Coping with Byzantine Processes and a Message Adversary: Modularity Helps!

Davide Frey, Michel Raynal, François Taïani, Timothé Albouy

Coping with Byzantine Processes and a Message Adversary: Modularity Helps!

Timothé Albouy, Davide Frey, Michel Raynal, François Taïani

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

February 10, 2022

Abstract

This paper explores how reliable broadcast can be implemented when facing a dual adversary that can both corrupt processes and remove messages. More precisely, we consider an asynchronous n-process message-passing systems in which up to t_b processes are Byzantine and where, at the network level, for each message broadcast by a correct process, an adversary can prevent up to t_m processes from receiving it (the integer t_m defines the power of the message adversary). So, differently from previous works, this work considers that not only computing entities can be faulty (Byzantine processes), but also that the network can lose messages. To this end, the paper first introduces a new basic communication abstraction denoted $k\ell$-cast, and studies its properties in this new bi-dimensional adversary context. Then, the paper deconstructs existing Byzantine-tolerant asynchronous broadcast algorithms and, with the help of the $k\ell$-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. The paper also shows that the condition $n > 3t_b + 2t_m$ is necessary and sufficient (with signatures) to design such reliable broadcast algorithms.

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. 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 [23]. If processes may fail, and in particular if failed processes may behave arbitrarily (a failure known as Byzantine [17, 22]), implementing reliable broadcast becomes far from trivial as Byzantine processes may collude to fool correct processes [24]. 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, 5, 8, 9, 14, 16, 18, 20, 21, 24]. Existing BRB algorithms typically assume they execute over a *reliable* point-to-point network, i.e. essentially 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 (as illustrated in the TCP protocol).

In this work, we take a drastic turn away from this usual assumption and explore how BRB might be provided when processes execute on an *unreliable* network that might lose point-to-point messages. Our motivation is threefold: (i) 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 have the potential to limit the performance of BRB algorithms drastically, a well-known phenomenon in systems research [11, 12, 30]. (ii) Second, re-transmissions require that correct processes be eventually able to receive messages and cannot, therefore, model the permanent disconnection of correct processes. (iii) 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 [27, 28][1], and then used (sometimes implicitly) in many works (e.g., [3, 4, 10, 25, 26, 28, 29]). 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.

Content of the paper. This paper combines a Message Adversary with Byzantine processes, and studies the implementation of Byzantine Reliable Broadcast (BRB) in an asynchronous fully connected network subject to this MA and to at most t_b Byzantine faults. The MA models lossy connections by preventing up to t_m 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.)

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 to mislead correct processes.

For clarity, in the remainder of the paper, we call "*implementation messages*" (or *imp-messages*) the point-to-point network messages used internally by a BRB algorithm. (The MA may suppress imp-messages.) We distinguish these imp-messages from the messages that 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 $k\ell$-cast, which appears to be a base building block to implement BRB abstractions (with or without the presence of an MA). This communication abstraction, which is based on a systematic dissociation of the predicate used to forward imp-messages from the predicate that allows a process to deliver an app-message, is at the heart of the work presented in the paper. When proving the $k\ell$-cast communication abstraction, the paper

[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 in [13], which considers static κ-connected networks. If the adversary selects statically, for each correct sender, t_m correct processes that do not receive any of this sender's messages, the proposed model includes Dolev's model where $\kappa = n - t_m$.

presents an in-depth analysis of the power of the two-dimensional adversary that constitutes the pair made up of at most t_b Byzantine processes + an MA of power t_m.

- Then, the paper deconstructs two BRB algorithms (Bracha's [8] and Imbs and Raynal's [16] algorithms) and reconstructs versions of them that tolerate *both* Byzantine processes and MA. In an interesting way, 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.[3]

Roadmap. The paper is composed of 7 sections and 4 appendices. Section 2 describes the underlying computing model. Section 3 presents the $k\ell$-cast abstraction and its properties. Section 4 defines the MA-tolerant BRB communication abstraction. Section 5 shows that thanks to the $k\ell$-cast abstraction, existing BRB algorithms can give rise to MA-tolerant BRB algorithms which, when $t_m = 0$, are more efficient than the BRB algorithms they originate from. Section 6 presents a signature-based implementation of $k\ell$-cast that possesses optimal guarantees. Finally, Section 7 concludes the paper. Due to page limitation, some contributions are presented in appendices: (1) some proofs, (2) the fact that $n > 3t_b + 2t_m$ is a necessary and sufficient condition (with/without signatures) for BRB in the presence of an MA, and (3) a numerical evaluation of the $k\ell$-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 an identity, and all the identities are different and known by the processes. To simplify the presentation and without loss of generality, we assume that i is the identity of p_i.

On the failure side, up to $t_b \geq 0$ processes can be Byzantine, where a Byzantine process is a process whose behavior does not follow the code specified by its algorithm [17, 22]. Let us notice that Byzantine processes can collude to fool the non-Byzantine processes (also called correct processes). Let us also notice that, in this model, the premature stop (crash) of a process is a Byzantine failure. In the following, given an execution, c denotes the number of processes that behave correctly in that execution. We always have $n - t_b \leq c \leq n$. While this number remains unknown to correct processes, it is used in the following to analyze and characterize (more precisely than using its worse value $n - t_b$) 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. The processes communicate by exchanging imp-messages through a fully connected asynchronous point-to-point communication network, which is assumed to be reliable in the sense it neither corrupts, duplicates nor creates imp-messages. As far as imp-messages losses are concerned, the network is under the control of an adversary (see below) that can suppress imp-messages.

Let MSG be an imp-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** $i \in \{1, \cdots, n\}$ **do** send MSG(v) **to** p_j **end for**". It is assumed that all the correct processes invoke

[3] As already said, the algorithm presented in [3] is a *signature-based* BRB algorithm that tolerates imp-message losses. Moreover, directly expressed on top of the *lowest level* network operations, namely the send and receive operations, it does not benefit from an underlying communication abstraction that would improve its efficiency.

ur_broadcast to send imp-messages, and we say that the imp-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 imp-message MSG(v). Moreover, due to its very nature, a Byzantine process can send fake imp-messages without using the macro-operation ur_broadcast.

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

As an example, let us consider a set D of correct processes, where $1 \leq |D| \leq t_m$, such that during some period of time, the MA suppresses all the imp-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.

3 $k\ell$-Cast Abstraction

3.1 Definition

$k\ell$-cast is a communication abstraction belonging to the many-to-many family[5]. Each of the $k\ell$-cast instances is defined by the values of four parameters denoted k', k, ℓ, and δ. $k\ell$-cast provides two operations denoted $k\ell$_cast and $k\ell$_deliver, and we consequently say that a process *$k\ell$-casts* and *$k\ell$-delivers* app-messages. Moreover, an app-message m has an identity id. (Typically, such an identity is a pair consisting of a process identity and a sequence number.) Intuitively, given an app-message m with identity id, $k\ell$-cast relates the number k of correct processes that $k\ell$-cast the pair (m, id) with the minimal number ℓ of correct processes that $k\ell$-deliver (m, id). More precisely, an object $k\ell$-cast (k',k,ℓ,δ) is defined by the following properties:

[4]This definition can be generalized for an operation ur_multicast that sends an imp-message to an arbitrary subset of processes if we suppose that the MA can still suppress up to t_m copies of this imp-message. In this context, the best way for correct processes to disseminate an imp-message is to communicate using the ur_broadcast operation.

[5]An example that belongs to this family is the binary reliable broadcast introduced in [19], 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.

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

Let us note that, when $k' = 0$, this specification does not prevent correct processes from $k\ell$-delivering an app-message $k\ell$-cast only by Byzantine processes. Let us also note that when δ is `true`, the $k\ell$-CONDITIONAL-NO-DUPLICITY property implies that all the app-messages m' involved in the $k\ell$-WEAK-GLOBAL-DELIVERY property are equal to m.

When one wants to implement the $k\ell$-cast communication abstraction, the main difficulty lies in the noise created by the imp-messages sent/forwarded by (unknown) Byzantine processes, while striving to have k' and ℓ as great as possible and k as small as possible. Let obj_1 and obj_2 be $k\ell$-cast objects defined by the parameters $(k'_1, k_1, \ell_1, \delta_1)$ and $(k'_2, k_2, \ell_2, \delta_2)$, respectively: obj_1 is *at least as strong* as obj_2 if $(k'_1 \geq k'_2) \wedge (k_1 \leq k_2) \wedge (\ell_1 \geq \ell_2) \wedge (\delta_2 \Rightarrow \delta_1)$. It is *strictly stronger* if additionally $(k'_1 > k'_2) \vee (k_1 < k_2) \vee (\ell_1 > \ell_2)$. As we will see, if obj_1 is strictly stronger than obj_2, it ensures more $k\ell$-deliveries than obj_2, while having weaker constraints on the number of required corresponding $k\ell$_cast operations invoked by correct processes.

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

Among the many possible implementations of $k\ell$-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 disseminates imp-messages to capture each process's perception of the system's state, and relies on *two thresholds* to drive this dissemination and decide delivery (a pattern also found for instance in Bracha's BRB algorithm [8]):

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

[7]In this paper, a quorum is a set of processes that (at the implementation level) ur-broadcast the same imp-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 $\frac{n+t_b}{2}$, the intersection of any two quorums contains, despite Byzantine processes, at least one correct process [8, 24].

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

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

The forwarding mechanism driven by q_f is instrumental in ensuring the $k\ell$-cast properties of the algorithm. Forwarding creates a phase transition in which, as soon as some critical "mass" of agreeing imp-messages accumulates within the system, a chain reaction ensures that a minimum number of correct processes eventually $k\ell$-deliver the corresponding app-message. Concretely, our algorithm takes the form of an object (SigFreeKLCast, Algorithm 1), instantiated using the function SigFreeKLCast$(q_d, q_f, single)$ according to three input parameters:

- q_d: the number of matching ur-broadcast imp-messages that must be received from distinct processes in order to $k\ell$-deliver an app-message.
- q_f: the number of matching ur-broadcast imp-messages that must be received from distinct processes in order to forward the received app-message.
- $single$: a Boolean that controls under which conditions $k\ell$-CONDITIONAL-NO-DUPLICITY is provided by the algorithm. When $single$ is `false`, the algorithm allows a single correct process to forward different app-message with the same identity id, otherwise it does not.

```
object SigFreeKLCast(q_d, q_f, single) is

(1)  operation kℓ_cast(m, id) is
(2)    if (MSG(−, id) not already ur-broadcast)
(3)      then ur_broadcast(MSG(m, id))
(4)    end if.

(5)  when MSG(m, id) is received do
       % ———————— forwarding step ————————
(6)    if (MSG(m, id) received from at least q_f processes
         ∧ ((¬single ∧ MSG(m, id) not already ur-broadcast)
           ∨ MSG(−, id) not already ur-broadcast))
(7)      then ur_broadcast(MSG(m, id))
(8)    end if;
       % ———————— delivery step ————————
(9)    if (MSG(m, id) received from at least q_d processes
         ∧ (−, id) not already kℓ-delivered)
(10)     then kℓ_deliver(m, id)
(11)   end if.

end object.
```

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

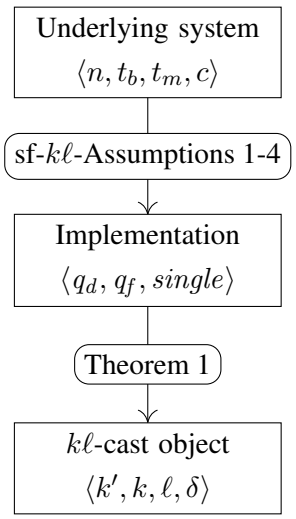


Figure 1: From the system parameters to a $k\ell$-cast implementation

The module exports operations $k\ell$_cast and $k\ell$_deliver. Given an app-message m with identity id, the operation $k\ell$_cast(m, id) ur-broadcasts the imp-message MSG(m, id) only if no identity-conflicting imp-message was previously ur-broadcast (lines 2-4). The reception of the imp-message MSG(m, id) by a process p_i entails the execution of two steps. The first one is a forwarding step, controlled by the size q_f of the forwarding quorum (lines 6-8), while the second is a $k\ell$-delivery step controlled by the size q_d of the $k\ell$-delivery quorum (lines 9-11).

