Byzantine Reliable Broadcast on Partially Connected Networks under Message Adversaries

***Abstract*—We consider the problem of implementing Message-Adversary-Tolerant Byzantine Reliable Broadcast (MBRB) on top of a network with an arbitrary topology, in the presence of both Byzantine nodes and a Message Adversary. The network consists of n distinct nodes where (i) up to t nodes may behave arbitrarily (Byzantine faults), and (ii) a message adversary can drop up to d messages for each local broadcast. We first investigate the solvability of the MBRB problem under different types of message adversaries and we prove lower bounds for different validity conditions. We show that such bounds are tight as we demonstrate the existence of a simple protocol for solving the MBRB problem. We then validate these theoretical findings by implementing and testing BRB protocols under various network topologies using Quantas, a discrete-time event simulator. We analyze how performance metrics— including average delivery time, fraction of correct nodes delivering the message, and communication overhead — vary as the adversarial power increases (i.e., by tuning t and d).**

***Index Terms*—Reliable Communication, Byzantine Failures, Message Adversary, Multi-hop Networks**

I. Introduction

Ensuring reliable communication (i.e., the ability of nodes to exchange information despite network uncertainty or faults) is a fundamental challenge in distributed systems. Although straightforward in benign environments, this task becomes considerably more complex under Byzantine failures, where processes may behave arbitrarily by altering, discarding, or generating fake messages [11], [10]. These risks are not just theoretical, but reflect real-world concerns in decentralized systems exposed to adversarial behavior and unpredictable conditions. At the heart of many distributed communication primitives lie two key abstractions: *Reliable Communication* (RC) and *Byzantine Reliable Broadcast* (BRB). RC guarantees that if a correct sender broadcasts a message m, then all correct processes eventually deliver m. Conversely, BRB removes the assumption of a correct sender and ensures that all correct nodes agree on the same message, even if the sender itself is Byzantine.

The Byzantine Generals Problem [13] inspired decades of research on BRB. Bracha and Toueg [5], [6] formalized BRB in asynchronous systems with authenticated links, proposing the well-known three-phase protocol (Send, Echo, Ready) for fully connected networks. Dolev [9], [10] later addressed arbitrary graphs by introducing RC abstraction, proving that RC is achievable if and only if the communication graph is $(2t+1)$-connected (where t is the number of Byzantine nodes). Bonomi et al. [4] showed that combining Dolev's RC with Bracha's BRB yields a correct BRB protocol for multi-hop networks.

Beyond Byzantine faults, recent research has investigated the impact of *network-level failures* through the concept of *message adversaries* (MA). Originally introduced by Santoro and Widmayer [14], [15] in synchronous settings, the MAs model adversarial behaviors that can arbitrarily drop or suppress messages according to predefined rules. As an example, a "tree MA" may allow only messages traveling through an (unknown) spanning tree, which can change at every communication round. Let us note that the MA failure abstraction differs substantially from Byzantine faults; indeed, Byzantine failures capture arbitrary behaviors occurring at the process level (that may cause some message loss) while the abstraction of an MA captures failures happening at the network communication layer, causing message losses even between pairs of correct processes.

Albouy et al. [3], [1], [2] recently combined these two adversarial dimensions, introducing a *dual-adversary model* that captures both Byzantine processes and a MA capable of suppressing up to d messages per local broadcast, meaning that whenever a nodes propagates a message to his neighbors, the MA is able of blocking up to d of those messages. They established tight feasibility bounds and proposed practical BRB algorithms for fully connected asynchronous networks. These results form the theoretical basis of our work, which extends the study of RC and BRB primitives over *multi-hop* topologies and evaluates their feasibility and performance under similar adversarial assumptions.

The rest of the paper is organized as follows. Section II surveys related work. Section III formalizes our system model and problem statement, followed by a consolidated table summarizing feasibility and impossibility results. Section IV derives lower bounds under the three message-adversary classes (MA1–MA3). Section V presents a flooding-based RC primitive in the authenticated (PKI) setting and proves its guarantees, while Section VI develops an unauthenticated counterpart. Section VII reports our Quantas-based experimental evaluation across multiple topologies and adversary parameters. Section VIII provides concluding remarks.

II. Related Work

The Reliable Communication (RC) and Byzantine Reliable Broadcast (BRB) problems have been extensively studied in the context of Byzantine fault-tolerant distributed systems. Dolev [9], [10] introduced the RC abstraction for arbitrary graphs, proving that RC is solvable if and only if the network is $(2t + 1)$-connected. His algorithm ensures that messages propagate along at least $t+1$ node-disjoint paths, guaranteeing

delivery despite Byzantine interference. Bonomi et al. [4] later demonstrated that combining Dolev's RC with Bracha's BRB [5], [6] yields a correct BRB protocol for multi-hop networks, provided the RC connectivity conditions and Bracha's Byzantine threshold are satisfied.

Recent work has expanded this classical model to include additional network-level failures, notably message adversaries (MA) capable of suppressing messages sent by correct nodes. Two contributions by Albouy et al. are particularly relevant to our work, as they provide both theoretical foundations and algorithms that we directly build upon.

Albouy et al. [2] proposed the first modular construction of a signature-free BRB algorithm, called *SignFreeK2LCast*, resilient to both Byzantine processes and a message adversary that can suppress up to d messages per broadcast. Their work introduces a new communication abstraction that clearly separates forwarding and delivery conditions, enabling the reconstruction of existing Byzantine-only BRB algorithms into MA-tolerant versions. Using this abstraction, they achieved delivery to at least $\ell \geq \left\lceil n - t - \frac{(n-t)d}{n-3t-d} \right\rceil$ correct nodes, where ℓ is the delivery power (the number of correct nodes guaranteed to deliver the broadcast message), n is the total number of nodes in the system, t is the maximum number of Byzantine nodes, and d is the message adversary's power (the maximum number of messages it can suppress per broadcast). This guarantee holds whenever $n > 3t+2d+2\sqrt{td}$. Moreover, when $d = 0$, the resulting algorithms are more communication-efficient than their original Byzantine-only counterparts.

In a subsequent work, Albouy et al. [1] introduced a cryptographic approach, by the name of *MBRBbroadcast*, that leverages erasure codes and vector commitments to drastically reduce communication costs. Assuming $n > 3t + 2d$, it guarantees delivery to at least $\ell \geq n - t - (1 + \epsilon)d$ correct nodes, for any arbitrarily small $\epsilon > 0$. The key innovation is to split the encoded message into fragments, allowing nodes that reconstruct the message to retransmit verified fragments efficiently without requiring the sender's signature on each fragment individually.

Both works assume fully connected asynchronous networks. By contrast, our study investigates the same dual-adversary model in *multi-hop topologies*, where sparse connectivity may significantly increase the power of the message adversary. We also introduce additional adversarial classes, such as node isolation and edge removal, to further explore the limits of reliable communication in realistic network settings.

III. System Model and Problem Statement

Processes and time. We consider a distributed system composed of a set $V = \{p_1, \ldots, p_n\}$ of n processes (also called *nodes*), each one identified by a unique integer identifier. Processes collaborate to run a distributed protocol $\mathcal{P}$ and such a protocol can be executed multiple times during the whole system lifetime. Time is measured according to a fictional global clock spanning the range of integers and not accessible by processes. We consider an asynchronous system that has no guarantees about communication delays.

Byzantine nodes. We assume that up to t processes can be Byzantine faulty [11] (i.e., they can exhibit arbitrary, possibly malicious, behavior). Processes that are not Byzantine faulty are said to be *correct*.

Communication Model. Processes communicate by exchanging messages. Nodes are connected between them and are arranged in an undirected graph $G = (V, E)$, where the set V represents the vertices of the graph, and E is the set of edges such that $(p, q) \in E$ if there exists a direct communication channel between p and q.

Definition 1 (Node degree)**.** *Given a connected system communication graph $G = (V, E)$, the* degree *of a node $p_i \in V$ (denoted $\delta(p_i)$) is the number of edges that are incident to p_i.*

Definition 2 (Vertex Cut)**.** *Given a connected system communication graph $G = (V, E)$, a* vertex cut *C of G is a subset of vertices whose removal disconnects G.*

Definition 3 (k-Connectivity)**.** *Given a system communication graph $G = (V, E)$, the* vertex connectivity $k(G)$ *(or* connectivity *in short) is the size of the smallest vertex cut. A graph is called* k-vertex-connected *(or* k-connected in short*) if its vertex connectivity is k or greater.*

Every process $p_i \in V$ can communicate directly only with its neighbors in G (i.e., p_i can communicate with a process p_j if and only if there exists an edge $(i, j) \in E$).

Every process p_i has access to two distinct communication primitives, namely *authenticated point-to-point channel* (or *authenticated link* in short) and *authenticated local broadcast*. More precisely, when considering an authenticated point-to-point link, a process p_i may invoke a $\langle al, Send|...\rangle$ event to send a message to one of its neighbors and a $\langle al, Deliver|...\rangle$ is triggered when p_i delivers a message from one of its neighbors. Authenticated links are reliable and guarantee that if a correct process p_i delivers a message sent from a correct process p_j then p_j previously sent it and every message m is delivered at most once to its intended destination.

The authenticated local broadcast primitive allows a correct process p_i to send simultaneously the same message m to all its neighbors[1] and it is characterized by the following events $\langle alb, ALBroadcast|...\rangle$ and $\langle alb, ALDeliver|...\rangle$. It guarantees that if a correct process p_i locally broadcasts a message m, then eventually every neighbor delivers m.

Message adversaries. We also assume the presence of a *Message Adversary* (MA) that can interfere with the underlying communication system by intercepting and dropping messages. We consider three different types of message adversaries, each characterized by the parameter d measuring the power of the adversary in catching and dropping messages. To emphasize this dependency, we denote them using the notation MA(d). The three message adversaries under consideration are:

[1]The authenticated local broadcast primitive can be easily constructed by leveraging authenticated point-to-point links by invoking a $\langle al, Send|...\rangle$ event on every link connecting p_i with one of its neighbors.

- *Message remover (MA1(d)):* This MA is the one considered in [1]. In this case, when a process p_i locally broadcasts a message m in its neighborhood, the MA can pick up to d messages and drop/block them. Let us note that this MA can target multiple senders and for each of them may drop up to d messages. Thus, eventually, every process may fail in delivering some message.
- *Node silent (MA2(d)):* This MA targets up to d distinct nodes and it can remove all incoming messages for them (i.e., it makes up to d processes deaf during an instance of the protocol $\mathcal{P}$). The set of affected nodes is fixed for the entire instance of the protocol $\mathcal{P}$ but may change between two different instances of the same protocol.
- *Edge remover (MA3(d)):* This MA can remove up to d edges from the network. The set of affected edges is fixed for the entire instance of the protocol $\mathcal{P}$ but may change between two different instances of the same protocol.

Cryptographic Assumptions. We consider two distinct cases:

- **Authenticated Model:** Nodes have access to a Public Key Infrastructure (PKI) that enables them to sign, encrypt, and verify messages. In this model, we assume that Byzantine nodes are computationally bounded and cannot forge a node's signature.
- **Unauthenticated Model:** Nodes do not have access to a PKI, meaning messages cannot be cryptographically signed or verified. This allows Byzantine nodes to arbitrarily modify or forge the content of messages.

A. Problem Statement

We aim to implement a *Message-Adversary-Tolerant Byzantine Reliable Broadcast* (MBRB) primitive that enables n nodes to eventually agree on a message m sent by a predefined sender, despite the presence of both Byzantine (malicious) nodes and a *Message Adversary* (MA) controlling the network. To achieve this, we first consider a *Reliable Communication* (RC) primitive that allows any two correct processes to reliably exchange messages, even if they are not directly connected in the communication graph G and then we move to the MBRB primitive.

Every process p_i can interact with the *Reliable Communication (RC)* primitive by means of two core events: $\langle rc, RCBroadcast|...\rangle$, invoked by the sender to initiate the dissemination of a message m, and $\langle rc, RCDeliver|...\rangle$, triggered at the receiving processes to indicate that message m, originally broadcast via *RC-broadcast*, has been successfully delivered.

