When and How to Aggregate Message Authentication Codes on Lossy Channels?

Eric Wagner[1,2], Martin Serror[1], Klaus Wehrle[2], and Martin Henze[3,1]

[1] Cyber Analysis & Defense, Fraunhofer FKIE,
Wachtberg, Germany {firstname.lastname}@fkie.fraunhofer.de
[2] Communication and Distributed Systems, RWTH Aachen University,
Aachen, Germany {lastname}@comsys.rwth-aachen.de
[3] Security and Privacy in Industrial Cooperation, RWTH Aachen University,
Aachen, Germany henze@spice.rwth-aachen.de

Abstract. Aggregation of message authentication codes (MACs) is a proven and efficient method to preserve valuable bandwidth in resource-constrained environments: Instead of appending a long authentication tag to each message, the integrity protection of multiple messages is aggregated into a single tag. However, while such aggregation saves bandwidth, a single lost message typically means that authentication information for multiple messages cannot be verified anymore. With the significant increase of bandwidth-constrained lossy communication, as applications shift towards wireless channels, it thus becomes paramount to study the impact of packet loss on the diverse MAC aggregation schemes proposed over the past 15 years to assess when and how to aggregate message authentication. Therefore, we empirically study all relevant MAC aggregation schemes in the context of lossy channels, investigating achievable goodput improvements, the resulting verification delays, processing overhead, and resilience to denial-of-service attacks. Our analysis shows the importance of carefully choosing and configuring MAC aggregation, as selecting and correctly parameterizing the right scheme can, e.g., improve goodput by 39 % to 444 %, depending on the scenario. However, since no aggregation scheme performs best in all scenarios, we provide guidelines for network operators to select optimal schemes and parameterizations suiting specific network settings.

Keywords: Message Authentication Code · MAC Aggregation · IoT

1 Introduction

With the proliferation of the (industrial) Internet of Things (IoT), more and more battery-operated devices, such as sensors and actuators, rely on wireless communications. Consequently, the number of devices sharing the same transmission medium (with a fixed capacity) is growing, imposing increasingly stringent bandwidth constraints on IoT applications [29]. At the same time, wireless communication further amplifies the need to adequately secure transmitted messages [26], most notably to ensure the integrity of transmitted critical information [8], which would have prevented *e.g.*, the 2015 and 2016 cyberattacks on the Ukrainian power grid [33]. However, establishing integrity protection requires additional bandwidth to transmit authentication tags, thus conflicting with the already hard-to-reach constraints of IoT communication. Therefore, a vital research topic for industry and academia centers around the question of how to use the shared limited transmission resources efficiently and still provide adequate security [25].

As a result, many efforts across protocol stacks have been proposed to reduce bandwidth overhead. Prominent examples include 6LoWPAN header compression [20] or, more recently, the record layer headers of DTLS 1.3 [22] and Compact TLS 1.3 [21]. Such protocol improvements can however not address the inherent overhead necessary to provide integrity protection. Considering, *e.g.*, desirable 128-bit security requires the integration of a 16-byte authentication tag into the message's payload. Moreover, since (industrial) IoT protocols such as IEEE 802.15.4, LoRaWAN, or Bluetooth Low Energy often rely on short messages, such Message Authentication Codes (MACs) typically occupy a significant portion of each message and, in some cases, do not even fit [18].

For at least 15 years, the well-established and time-proven concept of MAC aggregation has been known to alleviate these limitations [14]. The idea is simple yet effective: Instead of protecting the integrity of each message individually, a single authentication tag is responsible for protecting the integrity of multiple messages. Given a reliable channel, this approach works flawlessly and can be reduced to a trade-off between saved bandwidth and the verification delay for received messages: Aggregating integrity protection of more and more messages reduces the induced overhead until it becomes negligible but implies that the receiver has to wait for the reception of all messages affected by the aggregation before being able to check their integrity, resulting in significant delays if too many authentication tags are aggregated.

Over the years, different MAC aggregation schemes have been proposed to address weaknesses [11,15,13], split authentication tags over multiple messages [18], or provide progressive security guarantees [3,17,31]. And while various implementations of security concepts, such as message authentication [28], have been evaluated and compared by literature, such analyses of MAC aggregation schemes in realistic wireless, and thus lossy, settings are practically non-existent. Most importantly, current evaluations of MAC aggregation schemes neglect that losing a single message from a set of messages with aggregated authentication tags may have cascading effects depending on the chosen MAC aggregation scheme. This phenomenon becomes increasingly relevant as more and more communication transitions to low-bandwidth wireless, and thus lossy, channels in a diverse set of applications such as smart cities, underwater communication, or the (industrial) IoT [9]. Thus, MAC aggregation is arguably becoming even more critical for lossy channels than for its initial setting of reliable communication. However, research, thus far, did not provide sufficient address under which circumstances MAC aggregation on lossy channels is sensible and how to unlock its full potential. This knowledge is, however, crucial to optimally utilize scarce bandwidth in wireless scenarios with an ever-growing number of participating devices.

To address these shortcomings, this paper addresses the hitherto neglected analysis of the performance of relevant MAC aggregation schemes in the presence of lossy channels. We consider realistic wireless (industrial) IoT communication scenarios, which suffer from scarce transmission resources and significant packet losses, where we compare the performance of existing MAC aggregation schemes. Our analysis is thus a valuable contribution for security practitioners and researchers: On the one hand, it allows identifying suitable aggregation schemes depending on the considered scenario, and on the other hand, it reveals current shortcomings, which lay the foundation for identifying more effective approaches. Ultimately, we want to answer the questions of when MAC aggregation is sensible on lossy channels and how this aggregation should be performed by making the following contributions:

- We investigate the achievable goodput improvements of all MAC aggregation scheme known to us under various parameterizations in synthetic and real-world scenarios (Section 3 and Section 4);
- We further analyze the impact of MAC aggregation on decisive factors such as verification delay, processing times, memory cost, and the susceptibility to denial-of-service attacks (Section 5); and
- Finally, we provide actionable guidelines to help in deciding when and how current MAC aggregation schemes are best deployed (Section 6).

Availability Statement. To help in the decision process of which, if any, MAC aggregation scheme should be deployed in a concrete scenario, our tool to compare MAC aggregations schemes in concrete scenarios is available at: `https://github.com/fkie-cad/mac-aggregation-analysis-tool`

2 MAC Aggregation on Lossy Channels

Achieving integrity protection is a significant challenge in bandwidth-constrained environments. Even the tiniest message requires an authentication tag of several bytes (e.g., 16 bytes for 128-bit security), thus occupying considerable space in each message. MAC aggregation schemes, as presented in this section, try to alleviate this overhead by distributing the burden of authentication over multiple messages. In the following, we first define MACs (Section 2.1) before formally introducing the concept of MAC aggregation (Section 2.2). To conclude, we introduce the existing MAC aggregation schemes (Section 2.3) and motivate research into their applicability in lossy conditions (Section 2.4).

2.1 Message Authentication Codes

Message Authentication Codes (MACs) allow two communication partners to verify the integrity of exchanged messages using a pre-shared secret k [7]. This key k can be derived dynamically through a key exchange protocol or hardcoded at both communicating entities. To authenticate a message m, the sender uses the tag generation algorithm $Sig_k(m)$ to generate the corresponding authentication tag t. Upon reception of a message, the verification algorithm $Vrfy_k(m, t)$ enables the recipient to evaluate whether the received tag is valid. Typically, this verification is done by computing the tag $t^* = Sig_k(m^*)$ for the received message m^* and comparing it to the received tag t. A MAC scheme is considered secure if it is computationally infeasible to generate a (m,t)-pair that $Vrfy_k(\cdot)$ would accept without knowing the secret k. This requirement can be achieved by, *e.g.*, using keyed hash functions such as `HMAC-SHA256` to compute t. Thus, MACs provide integrity protection for communication channels, where they prevent any attacker not knowing k from undetectably manipulating the content of transmitted messages.

2.2 MAC Aggregation to Combat Bandwidth Scarcity

Traditional MAC schemes consume significant bandwidth in constrained environments. Over 15 years ago, the concept of MAC aggregation was promoted to combat these limitations [14]. The idea is elegant and effective: Instead of authenticating each message individually, a single tag is responsible for protecting the integrity of multiple messages. Thus, the overhead of each tag is distributed over multiple messages, saving valuable bandwidth.

