A Modular Approach to Construct Signature-Free BRB Algorithms under a Message Adversary

Timothé Albouy, Davide Frey, Michel Raynal, François Taïani

A Modular Approach to Construct Signature-Free BRB Algorithms under a Message Adversary

Timothé Albouy, Davide Frey, Michel Raynal, François Taïani

Univ Rennes, IRISA, CNRS, Inria, 35042 Rennes, France

Abstract

This paper explores how reliable broadcast can be implemented without signatures when facing a dual adversary that can both corrupt processes and remove messages. More precisely, we consider an asynchronous n-process message-passing system in which up to t processes are Byzantine and where, at the network level, for each message broadcast by a correct process, an adversary can prevent up to d processes from receiving it (the integer d defines the power of the message adversary). So, unlike previous works, this work considers that not only can computing entities be faulty (Byzantine processes), but, in addition, that the network can also lose messages. To this end, the paper adopts a modular strategy and first introduces a new basic communication abstraction denoted $k2\ell$-cast, which simplifies quorum engineering, and studies its properties in this new adversarial context. Then, the paper deconstructs existing signature-free Byzantine-tolerant asynchronous broadcast algorithms and, with the help of the $k2\ell$-cast communication abstraction, reconstructs versions of them that tolerate both Byzantine processes and message adversaries. Interestingly, these reconstructed algorithms are also more efficient than the Byzantine-tolerant-only algorithms from which they originate.

Keywords: Asynchronous system, Byzantine processes, Communication abstraction, Delivery predicate, Fault-Tolerance, Forwarding predicate, Message adversary, Message loss, Modularity, Quorum, Reliable broadcast, Signature-free algorithm, Two-phase commit.

1 Introduction

Context: reliable broadcast and message adversaries. Reliable broadcast (RB) is a fundamental abstraction in distributed computing that lies at the core of many higher-level constructions (including distributed memories, distributed agreement, and state machine replication). Essentially, RB requires that non-faulty (i.e., correct) processes agree on the set of messages they deliver so that this set includes at least all the messages that correct processes have broadcast.

In a failure-free system, implementing reliable broadcast on top of an asynchronous network is relatively straightforward [26]. If processes may fail, and in particular if failed processes may behave arbitrarily (a failure known as Byzantine [20, 25]), implementing reliable broadcast becomes far from trivial as Byzantine processes may collude to fool correct processes [28]. An algorithm that solves reliable broadcast in the presence of Byzantine processes is known as implementing BRB (Byzantine reliable broadcast).

BRB in asynchronous networks (in which no bound is known over message delays) has been extensively studied over the last forty years [1, 2, 6, 10, 11, 17, 19, 22, 21, 24, 28]. Existing BRB algorithms typically assume they execute over a *reliable* point-to-point network, i.e., a network in which sent messages are eventually received. This is a reasonable assumption as most unreliable networks can be made reliable using re-transmissions and acknowledgments (e.g. a timeout-free version of the TCP protocol).

This work takes a drastic turn away from this usual assumption and explores how BRB might be provided when processes execute on an *unreliable* network that might lose point-to-point messages. Our motivation is threefold: First, in volatile networks (e.g., mobile networks or networks under attack), processes might remain disconnected over long periods (e.g., weeks or months), leading in practice to considerable delays (a.k.a. tail latencies) when using re-transmissions. Because most asynchronous Byzantine-tolerant algorithms exploit intersecting quorums, these tail latencies can potentially limit the performance of BRB algorithms drastically, a well-known phenomenon in systems research [13, 14, 33]. Second, re-transmissions require that correct processes be eventually able to receive messages and cannot, therefore, model the permanent disconnection of correct processes. Finally, this question is interesting in its own right, as it opens up novel trade-offs between algorithm guarantees and network requirements, with potential application to the design of reactive distributed algorithms tolerant to both processes and network failures.

The impact of network faults on distributed algorithms has been studied in several works, in particular using the concept of message adversaries (MA). Message adversaries were initially introduced by N. Santoro and P. Widmayer in [30, 31][1], and then used (sometimes implicitly) in many works (e.g., [4, 3, 12, 16, 29, 27, 31, 32]). Initially proposed for synchronous networks, an MA may suppress point-to-point network messages according to rules that define its power. For instance, a tree MA in a synchronous network might suppress any message except those transiting on an (unknown) spanning tree of the network, with this spanning tree possibly changing in each round.

The message losses that an MA causes differ fundamentally from Byzantine faults. This is because an MA can affect the messages sent by any correct process, and can change the processes it targets during an execution, in contrast to Byzantine corruptions that are tied to a set of fixed processes (which is why MA faults are sometimes dubbed *transient* or *mobile*). For instance, it may be tempting to think that Byzantine-fault-tolerant (BFT) algorithms inherently tolerate message losses from correct processes because they can only afford to wait for at most $n-t$ messages (where n is the total number of processes, and t the upper bound on Byzantine processes). In an asynchronous network, a BFT algorithm could therefore miss up to t messages from correct processes, if those are delayed by the scheduler. This scenario only applies, however, in the particular circumstance where the t faulty processes send messages that are received and accepted as valid by correct recipients. This caveat is fundamental. If the faulty processes remain silent or send contradicting messages (if they are Byzantine), then a BFT algorithm cannot afford to lose t messages from correct processes.

Content of the paper. This paper combines a Message Adversary with Byzantine processes, and studies the signature-free implementation of Byzantine Reliable Broadcast (BRB) in an asynchronous, fully connected network subject to this MA and to at most t Byzantine faults. The MA models lossy connections by preventing up to d point-to-point messages from reaching their recipient every time a correct process seeks to communicate with the rest of the network.[2]

To limit as much as possible our working assumptions, we further assume that the computability power of the adversary is unbounded (except for the cryptography-based algorithm presented in Section 6), which precludes the use of signatures. (We do assume, however, that each point-to-point communication channel is authenticated.)[3]

[1] Where the terminology *communication failure model* and *ubiquitous faults* is used instead of MA. While we consider only message losses, the work of Santoro and Widmayer also considers message additions and corruptions.

[2] A close but different notion was introduced by Dolev in [15], which considers static κ-connected networks. If the adversary selects statically, for each correct sender, d correct processes that do not receive any of this sender's messages, the proposed model includes Dolev's model where $\kappa = n - d$.

[3] Let us mention that the problem of designing an MA-tolerant BRB has been solved in [4] by leveraging digital signatures within a monolithic algorithm. Finding a signature-free counterpart remained, however, an open question, which we answer positively in this paper using a modular strategy.

This represents a particularly challenging environment, as the MA may target different correct processes every time the network is used or focus indefinitely on the same (correct) victims. Further, the Byzantine processes may collude with the MA for maximal impact.

For clarity, in the remainder of the paper, we simply call *messages* the point-to-point network messages used internally by a BRB algorithm. (The MA may suppress these messages.) We distinguish these messages from the messages the BRB algorithm seeks to disseminate, which we call "*application messages*" (*app-messages* for short). In such a context, the paper presents the following results.

- It first introduces a new modular abstraction, named $k2\ell$-cast, which appears to be a foundational building block to implement BRB abstractions (with or without the presence of an MA). This communication abstraction systematically dissociates the predicate used to forward (network) messages from the predicate that triggers the delivery of an app-message, and lies at the heart of the work presented in the paper. When proving the $k2\ell$-cast communication abstraction, the paper presents an in-depth analysis of the power of an adversary that controls at most t Byzantine processes and an MA of power d.
- Then, the paper deconstructs two signature-free BRB algorithms (Bracha's [10] and Imbs and Raynal's [19] algorithms) and reconstructs versions of them that tolerate *both* Byzantine processes and MA. Interestingly, when considering Byzantine failures only, these deconstructed versions use smaller quorum sizes and are, therefore, more efficient than their initial counterparts.

So, this paper is not only the first to present signature-free BRB algorithms in the context of asynchrony and MA but also the first to propose an intermediary communication abstraction that allows us to obtain efficient BRB algorithms. For clarity, we give in Table 1 the list of acronyms and notations used in this paper.

Acronyms	**Meaning**
BRB	Byzantine-tolerant reliable broadcast
MA	Message adversary
MBRB	Message adversary- and Byzantine-tolerant reliable broadcast
Notations	**Meaning**
n	number of processes in the network
t	upper bound on the number of Byzantine processes
d	power of the message adversary
c	effective number of correct processes in a run ($n - t \leq c \leq n$)
k	minimal nb of correct processes that $k2\ell$-cast a message
ℓ	minimal nb of correct processes that $k2\ell$-deliver a message
k'	minimal nb of correct $k2\ell$-casts if there is a correct $k2\ell$-delivery
δ	`true` iff no-duplicity is guaranteed, `false` otherwise
q_d	size of the $k2\ell$-delivery quorum
q_f	size of the forwarding quorum
$single$	`true` iff only a single message can be endorsed, `false` otherwise

Table 1: Acronyms and notations

Roadmap. The paper is composed of 7 sections and four appendices. Section 2 describes the underlying computing model. Section 3 presents the $k2\ell$-cast abstraction and its properties. Section 4 defines the MA-tolerant BRB communication abstraction. Section 5 shows that thanks to the $k2\ell$-cast abstraction, existing BRB algorithms can give rise to MA-tolerant BRB algorithms which, when $d = 0$, are more efficient than the BRB algorithms they originate from. Section 6 presents a signature-based implementation of $k2\ell$-cast that possesses optimal guarantees. Finally, Section 7 concludes the paper. Due to

page limitations, some contributions are presented in appendices: namely some proofs and a numerical evaluation of the $k2\ell$-cast abstraction.

2 Computing Model

Process model. The system is composed of n asynchronous sequential processes denoted $p_1, ..., p_n$. Each process p_i has a distinct identity, known to other processes. For simplicity and without loss of generality, we assume that i is the identity of p_i.

In terms of faults, up to $t \geq 0$ processes can be Byzantine, where a Byzantine process is a process whose behavior does not follow the code specified by its algorithm [20, 25]. Byzantine processes may collude to fool non-Byzantine processes (also called correct processes). In this model, the premature stop (crash) of a process is a Byzantine failure. In the following, given an execution, c denotes the effective number of processes that behave correctly in that execution. We always have $n - t \leq c \leq n$. While this number remains unknown to correct processes, it is used to analyze and characterize (more precisely than using its worse value $n - t$) the guarantees provided by the proposed algorithms.

Finally, the processes have no access to random numbers, and their computability power is unbounded. Hence, the algorithms presented in the paper are deterministic and signature-free (except the signature-based algorithm presented in Section 6).

Communication model. Processes communicate by exchanging messages through a fully connected asynchronous point-to-point network, assumed to be reliable in the sense it neither corrupts, duplicates, nor creates messages. As far as messages losses are concerned, the network is under the control of an adversary (see below) that can suppress messages.

Let MSG be a message type and v the associated value. A process can invoke the best-effort broadcast macro-operation denoted ur_broadcast(MSG(v)), which is a shorthand for "**for all** $j \in \{1, \cdots, n\}$ **do** send MSG(v) **to** p_j **end for**". Correct processes are assumed to invoke ur_broadcast to send messages. When they do, we say that the messages are *ur-broadcast* and *received*. The operation ur_broadcast(MSG(v)) is not reliable. For example, if the invoking process crashes during its invocation, an arbitrary subset of processes receive the message MSG(v). Moreover, due to its very nature, a Byzantine process can send fake messages without using the macro-operation ur_broadcast.

Message adversary. Let d be an integer constant such that $0 \leq d < n - t$. The communication network is controlled by an MA (as defined in Section 1), which eliminates messages ur-broadcast by correct processes, so these messages are lost. More precisely, when a correct process invokes ur_broadcast(MSG(v)), the MA is allowed to arbitrarily suppress up to d copies of the message MSG(v) intended to correct processes[4]. This means that, although the sender is correct, up to d correct processes may miss the message MSG(v). The extreme case $d = 0$ corresponds to the case where no message is lost.

As an example, let us consider a set D of correct processes, where $1 \leq |D| \leq d$, such that during some period of time, the MA suppresses all the messages sent to them. It follows that, during this period of time, this set of processes appears as being input-disconnected from the other correct processes. Note that the size and the content of D can vary with time and are never known by the correct processes.

[4]Note that this message adversary is not limited to algorithms that use the ur_broadcast macro-operation. The same adversary can be equivalently defined for an operation ur_multicast that sends a message to a dynamically defined subset of processes (be it multiple recipients or only one in the case of unicast), by stipulating that the MA can still suppress up to d copies of this message. In this case, the most robust way for correct processes to disseminate a message is to send it to all processes, i.e. to fall back on a ur_broadcast operation.

3 $k2\ell$-Cast Abstraction

Signature-free BRB algorithms [8, 10, 19] often rely on successive waves of internal messages (e.g. the ECHO or READY messages of Bracha's algorithm [10]) to provide safety and liveness. Each wave is characterized by a threshold-based predicate that triggers the algorithm's next phase when fulfilled (e.g. enough ECHO messages for the same app-message m).

In this section, we introduce, implement, and prove a new modular abstraction, called $k2\ell$-cast, that encapsulates a wave/thresholding mechanism that is both Byzantine- and MA-tolerant. As previously announced, we then use this abstraction to reconstruct MA-tolerant BRB algorithms in Section 5 from two existing BRB algorithms [10, 19].

3.1 Definition

$k2\ell$-cast (for k-to-ℓ-cast) is a many-to-many communication abstraction[5]. Intuitively, it relates the number k of correct processes that send a message m (we say that these processes $k2\ell$-*cast* m) with the number ℓ of correct processes that deliver m (we say that they $k2\ell$-*deliver* m). Both k and ℓ are subject to thresholding constraints: enough correct processes must $k2\ell$-cast a message for it to be $k2\ell$-delivered at least once; and as soon as one (correct) $k2\ell$-delivery occurs, some minimal number of correct processes are guaranteed to $k2\ell$-deliver as well.

More formally, $k2\ell$-cast is a multi-shot abstraction, i.e. each app-message m that is $k2\ell$-cast or $k2\ell$-delivered is associated with an identity id. (Typically, such an identity is a pair consisting of a process identity and a sequence number.) It provides two operations, $k2\ell$_cast and $k2\ell$_deliver, whose behavior is defined by the values of four parameters: three integers k', k, ℓ, and a Boolean δ. This behavior is captured by the following six properties:

- Safety:
  - $k2\ell$-VALIDITY. If a correct process p_i $k2\ell$-delivers an app-message m with identity id, then at least k' correct processes $k2\ell$-cast m with identity id.
  - $k2\ell$-NO-DUPLICATION. A correct process $k2\ell$-delivers at most one app-message m with identity id.
  - $k2\ell$-CONDITIONAL-NO-DUPLICITY. If the Boolean δ is `true`, then no two different correct processes $k2\ell$-deliver different app-messages with the same identity id.
- Liveness[6]:
  - $k2\ell$-LOCAL-DELIVERY. If at least k correct processes $k2\ell$-cast an app-message m with identity id and no correct process $k2\ell$-casts an app-message $m' \neq m$ with identity id, then at least one correct process $k2\ell$-delivers the app-message m with identity id.
  - $k2\ell$-WEAK-GLOBAL-DELIVERY. If a correct process $k2\ell$-delivers an app-message m with identity id, then at least ℓ correct processes $k2\ell$-deliver an app-message m' with identity id (each of them possibly different from m).
  - $k2\ell$-STRONG-GLOBAL-DELIVERY. If a correct process $k2\ell$-delivers an app-message m with identity id, and no correct process $k2\ell$-casts an app-message $m' \neq m$ with identity id, then at least ℓ correct processes $k2\ell$-deliver the app-message m with identity id.

[5] An example of this family is the binary reliable broadcast introduced in [23], which is defined by specific delivery properties—not including MA-tolerance—allowing binary consensus to be solved efficiently with the help of a common coin.

[6] The liveness properties comprise a *local* delivery property that provides a necessary condition for the $k2\ell$-delivery of an app-message by at least *one* correct process, and two *global* delivery properties that consider the collective behavior of correct processes.

This specification is *parameterized* in the sense that each tuple (k', k, ℓ, δ) defines a specific communication abstraction with different guarantees. This versatility explains why the $k2\ell$-cast abstraction can be used to produce highly compact reconstructions of existing BRB algorithms, rendering them MA-tolerant in the process (using four and three lines of pseudo-code respectively, see Section 5). Despite this versatility, however, we will see in Section 3.2 that $k2\ell$-cast can be implemented using a single (parameterized) algorithm, underscoring the fundamental commonalities of MA-tolerant BRB algorithms.

Intuitively, the parameters k', k, and ℓ hobble the disruption power of the Byzantine/MA adversary by setting limits on the number of correct processes that are either required or guaranteed to be involved in one communication "wave" (corresponding to one identity id). k' sets the minimal number of correct processes that must $k2\ell$-cast for any $k2\ell$-delivery to occur: it thus limits the ability of the Byzantine/MA adversary to trigger spurious $k2\ell$-deliveries. The role of k is symmetrical. It guarantees that some $k2\ell$-delivery will necessarily occur if k correct processes $k2\ell$-cast some message. It thus prevents the adversary from silencing correct processes as soon as some critical mass of them participates. Finally, ℓ captures a "quite-a-few-or-nothing" guarantee that mirrors the traditional "all-or-nothing" delivery guarantee of traditional BRB. As soon as one correct $k2\ell$-delivery occurs (for some identity id), then ℓ correct processes must also $k2\ell$-deliver (with the same identity).

The fourth parameter, δ, is a flag that when `true` enforces agreement between $k2\ell$-deliveries. When $\delta =$ `true`, the $k2\ell$-CONDITIONAL-NO-DUPLICITY property implies that all the app-messages m' involved in the $k2\ell$-WEAK-GLOBAL-DELIVERY property are equal to m.

3.2 A Signature-Free Implementation of $k2\ell$-Cast

Among the many possible ways of implementing $k2\ell$-cast, this section presents a quorum-based[7] signature-free implementation[8] of the abstraction. To overcome the disruption caused by Byzantine processes and message losses from the MA, our algorithm uses the ur-broadcast primitive (cf. our communication model in Sec. 2) to accumulate and forward ENDORSE messages before deciding whether to deliver. Forwarding and delivery are triggered by *two thresholds* (a pattern also found, for instance, in Bracha's BRB algorithm [10]):

- A first threshold, q_d, triggers the delivery of an app-message m when enough ENDORSE messages supporting m have been received.
- A second threshold, q_f, which is lower than q_d, controls how ENDORSE messages are forwarded during the algorithm's execution.

Forwarding, which is controlled by q_f, amplifies how correct processes react to ENDORSE messages, and is instrumental to ensure the algorithm's liveness. As soon as some critical "mass" of agreeing ENDORSE messages accumulates within the system, forwarding triggers a chain reaction which guarantees that a minimum number of correct processes eventually $k2\ell$-deliver the corresponding app-message.

More concretely, our algorithm provides an object (SigFreeK2LCast, Alg. 1), instantiated using the function SigFreeK2LCast$(q_d, q_f, single)$, using three input parameters:

- q_d: the number of matching ENDORSE messages that must be received from distinct processes in order to $k2\ell$-deliver an app-message.

[7] In this paper, a quorum is a set of processes that (at the implementation level) ur-broadcast the same message. This definition takes quorums in their ordinary sense. In a deliberative assembly, a quorum is the minimum number of members that must vote the same way for an irrevocable decision to be taken. Let us notice that this definition does not require quorum intersection. However, if quorums have a size greater than $\frac{n+t}{2}$, the intersection of any two quorums contains, despite Byzantine processes, at least one correct process [10, 28].

[8] Another $k2\ell$-cast implementation, which uses digital signatures and allows to reach optimal values for k and ℓ, is presented in Section 6.

object SigFreeK2LCast(q_d, q_f, $single$) **is**

(1) **operation** $k2\ell$_cast(m, id) **is**
(2) **if** (ENDORSE$(-, id)$ not already ur-broadcast)
(3) **then** ur_broadcast(ENDORSE(m, id))
(4) **end if**.

(5) **when** ENDORSE(m, id) **is** received **do**
% ——————— forwarding step ———————
(6) **if** (ENDORSE(m, id) received from at least q_f processes
$\wedge$ ($(\neg single \wedge$ ENDORSE(m, id) not already ur-broadcast)
$\vee$ ENDORSE$(-, id)$ not already ur-broadcast))
(7) **then** ur_broadcast(ENDORSE(m, id))
(8) **end if**;
% ——————— delivery step ———————
(9) **if** (ENDORSE(m, id) received from at least q_d processes
$\wedge$ $(-, id)$ not already $k2\ell$-delivered)
(10) **then** $k2\ell$_deliver(m, id)
(11) **end if**.

end object.

