

OciorCOOL: Faster Byzantine Agreement and Reliable Broadcast

Jinyuan Chen

Abstract

COOL (Chen'21) is an error-free and deterministic Byzantine agreement protocol that achieves consensus on an ℓ -bit message with a communication complexity of $O(\max\{n\ell, nt \log t\})$ bits in four phases, given $n \geq 3t + 1$, for a network of n nodes, where up to t nodes may be dishonest. In this work we show that COOL can be optimized by reducing one communication round. The new protocol is called OciorCOOL. Additionally, building on OciorCOOL, we design an optimal reliable broadcast protocol that requires only six communication rounds.

I. INTRODUCTION

Byzantine agreement (BA) is a fundamental distributed consensus problem introduced around forty years ago [1]. In this problem, n distributed nodes seek to reach consensus on an ℓ -bit message, where up to t of the nodes may be dishonest. Byzantine agreement, together with its variants such as Byzantine broadcast (BB) and reliable broadcast (RBC), is believed to be an essential foundation of distributed systems and cryptography [1]–[13].

For the multi-valued error-free BA problem, significant efforts have been made to improve performance in terms of communication and round complexities, under the optimal resilience condition $n \geq 3t + 1$ [3], [8]–[11] (see Table I). In this direction, Chen designed the COOL protocol that achieves a communication complexity of $O(\max\{n\ell, nt \log t\})$ bits with four phases, under the optimal resilience of $n \geq 3t + 1$ [3]. In this work, we demonstrate that COOL can be optimized by eliminating one phase, thereby reducing the number of communication rounds. With fewer communication rounds, the new protocol, called OciorCOOL, is faster than the original COOL protocol.

COOL has been used as a building block in other consensus problems, such as BB [3], asynchronous BA [5], gradecast [6], validated Byzantine agreement [14], and RBC [15]. In this work, building on OciorCOOL, we design an error-free reliable broadcast protocol called OciorRBC, which requires only six communication rounds and improves upon the RBC protocol by Alhaddad et al. [15], which requires eight communication rounds (see Table II). The proposed OciorCOOL can be applied to other consensus problems, such as BB, asynchronous BA, gradecast, and validated Byzantine agreement to improve the round complexity.

The proposed OciorCOOL protocol is described in Algorithms 1 and 2, and its analysis is provided in Section II. The proposed OciorRBC protocol is described in Algorithm 3, and its analysis is provided in Section III. Table I and Table II provide the comparison between the proposed protocols and some other error-free protocols for the BA and RBC settings, respectively. Some definitions and primitives used in our protocols are provided in the following subsection.

A. Primitives

Information-theoretic (IT) protocol. A protocol that guarantees all of the required properties without using any cryptographic assumptions, such as signatures or hashing, is said to be *information-theoretically secure*. The proposed protocols are information-theoretically secure.

Error-free protocol. A protocol that guarantees all of the required properties in *all* executions is said to be *error-free*. The proposed protocols are error-free.

TABLE I

COMPARISON BETWEEN PROPOSED OCIOORCOOL AND SOME OTHER ERROR-FREE BYZANTINE AGREEMENT PROTOCOLS. $\mathcal{B}_r(1)$ DENOTES THE ROUND COMPLEXITY OF A BINARY BA. BY USING THE BINARY BA PROTOCOL IN [16], [17], WE HAVE $\mathcal{B}_r(1) = O(t)$.

Protocols	Resilience	Communication	Rounds	Error Free	Signature Free
Liang-Vaidya [8]	$n \geq 3t + 1$	$O(n\ell + n^4\sqrt{\ell} + n^6)$	$\Omega(\sqrt{\ell} + n^2)$	Yes	Yes
Ganesh-Patra [9]	$n \geq 3t + 1$	$O(n\ell + n^4)$	$O(t)$	Yes	Yes
Loveless et al. [10]	$n \geq 3t + 1$	$O(n\ell + n^4)$	$O(t)$	Yes	Yes
Nayak et al. [11]	$n \geq 3t + 1$	$O(n\ell + n^3)$	$O(t)$	Yes	Yes
Chen [3]	$n \geq 3t + 1$	$O(\max\{n\ell, nt \log t\})$	$5 + \mathcal{B}_r(1)$	Yes	Yes
OciORCOOL	$n \geq 3t + 1$	$O(\max\{n\ell, nt \log t\})$	$4 + \mathcal{B}_r(1)$	Yes	Yes

TABLE II

COMPARISON BETWEEN PROPOSED OCIOORBC AND SOME OTHER ERROR-FREE RELIABLE BROADCAST PROTOCOLS. “BALANCED COMMUNICATION” MEANS THAT COMMUNICATION OVERHEAD IS DISTRIBUTED EVENLY AMONG DISTRIBUTED NODES.

Protocols	Resilience	Communication	Rounds	Error Free	Signature Free
Bracha [18]	$n \geq 3t + 1$	$O(n^2 w)$	$O(1)$	Yes	Yes
Patra [12]	$n \geq 3t + 1$	$O(n w + n^4 \log n)$	$O(1)$	Yes	Yes
Nayak et al. [11]	$n \geq 3t + 1$	$O(n w + n^3 \log n)$	$O(1)$	Yes	Yes
Alhaddad et al. [15]	$n \geq 3t + 1$	$O(n w + n^2 \log n)$	8 (without balanced com.) 9 (with balanced com.)	Yes	Yes
OciORBC	$n \geq 3t + 1$	$O(n w + n^2 \log n)$	6 (without balanced com.) 7 (with balanced com.)	Yes	Yes

Error correction code (ECC). An (n, k) error correction coding scheme consists of an encoding scheme $\text{ECC}_{\text{Enc}} : \mathcal{B}^k \rightarrow \mathcal{B}^n$ and a decoding scheme $\text{ECC}_{\text{Dec}} : \mathcal{B}^{n'} \rightarrow \mathcal{B}^k$, where $\mathcal{B}$ denotes the alphabet of each symbol, for some n' . While $[y_1, y_2, \dots, y_n] \leftarrow \text{ECC}_{\text{Enc}}(n, k, w)$ outputs n encoded symbols, $y_j \leftarrow \text{ECC}_{\text{Enc}_j}(n, k, w)$ outputs the j th encoded symbol. An (n, k) error correction code can correct up to t Byzantine errors and simultaneously detect up to d Byzantine errors in n' symbol observations, given the conditions of $2t + d + k \leq n'$ and $n' \leq n$. Reed-Solomon (RS) code (cf. [19]) is one popular error correction code. The (n, k) RS code is operated over Galois Field $GF(2^c)$ under the constraint of $n \leq 2^c - 1$ (cf. [19]). Berlekamp-Welch algorithm and Euclid's algorithm are two efficient decoding algorithms for RS code [19]–[21].