Formally, MAC aggregation schemes can be defined as an extension of traditional MAC schemes. In a traditional MAC scheme, the tag t_i is computed over and transmitted alongside message m_i. For MAC aggregation schemes, the aggregated tag t_i^{agg}, which is transmitted alongside m_i, is computed by aggregating the integrity protection of multiple messages $m_{i-d}(d \in \mathcal{D})$ with an additional keyless function $Agg(\cdot)$, such that $t_i^{\text{agg}} = Agg(t_{i-d} | d \in \mathcal{D})$. We say that $\mathcal{D} \subset \mathbb{N}_0$ is the set of dependencies of a MAC aggregation scheme and, *e.g.*, $2 \in \mathcal{D}$ means that the tag t_{i-2} is included in the computation of the aggregated tag t_i^{agg}. Vice versa, the integrity of message m_{i-2} is protected by tags t_i.

Thus, aggregated authentication tags protect multiple messages. At the same time, each message is potentially protected by multiple tags as each (potentially shortened) tag may only be responsible for providing a fraction of the overall targeted security level. Since each tag aggregates integrity protection for multiple messages, aggregated MAC schemes result in, on average, shorter tags. In this context, the dependencies $\mathcal{D}$ describe how the reception of one message influences the verifiability of tags and the authenticity of surrounding messages. We say that if an aggregated MAC scheme has the dependencies $\mathcal{D}$, the generation and verification of tag t_i require knowledge of $\{m_{i-d} | d \in \mathcal{D}\}$, as $t_{i-d} = Sig_k(m_{i-d})$. Consequently, a message m_i blends into all tags $\{t_{i+d} | d \in \mathcal{D}\}$, and a tag t_i protects the integrity of all messages $\{m_{i-d} | d \in \mathcal{D}\}$.

A specific MAC aggregation scheme defines the underlying MAC scheme, the dependencies $\mathcal{D}$, and the aggregation function $Agg(\cdot)$. In the following, we consider a simple XOR of authentication tags for the aggregation function, *i.e.*, $t_i^{\text{agg}} = Agg(t_{i-d} | d \in \mathcal{D}) = \bigoplus_{d \in \mathcal{D}} t_{i-d}$. This aggregation of tags is efficient and has been shown to be secure [4]. If, for example, t_i and t_j provide 128-bit integrity protection for m_i and m_j, then $t^{\text{agg}} = t_i \oplus t_j$ provides 128-bit integrity protection for both messages m_i and m_j. However, the security of this aggregation function requires that the chosen MAC function is pseudorandom and includes a nonce for replay protection to prevent mix-and-match attacks within one set of jointly authenticated messages [11]. Consequently, MAC schemes based on universal hashing, such as UMAC [6], should not be used in combination with XOR-based MAC aggregation[4]. Most prominent MAC schemes, such as `HMAC-SHA256`, can, however, be securely used with XOR-based MAC aggregation if used in combination with nonce-based replay protection.

2.3 Introducing Existing MAC Aggregation Schemes

After formalizing the concept of MAC aggregation in Section 2.2, we now introduce the different sets of MAC aggregation schemes, grouped by their choice of dependencies $\mathcal{D}$ and computation of t^{agg}. We do, however, not focus on the exact aggregation function or the underlying MAC scheme, as those choices do not impact the scheme's susceptibility to packet loss. Under these aspects, we present all four classes of aggregation that cover, to the best of our knowledge, all proposed schemes. For this presentation, we assume XOR-based aggregation with `HMAC-SHA256` as a suitable MAC scheme (including an appended nonce for replay protection).

Traditional (Trad.): To quantify the performance of existing MAC aggregation schemes, we compare them to the baseline performance of traditional MAC schemes. Therefore, we consider a traditional MAC scheme that authenticates each message m_i with an individual tag t_i. This computation thus solely depends on m_i, *i.e.*, $\mathcal{D} = \{0\}$. As we target 128-bit security, the HMAC-SHA256 is truncated to 16 B.

Aggregated MAC (Agg(n)): The most prominent scheme is aggregated MAC as introduced in 2008 [14] and later extended to prevent reordering attacks [11], allow messages to occur multiple times [15], and identify faulty messages in an aggregate [13]. For these schemes, a tag t^{agg} is only appended to each n-th message, where n is the parameter for how many messages' authentication tags are aggregated together. For our evaluation, we consider the aggregation of two, four, eight, and sixteen tags, *i.e.*, $n \in 2, 4, 8, 16$ to cover a range of different parameterizations. For every n-th message, a tag is then computed by XORing the authentication tags of all considered messages, as formalized in the following:

$$t_i^{\text{agg}} = \bigoplus_{i-n<k\leq i} t_k \quad \text{for } i \equiv 0 \pmod{n}$$

[4] BP-MAC [32] (based on a Carter-Wegman construction), for example, is insecure if used in combination with XOR-based MAC aggregation. As each bit is authenticated individually and replay protection is only provided through a blinding tag, an attack can undetectably swap the x-th bits' values of two messages.

Compound MAC (Comp(n)): As the tags computed by $\texttt{Agg}(\cdot)$ are too long for some applications, Compound MAC is proposed that splits across multiple messages [18]. Thus, each message carries a shortened authentication tag, the length of which is inversely proportional to the number of aggregated messages, *i.e.*, $|t| = {}^{128}/_{n}$. For our analysis, we again consider $n \in 2, 4, 8, 16$. We formalize $\texttt{Comp}(\cdot)$ in the following, where $t[a : b]$ means the chunk from the a-th to the b-th bit of tag t:

$$t_i^{\text{agg}} = \bigoplus_{\lfloor \frac{i}{n} \rfloor \cdot (n-1) \leq k < \lfloor \frac{i}{n} \rfloor \cdot n} t_k[(k \bmod n) \cdot |t| : ((k+1) \bmod n) \cdot |t|]$$

Sliding Window-based Progressive MACs (SW(n,o)): Progressive MAC has been introduced to provide initially reduced security that is improved eventually upon message reception [3,17,24]. Therefore, each message is protected by a shortened tag that also verifies the integrity of the previous n messages. As $\texttt{SW}(\cdot)$ is not equipped to provide full security under packet loss, it can be compensated by additionally considering an overprovisioning factor o. This factor defines in percent how much security may be extended beyond the target, *i.e.*, $o = 100$ means that messages may have 256-security at the expense of longer tags as $|t^{\text{agg}}| = {}^{128}/_{n} \cdot (1 + {}^{o}/_{100})$. Here, we select a number of parameter combinations that perform best under various scenarios. The tag computation of $\texttt{SW}(\cdot)$ can be formalized as follows:

$$t_i^{\text{agg}} = \bigoplus_{i-n<k\leq i} t_k[k \cdot |t| : (k+1) \cdot |t|]$$

Randomized and Resilient Dependency Distribution (R2D2(n,g,o)): To address weaknesses of $\texttt{SW}(\cdot)$ in the presence of packet loss, $\texttt{R2D2}(\cdot)$ introduces dependencies that bound the effect a dropped packet can have on the verifiability of any other message [31]. Therefore, the parameter g is introduced, which defines how much security any message loses at most if a surrounding packet is lost, *i.e.*, $g = 1$ in combination with 2 B long packets means that any message can lose at most 16 bit of security. Furthermore, $\texttt{R2D2}(\cdot)$ randomizes the concrete dependency set $\mathcal{D}$ and assigns a different set to each bit of a tag. The final aggregate tag t^{agg} is thus a juxtaposition of bit-long tags t_j^{agg} and is defined as:

$$t_j^{\text{agg}} = \bigoplus_{0\leq k<|\mathcal{D}_j|} t_{i-\mathcal{D}_j[k]}[k * |t| + i]$$

with $\mathcal{D}_j[n]$ representing the n-th entry of j-th bit dependency set $\mathcal{D}_j$.

2.4 Interplay of Lossy Channels and MAC Aggregation

MAC aggregation can bring benefits to a wide range of constrained environments, such as Industrial Control Systems (ICSs), smart homes, smart city, or underwater networks. However, we see these targeted environments quickly shifting towards more and more lossy communication with protocols such as ZigBee, Sigfox, Bluetooth Low Energy, or LoRaWAN, to name only a few. This shift can significantly impact the performance of MAC aggregation schemes, especially considering Packet Error Rates (PERs) that can rise to 10 % and above for certain scenarios [31]. With MAC aggregation, a lost packet means that the receiver cannot authenticate the initially transmitted message and all other messages that depend on it. Arguably, MAC aggregation has become even more critical in the lossy settings than for reliable communications since these networks more often expose bandwidth constraints due to the high number of nodes sharing the same transmission medium. LoRaWAN, for example, is often limited to less than 10 kB or even 1 kB of throughput per hour per device. Despite this stringent requirement of conserving bandwidth in lossy networks, no accurate performance analysis of MAC aggregation in this context has been conducted thus far to the best of our knowledge. In the following section, we provide the first such analyses for the different MAC aggregation schemes presented in Section 2.3.