Algorithm 1: Signature-free $k2\ell$-cast (code for p_i)

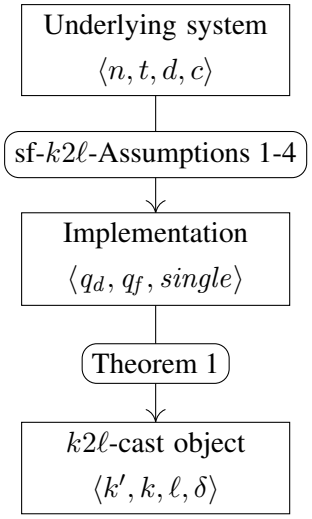


Figure 1: From the system parameters to a $k2\ell$-cast implementation

- q_f: the number of matching ENDORSE messages that must be received from distinct processes for the local p_i to endorse the corresponding app-message (if it has not yet).
- $single$: a Boolean that controls whether a given correct process can endorse different app-messages for the same identity id ($single =$ `false`), or not ($single =$ `true`).

The algorithm provides the operations $k2\ell$_cast and $k2\ell$_deliver. Given an app-message m with identity id, the operation $k2\ell$_cast(m, id) ur-broadcasts ENDORSE(m, id) provided p_i has not yet endorsed any different app-message for the same identity id (lines 2-4). When p_i receives a message ENDORSE(m, id), its executes two steps. If the forwarding quorum q_f has been reached, p_i first retransmits ENDORSE(m, id) (Forwarding step, lines 6-8). Then, if the $k2\ell$-delivery quorum q_d is attained, p_i $k2\ell$-delivers m (Delivery step, lines 9-11).

For brevity, we define $\alpha = n + q_f - t - d - 1$. Given an execution defined by the system parameters n, t, d, and c, Alg. 1 requires the following assumptions to hold for the input parameters q_f and q_d of a $k2\ell$-cast instance (a global picture linking all parameters is presented in Fig. 1). The prefix "sf" stands for signature-free.

- sf-$k2\ell$-Assumption 1: $c - d \geq q_d \geq q_f + t \geq 2t + 1$,
- sf-$k2\ell$-Assumption 2: $\alpha^2 - 4(q_f - 1)(n - t) \geq 0$,
- sf-$k2\ell$-Assumption 3: $\alpha(q_d - 1) - (q_f - 1)(n - t) - (q_d - 1)^2 > 0$,
- sf-$k2\ell$-Assumption 4: $\alpha(q_d - 1 - t) - (q_f - 1)(n - t) - (q_d - 1 - t)^2 \geq 0$.

In particular, the safety of Alg. 1 algorithm relies solely on sf-$k2\ell$-Assumption 1, while its liveness relies on all four of them. sf-$k2\ell$-Assumption 2 through 4 constrain the solutions of a second-degree inequality resulting from the combined action of the MA, the Byzantine processes, and the message-forwarding behavior of Alg. 1. We show in Appendix B that, in practical cases, these assumptions can be satisfied by a bound of the form $n > \lambda t + \xi d + f(t, d)$, where $\lambda, \xi \in \mathbb{N}$ and $f(t, 0) = f(0, d) = 0$. Together, the assumptions allow Alg. 1 to provide a $k2\ell$-cast abstraction (with values of the parameters k', k, ℓ, and δ defining a specific $k2\ell$-cast instance) as stated by the following theorem.

Theorem 1 ($k2\ell$-CORRECTNESS)**.** *If sf-$k2\ell$-Assumptions* 1–4 *are verified, Alg.* 1 *implements $k2\ell$-cast with the following guarantees:*

- $k2\ell$-VALIDITY *with* $k' = q_f - n + c$,
- $k2\ell$-NO-DUPLICATION,
- $k2\ell$-CONDITIONAL-NO-DUPLICITY *with* $\delta = \left(q_f > \frac{n+t}{2}\right) \vee \left(single \wedge q_d > \frac{n+t}{2}\right)$,
- $k2\ell$-LOCAL-DELIVERY *with* $k = \left\lfloor \frac{c(q_f-1)}{c-d-q_d+q_f} \right\rfloor + 1$,
- $\left\{ \begin{array}{ll} \textit{if } single = \texttt{false}, & k2\ell\text{-WEAK-GLOBAL-DELIVERY} \\ \textit{if } single = \texttt{true}, & k2\ell\text{-STRONG-GLOBAL-DELIVERY} \end{array} \right\}$ *with* $\ell = \left\lceil c\left(1 - \frac{d}{c-q_d+1}\right)\right\rceil$.

3.3 Proof of Algorithm 1

The proofs of the $k2\ell$-cast safety properties stated in Theorem 1 ($k2\ell$-VALIDITY, $k2\ell$-NO-DUPLICATION, and $k2\ell$-CONDITIONAL-NO-DUPLICITY) are fairly straightforward. To save space, these proofs (Lemmas 10-13) are provided in Appendix A.1.

The proofs of the $k2\ell$-cast liveness properties ($k2\ell$-LOCAL-DELIVERY, $k2\ell$-WEAK-GLOBAL-DELIVERY, $k2\ell$-STRONG-GLOBAL-DELIVERY) are sketched informally below (Lemmas 1-9). Their full development can be found in Appendix A.2.

When seeking to violate the liveness properties of $k2\ell$-cast, the attacker can use the MA to control in part how many ENDORSE messages are received by each correct process, thus interfering with the quorum mechanisms defined by q_d and q_f. To analyze the joint effect of this interference with Byzantine faults, our proofs consider seven well-chosen subsets of correct processes (A, B, C, U, F, NF, and NB, depicted in Fig. 2a).

These subsets are defined for an execution of Alg. 1 in which k_I correct processes $k2\ell$-cast (m, id) (the I in k_I is for "Initial"), and ℓ_e correct processes receive at least q_d message ENDORSE(m, id). The first three subsets, A, B, and C, partition correct processes based on the number of ENDORSE(m, id) messages they receive.

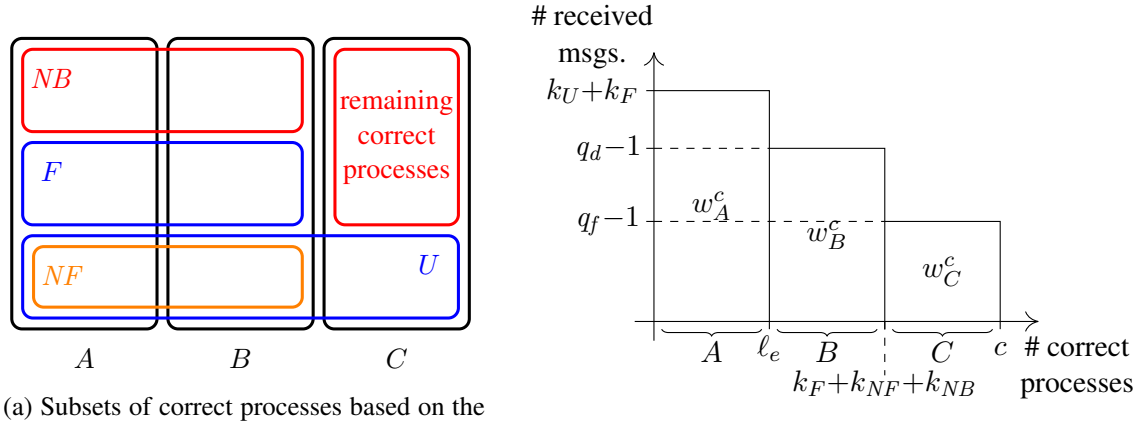
- A contains the ℓ_e correct processes that receive at least q_d ENDORSE(m, id) messages (be it from correct or from Byzantine processes), and thus $k2\ell$-deliver some message.[9]
- B contains the correct processes that receive at least q_f but less than q_d ENDORSE(m, id) messages and thus do not $k2\ell$-deliver (m, id).
- C contains the remaining correct processes that receive less than q_f ENDORSE(m, id) messages. They neither forward nor deliver any message for identity id (since $q_f \leq q_d$).

In our proofs, we count how many messages ENDORSE(m, id) ur-broadcast by correct processes are received by the processes of A (resp. B and C). We note these quantities w_A^c, w_B^c, and w_C^c, and use them to bootstrap our proofs using bounds on messages (see below).

The last four subsets intersect with A, B and C, and distinguish correct processes based on the ur-broadcast operations they perform.

- U consists of the correct processes that ur-broadcast ENDORSE(m, id) at line 3.

[9] Because of the condition at line 9, these processes do not necessarily $k2\ell$-deliver (m, id), but all do $k2\ell$-deliver an app-message for identity id.



(a) Subsets of correct processes based on the number of received ENDORSE messages (A, B and C) and based on their ur-broadcast actions (U, F, NF, and NB)

(b) Distribution of ENDORSE messages among correct processes of A, B, and C, sorted by decreasing number of ENDORSE messages received

Figure 2: Subsets of correct processes and distribution of ENDORSE messages among them

- F denotes the correct processes of $A \cup B$ that ur-broadcast ENDORSE(m, id) at line 7 (i.e., they perform forwarding).
- NF denotes the correct processes of $A \cup B$ that ur-broadcast ENDORSE(m, id) at line 3.
- NB denotes the correct processes of $A \cup B$ that never ur-broadcast ENDORSE(m, id), be it at line 3 or at line 7. These processes have received at least q_f messages ENDORSE(m, id), but do not forward ENDORSE(m, id), because they have already ur-broadcast ENDORSE(m', id) at line 3 or at line 7 for an app-message $m' \neq m$.

Proof strategy. We note $k_U = |U|$, $k_F = |F|$, $k_{NF} = |NF|$, $k_{NB} = |NB|$. Observe that $k_U \leq k_I$ and $k_{NF} \leq k_U$, since all (correct) processes in U and NF invoke $k2\ell$_cast. Also, $(k_U + k_F)$ represents the total number of correct processes that ur-broadcast a message ENDORSE(m, id). Fig. 2b illustrates how these quantities constrain the distribution of ENDORSE messages across A, B and C. Our core proof strategy consists in bounding the areas shown in Fig. 2b. (For instance, observe that $w_A^c \leq |A| \times (k_U + k_F)$, since each of the ℓ_e correct processes in A can receive at most one ENDORSE message from each of the $(k_U + k_F)$ correct processes that send them.) This reasoning on bounds yields a polynomial involving $\ell_e = |A|$, k_I, and k_U, whose roots can then be constrained to yield the liveness guarantees required by the $k2\ell$-cast specification.

Observation. In the same way we have bounded w_A^c, we can also bound w_B^c by observing that there are $(k_{NF} + k_{NB} + k_F - \ell_e)$ processes in B and that each can receive at most $q_d - 1$ ENDORSE messages. Similarly, we can bound w_C^c by observing that the $(c - k_{NF} - k_{NB} - k_F)$ processes of C can receive at most $q_f - 1$ ENDORSE messages. Thus:

$$w_A^c \leq (k_U + k_F)\ell_e, \tag{1}$$
$$w_B^c \leq (q_d - 1)(k_{NF} + k_{NB} + k_F - \ell_e), \tag{2}$$
$$w_C^c \leq (q_f - 1)(c - k_{NF} - k_{NB} - k_F). \tag{3}$$

Moreover, the MA cannot suppress more than d copies of each individual ENDORSE message ur-broadcast to the c correct processes. Thus, the total number of ENDORSE messages received by correct processes $(w_A^c + w_B^c + w_C^c)$ is such that:

$$w_A^c + w_B^c + w_C^c \geq (k_U + k_F)(c - d). \tag{4}$$

Lemma 1. $\ell_e \times (k_U + k_F - q_d + 1) \geq (k_U + k_F)(c - d - q_d + q_f) - c(q_f - 1) - k_{NB}(q_d - q_f)$.

Proof sketch. We get this result by combining (1), (2), (3) and (4), and using sf-$k2\ell$-Assumption 1 with the fact that $k_{NF} \leq k_U$. (Full derivations in Appendix A.2.) □

Lemma 2. *If no correct process $k2\ell$-casts (m', id) with $m' \neq m$, then no correct process forwards* ENDORSE(m', id) *at line* 7 *(and then $k_{NB} = 0$).* (Proof in Appendix A.2.)

Lemma 3 ($k2\ell$-LOCAL-DELIVERY)**.** *If at least $k = \left\lfloor \frac{c(q_f-1)}{c-d-q_d+q_f} \right\rfloor + 1$ correct processes $k2\ell$-cast an app-message m with identity id and no correct process $k2\ell$-casts any app-message m' with identity id such that $m \neq m'$, then at least one correct process p_i $k2\ell$-delivers m with identity id.*

Proof sketch. From the hypotheses, Lemma 2 helps us determine that $k_{NB} = 0$. Then, the property is proved by contraposition, by assuming that no correct process $k2\ell$-delivers (m, id), which leads us to $\ell_e = 0$. Using prior information and sf-$k2\ell$-Assumption 1, we can rewrite the inequality of Lemma 1 to get the threshold of $k2\ell$-casts above which there is at least one $k2\ell$-delivery. (Full derivations in Appendix A.2.) □

Lemma 4. $(single = \texttt{false}) \implies (k_{NB} = 0)$. (Proof in Appendix A.2.)

Lemma 5. *If at least one correct process $k2\ell$-delivers (m, id) and $x = k_U + k_F$ (the number of correct processes that ur-broadcast* ENDORSE(m, id) *at line* 3 *or* 7*), then $x \geq q_d - t$ and $x^2 - x(c - d + q_f - 1 - k_{NB}) \geq -(c - k_{NB})(q_f - 1)$.*

Proof sketch. We prove this lemma by counting the total number of messages (sent by Byzantine or correct processes) that are received by the processes of A, and by using (1), (3) (4), and sf-$k2\ell$-Assumption 1. (Full derivations in Appendix A.2.) □

Lemma 6. *If $k_{NB} = 0$, and at least one correct process $k2\ell$-delivers (m, id), then $k_U + k_F \geq q_d$.*

Proof sketch. Given that $k_{NB} = 0$, we can rewrite the inequality of Lemma 5, which gives us a second-degree polynomial (where $x = k_U + k_F$ is the unknown variable). We compute its roots and show that the smaller one contradicts Lemma 5, and that the larger one is greater than or equal to q_d. The fact that x must be greater than or equal to the larger root to satisfy Lemma 5 proves the lemma. (Full derivations in Appendix A.2.) □

Lemma 7. *If $k_{NB} = 0$ and $k_U + k_F \geq q_d$, then at least $\left\lceil c\left(1 - \frac{d}{c-q_d+1}\right)\right\rceil$ correct processes $k2\ell$-deliver some app-message with identity id (not necessarily m).*

Proof sketch. From the hypotheses, we can rewrite the inequality of Lemma 1 to get a lower bound on ℓ_e. Using sf-$k2\ell$-Assumption 3, we can determine that this lower bound is decreasing with the number of ur-broadcasts by correct processes ($x = k_U + k_F$). Hence, this lower bound is minimum when x is maximum, that is, when $x = c$. This gives us the minimum number of correct processes that $k2\ell$-deliver under the given hypotheses. (Full derivations in Appendix A.2.) □

Lemma 8 ($k2\ell$-WEAK-GLOBAL-DELIVERY)**.** *If $single = \texttt{false}$, and a correct process $k2\ell$-delivers an app-message m with identity id, then at least $\ell = \left\lceil c\left(1 - \frac{d}{c-q_d+1}\right)\right\rceil$ correct processes $k2\ell$-deliver an app-message m' with identity id (each possibly different from m).*

Proof sketch. As $single = \texttt{false}$ and one correct process $k2\ell$-delivers (m, id), Lemmas 4 and 6 apply, and we have $k_{NB} = 0$ and $k_U + k_F \geq q_d$. This provides the prerequisites for Lemma 7, which concludes the proof. (Full derivations in Appendix A.2.) □

Lemma 9 ($k2\ell$-STRONG-GLOBAL-DELIVERY)**.** *If $single = \texttt{true}$, and a correct process $k2\ell$-delivers an app-message m with identity id, and no correct process $k2\ell$-casts an app-message $m' \neq m$ with identity id, then at least $\ell = \left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes $k2\ell$-deliver m with identity id.*

Proof sketch. As $single = \texttt{true}$, Lemma 2 holds and implies that $k_{NB} = 0$. As above, Lemma 6 and Lemma 7 hold, yielding the lemma. (Full derivations in Appendix A.2.) □

4 BRB in the Presence of Message Adversary (MBRB): Definition

Before using the $k2\ell$-cast abstraction to reconstruct MA-tolerant BRB algorithms, we first specify what a Byzantine- and MA-tolerant broadcast should precisely achieve. We call such a broadcast an MBR-broadcast (for Message-adversarial Byzantine Reliable Broadcast), or MBRB for short. The MBRB abstraction provides two matching operations, mbrb_broadcast and mbrb_deliver. It is a multishot abstraction, i.e, it associates an identity $\langle sn, i\rangle$ (sequence number, sender identity) with each app-message, and assumes that correct processes never reuse the same sequence number for different mbrb_broadcast invocations.

When, at the application level, a process p_i invokes mbrb_broadcast(m, sn), where m is the app-message, we say it "mbrb-broadcasts (m, sn)". Similarly, when the invocation of mbrb_deliver by p_i returns the tuple (m, sn, j) to the client application (where p_j is the sender process), we say it "mbrb-delivers (m, sn, j)". So, the app-message are *mbrb-broadcast* and *mbrb-delivered*. Because of the MA, we cannot always guarantee that an app-message mbrb-delivered by a correct process is eventually received by all correct processes. Hence, in the MBR-broadcast specification, we introduce a variable ℓ_{MBRB} (reminiscent of the ℓ of $k2\ell$-cast) which indicates the strength of the global delivery guarantee of the primitive: if one correct process mbrb-delivers an app-message, then ℓ_{MBRB} correct processes eventually mbrb-deliver this app-message[10]. MBRB is defined by the following properties:

- Safety:
  - MBRB-VALIDITY. If a correct process p_i mbrb-delivers an app-message m from a correct process p_j with sequence number sn, then p_j mbrb-broadcast m with sequence number sn.
  - MBRB-NO-DUPLICATION. A correct process p_i mbrb-delivers at most one app-message from a process p_j with sequence number sn.
  - MBRB-NO-DUPLICITY. No two distinct correct processes mbrb-deliver different app-messages from a process p_i with the same sequence number sn.
- Liveness:
  - MBRB-LOCAL-DELIVERY. If a correct process p_i mbrb-broadcasts an app-message m with sequence number sn, then at least one correct process p_j eventually mbrb-delivers m from p_i with sequence number sn.
  - MBRB-GLOBAL-DELIVERY. If a correct process p_i mbrb-delivers an app-message m from a process p_j with sequence number sn, then at least ℓ_{MBRB} correct processes mbrb-deliver m from p_j with sequence number sn.

[10] If there is no MA (i.e. $d = 0$), we should have $\ell_{MBRB} = c \geq n - t$.

init: $obj_{\text{E}} \leftarrow$ SigFreeK2LCast($q_d{=}\lfloor\frac{n+t}{2}\rfloor{+}1$, $q_f{=}t{+}1$, $single{=}$`true`);
$obj_{\text{R}} \leftarrow$ SigFreeK2LCast($q_d{=}2t{+}d{+}1$, $q_f{=}t{+}1$, $single{=}$`true`).

(1) **operation** mbrb_broadcast(m, sn) **is** ur_broadcast(INIT(m, sn)).

(2) **when** INIT(m, sn) **is** received **from** p_j **do** $obj_{\text{E}}.k2\ell$_cast(ECHO(m), (sn, j)).

(3) **when** (ECHO(m), (sn, j)) **is** $obj_{\text{E}}.k2\ell$_delivered **do** $obj_{\text{R}}.k2\ell$_cast(READY(m), (sn, i)).

(4) **when** (READY(m), (sn, j)) **is** $obj_{\text{R}}.k2\ell$_delivered **do** mbrb_deliver(m, sn, j).