Online error correction (OEC). Online error correction is a variant of traditional error correction [22]. An (n, k) error correction code can correct up to t' Byzantine errors in n' symbol observations, provided the conditions of $2t' + k \leq n'$ and $n' \leq n$. However, in an asynchronous setting, a node might not be able to decode the message with n' symbol observations if $2t' + k > n'$. In such a case, the node can wait for one more symbol observation before attempting to decode again. This process repeats until the node successfully decodes the message. By setting the threshold as $n' \geq k + t$, OEC may perform up to t trials in the worst case before decoding the message.

Definition 1 (Byzantine agreement). In the Byzantine agreement protocol, the distributed nodes seek to reach agreement on a common value. The BA protocol guarantees the following properties:

- **Termination:** If all honest nodes receive their inputs, then every honest node eventually outputs a value and terminates.
- **Consistency:** If any honest node outputs a value w , then every honest node eventually outputs w .
- **Validity:** If all honest nodes input the same value w , then every honest node eventually outputs w .

Definition 2 (Reliable broadcast [18]). In a reliable broadcast protocol, a leader inputs a value and broadcasts it to distributed nodes, satisfying the following conditions:

- **Consistency:** If any two honest nodes output w' and w'' , respectively, then $w' = w''$.
- **Validity:** If the leader is honest and inputs a value w , then every honest node eventually outputs w .
- **Totality:** If one honest node outputs a value, then every honest node eventually outputs a value.

Definition 3 (Distributed multicast). In the problem of distributed multicast (DM), there exists a subset of nodes acting as senders multicasting the message over n nodes, where up to t nodes could be dishonest. Each node acting as a sender has an input message. A protocol is called as a DM protocol if the following property is guaranteed:

- **Validity:** If all honest senders input the same message w , every honest node eventually outputs w .

Honest-majority distributed multicast (HMDM): A DM problem is called as honest-majority DM if at least $t + 1$ senders are honest. HMDM was used previously as a building block for COOL protocol, i.e., Phase 4 of COOL [3], [4].

Definition 4 (Unique agreement). Unique agreement (UA) is a variant of Byzantine agreement problem operated over n nodes, where up to t nodes may be dishonest. In a UA protocol, each node inputs an initial value and seeks to make an output taking the form as (w, s, v) , where $s \in \{0, 1\}$ is a success indicator and $v \in \{0, 1\}$ is a vote. The UA protocol guarantees the following properties:

- **Unique Agreement:** If any two honest nodes output $(w', 1, *)$ and $(w'', 1, *)$, respectively, then $w' = w''$.
- **Majority Unique Agreement:** If any honest node outputs $(*, *, 1)$, then at least $t + 1$ honest nodes eventually output $(w, 1, *)$ for the same w .
- **Validity:** If all honest nodes input the same value w , then all honest nodes eventually output $(w, 1, 1)$.

II. OCORCOOL

This proposed OcorCOOL is a deterministic and error-free Byzantine agreement protocol for the synchronous setting. OcorCOOL doesn't rely on any cryptographic assumptions such as signatures or hashing. This proposed OcorCOOL protocol is an improvement on the previous COOL protocol, using three phases instead of four [3], [4].

A. Overview of OcorCOOL

The proposed OcorCOOL is described in Algorithm 2 and Algorithm 1. In the following, we provide an overview of the proposed protocol.

1) *Phases 1 and 2:* The first two phases use the proposed OcorUA algorithm (Algorithm 1) as a building block (Line 3 of Algorithm 2). OcorUA is a UA algorithm which ensures that: 1) if any two honest nodes output $(w', 1, *)$ and $(w'', 1, *)$, respectively, then $w' = w''$ (Unique Agreement); 2) if any honest node outputs $(w, 1, 1)$, then at least $t + 1$ honest nodes eventually output $(w, 1, *)$ (Majority Unique Agreement); and 3) if all honest nodes input the same value w , then all honest nodes eventually output $(w, 1, 1)$ (Validity).

After delivering outputs from OcorUA, Node i makes a vote and runs a binary BA consensus on the votes. This ensures sure that all honest nodes make the same decision on whether to terminate at Phase 2 or go to the next phase.

2) *Phase 3:* This phase uses distributed multicast as a building block. This phase ensures that the encoded symbols from the honest nodes can be calibrated using majority rule such that the symbols are encoded from the same message. In this way, all honest nodes with success indicators of zero can output the same decoded message.