3 Synthetic Measurements

We begin our analyses of MAC aggregations schemes by looking at synthetic measurements of simulated wireless channels. These measurements give us fine control over channel quality and payload length to investigate how these parameters influence the different MAC aggregation schemes. In Section 3.1, we first describe our setup before diving into the influence of channel quality and payload lengths in Sections 3.2 and 3.3, respectively. Finally, we look at the established challenge of determining optimal payload lengths for given channel qualities under the additional constraint that the received data must be authenticated.

3.1 Simulation Setup

For our synthetic measurements, we use the network simulator `ns-3` (version 3.37), giving us fine-grained control over the underlying communication channel. As communication protocol, we choose the IEEE 802.15.4 protocol commonly used in constrained wireless environments and included in `ns-3`, where we consider the most compact header of 5 B. For payload lengths varying between 1 B and the maximum supported 115 B, we simulate the communication between two static antennas placed 25 m apart and extract binary loss traces of which transmitted packets have been correctly received or not. We additionally vary the transmit power varying between −21 dBm and −16 dBm using 0.1 dBm steps to increase the signal-to-noise ratio progressively, thus improving the channel quality and reducing PER. We only transmit each message once and do implement acknowledgments or retransmission, as these features are not always available. For all combinations of transmit power and payload length, we simulated the transmission of 10 000 packets, of which we selected a random sequence of 5000 packets for each of the following analyses. In a standalone simulation, we

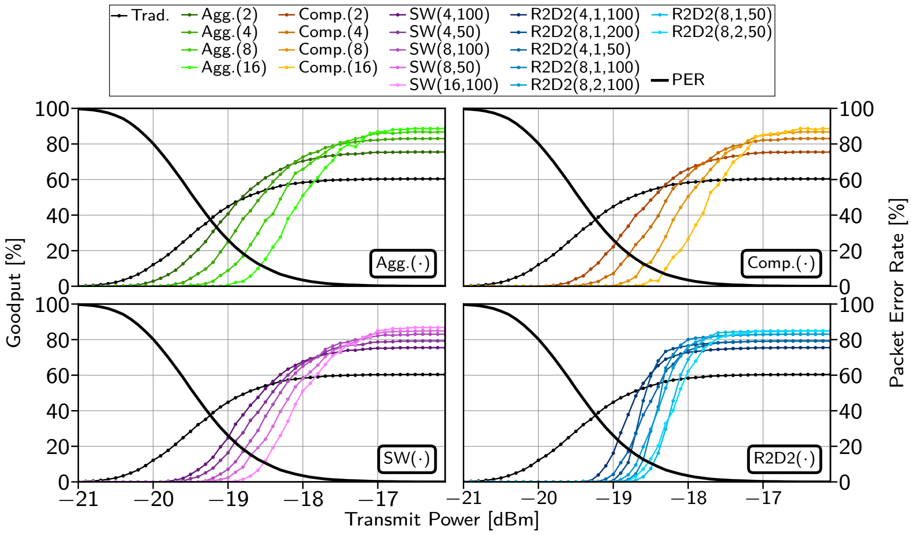

Fig. 1. Traditional MAC performs best with high packet error as all received data can be verified. For medium PERs, the aggregation of two tags with `Agg`(·) and various `R2D2`(·) parameterizations are preferable, while the aggregation of more messages with the simpler `Agg`(·) and `Comp`(·) scheme are is desirable with low PERs.

then implement the behavior of the different classes of MAC aggregation schemes and their selected parameterizations to extract which messages eventually become authenticated for a given binary loss trace. Our measurements focus on the achieved goodput by the different MAC aggregation schemes, where goodput is the ratio of received (and authenticated) payload bytes (*i.e.*, excluding header and authentication tag) to the number of transmitted bytes. We initially focus on goodput as performance metrics as it directly measures how efficient the transmission channel is utilized, the improvement of which is the main goal of MAC aggregation.

3.2 Influence of Channel Quality on Goodput

For an initial understanding of the different MAC aggregation schemes, we fixed the payload length to 48 B and gradually increased the transmission power, resulting in a slowly decreasing PER from 100 % to 0 %. Figure 1 shows our results.

We observe that all aggregation schemes exhibit the same general sigmoidal behavior: As the PER decreases, the achieved goodput increases slowly before increasing quickly and then leveling off. This behavior can be explained by the behavior of the packet delivery ratio (*i.e.*, the opposite of the packet error rate), which also increases first slowly and then rapidly as the channel quality improves. The interesting differences between the schemes and their parameterizations are thus defined by when and how goodput increases as the channel improves.

For the different aggregation schemes, we see that the maximally achieved goodput correlates inversely with the number of aggregated tags (parameter n). As a higher n results in, on average, shorter tags, a better maximal goodput can be achieved due to less overhead. Similarly, the transmit power where the goodput of the different schemes starts to take off also correlates with n. The increasing likelihood can explain this observation that at least one of the tags in an aggregate cannot be computed as the set of aggregated messages becomes larger. Thus, the parameterizations with the higher bandwidth saving potential also require a better channel (*i.e.*, lower PER) to be beneficial over more conservative parameterizations. Consequently, traditional MACs perform best with high PERs while exposing the overall worst goodput as PER approaches 0 %.

Comparing the performance of the different aggregation schemes, we observe that all schemes tend towards the same discrete goodput dictated by their average tag length. However, the goodput provided by $\texttt{Agg}(\cdot)$, $\texttt{Comp}(\cdot)$, and $\texttt{SW}(\cdot)$ increases earlier but more slowly with increasing transmit power in contrast to $\texttt{R2D2}(\cdot)$, which suddenly jumps up once the channel is good enough. The behavior of $\texttt{R2D2}(\cdot)$ can be explained by ideally distributing the effects of packet loss to surrounding messages, such that if security levels for a few messages become good enough to consider the message authenticated, surrounding messages are close to the threshold as well. Overall, for transmit powers up −18.9 dBm (PER=18.5 %), traditional MACs perform best as they are not handicapped by the many lost packets. Then, the aggregation of two messages with $\texttt{Agg}(\cdot)$ is best until, between −18.3 dBm (PER=8.5 %) and −17.1 dBm (PER=0.4 %), there are different parameterizations of $\texttt{R2D2}(\cdot)$ that perform best. As the PER reduces further, the selected scheme becomes, however, less critical, and the differences for the same average tag length are marginal. Here, simpler schemes with no overprovisioning, such as $\texttt{Agg}(\cdot)$ and $\texttt{Comp}(\cdot)$, are usually preferable. Consequently, it mostly depends on the channel quality, which aggregation scheme and parameterization achieve the best goodput.

3.3 Influence of Payload Length on Goodput

In Section 3.2, we consider a fixed payload length and slowly increase the transmit power to improve the signal-to-noise ratio. To better understand the behavior of the different MAC aggregation schemes, we now vary the payload length for a fixed transmit power of −18.3 dBm, where we have realistic PER between 1.5 and 10.9 % across the payload length range. We show our results in Figure 2.

With changing transmit power, we observe the same characteristics in the goodput curves of all aggregation schemes. Goodput first quickly increases before slowly dropping after reaching a maximum. This phenomenon can be explained by the overlapping effects of reduced relative overhead of authentication tags and growing numbers of unverifiable tags due to raised PERs with increased payload lengths. Thus, selecting the best MAC aggregation scheme depends on the underlying channel quality, packet lengths, and resulting variable PERs.