Definition 4 (Reliable Communication)**.** *Given a sender process p_s and a message m, a protocol $\mathcal{P}$ implements a Reliable Communication (RC) primitive if it satisfies the following three properties:*

- ***RC-No duplication:*** *A correct process p_i RC-deliver m at most once.*
- ***RC-No creation:*** *If a correct process p_i RC-deliver a message m, then m was RC-broadcast by p_s.*
- ***RC-Validity:*** *If a correct process p_s RC-broadcast a message m, then every correct processes p_i eventually RC-deliver m.*

Since Albouy et al. [1] proved that under the MA1, the **RC-Validity** property cannot be satisfied in a fully connected graph even assuming the presence of a PKI, we consider the following weaker specifications:

Definition 5 (Parametric Reliable Communication)**.** *Given a sender process p_s and a message m, a protocol $\mathcal{P}$ implements a* Parametric Reliable Communication *(PRC) primitive if it satisfies the following three properties:*

- ***RC-No duplication:*** *A correct process p_i RC-deliver m at most once.*
- ***RC-No creation:*** *If a correct process p_i RC-deliver a message m, then m was RC-broadcast by p_s.*
- ***Parametric RC-Validity:*** *If a correct process p_s RC-broadcast a message m, then at least $L \geq 1$ other correct processes eventually RC-deliver m.*

Definition 6 (Weak Reliable Communication)**.** *Weak Reliable Communication (WRC) is a special case of Parametric Reliable Communication (PRC) in which the validity parameter is set to its minimal value, i.e., L = 1.*

Definition 7 (Byzantine Reliable Broadcast)**.** *Given a sender process p_s and a message m, a protocol $\mathcal{P}$ implements a* Byzantine Reliable Broadcast *(BRB) primitive if it satisfies the following properties:*

- ***BRB-No duplication:*** *A correct process p_i BRB-deliver m at most once.*
- ***BRB-No creation:*** *If a correct process p_i BRB-deliver a message m, then m was BRB-broadcast by p_s.*
- ***BRB-Validity:*** *If a correct process p_s BRB-broadcast a message m, then every correct process p_i eventually BRB-deliver m.*
- ***BRB-Agreement:*** *If a correct process p_i BRB-deliver m, then every correct process eventually delivers m.*

Definition 8 (Message-Adversary Byzantine Reliable Broadcast)**.** *Given a sender process p_s and a message m, a protocol $\mathcal{P}$ implements a* Message-Adversary Byzantine Reliable Broadcast *(MBRB) primitive if it satisfies the following properties:*

- ***MBRB-No duplication:*** *A correct process p_i MBRB-deliver m at most once.*
- ***MBRB-No creation:*** *If a correct process p_i MBRB-deliver a message m, then m was MBRB-broadcast by p_s.*
- ***MBRB-No duplicity:*** *No two different correct nodes MBRB-deliver different application messages from node p_s.*
- ***MBRB-Local Delivery:*** *Suppose p_s is correct and MBRB-broadcast an application message m. Then at least one correct node p_j eventually MBRB-deliver m from node p_s.*

- ***MBRB-Global Delivery:*** *Suppose a correct node p_i MBRB-deliver an application message m from p_s. Then at least L correct nodes MBRB-deliver m from p_s.*

IV. Lower Bounds

Due to the lack of space, we report here only the main claims reporting the lower bound related to $MA1(d)$. Table I reports the overview of all the lower bounds considering other adversaries. Throughout this section, we consider a system composed of a set of processes represented by a communication graph $G = (V, E)$, where $n = |V|$ denotes the total number of processes and $k(G)$ denotes the vertex connectivity of G. Among the processes, up to t may be Byzantine.

Theorem 1 (Impossibility of RC under MA1 if $d > 0$)**.** *Let $MA1(d)$ be the message adversary that affects the system. It is impossible in an asynchronous system to implement a protocol $\mathcal{P}$ that implements* Reliable Communication *on G if $d > 0$.*

Proof: The claim trivially follows from the impossibility proved by Albouy et al. [1], [3] by observing that any protocol $\mathcal{P}$ implementing an RC primitive must satisfy the *RC-Validity*. Let D be a set of d nodes in V. Given the power of the Message Adversary to block at most d messages per node broadcast, it is always possible, every time a node p_i broadcasts a message, to block all messages directed to D. Therefore, regardless of the correct node actions, the nodes in D always remain disconnected from the rest of the network by the message adversary MA1. ∎

Theorem 2 (Impossibility of WRC under MA1 if $\delta(p_s) \leq t + d$)**.** *Let $MA1(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has access to a PKI, then it is impossible in an asynchronous system to implement a protocol $\mathcal{P}$ that implements* Weak Reliable Communication *on G if the broadcast source p_s has degree less than or equal to $t + d$ (i.e., $\delta(p_s) \leq t + d$).*

Proof: The claim trivially follows by observing that $MA1(d)$ and the t Byzantine processes may create a vertex cut around the broadcast source p_s as none of its correct neighbors ever receive any message. As a consequence, from the communication point of view, p_s is disconnected from the rest of the graph. This prevents any message propagation and the *Weak RC-Validity* property does not hold from which the claim follows. ∎

Corollary 1. *Let $MA1(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ has access to a PKI, it is necessary for the implementation of a* Weak Reliable Communication *(WRC) primitive that G has connectivity at least $t + d + 1$ (i.e., $k(G) \geq t + d + 1$).*

Proof: It trivially follows from Theorem 2. ∎

Corollary 2. *Let $MA1(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ has access to a PKI, it is necessary for the implementation of a* Parametric Reliable Communication *(PRC) primitive that G has connectivity at least $t + d + 1$ (i.e., $k(G) \geq t + d + 1$).*

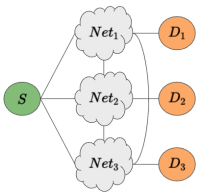


Fig. 1: Illustration of the network G with $n = 3(t+2d+x)+1$ nodes and connectivity $k(G) \geq t + d + x$. The orange sets D_i represent the groups of correct nodes that the message adversary $MA1(d)$ can isolate by blocking communications from any $p \in Net_i$ to every $q \in D_i$.

Proof: The corollary trivially follows from Theorem 2 and considering that *Parametric RC-Validity* is stronger than *Weak RC-Validity*. ∎

Theorem 3 (Impossibility of PRC When $L > n - t - ad$)**.** *Let $MA1(d)$ be the message adversary that affects the system. Assume that $k(G) \geq t + d + x$ and that $n > a(t + 2d + x)$ for some constant $a > 0$. Then, it is impossible to implement a protocol $\mathcal{P}$ that achieves Parametric Reliable Communication (PRC) on G in an asynchronous setting with a delivery guarantee parameter $L > n - t - ad$.*

Proof: Assume, for the sake of contradiction, that there exists a protocol $\mathcal{P}$ that ensures PRC with $L > n - t - ad$. Construct a graph G as illustrated in Figure 1, consisting of a distinct subgraphs $Net_1, \ldots, Net_a$ each fully connected and of size at least $t + d + x$ and a distinct sets $D_1, \ldots, D_a$ of at most d nodes. Since $MA1(d)$ can block all messages from a single process to up to d destinations, it can prevent any $p \in Net_i$ from reaching all nodes $q \in D_i$. Consequently, $MA1$ can isolate ad correct processes across the network, preventing them from delivering any messages. This contradicts the assumption that at least $L > n - t - ad$ correct processes deliver the message, as at most $L = n - t - ad$ correct nodes will deliver the message. Therefore, such a protocol $\mathcal{P}$ cannot exist. ∎

Theorem 4 (Impossibility of WRC under MA2 if $\delta(p_s) \leq t + d$)**.** *Let $MA2(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has access to a PKI, then it is impossible in an asynchronous system to implement a protocol $\mathcal{P}$ that implements* Weak Reliable Communication *on G if the broadcast source p_s has degree less than or equal to $t + d$ (i.e., $\delta(p_s) \leq t + d$).*

Proof: The claim directly follows by observing that if the source p_s has degree $\delta(p_s) = t + d$ then it is possible to find an execution where the t Byzantine processes are directly connected to p_s and the remaining d nodes in p_s's neighborhood are selected by $MA2(d)$, which drops all messages coming from p_s. This generates a network partition that prevents any correct process from delivering any message sent by p_s, which

-	RC		PRC		WRC	
type	PKI	No PKI	PKI	No PKI	PKI	No PKI
MA1	✗ [1], [3]		✻ $L > n - t - 2ad$ Th.3		✓ Th. 8 ▲ Cor. 1 ✻ if $\delta(p_s) \leq t + d$ Th. 2	✓ Th. 13 ▲ Cor. 1 ✻ if $\delta(p_s) \leq t + d$ Th. 2
MA2	✗ By def. of MA2		✓ Th. 9 ▲ Cor. 4	✓ Th. 14 ▲ Cor. 5	✓ Derived ▲ Cor. 3 ✻ if $\delta(p_s) \leq t + d$ Th. 4	
MA3	✓ Th. 11 ▲ Cor. 8	✓ Th. 16 ▲ Cor. 9 ✻ Th. 7	✓ Derived ▲ Cor.7		✓ Derived ▲ Cor.6 ✻ if $\delta(p_s) \leq t + d$ Th. 6	

-	BRB		MBRB	
type	PKI	no PKI	PKI	no PKI
MA1	✗ [1], [3]		✻ $L > n - t - 2ad$ Th.3	
MA2	✗ By def.		✓ Th.10	✓ Th.15
MA3	✓ Th.12	✓ Th.17	✓ Derived	

TABLE I: Summary of our results where ✓ = Feasible, ✗ = Impossibility results, ▲ The solution is optimal matching the lower bound, ✻ Condition specifying when impossibility applies (i.e., the boundary of the infeasible regime).

violates the *Weak RC-Validity* property. The claim follows. ■

Corollary 3. *Let $MA2(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ has access to a PKI, it is necessary for the implementation of a* Weak Reliable Communication *(WRC) primitive that G has connectivity at least $t + d + 1$ (i.e., $k(G) \geq t + d + 1$).*

Proof: It trivially follows by Theorem 4. ■

Corollary 4. *Let $MA2(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ has access to a PKI, it is necessary for the implementation of a* Parametric Reliable Communication *(PRC) primitive that G has connectivity at least $t + d + 1$ (i.e., $k(G) \geq t + d + 1$).*

Proof: The corollary trivially follows by Theorem 4 considering that *Parametric RC-Validity* is stronger than *Weak RC-Validity*. ■

Theorem 5 (Impossibility of PRC under MA2 with $L > t$ if $k(G) \leq 2t + d$)**.** *Let $MA2(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ doesn't have access to a PKI, then it is impossible in an asynchronous system to implement a protocol $\mathcal{P}$ that implements* Parametric Reliable Communication *on G with $L > t$ if the network G has connectivity $k(G) \leq 2t + d$.*

Proof: Assume, for contradiction, that there exists a protocol $\mathcal{P}$ that implements PRC on a multi-hop graph G with connectivity $k(G) \leq 2t + d$ in the unauthenticated setting (no PKI) with $L > t$. It is straightforward to observe that if the sender s has exactly $2t + d$ neighbors, then in the worst case at most t of them can be Byzantine and at most d can be disrupted by the message adversary, leaving at least t correct neighbors that will receive and forward s's message. Thus, at least $L = t$ correct nodes can immediately deliver the message once they obtain it. Now consider a generic correct node r that is not directly connected to s. By definition of $k(G)$, the number of pairwise node-disjoint paths connecting s and r is at most $2t+d$. Let $\mathcal{X}$ be a maximal family of such disjoint paths, with $|\mathcal{X}| \leq 2t + d$. Consider an execution where:

- t of the paths in $\mathcal{X}$ are corrupted by Byzantine processes (for example, by taking control of t nodes in $\mathcal{X}$);
- d further paths in $\mathcal{X}$ are disrupted by the message adversary $MA2(d)$ (for example, by disconnecting d nodes in $\mathcal{X}$).