For the sake of clarity, we define $\alpha = n + q_f - t_b - t_m - 1$. Given an execution defined by the system parameters n, t_b, t_m, and c, Algorithm 1 requires the following assumptions to hold for the input

parameters q_f and q_d of a $k\ell$-cast instance (a global picture linking all parameters is presented in Fig. 1). The prefix "sf" stands for signature-free.

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

In particular, the safety of Algorithm 1 algorithm relies solely on sf-$k\ell$-Assumption 1, while its liveness relies on all four of them. sf-$k\ell$-Assumption 2 through 4 constrain the solutions of a second-degree inequality resulting from the combined action of the MA, the Byzantine nodes, and the message-forwarding behavior of Algorithm 1. We show in Appendix B that, in practical cases, these assumptions can be satisfied by a bound of the form $n > \lambda t_b + \xi t_m + f(t_b, t_m)$, where $\lambda, \xi \in \mathbb{N}$ and $f(t_b, 0) = f(0, t_m) = 0$. Together, the assumptions allow Algorithm 1 to provide a $k\ell$-cast abstraction (with values of the parameters k', k, ℓ, and δ defining a specific $k\ell$-cast instance) as stated by the following theorem.

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

- $k\ell$-VALIDITY *with* $k' = q_f - n + c$,
- $k\ell$-NO-DUPLICATION,
- $k\ell$-CONDITIONAL-NO-DUPLICITY *with* $\delta = \left(q_f > \dfrac{n + t_b}{2}\right) \vee \left(single \wedge q_d > \dfrac{n + t_b}{2}\right)$,
- $k\ell$-LOCAL-DELIVERY *with* $k = \left\lfloor \frac{c(q_f - 1)}{c - t_m - q_d + q_f} \right\rfloor + 1$,
- $\begin{cases} \textit{if } single = \texttt{false}, & k\ell\text{-WEAK-GLOBAL-DELIVERY} \\ \textit{if } single = \texttt{true}, & k\ell\text{-STRONG-GLOBAL-DELIVERY} \end{cases}$ *with* $\ell = \left\lceil c\left(1 - \frac{t_m}{c - q_d + 1}\right)\right\rceil$.

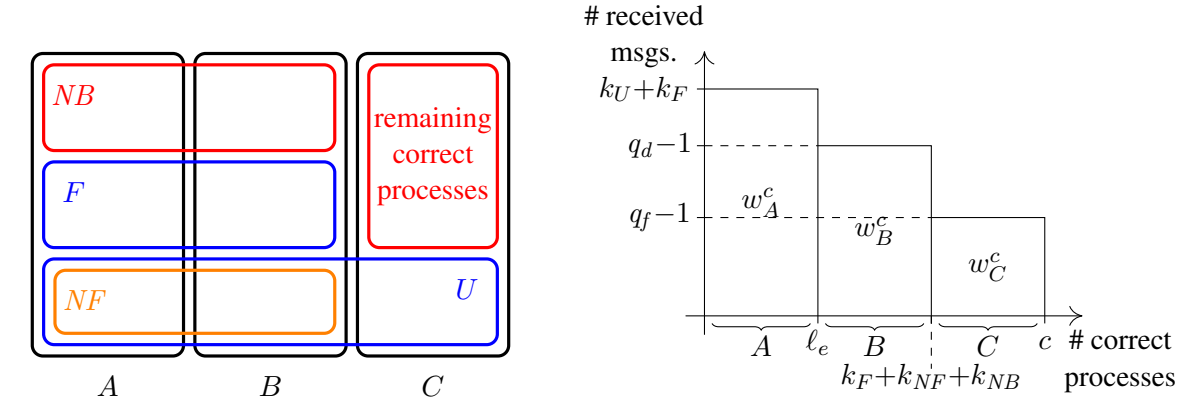
3.3 Proof of the Signature-Free $k\ell$-Cast Algorithm

The proofs of the $k\ell$-cast safety properties stated in Theorem 1 ($k\ell$-VALIDITY, $k\ell$-NO-DUPLICATION, and $k\ell$-CONDITIONAL-NO-DUPLICITY) are fairly straightforward. Due to page limitation, these proofs (Lemmas 10-13) are given in Appendix A.1.

Informal sketches of the proofs of the $k\ell$-cast liveness properties ($k\ell$-LOCAL-DELIVERY, $k\ell$-WEAK-GLOBAL-DELIVERY, $k\ell$-STRONG-GLOBAL-DELIVERY) are given below (Lemmas 1-9), but the full developments can be found in Appendix A.2. To violate the liveness properties of $k\ell$-cast, the attacker can control the distribution of imp-messages across correct processes to some extent, thanks to the MA. Considering an execution of our $k\ell$-cast algorithm in which k_I correct processes $k\ell$-cast (m, id), and ℓ_e correct processes receive at least q_d ur-broadcast imp-messages MSG(m, id)[9], we identify the subsets of correct processes depicted in Fig. 2a. Let us first identify subsets based on the number of MSG(m, id) imp-messages they receive.

- The subset A contains the ℓ_e correct processes that receive at least q_d MSG(m, id) imp-messages (whether it be from correct or from Byzantine processes).

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



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

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

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

- The subset B contains the correct processes that receive at least q_f but less than q_d MSG(m, id) imp-messages and thus do not $k\ell$-deliver (m, id).
- The subset C contains the remaining correct processes that receive less than q_f MSG(m, id) imp-messages.

The notation w_A^c (resp. w_B^c, w_C^c) denotes the total number of MSG(m, id) imp-messages ur-broadcast by correct processes and received by processes of A (resp. B, C). We also identify subsets based on the ur-broadcast operations they perform.

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

As previously, we denote the cardinality of each of these subsets as: $k_U = |U|$, $k_F = |F|$, $k_{NF} = |NF|$, $k_{NB} = |NB|$. We can then observe that $k_U \leq k_I$ and that $k_{NF} \leq k_U$, as all the (correct) processes in U and NF invoke $k\ell$-cast. Let us remark that $(k_U + k_F)$ represents the total number of correct processes that ur-broadcast an imp-message MSG(m, id). Fig. 2b presents a distribution of imp-messages across A, B and C.

Observation. By construction, we can bind w_A^c by observing that each of the ℓ_e correct processes in A can receive at most one imp-message from each of the $(k_U + k_F)$ correct processes that send them. We can also bind 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 $q_d - 1$ imp-messages. Similarly, we can bind w_C^c by observing that the

$(c - k_{NF} - k_{NB} - k_F)$ processes of C can receive at most $q_f - 1$ imp-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 can suppress t_m copies of the imp-message that should be received by the c correct processes. Thus, the total number of imp-messages received by correct processes $(w_A^c + w_B^c + w_C^c)$ is such that:

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

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

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

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

Lemma 3 ($k\ell$-LOCAL-DELIVERY)**.** *If at least $k = \left\lfloor \frac{c(q_f-1)}{c-t_m-q_d+q_f} \right\rfloor + 1$ correct processes $k\ell$-cast an app-message m with identity id and no correct process $k\ell$-casts any app-message m' with identity id such that $m \neq m'$, then at least one correct process p_i $k\ell$-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 $k\ell$-delivers (m, id), which leads us to $\ell_e = 0$. Using prior information and sf-$k\ell$-Assumption 1, we can rewrite the inequality of Lemma 1 to get the threshold of $k\ell$-casts above which there is at least one $k\ell$-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 $k\ell$-delivers (m, id) and $x = k_U + k_F$ (the number of correct processes that ur-broadcast* MSG(m, id) *at line* 3 *or* 7*), then $x \geq q_d - t_b$ and $x^2 - x(c - t_m + q_f - 1 - k_{NB}) \geq -(c - k_{NB})(q_f - 1)$.*

Proof sketch. We prove this lemma by counting the total number of messages (Byzantine or not) that are received by processes of A, and by using (1), (3) (4), and sf-$k\ell$-Assumption 1. (Full derivations in Appendix A.2.) □

Lemma 6. *If $k_{NB} = 0$, and at least one correct process $k\ell$-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 $\left\lceil c\left(1 - \frac{t_m}{c - q_d + 1}\right)\right\rceil$ correct processes $k\ell$-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-$k\ell$-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 $k\ell$-deliver under the given hypotheses. (Full derivations in Appendix A.2.) □

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

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

Proof sketch. As $single$ = `true`, Lemma 2 holds and implies that $k_{NB} = 0$. Like above, Lemma 6 and Lemma 7 hold, which leads us to the conclusion of the proof. (Full derivations in Appendix A.2.) □

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

The MBR-broadcast abstraction (for Message-adversarial Byzantine Reliable Broadcast) is composed of two matching operations denoted mbrb_broadcast and mbrb_deliver. It considers that an identity $\langle sn, i\rangle$ (sequence number, sender identity) is associated with each app-message, and assumes that any two app-messages mbrb-broadcast by the same correct process have different sequence numbers. Sequence numbers are one of the easiest ways to design "multi-shot" reliable broadcast algorithms (be the app-messages received in their sending order or not), when the mbrb_broadcast operation can be invoked multiple times by the same process.

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 returns the tuple (m, sn, j) to the client application, 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 $k\ell$-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]. MBR-broadcast 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.

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

- 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 different 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.

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 imp-messages sent to a fixed set D of t_m processes, these mbrb-delivery properties are the strongest that can be implemented. More generally, the best guaranteed value for ℓ_{MBRB} is $c-t_m$. So, the previous specification boils down to Bracha's specification [8] for $\ell_{MBRB} = c$.

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

This section revisits two signature-free BRB algorithms [8, 16] that were initially proposed in a pure Byzantine context (i.e. without any MA), and uses the $k\ell$-cast abstraction to transform them into Byzantine-MA-tolerant versions (hence they are MBRB algorithms). Moreover, when we take $t_m = 0$, our two revisited BRB algorithms are more efficient than the original algorithms that gave rise to them.

To make the reading easier for people who are familiar with the previous algorithms, the tag of the imp-messages (INIT, ECHO, READY, WITNESS) are the same in the original algorithms and in their revisited formulations.