Algorithm 2: $k2\ell$-cast-based reconstruction of Bracha's BRB algorithm (code of p_i)

It is implicitly assumed that a correct process does not use the same sequence number twice. Let us observe that, as at the implementation level, the MA can always suppress all the messages sent to a fixed set D of d processes, these mbrb-delivery properties are the strongest that can be implemented. More generally, the best-guaranteed value for ℓ_{MBRB} is $c - d$. So, the previous specification boils down to Bracha's specification [10] for $\ell_{MBRB} = c$.

5 $k2\ell$-Cast in Action: From Classical BRB to MA-Tolerant BRB (MBRB) Algorithms

This section uses $k2\ell$-cast to reconstruct two signature-free BRB algorithms [10, 19] initially introduced in a pure Byzantine context (i.e., without any MA). This reconstruction produces Byzantine-MA-tolerant versions of the initial algorithms that implement the MBRB specification of Section 4. Moreover, when $d = 0$, our two reconstructed BRB algorithms are strictly more efficient than the original algorithms that gave rise to them (they terminate earlier).

More precisely, the original and reconstructed versions of Bracha's BRB are identical in terms of communication cost, time complexity, and t-resilience (when $d = 0$). The same comparison holds for the original and reconstructed versions of Imbs and Raynal's BRB. However, both reconstructed BRB algorithms use smaller quorums than their original versions, and therefore require fewer messages to progress. In an actual network, this means a lower latency in practice, as practical networks typically exhibit a long tail distribution of latencies (a phenomenon well-studied by system and networking researchers [13, 14, 33]).

To help readers familiar with the initial algorithms, we use the same message types (INIT, ECHO, READY, WITNESS) as in the original publications. It has been shown in [4] that the MBRB problem can be solved if and only if $n > 3t + 2d$.

5.1 Bracha's BRB algorithm reconstructed

Reconstructed version. Bracha's BRB algorithm comprises three phases. When a process invokes brb_broadcast(m, sn), it disseminates the app-message m an INIT message (first phase). The reception of this message by a correct process triggers its participation in a second phase implemented by the exchange of messages tagged ECHO. Finally, when a process has received ECHO messages from "enough" processes, it enters the third phase, in which READY messages are exchanged, at the end of which it brb-delivers the app-message m. Alg. 2 is a reconstructed version of the Bracha's BRB, which assumes $n > 3t + 2d + 2\sqrt{td}$.

The algorithm requires two instances of $k2\ell$-cast, denoted obj_{E} and obj_{R}, associated with the ECHO messages and the READY messages, respectively. For both these objects, the Boolean $single$ is set to `true`. For the quorums, we have the following:

- obj_{E}: $q_f = t + 1$ and $q_d = \lfloor\frac{n+t}{2}\rfloor + 1$,
- obj_{R}: $q_f = t + 1$ and $q_d = 2t + d + 1$.

init: $obj_{\text{W}} \leftarrow$ SigFreeK2LCast($q_d = \lfloor \frac{n+3t}{2} \rfloor + 3d + 1$, $q_f = \lfloor \frac{n+t}{2} \rfloor + 1$, $single$=`false`).
(1) **operation** mbrb_broadcast(m, sn) **is** ur_broadcast(INIT(m, sn)).
(2) **when** INIT(m, sn) **is** received **from** p_j **do** $obj_{\text{W}}.k2\ell$_cast(WITNESS(m), (sn, j)).
(3) **when** (WITNESS(m), (sn, j)) **is** $obj_{\text{W}}.k2\ell$_delivered **do** mbrb_deliver(m, sn, j).

Algorithm 3: $k2\ell$-cast-based reconstruction of Imbs and Raynal's BRB algorithm (code of p_i)

The integer sn is the sequence number of the app-message m mbrb-broadcast by p_i. The identity of m is consequently the pair $\langle sn, i \rangle$.
Alg. 2 provides $\ell_{MBRB} = \left\lceil c \left(1 - \frac{d}{c-2t-d}\right) \right\rceil$ under:

- B87-Assumption: $n > 3t + 2d + 2\sqrt{td}$;

its proof of correctness can be found in Appendix B.1 (B87 stands for Bracha 1987).

Comparison (Table 2). When $d = 0$, both Bracha's algorithm and its reconstruction use the same quorum size for the READY phase. The quorums of the ECHO phase are however different (Table 2). As the algorithm requires $n > 3t$, we define $\Delta = n - 3t$ as the slack between the lower bound on n and the actual value of n. When considering the forwarding threshold q_f, we have $\lfloor \frac{n+t}{2} \rfloor + 1 = 2t + \lfloor \frac{\Delta}{2} \rfloor + 1 > t + 1$. As a result, the reconstruction of Bracha's algorithm always uses a lower forwarding threshold for ECHO messages than the original. It therefore forwards messages more rapidly and reaches the delivery quorum faster.

Threshold	**Original version** (ECHO phase)	$k2\ell$**-cast-based version** (obj_{E})
Forwarding q_f	$\lfloor \frac{n+t}{2} \rfloor + 1$	$t + 1$
Delivery q_d	$\lfloor \frac{n+t}{2} \rfloor + 1$	$\lfloor \frac{n+t}{2} \rfloor + 1$

Table 2: Bracha's original version vs. $k2\ell$-cast-based reconstruction when $d = 0$

5.2 Imbs and Raynal's BRB algorithm reconstructed

Reconstructed version. Imbs and Raynal's BRB is another BRB implementation, which achieves an optimal good-case latency (only two communication steps) at the cost of a non-optimal t-resilience. Its reconstructed version requires $n > 5t + 12d + \frac{2td}{t+2d}$.

The algorithm requires a single $k2\ell$-cast object, denoted obj_{W}, associated with the WITNESS message, and which is instantiated with $q_f = \lfloor \frac{n+t}{2} \rfloor + 1$ and $q_d = \lfloor \frac{n+3t}{2} \rfloor + 3d + 1$, and the Boolean $single =$ `false`. Similarly to Bracha's reconstructed BRB, an identity of app-message in this algorithm is a pair $\langle sn, i \rangle$ containing a sequence number sn and a process identity i.
Alg. 3 provides $\ell_{MBRB} = \left\lceil c \left(1 - \frac{d}{c - \lfloor \frac{n+3t}{2} \rfloor - 3d}\right) \right\rceil$ under:

- IR16-Assumption: $n > 5t + 12d + \frac{2td}{t+2d}$; (where $t + d > 0$)

its proof of correctness can be found in Appendix B.2 (IR16 stands for Imbs-Raynal 2016).

Comparison (Table 3). Table 3 compares Imbs and Raynal's original algorithm against its $k2\ell$-cast reconstruction for $d = 0$. Recall that this algorithm saves one communication step with respect to Bracha's at the cost of a weaker t-tolerance, i.e., it requires $n > 5t$. As for Bracha, let us define the slack between n and its minimum as $\Delta = n - 5t$, we have $\Delta \geq 1$.

- Let us first consider the size of the forwarding quorum (first line of the table). We have $n - 2t = 3t + \Delta$ and $\lfloor \frac{n+t}{2} \rfloor + 1 = 3t + \lfloor \frac{\Delta}{2} \rfloor + 1$. When $\Delta > 2$, we always have $\Delta > \lfloor \frac{\Delta}{2} \rfloor + 1$, it follows that the forwarding predicate of the reconstructed version is equal or weaker than the one of the original version.
- The same occurs for the size of the delivery quorum (second line of the table). We have $n - t = 4t + \Delta$ and $\lfloor \frac{n+3t}{2} \rfloor + 1 = 4t + \lfloor \frac{\Delta}{2} \rfloor + 1$. So both reconstructed quorums are lower than those of the original version when $\Delta > 2$, making the reconstructed algorithm quicker as soon as $n \geq 5t + 3$. The two versions behave identically for $5t + 3 \geq n \geq 5t + 2$ ($\Delta \in \{1, 2\}$).

Threshold	**Original version** (WITNESS phase)	$k2\ell$**-cast-based version** (obj_{W})
Forwarding q_f	$n - 2t$	$\left\lfloor \frac{n+t}{2} \right\rfloor + 1$
Delivery q_d	$n - t$	$\left\lfloor \frac{n+3t}{2} \right\rfloor + 1$

Table 3: Imbs and Raynal's original version vs. $k2\ell$-cast-based reconstruction when $d = 0$

5.3 Numerical evaluation of the MBRB algorithms

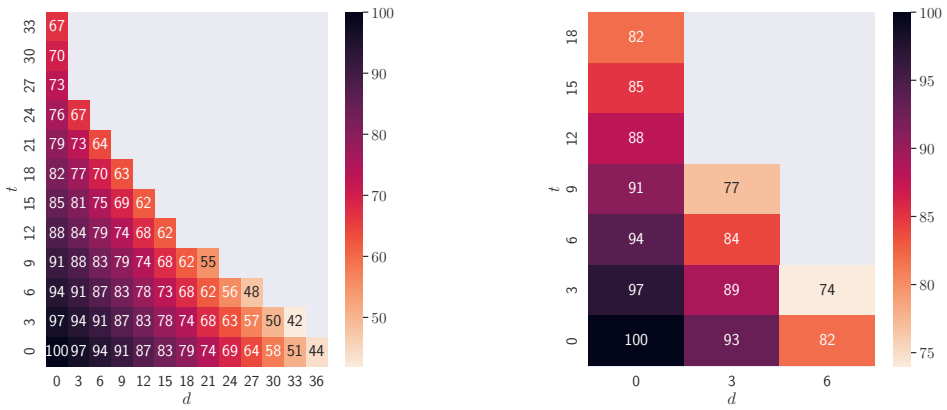
Fig. 3 provides a numerical evaluation of the delivery guarantees of both $k2\ell$-cast-based MBRB algorithms (Algs. 2 and 3) in the presence of Byzantine processes and an MA. Results were obtained for $n = 100$ and $c = n - t$, and show the values of ℓ_{MBRB} for different values of t and d. For instance, Fig. 3a shows that with 6 Byzantine processes and an MA suppressing up to 9 ur-broadcast messages, Alg. 2 ensures the MBRB-GLOBAL-DELIVERY property with $\ell_{MBRB} = 83$. The figures illustrate that the reconstructed Bracha algorithm performs in a broader range of parameter values, mirroring the bounds on n, t, and d captured by B87-Assumption and IR16-Assumption. Nonetheless, both algorithms exhibit values of ℓ_{MBRB} that can support real-world applications in the presence of an MA.

6 A Signature-Based Implementation of $k2\ell$-Cast

This section presents an implementation of $k2\ell$-cast based on digital signatures. The underlying model is the same as that of Section 2 (page 4), except that the computing power of the attacker is now bounded, which allows us to leverage asymmetric cryptography.

6.1 Algorithm

The signature-based algorithm is described in Alg. 4. It uses an asymmetric cryptosystem to sign messages and verify their authenticity. Every process has a public/private key pair. Public keys are known to everyone, but private keys are only known to their owner. (Byzantine processes may exchange their private keys.) Each process also knows the mapping between process indexes and associated public keys, and each process can produce a unique, valid signature for a given message, and check if a signature is valid.



(a) Reconstructed Bracha MBRB (Alg. 2) (b) Reconstructed Imbs-Raynal MBRB (Alg. 3)

Figure 3: Values of ℓ_{MBRB} for the reconstructed BRB algorithms when varying t and d ($n = 100$ and $c = n - t$) within the ranges that satisfy B87-Assumption and IR16-Assumption

It is a simple algorithm that ensures that an app-message must be $k2\ell$-cast by at least k correct processes to be $k2\ell$-delivered by at least ℓ correct processes. For the sake of simplicity, we say that a correct process p_i "ur-broadcasts a set of signatures" if it ur-broadcasts a BUNDLE$(m, id, sigs_i)$ in which $sigs_i$ contains the signatures at hand. A correct process p_i ur-broadcasts an app-message m with identity id at line 5 or line 11.

- If this occurs at line 5, p_i includes in the message it ur-broadcasts all the signatures it has already received for (m, id) plus its own signature.
- If this occurs at line 11, p_i has just received a message containing a set of signatures $sigs$ for the pair (m, id). The process p_i then aggregates in $sigs_i$ the valid signatures it just received with the ones it did know about beforehand (line 10).

This algorithm simply assumes: (the prefix "sb" stands for signature-based)

- sb-$k2\ell$-Assumption 1: $c > 2d$,
- sb-$k2\ell$-Assumption 2: $c - d \geq q_d \geq t + 1$.

Thanks to digital signatures, processes can relay the messages of other processes in Alg. 4. The algorithm, however, does not use forwarding in the same way Alg. 1 did: there is no equivalent of q_f here, that is, the only way to "endorse" an app-message (which, in this case, is equivalent to signing this app-message) is to invoke the $k2\ell$_cast operation. Furthermore, only one app-message can be endorsed by a correct process for a given identity (which is the equivalent of $single = \texttt{true}$ in the signature-free version).

Although this implementation of $k2\ell$-cast provides better guarantees than Alg. 1, using it to reconstruct signature-free BRB algorithms would be counter-productive. This is because signatures allow for MA-tolerant BRB algorithms that are more efficient in terms of round and message complexity than those that can be constructed using $k2\ell$-cast [4].

However, a signature-based $k2\ell$-cast does make sense in contexts in which many-to-many communication patterns are required [8], and, we believe, opens the path to novel ways to handle local state resynchronization resilient to Byzantine failures and message adversaries. For instance, we are using the following algorithm in our own work to design churn-tolerant money transfer systems tolerating Byzantine failures and temporary disconnections.

```
object SigBasedK2LCast(q_d) is
(1)  operation k2ℓ_cast(m, id) is
(2)    if ((−, id) not already signed by p_i) then
(3)      sig_i ← signature of (m, id) by p_i;
(4)      sigs_i ← {all valid signatures for (m, id) ur-broadcast by p_i} ∪ {sig_i};
(5)      ur_broadcast(BUNDLE(m, id, sigs_i));
(6)      check_delivery()
(7)    end if.
(8)  when BUNDLE(m, id, sigs) is received do
(9)    if (sigs contains valid signatures for (m, id) not already ur-broadcast by p_i) then
(10)     sigs_i ← {all valid signatures for (m, id) ur-broadcast by p_i}
           ∪ {all valid signatures for (m, id) in sigs};
(11)     ur_broadcast(BUNDLE(m, id, sigs_i));
(12)     check_delivery()
(13)   end if.
(14) internal operation check_delivery() is
(15)   if (p_i ur-broadcast at least q_d valid signatures for (m, id)
         ∧ (−, id) not already k2ℓ-delivered)
(16)     then k2ℓ_deliver(m, id)
(17)   end if.
end object.
```

Algorithm 4: $k2\ell$-cast implementation with signatures (code for p_i)

6.2 Guarantees

The proof of the following theorem can be found in Appendix C.

Theorem 2 ($k2\ell$-CORRECTNESS)**.** *If sb-$k2\ell$-Assumption* 1 *and* 2 *are verified, Alg.* 4 *implements $k2\ell$-cast with the following guarantees: (i) $k' = q_d - n + c$, (ii) $k = q_d$, (iii) $\ell = c - d$, and (iv) $\delta = q_d > \frac{n+t}{2}$.*

7 Conclusion

This paper discussed reliable broadcast in asynchronous systems where an adversary can control some Byzantine processes and can suppress messages. Its starting point was the design of generic reliable broadcast abstractions suited to applications that do not require total order on the delivery of application messages (distributed money transfers are such applications [7, 9, 18]). However, the ability to thwart an adversary controlling Byzantine processes and a message adversary is new. This approach can be applied to the design of a wide range of quorum-based distributed algorithms other than reliable broadcast. For instance, we conjecture that $k2\ell$-cast could benefit self-stabilizing and self-healing distributed systems [5], where a critical mass of messages from other processes is needed in order to re-synchronize the local state of a given process.

Acknowledgments

This work has been partially supported by the French ANR projects ByBloS (ANR-20-CE25-0002-01) and PriCLeSS (ANR-10-LABX-07-81) devoted to the design of modular distributed computing building blocks.

References

[1] I. Abraham, K. Nayak, L. Ren, and Z. Xiang. Good-case latency of Byzantine broadcast: a complete categorization. In *Proc. 40th ACM Symposium on Principles of Distributed Computing (PODC'21)*, pages 331–341. ACM Press, 2021.

[2] I. Abraham, L. Ren, and Z. Xiang. Good-case and bad-case latency of unauthenticated Byzantine broadcast: A complete categorization. In *Proc. 25th Int'l Conference on Principles of Distributed Systems (OPODIS'21)*, pages 5:1–5:20. LIPIcs, 2021.

[3] Y. Afek and E. Gafni. Asynchrony from synchrony. In *Proc. 14th Int'l Conference on Distributed Computing and Networking (ICDCN'13)*, Springer, pages 225–239, 2021.

[4] T. Albouy, D. Frey, M. Raynal, and F. Taïani. Asynchronous Byzantine reliable broadcast with a message adversary, 2022.

[5] K. Altisen, S. Devismes, S. Dubois, and F. Petit. *Introduction to distributed self-stabilizing algorithms*. Morgan & Claypool, 2019.

[6] H. Attiya and J. Welch. *Distributed computing: fundamentals, simulations and advanced topics*. Wiley-Interscience, 2004.

[7] A. Auvolat, D. Frey, M. Raynal, and F. Taïani. Money transfer made simple: a specification, a generic algorithm, and its proof. *Bulletin of EATCS (European Association of Theoretical Computer Science)*, 132:22–43, 2020.

[8] A. Auvolat, M. Raynal, and F. Taïani. Byzantine-tolerant set-constrained delivery broadcast. In *Proc. 23rd Int'l Conference on Principles of Distributed Systems (OPODIS'19)*, pages 6:1–6:23. LIPIcs, 2019.

[9] M. Baudet, G. Danezis, and A. Sonnino. Fastpay: high-performance Byzantine fault tolerant settlement. In *Proc. 2nd ACM Conference on Advances in Financial Technologies (AFT'20)*, page 163–177. ACM Press, 2020.

[10] G. Bracha. Asynchronous Byzantine agreement protocols. *Information & Computation*, 75(2):130–143, 1987.

[11] C. Cachin, R. Guerraoui, and L. Rodrigues. *Reliable and secure distributed programming*. Springer, 2011.

[12] B. Charron-Bost and A. Schiper. The heard-of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, 2009.

[13] X. Chen, H. Song, J. Jiang, C. Ruan, C. Li, S. Wang, G. Zhang, R. Cheng, and H. Cui. Achieving low tail-latency and high scalability for serializable transactions in edge computing. In *Proc. 16th European Conference on Computer Systems (EuroSys'21)*, pages 210–227. ACM Press, 2021.

[14] D. Didona and W. Zwaenepoel. Size-aware sharding for improving tail latencies in in-memory key-value stores. In *Proc. 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI'19)*, pages 79–94. USENIX Association, 2019.

[15] D. Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3:14–20, 1982.

[16] C. Dwork, D. Peleg, N. Pippenger, and E. Upfal. Fault tolerance in networks of bounded degree. *SIAM Journal of Computing*, 17(5):975–988, 1988.

[17] R. Guerraoui, J. Komatovic, P. Kuznetsov, Y.A. Pignolet, D.A. Seredinschi, and A. Tonkikh. Dynamic Byzantine reliable broadcast. In *Proc. 24th Int'l Conference on Principles of Distributed Systems (OPODIS'20)*, pages 23:1–23:18. LIPIcs, 2020.

[18] R. Guerraoui, P. Kuznetsov, M. Monti, M. Pavlovic, and D.A. Seredinschi. The consensus number of a cryptocurrency. In *Proc. 38th ACM Symposium on Principles of Distributed Computing (PODC'19)*, page 307–316. ACM Press, 2019.

[19] D. Imbs and M. Raynal. Trading t-resilience for efficiency in asynchronous Byzantine reliable broadcast. *Parallel Processing Letters*, 26(4):1650017:1–1650017:8, 2016.

[20] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[21] D. Malkhi and M.K. Reiter. Byzantine quorum systems. *Distributed Computing*, 11(4):203–213, 1998.

[22] A. Maurer, X. Défago, and S. Tixeuil. Communicating reliably in multi-hop dynamic networks despite Byzantine failures. In *Proc. 34th Symposium on Reliable Distributed Systems (SRDS'15)*, pages 238–245. IEEE Press, 2015.

[23] A. Mostéfaoui, H. Moumen, and M. Raynal. Signature-free asynchronous byzantine consensus with $t < n/3$ and $O(n^2)$ messages. In *Proc. 33th ACM Symposium on Principles of Distributed Computing (PODC'14)*, pages 2–9. ACM Press, 2014.