Since $\mathcal{X}$ contains at most $2t + d$ paths in total, at most t disjoint paths remain both intact and free of Byzantine. In an unauthenticated system, however, a correct receiver must collect at least $t + 1$ identical copies of the message over disjoint paths to distinguish genuine messages from forgeries (Dolev's condition). With only t such paths available, r cannot safely deliver. This contradicts the assumption about $L > t$, and we conclude that PRC cannot be implemented in a graph G with $k(G) \leq 2t + d$ and $L > t$. ■

Corollary 5. *Let $MA2(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ doesn't have access to a PKI, it is a necessary condition for the implementation of a* Parametric Reliable Communication *(PRC) primitive with $L > t$ that G has connectivity at least $2t + d + 1$ (i.e., $k(G) \geq 2t + d + 1$).*

Proof: The corollary trivially follows by Theorem 5. ■

Theorem 6 (Impossibility of WRC under MA3 if $\delta(p_s) \leq t + d$)**.** *Let $MA3(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has access to a PKI, then it is impossible in an asynchronous system to implement a protocol $\mathcal{P}$ that implements* Weak Reliable Communication *(WRC) on G if the broadcast source p_s has degree less than or equal to $t + d$ (i.e., $\delta(p_s) \leq t + d$).*

Proof: The claim directly follows by observing that if the source p_s has degree $\delta(p_s) = t + d$ then it is easily possible to find an execution where the t Byzantine processes are directly connected to p_s and the remaining d nodes in p_s's neighborhood are linked by edges selected by $MA3(d)$, which drop all messages coming from p_s. This induces a network partition that prevents any correct process from delivering any message sent by p_s. Hence, the *Weak RC-Validity* property is violated, and the claim follows. ■

Corollary 6. *Let $MA3(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ has access to a PKI, it is a necessary condition for the implementation of a* Weak Reliable Communication *(WRC) primitive that G has connectivity at least $t + d + 1$ (i.e., $k(G) \geq t + d + 1$).*

Proof: It trivially follows by Theorem 6. ∎

Corollary 7. *Let $MA3(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ has access to a PKI, it is a necessary condition for the implementation of a* Parametric Reliable Communication *(PRC) primitive that G has connectivity at least $t+d+1$ (i.e., $k(G) \geq t+d+1$).*

Proof: The corollary trivially follows by Theorem 6 and considering that *Parametric RC-Validity* is stronger than *Weak RC-Validity*. ∎

Corollary 8. *Let $MA3(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ has access to a PKI, it is a necessary condition for the implementation of a* Reliable Communication *(RC) primitive that G has connectivity at least $t+d+1$ (i.e., $k(G) \geq t+d+1$).*

Proof: The corollary trivially follows by Theorem 6 and considering that *RC-Validity* is stronger than *Weak RC-Validity*. ∎

Theorem 7 (Impossibility of RC under MA3 if $k(G) \leq 2t+d$)**.** *Let $MA3(d)$ be the message adversary that affects the system. Suppose that processes have no access to a PKI, then it is impossible in an asynchronous system to implement a protocol $\mathcal{P}$ that implements* Reliable Communication *on G if $k(G) \leq 2t + d$.*

Proof: We prove the theorem by contradiction. Assume that there exists a protocol $\mathcal{P}$ that implements a *Reliable Communication* (RC) primitive on a graph G with connectivity $k(G) \leq 2t+d$. This implies that $\mathcal{P}$ satisfies all the properties of Definition 4, including the *RC-Validity* property. Consider now a generic graph G where the sender s is connected to exactly $2t+d$ nodes, of which t are Byzantine and the remaining $t+d$ are correct. Without loss of generality, denote the neighbors of s as $neighbors(s) = \{t_1, \ldots, t_t, c_1, \ldots, c_{t+d}\}$, where each t_x represents a Byzantine node and each c_y represents a correct node.

Consider now the behavior of the message adversary $MA3(d)$, which may remove up to d edges from the graph. Suppose it removes the d edges connecting s to the correct nodes $\{c_{t+1}, \ldots, c_{t+d}\}$, leaving only the edges between s and $\{c_1, \ldots, c_t\}$ intact. In this case, the nodes $c_1, \ldots, c_t$ can safely *RC-deliver* the message, having received it directly from the sender. However, all the remaining correct nodes are now only connected to the sender s via $2t$ node-disjoint paths: t through correct nodes and t through Byzantine nodes. Since Byzantine nodes can behave arbitrarily, they may forward a forged message m' instead of the original message m. From the perspective of a receiver with access only to these $2t$ paths, it becomes impossible to distinguish between a genuine message and a forged one.

Therefore, in the absence of cryptographic guarantees, no algorithm can ensure RC when $k(G) \leq 2t + d$ under the influence of Byzantine nodes and a type $MA3(d)$ message adversary. ∎

Corollary 9. *Let $MA3(d)$ be the message adversary that affects the system. Assuming that every process $p_i \in V$ doesn't have access to a PKI, it is a necessary condition for the implementation of a* Reliable Communication *(RC) primitive that G has connectivity at least $2t + d + 1$ (i.e., $k(G) \geq 2t + d + 1$).*

Proof: It trivially follows by Theorem 7. ∎

V. Feasibility of a RC Primitive in Presence of a PKI

In this section, we present a simple *flooding-based* protocol to prove that the RC primitive can be implemented over a k-connected graph $G = (V, E)$ with k matching the lower bounds presented in Section IV. The protocol described below is designed to handle a single invocation of the Reliable Communication (RC) primitive for a unique sender–message pair (s, m). That is, we assume that during the execution, there is one correct sender p_s broadcasting a message m, and that the protocol instance is scoped to this specific communication. If p_s needs to distribute more messages, it may activate a new instance of the protocol $\mathcal{P}$ for every message sent.

Local Variables at node. Each node p_i maintains a local variable $delivered$, initially set to *false*. This flag ensures that each message m is *RC-delivered* at most once by p_i, preventing duplicate deliveries.

Functions available at node p_i. Each process p_i has access to two functions that allow it to communicate with the Public Key Infrastructure (PKI). *sign_private_key(m)* returns a signature S of the message m, signed with the private key of the invoking node, while *verify_signature(S')* verifies whether the received signature S' corresponds to a message genuinely signed by the sender s.

Working Flow. The protocol proceeds in two phases. In phase 1 (sender behavior), the sender p_s signs the message m using its private key to produce a signature S. It then broadcasts S to its neighbors using the authenticated local broadcast *alb* primitive, sets *delivered* to *true*, and triggers the *RC-deliver* event (line 8). In phase 2 (receiver behavior), upon delivering a message from the authenticated local broadcast *alb*, a node p_i verifies whether it has already delivered it. If not, it checks the authenticity of the signature. If the signature is valid and the message has not yet been delivered, p_i re-broadcasts the signature S', sets *delivered* to *true*, and delivers the decoded message m (line 15).

Throughout this section, we consider a system composed of a set of processes represented by a communication graph $G = (V, E)$, where $n = |V|$ denotes the total number of processes and $k(G)$ denotes the vertex connectivity of G. Among the processes, up to t may be Byzantine.

Algorithm 1: Reliable Communication in an Authenticated Model

1 **Init:**
2 *alb*: instance of authenticated link broadcast;
3 *delivered* = *false*;

4 **Upon Event** $\langle rc, Broadcast|m\rangle$**:**
5 *S* = *sign_private_key(m)*
6 **trigger** $\langle alb, ALBroadcast|S\rangle$;
7 *delivered* = *true*;
8 **trigger** $\langle rc, Deliver|m\rangle$;

9 **Upon Event** $\langle alb, ALDeliver|S'\rangle$**:**
10 **if** *not delivered* **then**
11 **if** verify_signature(S') == true **then**
12 **trigger** $\langle alb, ALBroadcast|S'\rangle$;
13 *delivered* = *true*;
14 *m* = *recover_message(S')*;
15 **trigger** $\langle rc, Deliver|m\rangle$

Lemma 1 (RC-No duplication)**.** *Let m be a message broadcast by a correct source p_s executing the protocol shown in Algorithm 1. If a correct process p_i executes the protocol shown in Algorithm 1 then p_i RC-deliver m at most once.*

Proof: From Algorithm 1, the *RC-deliver* event is triggered only in lines 8 and 15. In both cases, before triggering the event, the protocol sets $delivered = true$, making it impossible for a correct node that has already delivered a message to pass the condition at line 10 twice. ∎

Lemma 2 (RC-No creation)**.** *Let m be a message broadcast by a correct source p_s executing the protocol shown in Algorithm 1. If a correct process p_i executes the protocol shown in Algorithm 1 and it RC-deliver a message m, then m was RC-broadcast by p_s.*

Proof: Revisiting the two occurrences of the *RC-deliver* function, we observe that the first one (line 8) is contained within the *RC-broadcast* function itself. This means that executing line 8 is only possible if line 3 has already been executed.
As for line 15, a node can reach this point only after receiving a message S'. Since the sender must have broadcasted this message—otherwise, the `Check_sign()` function (line 5) would return `false`, preventing further execution, including line 15—it follows that a node can execute line 15 only if the sender has previously *RC-broadcast* the message. ∎

Lemma 3 (Weak RC-Validity)**.** *Let $G = (V, E)$ be the system communication graph connecting the $n = |V|$ processes participating in the system and let $k(G) \geq t + d + 1$ be the connectivity of G. Let t be the number of Byzantine processes in the system, and let $MA1(d)$ be the message adversary that affects the system. If every correct process executes the protocol shown in Algorithm 1 and a correct process p_s RC-broadcast a message m, then at least one other correct process eventually RC-deliver m.*

Proof: Let us consider the sender node p_s. While executing line 3, the following conditions hold:

- p_s has at least $t+d+1$ neighbors (due to the connectivity of G)
- At most t of its neighbors are Byzantine nodes.
- At most d of its neighbors can be affected by the message adversary ($MA1(d)$)

It follows that there is always at least one correct process p_i that is not affected by the message adversary $MA1(d)$, and that delivers m directly from p_s. Due to the properties of the local broadcast, p_i eventually receives the signed message S and successfully *RC-deliver* it (15), from which the claim follows. ∎

Theorem 8 (Feasibility of WRC Under MA1 with PKI)**.** *Let $k(G) \geq t + d + 1$ be the connectivity of G and let $MA1(d)$ be the message adversary that affects the system. The protocol shown in Algorithm 1 implements a Weak Reliable Communication primitive.*

Proof: The claim follows from Lemma 1, Lemma 2 and Lemma 3. ∎

Lemma 4 (Parametric RC-Validity)**.** *Let $MA2(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has access to a PKI. If $k(G) \geq t+d+1$, and every correct process executes the protocol in Algorithm 1 then if a correct process p_s RC-broadcast a message m, at least $L = n - t - d$ other correct processes eventually RC-deliver m.*

Proof: When the sender p_s broadcasts the signed message S to all its neighbors, the following conditions hold:

- At most t of its neighbors are Byzantine nodes.
- At most d of its neighbors can be affected by the message adversary $MA2(d)$. In particular, the adversary can affect at most d of the correct neighbors by blocking all their incoming messages. Let's call this set of nodes D.