5.1 Bracha's BRB algorithm revisited

Revisited version. Bracha's BRB algorithm is a kind of *three-phase commit* algorithm. When a process invokes $\mathsf{brb\_broadcast}(m, sn)$, it disseminates the app-message m with the help of an imp-message tagged INIT (first phase). The reception of this imp-message by a correct process entails its participation in a second phase implemented by the exchange of imp-messages tagged ECHO. Finally, when a process has received ECHO imp-messages from "enough" processes, it enters the third phase implemented by the exchange of imp-messages tagged READY, at the end of which it brb-delivers the app-message m. Algorithm 2 is a revisited version of the Bracha's BRB which assumes $n > 3t_b + 2t_m + 2\sqrt{t_b t_m}$.

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

• obj_{E}: $q_f = t_b + 1$ and $q_d = \lfloor \frac{n+t_b}{2} \rfloor + 1$, • obj_{R}: $q_f = t_b + 1$ and $q_d = 2t_b + t_m + 1$.

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 $\langle sn, i \rangle$.

Algorithm 2 provides $\ell_{MBRB} = \left\lceil c\left(1 - \frac{t_m}{c-2t_b-t_m}\right)\right\rceil$ under:

• B87-Assumption: $n > 3t_b + 2t_m + 2\sqrt{t_b t_m}$;

its proof of correctness can be found in Appendix B.1.

init: $obj_{\text{E}} \leftarrow$ SigFreeKLCast(q_d=$\lfloor \frac{n+t_b}{2} \rfloor$+1, q_f=t_b+1, $single$=`true`);
$obj_{\text{R}} \leftarrow$ SigFreeKLCast(q_d=2t_b+t_m+1, q_f=t_b+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** $obj_{\text{E}}.k\ell$_cast(ECHO(m), (sn, j)).

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

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

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

Comparison. When $t_m = 0$, the quorum sizes in Bracha's algorithm and its revisited version are the same for the READY phase. Those for the ECHO phase appear instead in Table 1. As the algorithm requires $n > 3t_b$, we define $\Delta = n - 3t_b$ as the slack between the lower bound on n and the actual value of n. When considering the delivery threshold q_f, we have $\lfloor \frac{n+t_b}{2} \rfloor + 1 = 2t_b + \lfloor \frac{\Delta}{2} \rfloor + 1 > t_b + 1$. As a result, the revisited version of Bracha's algorithm always has a lower forwarding threshold than the original. This causes it to forward messages more rapidly and therefore reach the delivery quorum faster, even if the two delivery quorums are the same.

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

Table 1: Bracha's original version vs. $k\ell$-cast-based rewriting when $t_m = 0$

5.2 Imbs and Raynal's BRB algorithm revisited

Revisited version. Imbs and Raynal's BRB is another implementation of Byzantine reliable broadcast which achieves an optimal good-case latency (only two communication steps) at the cost of a non-optimal t_b-resilience. The revisited version of Imbs and Raynal's BRB algorithm requires $n > 5t_b + 12t_m + \frac{2t_b t_m}{t_b + 2t_m}$.

init: $obj_{\text{W}} \leftarrow$ SigFreeKLCast(q_d=$\lfloor \frac{n+3t_b}{2} \rfloor + 3t_m + 1$, q_f=$\lfloor \frac{n+t_b}{2} \rfloor + 1$, $single$=`false`).

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

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

(3) **when** (WITNESS(m), (sn, j)) **is** $obj_{\text{W}}.k\ell$_delivered **do** mbrb_deliver(m, sn, j).

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

The algorithm requires a single $k\ell$-cast object, denoted obj_{W}, associated with the WITNESS imp-message, and which is instantiated with $q_f = \lfloor \frac{n+t_b}{2} \rfloor + 1$ and $q_d = \lfloor \frac{n+3t_b}{2} \rfloor + 3t_m + 1$, and the Boolean $single =$ `false`. Similarly to Bracha's revisited 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.

Algorithm 3 provides $\ell_{MBRB} = \left\lceil c \left(1 - \frac{t_m}{c - \left\lfloor \frac{n+3t_b}{2} \right\rfloor - 3t_m} \right) \right\rceil$ under:

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

its proof of correctness can be found in Appendix B.2.

Comparison. The original Imbs and Raynal's algorithm and its $k\ell$-revisited version are compared for $t_m = 0$ in Table 2. Let us recall that this algorithm saves one communication step with respect to Bracha's algorithm at the cost of a weaker t_b-tolerance, namely it requires $n > 5t_b$. Like for Bracha, let us define the slack between n and its minimum as $\Delta = n - 5t_b$, we have $\Delta \geq 1$.

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

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

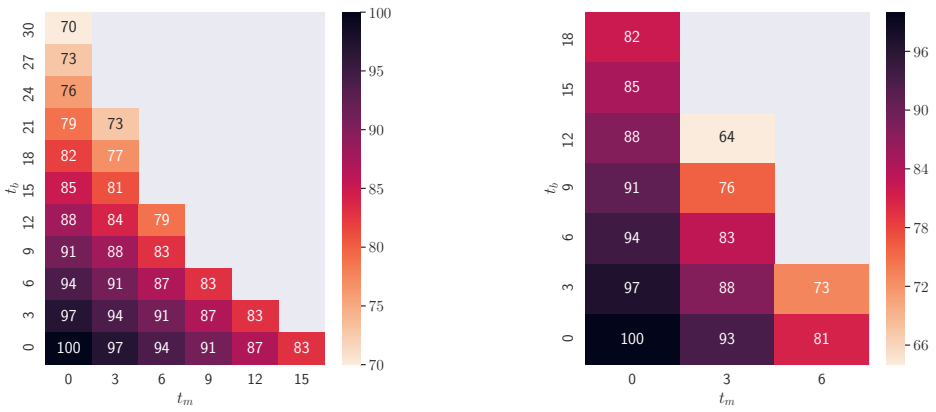
Table 2: Imbs and Raynal's original version vs. $k\ell$-cast-based rewriting when $t_m = 0$

5.3 Numerical evaluation of the MBRB algorithms

Fig. 3 provides a numerical evaluation of the delivery guarantees of both $k\ell$-cast-based MBRB algorithms (Algorithms 2 and 3) in the presence of Byzantine processes and an MA. Results were obtained for $n = 100$, and show the values of ℓ_{MBRB} for different combinations of t_b and t_m. For instance, Fig. 3a shows that with 6 Byzantine processes, and an MA suppressing up to 9 ur-broadcast imp-messages, Algorithm 2 ensures that the MBRB-GLOBAL-DELIVERY property is verified with $\ell_{MBRB} = 83$. The figures illustrate that the revisited Bracha algorithm performs on a wider range of parameter values as indicated by the bounds on n, t_b, and t_m in assumptions 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. Appendix E presents more in-depth results on both algorithms and their constituent $k\ell$-cast instances.

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

This section presents an implementation of $k\ell$-cast based on digital signatures. The underlying model is the same as the one presented in Section 2 (page 3), except that we assume that the computing power of the attacker is bounded, which allows us to leverage asymmetric cryptography.



(a) Revisited Bracha algorithm (Alg. 2) (b) Revisited Imbs-Raynal algorithm (Alg. 3)

Figure 3: Provided values of ℓ_{MBRB} for the revisited BRB algorithms with varying values of t_b and t_m ($n = 100$)

Although this signature-based implementation of $k\ell$-cast provides better guarantees than its signature-free counterpart (Algorithm 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 [3].

However, a signature-based $k\ell$-cast does make sense in contexts in which many-to-many communication patterns are required [7], and we believe opens the path to novel ways to handle local state resynchronization resilient to Byzantine failures and message adversaries.

6.1 Algorithm

The signature-based algorithm is described in Algorithm 4. It uses an asymmetric cryptosystem to sign imp-messages and verify their authenticity. Every process has a public/private key pair. We suppose that the public keys are known by everyone and that each correct process is the only one to know its private key (while Byzantine processes can 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 imp-message, and check if a signature is valid.

It is a simple and classical algorithm taking into account the fact that an app-message must be $k\ell$-cast by at least k correct processes to be $k\ell$-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 imp-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 an imp-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-$k\ell$-Assumption 1: $c > 2t_m$,
- sb-$k\ell$-Assumption 2: $c - t_m \geq q_d \geq t_b + 1$.

```
object SigBasedKLCast(q_d) is

(1)  operation kℓ_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 kℓ-delivered)
(16)     then kℓ_deliver(m, id)
(17)   end if.

end object.
```

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

Similarly to the signature-free $k\ell$-cast implementation presented in Section 3.2, this $k\ell$-cast object can be instantiated using the function called SigBasedKLCast(q_d), which only takes as a parameter the size of the delivery quorum q_d. Thanks to digital signatures, processes can relay the imp-messages of other processes in this signature-based implementation, however, it does not use forwarding in the same sense as in its signature-free free counterpart: there is no equivalent of q_f in this algorithm, 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 $k\ell$_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).

6.2 Guarantees

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

Theorem 2 ($k\ell$-CORRECTNESS)**.** *If sb-$k\ell$-Assumption* 1 *and* 2 *are verified, Algorithm* 4 *implements $k\ell$-cast with the following guarantees: (i) $k' = q_d - n + c$, (ii) $k = q_d$, (iii) $\ell = c - t_m$, and (iv) $\delta = q_d > \frac{n+t_b}{2}$.*

7 Conclusion

This paper discussed reliable broadcast in asynchronous systems where failures are under the control of a two-dimensional adversary: some processes can be Byzantine and imp-messages can be suppressed. 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 (as shown in [6, 15], distributed money transfers are such applications).

Applying the $k\ell$-cast-based approach for local-state process re-synchronization seems worth studying. More generally, the construction of quorums able to thwart a two-dimensional adversary (Byzantine

processes + MA) is a new approach that can be applied to the design of a wide range of quorum-based distributed algorithms other than reliable broadcast.