[24] K. Nayak, L. Ren, E. Shi, N.H. Vaidya, and Z. Xiang. Improved extension protocols for Byzantine broadcast and agreement. In *Proc. 34rd Int'l Symposium on Distributed Computing (DISC'20)*, pages 28:1–28:17. LIPIcs, 2020.

[25] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27:228–234, 1980.

[26] M. Raynal. *Distributed algorithms for message-passing systems*. Springer, 2013.

[27] M. Raynal. Message adversaries. In *Encyclopedia of Algorithms*, pages 1272–1276. Springer, 2016.

[28] M. Raynal. *Fault-tolerant message-passing distributed systems: an algorithmic approach*. Springer, 2018.

[29] M. Raynal and J. Stainer. Synchrony weakened by message adversaries vs asynchrony restricted by failure detectors. In *Proc. 32nd ACM Symposium on Principles of Distributed Computing (PODC'13)*, pages 166–175. ACM Press, 2013.

[30] N. Santoro and P. Widmayer. Time is not a healer. In *Proc. 6th Annual Symposium on Theoretical Aspects of Computer Science (STACS'89)*, pages 304–316. Springer, 1989.

[31] N. Santoro and P. Widmayer. Agreement in synchronous networks with ubiquitous faults. *Theoretical Computer Science*, 384(2-3):232–249, 2007.

[32] L. Tseng, Q. Zhang, S. Kumar, and Y. Zhang. Exact consensus under global asymmetric Byzantine links. In *Proc. 40th IEEE Int'l Conference on Distributed Computing Systems (ICDCS 2020)*, pages 721–731. IEEE Press, 2020.

[33] L. Yang, S.J. Park, M. Alizadeh, S. Kannan, and D. Tse. DispersedLedger: high-throughput Byzantine consensus on variable bandwidth networks. In *Prof. 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI'22)*, pages 493–512. USENIX Association, 2022.

A Proof of the Signature-Free $k2\ell$-cast Implementation (Algorithm 1)

A.1 Safety Proof

Lemma 10. *If a correct process p_i $k2\ell$-delivers (m, id), then at least $(q_f - n + c)$ correct processes have ur-broadcast* ENDORSE(m,id) *at line* 3.

Proof. If p_i $k2\ell$-delivers (m, id) at line 10, then it received q_d copies of ENDORSE(m, id) (because of the predicate at line 9). The effective number of Byzantine processes in the system is $n - c$, such that $0 \leq n - c \leq t$. Therefore, p_i must have received at least $q_d - n + c$ (which is strictly positive because $q_d \geq q_f > t \geq n - c$ by sf-$k2\ell$-Assumption 1) messages ENDORSE(m, id) that correct processes ur-broadcast, either during a $k2\ell$_cast(m, id) invocation at line 3, or during a forwarding step at line 7. There are two cases.

- If no correct process has forwarded ENDORSE(m, id) at line 7, then at least $q_d - n + c \geq q_f - n + c$ (as $q_d \geq q_f$ by sf-$k2\ell$-Assumption 1) correct processes have ur-broadcast ENDORSE(m, id) at line 3.
- If at least one correct process forwarded ENDORSE(m, id), then let us consider p_j, the first correct process that forwards ENDORSE(m, id). Because of the predicate at line 6, p_j must have received at least q_f distinct copies of the ENDORSE(m, id) message, out of which at most $n - c$ have been ur-broadcast by Byzantine processes, and at least $q_f - n + c$ (which is strictly positive because $q_f > t \geq n - c$ by sf-$k2\ell$-Assumption 1) have been sent by correct processes. Moreover, as p_j is the first correct process that forwards ENDORSE(m, id), all of the $q_f - n + c$ ENDORSE messages it receives from correct processes must have been sent at line 3. □

Lemma 11 ($k2\ell$-VALIDITY)**.** *If a correct process p_i $k2\ell$-delivers an app-message m with identity id, then at least $k' = q_f - n + c$ correct processes have $k2\ell$-cast m with id.*

Proof. The condition at line 2 implies that the correct processes that ur-broadcast ENDORSE(m, id) at line 3 constitute a subset of those that $k2\ell$-cast (m, id). Thus, by Lemma 10, their number is at least $k' = q_f - n + c$. □

Lemma 12 ($k2\ell$-NO-DUPLICATION)**.** *A correct process p_i $k2\ell$-delivers an app-message m with identity id at most once.*

Proof. This property derives trivially from the predicate at line 9. □

Lemma 13 ($k2\ell$-CONDITIONAL-NO-DUPLICITY)**.** *If the Boolean $\delta = \big((q_f > \frac{n+t}{2}) \vee (single \wedge q_d > \frac{n+t}{2})\big)$ is* `true`*, then no two different correct processes $k2\ell$-deliver different app-messages with the same identity id.*

Proof. Let p_i and p_j be two correct processes that respectively $k2\ell$-deliver (m, id) and (m', id). We want to prove that, if the predicate $\big((q_f > \frac{n+t}{2}) \vee (single \wedge q_d > \frac{n+t}{2})\big)$ is satisfied, then $m = m'$. There are two cases.

- Case $\big(q_f > \frac{n+t}{2}\big)$.

  We denote by A and B the sets of correct processes that have respectively ur-broadcast ENDORSE(m, id) and ENDORSE(m', id) at line 3. By Lemma 10, we know that $|A| \geq q_f - n + c > \frac{n+t}{2} - n + c$ and $|B| \geq q_f - n + c > \frac{n+t}{2} - n + c$. As A and B contain only correct processes, we have $|A \cap B| > 2(\frac{n+t}{2} - n + c) - c = t - n + c \geq t - t = 0$. Hence, at least one correct process p_x has ur-broadcast both ENDORSE(m, id) and ENDORSE(m', id) at line 3. But because of the predicate at line 2, p_x ur-broadcasts at most one message ENDORSE$(-, id)$ at line 3. We conclude that m is necessarily equal to m'.

- Case $\left(single \wedge q_d > \frac{n+t}{2}\right)$.

  Thanks to the predicate at line 9, we can assert that p_i and p_j must have respectively received at least q_d distinct copies of ENDORSE(m, id) and ENDORSE(m', id), from two sets of processes, that we respectively denote A and B, such that $|A| \geq q_d > \frac{n+t}{2}$ and $|B| \geq q_d > \frac{n+t}{2}$. We have $|A \cap B| > 2\frac{n+t}{2} - n = t$. Hence, at least one correct process p_x has ur-broadcast both ENDORSE(m, id) and ENDORSE(m', id). But because of the predicates at lines 2 and 6, and as $single = \texttt{true}$, p_x ur-broadcasts at most one message ENDORSE$(-, id)$, either during a $k2\ell$_cast(m, id) invocation at line 3 or during a forwarding step at line 7. We conclude that m is necessarily equal to m'. □

A.2 Liveness Proof

Lemma 1. $\ell_e \times (k_U + k_F - q_d + 1) \geq (k_U + k_F)(c - d - q_d + q_f) - c(q_f - 1) - k_{NB}(q_d - q_f)$.

Proof. Combining (1), (2), (3) and (4) yields:

$$\begin{aligned}(k_U + k_F)\ell_e + (q_d - 1)(k_{NF} + k_{NB} + k_F - \ell_e) + \\ (q_f - 1)(c - k_{NF} - k_{NB} - k_F) \geq (k_U + k_F)(c - d),\\ \ell_e \times (k_U + k_F - q_d + 1) \geq (k_U + k_F)(c - d) - (q_d - 1)(k_{NF} + k_{NB} + k_F) - \\ (q_f - 1)(c - k_{NF} - k_{NB} - k_F),\\ \geq (k_U + k_F)(c - d) - (q_d - q_f)(k_{NF} + k_{NB} + k_F) - c(q_f - 1).\end{aligned}$$

Using sf-$k2\ell$-Assumption 1, we have $q_d - q_f \geq 0$. By definition, we also have $k_{NF} \leq k_U$, which yields:

$$\begin{aligned}\ell_e \times (k_U + k_F - q_d + 1) &\geq (k_U + k_F)(c - d) - (q_d - q_f)(k_U + k_F + k_{NB}) - c(q_f - 1),\\ &\geq (k_U + k_F)(c - d - q_d + q_f) - c(q_f - 1) - k_{NB}(q_d - q_f).\end{aligned}$$ □

Lemma 2. *If no correct process $k2\ell$-casts (m', id) with $m' \neq m$, then no correct process forwards* ENDORSE(m', id) *at line 7 (and then $k_{NB} = 0$).*

Proof. Assume there is a correct process that ur-broadcasts ENDORSE(m', id) at line 7 with $m' \neq m$. Let us consider the first such process p_i. To execute line 7, p_i must first receive q_f messages ENDORSE(m', id) from distinct processes. Since $q_f > t$ (sf-$k2\ell$-Assumption 1), at least one of these processes, p_j, is correct. Since p_i is the first correct process to forward ENDORSE(m', id) at line 7, the ENDORSE(m', id) message of p_j must come from line 3, and p_j must have $k2\ell$-cast (m', id). We have assumed that no correct process $k2\ell$-cast $m' \neq m$, therefore $m' = m$. Contradiction.

We conclude that, under these assumptions, no correct process ur-broadcasts ENDORSE(m', id) with $m' \neq m$, be it at line 3 (by assumption) or at line 7 (shown by this proof). As a result, $k_{NB} = 0$. □

Lemma 3 ($k2\ell$-LOCAL-DELIVERY)**.** *If at least $k = \left\lfloor \frac{c(q_f-1)}{c-d-q_d+q_f} \right\rfloor + 1$ correct processes $k2\ell$-cast an app-message m with identity id and no correct process $k2\ell$-casts any app-message m' with identity id such that $m \neq m'$, then at least one correct process p_i $k2\ell$-delivers m with identity id.*

Proof. Let us assume that no correct process $k2\ell$-casts (m', id) with $m' \neq m$. No correct process therefore ur-broadcasts ENDORSE(m', id) with $m' \neq m$ at line 3. Lemma 2 also applies and no correct process forwards ENDORSE(m', id) with $m' \neq m$ at line 7 either, so $k_{NB} = 0$. Because no correct process ur-broadcasts ENDORSE(m', id) with $m' \neq m$ whether at line 3 or 7, a correct process receives at most t messages ENDORSE(m', id) (all coming from Byzantine processes). As by sf-$k2\ell$-Assumption 1, $t < q_d$, no correct process $k2\ell$-delivers (m', id) with $m' \neq m$ at line 10.

We now prove the contraposition of the Lemma. Let us assume no correct process $k2\ell$-delivers (m, id). Since, by our earlier observations, no correct process $k2\ell$-delivers (m', id) with $m' \neq m$ either, the condition at line 9 implies that no correct process ever receives at least q_d ENDORSE(m, id), and therefore $\ell_e = 0$. By Lemma 1 we have $c(q_f-1) \geq (k_U+k_F)(c-d-q_d+q_f)$. sf-$k2\ell$-Assumption 1 implies that $c-d-q_d \geq 0 \iff c-d-q_d+q_f > 0$ (as $q_f \geq t+1 \geq 1$), leading to $k_U+k_F \leq \frac{c(q_f-1)}{c-d-q_d+q_f}$. Because of the condition at line 2, a correct process p_j that has $k2\ell$-cast (m, id) but has not ur-broadcast ENDORSE(m, id) at line 3 has necessarily ur-broadcast ENDORSE(m, id) at line 7. We therefore have $k_I \leq k_U + k_F$, which gives $k_I \leq \frac{c(q_f-1)}{c-d-q_d+q_f}$. By contraposition, if $k_I > \frac{c(q_f-1)}{c-d-q_d+q_f}$, then at least one correct process must $k2\ell$-deliver (m, id). Hence, we have $k = \left\lfloor \frac{c(q_f-1)}{c-d-q_d+q_f} \right\rfloor + 1$. □

Lemma 4. $(single = \texttt{false}) \implies (k_{NB} = 0)$.

Proof. Let us consider a correct process $p_i \in A \cup B$. If we assume $p_i \notin F$, p_i never executes line 7 by definition. Because $p_i \in A \cup B$, p_i has received at least q_f messages ENDORSE(m, id), and therefore did not fulfill the condition at line 6 when it received its $q_f{}^{\text{th}}$ message ENDORSE(m, id). As $single = \texttt{false}$ by Lemma assumption, to falsify this condition, p_i must have had already ur-broadcast ENDORSE(m, id) when this happened. Because p_i never executes line 7, this implies that p_i ur-broadcasts ENDORSE(m, id) at line 3, and therefore $p_i \in NF$. This reasoning proves that $A \cup B \setminus F \subseteq NF$. As the sets F, NF and NB partition $A \cup B$, this shows that $NB = \emptyset$, and $k_{NB} = |\emptyset| = 0$. □

Lemma 5. *If at least one correct process $k2\ell$-delivers (m, id) and $x = k_U + k_F$ (the number of correct processes that ur-broadcast* ENDORSE(m, id) *at line* 3 *or* 7*), then $x \geq q_d - t$ and $x^2 - x(c - d + q_f - 1 - k_{NB}) \geq -(c - k_{NB})(q_f - 1)$.*

Proof. Let us write w_A^b the total number of ENDORSE(m, id) messages from Byzantine processes received by the processes of A, and $w_A = w_A^c + w_A^b$ the total of number ENDORSE(m, id) messages received by the processes of A, whether these ENDORSE messages originated from correct or Byzantine senders. By definition, $w_A^b \leq t\ell_e$ and $w_A \geq q_d\ell_e$. By combining these two inequalities with (1) on w_A^c we obtain:

$$\begin{aligned} q_d\ell_e \leq w_A = w_A^c + w_A^b &\leq (k_U + k_F)\ell_e + t\ell_e = (t + k_U + k_F)\ell_e, \\ q_d &\leq t + k_U + k_F, && \text{(as } \ell_e > 0\text{)} \\ q_d - t &\leq k_U + k_F = x. && (5) \end{aligned}$$

This proves the first inequality of the lemma. The processes in $A \cup B$ each receive at most $k_U + k_F$ distinct ENDORSE(m, id) messages from correct processes, so we have $w_A^c + w_B^c \leq (k_{NF} + k_F + k_{NB})(k_U + k_F)$. Combined with the inequalities (3) on w_C^c and (4) on $w_A^c + w_B^c + w_C^c$ that remain valid in this case, we now have:

$$\begin{aligned} &(k_{NF} + k_F + k_{NB})(k_U + k_F) + (q_f - 1)(c - k_{NF} - k_{NB} - k_F) \geq (k_U + k_F)(c - d), \\ &(k_{NF} + k_F + k_{NB})(k_U + k_F - q_f + 1) \geq (k_U + k_F)(c - d) - c(q_f - 1). && (6) \end{aligned}$$

Let us determine the sign of $(k_U + k_F - q_f + 1)$. We derive from (5):

$$\begin{aligned} k_U + k_F - q_f + 1 &\geq q_d - t - q_f + 1 \\ &\geq 1 > 0. && \text{(as } q_d - q_f \geq t \text{ by sf-}k2\ell\text{-Assumption 1)} \end{aligned}$$

As $(k_U + k_F - q_f + 1)$ is positive and we have $k_U \geq k_{NF}$ by definition, we can transform (6) into:

$$\begin{aligned} (k_U + k_F + k_{NB})(k_U + k_F - q_f + 1) &\geq (k_U + k_F)(c - d) - c(q_f - 1), \\ (x + k_{NB})(x - q_f + 1) &\geq x(c - d) - c(q_f - 1), && \text{(as } x = k_U + k_F\text{)} \\ x^2 - x(c - d + q_f - 1 - k_{NB}) &\geq -(c - k_{NB})(q_f - 1). && \square \end{aligned}$$

Lemma 6. *If $k_{NB} = 0$, and at least one correct process $k2\ell$-delivers (m, id), then $k_U + k_F \geq q_d$.*

Proof. By Lemma 5 we have:

$$x^2 - x(c - d + q_f - 1 - k_{NB}) \geq -(c - k_{NB})(q_f - 1), \tag{7}$$

As (7) holds for all, values of $c \in [n - t, n]$, we can in particular consider $c = n - t$. Moreover, as by hypothesis, $k_{NB} = 0$, we have.

$$\begin{aligned} &x^2 - x(n - t - d + q_f - 1) + (q_f - 1)(n - t) \geq 0, \\ &x^2 - \alpha x + (q_f - 1)(n - t) \geq 0. \qquad \text{(by definition of } \alpha) \end{aligned} \tag{8}$$

Let us first observe that the discriminant of the second-degree polynomial in (8) is non negative, i.e. $\alpha^2 - 4(q_f - 1)(n - t) \geq 0$ by sf-$k2\ell$-Assumption 2. This allows us to compute the two real-valued roots as follows:

$$r_0 = \frac{\alpha}{2} - \frac{\sqrt{\alpha^2 - 4(q_f - 1)(n - t)}}{2} \quad \text{and} \quad r_1 = \frac{\alpha}{2} + \frac{\sqrt{\alpha^2 - 4(q_f - 1)(n - t)}}{2}.$$

Thus (8) is satisfied if and only if $x \leq r_0 \vee x \geq r_1$.

- Let us prove $r_0 \leq q_d - 1 - t$. We need to show that:

$$\begin{aligned} \frac{\alpha}{2} - \frac{\sqrt{\alpha^2 - 4(q_f - 1)(n - t)}}{2} &\leq q_d - 1 - t \\ \frac{\alpha}{2} - (q_d - 1) + t &\leq \frac{\sqrt{\alpha^2 - 4(q_f - 1)(n - t)}}{2} \\ \frac{\sqrt{\alpha^2 - 4(q_f - 1)(n - t)}}{2} &\geq \frac{\alpha}{2} - (q_d - 1) + t \\ \sqrt{\alpha^2 - 4(q_f - 1)(n - t)} &\geq \alpha - 2(q_d - 1) + 2t. \end{aligned}$$

  The inequality is trivially satisfied if $\alpha - 2(q_d - 1) + 2t < 0$. For all other cases, we need to verify that:

$$\begin{aligned} \alpha^2 - 4(q_f - 1)(n - t) &\geq (\alpha - 2(q_d - 1) + 2t)^2, \\ \alpha^2 - 4(q_f - 1)(n - t) &\geq \alpha^2 + 4(q_d - 1)^2 + 4t^2 - 4\alpha(q_d - 1) + 4\alpha t - 8t(q_d - 1), \\ -4(q_f - 1)(n - t) &\geq 4(q_d - 1)^2 + 4t^2 - 4\alpha(q_d - 1) + 4\alpha t - 8t(q_d - 1), \\ -(q_f - 1)(n - t) &\geq (q_d - 1)^2 + t^2 - \alpha(q_d - 1) + \alpha t - 2t(q_d - 1), \\ -(q_f - 1)(n - t) &\geq (q_d - 1 - t)^2 - \alpha(q_d - 1 - t), \end{aligned}$$

  and thus $\alpha(q_d - 1 - t) - (q_f - 1)(n - t) - (q_d - 1 - t)^2 \geq 0$, which is true by sf-$k2\ell$-Assumption 4.

- Let us prove $r_1 > q_d - 1$. We want to show that:

$$\frac{\alpha}{2} + \frac{\sqrt{\alpha^2 - 4(q_f - 1)(n - t)}}{2} > q_d - 1$$

  Let us rewrite the inequality as follows:

$$\begin{aligned} \alpha + \sqrt{\alpha^2 - 4(q_f - 1)(n - t)} &> 2(q_d - 1) \\ \sqrt{\alpha^2 - 4(q_f - 1)(n - t)} &> 2(q_d - 1) - \alpha \end{aligned}$$

The inequality is trivially satisfied if $2(q_d - 1) - \alpha < 0$. For all other cases, we can take the squares as follows:

$$\begin{aligned}
\alpha^2 - 4(q_f - 1)(n - t) &> (2(q_d - 1) - \alpha)^2, \\
\alpha^2 - 4(q_f - 1)(n - t) &> 4(q_d - 1)^2 + \alpha^2 - 4\alpha(q_d - 1), \\
-4(q_f - 1)(n - t) &> 4(q_d - 1)^2 - 4\alpha(q_d - 1), \\
4\alpha(q_d - 1) - 4(q_f - 1)(n - t) - 4(q_d - 1)^2 &> 0, \\
\alpha(q_d - 1) - (q_f - 1)(n - t) - (q_d - 1)^2 &> 0,
\end{aligned}$$

which is true by sf-$k2\ell$-Assumption 3.