In this case, since the network G has connectivity $k(G) \geq t + d + 1$, there exist at least $t + d + 1$ node-disjoint paths between any pair of nodes (p_i, p_j). This implies that there exists a subset $L = correct\_nodes \setminus D$ of correct nodes that remain connected to the sender p_s through paths that avoid both Byzantine nodes and the set D of affected nodes. As a result, the message m can propagate along these unaffected paths, ensuring that the nodes in L eventually receive and deliver it. ∎

Theorem 9 (Feasibility of PRC Under MA2 with PKI)**.** *Let $MA2(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has access to a PKI. If $k(G) \geq t+d+1$ then the protocol in Algorithm 1 implements a* Parametric Reliable Communication *primitive with $L \geq n - t - d$.*

Proof: The claim follows from Lemma 1, Lemma 2 and Lemma 4. ∎

Theorem 10 (Feasibility of MBRB Under MA2 with PKI)**.** *Let $MA2(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has access to a PKI. If $k(G) \geq t + d + 1$ and $n > 3t + 2d$, then the protocol* MBRBbroadcast *introduced by Albouy et al. [1], combined with the protocol in Algorithm 1 as RC instance, solves MBRB.*

Proof: The claim follows from the observation that Theorem 9 provides the same delivery guarantees as the communication primitive $comm(m_1, m_2, \ldots, m_n)$ used in *MBRBbroadcast*. In particular, Algorithm 1 guarantees that when a correct nodes broadcast a message, at least $L \geq n - t - (1 + \epsilon)d$ correct processes deliver the message m. ∎

Lemma 5 (RC-Validity)**.** *Let $k(G) \geq t + d + 1$ be the connectivity of G and let $MA3(d)$ be the message adversary that affects the system. If every correct process executes the protocol shown in Algorithm 1 and a correct process p_s RC-broadcast a message m, then every correct process eventually RC-delivers it.*

Proof: Let us note that, in the presence of a PKI, Byzantine nodes are unable to forge signatures. As a result, even if they attempt to tamper with or inject fake messages, such messages are rejected by correct nodes during signature verification. Consequently, Byzantine processes can only hinder progress by refusing to relay valid messages, but they cannot corrupt the content or impersonate the sender. Now consider the message adversary $MA3(d)$, which is allowed to remove up to d edges from the network. Since the graph has connectivity $k(G) \geq t + d + 1$, even after the removal of any d edges by $MA3$, the remaining graph retains connectivity of at least $t + 1$. By Menger's Theorem, this guarantees the existence of at least $t + 1$ internally node-disjoint paths between any pair of correct nodes in the remaining graph. In particular, for every correct receiver p_i, there exist at least $t + 1$ node-disjoint paths from the sender s to p_i, despite the presence of both Byzantine nodes and $MA3$. Since at most t of these paths may traverse Byzantine nodes, at least one such path remains entirely controlled by correct processes. Along this path, the signed message broadcast by p_s eventually reaches p_i, allowing it to pass the signature check and be delivered. Therefore, every correct node eventually receives and *RC-deliver* the message, satisfying the RC-Validity property. ∎

Theorem 11 (Feasibility of RC Under MA3 with PKI)**.** *Let $k(G) \geq t + d + 1$ be the connectivity of G and let $MA3(d)$ be the message adversary that affects the system. The protocol shown in Algorithm 1 implements a Reliable Communication primitive.*

Proof: The claim follows from Lemma 1, Lemma 2 and Lemma 5. ∎

Algorithm 2 implements Bracha's Byzantine Reliable Broadcast protocol [6], adapted to use one of the reliable communication (RC) primitives introduced in this paper—either Algorithm 1 or Algorithm 3—depending on whether a public key infrastructure (PKI) is assumed.

Algorithm 2: Bracha's BRB algorithm

```
Init:
    rc : instance of reliable communication protocol;
    bracha_rc : instance of Bracha's algorithm with rc as
      a reliable communication primitive;

Upon Event ⟨Broadcast|m⟩:
    trigger ⟨bracha_rc, Broadcast|m⟩;

Upon Event ⟨bracha_rc, Deliver|m⟩:
    trigger ⟨Deliver|m⟩;
```

Theorem 12 (Feasibility of BRB Under MA3 with PKI)**.** *Let $k(G) \geq t + d + 1$ be the connectivity of G and let $MA3(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has access to a PKI. If $n > max(3t, t + d)$ and $k(G) \geq t + d + 1$, then Algorithm 2 (Bracha adapted with RC primitive), combined with the protocol in Algorithm 1 as RC instance, solves BRB.*

Proof: The claim follows from Theorem 11, which guarantees that whenever a correct process p rc-broadcasts a message m, all correct processes eventually deliver m. Therefore, by using Algorithm 1 as the underlying reliable channel abstraction within Bracha's broadcast protocol (Algorithm 2), we obtain a correct implementation of Byzantine Reliable Broadcast. ∎

VI. Feasibility of a RC Primitive without the Presence of a PKI

In this section, we will use Dolev's broadcast protocol [10], Algorithm 3, to prove that the RC primitive can be implemented over a k-connected graph $G = (V, E)$ with k matching the lower bounds presented in Section IV. The protocol described below is designed to handle a single invocation of the Reliable Communication (RC) primitive for a unique sender–message pair (s, m). That is, we assume that during the execution, there is one correct sender p_s broadcasting a message m, and that the protocol instance is scoped to this specific communication.

Local Variables at node p_i. Each node p_i maintains two local variables. *delivered*, initially set to *false*, is used to avoid delivering the same message multiple times, while *paths*, initialized to the empty set, is used to track the distinct node-disjoint paths through which a message m has been received.

Working Flow. The algorithm unfolds in two main phases. In phase 1 (sender behavior), the sender s invokes *ALBroadcast* with the initial message and an empty path, sets *delivered* to *true*, and triggers *RC-deliver* (line 8).

In phase 2 (receiver behavior), when a node p_i receives a message of the form [m, *path*] from a neighbor p_j, it appends p_j to the path and records the resulting path in *paths*. It

then rebroadcasts the message with the updated path. If p_i eventually observes t + 1 node-disjoint paths linking it to the source and has not yet delivered the message, it sets *delivered* to *true* and triggers *RC-deliver* (line 12).

Algorithm 3: Dolev's RC algorithm

```
1  Init:
2      alb : instance of authenticated link broadcast;
3      delivered = False;
4      paths = ∅;

5  Upon Event ⟨dolev, Broadcast|m⟩:
6      trigger ⟨alb, ALBroadcast|[m, []]⟩;
7      delivered = true;
8      trigger ⟨dolev, Deliver|m⟩

9  Upon Event ⟨alb, ALDeliver|p_j, [m, path]⟩:
10     if p_j = sender and path = [] and
         delivered = false then
11         delivered = true;
12         trigger ⟨dolev, Deliver|m⟩
13     paths.insert(path+[p_j]);
14     trigger ⟨alb, ALBroadcast|[m, path + [p_j]]⟩;

15 Upon Event ( self is connected to the source through
     t+1 node-disjoint paths contained in paths) and
     delivered=false:
16     delivered = true;
17     trigger ⟨dolev, Deliver|m⟩
```

Throughout this section, we consider a system composed of a set of processes represented by a communication graph $G = (V, E)$, where $n = |V|$ denotes the total number of processes and $k(G)$ denotes the vertex connectivity of G. Among the processes, up to t may be Byzantine.

Lemma 6 (RC-No duplication)**.** *Let m be a message broadcast by a correct source p_s executing the protocol shown in Algorithm 3. If a correct process p_i executes the protocol shown in Algorithm 1 then p_i RC-deliver m at most once.*

Proof: Given Algorithm 3, the function *RC-deliver* is executed only at line 8 at line 12 or at lien 17. Line 8 is executed only by the original sender p_s, while line 12 and 17 instead can be executed by every node. In all cases, before executing the *RC-deliver* function, the algorithm sets the variable $delivered = True$ making in this way impossible for a node to *RC-deliver* a message more than once. ∎

Lemma 7 (RC-No creation)**.** *Let m be a message broadcast by a correct source p_s executing the protocol shown in Algorithm 3. If a correct process p_i executes the protocol shown in Algorithm 3 and it RC-deliver a message m, then m was RC-broadcast by p_s.*

Proof: Revisiting the line where the function *RC-deliver* appears, we observe that the instance at line 8 is nested within the function *RC-broadcast* itself. As a result, executing this line inherently entails executing line 4, where the broadcast function is invoked.
For line 12, the condition can only be satisfied if a node receives the message directly from the sender. Since the Byzantine nodes can forge messages but cannot impersonate other nodes due to the authenticated links, if a correct node delivers a message m at line 12, it must be the case that the sender indeed broadcast m earlier.
For line 17, to satisfy the condition at line 15, a node must receive the message through at least $t+1$ node-disjoint paths. Since Byzantine nodes can control at most t of the $2t+1$ paths, this guarantees that the message m has been received from at least $t + 1$ correct nodes. Given that correct nodes follow the protocol, it follows that the sender must have previously broadcast the message m. ∎

Lemma 8 (Weak RC-Validity)**.** *Let $MA1(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has no access to a PKI. If $\delta(p_s) \geq t + d + 1$, and every correct process executes Algorithm 3. Then if a correct process p_s RC-broadcast a message m, at least one other correct processes eventually RC-deliver m.*

Proof: The proof follows directly from the observation that if $\delta(p_s) \geq t+d+1$, then at most t of the neighbors of p_s can be Byzantine. When p_s broadcasts the message m, the adversary can block at most d of these transmissions. Thus, at least one correct neighbor must receive and deliver m at line 12. ∎

Theorem 13 (Feasibility of WRC Under MA1 without PKI)**.** *Let $MA1(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has no access to a PKI. If $\delta(p_s) \geq t+d+1$ then the protocol in Algorithm 3 implements a* Weak Reliable Communication *primitive.*

Proof: The claim follows from Lemma 6, Lemma 7 and Lemma 8. ∎

Lemma 9 (Parametric RC-Validity)**.** *Let $MA2(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has no access to a PKI. If $k(G) \geq 2t+d+1$, and every correct process executes Algorithm 3. Then if a correct process p_s RC-broadcast a message m, at least $L = n - t - d$ other correct processes eventually RC-deliver m.*

Proof: Let G be the communication graph with vertex connectivity $k(G) \geq 2t + d + 1$. By Menger's Theorem, this implies that for any pair of nodes (u, v), there exist at least $2t + d + 1$ internally node-disjoint paths connecting them. In particular, there are at least $2t + d + 1$ node-disjoint paths connecting the sender to every correct node in the network. Recall that the message adversary $MA2$ can isolate at most d correct nodes by blocking all incoming messages directed toward them. Let $D \subseteq$ *correct_nodes* denote the set of such isolated nodes, and define $L =$ *correct_nodes* $\setminus D$ as the set of correct nodes not affected by the adversary. Because at most t nodes are Byzantine and at most d correct nodes are

isolated, any node $p_i \in L$ is connected to the sender via at least $(2t + d + 1) - t - d = t + 1$ node-disjoint paths that exclude both Byzantine nodes and isolated nodes in D. This satisfies the conditions established by Dolev for Reliable Communication in the presence of up to t Byzantine processes. Consequently, the message broadcast by a correct sender can propagate through a sufficient number of disjoint and reliable paths to reach all nodes in L, ensuring that each of them eventually *RC-delivers* the message. ■

Theorem 14 (Feasibility of PRC Under MA2 without PKI)**.** *Let $MA2(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has no access to a PKI. If $k(G) \geq 2t+d+1$ then the protocol in Algorithm 3 implements a* Parametric Reliable Communication *primitive with $L \geq n - t - d$.*

Proof: The claim follows from Lemma 6, Lemma 7 and Lemma 9. ■

Theorem 15 (Feasibility of MBRB Under MA2 without PKI)**.** *Let $MA2(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ has no access to a PKI. If $k(G) \geq 2t+d+1$ and $n > 3t+2d+2\sqrt{td}$, then the protocol* SigFreeK2LCast *introduced by Albouy et al. [2], combined with the protocol in Algorithm 3 as RC instance, solves MBRB with delivery power $L \geq \lceil n - t - \frac{(n-t)d}{n-3t-d} \rceil$.*

Proof: The claim follows from the observation that Algorithm 3 provides the same delivery guarantees that *SigFreeK2LCast* assumes from its $ur\_broadcast(m)$ primitive. Specifically, Theorem 14 ensures that when a correct process invokes a broadcast under MA2(d), at least $L \geq \lceil n-t-\frac{(n-t)d}{n-3t-d} \rceil$ correct processes receive m. Therefore, Algorithm 3 can be used as a drop-in replacement for $ur\_broadcast(m)$ and the proofs of *SigFreeK2LCast* carry over unchanged. ■

Lemma 10 (RC-Validity)**.** *Let $k(G) \geq 2t + d + 1$ be the connectivity of G and let $MA3(d)$ be the message adversary that affects the system. If every correct process executes the protocol shown in Algorithm 3 and a correct process p_s RC-broadcast a message m, then every correct process eventually RC-delivers it.*

Proof: Let G be the communication graph with connectivity $k(G) \geq 2t + d + 1$, and consider a message adversary $MA3(d)$ that removes up to d edges from the graph. After such removals, the resulting graph still has connectivity at least $2t + 1$. By Menger's Theorem, this guarantees the existence of at least $2t + 1$ internally node-disjoint paths between any pair of correct nodes in the remaining graph. In particular, for every correct receiver p_i, there exist at least $2t + 1$ node-disjoint paths from the sender p_s to p_i, despite the presence of both Byzantine nodes and $MA3$. Since at most t of these paths may traverse Byzantine nodes, the remaining at least $t+1$ are entirely composed of correct nodes. This satisfies the conditions established by Dolev for Reliable Communication in the presence of up to t Byzantine processes. Thus, each correct node p_i will eventually receive the broadcast message m from s via at least $t + 1$ independent and trustworthy paths, allowing it to safely *RC-deliver* m. Therefore, the algorithm satisfies the RC-Validity property under $MA3(d)$ in a $(2t + d + 1)$-connected graph. ■

Theorem 16 (Feasibility of RC Under MA3 without PKI)**.** *Let $k(G) \geq 2t + d + 1$ be the connectivity of G and let $MA3(d)$ be the message adversary that affects the system. The protocol shown in Algorithm 3 implements a Reliable Communication primitive.*

Proof: The claim follows from Lemma 6, Lemma 7 and Lemma 10. ■

Theorem 17 (Feasibility of BRB Under MA3 without PKI)**.** *Let $MA3(d)$ be the message adversary that affects the system. Suppose that every process $p_i \in V$ doesn't have access to a PKI. If $n > max(3t, 2t + d)$ and $k(G) \geq 2t + d + 1$ then Algorithm 2, combined with the protocol in Algorithm 3 as RC instance, solves BRB.*

Proof: The claim follows from Theorem 16, which guarantees that whenever a correct process p rc-broadcasts a message m, all correct processes eventually deliver m. Therefore, by using Algorithm 3 as the underlying reliable channel abstraction within Bracha's broadcast protocol (Algorithm 2), we obtain a correct implementation of Byzantine Reliable Broadcast. ■

VII. Experimental Evaluation

Simulation Environment. All simulations were conducted using QUANTAS [12], a timestep-based simulator designed for the quantitative performance analysis of distributed algorithms. By operating in discrete time steps, QUANTAS ensures that results are independent of specific network conditions (e.g., connection delays), operating system architectures, and hardware characteristics such as processor speed, memory size, or number of cores. This abstraction enables fair and consistent comparisons across different algorithmic solutions.