Algorithm 1 OciorUA protocol with identifier ID. Code is shown for $S_i, i \in [n]$

```

1: input  $w_i$ 
2: Initially set  $k \leftarrow \lfloor \frac{t}{5} \rfloor + 1$ ;  $w^{(i)} \leftarrow w_i$  // set the initial value of  $w^{(i)}$ 
3:  $[y_1^{(i)}, y_2^{(i)}, \dots, y_n^{(i)}] \leftarrow \text{ECCEnc}(n, k, w_i)$  // ECC encoding

Phase 1
4: send ("SYMBOL", ID,  $(y_j^{(i)}, y_i^{(i)})$ ) to  $S_j, \forall j \in [n]$  // exchange coded symbols;  $[n] := \{1, 2, \dots, n\}$ 
5: for  $j = 1 : n$  do //set link indicator
6:   if  $(y_j^{(j)}, y_j^{(j)}) = (y_i^{(i)}, y_j^{(i)})$  then  $u_i(j) \leftarrow 1$  else  $u_i(j) \leftarrow 0$ 
7: if  $\sum_{j=1}^n u_i(j) \geq n - t$  then  $s_i \leftarrow 1$  else  $s_i \leftarrow 0$ ;  $w^{(i)} \leftarrow \perp$  //set success indicator
8: send ("SI", ID,  $s_i$ ) to all nodes // exchange success indicators
9: set  $S_1 = \{j : s_j = 1, j \in [n]\}$  and  $S_0 = \{j : s_j = 0, j \in [n]\}$ , based on received success indicators  $\{s_j\}_{j=1}^n$ .

Phase 2
10: if  $s_i = 1$  then
11:   set  $u_i(j) \leftarrow 0, \forall j \in S_0$  // mask identified errors
12:   if  $\sum_{j=1}^n u_i(j) < n - t$  then
13:     set  $s_i \leftarrow 0$ ;  $w^{(i)} \leftarrow \perp$  //update success indicator
14:     send ("NewSI", ID,  $s_i$ ) to all nodes // exchange updated success indicators
15: update  $s_j \leftarrow s$  if receiving message ("NewSI", ID,  $s$ ) from  $S_j, \forall j \in S_1$  // update success indicators
16: update  $S_0$  and  $S_1$  based on the updated success indicators  $\{s_j\}_j$  // update  $S_0$  and  $S_1$ 
17: if  $|S_1| \geq 2t + 1$  then  $v_i \leftarrow 1$  else  $v_i \leftarrow 0$  // set the vote value
18: output  $[w^{(i)}, s_i, v_i, S_0, S_1, [y_1^{(1)}, y_2^{(2)}, \dots, y_n^{(n)}], [y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(n)}]]$  //  $w^{(i)}, s_i, v_i$  are three UA outputs

```

Algorithm 2 OciorCOOL protocol for BA with identifier ID. Code is shown for $S_i, i \in [n]$

```

1: input a non-empty value  $w_i$ 
2: Initially set  $k \leftarrow \lfloor \frac{t}{5} \rfloor + 1$ 

Phase 1 and Phase 2
3:  $[w^{(i)}, s_i, v_i, S_0, S_1, [y_1^{(1)}, y_2^{(2)}, \dots, y_n^{(n)}], [y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(n)}]] \leftarrow \text{OciorUA}[\text{ID}](w_i)$  // UA with two phases; see Algorithm 1
4:  $v^* \leftarrow \text{BBA}[\text{ID}](v_i)$  // BBA is a binary BA consensus on  $n$  votes  $\{v_1, v_2, \dots, v_n\}$ , by using protocol from [16], [17]
5: if  $v^* = 0$  then // if the output of binary BA is 0, set  $w^{(i)}$  as a default value
6:   output  $\perp$  and terminate
7: else
8:   go to next phase

Phase 3
9: if  $s_i = 0$  then // HMDM algorithm with one phase
10:   update  $y_i^{(i)} \leftarrow \text{Majority}(\{y_j^{(j)} : j \in S_1\})$  // update its coded symbol with majority rule
11:   send ("CORRECT", ID,  $y_i^{(i)}$ ) to the nodes in  $S_0$  // broadcast updated symbol
12:   update  $y_j^{(j)} \leftarrow y$  if receiving message ("CORRECT", ID,  $y$ ) from  $S_j, \forall j \in S_0$  // update coded symbol
13:    $w^{(i)} \leftarrow \text{ECCDec}(n, k, [y_1^{(1)}, y_2^{(2)}, \dots, y_n^{(n)}])$  // ECC decoding with updated symbols
14: output  $w^{(i)}$  and terminate

```

B. Analysis of OciorCOOL

This proposed OciorCOOL uses three phases, while the previous COOL protocol uses four phases [3], [4]. We will prove that, even with less number of phases, OciorCOOL still guarantees the termination, validity, and consistency properties. This proof will follow the original definitions in [3], [4] and will use some results in [3], [4].

For the ease of notation, we use $s_i^{[p]}$ to denote the value of s_i updated in Phase p , and use $u_i^{[p]}(j)$ to denote the values of $u_i(j)$ updated in Phase p , for $p \in \{1, 2\}$. $\mathcal{F}$ is defined as the set of indices of all dishonest nodes. In the analysis we just focus on the case with $|\mathcal{F}| = t$. It is worth noting that, the easier case with $|\mathcal{F}| = t'$ for $t' < t$ is indistinguishable from the case with $|\mathcal{F}| = t$ in which $t - t'$ out of t dishonest nodes act normally like honest nodes. Therefore, if a protocol guarantees the termination, validity, and consistency properties in the extreme case with $|\mathcal{F}| = t$, it also guarantees those properties in the easier case with $|\mathcal{F}| < t$.

We define some groups of honest nodes as

$$\mathcal{A}_l \triangleq \{i : \mathbf{w}_i = \bar{\mathbf{w}}_l, i \notin \mathcal{F}, i \in [1 : n]\}, \quad l \in [1 : \eta] \quad (1)$$

$$\mathcal{A}_l^{[p]} \triangleq \{i : s_i^{[p]} = 1, \mathbf{w}_i = \bar{\mathbf{w}}_l, i \notin \mathcal{F}, i \in [1 : n]\}, \quad l \in [1 : \eta^{[p]}], \quad p \in \{1, 2\} \quad (2)$$

$$\mathcal{B}^{[p]} \triangleq \{i : s_i^{[p]} = 0, i \notin \mathcal{F}, i \in [1 : n]\}, \quad p \in \{1, 2\} \quad (3)$$

for some different non-empty ℓ -bit values $\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \dots, \bar{\mathbf{w}}_\eta$ and some non-negative integers $\eta, \eta^{[1]}, \eta^{[2]}$ such that $\eta^{[2]} \leq \eta^{[1]} \leq \eta$. Group $\mathcal{A}_l$ (and Group $\mathcal{A}_l^{[p]}$) can be divided into some possibly overlapping sub-groups defined as