We now know that $r_0 \leq q_d - 1 - t$ and that $r_1 > q_d - 1$. In addition, as $x \leq r_0 \vee x \geq r_1$, we have $x \leq q_d - t - 1 \vee x > q_d - 1$. But Lemma 5 states that $x \geq q_d - t$, which is incompatible with $x \leq q_d - t - 1$. So we are left with $x > q_d - 1$, which implies, as q_d and x are integers that $x \geq q_d$, thus proving the lemma for $c = n - t$.

Let us now consider the set E_0 of all executions in which t processes are Byzantine, and therefore $c = n - t$, and a set E_c of executions in which there are fewer Byzantine processes, and thus $c > n - t$ correct processes. We show that $E_c \subseteq E_0$ in that a Byzantine process can always simulate the behavior of a correct process. In particular, if the simulated correct process is not subject to the message adversary, the simulating Byzantine process simply operates like a correct process. If, on the other hand, the simulated correct process misses some messages as a result of the message adversary, the Byzantine process can also simulate missing such messages. As a result, the executions that can happen when $c > n - t$ can also happen when $c = n - t$. Thus our result proven for $c = n - t$ can be extended to all possible values of c. □

Lemma 7. *If $k_{NB} = 0$ and $k_U + k_F \geq q_d$, then at least $\left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes $k2\ell$-deliver some app-message with identity id (not necessarily m).*

Proof. As $k_{NB} = 0$ and $k_U + k_F \geq q_d$, we can rewrite the inequality of Lemma 1 into:

$$\ell_e \times (k_U + k_F - q_d + 1) \geq (k_U + k_F)(c - d - q_d + q_f) - c(q_f - 1).$$

From $k_U + k_F \geq q_d$ we derive $k_U + k_F - q_d + 1 > 0$, and we transform the above inequality into:

$$\ell_e \geq \frac{(k_U + k_F)(c - d - q_d + q_f) - c(q_f - 1)}{k_U + k_F - q_d + 1}.$$

Let us now focus on the case in which $c = n - t$, we obtain:

$$\ell_e \geq \frac{(k_U + k_F)(n - t - d - q_d + q_f) - (n - t)(q_f - 1)}{k_U + k_F - q_d + 1}.$$

The right side of the inequality is of the form:

$$\ell_e \geq \frac{\phi x - \beta}{x - \gamma} = \phi + \frac{\phi\gamma - \beta}{x - \gamma} \tag{9}$$

with:

$$\begin{aligned}
x &= k_U + k_F, \\
\gamma &= q_d - 1, \\
\alpha &= n - t - d + q_f - 1, \\
\phi &= n - t - d - q_d + q_f, \\
\beta &= c(q_f - 1).
\end{aligned}$$

Since, by hypothesis, $x = k_U + k_F \geq q_d$, we have:

$$x - \gamma = k_U + k_F - q_d + 1 > 0. \tag{10}$$

We also have:

$$\begin{aligned}
\phi\gamma - \beta &= (\alpha - \gamma)\gamma - c(q_f - 1) = \alpha\gamma - \gamma^2 - c(q_f - 1), \\
&= \alpha(q_d - 1) - (q_d - 1)^2 - (n - t)(q_f - 1) > 0, && \text{(by sf-}k2\ell\text{-Assumption 3)} \\
\phi\gamma - \beta &> 0. && (11)
\end{aligned}$$

Injecting (10) and (11) into (9), we conclude that $\phi + \frac{\phi\gamma - \beta}{x - \gamma}$ is a *decreasing hyperbole* defined over $x \in]\gamma, \infty]$ with *asymptotic value* ϕ when $x \to \infty$. As x is a number of correct processes, $x \leq c$. The decreasing nature of the right-hand side of (9) leads us to: $\ell_e \geq \phi + \frac{\phi\gamma - \beta}{c - \gamma} = \frac{\phi c - \beta}{c - \gamma} \geq \frac{c(c - d - q_d + q_f) - c(q_f - 1)}{c - q_d + 1} \geq c \times \frac{c - d - q_d + 1}{c - q_d + 1} = c\left(1 - \frac{d}{c - q_d + 1}\right)$.

Since ℓ_e is a positive integer, we conclude that at least $\ell_{\mathsf{min}} = \left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes receive at least q_d message ENDORSE(m, id) at line 9. As each of these processes either $k2\ell$-delivers (m, id) when this first happens, or has already $k2\ell$-delivered another app-message $m' \neq m$ with identity id, we conclude that at least ℓ_{min} correct processes $k2\ell$-deliver some app-message (whether it be m or $m' \neq m$) with identity id when $c = n - t$. The reasoning for extending this result to any value of $c \in [n - t, n]$ is identical to the one at the end of the proof of Lemma 6 just above. □

Lemma 8 ($k2\ell$-WEAK-GLOBAL-DELIVERY)**.** *If $single =$ `false`, and a correct process $k2\ell$-delivers an app-message m with identity id, then at least $\ell = \left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes $k2\ell$-deliver an app-message m' with identity id (each possibly different from m).*

Proof. Let us assume $single =$ `false`, and one correct process $k2\ell$-delivers (m, id). By Lemma 4, $k_{NB} = 0$. The prerequisites for Lemma 6 are verified, and therefore $k_U + k_F \geq q_d$. This provides the prerequisites for Lemma 7, from which we conclude that at least $\ell = \left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes $k2\ell$-deliver an app-message m' with identity id, which concludes the proof of the lemma. □

Lemma 9 ($k2\ell$-STRONG-GLOBAL-DELIVERY)**.** *If $single =$ `true`, and a correct process $k2\ell$-delivers an app-message m with identity id, and no correct process $k2\ell$-casts an app-message $m' \neq m$ with identity id, then at least $\ell = \left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes $k2\ell$-deliver m with identity id.*

Proof. Let us assume that *(i)* $single =$ `true`, *(ii)* no correct process $k2\ell$-casts (m', id) with $m' \neq m$, and *(iii)* one correct process $k2\ell$-delivers (m, id). Lemma 2 holds and implies that $k_{NB} = 0$. From there, as above, Lemmas 6 and 7 hold, and at least $\ell = \left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes $k2\ell$-deliver an app-message for identity id.

By hypothesis, no correct process ur-broadcasts ENDORSE(m', id) at line 3 with $m' \neq m$. Similarly, because of Lemma 2, no correct process ur-broadcasts ENDORSE(m', id) at line 7 with $m' \neq m$. As a result, a correct process can receive at most receive t messages ENDORSE(m', id) at line 9 (all from Byzantine processes). As $q_d > t$ (by sf-$k2\ell$-Assumption 1), the condition of line 9 never becomes true for $m' \neq m$, and as result no correct process delivers an app-message $m' \neq m$ with identity id. All processes that $k2\ell$-deliver an app-message with identity id, therefore, $k2\ell$-deliver m, which concludes the lemma. □

B Proof of the Signature-Free MBRB Implementations

The proofs that follow use integer arithmetic. Given a real number x and an integer i, let us recall that $x-1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x+1$, $\lfloor x+i \rfloor = \lfloor x \rfloor + i$, $\lceil x+i \rceil = \lceil x \rceil + i$, $\lfloor -x \rfloor = -\lceil x \rceil$, $(i > x) \iff (i \geq \lfloor x \rfloor + 1)$, $(i < x) \iff (i \leq \lceil x \rceil - 1)$.

B.1 Proof of MBRB with Bracha's reconstructed algorithm (Algorithm 2)

B.1.1 Instantiating the parameters of the $k2\ell$-cast objects

In Alg. 2 (page 12), we instantiate the $k2\ell$-cast objects obj_{E} and obj_{R} using the signature-free implementation presented in Section 3.2. Let us mention that, given that $obj_{\text{E}}.single = obj_{\text{R}}.single = \texttt{true}$, then we use the strong variant of the global-delivery property of $k2\ell$-cast ($k2\ell$-STRONG-GLOBAL-DELIVERY) for both objects obj_{E} and obj_{R}. Moreover, according to the definitions of k', k, ℓ and δ (page 5) and their values stated in Theorem 1, we have:

- $obj_{\text{E}}.k' = obj_{\text{E}}.q_f - n + c = t + 1 - n + c \geq t + 1 - t = 1$,
- $obj_{\text{E}}.k = \left\lfloor \dfrac{c(obj_{\text{E}}.q_f - 1)}{c - d - obj_{\text{E}}.q_d + obj_{\text{E}}.q_f} \right\rfloor + 1 = \left\lfloor \dfrac{c(t+1-1)}{c - d - \lfloor \frac{n+t}{2} \rfloor - 1 + t + 1} \right\rfloor + 1$
  $= \left\lfloor \dfrac{ct}{c - d - \lfloor \frac{n-t}{2} \rfloor} \right\rfloor + 1$,
- $obj_{\text{E}}.\ell = \left\lceil c\left(1 - \dfrac{d}{c - obj_{\text{E}}.q_d + 1}\right) \right\rceil = \left\lceil c\left(1 - \dfrac{d}{c - \lfloor \frac{n+t}{2} \rfloor - 1 + 1}\right) \right\rceil$
  $= \left\lceil c\left(1 - \dfrac{d}{c - \lfloor \frac{n+t}{2} \rfloor}\right) \right\rceil$,
- $obj_{\text{E}}.\delta = \left(\left(obj_{\text{E}}.q_f > \dfrac{n+t}{2}\right) \vee \left(obj_{\text{E}}.single \wedge obj_{\text{E}}.q_d > \dfrac{n+t}{2}\right)\right)$
  $= \left(\left(t+1 > \dfrac{n+t}{2}\right) \vee \left(\texttt{true} \wedge \left\lfloor \dfrac{n+t}{2} \right\rfloor + 1 > \dfrac{n+t}{2}\right)\right)$
  $= (\texttt{false} \vee (\texttt{true} \wedge \texttt{true})) = \texttt{true}$,
- $obj_{\text{R}}.k' = obj_{\text{R}}.q_f - n + c = t + 1 - n + c \geq t + 1 - t = 1$,
- $obj_{\text{R}}.k = \left\lfloor \dfrac{c(obj_{\text{R}}.q_f - 1)}{c - d - obj_{\text{R}}.q_d + obj_{\text{R}}.q_f} \right\rfloor + 1 = \left\lfloor \dfrac{c(t+1-1)}{c - d - 2t - d - 1 + t + 1} \right\rfloor + 1$
  $= \left\lfloor \dfrac{ct}{c - 2d - t} \right\rfloor + 1$,
- $obj_{\text{R}}.\ell = \left\lceil c\left(1 - \dfrac{d}{c - obj_{\text{R}}.q_d + 1}\right) \right\rceil = \left\lceil c\left(1 - \dfrac{d}{c - 2t - d - 1 + 1}\right) \right\rceil$
  $= \left\lceil c\left(1 - \dfrac{d}{c - 2t - d}\right) \right\rceil$,
- $obj_{\text{R}}.\delta = \left(\left(obj_{\text{R}}.q_f > \dfrac{n+t}{2}\right) \vee \left(obj_{\text{R}}.single \wedge obj_{\text{R}}.q_d > \dfrac{n+t}{2}\right)\right)$
  $= \left(\left(t+1 > \dfrac{n+t}{2}\right) \vee \left(\texttt{true} \wedge 2t + d + 1 > \dfrac{n+t}{2}\right)\right) \in \{\texttt{true}, \texttt{false}\}$

We recall that parameter δ controls the conditional no-duplicity property. The value for $obj_{\text{E}}.\delta$ is `true`, but that of value for $obj_{\text{R}}.\delta$ may be either `true` or `false` depending on the values of n, t, and d. This is fine because, in Bracha's reconstructed algorithm (Alg. 2), it is the first round (obj_{E}) that ensures no-duplicity. Once this has happened, the second round (obj_{R}) does not need to provide no-duplicity but only needs to guarantee the termination properties of local and global delivery. This observation allows obj_{R} to operate with lower values of q_d and q_f.

Finally, we observe that for Alg. 2, sf-$k2\ell$-Assumption 1 through 4 are all satisfied by B87-Assumption $n > 3t + 2d + 2\sqrt{td}$. We prove this fact in Appendix 14. In the following, we prove that $3t + 2d + 2\sqrt{td} \geq 2t + d + \sqrt{t^2 + 6td + d^2} \geq 3t + 2d$.

Observation 1. *For $d, t \in \mathbb{N}_0$ non-negative integers, we have:*

$$3t + 2d + 2\sqrt{td} \geq 2t + d + \sqrt{t^2 + 6td + d^2} \geq 3t + 2d.$$

Proof. Let us start by proving the first inequality.

$$\begin{aligned}
t^2 + 6td + d^2 + 4\sqrt{td}(t + d) &\geq t^2 + 6td + d^2,\\
t^2 + d^2 + 4td + 4t\sqrt{td} + 4d\sqrt{td} + 2td &\geq t^2 + 6td + d^2,\\
(t + d + 2\sqrt{td})^2 &\geq t^2 + 6td + d^2,\\
t + d + 2\sqrt{td} &\geq \sqrt{t^2 + 6td + d^2},\\
3t + 2d + 2\sqrt{td} &\geq 2t + d + \sqrt{t^2 + 6td + d^2}.
\end{aligned}$$

Let us then prove the second inequality:

$$\begin{aligned}
t^2 + 6td + d^2 &\geq t^2 + 2td + d^2 = (t + d)^2,\\
\sqrt{t^2 + 6td + d^2} &\geq t + d,\\
2t + d + \sqrt{t^2 + 6td + d^2} &\geq 3t + 2d.
\end{aligned}$$

□

B.1.2 Proof of satisfaction of the assumptions of Algorithm 1

In this section, we prove that all the assumptions of the signature-free $k2\ell$-cast implementation presented in Alg. 1 (page 7) are well respected for the two $k2\ell$-cast instances used in Alg. 2 (obj_{E} and obj_{R}).

Lemma 14. *Alg. 1's assumptions are well respected for obj_{E}.*

Proof. Let us recall that $q_f = t + 1$ and $q_d = \lfloor \frac{n+t}{2} \rfloor + 1$ for obj_{E}.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 1 ($c - d \geq obj_{\text{E}}.q_d \geq obj_{\text{E}}.q_f + t \geq 2t + 1$):
  By B87-Assumption and Observation 1, we have the following:

$$\begin{aligned}
c - d \geq n - t - d &= \frac{2n - 2t - 2d}{2}, && \text{(by definition of } c)\\
&> \frac{n + 3t + 2d - 2t - 2d}{2} = \frac{n + t}{2}, && \text{(as } n > 3t + 2d)\\
&\geq \left\lfloor \frac{n + t}{2} \right\rfloor + 1. && (12)
\end{aligned}$$

  We also have:

$$\begin{aligned}
\left\lfloor \frac{n + t}{2} \right\rfloor + 1 &\geq \left\lfloor \frac{3t + 2d + 1 + t}{2} \right\rfloor + 1, && \text{(as } n > 3t + 2d)\\
&\geq \lfloor 2t + d + 1/2 \rfloor + 1 = 2t + d + 1 \geq 2t + 1. && (13)
\end{aligned}$$

By combining (12) and (13), we get:

$$c - d \geq \left\lfloor \frac{n+t}{2} \right\rfloor + 1 \geq 2t + 1 \geq 2t + 1,$$
$$c - d \geq obj_{\mathrm{E}}.q_d \geq obj_{\mathrm{E}}.q_f + t \geq 2t + 1. \qquad \text{(sf-}k2\ell\text{-Assumption 1)}$$

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 2 ($\alpha^2 - 4(obj_{\mathrm{E}}.q_f - 1)(n-t) \geq 0$):

  Let us recall that, for object obj_{E}, we have $q_f = t + 1$ and $q_d = \lfloor \frac{n+t}{2} \rfloor + 1$. We therefore have $\alpha = n + q_f - t - d - 1 = n - d$. Let us now consider the quantity:

$$\begin{aligned}\Delta = \alpha^2 - 4(q_f - 1)(n-t) &= (n-d)^2 - 4t(n-t) \\ &= 4t^2 + d^2 + n^2 + n\,(-4t - 2d)\end{aligned}$$

  The inequality is satisfied if $n > 2\sqrt{td} + 2t + d$, which is clearly the case as $n > 3t + 2d + 2\sqrt{td}$. This proves sf-$k2\ell$-Assumption 2.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 3 ($\alpha(obj_{\mathrm{E}}.q_d - 1) - (obj_{\mathrm{E}}.q_f - 1)(n-t) - (obj_{\mathrm{E}}.q_d - 1)^2 > 0$):

  Let us consider the quantity on the left-hand side of sf-$k2\ell$-Assumption 3 and substitute $q_f = t+1$, $q_d = \lfloor \frac{n+t}{2} \rfloor + 1$:

$$\begin{aligned}&\alpha(q_d - 1) - (q_f - 1)(n-t) - (q_d - 1)^2, \\ &= (n + q_f - t - d - 1)(q_d - 1) - (q_f - 1)(n-t) - (q_d - 1)^2, \\ &= (n-d)\left(\left\lfloor \frac{n+t}{2} \right\rfloor\right) - t(n-t) - \left(\left\lfloor \frac{n+t}{2} \right\rfloor\right)^2. \end{aligned} \qquad (14)$$

  We now observe that $\left(\lfloor \frac{n+t}{2} \rfloor\right) = \left(\frac{n+t-\epsilon}{2}\right)$ with $\epsilon = 0$ if $n + t = 2k$ is even, and $\epsilon = 1$ if $n + t = 2k + 1$ is odd. We thus rewrite (14) as follows:

$$\begin{aligned}&(n-d)\left(\frac{n+t-\epsilon}{2}\right) - t(n-t) - \left(\frac{n+t-\epsilon}{2}\right)^2, \\ &= \frac{n+t-\epsilon}{2} \times \frac{2n - 2d - n - t + \epsilon}{2} - t(n-t), \\ &= \frac{(n+t-\epsilon)(n-2d-t+\epsilon) - 4t(n-t)}{4}, \\ &= \frac{n^2 - t^2 - 2td + 2t\epsilon - 2nd + 2d\epsilon - \epsilon^2 - 4nt + 4t^2}{4}, \\ &= \frac{n^2 + 3t^2 - 2td - 2n(d+2t) + \epsilon(2t + 2d - \epsilon)}{4}.\end{aligned}$$

  As we want to show that the above quantity is positive, the result will not change if we multiply it by 4:

$$n^2 + 3t^2 - 2td - 2n(d+2t) + \epsilon(2t + 2d - \epsilon) > 0. \qquad (15)$$

  We now solve the inequality to obtain:

$$n > 2t + d + \sqrt{t^2 + 6td + d^2 - \epsilon(2t + 2d - \epsilon)}.$$

We observe that, for $t+d \geq 1$, the quantity $-\epsilon(2t+2d-\epsilon)$ is strictly negative if $\epsilon = 1$, therefore if $\epsilon = 1 \vee t+d \geq 1$:

$$\begin{aligned} n &> 3t+2d+2\sqrt{td}, \\ &\geq t+d+\sqrt{t^2+6td+d^2}, && \text{(by Observation 1)} \\ &\geq 2t+d+\sqrt{t^2+6td+d^2-\epsilon(2t+2d-\epsilon)}. \end{aligned}$$

This leaves out the case $(t=d=0) \wedge (n = 2k+1 \text{ is odd})$, for which we can show that (15) is positive or null for $n \geq 1$:

$$\begin{aligned} (15): \quad & n^2+3t^2-2td-2n(d+2t)+\epsilon(2t+2d-\epsilon), \\ & = n^2-1 \geq 0 \text{ for } n \geq 1. \end{aligned}$$

This completes the proof of sf-$k2\ell$-Assumption 3.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 4 ($\alpha(obj_{\mathrm{E}}.q_d-1-t)-(obj_{\mathrm{E}}.q_f-1)(n-t)-(obj_{\mathrm{E}}.q_d-1-t)^2 \geq 0$):

  Let us consider the quantity on the left-hand side of sf-$k2\ell$-Assumption 4 and substitute $q_f = tb+1$, $q_d = \lfloor \frac{n+t}{2} \rfloor + 1$:

$$\begin{aligned} & \alpha(q_d-1-t)-(q_f-1)(n-t)-(q_d-1-t)^2, \\ &= (n+q_f-t-d-1)(q_d-1-t)-(q_f-1)(n-t)-(q_d-1-t)^2, \\ &= (n-d)\left(\left\lfloor \frac{n+t}{2} \right\rfloor - t\right) - t(n-t) - \left(\left\lfloor \frac{n+t}{2} \right\rfloor - t\right)^2. \end{aligned} \tag{16}$$