Message Adversary implementation. The influence of the Message Adversary (MA) on message transmission is modeled through the *broadcast* procedure. When a node p broadcasts a message m (originating either from the source s or from another node), the MA acts according to one of the strategies discussed in Section III.

Byzantine Node Modeling. Given the timestep-based nature of the simulation and the assumption of a Public Key Infrastructure (PKI), we model Byzantine nodes as silent nodes, that is, they do not send any messages. This modeling choice is justified by two main reasons. First, under a PKI, any forged message claiming to originate from the sender can be trivially detected by correct nodes through signature verification. Second, since the simulator operates in discrete time steps, malicious attempts to flood the network with spurious messages have no practical effect on the simulation's behavior or message scheduling. As a result, simulating Byzantine nodes as silent captures the essence of their adversarial behavior without introducing unnecessary complexity.

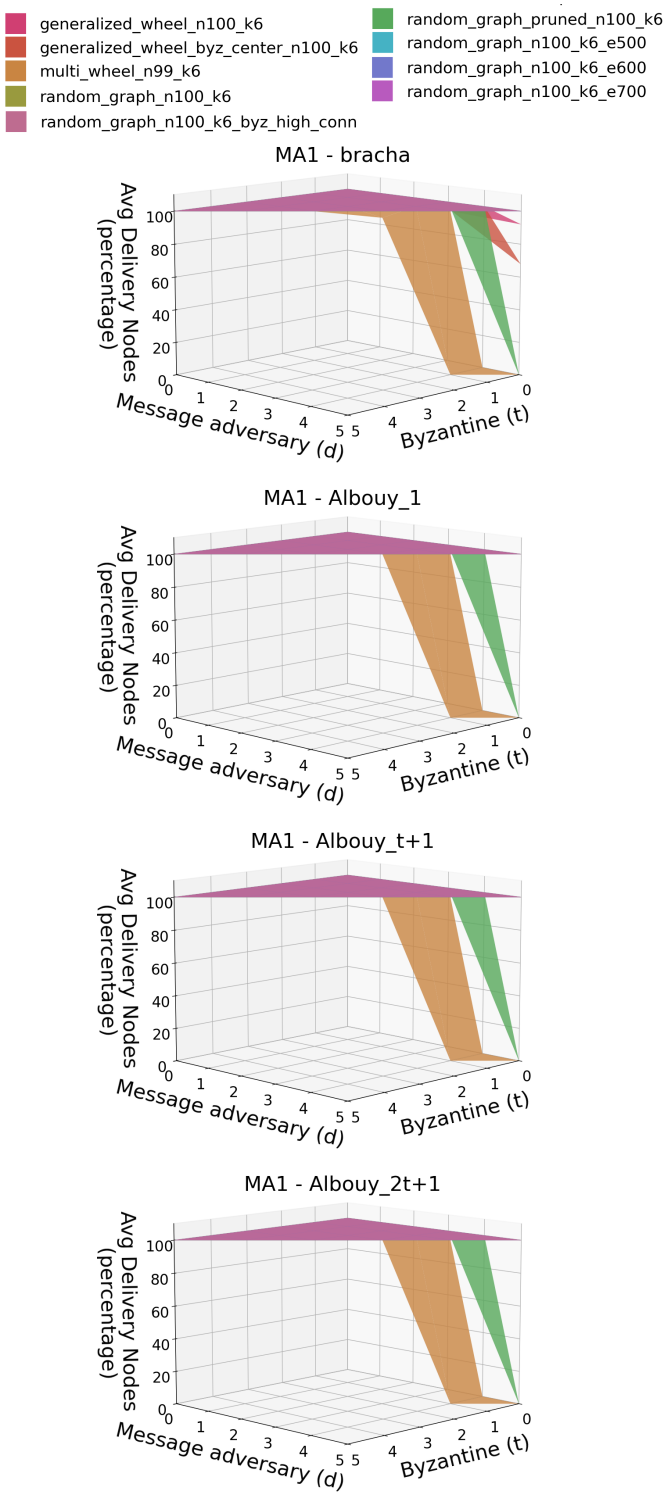


Fig. 2: 3D Plots comparing the avg_CND of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA1 .

Algorithms under test. We evaluate two Byzantine reliable broadcast protocols: our instantiation of the Bracha protocol (Algorithm 2), which uses our reliable communication primitive (Algorithm 1) as its RC component, and the near-optimal algorithm of Albouy et al. [1]. The latter leverages erasure coding: the sender encodes a message m into fragments such that any x fragments suffice to reconstruct m. Because the reconstruction threshold x (minimum number of fragments needed to reconstruct the original message) shapes both message and bit complexity, we consider three instantiations: *Albouy_1* with $x = 1$, *Albouy_t+1* with $x = t + 1$ and *Albouy_2t+1* with $x = 2t + 1$.

A. Research Questions and Metrics

In this section, we investigate the performance of BRB/MBRB protocols when running in the presence of both Byzantine processes and an MA.

To this end, we analyzed various network topologies and evaluated the algorithms using several performance metrics. We considered three distinct network structures: *(i)* multipartite wheel, *(ii)* generalized wheel, and *(iii)* Erdős–Rényi (ER) random graphs. We also explored key structural variations, including:

- *Targeted Byzantine selection*: prioritizing the selection of Byzantine nodes with the highest degree instead of random choice.
- *Edge pruning in random graphs*: removing edges randomly while preserving k-connectivity.
- *Network density variation*: starting from the initial Erdős–Rényi graph with 1000 edges, we randomly removed edges (always preserving k-connectivity) to obtain graphs with 700, 600, and 500 edges.

Random graphs were generated using the Erdős–Rényi model via the existing NetworkX functions [7][8]. Each topology is composed of $n = 100$ nodes and has a network connectivity $k = 6$. For each topology and for each adversarial configuration of (t, d) such that $t, d \in \{0, \ldots, k-1\}, t + d < k$, we conducted 25 independent test runs, the minimum required to achieve a 99% confidence level on almost all our results. Final results were obtained by aggregating statistics across the total tests:

- *Average delivery time* `avg_TTD`: For each message m, we computed the average time taken for all correct nodes that successfully delivered m. These values were averaged over all messages in a test, and then across all tests.
- *Average percentage of correct nodes delivering a message* `avg_CND`: For each message, we measured the fraction of correct nodes that delivered it. These percentages were averaged over all messages in a test, and then across all tests.
- *Total number of messages sent* `tot_MSGS`: The number of messages sent per test, including retransmissions and those blocked by the Message Adversary. We report the average across all tests.

Note on plotted adversaries. For the sake of readability and to avoid interrupting the flow of the discussion, we present

only the plots corresponding to MA1 in the main body of the paper. The complete set of plots for MA2 and MA3 has been moved to Appendix A.

RQ1 (Delivery rate). What is the average percentage of correct nodes that deliver (`avg_CND`) across topologies when varying (t, d) and the Message Adversary? Figure 2 shows that `avg_CND` stays at 100% across all (t, d) for every topology and algorithm, except for *multipartite wheel* and *random graph pruned* when d approaches $k - 1$. This drop is structural: in those graphs many nodes have degree exactly k; with MA1, when an intermediate node forwards to its $k - 1$ neighbors (excluding the predecessor), the adversary can drop all those $k - 1$ transmissions, stalling propagation.

RQ2 (Time to deliver). What is the average time to deliver (`avg_TTD`) across topologies? As depicted in Figure 3, all algorithms terminate in approximately 4-5 steps on *generalized wheel* and *ER*, while *multipartite wheel* requires substantially more steps due to its larger diameter (ring-like topology). The sharp "drop" in `avg_TTD` at high d simply reflects the 0% delivery cases in RQ1 (no delivery was represented with time = 0). Under MA2/MA3, the `avg_TTD` surfaces are essentially identical to MA1, meaning that the adversary type does not materially affect latency once delivery is possible.

RQ3 (Messages vs. bits). How do erasure-coded protocols compare to Bracha in terms of the number of messages and bits sent? Looking at Figure 4, the trend lines for the three algorithms are very similar. Bracha's algorithm remains essentially flat across all (t,d) combinations, while the other two vary slightly but hover around the same value. The key difference lies in the absolute number of messages sent: Bracha sends on the order of 0.1–0.4 million messages, whereas Albouy's algorithms send on the order of tens of millions (10–50 million)—roughly 100× more. This is expected: Bracha transmits a single message m, whereas Albouy splits m into $n = 100$ fragments and therefore sends about 100 messages per original message.

In terms of bits transmitted, the picture is more nuanced. Ignoring, for simplicity, the bits used for digital signatures and other PKI overhead, Bracha incurs $O(|m|)$ bits per message. Albouy's cost depends not only on m but also on the Error Correction Code (ECC). As explained in Albouy's paper [1], when an ECC is used to split a message into fragments it is important to select x, the minimum number of fragments required to reconstruct the original message. Since $|ECC(m)| \approx \frac{n}{x}|m|$, we should choose $x = \Theta(n)$ (i.e., x proportional to n) so that each fragment carries $O(|m|/n)$ bits and the total number of transmitted bits is on the same order as Bracha's, namely $O(|m|)$.

In our experiments, we instantiated *Albouy_1* with $x = 1$ and *Albouy_2t+1* with $x = 2t + 1$; both therefore exceed Bracha in total bits, but as we will discuss in RQ8, it is possible to increase the value of x without decreasing the performance of the system.