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

[2] Abraham I., Ren L., and Xiang Z., Good-case and bad-case latency of unauthenticated Byzantine broadcast: a complete characterization. *Proc. 25th Conference on Principles of Distributed Systems (OPODIS'21)*, LIPIcs, Vol. xxx, 20 pages (2021)

[3] Albouy T., Frey D., Raynal M., and Taïani F., Byzantine-tolerant reliable broadcast in the presence of silent churn. *Proc. 23th Int'l Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'21)* Springer LNCS 13046, pp. 21-33 (2021)

[4] Afek Y. and Gafni E., Asynchrony from synchrony. *Proc. Int'l Conference on Distributed Computing and Networking (ICDCN'13)*, Springer LNCS 7730, pp. 225-239, (2013)

[5] Attiya H. and Welch J., *Distributed computing: fundamentals, simulations and advanced topics*, (2d Edition), Wiley-Interscience, 414 pages (2004)

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

[7] Auvolat A., Raynal M., Taïani F., Byzantine-Tolerant Set-Constrained Delivery Broadcast. *Proc. 23rd International Conference on Principles of Distributed Systems (OPODIS 2019)*, LIPIcs 153, pp. 6:1-6:23 (2019)

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

[9] Cachin Ch., Guerraoui R., and Rodrigues L., *Reliable and secure distributed programming*, Springer, 367 pages, ISBN 978-3-642-15259-7 (2011)

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

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

[12] Didona D., Zwaenepoel W., Size-aware Sharding For Improving Tail Latencies in In-memory Key-value Stores. *Prod. 16th USENIX Symposium on Networked Systems Design and Implementation, (NSDI 2019)*, USENIX Association, pp. 79-94 (2019)

[13] Dolev D., The Byzantine generals strike again. *Journal of Algorithms*, 3:14-20 (1982)

[14] Guerraoui G., Komatovic J., Kuznetsov P., Pignolet P.A., Seredinschi D.-A., and Tonkikh A., Dynamic Byzantine reliable broadcast. *Proc. 24th Int'l Conference on Principles of Distributed Systems (OPODIS'20)*, LIPIcs Vol. 184, Article 23, 18 pages (2020)

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

[16] Imbs D. and Raynal M., Trading t-resilience for efficiency in asynchronous Byzantine reliable broadcast. *Parallel Processing Letters*, Vol. 26(4), 8 pages (2016)

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

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

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

[20] Malkhi D. and Reiter M.K., Byzantine Quorum Systems. *Distributed Computing*, 11(4): 203–213 (1998)

[21] Nayak K., Ren L., Shi E., Vaidya N.H., Xiang Z., Improved extension protocols for Byzantine broadcast and agreement. *Proc. 34rd Int'l Symposium on Distributed Computing (DISC'20)*, LIPIcs Vol. 179, Article 28, 16 pages (2020)

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

[23] Raynal M., *Distributed algorithms for message-passing systems*. Springer, 510 pages, ISBN 978-3-642-38122-5 (2013)

[24] Raynal M., *Fault-tolerant message-passing distributed systems: an algorithmic approach.* Springer, 480 pages, ISBN 978-3-319-94140-0 (2018)

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

[26] Raynal M., Message adversaries. *Encyclopedia of Algorithms*, Springer (2015)

[27] Santoro N. and Widmayer P., Time is not a healer. *Proc. 6th Annual Symposium on Theoretical Aspects of Computer Science (STACS'89)*, Springer LNCS 349, pp. 304-316 (1989)

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

[29] Tseng L., Zhang Q., Kumar S., and Zhang Y., Exact Consensus under Global Asymmetric Byzantine Links. *Proc. 40th IEEE International Conference on Distributed Computing Systems (ICDCS 2020)*, pp. 721-731 (2020)

[30] Yang L., Park S. J., Alizadeh M., Kannan S., Tse D., DispersedLedger: High-Throughput Byzantine Consensus on Variable Bandwidth Networks. *Prof. 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI'22)*, USENIX Association, pp. 493-512 (2022)

A Proof of the Signature-Free $k\ell$-cast Implementation

A.1 Safety Proof

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

Proof. If p_i $k\ell$-delivers (m, id) at line 10, then it received q_d copies of MSG(m, id) (because of the predicate at line 9). The effective number of Byzantine processes in the system is $n - c$, such that $0 \leq n - c \leq t_b$. Therefore, p_i must have received at least $q_d - n + c$ (which is strictly positive because $q_d \geq q_f > t_b \geq n - c$ by sf-$k\ell$-Assumption 1) imp-messages MSG(m, id) that correct processes ur-broadcast, either during a $k\ell$_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 MSG(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-$k\ell$-Assumption 1) correct processes have ur-broadcast MSG(m, id) at line 3.
- If at least one correct process forwarded MSG(m, id), then let us consider p_j, the first correct process that forwards MSG(m, id). Because of the predicate at line 6, p_j must have received at least q_f distinct copies of the MSG(m, id) imp-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 $q_f > t_b \geq n - c$ by sf-$k\ell$-Assumption 1) have been sent by correct processes. Moreover, as p_j is the first correct process that forwards MSG(m, id), all of the $q_f - n + c$ imp-messages it receives from correct processes must have been sent at line 3. □

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

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

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

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

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

Proof. Let p_i and p_j be two correct processes that respectively $k\ell$-deliver (m, id) and (m', id). We want to prove that, if the predicate $\big((q_f > \frac{n+t_b}{2}) \vee (single \wedge q_d > \frac{n+t_b}{2})\big)$ is satisfied, then $m = m'$. There are two cases.

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

  We denote by A and B the sets of correct processes that have respectively ur-broadcast MSG(m, id) and MSG(m', id) at line 3. By Lemma 10, we know that $|A| \geq q_f - n + c > \frac{n+t_b}{2} - n + c$ and $|B| \geq q_f - n + c > \frac{n+t_b}{2} - n + c$. As A and B contain only correct processes, we have $|A \cap B| > 2(\frac{n+t_b}{2} - n + c) - c = t_b - n + c \geq t_b - t_b = 0$. Hence, at least one correct process p_x has ur-broadcast both MSG(m, id) and MSG(m', id) at line 3. But because of the predicate at line 2, p_x ur-broadcasts at most one imp-message MSG$(-, id)$ at line 3. We conclude that m is necessarily equal to m'.

- Case $\big(single \wedge q_d > \frac{n+t_b}{2}\big)$.

  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 $\text{MSG}(m, id)$ and $\text{MSG}(m', id)$, from two sets of processes, that we respectively denote A and B, such that $|A| \geq q_d > \frac{n+t_b}{2}$ and $|B| \geq q_d > \frac{n+t_b}{2}$. We have $|A \cap B| > 2\frac{n+t_b}{2} - n = t_b$. Hence, at least one correct process p_x has ur-broadcast both $\text{MSG}(m, id)$ and $\text{MSG}(m', id)$. But because of the predicates at lines 2 and 6, and as $single = \texttt{true}$, p_x ur-broadcasts at most one imp-message $\text{MSG}(-, id)$, either during a $k\ell\_\texttt{cast}(m, id)$ 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. $\ell_e \times (k_U + k_F - q_d + 1) \geq (k_U + k_F)(c - t_m - q_d + q_f) - c(q_f - 1) - k_{NB}(q_d - q_f)$.

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

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

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

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

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

Proof. Assume there is a correct process that ur-broadcasts $\text{MSG}(m', id)$ 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 imp-messages $\text{MSG}(m', id)$ from distinct processes. Since $q_f > t_b$ (sf-$k\ell$-Assumption 1), at least one of these processes, p_j, is correct. Since p_i is the first correct process to forward $\text{MSG}(m', id)$ at line 7, the $\text{MSG}(m', id)$ imp-message of p_j must come from line 3, and p_j must have $k\ell$-cast (m', id). We have assumed that no correct process $k\ell$-cast $m' \neq m$, therefore $m' = m$. Contradiction.

We conclude that, under these assumptions, no correct process ur-broadcasts $\text{MSG}(m', id)$ 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 ($k\ell$-LOCAL-DELIVERY)**.** *If at least $k = \left\lfloor \frac{c(q_f - 1)}{c - t_m - q_d + q_f} \right\rfloor + 1$ correct processes $k\ell$-cast an app-message m with identity id and no correct process $k\ell$-casts any app-message m' with identity id such that $m \neq m'$, then at least one correct process p_i $k\ell$-delivers m with identity id.*

Proof. Let us assume that no correct process $k\ell$-casts (m', id) with $m' \neq m$. No correct process therefore ur-broadcasts $\text{MSG}(m', id)$ with $m' \neq m$ at line 3. Lemma 2 also applies and no correct process forwards $\text{MSG}(m', id)$ with $m' \neq m$ at line 7 either, so $k_{NB} = 0$. Because no correct process ur-broadcasts $\text{MSG}(m', id)$ with $m' \neq m$ whether at line 3 or 7, a correct process receives at most t_b imp-messages $\text{MSG}(m', id)$ (all coming from Byzantine processes). As by sf-$k\ell$-Assumption 1, $t_b < q_d$, no correct process $k\ell$-delivers (m', id) with $m' \neq m$ at line 10.

We now prove the contraposition of the Lemma. Let us assume no correct process $k\ell$-delivers (m, id). Since, by our earlier observations, no correct process $k\ell$-delivers (m', id) with $m' \neq m$ either,

the condition at line 9 implies that no correct process ever receives at least q_d MSG(m, id), and therefore $\ell_e = 0$. By Lemma 1 we have $c(q_f - 1) \geq (k_U + k_F)(c - t_m - q_d + q_f)$. sf-$k\ell$-Assumption 1 implies that $c - t_m - q_d \geq 0 \iff c - t_m - q_d + q_f > 0$ (as $q_f \geq t_b + 1 \geq 1$), leading to $k_U + k_F \leq \frac{c(q_f-1)}{c-t_m-q_d+q_f}$. Because of the condition at line 2, a correct process p_j that has $k\ell$-cast (m, id) but has not ur-broadcast MSG(m, id) at line 3 has necessarily ur-broadcast MSG(m, id) at line 7. We therefore have $k_I \leq k_U + k_F$, which gives $k_I \leq \frac{c(q_f-1)}{c-t_m-q_d+q_f}$. By contraposition, if $k_I > \frac{c(q_f-1)}{c-t_m-q_d+q_f}$, then at least one correct process must $k\ell$-deliver (m, id). Hence, we have $k = \left\lfloor \frac{c(q_f-1)}{c-t_m-q_d+q_f} \right\rfloor + 1$. □

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 imp-messages MSG(m, id), and therefore did not fulfill the condition at line 6 when it received its $q_f{}^{\text{th}}$ imp-message MSG(m, id). As $single = \texttt{false}$ by Lemma assumption, to falsify this condition, p_i must have had already ur-broadcast MSG(m, id) when this happened. Because p_i never executes line 7, this implies that p_i ur-broadcasts MSG(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 $k\ell$-delivers (m, id) and $x = k_U + k_F$ (the number of correct processes that ur-broadcast* MSG(m, id) *at line* 3 *or* 7*), then $x \geq q_d - t_b$ and $x^2 - x(c - t_m + q_f - 1 - k_{NB}) \geq -(c - k_{NB})(q_f - 1)$.*

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

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

This proves the first inequality of the lemma. The processes in $A \cup B$ each receive at most $k_U + k_F$ distinct MSG(m, id) imp-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:

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

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

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

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:

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

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

Proof. By Lemma 5 we have:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Let us rewrite the inequality as follows:

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

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

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

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

We now know that $r_0 \leq q_d - 1 - t_b$ and that $r_1 > q_d - 1$. In addition, as $x \leq r_0 \vee x \geq r_1$, we have $x \leq q_d - t_b - 1 \vee x > q_d - 1$. But Lemma 5 states that $x \geq q_d - t_b$, which is incompatible with $x \leq q_d - t_b - 1$. 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 $c = n - t_b$.

Let us now consider the set E_0 of all executions in which t_b processes are Byzantine, and therefore $c = n - t_b$, and a set E_c of executions in which there are fewer Byzantine processes, and thus $c > n - t_b$ 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 $c > n - t_b$ can also happen when $c = n - t_b$. Thus our result proven for $c = n - t_b$ 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 $\left\lceil c\left(1 - \frac{t_m}{c - q_d + 1}\right)\right\rceil$ correct processes $k\ell$-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:

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

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:

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

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

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

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:

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

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:

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

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: $\ell_e \geq \phi + \frac{\phi\gamma-\beta}{c-\gamma} = \frac{\phi c-\beta}{c-\gamma} \geq \frac{c(c-t_m-q_d+q_f)-c(q_f-1)}{c-q_d+1} \geq c \times \frac{c-t_m-q_d+1}{c-q_d+1} = c\left(1 - \frac{t_m}{c-q_d+1}\right)$.