Furthermore, we can see that not all aggregation schemes can be employed for short payload lengths. Traditional MAC and $\texttt{Agg}(\cdot)$ append 16-byte authen-

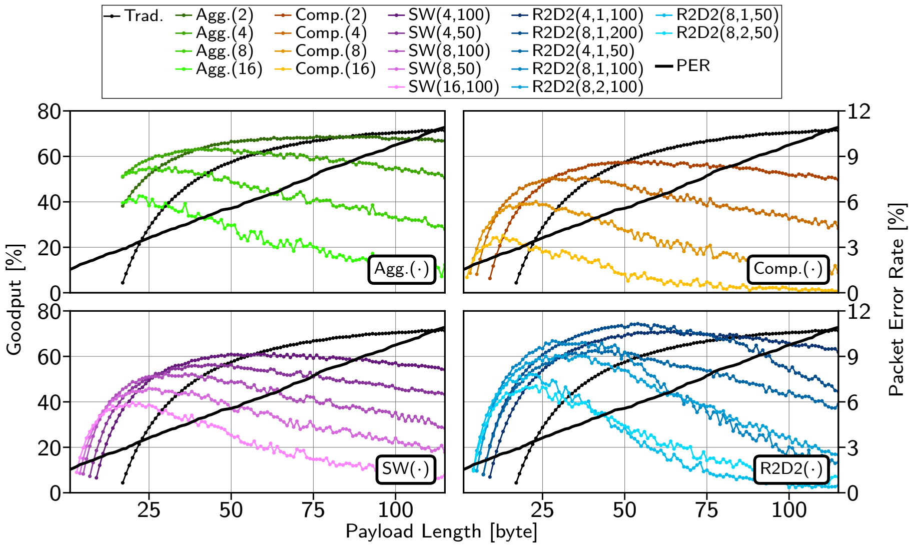

Fig. 2. For larger payloads, the PER increases and the relative overhead of authentication tags decrease. Therefore, different schemes and parameterizations are optimal depending on payload lengths.

tication tags (to a fraction of all) messages and thus require payload lengths of at least 17 B. The other aggregation schemes append a shortened tag to all messages, but the size of these tags also dictates how small messages can be. Thus, if transmitted packets can only carry a few bytes of payload, such as the unreliable CAN bus protocol, which supports at most 8-byte payloads and has no header fields intended for integrity protection, the choice of available MAC aggregation scheme shrinks.

Moreover, we observe different optimal payload lengths w.r.t. to goodput for the distinct schemes and parameterizations. While using the maximal payload length of 114 B yields the optimal goodput of 71.7 % for traditional MACs, the overall maximal goodput of 74.4 % is achieved by R2D2(8,1,200) with a payload length of 54 B. Hence, investigating the combined impact of packet lengths and MAC aggregation under varying conditions is essential to determine optimal network configurations in novel deployments.

3.4 Optimal Packet Lengths for Authenticated Data

Prior results indicate that considering the MAC aggregation scheme is crucial when optimizing packet lengths for a given channel. This search for optimal payload length gathered interest in the past [23,30,1,16] to make use of limited bandwidth availability or optimize the lifetimes of battery-powered devices. As resource-constrained devices consume most of their power for wireless transmissions [27], optimizing goodput is essential for improving device lifetimes. Assum-

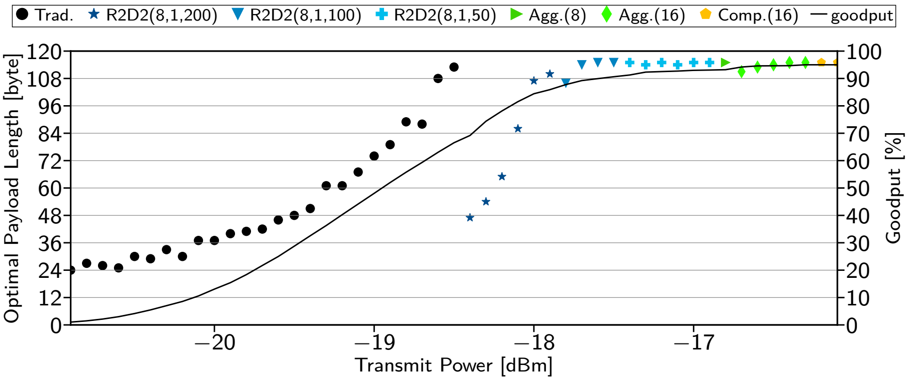

Fig. 3. Different MAC (aggregation) schemes achieve a higher goodput as channel quality improves under optimal payload lengths. Unintuitively, changing a scheme can result in a reduced optimal payload length even if the channel improves.

ing constant energy consumption for each transmitted bit at a given transmit power, the optimal combination of payload lengths and MAC aggregation scheme also optimizes device lifetimes. These packet length optimizations, thus far, only looked at received data and not received *and authenticated* data. Assuming the imperative requirement of authenticated data, we search for the optimal payload lengths to optimize goodput across varying channel qualities, considering the different MAC aggregation schemes. Our results are shown in Figure 3.

For low transmission powers, *i.e.*, low signal-to-noise ratio, we see that traditional MACs,*i.e.*, no aggregation, perform best. This behavior can be explained by the initially high PER, even for small messages, such that aggregated tags have a high risk of being composed of at least one message that did not arrive. Here, the behavior observed in our setup matches related work [23,30,1,16] in that optimal payload lengths are initially short and then slowly increase as the transmit power is increased.

As the transmission channel improves, message aggregation starts to pay off since the benefits of shorter tags outweigh the risk of received data that cannot be authenticated. Here, initially, between -18.4 and −17.2 dBM, R2D2(·) under various parameterizations performs best. However, the best MAC aggregation scheme does not only change with better channels; the optimal payload also decreases on each change before slowly increasing again. Therefore, the optimal payload length for a transmit power of −18.5 dBM is 113 B (with traditional MACs), but for a slightly higher transmit power of −18.4 dBM, it drops down to 47 B (for R2D2(8,1,100)). We can observe this same phenomenon for other changes between MAC aggregation schemes, and it is more or less pronounced depending on the header sizes, where the static overhead of larger packet headers dampens the drop in optimal packet sizes.

Overall, we can see that the average tag lengths of the optimal schemes shrink for higher transmission power. Looking at the achieved goodput by the

Scenario	Duration	Protocol	Header	Data	#pkts	PER	Src
ICS	8 hours	IEEE 802.15.4	11 B	20 B	57648	4.79%	[12]
Office	22 hours	BLE	10 B	32 B	79032	3.22%	[12]
Smart City (sta.)	131 days	LoRaWAN	13 B	16 B	18790	1.97%	[5]
Smart City (mob.)	250 days	LoRaWAN	13 B	24 B	17415	7.09%	[19]
Underwater	327 min	GUWMANET	31 bit	16 B	334	16.46%	[10]

Table 1. Limited bandwidth availability for integrity protection is a serious challenge across a wide range of lossy environments.

respective optimal scheme, we see a sigmoid curve that instead of leveling off at 82.5 % if only using traditional MACs, the different MAC aggregation schemes boosts this achievable goodput to 95.0 % as the PER approaches 0. However, it must also be understood that transmitting with optimal payload lengths is often not an option in practice. Here, the (established) applications and protocols often dictate the payload lengths, *e.g.*, a sensor may only have a single reading that should be transmitted quickly and thus has no other data to fill into the payload. Therefore, and because real wireless channels change over time, it is necessary to investigate MAC aggregation in real-world scenarios.

4 MAC Aggregation in Real-World Scenarios

Thus far, we have analyzed MAC aggregation schemes in controlled synthetic environments. While these analyses gave us insights into the behavior and nuances of the different schemes, they do not necessarily represent the entire story for realistic deployments. Here, we often have predetermined payload lengths dictated by available data or protocol specifications. Also, channel qualities vary dynamically over time, especially if some communication partners are mobile. In the following subsection, we first introduce distinct real-world scenarios, which we subsequently use to evaluate and compare the performance of the MAC aggregation schemes (*cf.* Section 2.3) under realistic conditions.

4.1 Description of the Scenarios

For our realistic measurements, we rely on network traces collected from real-world scenarios. Each trace has constant payload lengths and transmission configurations, and we extract a binary loss trace of which transmitted packets have been correctly received or not. This trace is then fed into our simulation to analyze the MAC aggregation schemes. We summarize the scenarios in Table 1 and briefly introduce them in the following subsections.

Industrial Control System (ICS) Scenario. For the first scenario, we look at a measurement campaign of wireless communication in a 3600 m^2 production hall with nearly a billion transmitted packets [12]. We select a single representative link from the various configurations using the IEEE 802.15.4 protocol with a payload length of 20 B. Our trace covers a total of 8 h of traffic on a typical workday with an overall PER of 4.79 %. In this scenario, we observe primarily short bursts of packet loss with channel quality changing mostly over longer time windows (hours), while phases of high error rates (upwards of 50 %) are possible for several minutes.

Office Scenario. With the same measurement setup as for the ICS scenario, wireless links between nodes placed in various office rooms on a single floor have been measured [12]. Here, we select a Bluetooth Low Energy (BLE) communication link with 32 B payloads over a 22 h window during a workday. We observe a relatively constant error distribution with short error bursts of a few packets each and an overall PER of 3.22 %.

Smart City (Stationary) Scenario. Our first smart city scenario is based on the LoED dataset [5], where nine LoRaWAN gateways were placed in central London. We focus on the 18790 packets transmitted by a single stationary sender and received by any of the gateways. With an overall PER of 1.97 %, we see primarily isolated packet loss due to long idle times between two transmissions, and the channel only experiences long-term changes in quality over several days, potentially due to altering weather conditions.