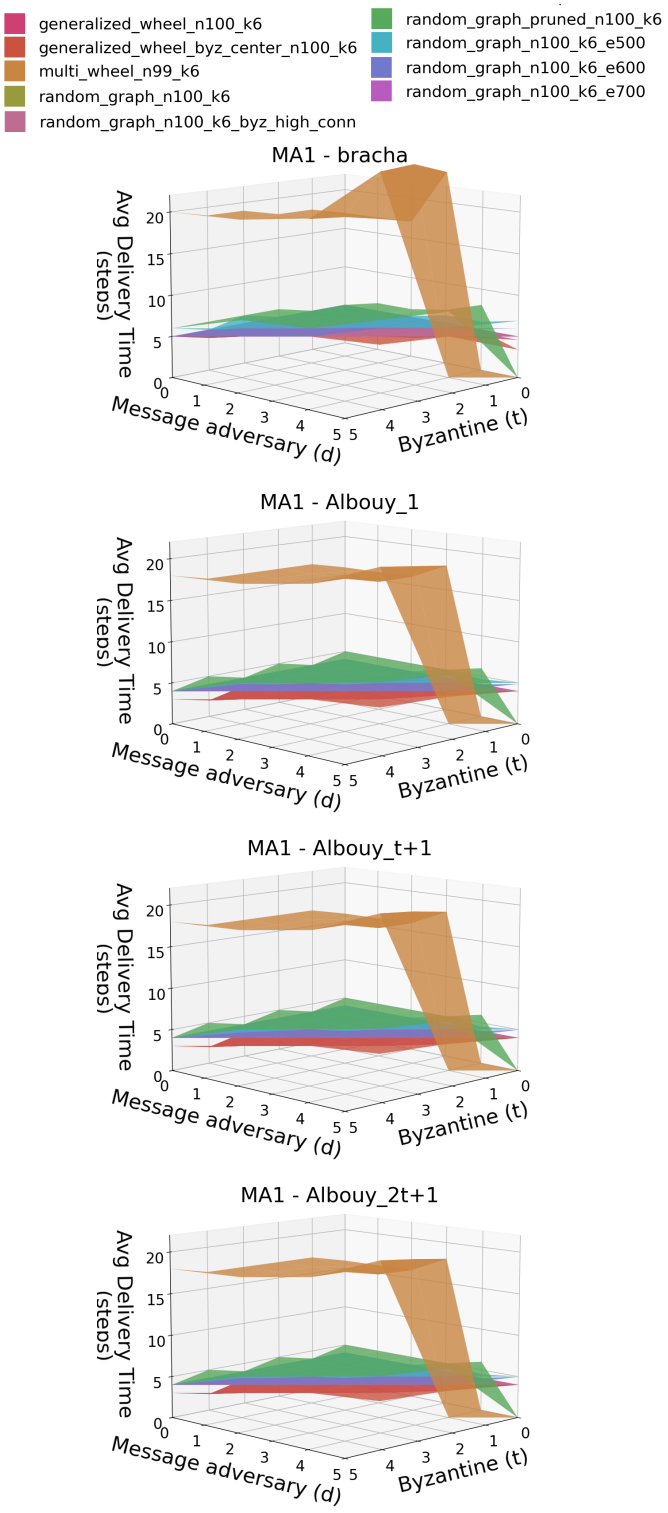


Fig. 3: 3D Plots comparing the avg_TTD of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA1.

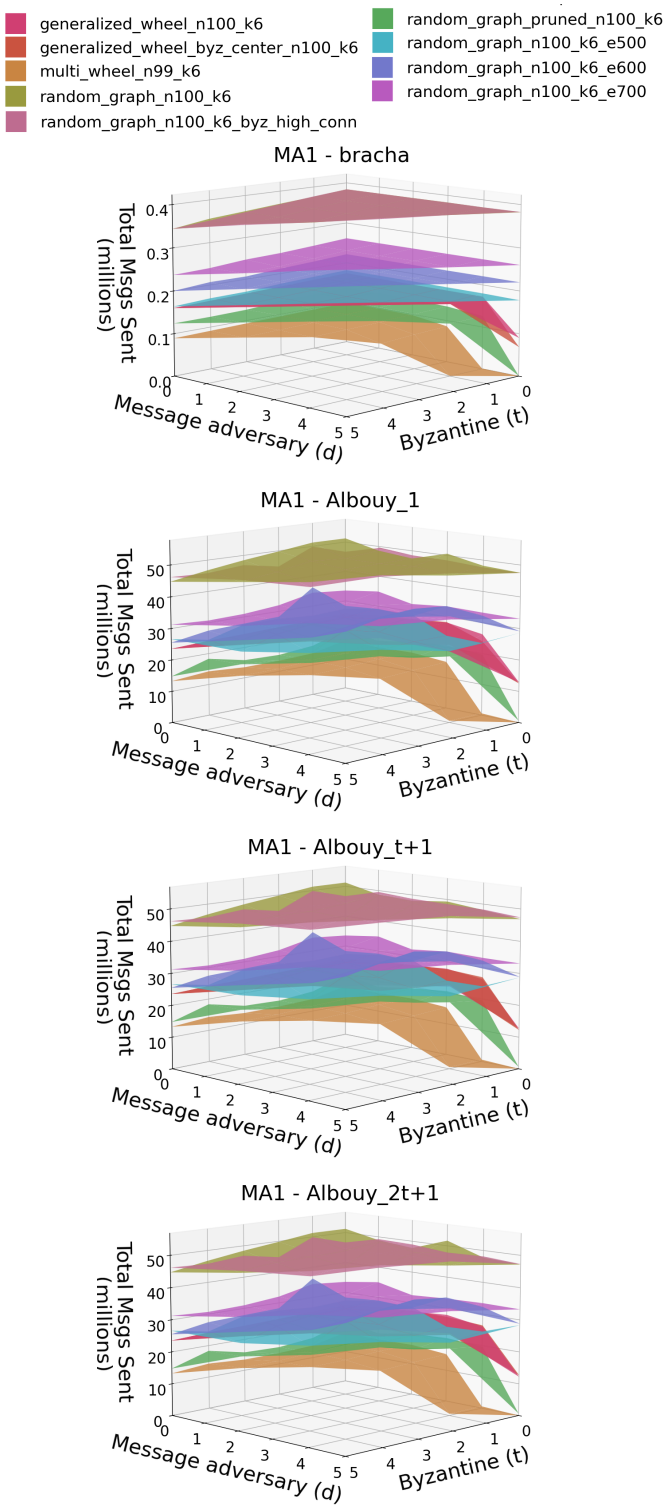


Fig. 4: 3D Plots comparing the tot_MSGS of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA1.

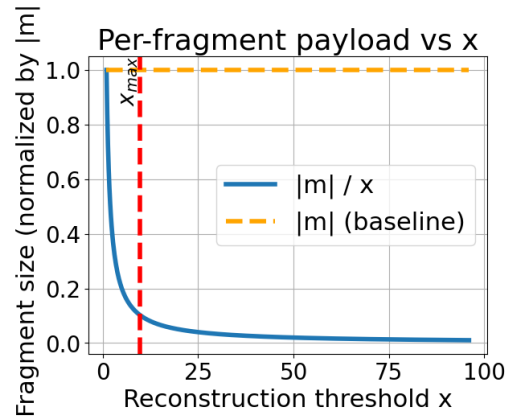
RQ4 (Best-performing families). Do Erdős–Rényi random graphs consistently outperform generalized/multipartite wheels on `avg_TTD` and communication cost? ER graphs do not uniformly dominate wheels. *Multipartite wheel* is the worst (both `avg_TTD` and occasional delivery failures under MA1 at $d \approx k-1$). Between *generalized wheel* and *ER*, the generalized wheel is slightly faster on `avg_TTD` (central hubs accelerate dissemination), while ER is comparable, about one extra step on average.

RQ5 (Edge pruning at fixed connectivity). In ER random graphs, does pruning while preserving $k(G)$ improve performance without harming delivery? As we saw in Figure 2, when $d << k$ the all the correct nodes deliver the message (like for the non-pruned random graph, even if they always have a delivery of 100% even for $d \sim k$). The main difference can be seen when looking at Figure 4 where the lower network density significantly lowers the number of messages exchanged in the network. On ER random graphs, pruning edges *while preserving* $k(G)$ leaves `avg_CND` unchanged (until the MA1 boundary $d \approx k-1$), and keeps `avg_TTD` essentially flat. However, `tot_MSGS` drops substantially: pruning edges reduces each node's broadcast fan-out, eliminating many redundant transmissions to the same neighbors.

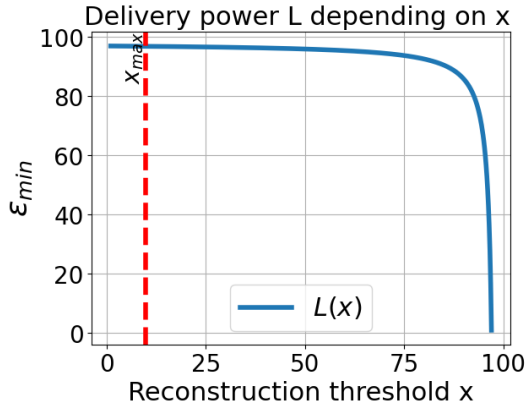
RQ6 (Density sweet spot). How does edge density affect performance, and where is the threshold beyond which extra edges cease to help (or start to hurt) efficiency? Continuing from the previous point, edge density strongly drives communication cost (`tot_MSGS`)—and thus bits—because each broadcast fans out to more neighbors, increasing per-round transmissions. The upside is resilience: a higher degree adds path redundancy and typically reduces diameter, making it harder for MA1 to stall propagation near the boundary $d \approx k-1$. By contrast, under MA2/MA3, we observe little sensitivity to density even when $d \sim k$. Consistent with Figure 2, when $d = 5 = k-1$ the pruned ER graph (with many nodes at degree $\approx k$) yields `avg_CND`$= 0\%$; in topologies where most nodes have degree well above k, MA1 cannot suppress all outgoing links from a forwarder, and delivery remains robust.

When considering MA1, if the objective is to maximize `avg_CND`, retain higher edge density; if the objective is to minimize `tot_MSGS`, prune edges while preserving $k(G)$, accepting a higher non-delivery risk as d approaches $k-1$.

RQ7 (Adversarial sensitivity to t vs. d). Under fixed topology and connectivity, how do incremental increases in t versus in d differentially impact `avg_CND`, `avg_TTD` and the average number of messages sent (`tot_MSGS`) (i.e., what is the relative sensitivity to t and to d)? With PKI, Byzantine nodes contribute little beyond withholding messages; increases in t therefore have a mild effect until t approaches the connectivity ceiling. By contrast, increases in d (MA1 drops) directly reduce effective fan-out each round and are far more damaging: `avg_CND` degrades sooner, and



(a)



(b)

Fig. 5: 2D plots showing how the fragment size and the delivery power L vary over x. The red dotted line represents (for a network with $(n = 100, t = 2, d = 1, \epsilon = 0.1)$) the maximum value of x that satisfies the conditions imposed by Albouy.

`avg_TTD` rises (or collapses to zero in the non-delivery regime). When considering instead MA2 and MA3 the impact of d is still bigger than t but considerably less harmful than MA1.

RQ8 (Impact of ECC). As expected, `tot_MSGS` is largely *insensitive* to the ECC threshold x (same dissemination pattern, different payload per message). The *bit* cost scales as $|ECC(m)| \approx (n/x)|m|$, so each fragment carries $\approx |m|/x$ payload bits. For our instantiations $x = 1$, $x = t + 1$ and $x = 2t + 1$, the payload blowup factors are n, $n/(t + 1)$ and $n/(2t + 1)$, respectively—both exceeding Bracha's $O(|m|)$. Choosing $x = \Omega(n)$ (constant rate) restores $O(|m|)$ total payload while keeping fragment size $O(|m|/n)$; in our simulations, moving from $x = 1$ to $x = t+1$ and then to $x = 2t+1$ did not materially harm `avg_CND` or `avg_TTD`, suggesting room to increase x further.

Total bits sent and delivery power L. **How does the Error Correction Code (ECC) affect the performance of the Albouy et al. Algorithm?** As you can see from Fig. 5(a), the fragment size decreases hyperbolically with x. In Albouy_1 ($x = 1$), each fragment f has $|f| = |m|$, so the total payload bits are $\approx$ 100 times Bracha's. In Albouy_2t+1, each fragment has $|f| = |m|/(2t + 1)$, cutting the Albouy_1 payload by a factor $\approx (2t + 1)$ yet still above Bracha and short of the $O(|m|)$ regime achievable with $x = \Omega(n)$. Following Albouy et al. [1], to guarantee delivery power $L \geq n - t - (1 + \varepsilon)d$, a sufficient condition is $x \leq \min\{n - t - 2d, \frac{\varepsilon}{1+\varepsilon}(n - t - d) + 1\}$. As you can see from Fig. 5(b), for $(n, t, d) = (100, 2, 1)$ the admissible range of x expands with larger ε: increasing ε (or decreasing d) permits larger x (and thus smaller fragment size), whereas pushing x beyond the depicted threshold forfeits the stated delivery-power guarantee. For a fixed $\varepsilon = 0.1$, we define $x_{\max}$ as the largest threshold value consistent with Albouy's conditions. In the plots, the red dotted line highlights this bound. The trade-off is clear: increasing ε shifts $x_{\max}$ to the right, enabling smaller fragments but reducing the guaranteed delivery power L; conversely, decreasing ε shifts $x_{\max}$ to the left, ensuring higher delivery power but at the expense of larger fragments and a higher total communication cost.