  Like before, we observe that $\left(\lfloor \frac{n+t}{2} \rfloor\right) = \left(\frac{n+t-\epsilon}{2}\right)$ with $\epsilon = 0$ if $n+t = 2k$ is even, and $\epsilon = 1$ if $n+t = 2k+1$ is odd. We thus rewrite (16) as follows:

$$\begin{aligned} & (n-d)\left(\frac{n+t-\epsilon}{2} - t\right) - t(n-t) - \left(\frac{n+t-\epsilon}{2} - t\right)^2, \\ &= (n-d)\cdot\frac{n-t-\epsilon}{2} - t(n-t) - \left(\frac{n-t-\epsilon}{2}\right)^2, \\ &= \frac{n-t-\epsilon}{2}\cdot\frac{2n-2d-n+t+\epsilon}{2} - t(n-t), \\ &= \frac{(n-t-\epsilon)(n-2d+t+\epsilon)-4nt+4t^2}{4}, \\ &= \frac{-t^2+2td-2t\epsilon+2d\epsilon-2dn-\epsilon^2+n^2}{4}. \end{aligned}$$

  As we want to show that the above quantity is non-negative, the result will not change if we multiply it by 4:

$$-t^2+2td-2t\epsilon+2d\epsilon-\epsilon^2-2dn+n^2.$$

  We then solve the inequality to obtain: $n \geq \mathsf{max}(t+\epsilon, -t+2d-\epsilon)$, which is clearly satisfied as $n \geq 3t+2d+2\sqrt{td}+1$. This proves all previous inequality and thus sf-$k2\ell$-Assumption 4. □

Lemma 15. *Alg. 1's assumptions are well respected for obj_{R}.*

Proof. Let us recall that $q_f = t + 1$ and $q_d = 2t + d + 1$ for obj_{R}. Let us observe that we have then $q_d - q_f - t - d = 0$.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 1 ($c - d \geq obj_{\mathrm{R}}.q_d \geq obj_{\mathrm{R}}.q_f + t \geq 2t + 1$):

  From Observation 1, we have:

  $$\begin{aligned}
  &c - d \geq n - t - d \geq 3t + 2d + 1 - t - d \geq 2t + d + 1, && \text{(as } n > 3t + 2d\text{)}\\
  &c - d \geq 2t + d + 1 \geq 2t + 1 \geq 2t + 1,\\
  &c - d \geq obj_{\mathrm{R}}.q_d \geq obj_{\mathrm{R}}.q_f + t \geq 2t + 1. && \text{(sf-}k2\ell\text{-Assumption 1)}
  \end{aligned}$$

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 2 ($\alpha^2 - 4(obj_{\mathrm{R}}.q_f - 1)(n - t) \geq 0$):

  Let us recall that, for object obj_{R}, we have $q_f = t+1$ and $q_d = 2t+d+1$. As sf-$k2\ell$-Assumption 2 depends on q_d but not on q_f, and since $obj_{\mathrm{E}}.q_f = obj_{\mathrm{R}}.q_f$, we refer the reader to the proof we gave in Lemma 14 for obj_{E}.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 3 ($\alpha(obj_{\mathrm{R}}.q_d - 1) - (obj_{\mathrm{R}}.q_f - 1)(n - t) - (obj_{\mathrm{R}}.q_d - 1)^2 > 0$):

  Let us consider the quantity on the left-hand side of sf-$k2\ell$-Assumption 3:

  $$\begin{aligned}
  &\alpha(q_d - 1) - (q_f - 1)(n - t) - (q_d - 1)^2,\\
  &= (n + q_f - t - d - 1)(q_d - 1) - (q_f - 1)(n - t) - (q_d - 1)^2,\\
  &= (n - d)(2t + d) - t(n - t) - (2t + d)^2,\\
  &= 2nt + nd - 2td - d^2 - nt + t^2 - 4t^2 - d^2 - 4td,\\
  &= n(t + d) - 6td - 2d^2 - 3t^2,\\
  &= n(t + d) - (6td + 2d^2 + 3t^2). && (17)
  \end{aligned}$$

  Then, we observe that we can lower bound the quantity on the left side of (17) by substituting B87-Assumption, i.e. $n > 3t + 2d + 2\sqrt{td} \geq 2t + d + \sqrt{t^2 + 6td + d^2}$. For convenience, in the following we write $\rho = t^2 + 6td + d^2$, thus $n > 2t + d + \sqrt{\rho}$. We get:

  $$\begin{aligned}
  &n(t + d) - (3t^2 + 6td + 2d^2),\\
  &> (2t + d + \sqrt{\rho})(t + d) - (3t^2 + 6td + 2d^2),\\
  &= \sqrt{\rho}(t + d) - d^2 - t^2 - 3td.
  \end{aligned}$$

  We now want to show that the above quantity is positive or null, i.e.:

  $$\sqrt{\rho}(t + d) - d^2 - t^2 - 3td \geq 0. \qquad (18)$$

  We now rewrite (18) as follows:

  $$\begin{aligned}
  \sqrt{\rho}(t + d) &\geq d^2 + t^2 + 2td + td,\\
  \sqrt{\rho}(t + d) &\geq (d + t)^2 + td,\\
  (t^2 + 6td + d^2)(t + d)^2 &\geq ((d + t)^2 + td)^2, && \text{(as } (d + t)^2 + td \geq 0\text{)}\\
  ((t + d)^2 + 4td)(t + d)^2 &\geq ((d + t)^2 + td)^2,\\
  (t + d)^4 + 4td(t + d)^2 &\geq (d + t)^4 + (td)^2 + 2td(t + d)^2,\\
  2td(t + d)^2 &\geq (td)^2,\\
  2td(t^2 + d^2 + 2td) &\geq (td)^2,\\
  2td(t^2 + d^2) + 4(td)^2 &\geq (td)^2,\\
  2td(t^2 + d^2) + 3(td)^2 &\geq 0.
  \end{aligned}$$

This proves (18) and all previous inequalities and ultimately sf-$k2\ell$-Assumption 3.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 4 ($\alpha(obj_{\text{R}}.q_d-1-t)-(obj_{\text{R}}.q_f-1)(n-t)-(obj_{\text{R}}.q_d-1-t)^2 \geq 0$):

  Let us consider the quantity on the left-hand side of sf-$k2\ell$-Assumption 4:

$$\begin{aligned}
&\alpha(q_d-1-t)-(q_f-1)(n-t)-(q_d-1-t)^2, && (19)\\
&=(n+q_f-t-d-1)(q_d-1-t)-(q_f-1)(n-t)-(q_d-1-t)^2,\\
&=(n+-d)(t+d)-t(n-t)-(t+d)^2,\\
&=(t+d)(n+-2d-t)-t(n-t),\\
&=nt+nd-2td-2d^2-t^2-td-nt+t^2,\\
&=nd-3td-2d^2,\\
&=nd-3td-2d^2,\\
&=d(n-3t-2d). && (20)
\end{aligned}$$

  Like before, we observe that we can lower bound the quantity on the left side of (20) by substituting B87-Assumption, i.e., $n > 3t+2d+2\sqrt{td} \geq 3t+2d$, so we have:

$$\begin{aligned}
(20):\quad & d(n-3t-2d)\\
&> d(3t+2d-2d-3t)=0. && (21)
\end{aligned}$$

  which recursively proves that (19) is positive or zero and thus sf-$k2\ell$-Assumption 4. □

B.1.3 Correctness proof

This section proves the following theorem:

Theorem 3 (MBRB-CORRECTNESS)**.** *If B87-Assumption is verified, then Alg.* 2 *implements* MBRB *with the guarantee* $\ell_{MBRB} = \left\lceil c\left(1-\frac{d}{c-2t-d}\right)\right\rceil$.

The proof follows from the next lemmas.

Lemma 16. $c-d \geq obj_{\text{E}}.k$.

Proof. We want to show that:

$$c-d \geq \left\lfloor \frac{ct}{c-d-\lfloor\frac{n-t}{2}\rfloor}\right\rfloor + 1 = obj_{\text{E}}.k. \tag{22}$$

As the left-hand side is also an integer, we can rewrite (22) as follows:

$$\begin{aligned}
c-d &> \frac{ct}{c-d-\lfloor\frac{n-t}{2}\rfloor}, && (23)\\
(c-d)(c-d-\lfloor\frac{n-t}{2}\rfloor) &> ct. && (\text{as } (c-d-\lfloor\tfrac{n-t}{2}\rfloor)>0)
\end{aligned}$$

We now observe that $\left(\lfloor\frac{n+t}{2}\rfloor\right) = \left(\frac{n+t-\epsilon}{2}\right)$ with $\epsilon = 0$ if $n+t = 2k$ is even, and $\epsilon = 1$ if $n+t = 2k+1$ is odd, which leads us to:

$$\begin{aligned}
(c-d)(c-d-\frac{n-t-\epsilon}{2}) &> ct,\\
(c-d)(2c-2d-n+t+\epsilon) &> 2ct,\\
(c-d)(2c-2d-n+t+\epsilon)-2ct &> 0.
\end{aligned}$$

Like for the proofs of Lemma 6 and Lemma 7, we leverage the fact that the executions that can happen when $c > n - t$ can also occur when $c = n - t$. We thus rewrite our inequality for $c = n - t$:

$$\begin{aligned}
(n-t-d)(n-t-2d+\epsilon) - 2(n-t)t > 0,\\
(n-t)(n-t-2d+\epsilon-2t) - d(n-t-2d+\epsilon) > 0,\\
(n-t)^2 + (n-t)(-2d+\epsilon-2t) - d(n-t-2d+\epsilon) > 0,\\
n^2 + t^2 - 2nt - 2nd + n\epsilon - 2nt + 2td - t\epsilon + 2t^2 - nd + td + 2d^2 - \epsilon d > 0,\\
n^2 + 3t^2 - 4nt - 3nd + n\epsilon + 3td - t\epsilon + 2d^2 - \epsilon d > 0,\\
n^2 - n(4t + 3d - \epsilon) + 3t^2 + 3td + 2d^2 - \epsilon(t+d) > 0.
\end{aligned}$$

We now solve the second-degree inequality with respect to n. It is easy to see that the discriminant is non-negative for non-negative values of t and d. So we obtain:

$$\begin{aligned}
n > 2t + \frac{3d}{2} - \frac{\epsilon}{2} + \frac{\sqrt{4t^2 + 12td - 4t\epsilon + d^2 - 2d\epsilon + \epsilon^2}}{2},\\
-4t - 3d + \epsilon + 2n - \sqrt{4t^2 + 12td - 4t\epsilon + d^2 - 2d\epsilon + \epsilon^2} > 0,
\end{aligned}$$

which is implied by the following as $n \geq 3t + 2d + 2\sqrt{td} + 1$:

$$\begin{aligned}
4\sqrt{td} + 2t + d + 2 + \epsilon - \sqrt{4t^2 + 12td - 4t\epsilon + d^2 - 2d\epsilon + \epsilon^2} > 0,\\
4\sqrt{td} + 2t + d + 2 + \epsilon > \sqrt{4t^2 + 12td - 4t\epsilon + d^2 - 2d\epsilon + \epsilon^2}.
\end{aligned}$$

Taking the squares as both the argument of the square root and the left-hand side are non-negative leads to:

$$\begin{aligned}
\left(4\sqrt{td} + 2t + d + \epsilon + 2\right)^2 > 4t^2 + 12td - 4t\epsilon + d^2 - 2d\epsilon + \epsilon^2,\\
16t^{\frac{3}{2}}\sqrt{d} + 8\sqrt{t}d^{\frac{3}{2}} + 8\sqrt{t}\sqrt{d}\epsilon + 16\sqrt{t}\sqrt{d} + 4t^2 + 20td + 4t\epsilon + 8t + d^2 + 2d\epsilon\\
+ 4d + \epsilon^2 + 4\epsilon + 4 > 4t^2 + 12td - 4t\epsilon + d^2 - 2d\epsilon + \epsilon^2,
\end{aligned}$$

which simplifies to:

$$\begin{aligned}
16t^{\frac{3}{2}}\sqrt{d} + 8\sqrt{t}d^{\frac{3}{2}} + 8\sqrt{t}\sqrt{d}\epsilon + 16\sqrt{t}\sqrt{d} + 8td + 8t\epsilon + 8t + 4d\epsilon\\
+ 4d + 4\epsilon + 4 > 0. \qquad (24)
\end{aligned}$$

We can then easily observe that the left-hand side of (24) is strictly positive, thereby proving all previous inequalities and thus the lemma. □

Lemma 17. $obj_{\text{E}}.\ell \geq obj_{\text{R}}.k$.

Proof. We need to prove:

$$obj_{\text{E}}.\ell = \left\lceil c\left(1 - \frac{d}{c - \lfloor \frac{n+t}{2} \rfloor}\right)\right\rceil \geq \left\lfloor \frac{ct}{c-2d-t} \right\rfloor + 1 = obj_{\text{R}}.k. \qquad (25)$$

We observe that $x \geq \lfloor m \rfloor + 1$ if and only if $x > m$, and that $m \geq \lfloor m \rfloor$. Therefore (25) is implied by the following:

$$\begin{aligned}
c\left(1 - \frac{d}{c - \frac{n+t-\epsilon}{2}}\right) \geq \frac{ct}{c-t-2d},\\
c - \frac{2dc}{2c - t - n + \epsilon} > \frac{ct}{c-t-2d}.
\end{aligned}$$

As both denominators are positive, we can solve:

$$
\begin{aligned}
&-t\left(-t+2c+\epsilon-n\right)-2d\left(-t-2d+c\right)+\left(-t-2d+c\right)\left(-t+2c+\epsilon-n\right)>0,\\
&-t\left(-t+2c+\epsilon-n\right)+\left(-t-2d+c\right)\left(-t-2d+2c+\epsilon-n\right)>0,\\
&-t\left(-2t+c+\epsilon\right)+\left(-t-2d+c\right)\left(-2t-2d+c+\epsilon\right)>0, && \text{(as } c\geq n-t\text{)}\\
&-t\left(-2t+c+\epsilon\right)+\left(-t-2d+c\right)\left(-2t-2d+c+\epsilon\right)>0,\\
&-t\left(-t+2c-n\right)+\epsilon\left(-3t-2d+c\right)+\left(-t-2d+c\right)^2>0,\\
&t^2-2tc+tn+\epsilon\left(-3t-2d+c\right)+\left(-t-2d+c\right)^2>0,\\
&t^2-2tc+t\left(2\sqrt{t}\sqrt{d}+3t+2d+1\right)+\epsilon\left(-3t-2d+c\right)+\left(-t-2d+c\right)^2>0,\\
& && \text{(as } n\geq 3t+2d+2\sqrt{td}\text{)}\\
&2t^{\frac{3}{2}}\sqrt{d}+4t^2+2td-2tc+t+\epsilon\left(-3t-2d+c\right)+\left(-t-2d+c\right)^2>0,\\
&2t^{\frac{3}{2}}\sqrt{d}+5t^2+6td-4tc+t+4d^2-4dc+c^2+\epsilon\left(-3t-2d+c\right)>0.
\end{aligned}
$$

We now consider the two possible values of ϵ:

- $\epsilon=0$:

$$2t^{\frac{3}{2}}\sqrt{d}+5t^2+6td-4tc+t+4d^2-4dc+c^2>0 \tag{26}$$

We solve the inequality with respect to c to obtain (when the discriminant is positive):

$$c>2t+2d+\sqrt{-2t^{\frac{3}{2}}\sqrt{d}-t^2+2td-t}$$

which we prove by observing that $c\geq n-t\geq 2t+2d+2\sqrt{td}+1$ and that:

$$2t+2d+2\sqrt{td}+1>2t+2d+\sqrt{-2t^{\frac{3}{2}}\sqrt{d}-t^2+2td-t},$$

as all terms except $2td$ inside the square root are negative. When the discriminant is negative (e.g. for $d=0$), inequality (26) is satisfied for all values of c.

- $\epsilon=1$:

In this case, we obtain:

$$2t^{\frac{3}{2}}\sqrt{d}+5t^2+6td-4tc-2t+4d^2-4dc-2d+c^2+c>0,$$

which is implied by a negative discriminant or by:

$$c>2t+2d+\sqrt{-2t^{\frac{3}{2}}\sqrt{d}-t^2+2td+1/4}-\frac{1}{2}.$$

Like before, we simply observe that:

$$
\begin{aligned}
2\sqrt{td}&\geq\sqrt{2td}+\frac{1}{2}-\frac{1}{2},\\
&\geq\sqrt{2td+1/4}-\frac{1}{2},\\
&\geq\sqrt{-2t^{\frac{3}{2}}\sqrt{d}-t^2+2td+1/4}-\frac{1}{2},
\end{aligned}
$$

thereby proving the second case and the lemma. □

Lemma 18 (MBRB-VALIDITY)**.** *If a correct process p_i mbrb-delivers an app-message m from a correct process p_j with sequence number sn, then p_j mbrb-broadcast m with sequence number sn.*

Proof. If p_i mbrb-delivers (m, sn, j) at line 1, then it $k2\ell$-delivered $(\text{READY}(m), (sn, j))$ using obj_{R}. From $k2\ell$-VALIDITY, and as $obj_{\text{R}}.k' = 1$, we can assert that at least one correct process p_x $k2\ell$-cast $(\text{READY}(m), (sn, j))$ at line 3, after having $k2\ell$-delivered $(\text{ECHO}(m), (sn, j))$ using obj_{E}. Again, from $k2\ell$-VALIDITY, we can assert that at least $obj_{\text{E}}.k' = 1$ correct process p_y $k2\ell$-cast $(\text{ECHO}(m), (sn, j))$ at line 2, after having received an $\text{INIT}(m, sn)$ message from p_j. And as p_j is correct and the network channels are authenticated, then p_j has ur-broadcast $\text{INIT}(m, sn)$ at line 1, during a mbrb_broadcast(m, sn) invocation. □

Lemma 19 (MBRB-NO-DUPLICATION)**.** *A correct process p_i mbrb-delivers at most one app-message from a process p_j with sequence number sn.*

Proof. By $k2\ell$-NO-DUPLICATION, we know that a correct process p_i can $k2\ell$-deliver at most one $\text{READY}(-)$ with identity (sn, j). Therefore, p_i can mbrb-deliver only one app-message from p_j with sequence number sn. □

Lemma 20 (MBRB-NO-DUPLICITY)**.** *No two different correct processes mbrb-deliver different app-messages from a process p_i with the same sequence number sn.*

Proof. We proceed by contradiction. Let us consider two correct processes p_w and p_x that respectively mbrb-deliver (m, sn, i) and (m', sn, i) at line 4, such that $m \neq m'$. It follows that p_w and p_x respectively $k2\ell$-delivered $(\text{READY}(m), (sn, i))$ and $(\text{READY}(m'), (sn, i))$ using obj_{R}.