Smart City (Mobile) Scenario. In this scenario, mobile LoRaWAN senders transmit to a total of nine stationary gateways for 250 days. Specifically, the sender was mounted to the top of a garbage truck driving through a 200 km^2 area in the city of Bonn [19]. We observe burstier errors and overall channel qualities changing significantly over days and weeks. The burstiness is likely due to the sender quickly entering and exiting the line of sight of a gateway, while the long-term changes changing presumably again relate to the weather conditions.

Underwater Scenario. Finally, we consider acoustic underwater communication, with a trace of 334 16-byte messages being transmitted over 327 min between two stationary nodes placed in the sea [10]. The measurements were conducted during moderately rough weather conditions. Despite an overall high PER of 16.46 %, most of these errors occurred during long bursts interspersed with periods of high packet delivery rates.

4.2 Evaluating MAC Aggregation in Realistic Scenarios

We now analyze MAC aggregation schemes in the different realistic scenarios introduced in the previous section. These scenarios are characterized by dynamic channels, differing communication protocols, and prespecified header and payload lengths. For each scenario, we analyze the goodput (*i.e.*, the amount of

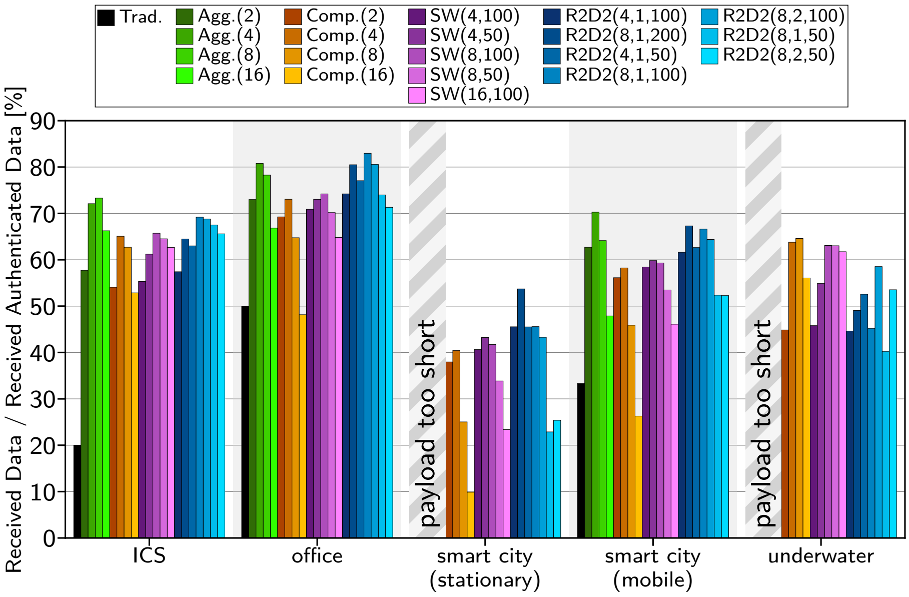

Fig. 4. Different MAC aggregation schemes and parameterizations perform best, depending on the payload lengths, error burstiness, and overall PERs, such that the right scheme selection is non-trivial but crucial for the optimal use of constrained channels.

received and authenticated data) in Figure 4. We express the goodput as a percentage of the total amount of received payload data if no integrity protection was used. For the urban (static) and underwater scenarios, traditional MACs and Agg(·) cannot be included since the tags do not fit into the available payload.

We see different MAC aggregation schemes performing best in the synthetic measurements, depending on the investigated scenario. Payload lengths influence the concrete parameterization but do not directly correlate with which scheme performs best under the relatively small variations observed across the different scenarios. The best-performing MAC aggregation scheme thus depends primarily on two factors: overall PER and burstiness.

For the industry and urban (mobile) scenario, where overall PER is low and burstiness relatively high, Agg(·) performs best. During a burst, most packets in one set of aggregated messages are lost, while otherwise, most sets are received entirely and can be authenticated. For the office and urban (static) scenarios, R2D2(·) performs best due to the high PER and the short error bursts, where often only a single packet is lost. However, higher PERs do not immediately mean that R2D2(·) performs best (until traditional MACs are more favorable), as suggested by the synthetic scenarios. The long burst, where no traffic passes, in combination with a relatively good delivery ratio otherwise mean that SW(·) and Comp(·) perform best in the underwater scenario. Overall, the best MAC aggregation scheme can achieve relative improvements of up to 24.2 % better goodput compared to the second best scheme.

However, more important than selecting the scheme is using the correct parameterization. If the wrong parameters are used for the best-performing MAC aggregation scheme, performance can drop by an average between 14.0 and 57.4 % in the worst case. With the PER of the different scenarios ranging between 1.97 and 16.46 %, we have parameterizations that result in average tag lengths of 4 B performing best, which is more or less in line with the synthetic measurements from Section 3.2.

Overall, we see that PER and error burstiness play a significant role in finding the best scheme and parameterizations. Due to relatively small variance in payload lengths across the scenarios, we, however, cannot confirm its low impact in selecting the best schemes. Nevertheless, we know that the potential gains achieved through MAC aggregation shrink with larger payloads. Most importantly, we can conclude that adequate parameterization is more important than finding the best MAC aggregation scheme. Ultimately, both optimizations have a non-negligible effect on the achievable goodput.

5 Beyond Goodput as Evaluation Metric

Optimizing the goodput of MAC aggregation is the main goal for most scenarios. However, other effects must also be considered when choosing the MAC aggregation scheme, such as verification delay, processing overhead, and susceptibility to jamming attacks. In the following subsections, we compare the different MAC aggregation schemes (*cf.* Sec. 2.3) w.r.t. these effects.

5.1 Average Delay until Authentication

First, we look at the authentication delay for the different MAC aggregation schemes. Traditional authentication tags can be verified immediately upon message reception, so no delay occurs due to waiting for additional data. With MAC aggregation, on the other hand, we need to wait until all messages depending on a specific tag have been received to verify it, which might introduce significant delays. To analyze these effects, we plotted the delay from the measurements on all traces from Section 4.1 as a CDF in Figure 5.

We see major differences in the behavior of the different aggregation schemes for these measurements. `Agg`(·) and `Comp`(·) periodically verify a set of prior messages together, such that a range of different delays occur with the same frequency. The concrete span of possible delay is then proportional to the parameter n of how many tags are aggregated together.

`SW`(·), on the other hand, verifies messages continuously with an almost constant delay. This delay only varies if some messages get lost, which incurs a verification delay for surrounding messages. This behavior is beneficial for applications requiring periodic messages with practically no jitter, *e.g.*, control algorithms in ICSs relying on a constant delay of the received information. `R2D2`(·)

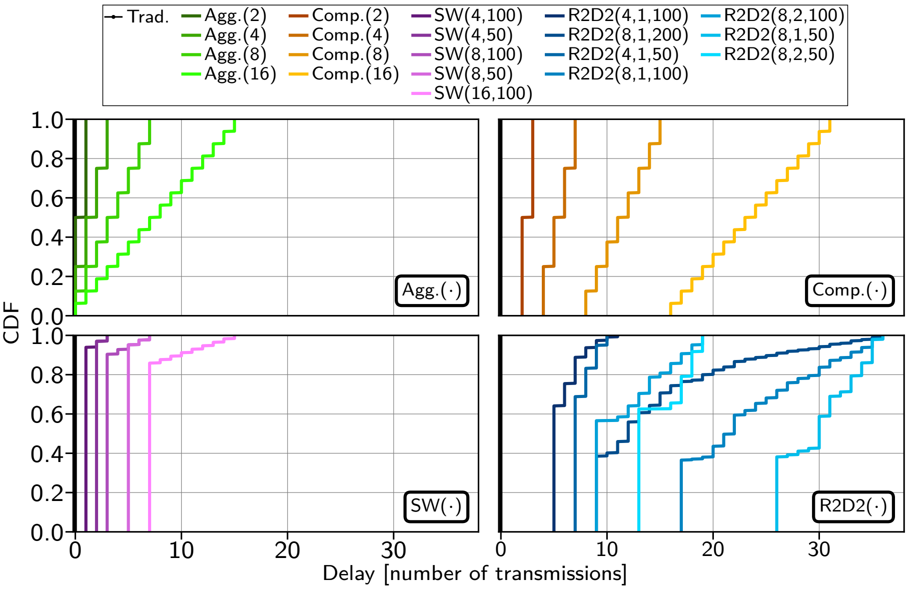

Fig. 5. Verification delay is an inherent drawback of MAC aggregation. For scenarios where verification delay is critical, SW(·) does, however, provide highly consistent delays which control algorithms can thus anticipate.

shows similar behavior for about half of all the received messages, while the rest have increasing delays. Again, the packet loss is responsible for higher delays, but since R2D2(·) distributed the effects of packet losses over multiple packets, more of them experience delayed verification. Moreover, the magnitude of these delays correlated with the overprovisioning factor o, allowing late authentication for messages that could otherwise not be authenticated.