Since ℓ_e is a positive integer, we conclude that at least $\ell_{\mathsf{min}} = \left\lceil c\left(1 - \frac{t_m}{c-q_d+1}\right)\right\rceil$ correct processes receive at least q_d imp-message $\text{MSG}(m, id)$ at line 9. As each of these processes either $k\ell$-delivers (m, id) when this first happens, or has already $k\ell$-delivered another app-message $m' \neq m$ with identity id, we conclude that at least ℓ_{min} correct processes $k\ell$-deliver some app-message (whether it be m or $m' \neq m$) with identity id when $c = n - t_b$. The reasoning for extending this result to any value of $c \in [n - t_b, n]$ is identical to the one at the end of the proof of Lemma 6 just above. □

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

Proof. Let us assume *single* $=$ `false`, and one correct process $k\ell$-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 $\ell = \left\lceil c\left(1 - \frac{t_m}{c-q_d+1}\right)\right\rceil$ correct processes $k\ell$-deliver an app-message m' with identity id, which concludes the proof of the lemma. □

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

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

By hypothesis, no correct process ur-broadcasts $\text{MSG}(m', id)$ at line 3 with $m' \neq m$. Similarly, because of Lemma 2, no correct process ur-broadcasts $\text{MSG}(m', id)$ at line 7 with $m' \neq m$. As a result, a correct process can receive at most receive t_b imp-messages $\text{MSG}(m', id)$ at line 9 (all from Byzantine processes). As $q_d > t_b$ (by sf-$k\ell$-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 $k\ell$-deliver an app-message with identity id therefore $k\ell$-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 revisited algorithm

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

In Algorithm 2 (page 12), we instantiate the $k\ell$-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 $k\ell$-cast ($k\ell$-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:

- $obj_{\text{E}}.k' = obj_{\text{E}}.q_f - n + c = t_b + 1 - n + c \geq t_b + 1 - t_b = 1$,
- $obj_{\text{E}}.k = \left\lfloor \dfrac{c(obj_{\text{E}}.q_f - 1)}{c - t_m - obj_{\text{E}}.q_d + obj_{\text{E}}.q_f} \right\rfloor + 1 = \left\lfloor \dfrac{c(t_b + 1 - 1)}{c - t_m - \lfloor \frac{n+t_b}{2} \rfloor - 1 + t_b + 1} \right\rfloor + 1$
  $= \left\lfloor \dfrac{ct_b}{c - t_m - \lfloor \frac{n-t_b}{2} \rfloor} \right\rfloor + 1$,
- $obj_{\text{E}}.\ell = \left\lceil c\left(1 - \dfrac{t_m}{c - obj_{\text{E}}.q_d + 1}\right) \right\rceil = \left\lceil c\left(1 - \dfrac{t_m}{c - \lfloor \frac{n+t_b}{2} \rfloor - 1 + 1}\right) \right\rceil$
  $= \left\lceil c\left(1 - \dfrac{t_m}{c - \lfloor \frac{n+t_b}{2} \rfloor}\right) \right\rceil$,
- $obj_{\text{E}}.\delta = \left(\left(obj_{\text{E}}.q_f > \dfrac{n+t_b}{2}\right) \vee \left(obj_{\text{E}}.single \wedge obj_{\text{E}}.q_d > \dfrac{n+t_b}{2}\right)\right)$
  $= \left(\left(t_b + 1 > \dfrac{n+t_b}{2}\right) \vee \left(\texttt{true} \wedge \left\lfloor \dfrac{n+t_b}{2} \right\rfloor + 1 > \dfrac{n+t_b}{2}\right)\right)$
  $= (\texttt{false} \vee (\texttt{true} \wedge \texttt{true})) = \texttt{true}$,
- $obj_{\text{R}}.k' = obj_{\text{R}}.q_f - n + c = t_b + 1 - n + c \geq t_b + 1 - t_b = 1$,
- $obj_{\text{R}}.k = \left\lfloor \dfrac{c(obj_{\text{R}}.q_f - 1)}{c - t_m - obj_{\text{R}}.q_d + obj_{\text{R}}.q_f} \right\rfloor + 1 = \left\lfloor \dfrac{c(t_b + 1 - 1)}{c - t_m - 2t_b - t_m - 1 + t_b + 1} \right\rfloor + 1$
  $= \left\lfloor \dfrac{ct_b}{c - 2t_m - t_b} \right\rfloor + 1$,
- $obj_{\text{R}}.\ell = \left\lceil c\left(1 - \dfrac{t_m}{c - obj_{\text{R}}.q_d + 1}\right) \right\rceil = \left\lceil c\left(1 - \dfrac{t_m}{c - 2t_b - t_m - 1 + 1}\right) \right\rceil$
  $= \left\lceil c\left(1 - \dfrac{t_m}{c - 2t_b - t_m}\right) \right\rceil$,
- $obj_{\text{R}}.\delta = \left(\left(obj_{\text{R}}.q_f > \dfrac{n+t_b}{2}\right) \vee \left(obj_{\text{R}}.single \wedge obj_{\text{R}}.q_d > \dfrac{n+t_b}{2}\right)\right)$
  $= \left(\left(t_b + 1 > \dfrac{n+t_b}{2}\right) \vee \left(\texttt{true} \wedge 2t_b + t_m + 1 > \dfrac{n+t_b}{2}\right)\right) \in \{\texttt{true}, \texttt{false}\}$

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_b, and t_m. This is fine because, in Bracha's revisited algorithm (Algorithm 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 Algorithm 2, sf-$k\ell$-Assumption 1 through 4 are all satisfied by B87-Assumption $n > 3t_b + 2t_m + 2\sqrt{t_b t_m}$. We prove this fact in Appendix 14. In the following, we prove that $3t_b + 2t_m + 2\sqrt{t_b t_m} \geq 2t_b + t_m + \sqrt{t_b^2 + 6t_b t_m + t_m^2} \geq 3t_b + 2t_m$.

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

$$3t_b + 2t_m + 2\sqrt{t_b t_m} \geq 2t_b + t_m + \sqrt{t_b^2 + 6t_b t_m + t_m^2} \geq 3t_b + 2t_m.$$

Proof. Let us start by proving the first inequality.

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

Let us then prove the second inequality:

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

□

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 $k\ell$-cast implementation presented in Algorithm 1 (page 6) are well respected for the two $k\ell$-cast instances used in Algorithm 2 (obj_{E} and obj_{R}).

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

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

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

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

  We also have:

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

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned}&(n - t_m)\left(\frac{n + t_b - \epsilon}{2}\right) - t_b(n - t_b) - \left(\frac{n + t_b - \epsilon}{2}\right)^2, \\ &= \frac{n + t_b - \epsilon}{2} \times \frac{2n - 2t_m - n - t_b + \epsilon}{2} - t_b(n - t_b), \\ &= \frac{(n + t_b - \epsilon)(n - 2t_m - t_b + \epsilon) - 4t_b(n - t_b)}{4}, \\ &= \frac{n^2 - t_b^2 - 2t_b t_m + 2t_b\epsilon - 2nt_m + 2t_m\epsilon - \epsilon^2 - 4nt_b + 4t_b^2}{4}, \\ &= \frac{n^2 + 3t_b^2 - 2t_b t_m - 2n(t_m + 2t_b) + \epsilon(2t_b + 2t_m - \epsilon)}{4}.\end{aligned}$$

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

$$n^2 + 3t_b^2 - 2t_b t_m - 2n(t_m + 2t_b) + \epsilon(2t_b + 2t_m - \epsilon) > 0. \qquad (15)$$

  We now solve the inequality to obtain:

$$n > 2t_b + t_m + \sqrt{t_b^2 + 6t_b t_m + t_m^2 - \epsilon(2t_b + 2t_m - \epsilon)}.$$

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

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

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

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

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

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

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

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

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

$$\begin{aligned} & (n - t_m)\left(\frac{n + t_b - \epsilon}{2} - t_b\right) - t_b(n - t_b) - \left(\frac{n + t_b - \epsilon}{2} - t_b\right)^2, \\ &= (n - t_m) \cdot \frac{n - t_b - \epsilon}{2} - t_b(n - t_b) - \left(\frac{n - t_b - \epsilon}{2}\right)^2, \\ &= \frac{n - t_b - \epsilon}{2} \cdot \frac{2n - 2t_m - n + t_b + \epsilon}{2} - t_b(n - t_b), \\ &= \frac{(n - t_b - \epsilon)(n - 2t_m + t_b + \epsilon) - 4nt_b + 4t_b^2}{4}, \\ &= \frac{-t_b^2 + 2t_b t_m - 2t_b\epsilon + 2t_m\epsilon - 2t_m n - \epsilon^2 + n^2}{4}. \end{aligned}$$

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

$$-t_b^2 + 2t_b t_m - 2t_b\epsilon + 2t_m\epsilon - \epsilon^2 - 2t_m n + n^2.$$

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

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

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

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

  From Observation 1, we have:

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

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

  Let us recall that, for object obj_{R}, we have $q_f = t_b + 1$ and $q_d = 2t_b + t_m + 1$. As sf-$k\ell$-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-$k\ell$-Assumption* 3 ($\alpha(obj_{\mathrm{R}}.q_d - 1) - (obj_{\mathrm{R}}.q_f - 1)(n - t_b) - (obj_{\mathrm{R}}.q_d - 1)^2 > 0$):

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

  $$\begin{aligned}
  &\alpha(q_d - 1) - (q_f - 1)(n - t_b) - (q_d - 1)^2,\\
  &= (n + q_f - t_b - t_m - 1)(q_d - 1) - (q_f - 1)(n - t_b) - (q_d - 1)^2,\\
  &= (n - t_m)(2t_b + t_m) - t_b(n - t_b) - (2t_b + t_m)^2,\\
  &= 2nt_b + nt_m - 2t_bt_m - t_m^2 - nt_b + t_b^2 - 4t_b^2 - t_m^2 - 4t_bt_m,\\
  &= n(t_b + t_m) - 6t_bt_m - 2t_m^2 - 3t_b^2,\\
  &= n(t_b + t_m) - (6t_bt_m + 2t_m^2 + 3t_b^2). \qquad (17)
  \end{aligned}$$

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

  $$\begin{aligned}
  &n(t_b + t_m) - (3t_b^2 + 6t_bt_m + 2t_m^2),\\
  &> (2t_b + t_m + \sqrt{\rho})(t_b + t_m) - (3t_b^2 + 6t_bt_m + 2t_m^2),\\
  &= \sqrt{\rho}(t_b + t_m) - t_m^2 - t_b^2 - 3t_bt_m.
  \end{aligned}$$

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

  $$\sqrt{\rho}(t_b + t_m) - t_m^2 - t_b^2 - 3t_bt_m \geq 0. \qquad (18)$$

  We now rewrite (18) as follows:

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

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

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

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

$$\begin{aligned}
&\alpha(q_d - 1 - t_b) - (q_f - 1)(n - t_b) - (q_d - 1 - t_b)^2, && (19)\\
&= (n + q_f - t_b - t_m - 1)(q_d - 1 - t_b) - (q_f - 1)(n - t_b) - (q_d - 1 - t_b)^2,\\
&= (n + -t_m)(t_b + t_m) - t_b(n - t_b) - (t_b + t_m)^2,\\
&= (t_b + t_m)(n + -2t_m - t_b) - t_b(n - t_b),\\
&= nt_b + nt_m - 2t_bt_m - 2t_m^2 - t_b^2 - t_bt_m - nt_b + t_b^2,\\
&= nt_m - 3t_bt_m - 2t_m^2,\\
&= nt_m - 3t_bt_m - 2t_m^2,\\
&= t_m(n - 3t_b - 2t_m). && (20)
\end{aligned}$$

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

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

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

B.1.3 Correctness proof

This section proves the following theorem:

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

The proof follows from the next lemmas.