From $k2\ell$-VALIDITY, and as $obj_{\text{R}}.k' \geq 1$, we can assert that two correct processes p_y and p_z respectively $k2\ell$-cast $(\text{READY}(m), (sn, i))$ and $(\text{READY}(m'), (sn, i))$ at line 3, after having respectively $k2\ell$-delivered $(\text{ECHO}(m), (sn, i))$ and $(\text{ECHO}(m'), (sn, i))$ using obj_{E}. But as $obj_{\text{E}}.\delta = \texttt{true}$, then, by $k2\ell$-CONDITIONAL-NO-DUPLICITY, we know that $m = m'$. There is a contradiction. □

Lemma 21 (MBRB-LOCAL-DELIVERY)**.** *If a correct process p_i mbrb-broadcasts an app-message m with sequence number sn, then at least one correct process p_j eventually mbrb-delivers m from p_i with sequence number sn.*

Proof. If p_i mbrb-broadcasts (m, sn) at line 1, then it invokes ur-broadcasts $\text{INIT}(m, sn)$. By the definition of the MA, the message $\text{INIT}(m, sn)$ is then received by at least $c - d$ correct processes at line 2, which then $k2\ell$-cast $(\text{ECHO}(m), sn, i)$. As p_i is correct and ur-broadcasts only one message $\text{INIT}(-, sn)$, then no correct process $k2\ell$-casts any different $(\text{ECHO}(-), sn, i)$. Moreover, thanks to Lemma 16, we know that:

$$c - d \geq obj_{\text{E}}.k = \left\lfloor \frac{ct}{c - d - \lfloor \frac{n-t}{2} \rfloor} \right\rfloor + 1.$$

Hence, from $k2\ell$-LOCAL-DELIVERY and $k2\ell$-STRONG-GLOBAL-DELIVERY, at least $obj_{\text{E}}.\ell = \left\lceil c \left(1 - \frac{d}{c - \lfloor \frac{n+t}{2} \rfloor}\right) \right\rceil$ correct processes eventually $k2\ell$-deliver $(\text{ECHO}(m), (sn, i))$ using obj_{E} and then $k2\ell$-cast $(\text{READY}(m), (sn, i))$ using obj_{R} at line 3. By $k2\ell$-VALIDITY, and as $obj_{\text{R}}.k' \geq 1$, then no correct process can $k2\ell$-cast a different $(\text{READY}(-), (sn, i))$, because otherwise it would mean that at least one correct process would have $k2\ell$-cast a different $(\text{ECHO}(-), (sn, i))$, which is impossible (see before). Moreover, thanks to Lemma 17, we know that:

$$\left\lceil c \left(1 - \frac{d}{c - \lfloor \frac{n+t}{2} \rfloor}\right) \right\rceil = obj_{\text{E}}.\ell \geq obj_{\text{R}}.k = \left\lfloor \frac{ct}{c - 2d - t} \right\rfloor + 1.$$

Therefore, $k2\ell$-LOCAL-DELIVERY applies and we know that at least one correct processes eventually $k2\ell$-delivers $(\text{READY}(m), (sn, i))$ using obj_{R} and then mbrb-delivers (m, sn, i) at line 4. □

Lemma 22 (MBRB-GLOBAL-DELIVERY)**.** *If a correct process p_i mbrb-delivers an app-message m from a process p_j with sequence number sn, then at least $\ell_{MBRB} = \left\lceil c\left(1 - \frac{d}{c-2t-d}\right)\right\rceil$ correct processes mbrb-deliver m from p_j with sequence number sn.*

Proof. If p_i mbrb-delivers (m, sn, j) at line 4, then it has $k2\ell$-delivered $(\text{READY}(m), (sn, j))$ using obj_{R}. From $k2\ell$-VALIDITY, we know that at least $obj_{\text{R}}.k' \geq 1$ correct process $k2\ell$-cast $(\text{READY}(m), (sn, j))$ using obj_{R} at line 3 and thus $k2\ell$-delivered $(\text{ECHO}(m), (sn, j))$ using obj_{E}. From $k2\ell$-CONDITIONAL-NO-DUPLICITY, and as $obj_{\text{E}}.\delta = \texttt{true}$, we can state that no correct process $k2\ell$-delivers any $(\text{ECHO}(m'), (sn, j))$ where $m' \neq m$ using obj_{E}, so no correct process $k2\ell$-casts any $(\text{READY}(m'), (sn, j))$ where $m' \neq m$ using obj_{R} at line 3. It means that $k2\ell$-STRONG-GLOBAL-DELIVERY applies, and we can assert that at least $obj_{\text{R}}.\ell = \left\lceil c\left(1 - \frac{d}{c-2t-d}\right)\right\rceil = \ell_{MBRB}$ correct processes eventually $k2\ell$-deliver $(\text{READY}(m), (sn, j))$ using obj_{R} and thus mbrb-deliver (m, sn, j) at line 4. □

B.2 Proof of MBRB with Imbs and Raynal's reconstructed algorithm (Algorithm 3)

B.2.1 Instantiating the parameters of the $k2\ell$-cast object

In Alg. 3 (page 13), we instantiate the $k2\ell$-cast object obj_{W} using the signature-free implementation presented in Section 3.2 with parameters $q_d = \left\lfloor \frac{n+3t}{2} \right\rfloor + 3d + 1$, $q_f = \left\lfloor \frac{n+t}{2} \right\rfloor + 1$, and $single = \texttt{false}$. Based on Theorem 1 (page 8), these parameters lead to the following values for k', k, ℓ and δ.

- $obj_{\text{W}}.k' = obj_{\text{W}}.q_f - n + c = \left\lfloor \frac{n+t}{2} \right\rfloor + 1 - n + c$
  $\geq \left\lfloor \frac{n+t}{2} \right\rfloor + 1 - n + n - t = \left\lfloor \frac{n-t}{2} \right\rfloor + 1,$
- $obj_{\text{W}}.k = \left\lfloor \frac{c(obj_{\text{W}}.q_f - 1)}{c - d - obj_{\text{W}}.q_d + obj_{\text{W}}.q_f} \right\rfloor + 1$
  $= \left\lfloor \frac{c(\lfloor \frac{n+t}{2} \rfloor + 1 - 1)}{c - d - (\lfloor \frac{n+3t}{2} \rfloor + 3d + 1) + \lfloor \frac{n+t}{2} \rfloor + 1} \right\rfloor + 1$
  $= \left\lfloor \frac{c\lfloor \frac{n+t}{2} \rfloor}{c - t - 4d} \right\rfloor + 1,$
- $obj_{\text{W}}.\ell = \left\lceil c\left(1 - \frac{d}{c - obj_{\text{E}}.q_d + 1}\right)\right\rceil = \left\lceil c\left(1 - \frac{d}{c - (\lfloor \frac{n+3t}{2} \rfloor + 3d + 1) + 1}\right)\right\rceil$
  $= \left\lceil c\left(1 - \frac{d}{c - \lfloor \frac{n+3t}{2} \rfloor - 3d}\right)\right\rceil,$
- $obj_{\text{W}}.\delta = \left(\left(obj_{\text{W}}.q_f > \frac{n+t}{2}\right) \vee \left(obj_{\text{W}}.single \wedge obj_{\text{W}}.q_d > \frac{n+t}{2}\right)\right)$
  $= \left(\left(\left\lfloor \frac{n+t}{2} \right\rfloor + 1 > \frac{n+t}{2}\right) \vee \left(\texttt{false} \wedge \left\lfloor \frac{n+3t}{2} \right\rfloor + 3d + 1 > \frac{n+t}{2}\right)\right)$
  $= (\texttt{true} \vee (\texttt{false} \wedge \texttt{true})) = \texttt{true}.$

Finally, we observe that for Alg. 3, sf-$k2\ell$-Assumption 1 through 4 are all satisfied by IR16-Assumption ($n > 5t + 12d + \frac{2td}{t+2d}$), as we prove in Appendix B.2.2.

B.2.2 Proof of satisfaction of the assumptions of Algorithm 1

This section proves that all the assumptions of the signature-free $k2\ell$-cast implementation presented in Alg. 1 (page 7) are well respected for the $k2\ell$-cast instance used in Alg. 3 (obj_{W}).

Lemma 23. *Alg. 1's sf-$k2\ell$-Assumptions are well respected for obj_{W}.*

Proof. Let us recall that $q_f = \left\lfloor \frac{n+t}{2} \right\rfloor + 1$ and $q_d = \left\lfloor \frac{n+3t}{2} \right\rfloor + 3d + 1$ for object obj_{W}.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 1 ($c - d \geq obj_{\mathrm{W}}.q_d \geq obj_{\mathrm{W}}.q_f + t \geq 2t + 1$):
  From IR16-Assumption ($n > 5t + 12d + \frac{2td}{t+2d}$), we get that $n > 5t + 8d$, which yields:

$$\begin{aligned} c - d \geq n - t - d &= \frac{2n - 2t - 2d}{2}, && \text{(by definition of } c\text{)} \\ &> \frac{n + 5t + 8d - 2t - 2d}{2} = \frac{n + 3t}{2}, && \text{(as } n > 5t + 8d\text{)} \\ &\geq \left\lfloor \frac{n + 3t + 6d}{2} \right\rfloor + 1 = \left\lfloor \frac{n + 3t}{2} \right\rfloor + 3d + 1. && (27) \end{aligned}$$

  We also have:

$$\begin{aligned} \left\lfloor \frac{n + 3t}{2} \right\rfloor + 1 &> \left\lfloor \frac{5t + 8d + 3t}{2} \right\rfloor + 1 = 4t + 4d + 1, && \text{(as } n > 5t + 8d\text{)} \\ &\geq 2t + 1. && (28) \end{aligned}$$

  By combining (27) and (28), we obtain:

$$\begin{aligned} &c - d \geq \left\lfloor \frac{n + 3t}{2} \right\rfloor + 3d + 1 \geq \left\lfloor \frac{n + 3t}{2} \right\rfloor + 1 \geq 2t + 1, \\ &c - d \geq obj_{\mathrm{W}}.q_d \geq obj_{\mathrm{W}}.q_f + t \geq 2t + 1. \end{aligned}$$

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 2 ($\alpha^2 - 4(obj_{\mathrm{W}}.q_f - 1)(n - t) \geq 0$):
  Let us recall that for object obj_{W} we have $q_f = \left\lfloor \frac{n+t}{2} \right\rfloor + 1$ and $q_d = \left\lfloor \frac{n+3t}{2} \right\rfloor + 3d + 1$. We therefore have $\alpha = \left\lfloor \frac{3n-t}{2} \right\rfloor - d$. Let us now consider the following quantity:

$$\begin{aligned} \Delta &= \alpha^2 - 4(q_f - 1)(n - t), \\ &= \left(\left\lfloor \frac{3n - t}{2} \right\rfloor - d \right)^2 - 4 \left\lfloor \frac{n + t}{2} \right\rfloor (n - t). && (29) \end{aligned}$$

  We now observe that $\left(\left\lfloor \frac{m}{2} \right\rfloor\right) = \left(\frac{m-\epsilon}{2}\right)$ with $\epsilon = 0$ if $m = 2k$ is even, and $\epsilon = 1$ if $m = 2k + 1$ is

odd. We thus rewrite (29) as follows:

$$\begin{aligned}
&\left(\frac{3n-t-\epsilon}{2}-d\right)^2-4\frac{n+t-\epsilon}{2}(n-t),\\
&=\left(\frac{3n-t-\epsilon-2d}{2}\right)^2-4\frac{n+t-\epsilon}{2}(n-t),\\
&=\frac{t^2+4td+2t\epsilon-6tn+4d^2+4d\epsilon-12dn+\epsilon^2-6\epsilon n+9n^2}{4}\\
&\quad+\frac{8t^2-8t\epsilon+8\epsilon n-8n^2}{4},\\
&=\frac{9t^2+4td-6t\epsilon-6tn+4d^2+4d\epsilon-12dn+\epsilon^2+2\epsilon n+n^2}{4},\\
&=\frac{9t^2-6tn+n^2+4td-12dn+4d^2+4d\epsilon-6t\epsilon+\epsilon^2+2\epsilon n}{4},\\
&=\frac{(n-3t)^2+4d(t-3n+d)+\epsilon(4d-6t+\epsilon+2n)}{4}.
\end{aligned}$$

We now multiply by 4 and solve the inequality:

$$\begin{aligned}
&n^2-6n(t+2d)+9t^2+4td+4d^2+\epsilon\left(-6t+4d+\epsilon+2n\right)\geq 0,\\
&n\geq 3t+4\sqrt{d}\sqrt{2t+2d-\epsilon}+6d-\epsilon.
\end{aligned}\tag{30}$$

By IR16-Assumption we have $n > 5t+12d+\frac{2td}{t+2d}$. To prove (30), we therefore show that $5t+12d+\frac{2td}{t+2d}\geq 3t+4\sqrt{d}\sqrt{2t+2d}+6d$:

$$\begin{aligned}
&5t+12d+\frac{2td}{t+2d}\geq 3t+4\sqrt{d}\sqrt{2t+2d}+6d\\
\iff&2t+6d+\frac{2td}{t+2d}\geq 4\sqrt{d}\sqrt{2t+2d}\\
\iff&\left(2t+6d+\frac{2td}{t+2d}\right)^2\geq 16d(2t+2d)\iff\\
\iff&-16d\left(t+2d\right)\left(2t+2d\right)+\left(2td+2t\left(t+2d\right)+6d\left(t+2d\right)\right)^2\geq 0\\
\iff&4t^4+48t^3d+192t^2d^2-32t^2d+288td^3-96td^2+144d^4-64d^3\geq 0.
\end{aligned}\tag{31}$$

We observe that (31) holds as $144d^4\geq 64d^3$, $288td^3\geq 96td^2$, and $192t^2d^2\geq 32t^2d$, therefore proving sf-$k2\ell$-Assumption 2.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 3 ($\alpha(obj_{\mathrm{w}}.q_d-1)-(obj_{\mathrm{w}}.q_f-1)(n-t)-(obj_{\mathrm{w}}.q_d-1)^2>0$):

  Let us consider the quantity on the left-hand side of sf-$k2\ell$-Assumption 3 and substitute $q_f=\left\lfloor\frac{n+t}{2}\right\rfloor+1$, $q_d=\left\lfloor\frac{n+3t}{2}\right\rfloor+3d+1$, and $\alpha=\left\lfloor\frac{3n-t}{2}\right\rfloor-d$:

$$\begin{aligned}
&\alpha(q_d-1)-(q_f-1)(n-t)-(q_d-1)^2,\\
&=\left(\left\lfloor\frac{3n-t}{2}\right\rfloor-d\right)\left(\left\lfloor\frac{n+3t}{2}\right\rfloor+3d\right)-\left(\left\lfloor\frac{n+t}{2}\right\rfloor\right)(n-t)\\
&\quad-\left(\left\lfloor\frac{n+3t}{2}\right\rfloor+3d\right)^2.
\end{aligned}$$

We now observe that $\left(\left\lfloor \frac{m}{2} \right\rfloor\right) = \left(\frac{m-\epsilon}{2}\right)$ with $\epsilon = 0$ if $m = 2k$ is even, and $\epsilon = 1$ if $m = 2k+1$ is odd, and rewrite the expression accordingly:

$$\begin{aligned}
&\frac{3n - t - 2d - \epsilon}{2} \cdot \frac{n + 3t + 6d - \epsilon}{2} - \frac{(n + t - \epsilon)(n - t)}{2} \\
&- \left(\frac{n + 3t + 6d - \epsilon}{2}\right)^2, \\
&= \frac{(n + 3t + 6d - \epsilon)(3n - t - 2d - \epsilon - n - 3t - 6d + \epsilon)}{4} - \frac{(n + t - \epsilon)(n - t)}{2}, \\
&= \frac{(n + 3t + 6d - \epsilon)(2n - 4t - 8d)}{4} - \frac{(n + t - \epsilon)(n - t)}{2}, \\
&= \frac{-12t^2 - 48td + 4t\epsilon + 2tn - 48d^2 + 8d\epsilon + 4dn - 2\epsilon n + 2n^2 + 2t^2 - 2t\epsilon + 2\epsilon n - 2n^2}{4}, \\
&= \frac{-10t^2 - 48td + 2t\epsilon + 2tn - 48d^2 + 8d\epsilon + 4dn}{4}.
\end{aligned}$$

As the coefficients of n are all positive, we can lower-bound the quantity using $n > 5t + 12d + \frac{2td}{t+2d}$:

$$\begin{aligned}
&\frac{-10t^2 - 48td - 48d^2 + 2n(t + 2d) + 2\epsilon(t + 8d)}{4}, \\
&= \frac{-10t^2 - 48td - 48d^2 + 2(5t + 12d + \frac{2td}{t+2d})(t + 2d) + 2\epsilon(t + 8d)}{4}, \\
&= \frac{-10t^2 - 48td - 48d^2 + 10t^2 + 44td + 48d^2 + 4td + 2\epsilon(t + 8d)}{4}, \\
&= \frac{\epsilon(t + 8d)}{2} \geq 0,
\end{aligned}$$

which proves all previous inequalities and thus sf-$k2\ell$-Assumption 3.

- *Proof of satisfaction of sf-$k2\ell$-Assumption* 4 ($\alpha(obj_{\mathrm{w}}.q_d - 1 - t) - (obj_{\mathrm{w}}.q_f - 1)(n - t) - (obj_{\mathrm{w}}.q_d - 1 - t)^2 \geq 0$):

  Let us consider the quantity on the left-hand side of sf-$k2\ell$-Assumption 4 and substitute $q_f = \left\lfloor \frac{n+t}{2} \right\rfloor + 1$, $q_d = \left\lfloor \frac{n+3t}{2} \right\rfloor + 3d + 1$, and $\alpha = \left\lfloor \frac{3n-t}{2} \right\rfloor - d$:

$$\begin{aligned}
&\alpha(q_d - 1 - t) - (q_f - 1)(n - t) - (q_d - 1 - t)^2, \\
&= \left(\left\lfloor \frac{3n - t}{2} \right\rfloor - d\right)\left(\left\lfloor \frac{n + 3t}{2} \right\rfloor + 3d - t\right) - \left(\left\lfloor \frac{n + t}{2} \right\rfloor\right)(n - t) \\
&- \left(\left\lfloor \frac{n + 3t}{2} \right\rfloor + 3d - t\right)^2.
\end{aligned}$$

  We now observe that $\left(\left\lfloor \frac{m}{2} \right\rfloor\right) = \left(\frac{m-\epsilon}{2}\right)$ with $\epsilon = 0$ if $m = 2k$ is even, and $\epsilon = 1$ if $m = 2k+1$ is

odd, and rewrite the expression accordingly:

$$
\begin{aligned}
&= \left(\frac{3n-t-\epsilon}{2} - d\right)\left(\frac{n+3t-\epsilon}{2} + 3d - t\right) - \left(\frac{n+t-\epsilon}{2}\right)(n-t) \\
&\quad - \left(\frac{n+3t-\epsilon}{2} + 3d - t\right)^2, \\
&= \left(\frac{3n-t-2d-\epsilon}{2}\right)\left(\frac{n+t+6d-\epsilon}{2}\right) - \left(\frac{n+t-\epsilon}{2}\right)(n-t) \\
&\quad - \left(\frac{n+t+6d-\epsilon}{2}\right)^2, \\
&= \frac{(n+t+6d-\epsilon)(3n-t-2d-\epsilon-n-t-6d+\epsilon)}{4} - \left(\frac{(n+t-\epsilon)(n-t)}{2}\right), \\
&= \frac{(n+t+6d-\epsilon)(2n-2t-8d)}{4} - \left(\frac{(n+t-\epsilon)(n-t)}{2}\right), \\
&= \frac{(n+t+6d-\epsilon)(n-t-4d) - (n+t-\epsilon)(n-t)}{2}, \\
&= \frac{-10td - 24d^2 + 4d\epsilon + 2dn}{2}.
\end{aligned}
$$

As the coefficients of n are all positive, we can lower bound using $n > 5t+12d+\frac{2td}{t+2d} > 5t+12d$ to obtain:

$$
\begin{aligned}
&= \frac{-10td - 24d^2 + 4d\epsilon + 2d(5t+12d)}{2}, \\
&= \frac{-10td - 24d^2 + 4d\epsilon + 10td + 24d^2)}{2}, \\
&= 2d\epsilon \geq 0,
\end{aligned}
$$

which proves sf-$k2\ell$-Assumption 4. □

B.2.3 Correctness proof

This section proves the following theorem:

Theorem 4 (MBRB-CORRECTNESS)**.** *If IR16-Assumption is verified, then Alg.* 3 *implements* MBRB *with the guarantee* $\ell_{MBRB} = \left\lceil c\left(1 - \frac{d}{c - \left\lfloor \frac{n+3t}{2} \right\rfloor - 3d}\right)\right\rceil$.

The proof follows from the next lemmas.

Lemma 24. $c - d \geq obj_{\text{W}}.k$.