$$\mathcal{A}_{l,j} \triangleq \{i : i \in \mathcal{A}_l, \mathbf{h}_i^\top \bar{\mathbf{w}}_l = \mathbf{h}_i^\top \bar{\mathbf{w}}_j\}, \quad j \neq l, j, l \in [1 : \eta] \quad (4)$$

$$\mathcal{A}_{l,l} \triangleq \mathcal{A}_l \setminus \{\cup_{j=1, j \neq l}^\eta \mathcal{A}_{l,j}\}, \quad l \in [1 : \eta] \quad (5)$$

$$\mathcal{A}_{l,j}^{[p]} \triangleq \{i : i \in \mathcal{A}_l^{[p]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_l = \mathbf{h}_i^\top \bar{\mathbf{w}}_j\}, \quad j \neq l, j, l \in [1 : \eta^{[p]}], \quad p \in \{1, 2\} \quad (6)$$

$$\mathcal{A}_{l,l}^{[p]} \triangleq \mathcal{A}_l^{[p]} \setminus \{\cup_{j=1, j \neq l}^{\eta^{[p]}} \mathcal{A}_{l,j}^{[p]}\}, \quad l \in [1 : \eta^{[p]}], \quad p \in \{1, 2\} \quad (7)$$

where $\mathbf{h}_i$ is the encoding vector of error correction code such that the i th encoded symbol is computed as $y_i = \mathbf{h}_i^\top \mathbf{w}$, given the input vector $\mathbf{w}$, for $i \in [1 : n]$.

The main results of OciorCOOL are summarized in the following Theorems 1-4. Theorems 1-3 reveals that, given $n \geq 3t + 1$, the termination, validity and consistency conditions are all satisfied in all executions (*error-free*). Theorems 1-3 hold true without using any cryptographic assumptions (*signature-free*). Furthermore, Theorems 1-3 hold true even if the adversary has unbounded computational power (*information-theoretically secure*).

Theorem 1 (Termination). *Given $n \geq 3t + 1$, if all honest nodes receive their inputs, then every honest node eventually outputs a message and terminates in OciorCOOL.*

Proof. In OciorCOOL, if all honest nodes receive their inputs, all honest nodes eventually output messages and terminate together in Line 6 of Phase 2 or terminate together in Line 14 of Phase 3 in Algorithm 2. $\square$

Theorem 2 (Validity). *Given $n \geq 3t + 1$, if all honest nodes input the same value $\mathbf{w}$, then in OciorCOOL every honest node eventually outputs $\mathbf{w}$.*

Proof. If all honest nodes input the same value $\mathbf{w}$, then in Phase 1 each honest node eventually sets its success indicator as 1 (see Line 7 of Algorithm 1); and then in Phase 2 each honest node eventually keeps its success indicator as 1 (see Lines 10-14 of Algorithm 1). Thus, each honest node eventually goes to Phase 3 (see Line 8 of Algorithm 2) and then directly jumps to Line 14 of Algorithm 2. Therefore, if all honest nodes input the same value $\mathbf{w}$, then every honest node eventually outputs $\mathbf{w}$. $\square$

Theorem 3 (Consistency). *Given $n \geq 3t + 1$, all honest nodes eventually reach the same agreement in OciorCOOL.*

Proof. In OciorCOOL, if the output of binary BA (BBA) is 0, then every honest node outputs the default value $\perp$ (see Line 6 of Algorithm 2), satisfying the consistency condition. In the following, we will focus on the case where the output of BBA is 1.

From Lemma 1, if the output of BBA is 1, then at least $t + 1$ honest nodes have sent out success indicators as 1 in Phase 2. Furthermore, from Lemma 3, it holds true that $\eta^{[2]} \leq 1$, i.e., all honest nodes that have sent out success indicators as 1 in Phase 2 should belong to the same group $\mathcal{A}_1^{[2]}$, if any, where $\mathcal{A}_1^{[2]} \triangleq \{i : s_i^{[2]} = 1, \mathbf{w}_i = \bar{\mathbf{w}}_1, i \notin \mathcal{F}, i \in [1 : n]\}$ for some $\bar{\mathbf{w}}_1$ (see (2)). By combining the results of Lemma 1 and Lemma 3, if the output of BBA is 1, then the following conclusions are true

$$\eta^{[2]} = 1 \quad (8)$$

$$|\mathcal{A}_1^{[2]}| \geq t + 1 \quad (9)$$

$$\mathbf{w}_i = \bar{\mathbf{w}}_1, \quad \forall i \in \mathcal{A}_1^{[2]}. \quad (10)$$

If the output of BBA is 1, and given the conclusions in (8)-(10), then from Lemma 2 it is guaranteed that every honest node eventually outputs the same value $\bar{w}_1$ in Phase 3. $\square$

Theorem 4 (Communication, Round, and Resilience). *The proposed OciorCOOL is an error-free, signature-free, information-theoretically secure BA protocol that achieves the consensus on an ℓ -bit message with optimal resilience of $n \geq 3t + 1$, asymptotically optimal round complexity of $O(t)$ rounds, and asymptotically optimal communication complexity of $O(\max\{n\ell, nt \log t\})$ bits, simultaneously.*

Proof. Theorems 1-3 reveals that, given $n \geq 3t + 1$, the termination, validity and consistency conditions are all satisfied in all executions (*error-free*) in OciorCOOL. The round complexity of OciorCOOL is dominated by that of the binary BA algorithm, which is $O(t)$ rounds. The communication complexity of OciorCOOL is $O(\max\{n\ell, nt \log t\})$, similar to that of the COOL protocol [3], [4]. $\square$

C. Some lemmas

Below we provide some lemmas used in our proofs. Note that some lemmas are directly from [4].

