cybersecurity.

RAMÓN CAJAL-PÉREZ received the bachelor's and M.S. degrees in telecommunications engineering from the University of Zaragoza (UZ), Spain, in 2023 and 2025, respectively. In 2024, he joined the Department of Electronics Engineering and Communications, UZ, where he is currently a N4-Novel Investigator. His current research interest includes network communication.