In summary, the average delay until authentication of the different authentication schemes strongly differs. While Agg(·) offers, on average, the lowest delays, SW(·) has the most constant delays. On the other hand, R2D2(·) offers the best goodput for many scenarios with higher PER while messages have higher and more varying verification delays. Selecting the best aggregation scheme according to this delay thus depends on which balance the concrete application scenario demands between the goodput reduction and the type of verification delay.

5.2 Performance and Memory Overhead

Many of the considered scenarios involve resource-constrained IoT devices where substantial additional processing and memory overhead from the MAC aggregation scheme could significantly impact performance. Hence, we measure and compare the processing delay and memory overhead for tag computation and buffering by the different schemes. We conducted the analysis on the Arm Cortex

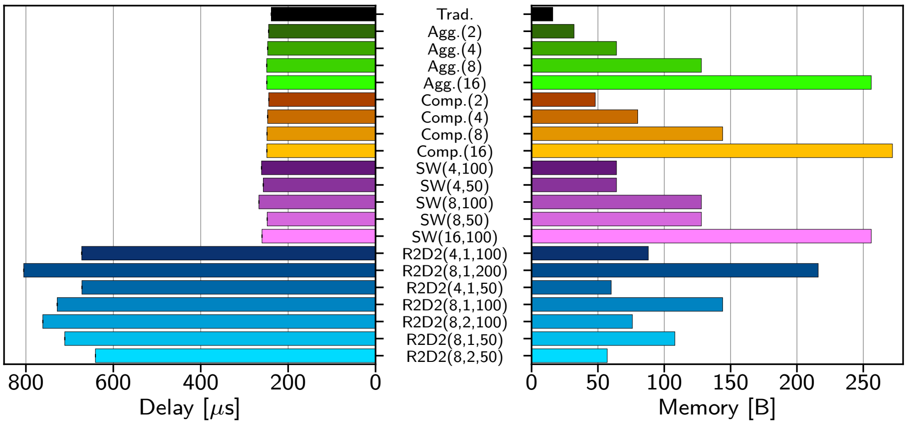

Fig. 6. Only R2D2($\cdot$) introduces significant processing overhead over traditional MACs. Memory overhead, on the other hand, is mostly dependent on how many tags are aggregated together but so small that it should rarely be a decisive factor.

M3 processor of a Zolertia RE-Mote board, a common choice to evaluate realistic resource-constrained hardware. As a baseline, we capture the time to authenticate a single 32 B message with hardware-accelerated HMAC-SHA256, which is the underlying MAC scheme used for the aggregation schemes as well. We averaged the average processing times over 16 tag generations (not all schemes do the same computations for each message) and repeated this measurement 30 times. For the memory overhead, we measure the memory necessary to buffer tags before their aggregation, as all other memory overhead is implementation-dependent and is mostly optimized away by the compiler. The results of both measurements are presented in Figure 6.

Regarding processing times, we see only marginal overhead for all aggregation schemes except R2D2($\cdot$). There, we have a 168 to 237 % increase in processing times compared to the baseline, where the differences across parameterizations are mostly insignificant. This overhead stems from the bitwise processing of R2D2($\cdot$), which requires a significant amount of XOR and bitshift operations. This processing overhead is, however, mostly only impactful for applications that run on slower hardware and have tight latency requirements, especially considering that the sender *and* receiver must conduct this additional processing.

We see a different picture across the MAC aggregation schemes for the memory overhead. For Agg($\cdot$), Comp($\cdot$), and SW($\cdot$), the needed memory depends on the message history. More tags must be stored concurrently for shorter aggregated tags, resulting in higher memory overhead. Here, the bitwise processing of R2D2($\cdot$) helps to partially process tags when new messages arise. Consequently, the memory depends mainly on the overprovisioning factor and less on the number of aggregated tags. The magnitude of the required additional memory for

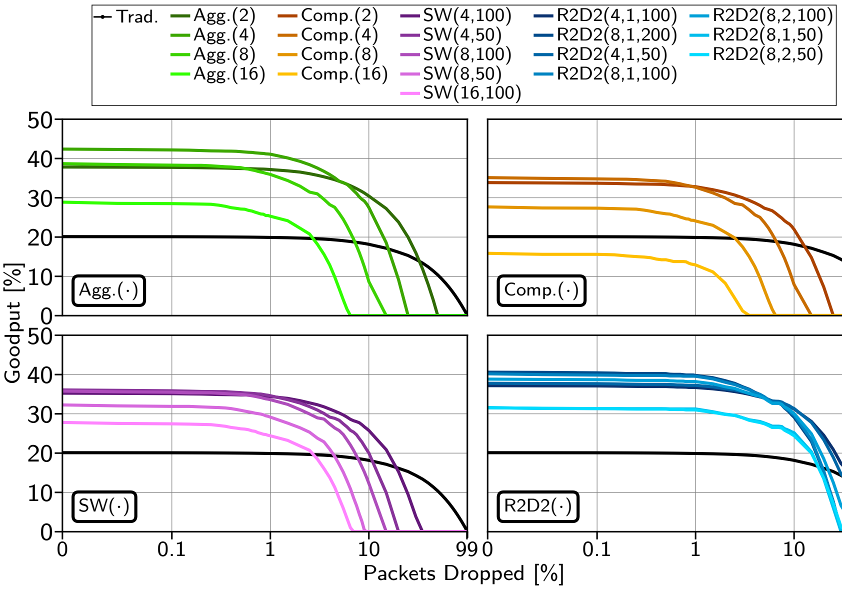

Fig. 7. `R2D2`(·) shows significantly increased resilience to denial-of-service attacks through selective jamming, especially if an attacker jams less than 10 % of messages to remain stealthy or conserve energy.

MAC aggregation schemes is, however, small enough that it should rarely influence the decision on which scheme should be deployed.

5.3 Resilience to Adversarial Interference

In our final analysis, we compare the resilience of MAC aggregation schemes to selective jamming attacks. Selective jamming refers to jamming specific messages to prevent their correct reception which enables stealthy and energy-saving attacks as dropped packets are hardly distinguishable from random packet loss [34,2]. In the context of MAC aggregation schemes, a sophisticated attacker can amplify the effects of a denial-of-service attack due to the employed MAC aggregation. For example, for `Agg`(16), it suffices to jam every 16th packet to reduce the (authenticated) goodput of the channel to zero.

For our measurements, we considered the trace from the urban (mobile) scenario introduced previously, as its payload is large enough for all schemes, and urban settings provide easy access to potential attackers. For each aggregation scheme, we developed the optimal jamming attack strategy to minimize the goodput at the receiver. In Figure 7, we show how the achieved goodput of the different aggregation schemes is impacted by increased attacking capabilities.

The x-axis represents the number of overall dropped packets in percent on a logarithmic scale. For traditional authentication, we see, as expected, that the channel can still transmit authenticated data as long as not the entire channel is jammed. In general, we note that shorter average tags are more susceptible to selective jamming attacks, as each tag requires many received messages to become verifiable. Considering the shortest tags (n=16) for `Agg`(·), `Comp`(·), and `SW`(·), we see that dropping between 27 and 29 % targeted packet already suffices to prevent all data transmission over the channel.

The behavior of `R2D2`(·) requires, however, a separate analysis since one of the protocol's design goals is resilience against jamming attacks. Therefore, the exact dependencies between tags and messages are kept secret, such that attackers can only design their strategy to inflict the most damage for the average dependency selection. Furthermore, the design of `R2D2`(·) explicitly distributes the effects of packet losses (malicious or not) over many packets, thus cushioning the impact of selective jamming. Hence, up to 15 % of packets need to be dropped to reduce goodput by even 20 %. However, once a critical mass of packet loss occurs, such distribution no longer suffices for compensation, and the goodput quickly drops.

Overall, we can say that `R2D2`(·) is the most resilient scheme in the presence of a selective jammer. Considering our entire analysis, no scheme is an outright winner, and each scheme has its benefits. To summarize these findings, guide operators toward the right MAC aggregation scheme, and identify open research questions, we provide general recommendations in the following section.