Lemma 1. *In OciorCOOL, if the output of the binary BA is 1, then at least $t + 1$ honest nodes have sent out success indicators as 1 in Phase 2.*

Proof. If the output of BBA is 1, then at least one honest node has voted 1 in Line 17. Otherwise, BBA would deliver an output of 0. When one honest node has voted as $v_i = 1$, it means that this node has seen $|\mathbb{S}_1| \geq 2t + 1$ (see Line 8), which reveals that at least $t + 1$ honest nodes have sent out success indicators as ones in Phase 2, where $\mathbb{S}_1$ denotes the indexes of nodes who sent their success indicators as 1. $\square$

Lemma 2. *In OciorCOOL, if the output of BBA is 1, and given the conclusions in (8)-(10), then every honest node eventually outputs the same value $\bar{w}_1$ in Phase 3.*

Proof. If the output of BBA is 1, then all honest nodes go to Phase 3 (see Line 8 of Algorithm 2). In this case, all nodes within $\mathcal{A}_1^{[2]}$ directly jumps to Line 14 of Algorithm 2 and output $\bar{w}_1$. With the conclusions in (8)-(10), it can be shown that all honest nodes outside $\mathcal{A}_1^{[2]}$ will output $\bar{w}_1$ as well, thanks to the honest-majority distributed multicast (HMDM) protocol of Phase 3 (see Lines 9-13 of Algorithm 2). Phase 3 is simply an honest-majority distributed multicast protocol. In an honest-majority distributed multicast protocol defined in Definition 3, if all honest senders input the same message $\bar{w}_1$ and at least $t + 1$ senders are honest, then every honest node eventually outputs $\bar{w}_1$.

Specifically, in Line 10 of Algorithm 2, each honest node with success indicator being 0, e.g., Node i with $s_i^{[2]} = 0$, updates the value of $y_i^{(i)}$ as $y_i^{(i)} \leftarrow \text{Majority}(\{y_j^{(j)} : j \in \mathbb{S}_1\}) = \mathbf{h}_i^\top \bar{w}_1$ based on the majority rule, due to the conclusions $\mathcal{A}_1^{[2]} \subseteq \mathbb{S}_1$ and $|\mathcal{A}_1^{[2]}| > |\mathcal{F}|$ (see conclusions in (8)-(10)). After this step, for any honest Node i , the value of $y_i^{(i)}$ becomes $y_i^{(i)} = \mathbf{h}_i^\top \bar{w}_1$ that is encoded with $\bar{w}_1$. Then, in Line 11, each Node i with $s_i^{[2]} = 0$ sends the (“CORRECT”, ID, $y_i^{(i)}$) to the nodes within $\mathbb{S}_0$, where $\mathbb{S}_0 = \{j : s_j = 0, j \in [1 : n]\}$. After the step in Line 12, each honest node within $\mathbb{S}_0$ updates $y_j^{(j)}$ if receiving message (“CORRECT”, ID, $y_j^{(j)}$) from $S_j, \forall j \in \mathbb{S}_0$. After this step, for each honest node within $\mathbb{S}_0$, it is guaranteed that each symbol $y_j^{(j)}$ sent from an honest node is encoded with $\bar{w}_1$, i.e., $y_j^{(j)} = \mathbf{h}_j^\top \bar{w}_1$, for any $j \notin \mathcal{F}$. Then, in Line 13, each honest node within $\mathbb{S}_0$ decodes a message based on the updated symbols $\{y_1^{(1)}, y_2^{(2)}, \dots, y_n^{(n)}\}$, where at least $n - t$ symbols are encoded with $\bar{w}_1$. Since error correction code can correct up to t errors, each honest Node i within $\mathbb{S}_0$ should decode the message as $\bar{w}_1$, given that the number of symbols that are not encoded with the message $\bar{w}_1$ is no more than t . Thus, in this case every honest node eventually outputs the same message $\bar{w}_1$. $\square$

Lemma 3. *For the proposed OciorCOOL with $n \geq 3t + 1$, it holds true that $\eta^{[2]} \leq 1$.*

Proof. At first, from Lemma 5 it is concluded that $\eta^{[2]} \leq 2$. Next, we argue that the case of $\eta^{[2]} = 2$ does not exist. Let us assume that $\eta^{[2]} = 2$. Under the assumption of $\eta^{[2]} = 2$, it holds true that $\eta^{[1]} = 2$

(see Lemma 6). However, if $\eta^{[1]} = 2$ then it implies that $\eta^{[2]} \leq 1$ (see Lemma 7), which contradicts the assumption of $\eta^{[2]} = 2$. Therefore, the case of $\eta^{[2]} = 2$ does not exist, which, together with the result $\eta^{[2]} \leq 2$, concludes that $\eta^{[2]} \leq 1$. $\square$

Lemma 4. [4, Lemma 7] For $\eta \geq \eta^{[1]} \geq 2$, the following inequalities hold true

$$|\mathcal{A}_{l,j}| + |\mathcal{A}_{j,l}| < k, \quad \forall j \neq l, j, l \in [1 : \eta] \quad (11)$$

$$|\mathcal{A}_{l,j}^{[1]}| + |\mathcal{A}_{j,l}^{[1]}| < k, \quad \forall j \neq l, j, l \in [1 : \eta^{[1]}] \quad (12)$$

where k is a parameter of (n, k) error correction code, which is set here as $k = \lfloor \frac{t}{5} \rfloor + 1$.

Lemma 5. [4, Lemma 11] For the proposed OciorCOOL with $n \geq 3t + 1$, it holds true that $\eta^{[2]} \leq 2$.

Lemma 6. [4, Lemma 13] For the proposed OciorCOOL with $n \geq 3t + 1$, if $\eta^{[2]} = 2$, then it holds true that $\eta^{[1]} = 2$.

Lemma 7. For the proposed OciorCOOL with $n \geq 3t + 1$, if $\eta^{[1]} = 2$ then it holds true that $\eta^{[2]} \leq 1$.