Lemma 16. $c - t_m \geq obj_{\mathrm{E}}.k$.

Proof. We want to show that:

$$c - t_m \geq \left\lfloor \frac{ct_b}{c - t_m - \lfloor \frac{n - t_b}{2} \rfloor} \right\rfloor + 1 = obj_{\mathrm{E}}.k. \tag{22}$$

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

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

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

$$\begin{aligned}
(c - t_m)(c - t_m - \frac{n - t_b - \epsilon}{2}) &> ct_b,\\
(c - t_m)(2c - 2t_m - n + t_b + \epsilon) &> 2ct_b,\\
(c - t_m)(2c - 2t_m - n + t_b + \epsilon) - 2ct_b &> 0.
\end{aligned}$$

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

$$\begin{aligned}
(n - t_b - t_m)(n - t_b - 2t_m + \epsilon) - 2(n - t_b)t_b > 0,\\
(n - t_b)(n - t_b - 2t_m + \epsilon - 2t_b) - t_m(n - t_b - 2t_m + \epsilon) > 0,\\
(n - t_b)^2 + (n - t_b)(-2t_m + \epsilon - 2t_b) - t_m(n - t_b - 2t_m + \epsilon) > 0,\\
n^2 + t_b^2 - 2nt_b - 2nt_m + n\epsilon - 2nt_b + 2t_bt_m - t_b\epsilon + 2t_b^2 - nt_m + t_bt_m + 2t_m^2 - \epsilon t_m > 0,\\
n^2 + 3t_b^2 - 4nt_b - 3nt_m + n\epsilon + 3t_bt_m - t_b\epsilon + 2t_m^2 - \epsilon t_m > 0,\\
n^2 - n(4t_b + 3t_m - \epsilon) + 3t_b^2 + 3t_bt_m + 2t_m^2 - \epsilon(t_b + t_m) > 0.
\end{aligned}$$

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_b and t_m. So we obtain:

$$\begin{aligned}
n > 2t_b + \frac{3t_m}{2} - \frac{\epsilon}{2} + \frac{\sqrt{4t_b^2 + 12t_bt_m - 4t_b\epsilon + t_m^2 - 2t_m\epsilon + \epsilon^2}}{2},\\
-4t_b - 3t_m + \epsilon + 2n - \sqrt{4t_b^2 + 12t_bt_m - 4t_b\epsilon + t_m^2 - 2t_m\epsilon + \epsilon^2} > 0,
\end{aligned}$$

which is implied by the following as $n \geq 3t_b + 2t_m + 2\sqrt{t_bt_m} + 1$:

$$\begin{aligned}
4\sqrt{t_bt_m} + 2t_b + t_m + 2 + \epsilon - \sqrt{4t_b^2 + 12t_bt_m - 4t_b\epsilon + t_m^2 - 2t_m\epsilon + \epsilon^2} > 0,\\
4\sqrt{t_bt_m} + 2t_b + t_m + 2 + \epsilon > \sqrt{4t_b^2 + 12t_bt_m - 4t_b\epsilon + t_m^2 - 2t_m\epsilon + \epsilon^2}.
\end{aligned}$$

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

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

which simplifies to:

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

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:

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

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:

$$\begin{aligned}
c\left(1 - \frac{t_m}{c - \frac{n+t_b-\epsilon}{2}}\right) \geq \frac{ct_b}{c - t_b - 2t_m},\\
c - \frac{2t_mc}{2c - t_b - n + \epsilon} > \frac{ct_b}{c - t_b - 2t_m}.
\end{aligned}$$

As both denominators are positive, we can solve:

$$
\begin{aligned}
&-t_b\left(-t_b+2c+\epsilon-n\right)-2t_m\left(-t_b-2t_m+c\right)+\left(-t_b-2t_m+c\right)\left(-t_b+2c+\epsilon-n\right)>0,\\
&-t_b\left(-t_b+2c+\epsilon-n\right)+\left(-t_b-2t_m+c\right)\left(-t_b-2t_m+2c+\epsilon-n\right)>0,\\
&-t_b\left(-2t_b+c+\epsilon\right)+\left(-t_b-2t_m+c\right)\left(-2t_b-2t_m+c+\epsilon\right)>0, && \text{(as } c\geq n-t_b\text{)}\\
&-t_b\left(-2t_b+c+\epsilon\right)+\left(-t_b-2t_m+c\right)\left(-2t_b-2t_m+c+\epsilon\right)>0,\\
&-t_b\left(-t_b+2c-n\right)+\epsilon\left(-3t_b-2t_m+c\right)+\left(-t_b-2t_m+c\right)^2>0,\\
&t_b^2-2t_bc+t_bn+\epsilon\left(-3t_b-2t_m+c\right)+\left(-t_b-2t_m+c\right)^2>0,\\
&t_b^2-2t_bc+t_b\left(2\sqrt{t_b}\sqrt{t_m}+3t_b+2t_m+1\right)+\epsilon\left(-3t_b-2t_m+c\right)+\left(-t_b-2t_m+c\right)^2>0,\\
& && \text{(as } n\geq 3t_b+2t_m+2\sqrt{t_bt_m}\text{)}\\
&2t_b^{\frac{3}{2}}\sqrt{t_m}+4t_b^2+2t_bt_m-2t_bc+t_b+\epsilon\left(-3t_b-2t_m+c\right)+\left(-t_b-2t_m+c\right)^2>0,\\
&2t_b^{\frac{3}{2}}\sqrt{t_m}+5t_b^2+6t_bt_m-4t_bc+t_b+4t_m^2-4t_mc+c^2+\epsilon\left(-3t_b-2t_m+c\right)>0.
\end{aligned}
$$

We now consider the two possible values of ϵ:

- $\epsilon=0$:

$$2t_b^{\frac{3}{2}}\sqrt{t_m}+5t_b^2+6t_bt_m-4t_bc+t_b+4t_m^2-4t_mc+c^2>0 \tag{26}$$

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

$$c>2t_b+2t_m+\sqrt{-2t_b^{\frac{3}{2}}\sqrt{t_m}-t_b^2+2t_bt_m-t_b}$$

  which we prove by observing that $c\geq n-t_b\geq 2t_b+2t_m+2\sqrt{t_bt_m}+1$ and that:

$$2t_b+2t_m+2\sqrt{t_bt_m}+1>2t_b+2t_m+\sqrt{-2t_b^{\frac{3}{2}}\sqrt{t_m}-t_b^2+2t_bt_m-t_b},$$

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

- $\epsilon=1$:

  In this case we obtain:

$$2t_b^{\frac{3}{2}}\sqrt{t_m}+5t_b^2+6t_bt_m-4t_bc-2t_b+4t_m^2-4t_mc-2t_m+c^2+c>0,$$

  which is implied by a negative discriminant or by:

$$c>2t_b+2t_m+\sqrt{-2t_b^{\frac{3}{2}}\sqrt{t_m}-t_b^2+2t_bt_m+1/4}-\frac{1}{2}.$$

  Like before we simply observe that:

$$
\begin{aligned}
2\sqrt{t_bt_m}&\geq\sqrt{2t_bt_m}+\frac{1}{2}-\frac{1}{2},\\
&\geq\sqrt{2t_bt_m+1/4}-\frac{1}{2},\\
&\geq\sqrt{-2t_b^{\frac{3}{2}}\sqrt{t_m}-t_b^2+2t_bt_m+1/4}-\frac{1}{2},
\end{aligned}
$$

  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 $k\ell$-delivered $(\text{READY}(m), (sn, j))$ using obj_{R}. From $k\ell$-VALIDITY, and as $obj_{\text{R}}.k' = 1$, we can assert that at least one correct process p_x $k\ell$-cast $(\text{READY}(m), (sn, j))$ at line 3, after having $k\ell$-delivered $(\text{ECHO}(m), (sn, j))$ using obj_{E}. Again, from $k\ell$-VALIDITY, we can assert that at least $obj_{\text{E}}.k' = 1$ correct process p_y $k\ell$-cast $(\text{ECHO}(m), (sn, j))$ at line 2, after having received an $\text{INIT}(m, sn)$ imp-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 $k\ell$-NO-DUPLICATION, we know that a correct process p_i can $k\ell$-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 $k\ell$-delivered $(\text{READY}(m), (sn, i))$ and $(\text{READY}(m'), (sn, i))$ using obj_{R}.

From $k\ell$-VALIDITY, and as $obj_{\text{R}}.k' \geq 1$, we can assert that two correct processes p_y and p_z respectively $k\ell$-cast $(\text{READY}(m), (sn, i))$ and $(\text{READY}(m'), (sn, i))$ at line 3, after having respectively $k\ell$-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 $k\ell$-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 imp-message $\text{INIT}(m, sn)$ is then received by at least $c - t_m$ correct processes at line 2, which then $k\ell$-cast $(\text{ECHO}(m), sn, i)$. As p_i is correct and ur-broadcasts only one imp-message $\text{INIT}(-, sn)$, then no correct process $k\ell$-casts any different $(\text{ECHO}(-), sn, i)$. Moreover, thanks to Lemma 16, we know that:

$$c - t_m \geq obj_{\text{E}}.k = \left\lfloor \frac{ct_b}{c - t_m - \lfloor \frac{n-t_b}{2} \rfloor} \right\rfloor + 1.$$

Hence, from $k\ell$-LOCAL-DELIVERY and $k\ell$-STRONG-GLOBAL-DELIVERY, at least $obj_{\text{E}}.\ell = \left\lceil c\left(1 - \frac{t_m}{c - \lfloor \frac{n+t_b}{2} \rfloor}\right)\right\rceil$ correct processes eventually $k\ell$-deliver $(\text{ECHO}(m), (sn, i))$ using obj_{E} and then $k\ell$-cast $(\text{READY}(m), (sn, i))$ using obj_{R} at line 3. By $k\ell$-VALIDITY, and as $obj_{\text{R}}.k' \geq 1$, then no correct process can $k\ell$-cast a different $(\text{READY}(-), (sn, i))$, because otherwise it would mean that at least one correct process would have $k\ell$-cast a different $(\text{ECHO}(-), (sn, i))$, which is impossible (see before). Moreover, thanks to Lemma 17, we know that:

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

Therefore, $k\ell$-LOCAL-DELIVERY applies and we know that at least one correct processes eventually $k\ell$-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 $\ell_{MBRB} = \left\lceil c\left(1 - \frac{t_m}{c-2t_b-t_m}\right)\right\rceil$ 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 $k\ell$-delivered $(\text{READY}(m), (sn, j))$ using obj_{R}. From $k\ell$-VALIDITY, we know that at least $obj_{\text{R}}.k' \geq 1$ correct process $k\ell$-cast $(\text{READY}(m), (sn, j))$ using obj_{R} at line 3 and thus $k\ell$-delivered $(\text{ECHO}(m), (sn, j))$ using obj_{E}. From $k\ell$-CONDITIONAL-NO-DUPLICITY, and as $obj_{\text{E}}.\delta = \texttt{true}$, we can state that no correct process $k\ell$-delivers any $(\text{ECHO}(m'), (sn, j))$ where $m' \neq m$ using obj_{E}, so no correct process $k\ell$-casts any $(\text{READY}(m'), (sn, j))$ where $m' \neq m$ using obj_{R} at line 3. It means that $k\ell$-STRONG-GLOBAL-DELIVERY applies, and we can assert that at least $obj_{\text{R}}.\ell = \left\lceil c\left(1 - \frac{t_m}{c-2t_b-t_m}\right)\right\rceil = \ell_{MBRB}$ correct processes eventually $k\ell$-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 revisited algorithm

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

In Algorithm 3 (page 12), we instantiate the $k\ell$-cast object obj_{W} using the signature-free implementation presented in Section 3.2 with parameters $q_d = \left\lfloor \frac{n+3t_b}{2} \right\rfloor + 3t_m + 1$, $q_f = \left\lfloor \frac{n+t_b}{2} \right\rfloor + 1$, and $single = \texttt{false}$. Based on Theorem 1 (page 7), these parameters lead to the following values for k', k, ℓ and δ.