RQ9 (Theory vs. practice). How do results change when omniscience assumptions are removed for the adversaries? Do practical settings materially lessen the observed impact of t and d? When we move from the *theoretical* MA1 (omniscient, worst-case dropping) to a *practical* setting in which the adversary is *non-omniscient* and drops are selected without global knowledge, delivery becomes dramatically easier. We observed `avg_CND` $\approx 100\%$ across the tested topologies and (t, d), even in regimes where the worst-case theory predicts that delivery cannot be guaranteed. This gap is fully explained by our modeling assumptions: (i) PKI renders Byzantine nodes effectively silent (no profitable forgeries), and (ii) message removals are not adaptively targeted to choke critical cut-sets in each round. In a setting where losses resemble random faults rather than an adaptive scheduler, the probability that every forwarder's $k - 1$ outgoing edges are simultaneously suppressed becomes negligible, so the redundancy of the graph (and of the protocol's fan-out) suffices to sustain propagation.

VIII. Conclusion

This paper provided a comprehensive view of the possibility of implementing RC and MBRB primitives on top of multi-hop networks and in a multi-adversary model that considers both Byzantine failures and message adversaries. To the best of our knowledge, this is the first paper looking at the problem in such settings and providing a complete overview.

Our results (Table I) show that multi-hop structures introduce significant complexity. For $MA1$, propagation remains unresolved, but we establish necessary conditions and impossibility regimes. For $MA2$ and $MA3$, we give complete solutions and tight bounds. In addition, we also provided an experimental evaluation aimed at studying the performance of correct implementations.

References

[1] Timothé Albouy, Davide Frey, Ran Gelles, Carmit Hazay, Michel Raynal, Elad Michael Schiller, François Taïani, and Vassilis Zikas. Near-optimal communication byzantine reliable broadcast under a message adversary. In *28th International Conference on Principles of Distributed Systems (OPODIS 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

[2] Timothé Albouy, Davide Frey, Michel Raynal, and François Taïani. A modular approach to construct signature-free brb algorithms under a message adversary. *arXiv preprint arXiv:2204.13388*, 2022.

[3] Timothé Albouy, Davide Frey, Michel Raynal, and François Taïani. Asynchronous byzantine reliable broadcast with a message adversary. *Theoretical Computer Science*, 978:114110, 2023.

[4] Silvia Bonomi, Jérémie Decouchant, Giovanni Farina, Vincent Rahli, and Sébastien Tixeuil. Practical byzantine reliable broadcast on partially connected networks. In *41st IEEE International Conference on Distributed Computing Systems, ICDCS 2021, Washington DC, USA, July 7-10, 2021*, pages 506–516. IEEE, 2021. doi:10.1109/ICDCS51616.2021.00055.

[5] Gabriel Bracha and Sam Toueg. Resilient consensus protocols. In *Proceedings of the second annual ACM symposium on Principles of distributed computing*, pages 12–26, 1983.

[6] Gabriel Bracha and Sam Toueg. Asynchronous consensus and broadcast protocols. *J. ACM*, 32(4):824–840, October 1985. doi:10.1145/4221.214134.

[7] NetworkX Developers. Erdős–rényi graph example. https://networkx.org/documentation/stable/auto_examples/graph/plot_erdos_renyi.html.

[8] NetworkX Developers. is k edge connected. https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.connectivity.edge_augmentation.is_k_edge_connected.html.

[9] Danny Dolev. Unanimity in an unknown and unreliable environment. In *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, pages 159–168. IEEE, 1981.

[10] Danny Dolev. The byzantine generals strike again. *Journal of algorithms*, 3(1):14–30, 1982.

[11] Leslie Lamport, Robert Shostak, and Marshall Pease. *The Byzantine generals problem*, page 203–226. Association for Computing Machinery, New York, NY, USA, 2019. URL: https://doi.org/10.1145/3335772.3335936.

[12] Joseph Oglio, Kendric Hood, Mikhail Nesterenko, and Sébastien Tixeuil. Quantas: quantitative user-friendly adaptable networked things abstract simulator. In *Proceedings of the 2022 Workshop on Advanced tools, programming languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems*, pages 40–46, 2022.

[13] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980.

[14] Nicola Santoro and Peter Widmayer. Time is not a healer: Preliminary version. In *STACS 89: 6th Annual Symposium on Theoretical Aspects of Computer Science Paderborn, FRG, February 16–18, 1989 Proceedings 6*, pages 304–313. Springer, 1989.

[15] Nicola Santoro and Peter Widmayer. Distributed function evaluation in the presence of transmission faults. In *Algorithms: International Symposium SIGAL'90 Tokyo, Japan, August 16–18, 1990 Proceedings*, pages 358–367. Springer, 1990.

Appendix A. Plots for MA2 and MA3

A. MA2 Plots

We now report the results obtained under MA2.

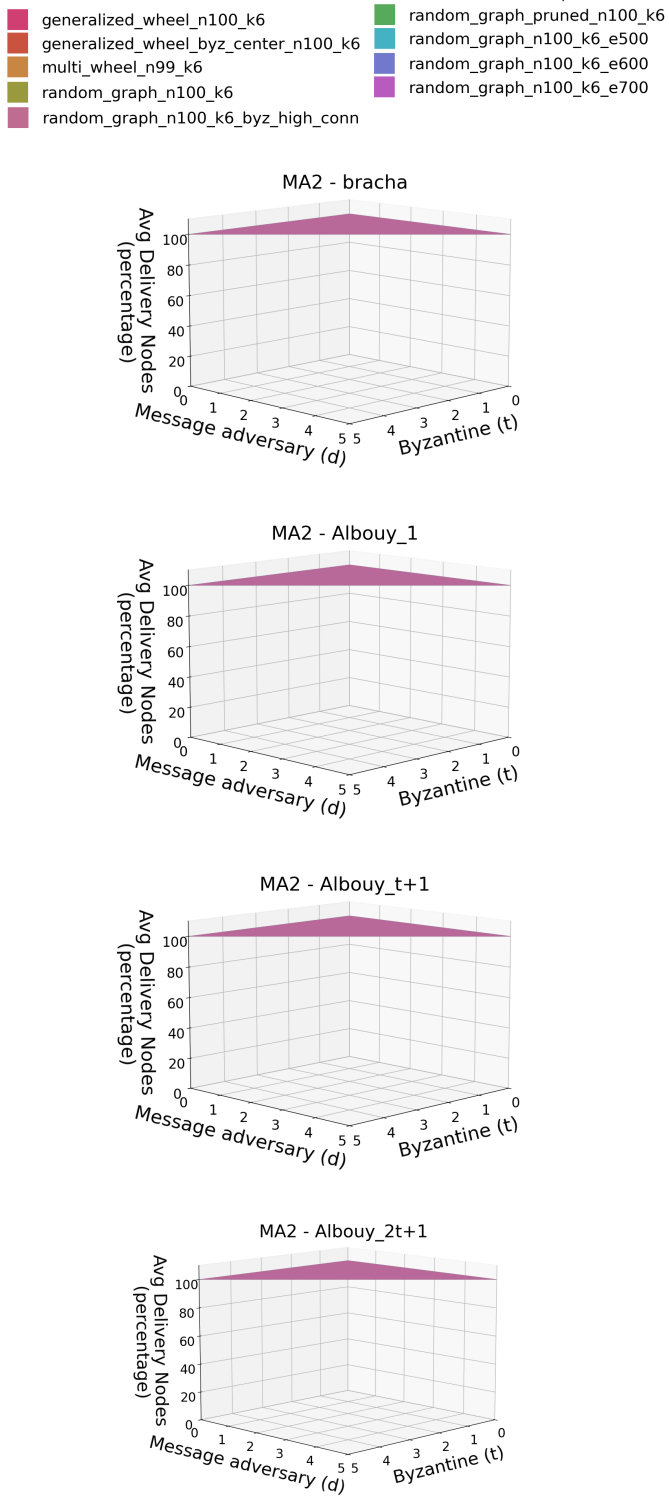


Fig. 6: 3D Plots comparing the avg_CND of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA2.

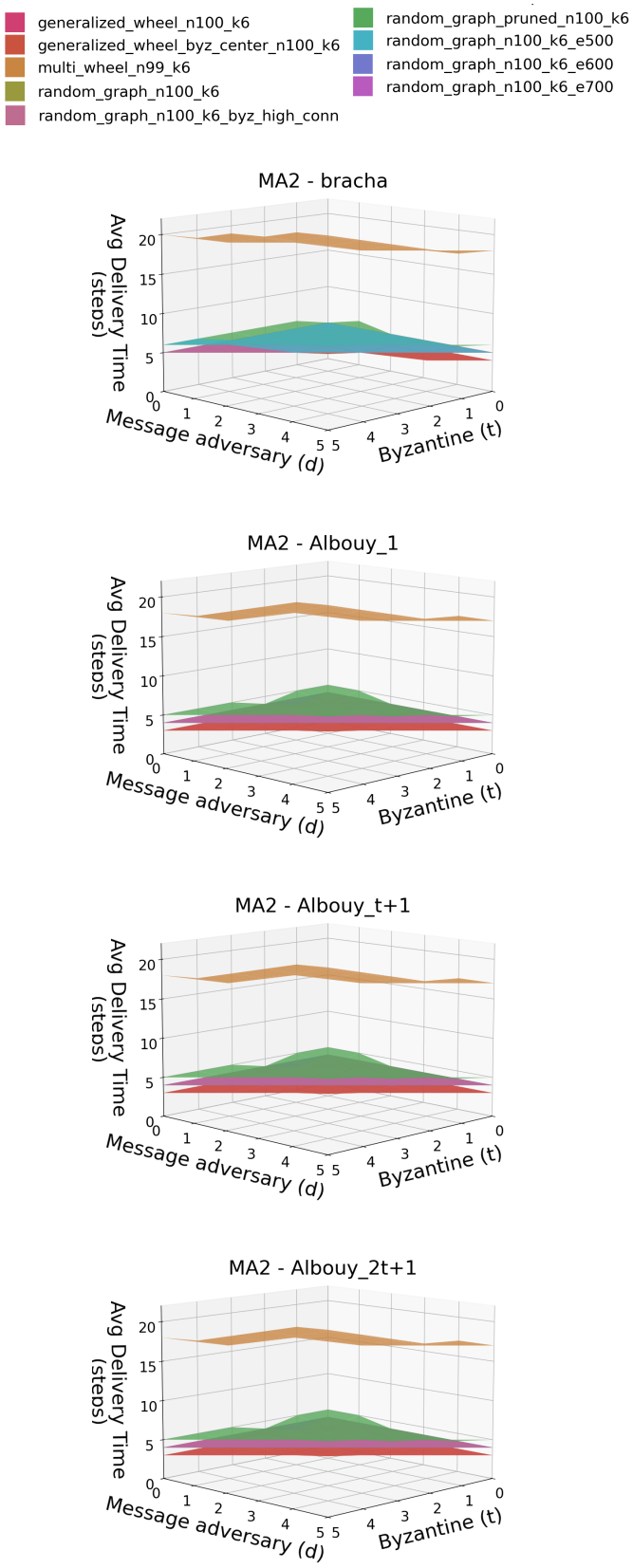


Fig. 7: 3D Plots comparing the avg_TTD of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA2.