Proof. We will consider the assumption of $\eta^{[1]} = 2$. Given this assumption the definition in (1)-(3) implies that

$$\mathcal{A}_1^{[1]} = \{i : s_i^{[1]} = 1, \mathbf{w}_i = \bar{\mathbf{w}}_1, i \notin \mathcal{F}, i \in [1 : n]\} \quad (13)$$

$$\mathcal{A}_2^{[1]} = \{i : s_i^{[1]} = 1, \mathbf{w}_i = \bar{\mathbf{w}}_2, i \notin \mathcal{F}, i \in [1 : n]\} \quad (14)$$

$$\mathcal{A}_{1,2}^{[1]} = \{i : i \in \mathcal{A}_1^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_1 = \mathbf{h}_i^\top \bar{\mathbf{w}}_2\} \quad (15)$$

$$\mathcal{A}_{1,1}^{[1]} = \mathcal{A}_1^{[1]} \setminus \mathcal{A}_{1,2}^{[1]} = \{i : i \in \mathcal{A}_1^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_1 \neq \mathbf{h}_i^\top \bar{\mathbf{w}}_2\} \quad (16)$$

$$\mathcal{A}_{2,1}^{[1]} = \{i : i \in \mathcal{A}_2^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_2 = \mathbf{h}_i^\top \bar{\mathbf{w}}_1\} \quad (17)$$

$$\mathcal{A}_{2,2}^{[1]} = \mathcal{A}_2^{[1]} \setminus \mathcal{A}_{2,1}^{[1]} = \{i : i \in \mathcal{A}_2^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_2 \neq \mathbf{h}_i^\top \bar{\mathbf{w}}_1\} \quad (18)$$

$$\mathcal{B}^{[1]} = \{i : s_i^{[1]} = 0, i \notin \mathcal{F}, i \in [1 : n]\} = \{i : i \in [1 : n], i \notin \mathcal{F} \cup \mathcal{A}_1^{[1]} \cup \mathcal{A}_2^{[1]}\}. \quad (19)$$

Since $|\mathcal{A}_1| + |\mathcal{A}_2| = n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|$, it is true that at least one of the following cases is satisfied:

$$\text{Case 1: } |\mathcal{A}_2| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2} \quad (20)$$

$$\text{Case 2: } |\mathcal{A}_1| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}. \quad (21)$$

1) Analysis for Case 1: We will first consider Case 1 and prove that $|\mathcal{A}_2^{[2]}| = 0$ under this case. Let us define $\mathcal{U}_i^{[p]}$ as a set of links that are matched with Node i at Phase p , that is,

$$\mathcal{U}_i^{[p]} := \{j : u_i^{[p]}(j) = 1, j \in [1 : n]\}, \quad \text{for } i \in [1 : n], p \in \{1, 2\}. \quad (22)$$

Then, for any $i \in \mathcal{A}_2^{[1]}$, the size of $\mathcal{U}_i^{[2]}$ can be bounded as

$$|\mathcal{U}_i^{[2]}| = \sum_{j \in [1:n]} u_i^{[2]}(j) \quad (23)$$

$$= \sum_{j \in [1:n] \setminus \mathcal{B}^{[1]}} u_i^{[2]}(j) \quad (24)$$

$$= \sum_{j \in [1:n] \setminus \{\mathcal{B}^{[1]} \cup \mathcal{A}_{1,1}^{[1]}\}} u_i^{[2]}(j) \quad (25)$$

$$= \sum_{j \in \{\mathcal{A}_{1,2}^{[1]} \cup \mathcal{A}_{2,1}^{[1]} \cup \mathcal{A}_{2,2}^{[1]} \cup \mathcal{F}\}} u_i^{[2]}(j) \quad (26)$$

$$\leq |\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}| + |\mathcal{A}_{2,2}^{[1]}| + |\mathcal{F}| \quad (27)$$

where (24) stems from the fact that $s_j^{[1]} = 0$ for any $j \in \mathcal{B}^{[1]}$ (see (19)), which suggests that $u_i^{[2]}(j) = 0$ for any $i \in \mathcal{A}_2^{[1]}$ (see Line 11 of Algorithm 1); (25) results from the identity that $\mathbf{h}_j^\top \bar{\mathbf{w}}_1 \neq \mathbf{h}_j^\top \bar{\mathbf{w}}_2$ for any $j \in \mathcal{A}_{1,1}^{[1]}$ (see (16)), which implies that $u_i^{[1]}(j) = u_i^{[2]}(j) = 0$ for any $j \in \mathcal{A}_{1,1}^{[1]}$ and $i \in \mathcal{A}_2^{[1]}$ (see Line 6 of Algorithm 1); (26) is from the fact that $[1 : n] = \mathcal{A}_1^{[1]} \cup \mathcal{A}_2^{[1]} \cup \mathcal{B}^{[1]} \cup \mathcal{F} = \mathcal{A}_{1,1}^{[1]} \cup \mathcal{A}_{1,2}^{[1]} \cup \mathcal{A}_{2,1}^{[1]} \cup \mathcal{A}_{2,2}^{[1]} \cup \mathcal{B}^{[1]} \cup \mathcal{F}$.

From Lemma 4, the summation of $|\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}|$ in (27) can be bounded as

$$|\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}| \leq k - 1. \quad (28)$$