- $obj_{\text{W}}.k' = obj_{\text{W}}.q_f - n + c = \left\lfloor \frac{n+t_b}{2} \right\rfloor + 1 - n + c$
  $\geq \left\lfloor \frac{n+t_b}{2} \right\rfloor + 1 - n + n - t_b = \left\lfloor \frac{n-t_b}{2} \right\rfloor + 1,$
- $obj_{\text{W}}.k = \left\lfloor \frac{c(obj_{\text{W}}.q_f - 1)}{c - t_m - obj_{\text{W}}.q_d + obj_{\text{W}}.q_f} \right\rfloor + 1$
  $= \left\lfloor \frac{c(\left\lfloor \frac{n+t_b}{2} \right\rfloor + 1 - 1)}{c - t_m - (\left\lfloor \frac{n+3t_b}{2} \right\rfloor + 3t_m + 1) + \left\lfloor \frac{n+t_b}{2} \right\rfloor + 1} \right\rfloor + 1$
  $= \left\lfloor \frac{c\left\lfloor \frac{n+t_b}{2} \right\rfloor}{c - t_b - 4t_m} \right\rfloor + 1,$
- $obj_{\text{W}}.\ell = \left\lceil c\left(1 - \frac{t_m}{c - obj_{\text{E}}.q_d + 1}\right)\right\rceil = \left\lceil c\left(1 - \frac{t_m}{c - (\left\lfloor \frac{n+3t_b}{2} \right\rfloor + 3t_m + 1) + 1}\right)\right\rceil$
  $= \left\lceil c\left(1 - \frac{t_m}{c - \left\lfloor \frac{n+3t_b}{2} \right\rfloor - 3t_m}\right)\right\rceil,$
- $obj_{\text{W}}.\delta = \left(\left(obj_{\text{W}}.q_f > \frac{n+t_b}{2}\right) \vee \left(obj_{\text{W}}.single \wedge obj_{\text{W}}.q_d > \frac{n+t_b}{2}\right)\right)$
  $= \left(\left(\left\lfloor \frac{n+t_b}{2} \right\rfloor + 1 > \frac{n+t_b}{2}\right) \vee \left(\texttt{false} \wedge \left\lfloor \frac{n+3t_b}{2} \right\rfloor + 3t_m + 1 > \frac{n+t_b}{2}\right)\right)$
  $= (\texttt{true} \vee (\texttt{false} \wedge \texttt{true})) = \texttt{true}.$

Finally, we observe that for Algorithm 3, sf-$k\ell$-Assumption 1 through 4 are all satisfied by IR16-Assumption ($n > 5t_b + 12t_m + \frac{2t_b t_m}{t_b + 2t_m}$), 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 $k\ell$-cast implementation presented in Algorithm 1 (page 6) are well respected for the $k\ell$-cast instance used in Algorithm 3 (obj_{W}).

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

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

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

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

  We also have:

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

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

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

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

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

  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:

$$\begin{aligned}
&\left(\frac{3n-t_b-\epsilon}{2}-t_m\right)^2-4\frac{n+t_b-\epsilon}{2}(n-t_b),\\
&=\left(\frac{3n-t_b-\epsilon-2t_m}{2}\right)^2-4\frac{n+t_b-\epsilon}{2}(n-t_b),\\
&=\frac{t_b^2+4t_bt_m+2t_b\epsilon-6t_bn+4t_m^2+4t_m\epsilon-12t_mn+\epsilon^2-6\epsilon n+9n^2}{4}\\
&\quad+\frac{8t_b^2-8t_b\epsilon+8\epsilon n-8n^2}{4},\\
&=\frac{9t_b^2+4t_bt_m-6t_b\epsilon-6t_bn+4t_m^2+4t_m\epsilon-12t_mn+\epsilon^2+2\epsilon n+n^2}{4},\\
&=\frac{9t_b^2-6t_bn+n^2+4t_bt_m-12t_mn+4t_m^2+4t_m\epsilon-6t_b\epsilon+\epsilon^2+2\epsilon n}{4},\\
&=\frac{(n-3t_b)^2+4t_m(t_b-3n+t_m)+\epsilon(4t_m-6t_b+\epsilon+2n)}{4}.
\end{aligned}$$

We now multiply by 4 and solve the inequality:

$$\begin{aligned}
&n^2-6n(t_b+2t_m)+9t_b^2+4t_bt_m+4t_m^2+\epsilon\left(-6t_b+4t_m+\epsilon+2n\right)\geq 0,\\
&n\geq 3t_b+4\sqrt{t_m}\sqrt{2t_b+2t_m-\epsilon}+6t_m-\epsilon. \qquad (30)
\end{aligned}$$

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

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

We observe that (31) holds as $144t_m^4\geq 64t_m^3$, $288t_bt_m^3\geq 96t_bt_m^2$, and $192t_b^2t_m^2\geq 32t_b^2t_m$, therefore proving sf-$k\ell$-Assumption 2.

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

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

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

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:

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

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

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

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

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

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

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

  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:

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

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

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

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

B.2.3 Correctness proof

This section proves the following theorem:

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

The proof follows from the next lemmas.

Lemma 24. $c - t_m \geq obj_{\mathrm{W}}.k$.

Proof. This proof is presented in reverse order: we start from the result we want to prove, and we 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:

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

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

$$
\left[2t_b + 5t_m - \sqrt{4t_b^2 + 18t_b t_m + 17t_m^2}, 2t_b + 5t_m + \sqrt{4t_b^2 + 18t_b t_m + 17t_m^2}\right].
$$

We now need to show that:

$$
c > 2t_b + 5t_m + \sqrt{4t_b^2 + 18t_b t_m + 17t_m^2}.
$$

By IR16-Assumption, we know that:

$$
n \geq 5t_b + 12t_m + \frac{2t_b t_m}{t_b + 2t_m} + 1,
$$

and thus that:

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

So we want to show that:

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

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

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

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 $k\ell$-delivered $(\text{WITNESS}(m), (sn, j))$ using obj_{W}. From $k\ell$-VALIDITY, and as $obj_{\text{R}}.k' \geq 1$, we can assert that at least one correct process p'_i $k\ell$-cast $(\text{WITNESS}(m), (sn, j))$ at line 2, after having received an $\text{INIT}(m, sn)$ imp-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 $k\ell$-NO-DUPLICATION, we know that a correct process p_i can $k\ell$-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 different 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 $k\ell$-CONDITIONAL-NO-DUPLICITY, we know that no two correct processes can $k\ell$-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 imp-message $\text{INIT}(m, sn)$ is then received by at least $c - t_m$ correct processes at line 2, which then $k\ell$-cast $(\text{WITNESS}(m), sn, i)$. But thanks to Lemma 24, we know that:

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

As p_i is correct and ur-broadcasts only one imp-message $\text{INIT}(-, sn)$, then no correct process $k\ell$-casts any different $(\text{WITNESS}(-), sn, i)$, $k\ell$-LOCAL-DELIVERY applies and at least one correct processes eventually $k\ell$-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 $\ell_{MBRB} = \left\lceil c\left(1 - \frac{t_m}{c - \left\lfloor \frac{n+3t_b}{2} \right\rfloor - 3t_m}\right)\right\rceil$ 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 $k\ell$-delivered $(\text{WITNESS}(m), (sn, j))$ using obj_{W}. As $obj_{\text{W}}.\delta =$ `true`, we can assert from $k\ell$-WEAK-GLOBAL-DELIVERY and $k\ell$-CONDITIONAL-NO-DUPLICITY that at least $obj_{\text{W}}.\ell = \left\lceil c\left(1 - \frac{t_m}{c - q_d + 1}\right)\right\rceil$ correct processes eventually $k\ell$-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 $obj_{\text{W}}.\ell = \left\lceil c\left(1 - \frac{t_m}{c - \left\lfloor \frac{n+3t_b}{2} \right\rfloor - 3t_m}\right)\right\rceil = \ell_{MBRB}$ thus proving the lemma. □

C Proof of the Signature-Based $k\ell$-cast Implementation

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 $n - t_b$. 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 Algorithm 4:

- sb-$k\ell$-Assumption 1: $c > 2t_m$,
- sb-$k\ell$-Assumption 2: $c - t_m \geq q_d \geq t_b + 1$.

C.1 Safety Proof

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

Proof. If p_i $k\ell$-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 $0 \leq n - c \leq t_b$. Therefore, p_i must have sent at least $q_d - n + c$ (which, due to sb-$k\ell$-Assumption 2, is strictly positive because $q_d > t_b \geq n - c$) valid distinct signatures for (m, id) that correct processes made at line 3, during a $k\ell$_cast(m, id) invocation. □

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

Proof. The condition at line 2 implies that the correct processes that $k\ell$-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 ($k\ell$-NO-DUPLICATION)**.** *A correct process $k\ell$-delivers at most one app-message m with identity id.*

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

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

Proof. Let p_i and p_j be two correct processes that respectively $k\ell$-deliver (m, id) and (m', id). We want to prove that, if the predicate $(q_d > \frac{n+t_b}{2})$ 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 $|A| \geq q_d > \frac{n+t_b}{2}$ and $|B| \geq q_d > \frac{n+t_b}{2}$. We have $|A \cap B| > 2\frac{n+t_b}{2} - n = t_b$. 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 $k\ell$_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 $c - t_m$ 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 $c - t_m$ 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 $c - t_m$ correct processes that we denote A. We now show that the proposition is verified at $z + 1$: at least $c - t_m$ 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$(m, id, sigs)$ imp-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 $c - t_m$ correct processes that we denote B. We have $|A \cap B| = 2(c - t_m) - c = c - 2t_m > 2t_m - 2t_m = 0$ (from sb-$k\ell$-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 $c - t_m$ correct processes at line 8. □

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

Proof. Looking for a contradiction, let us suppose that a correct process p_i $k\ell$-delivers (m, id) while no correct process $k\ell$-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_b are made by Byzantine processes. As $q_d > t_b$ (sb-$k\ell$-Assumption 2), we know that $q_d - t_b > 0$. Hence, at least one correct process must have $k\ell$-cast (m, id). Contradiction. □

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

Proof. As no correct process $k\ell$-casts an app-message $m' \neq m$ with identity id, then Lemma 35 holds, and no correct process can $k\ell$-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 $k\ell$_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 $c - t_m$ 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 $k\ell$-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 $k\ell$-delivered (m, id) at line 16. □

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

Proof. If p_i $k\ell$-delivers (m, id) at line 16, then it has necessarily ur-broadcast the BUNDLE$(m, id, sigs)$ imp-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 $c - t_m$ correct processes, that we denote A, eventually receives this BUNDLE$(m, id, sigs)$ imp-message at line 8. If some processes of A do not pass the condition at line 9 upon receiving this BUNDLE$(m, id, sigs)$ imp-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 $c - t_m = \ell$, $k\ell$-deliver some app-message for identity id at line 16, whether it be m or any other app-message. □

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

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

D Necessary and Sufficient Condition for MBR-Broadcast

This section shows that $n > 3t_b + 2t_m$ is necessary and sufficient to build BRB algorithm in the presence of an MA (i.e. MBRB). Intuitively, the constraint $n \geq 3t_b$ comes from the Byzantine processes while the constraint $n \geq 2t_m$ comes from the MA (this constraint prevents the MA from partitioning the system). Their "addition" $n > 3t_b + 2t_m$ comes from the fact Byzantine failures and MA are not reducible to each other.