6 Guidelines on Employing MAC Aggregation

In general, MAC aggregation shows promising potential to boost available bandwidth on lossy channels for various scenarios. However, not every scenario benefits from MAC aggregation compared to traditional MACs. More importantly, choosing the correct scheme and parameters is decisive in answering the questions of *when* and *how* to use MAC aggregation. Therefore, in the following, we deepen this discussion towards providing general guidelines on employing MAC aggregation based on our empirical measurements.

6.1 When to Use MAC Aggregation on Lossy Channels?

From our analysis, it is evident that MAC aggregation reliably improves goodput for relatively high PERs of 10 % or below. In cases where the PER is higher, it is often more beneficial to rely on traditional MACs or, at most, aggregate MACs for no more than two messages (*i.e.*, setting the parameter n to 2). However, for high PERs due to long error bursts where hardly any traffic arrives, MAC aggregation can still be beneficial (*cf.* Section 4.2).

Furthermore, we investigated the relationship between payload lengths and the resulting benefits of MAC aggregation. For instance, in scenarios involving 200 B payloads and minimalistic 5 B headers, a MAC aggregation scheme aggregating 16 tags (*i.e.*, $n = 16$) could still generate a 7.3 % goodput improvement. Consequently, we conclude that MAC aggregation, in general, offers the most substantial benefits for short payload lengths, up to a few hundred bytes, and moderate PERs of up to 10 %. As substantiated by the real-world scenarios (*cf.* Section 4.1), this is precisely the kind of communication that occurs in many (industrial) IoT scenarios, leading to the question of how to use MAC aggregation in such scenarios to gain the most benefit.

6.2 How to Employ MAC Aggregation on Lossy Channels?

In our evaluations of the goodput improvements that different MAC aggregation schemes and parameterizations can bring in real-world scenarios (*cf.* Section 4.2), we have seen that no aggregation scheme is a clear-cut winner (even when solely focusing on goodput as an evaluation metric). Moreover, we have seen that the correct parameterization for a given scenario is crucial to achieving optimal performance. These observations thus warrant a more nuanced discussion of when to use which MAC aggregation scheme and with which parameters.

Focusing solely on goodput, we see that generally `R2D2`$(\cdot)$ achieves the highest performance for PER between 0.4 and 8.5 %, especially when packet errors occur as short bursts. For lower PERs and traffic with longer error bursts, the better performance and simplicity of `Agg`$(\cdot)$ is often preferable. If the periodic 16-byte tags for `Agg`$(\cdot)$ are not supported by the application (*e.g.*, due to fixed message sizes), `Comp`$(\cdot)$ is a good alternative to realize a constant tag size across all messages. Considering the parameterizations, a high n has the potential to realize better goodput, but only if the PER is relatively low. For the overprovisioning factor o of `SW`$(\cdot)$ and `R2D2`$(\cdot)$, 100 is usually the best or least a decent choice. `R2D2`$(\cdot)$'s g-factor is best set to 1 in those scenarios where `R2D2`$(\cdot)$ achieves the best goodput. Overall, `SW`$(\cdot)$ rarely outperforms the other schemes if only considering goodput since it is not designed for lossy communication [31]. Nevertheless, it can still be a sensitive choice when also considering *e.g.*, verification delays.

One disadvantage of MAC aggregation compared to traditional MACs is the inherent verification delay which we investigated in Section 5.1. This delay occurs as most messages cannot be verified directly upon reception and thus need to be buffered or processed optimistically [31], *i.e.*, processed under the assumption of being genuine before full integrity verification. This risk can be reduced by the two progressive schemes `SW`$(\cdot)$ and `R2D2`$(\cdot)$, already providing some, yet reduced, security guarantees immediately upon message reception. Furthermore, if an application requires complete message verification, `SW`$(\cdot)$ provides deterministic verification delays, beneficial for real-time control.

Concerning other potential dimensions for selecting the best MAC aggregation scheme for a given scenario, memory overhead is so small that it should rarely be a decisive factor. When interested in optimizing processing overhead, only `R2D2`$(\cdot)$ shows a clear disadvantage (*cf.* Section 5.2) compared to the other aggregation schemes. Finally, if resilience to denial-of-service attacks through selective jamming is essential, `R2D2`$(\cdot)$ shows clear advantages over the other schemes. However, if another scheme must be used (*e.g.*, due to the excessive processing overhead of `R2D2`$(\cdot)$), then lowering the parameter n can reduce the effects of attacks at the cost of reduced goodput under normal operation.

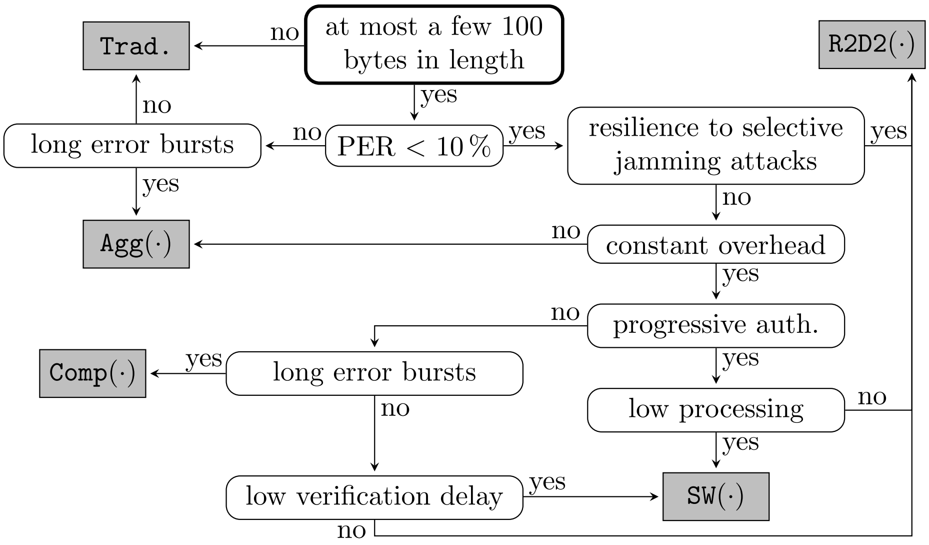

Fig. 8. The optimal MAC aggregation scheme depends on many different characteristics. This decision diagram assists in this selection process.

6.3 Selecting an MAC Aggregation Scheme

We observe that often many dimensions must be considered to decide when and how to perform MAC aggregation. To help operators in their decision process, we provide two forms of assistance. First, we provide a decision diagram to select the right MAC aggregation scheme in Figure 8 based on basic network characteristics and feature demands. Secondly, and for more detailed analysis, we provide an evaluation tool[5] to aid further in this decision process. Our evaluation tool takes as input the header and payload lengths as well as an example binary loss trace, *i.e.*, a series of 1s and 0s for received and dropped packets, respectively. It provides a comparison of all MAC aggregation schemes and their parameterizations (as analyzed in this paper) for the given scenario. In combination with these tools, our guidelines support operators in deciding when and how to employ MAC aggregation and help researchers to identify further opportunities to optimize existing MAC aggregation schemes.

7 Conclusion

MAC aggregation effectively saves valuable bandwidth in resource-constrained networks by shifting integrity protection from single to multiple packets. However, as shown in this paper, the potential benefits of MAC aggregation strongly depend on the individual network scenario. In particular, the effects of (bursty)

[5] `https://github.com/fkie-cad/mac-aggregation-analysis-tool`.

packet losses, as experienced in wireless communication, severely impact the performance of MAC aggregation. Therefore, we specifically address the research question of *when* and *how* to aggregate MACs by comparing existing aggregation schemes in synthetic and real-world scenarios. Our empirical results indicate that, in general, MAC aggregation is particularly effective in scenarios with relatively reliable communication (*i.e.*, with PERs below 10 %) and for short payload lengths (*i.e.*, below a few hundred bytes). Most importantly, however, correctly parameterizing MAC aggregation is even more critical than choosing the right scheme. Moreover, other optimization metrics than goodput may limit the choice of applicable MAC aggregation schemes and thus need to be considered. With our detailed guidelines and our public evaluation tool, we intend to support operators in deciding when and how to employ MAC aggregation for their applications and researchers to improve MAC aggregation further, ultimately strengthening security even under adverse networking conditions.

Acknowledgements

We thank Thomas Hänel, Michael Rademacher, and Michael Goetz for providing access to their datasets. We thank Jan Bauer, Michael Rademacher, and our anonymous reviewers for valuable feedback that improved this paper. This work is partially funded by the project MUM2, partially funded by the German Federal Ministry of Economic Affairs and Climate Action (BMWK) with contract number 03SX543B managed by the Project Management Jülich (PTJ), and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. The authors are responsible for the contents of this work.