From Lemma 8, the term $|\mathcal{A}_{2,2}^{[1]}|$ in (27) can be upper bounded by

$$|\mathcal{A}_{2,2}^{[1]}| \leq 2(k - 1). \quad (29)$$

At this point, for any $i \in \mathcal{A}_2^{[1]}$, the size of $\mathcal{U}_i^{[2]}$ can be bounded as

$$|\mathcal{U}_i^{[2]}| \leq |\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}| + |\mathcal{A}_{2,2}^{[1]}| + |\mathcal{F}| \quad (30)$$

$$\leq k - 1 + 2(k - 1) + |\mathcal{F}| \quad (31)$$

$$\leq 3(k - 1) + t \quad (32)$$

$$\leq 2t \quad (33)$$

$$< n - t$$

where (30) is from (27); (31) is from Lemma 4 and Lemma 8; (32) uses the value of the parameter k , i.e., $k = \lfloor t/5 \rfloor + 1$; and the last inequality uses the condition of $n \geq 3t + 1 > 3t$. With the result in (33), it is concluded that Node i , for any $i \in \mathcal{A}_2^{[1]}$, sets $s_i^{[2]} = 0$ at Phase 2 due to the derived result $|\mathcal{U}_i^{[2]}| < n - t$ (see Line 13 of Algorithm 1). Therefore, it can be concluded that $\mathcal{A}_2^{[1]}$ is in the list of $\mathbb{S}_0$, that is,

$$\mathcal{A}_2^{[1]} \subseteq \mathbb{S}_0 \quad (34)$$

at the end of Phase 2. In other words, there exists *at most* 1 group of honest nodes who input the same message and set their success indicators as ones at the end of Phase 2, while the honest nodes outside this group set their success indicators as zeros at the end of Phase 2, that is, $\eta^{[2]} \leq 1$, for Case 1.

2) *Analysis for Case 2:* By interchange the roles of $\mathcal{A}_1$ and $\mathcal{A}_2$, one can easily follow the proof for Case 1 and show that

$$\mathcal{A}_1^{[1]} \subseteq \mathbb{S}_0 \quad (35)$$

for Case 2. Then it completes the proof of this lemma. $\square$

Lemma 8. *Given the condition of $|\mathcal{A}_i| \leq \frac{n-|\mathcal{F}|-\sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}$, and for $\eta \geq 2$, the following conclusion is true*

$$|\mathcal{A}_{i,i}^{[1]}| \leq 2(k-1) \quad (36)$$

for $i \in \{1, 2\}$.

Proof. Without loss of generality, we will focus on the proof of $|\mathcal{A}_{i,i}^{[1]}| \leq 2(k-1)$ for the case with $i = 2$, under the condition of $|\mathcal{A}_i| \leq \frac{n-|\mathcal{F}|-\sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}$. The proof for the case with $i = 1$ is similar and thus omitted here.

At first let us consider the case with $\eta \geq 3$, or equivalently, $\sum_{l=3}^{\eta} |\mathcal{A}_l| > 0$. We will at first argue that $|\mathcal{A}_{2,2}^{[1]}|$ is upper bounded by

$$|\mathcal{A}_{2,2}^{[1]}| \leq \frac{(k-1) \cdot \sum_{l=3}^{\eta} |\mathcal{A}_l|}{n-t-|\mathcal{F}|-|\mathcal{A}_2|}. \quad (37)$$

From the definitions in (14) and (18), it is true that $s_i^{[1]} = 1, \forall i \in \mathcal{A}_{2,2}^{[1]}$. We recall that, Node i needs to see

$$\sum_{j \in [1:n]} u_i^{[1]}(j) \geq n-t, \quad \forall i \in \mathcal{A}_{2,2}^{[1]} \quad (38)$$

in order to set $s_i^{[1]} = 1$ at Phase 1 (see Line 7 of Algorithm 1). The condition in (38) implies that

$$\sum_{j \in [1:n] \setminus \{\mathcal{F} \cup \mathcal{A}_2\}} u_i^{[1]}(j) \geq n-t-|\mathcal{F}|-|\mathcal{A}_2|, \quad \forall i \in \mathcal{A}_{2,2}^{[1]}. \quad (39)$$

Furthermore, it is true that $u_i^{[1]}(j) = 0$ for any $i \in \mathcal{A}_{2,2}^{[1]}, j \in \mathcal{A}_1$, due to identity of $\mathbf{h}_i^T \bar{\mathbf{w}}_1 \neq \mathbf{h}_i^T \bar{\mathbf{w}}_2$ for any $i \in \mathcal{A}_{2,2}^{[1]}$ (see (18)). Then, the condition in (39) can be modified as

$$\sum_{j \in [1:n] \setminus \{\mathcal{F} \cup \mathcal{A}_2 \cup \mathcal{A}_1\}} u_i^{[1]}(j) \geq n-t-|\mathcal{F}|-|\mathcal{A}_2|, \quad \forall i \in \mathcal{A}_{2,2}^{[1]}. \quad (40)$$

Since $[1:n] \setminus \{\mathcal{F} \cup \mathcal{A}_2 \cup \mathcal{A}_1\} = \cup_{l=3}^{\eta} \mathcal{A}_l$, the condition in (40) can be expressed as

$$\sum_{j \in \cup_{l=3}^{\eta} \mathcal{A}_l} u_i^{[1]}(j) \geq n-t-|\mathcal{F}|-|\mathcal{A}_2|, \quad \forall i \in \mathcal{A}_{2,2}^{[1]}, \quad (41)$$

which also gives the following bound

$$\sum_{i \in \mathcal{A}_{2,2}^{[1]}} \sum_{j \in \cup_{l=3}^{\eta} \mathcal{A}_l} u_i^{[1]}(j) \geq (n-t-|\mathcal{F}|-|\mathcal{A}_2|) \cdot |\mathcal{A}_{2,2}^{[1]}|. \quad (42)$$