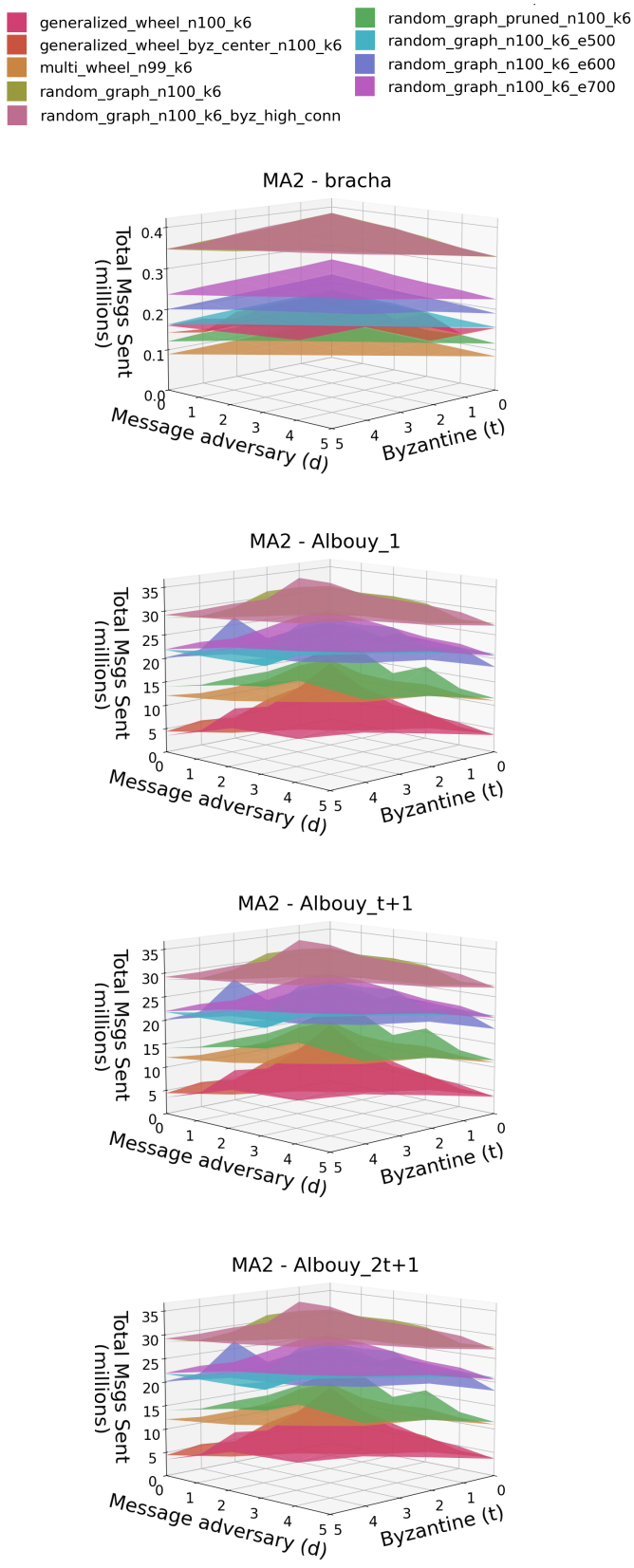


Fig. 8: 3D Plots comparing the tot_MSGS of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA2.

B. MA3 Plots

We now report the results obtained under MA3.

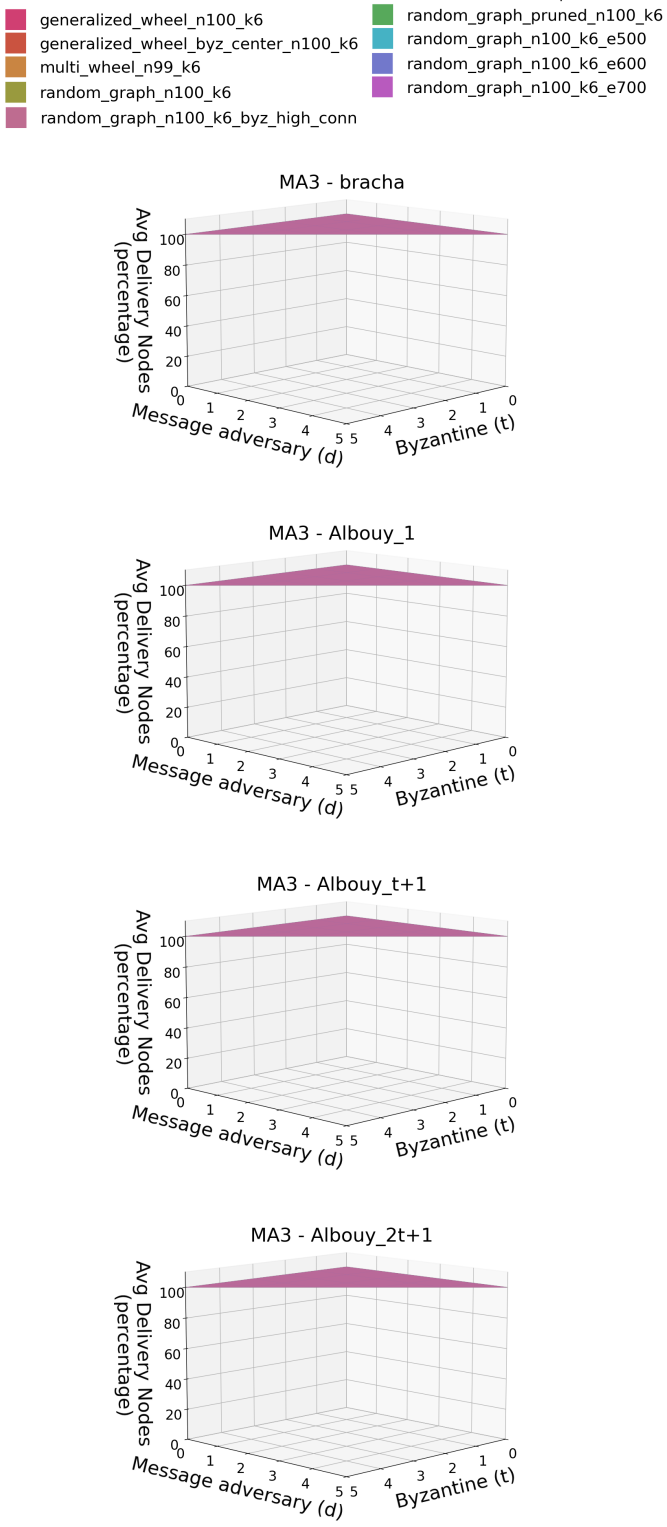


Fig. 9: 3D Plots comparing the avg_CND of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA2.

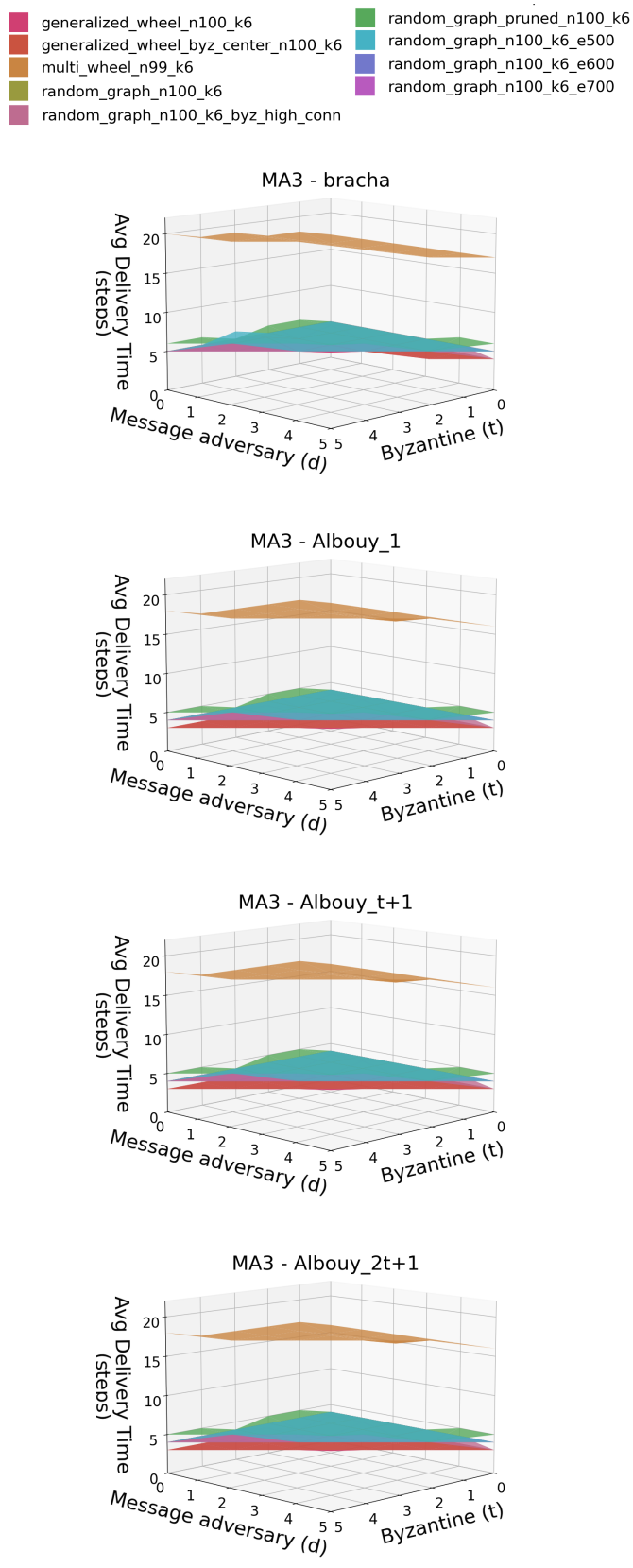


Fig. 10: 3D Plots comparing the avg_TTD of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA2.

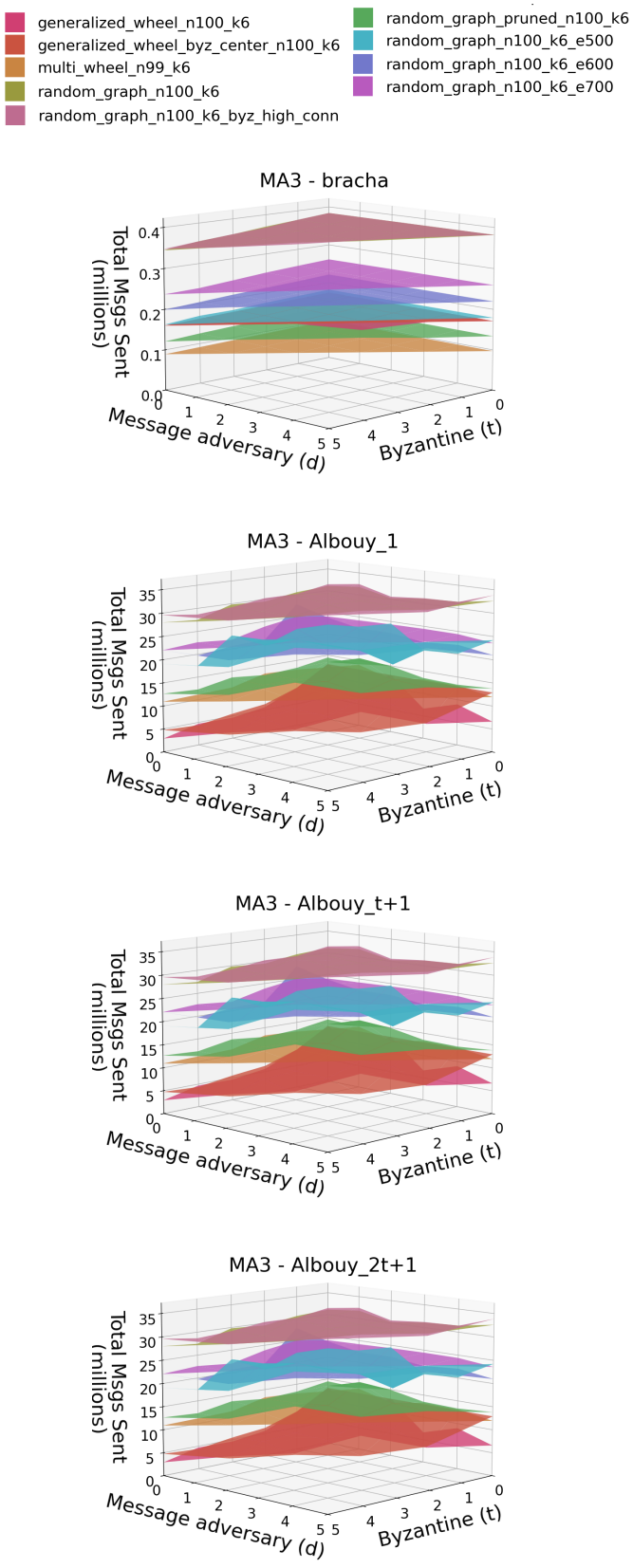


Fig. 11: 3D Plots comparing the tot_MSGS of the four algorithms (Bracha, Albouy_1, Albouy_t+1 and Albouy_2t+1) across all topology when varying (t, d) and considering MA2.