Proof. This proof is presented in reverse order: we start with the result we want to prove and finish with a proposition we know to be true. In this manner, given two consecutive propositions, we only need that

the latter implies the former and not necessarily the converse. We want to show that:

$$
\begin{aligned}
c - d &\geq \left\lfloor \frac{c\left\lfloor \frac{n+t}{2} \right\rfloor}{c-t-4d} \right\rfloor + 1 = obj_{\mathrm{W}}.k, \\
c - d &> \frac{c\left\lfloor \frac{n+t}{2} \right\rfloor}{c-t-4d}, && (\text{as } x \geq \lfloor y \rfloor + 1 \iff x > y) \\
c - d &> \frac{c\frac{n+t}{2}}{c-t-4d}, \\
c - d &> \frac{c(n+t)}{2(c-t-4d)}, \\
c - d &> \frac{c(n+t)}{2c-2t-8d}, \\
(c-d)(2c-2t-8d) &> c(n+t), && (\text{as } 2c-2t-8d > 0 \text{ by IR16-Assumption}) \\
(c-d)(2c-2t-8d) &> c(c-2t) \geq c(n+t), && (\text{as } n \leq c+t) \\
(c-d)(2c-2t-8d) - c(c-2t) &> 0, \\
c^2 + 2td - 4tc + 8d^2 - 10dc &> 0, \\
2td + 8d^2 + c^2 + c\,(-4t-10d) &> 0.
\end{aligned}
$$

The left-hand side of the above inequality is a second-degree polynomial, whose roots we can solve:

$$\left[2t + 5d - \sqrt{4t^2 + 18td + 17d^2}, 2t + 5d + \sqrt{4t^2 + 18td + 17d^2}\right].$$

We now need to show that:

$$c > 2t + 5d + \sqrt{4t^2 + 18td + 17d^2}.$$

By IR16-Assumption, we know that:

$$n \geq 5t + 12d + \frac{2td}{t+2d} + 1,$$

and thus that:

$$
\begin{aligned}
n &\geq 5t + 12d + 1, \\
c &\geq 4t + 12d + 1.
\end{aligned}
$$

So we want to show that:

$$
\begin{aligned}
4t + 12d + 1 &> 2t + 5d + \sqrt{4t^2 + 18td + 17d^2}, \\
2t + 7d + 1 &> \sqrt{4t^2 + 18td + 17d^2}.
\end{aligned}
$$

It is easy to see that the right-hand side of the above inequality is non-negative, so we get:

$$
\begin{aligned}
(2t + 7d + 1)^2 &> 4t^2 + 18td + 17d^2, \\
4t^2 + 28td + 4t + 49d^2 + 14d + 1 &> 4t^2 + 18td + 17d^2, \\
10td + 4t + 32d^2 + 14d + 1 &> 0.
\end{aligned}
$$

This concludes the proof. □

Lemma 25 (MBRB-VALIDITY)**.** *If a correct process p_i mbrb-delivers an app-message m from a correct process p_j with sequence number sn, then p_j mbrb-broadcast m with sequence number sn.*

Proof. If p_i mbrb-delivers (m, sn, j) at line 1, then it $k2\ell$-delivered $(\text{WITNESS}(m), (sn, j))$ using obj_{W}. From $k2\ell$-VALIDITY, and as $obj_{\text{R}}.k' \geq 1$, we can assert that at least one correct process p'_i $k2\ell$-cast $(\text{WITNESS}(m), (sn, j))$ at line 2, after having received an $\text{INIT}(m, sn)$ message from p_j. And as p_j is correct and the network channels are authenticated, then p_j has ur-broadcast $\text{INIT}(m, sn)$ at line 1, during a mbrb_broadcast(m, sn) invocation. □

Lemma 26 (MBRB-NO-DUPLICATION)**.** *A correct process p_i mbrb-delivers at most one app-message from a process p_j with sequence number sn.*

Proof. By $k2\ell$-NO-DUPLICATION, we know that a correct process p_i can $k2\ell$-deliver at most one $\text{READY}(-)$ with identity (sn, j). Therefore, p_i can mbrb-deliver only one app-message from p_j with sequence number sn. □

Lemma 27 (MBRB-NO-DUPLICITY)**.** *No two distinct correct processes mbrb-deliver different app-messages from a process p_i with the same sequence number sn.*

Proof. As $obj_{\text{W}}.\delta =$ `true`, then, by $k2\ell$-CONDITIONAL-NO-DUPLICITY, we know that no two correct processes can $k2\ell$-deliver two different app-messages with the same identity using obj_{W} at line 3. Hence, no two correct processes mbrb-deliver different app-messages for a given sequence number sn and sender p_i. □

Lemma 28 (MBRB-LOCAL-DELIVERY)**.** *If a correct process p_i mbrb-broadcasts an app-message m with sequence number sn, then at least one correct process p_j eventually mbrb-delivers m from p_i with sequence number sn.*

Proof. If p_i mbrb-broadcasts (m, sn) at line 1, then it invokes ur-broadcasts $\text{INIT}(m, sn)$. By the definition of the MA, the message $\text{INIT}(m, sn)$ is then received by at least $c - d$ correct processes at line 2, which then $k2\ell$-cast $(\text{WITNESS}(m), sn, i)$. But thanks to Lemma 24, we know that:

$$c - d \geq obj_{\text{W}}.k = \left\lfloor \frac{c\left\lfloor \frac{n+t}{2} \right\rfloor}{c - t - 4d} \right\rfloor + 1.$$

As p_i is correct and ur-broadcasts only one message $\text{INIT}(-, sn)$, then no correct process $k2\ell$-casts any different $(\text{WITNESS}(-), sn, i)$, $k2\ell$-LOCAL-DELIVERY applies and at least one correct processes eventually $k2\ell$-delivers $(\text{WITNESS}(m), (sn, i))$ using obj_{W} and thus mbrb-delivers (m, sn, i) at line 3. □

Lemma 29 (MBRB-GLOBAL-DELIVERY)**.** *If a correct process p_i mbrb-delivers an app-message m from a process p_j with sequence number sn, then at least $\ell_{MBRB} = \left\lceil c\left(1 - \frac{d}{c - \left\lfloor \frac{n+3t}{2} \right\rfloor - 3d}\right)\right\rceil$ correct processes mbrb-deliver m from p_j with sequence number sn.*

Proof. If p_i mbrb-delivers (m, sn, j) at line 3, then it has $k2\ell$-delivered $(\text{WITNESS}(m), (sn, j))$ using obj_{W}. As $obj_{\text{W}}.\delta =$ `true`, we can assert from $k2\ell$-WEAK-GLOBAL-DELIVERY and $k2\ell$-CONDITIONAL-NO-DUPLICITY that at least $obj_{\text{W}}.\ell = \left\lceil c\left(1 - \frac{d}{c - q_d + 1}\right)\right\rceil$ correct processes eventually $k2\ell$-deliver $(\text{WITNESS}(m), (sn, j))$ using obj_{W} and thus mbrb-deliver (m, sn, j) at line 3. By substituting the values of q_f and q_d, we obtain $obj_{\text{W}}.\ell = \left\lceil c\left(1 - \frac{d}{c - \left\lfloor \frac{n+3t}{2} \right\rfloor - 3d}\right)\right\rceil = \ell_{MBRB}$ thus proving the lemma. □

C Proof of the Signature-Based $k2\ell$-Cast Implementation (Algorithm 4)

For the proofs provided in this section, let us remind that, given two sets A and B, we have $|A \cap B| = |A| + |B| - |A \cup B|$. Moreover, the number of correct processes c is superior or equal to $n - t$. Additionally, if A and B are both sets containing a majority of correct processes, we have $|A \cup B| \leq c$, which implies that $|A \cap B| \geq |A| + |B| - c$. Furthermore, let us remind the assumptions of Alg. 4:

- sb-$k2\ell$-Assumption 1: $c > 2d$,
- sb-$k2\ell$-Assumption 2: $c - d \geq q_d \geq t + 1$.

C.1 Safety Proof

Lemma 30. *If a correct process p_i $k2\ell$-delivers (m, id), then at least $q_d - n + c$ correct processes have signed (m, id) at line 3.*

Proof. If p_i $k2\ell$-delivers (m, id) at line 16, then it sent q_d valid signatures for (m, id) (because of the predicate at line 15). The effective number of Byzantine processes in the system is $n - c$, such that $0 \leq n - c \leq t$. Therefore, p_i must have sent at least $q_d - n + c$ (which, due to sb-$k2\ell$-Assumption 2, is strictly positive because $q_d > t \geq n - c$) valid distinct signatures for (m, id) that correct processes made at line 3, during a $k2\ell$_cast(m, id) invocation. □

Lemma 31 ($k2\ell$-VALIDITY)**.** *If a correct process p_i $k2\ell$-delivers an app-message m with identity id, then at least $k' = q_d - n + c$ correct processes $k2\ell$-cast m with identity id.*

Proof. The condition at line 2 implies that the correct processes that $k2\ell$-cast (m, id) constitute a superset of those that signed (m, id) at line 3. Thus, by Lemma 30, their number is at least $k' = q_d - n + c$. □

Lemma 32 ($k2\ell$-NO-DUPLICATION)**.** *A correct process $k2\ell$-delivers at most one app-message m with identity id.*

Proof. This property derives trivially from the predicate at line 15. □

Lemma 33 ($k2\ell$-CONDITIONAL-NO-DUPLICITY)**.** *If the Boolean $\delta = q_d > \frac{n+t}{2}$ is* `true`*, then no two different correct processes $k2\ell$-deliver different app-messages with the same identity id.*

Proof. Let p_i and p_j be two correct processes that respectively $k2\ell$-deliver (m, id) and (m', id). We want to prove that, if the predicate $(q_d > \frac{n+t}{2})$ is satisfied, then $m = m'$.

Thanks to the predicate at line 15, we can assert that p_i and p_j must have respectively sent at least q_d valid signatures for (m, id) and (m', id), made by two sets of processes, that we respectively denote A and B, such that $|A| \geq q_d > \frac{n+t}{2}$ and $|B| \geq q_d > \frac{n+t}{2}$. We have $|A \cap B| > 2\frac{n+t}{2} - n = t$. Hence, at least one correct process p_x has signed both (m, id) and (m', id). But because of the predicates at lines 2, p_x signed at most one couple $(-, id)$ during a $k2\ell$_cast(m, id) invocation at line 3. We conclude that m is necessarily equal to m'. □

C.2 Liveness Proof

Lemma 34. *All signatures made by correct processes at line 3 are eventually received by at least $c - d$ correct processes at line 8.*

Proof. Let $\{s_1, s_2, ...\}$ be the set of all signatures for (m, id) made by correct processes at line 3. We first show by induction that, for all z, at least $c - d$ correct processes receive all signatures $\{s_1, s_2, ..., s_z\}$ at line 8.

Base case $z = 0$. As no correct process signed (m, id), the proposition is trivially satisfied.

Induction. We suppose that the proposition is verified at z: signatures $s_1, s_2, ..., s_z$ are received by a set of at least $c - d$ correct processes that we denote A. We now show that the proposition is verified at $z + 1$: at least $c - d$ correct processes eventually receive all signatures $s_1, s_2, ..., s_{z+1}$.

The correct process that makes the signature s_{z+1} ur-broadcasts a BUNDLE$(m, id, sigs)$ message (at line 5) where $sigs$ contains s_{z+1}. From the definition of the MA, BUNDLE$(m, id, sigs)$ is eventually received by a set of at least $c-d$ correct processes that we denote B. We have $|A \cap B| = 2(c-d)-c = c - 2d > 2d - 2d = 0$ (from sb-$k2\ell$-Assumption 1). Hence, at least one correct process p_j eventually receives all signatures $s_1, s_2, ..., s_{z+1}$, and thereafter ur-broadcasts BUNDLE$(m, id, sigs')$ where $\{s_1, s_2, ..., s_{z+1}\} \subseteq sigs'$. Again, from the definition of the MA, BUNDLE$(m, id, sigs')$ is eventually received by a set of at least $c - d$ correct processes at line 8. □

Lemma 35. *If no correct process $k2\ell$-casts (m, id) at line 1, then no correct process $k2\ell$-delivers (m, id) at line 16.*

Proof. Looking for a contradiction, let us suppose that a correct process p_i $k2\ell$-delivers (m, id) while no correct process $k2\ell$-cast (m, id). Because of the condition at line 15, p_i must have ur-broadcast at least q_d valid signatures for (m, id), out of which at most t are made by Byzantine processes. As $q_d > t$ (sb-$k2\ell$-Assumption 2), we know that $q_d - t > 0$. Hence, at least one correct process must have $k2\ell$-cast (m, id). Contradiction. □

Lemma 36 ($k2\ell$-LOCAL-DELIVERY)**.** *If at least $k = q_d$ correct processes $k2\ell$-cast an app-message m with identity id and no correct process $k2\ell$-casts an app-message $m' \neq m$ with identity id, then at least one correct process p_i $k2\ell$-delivers the app-message m with identity id.*

Proof. As no correct process $k2\ell$-casts an app-message $m' \neq m$ with identity id, then Lemma 35 holds, and no correct process can $k2\ell$-deliver (m', id) where $m' \neq m$. Moreover, no correct process can sign (m', id) where $m' \neq m$ at line 3, and thus all $k = q_d$ correct processes that invoke $k2\ell$_cast(m, id) at line 1 also pass the condition at line 2, and then sign (m, id) at line 3. From Lemma 34, we can assert that all q_d signatures are received at line 8 by a set of at least $c - d$ correct processes, that we denote A. Let us consider p_j, one of the processes of A. There are two cases:

- If p_j passes the condition at line 9, then it sends all q_d signatures at line 11, then invokes check_delivery() at line 12, passes the condition at line 15 (if it was not already done before) and $k2\ell$-delivers (m, id) at line 16;
- If p_j does not pass the condition at line 9, then it means that it has already sent all q_d signatures before, whether it be at line 5 or 11, but after that, it necessarily invoked check_delivery() (at line 6 or 12, respectively), passed the condition at line 15 (if it was not already done before) and $k2\ell$-delivered (m, id) at line 16. □

Lemma 37 ($k2\ell$-WEAK-GLOBAL-DELIVERY)**.** *If a correct process $k2\ell$-delivers an app-message m with identity id, then at least $\ell = c - d$ correct processes $k2\ell$-deliver an app-message m' with identity id (each of them possibly different from m).*

Proof. If p_i $k2\ell$-delivers (m, id) at line 16, then it has necessarily ur-broadcast the BUNDLE$(m, id, sigs)$ message containing the q_d valid signatures before, whether it be at line 5 or 11. From the definition of the MA, a set of at least $c - d$ correct processes, that we denote A, eventually receives this BUNDLE$(m, id, sigs)$ message at line 8. If some processes of A do not pass the condition at line 9 upon receiving this BUNDLE$(m, id, sigs)$ message, it means that they already ur-broadcast all signatures of $sigs$. Thus, in every scenario, all processes of A eventually ur-broadcast all signatures of $sigs$ at line 5 or 11. After that, all processes of A necessarily invoke the check_delivery() operation at line 6 or 12, respectively, and then evaluate the condition at line 15. Hence, all correct processes of A, which are at

least $c - d = \ell$, $k2\ell$-deliver some app-message for identity id at line 16, whether it be m or any other app-message. □

Lemma 38 ($k2\ell$-STRONG-GLOBAL-DELIVERY)**.** *If a correct process $k2\ell$-delivers an app-message m with identity id, and no correct process $k2\ell$-casts an app-message $m' \neq m$ with identity id, then at least $\ell = c - d$ correct processes $k2\ell$-deliver m with identity id.*

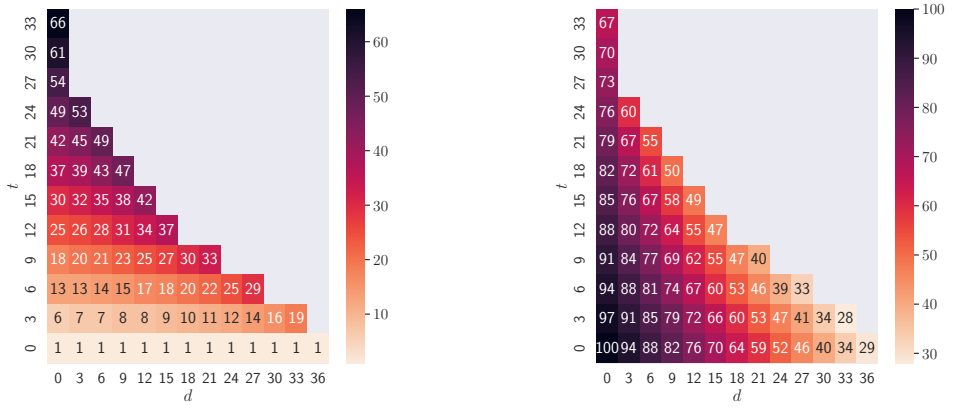
Proof. If a correct process $k2\ell$-delivers (m, id) at line 16, then by Lemma 37, we can assert that at least $\ell = c - d$ correct process eventually $k2\ell$-deliver some app-message (not necessarily m) with identity id. Moreover, as no correct process $k2\ell$-casts (m', id) with $m' \neq m$, then Lemma 35 holds, and we conclude that all ℓ correct processes $k2\ell$-deliver (m, id). □

D Numerical Evaluation

This section presents additional numerical results that complement those of Section 5.3, and provides concrete lower-bound values for the k and ℓ parameters of the $k2\ell$-cast objects used in the reconstructed Bracha MBRB algorithm (Alg. 2, page 12). Results were obtained by considering a network with $n = 100$ processes and varying values of t and d. Fig. 4 and Fig. 5 present the values of k and ℓ for the obj_{E} and obj_{R} of Alg. 2.

The numbers in each cell show the value of k (Figs. 4a and 5a), resp. ℓ (Figs. 4b and 5b) that is required, resp. guaranteed, by the corresponding $k2\ell$-cast object. The two plots show the two roles of the two $k2\ell$-cast objects. The first, obj_{E}, needs to provide agreement among the possibly different messages sent by Byzantine processes (Fig. 4). As a result, it can operate in a more limited region of the parameter space. obj_{R}, on the other hand, would, in principle, be able to support larger values of d and t, but it needs to operate in conjunction with obj_{E} (Fig. 5).

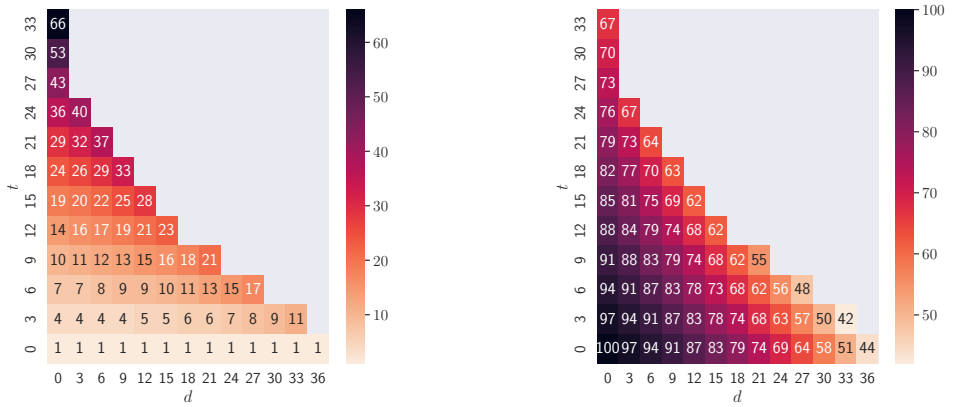
Fig. 3b on page 15 already displays the values of ℓ provided by obj_{W} in the Imbs-Raynal algorithm. Fig. 6 complements it by showing the required values of k for obj_{W}. The extra constraint introduced by chaining the two objects suggests that a single $k2\ell$-cast algorithm could achieve better performance. But this is not the case if we examine the performance of the reconstructed Imbs-Raynal algorithm depicted in Fig. 3b. The reason lies in the need for higher quorum values in obj_{W} due to $single = \texttt{false}$. In the future, we plan to investigate if variants of this algorithm can achieve tighter bounds and explore the limits of signature-free $k2\ell$-cast-based broadcast in the presence of an MA and Byzantine processes.



(a) Minimum required k

(b) Minimum provided ℓ

Figure 4: Required values of k and provided values of ℓ for obj_{E} in the reconstructed Bracha BRB algorithm with varying values of t and d within the ranges that satisfy B87-Assumption



(a) Minimum required k

(b) Minimum provided ℓ

Figure 5: Required values of k and provided values of ℓ for obj_{R} in the reconstructed Bracha BRB algorithm with varying values of t and d within the ranges that satisfy B87-Assumption

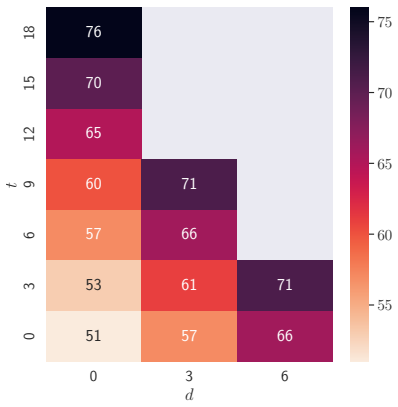


Figure 6: Required values of k for obj_{W} in the reconstructed Imbs & Raynal BRB algorithm with varying values of t and d within the ranges that satisfy IR16-Assumption