Definition. An algorithm implementing a broadcast communication abstraction is *event-driven* if, as far as the correct processes are concerned, only (i) the invocation of the broadcast operation that is built or (ii) the reception of an imp-message–sent by a correct or a Byzantine process– can generate the sending of imp-messages (realized with the underlying unreliable ur_broadcast operation).

Theorem 5 (MBRB-Necessary-condition)**.** *When $n \leq 3t_b + 2t_m$, there is no event-driven algorithm implementing the* MBR-broadcast *communication abstraction on top of an n-process asynchronous system in which up to t_b processes may be Byzantine and where an MA may suppress up to t_m copies of each imp-message ur-broadcast by a process.*

Proof. Without loss of generality, the proof considers the case $n = 3t_b + 2t_m$. Let us partition the n processes into five sets Q_1, Q_2, Q_3, D_1, and D_2, such that $|D_1| = |D_2| = t_m$ and $|Q_1| = |Q_2| = |Q_3| = t$.[11] So, when considering the sets Q_1, Q_2, and Q_3, there are executions in which all the processes of either Q_1 or Q_2 or Q_3 can be Byzantine, while the processes of the two other sets are not.

The proof is by contradiction. So, assuming that there is an event-driven algorithm A that builds the MBR-broadcast abstraction for $n = 3t_b + 2t_m$, let us consider an execution E of A in which the processes of Q_1, Q_2, D_1, and Q_2 are not Byzantine while all the processes of Q_3 are Byzantine.

Let us observe that the MA can isolate up to t_m processes by preventing them from receiving any imp-message. Without loss of generality, let us assume that the adversary isolates a set of t_m correct processes not containing the sender of the app-message. As A is event-driven, these t_m isolated processes do not send imp-messages during the execution E of A. As a result, no correct process can expect imp-messages from more than $(n - t_b - t_m)$ different processes without risking being blocked forever. Thanks to the assumption $n = 3t_b + 2t_m$, this translates as "no correct process can expect imp-messages from more than $(2t_b + t_m)$ different processes without risking to be blocked forever".

In the execution E, the (Byzantine) processes of Q_3 simulate the mbrb-broadcast of an app-message such that this app-message appears as being mbrb-broadcast by one of them and is mbrb-delivered as the app-message m to the processes of Q_1 (hence the processes of Q_3 appear, to the processes of Q_1, as if they were correct) and as the app-message $m' \neq m$ to the processes of Q_2 (hence, similarly to

[11] For the case $n < 3t_b + 2t_m$, the partition is such that $\max(|Q_1|, |D_2|) \leq t_m$ and $\max(|Q_1|, |Q_2|, |Q_3|) \leq t_b$.

the previous case, the processes of Q_3 appear to the processes of Q_2 as if they were correct). Let us call m-messages (resp., m'-messages) the imp-messages generated by the event-driven algorithm A that entails the mbrb-delivery of m (resp., m'). Moreover, the execution E is such that:

- concerning the m-messages: the MA suppresses all the m-messages sent to the processes of D_2, and asynchrony delays the reception of all the m-messages sent to Q_2 until some time τ defined below[12]. So, as $|Q_1 \cup D_1 \cup Q_3| = n - t_b - t_m = 2t_b + t_m$, Algorithm A will cause the processes of Q_1 and D_1 to mbrb-deliver m[13].
- concerning the m'-messages: the MA suppresses all the m'-messages sent to the processes of D_1, and the asynchrony delays the reception of all the m'-messages sent to Q_1 until time τ. As previously, as $|Q_2 \cup D_2 \cup Q_3| = n - t_b - t_m = 2t_b + t_m$, Algorithm A will cause the processes of Q_2 and D_2 to mbrb-deliver m'.
- Finally, the time τ occurs after the mbrb-delivery of m by the processes of D_1 and Q_1, and after the mbrb-delivery of m' by the processes of D_2 and Q_2.

It follows that different non-Byzantine processes mbrb-deliver different app-messages for the same mbrb-broadcast (or a fraudulent simulation of it) issued by a Byzantine process (with possibly the help of other Byzantine processes). This contradicts the MBRB-No-Duplicity property, which concludes the proof of the theorem. □

Theorem 6 (MBRB-Failure-Tolerance-Optimality)**.** *The condition $n > 3t_b + 2t_m$ is both necessary and sufficient to build an event-driven* MBRB *algorithm in an n-process asynchronous message-passing system in which up to t_b process are Byzantine and the network is controlled by a t_m-MA.*

Proof. Theorem 5 has shown the the condition $n > 3t_b + 2t_m$ is necessary. The sufficiency comes from the existence of the algorithm presented in [3] which, assuming $n > 3t_b + 2t_m$, builds (with the help of signatures) the MBRB abstraction despite asynchrony, MA, and Byzantine processes. □

E 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 $k\ell$-cast objects used in the revisited Bracha MBRB algorithm (Algorithm 2, page 12). Results were obtained by considering a network with $n = 100$ processes and varying values of t_b and t_m. Fig. 4 and Fig. 5 present the values of k and ℓ for the obj_{E} and obj_{R} of Algorithm 2.

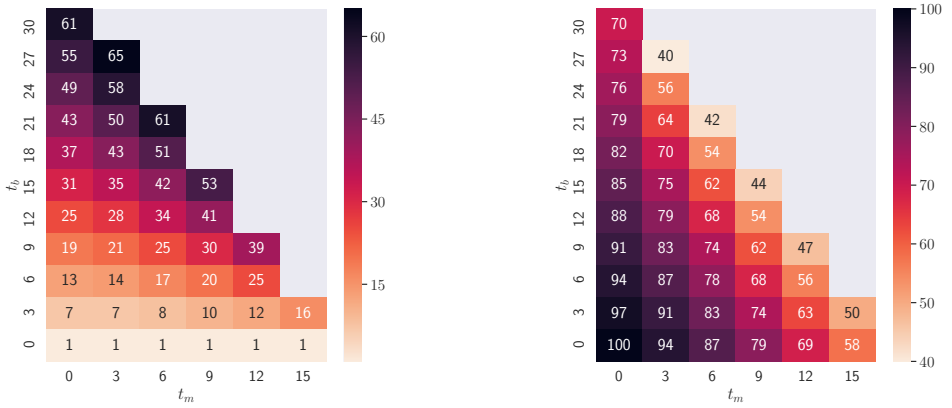
The numbers in each cell show the value of k (Figs. 4a and 5a), resp. ℓ (Figs. 4b and 5b), that is required, resp. guaranteed, by the corresponding $k\ell$-cast object. The two plots show the two different roles of the two $k\ell$-cast objects. The first, obj_{E}, needs to provide agreement among the 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 t_m and t_b, but it needs to operate in conjunction with obj_{E} (Fig. 5).

Figure 3b on page 14 already displays the values of ℓ provided by obj_{W} in the IR algorithm. Figure 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 $k\ell$-cast algorithm could achieve better performance. But this is not the case if we examine the performance of the revisited Imbs-Raynal algorithm depicted in Fig. 3b.

[12] In an equivalent way, we could also say that asynchrony delays the reception of all the m-messages sent to $D_2 \cup Q_2$ until time τ. The important point is here that, due to the assumed existence of Algorithm A, the processes of Q_1 and D_1 mbrb-deliver m with m-messages from at most $2t_b + t_m$ different processes.

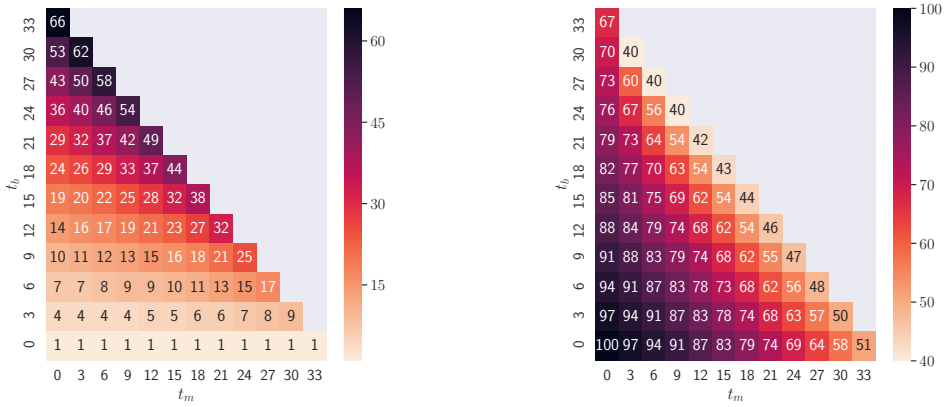
[13] Let us notice that this is independent of the fact that the processes in Q_3 are Byzantine or not.

The reason lies in the need for higher quorum values in obj_{W} due to the fact the $single = \texttt{false}$. In the future, we plan to investigate if variants of this algorithm can achieve tighter bounds and explore the limits of signature-free $k\ell$-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 revisited Bracha BRB algorithm with varying values of t_b and t_m



(a) Minimum required k (b) Minimum provided ℓ

Figure 5: Required values of k and provided values of ℓ for obj_{R} in the revisited Bracha BRB algorithm with varying values of t_b and t_m

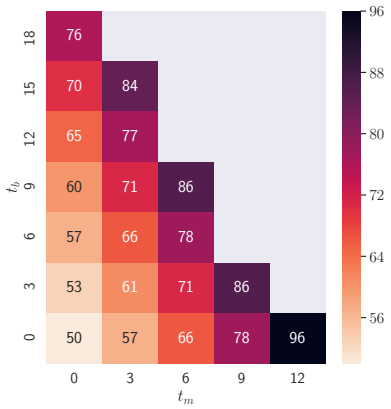


Figure 6: Required values of k for obj_{W} in the revisited Imbs & Raynal BRB algorithm with varying values of t_b and t_m