References

1. Akbas, A., Yildiz, H.U., Tavli, B., Uludag, S.: Joint Optimization of Transmission Power Level and Packet Size for WSN Lifetime Maximization. IEEE Sensors Journal **16**(12) (2016). https://doi.org/10.1109/JSEN.2016.2548661
2. Aras, E., Small, N., Ramachandran, G.S., Delbruel, S., Joosen, W., Hughes, D.: Selective Jamming of LoRaWAN using Commodity Hardware. In: Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous). ACM (2017). https://doi.org/10.1145/3144457.3144478
3. Armknecht, F., Walther, P., Tsudik, G., Beck, M., Strufe, T.: ProMACs: Progressive and Resynchronizing MACs for Continuous Efficient Authentication of Message Streams. In: Proceedings of the Conference on Computer and Communications Security (CCS). ACM (2020). https://doi.org/10.1145/3372297.3423349
4. Bellare, M., Guérin, R., Rogaway, P.: XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions. In: Proceedings of the Annual International Cryptology Conference (CRYPTO). Springer (1995). https://doi.org/10.1007/3-540-44750-4_2

5. Bhatia, L., Breza, M., Marfievici, R., McCann, J.A.: Dataset: LoED: The LoRaWAN at the Edge Dataset. In: Proceedings of the Third Workshop on Data: Acquisition To Analysis (2020)
6. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and secure message authentication. In: Proceedings of the Annual International Cryptology Conference (CRYPTO). Springer (1999). https://doi.org/10.1007/3-540-48405-1_14
7. Boneh, D., Shoup, V.: A graduate course in applied cryptography (2023)
8. Castellanos, J.H., Antonioli, D., Tippenhauer, N.O., Ochoa, M.: Legacy-Compliant Data Authentication for Industrial Control System Traffic. In: Proceedings of the International Conference on Applied Cryptography and Network Security (ANCS). Springer (2017). https://doi.org/10.1007/978-3-319-61204-1_33
9. Chen, Y., Kunz, T.: Performance Evaluation of IoT Protocols under a Constrained Wireless Access Network. In: Proceedings of International Conf. on Selected Topics in Mobile & Wireless Networking (MoWNeT). IEEE (2016). https://doi.org/10.1109/MoWNet.2016.7496622
10. Dol, H., Blom, K., Sotnik, D., Nissen, I., Otnes, R., Komulainen, A., Campagnaro, F.: EDA-SALSA: Development of a self-reconfigurable protocol stack for robust underwater acoustic networking. In: IEEE/MTS Oceans (2023). https://doi.org/10.1109/OCEANSLimerick52467.2023.10244330
11. Eikemeier, O., Fischlin, M., Götzmann, J.F., Lehmann, A., Schröder, D., Schröder, P., Wagner, D.: History-Free Aggregate Message Authentication Codes. In: Proceedings of the International Conference on Security and Cryptography for Networks (SCN). Springer (2010). https://doi.org/10.1007/978-3-642-15317-4_20
12. Hänel, T., Brüggemann, L., Loske, F., Aschenbruck, N.: Long-Term Wireless Sensor Network Deployments in Industry and Office Scenarios. In: Proceedings of the 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE (2021). https://doi.org/10.1109/WoWMoM51794.2021.00024
13. Hirose, S., Shikata, J.: Non-adaptive Group-Testing Aggregate MAC Scheme. In: Proceedings of the International Conference on Information Security Practice and Experience (ISPEC). Springer (2018). https://doi.org/10.1007/978-3-319-99807-7_22
14. Katz, J., Lindell, A.Y.: Aggregate Message Authentication Codes. In: Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA). Springer (2008). https://doi.org/10.1007/978-3-540-79263-5_10
15. Kolesnikov, V.: MAC Aggregation with Message Multiplicity. In: International Conference on Security and Cryptography for Networks (SCN). Springer (2012). https://doi.org/10.1007/978-3-642-32928-9_25
16. Kurt, S., Yildiz, H.U., Yigit, M., Tavli, B., Gungor, V.C.: Packet Size Optimization in Wireless Sensor Networks for Smart Grid Applications. IEEE Transactions on Industrial Electronics **64**(3) (2016). https://doi.org/10.1109/TIE.2016.2619319
17. Li, H., Kumar, V., Park, J.M., Yang, Y.: Cumulative Message Authentication Codes for Resource-Constrained IoT Networks. IEEE Internet of Things Journal (2021). https://doi.org/10.1109/JIOT.2021.3074054
18. Nilsson, D.K., Larson, U.E., Jonsson, E.: Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes. In: Proceedings of the 68th Vehicular Technology Conference (VTC-Fall). IEEE (2008). https://doi.org/10.1109/VETECF.2008.259
19. Rademacher, M., Linka, H., Horstmann, T., Henze, M.: Path Loss in Urban LoRa Networks: A Large-Scale Measurement Study. In: Proceedings

of the 94th Vehicular Technology Conference (VTC-Fall). IEEE (2021). https://doi.org/10.1109/VTC2021-Fall52928.2021.9625531

20. Raza, S., Trabalza, D., Voigt, T.: 6LoWPAN Compressed DTLS for CoAP. In: Proceedings of the 8th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE (2012). https://doi.org/10.1109/DCOSS.2012.55
21. Rescorla, E., Barnes, R., Tschofenig, H., Schwartz, B.: Compact TLS 1.3. Internet-Draft, IETF (2023)
22. Rescorla, E., Tschofenig, H., Modadugu, N.: The Datagram Transport Layer Security (DTLS) Protocol Version 1.3. RFC 9147, IETF (2022)
23. Sankarasubramaniam, Y., Akyildiz, I.F., McLaughlin, S.: Energy efficiency based packet size optimization in wireless sensor networks. In: Proceedings of the First International Workshop on Sensor Network Protocols and Applications (SNPA). IEEE (2003). https://doi.org/10.1109/SNPA.2003.1203351
24. Schmandt, J., Sherman, A.T., Banerjee, N.: Mini-MAC: Raising the bar for vehicular security with a lightweight message authentication protocol. Vehicular Communications **9** (2017). https://doi.org/10.1016/j.vehcom.2017.07.002
25. Seferagić, A., Famaey, J., De Poorter, E., Hoebeke, J.: Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things. Sensors **20**(2) (2020). https://doi.org/10.3390/s20020488
26. Serror, M., Hack, S., Henze, M., Schuba, M., Wehrle, K.: Challenges and Opportunities in Securing the Industrial Internet of Things. IEEE Transactions on Industrial Informatics **17**(5) (2021). https://doi.org/10.1109/TII.2020.3023507
27. Shaikh, F.K., Zeadally, S.: Energy harvesting in wireless sensor networks: A comprehensive review. Renewable and Sustainable Energy Reviews **55** (2016). https://doi.org/10.1016/j.rser.2015.11.010
28. Simplicio Jr, M.A., De Oliveira, B.T., Margi, C.B., Barreto, P.S., Carvalho, T.C., Näslund, M.: Survey and comparison of message authentication solutions on wireless sensor networks. Ad Hoc Networks **11**(3) (2013)
29. Vitturi, S., Zunino, C., Sauter, T.: Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G. Proceedings of the IEEE **107**(6) (2019). https://doi.org/10.1109/JPROC.2019.2913443
30. Vuran, M.C., Akyildiz, I.F.: Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks. In: Proceedings of the Conference on Computer Communications (INFOCOM). IEEE (2008). https://doi.org/10.1109/INFOCOM.2008.54
31. Wagner, E., Bauer, J., Henze, M.: Take a Bite of the Reality Sandwich: Revisiting the Security of Progressive Message Authentication Codes. In: Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks. ACM (2022). https://doi.org/10.1145/3507657.3528539
32. Wagner, E., Serror, M., Wehrle, K., Henze, M.: BP-MAC: Fast Authentication for Short Messages. In: Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec). ACM (2022). https://doi.org/10.1145/3507657.3528554
33. Whitehead, D.E., Owens, K., Gammel, D., Smith, J.: Ukraine Cyber-Induced Power Outage: Analysis and Practical Mitigation Strategies. In: Proceedings of the Conference for Protective Relay Engineers (CPRE). IEEE (2017). https://doi.org/10.1109/CPRE.2017.8090056
34. Wilhelm, M., Martinovic, I., Schmitt, J.B., Lenders, V.: Short paper: reactive jamming in wireless networks: how realistic is the threat? In: Proceedings of the fourth ACM conference on Wireless network security (WiSec). ACM (2011). https://doi.org/10.1145/1998412.1998422