On the other hand, for any $j \in \mathcal{A}_{l^*}$, $l^* \neq l$ and $l^*, l \in [1:\eta]$, the term $\sum_{i \in \mathcal{A}_l^{[1]}} u_i^{[1]}(j)$ can be upper bounded by

$$\begin{aligned} \sum_{i \in \mathcal{A}_l^{[1]}} u_i^{[1]}(j) &= \sum_{i \in \mathcal{A}_{l,l^*}^{[1]}} u_i^{[1]}(j) + \sum_{i \in \mathcal{A}_l^{[1]} \setminus \mathcal{A}_{l,l^*}^{[1]}} u_i^{[1]}(j) \\ &= \sum_{i \in \mathcal{A}_{l,l^*}^{[1]}} u_i^{[1]}(j) \end{aligned} \quad (43)$$

$$\begin{aligned} &\leq |\mathcal{A}_{l,l^*}^{[1]}| \\ &\leq k-1 \end{aligned} \quad (44)$$

where (43) results from the fact that $\mathbf{h}_i^\top \bar{\mathbf{w}}_{l^*} \neq \mathbf{h}_i^\top \bar{\mathbf{w}}_l$ for $i \in \mathcal{A}_l^{[1]} \setminus \mathcal{A}_{l,l^*}^{[1]}$ (see (6)), which implies that $u_i^{[1]}(j) = 0$ for $i \in \mathcal{A}_l^{[1]} \setminus \mathcal{A}_{l,l^*}^{[1]}$, $j \in \mathcal{A}_{l^*}$ and $l^* \neq l$; and the last inequality in (44) follows from Lemma 4. With the result in (44), we can bound that

$$\sum_{i \in \mathcal{A}_{2,2}^{[1]}} u_i^{[1]}(j) \leq \sum_{i \in \mathcal{A}_2^{[1]}} u_i^{[1]}(j) \leq k - 1, \quad \forall j \in \cup_{l=3}^{\eta} \mathcal{A}_l \quad (45)$$

which also gives the following bound

$$\sum_{j \in \cup_{l=3}^{\eta} \mathcal{A}_l} \sum_{i \in \mathcal{A}_{2,2}^{[1]}} u_i^{[1]}(j) \leq (k - 1) \cdot \sum_{l=3}^{\eta} |\mathcal{A}_l| \quad (46)$$

By combining the results of (42) and (46), the following bound is obvious

$$(n - t - |\mathcal{F}| - |\mathcal{A}_2|) \cdot |\mathcal{A}_{2,2}^{[1]}| \leq (k - 1) \cdot \sum_{l=3}^{\eta} |\mathcal{A}_l| \quad (47)$$

which also implies that

$$|\mathcal{A}_{2,2}^{[1]}| \leq \frac{(k - 1) \cdot \sum_{l=3}^{\eta} |\mathcal{A}_l|}{n - t - |\mathcal{F}| - |\mathcal{A}_2|}, \quad (48)$$

where $n - t - |\mathcal{F}| - |\mathcal{A}_2| > 0$ holds true under the conditions of $|\mathcal{A}_2| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}$ and $n \geq 3t + 1$.

From the result in (48) we have the following bound

$$\begin{aligned} |\mathcal{A}_{2,2}^{[1]}| &\leq \frac{(k - 1) \cdot \sum_{l=3}^{\eta} |\mathcal{A}_l|}{n - t - |\mathcal{F}| - |\mathcal{A}_2|} \\ &\leq \frac{(k - 1) \cdot \sum_{l=3}^{\eta} |\mathcal{A}_l|}{n - t - |\mathcal{F}| - \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}} \end{aligned} \quad (49)$$

$$\begin{aligned} &= \frac{2(k - 1) \cdot \sum_{l=3}^{\eta} |\mathcal{A}_l|}{n - 2t - |\mathcal{F}| + \sum_{l=3}^{\eta} |\mathcal{A}_l|} \\ &= \frac{2(k - 1)}{\frac{n - 2t - |\mathcal{F}|}{\sum_{l=3}^{\eta} |\mathcal{A}_l|} + 1} \\ &\leq \frac{2(k - 1)}{0 + 1} \end{aligned} \quad (50)$$

$$= 2(k - 1) \quad (51)$$

where (49) uses the condition $|\mathcal{A}_2| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}$; (50) is derived from the identity that $\frac{n - 2t - |\mathcal{F}|}{\sum_{l=3}^{\eta} |\mathcal{A}_l|} > 0$ in this case with $\eta \geq 3$, and given the condition of $n \geq 3t + 1$.

Let us now consider the case with $\eta = 2$. In this case we will prove that $|\mathcal{A}_{2,2}^{[1]}| = 0$ under the condition of $|\mathcal{A}_2| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}$. Let us assume that $|\mathcal{A}_{2,2}^{[1]}| > 0$. Then, by following the steps in (38)-(41), and given $\eta = 2$ in this case, we have

$$0 = \sum_{j \in \cup_{l=3}^{\eta} \mathcal{A}_l} u_i^{[1]}(j) \geq n - t - |\mathcal{F}| - |\mathcal{A}_2|, \quad \forall i \in \mathcal{A}_{2,2}^{[1]}. \quad (52)$$

The bound in (52) apparently contradicts the condition of $|\mathcal{A}_2| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2} < n - t - |\mathcal{F}|$. In other words, the assumption of $|\mathcal{A}_{2,2}^{[1]}| > 0$ leads to a contradiction. Therefore, it is true that $|\mathcal{A}_{2,2}^{[1]}| = 0$ under the condition of $|\mathcal{A}_2| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}$, for this case with $\eta = 2$. At this point we complete the proof. $\square$

Algorithm 3 OciorRBC protocol with identifier (ID, l) . Code is shown for S_i .

*// ** OciorRBC can be slightly modified to a reliable broadcast protocol without balancing the communication between the leader the other nodes. In this case, in the initial phase the leader just broadcasts the whole message to each node. ***

Initial phase

```

1: Initially set  $k \leftarrow \lfloor \frac{t}{5} \rfloor + 1$ ;  $I_{\text{oec}} \leftarrow 0$ ;  $I_{\text{oecfinal}} \leftarrow 0$ ;  $\mathbb{Z}_{\text{oec}} \leftarrow \{\}$ ;  $\mathbb{Y}_{\text{oec}} \leftarrow \{\}$ ;  $\mathbb{U}_0 \leftarrow \{\}$ ;  $\mathbb{U}_1 \leftarrow \{\}$ ;  $\mathbb{S}_0^{[1]} \leftarrow \{\}$ ;  $\mathbb{S}_1^{[1]} \leftarrow \{\}$ ;  $\mathbb{S}_0^{[2]} \leftarrow \{\}$ ;  $\mathbb{S}_1^{[2]} \leftarrow \{\}$ ;  $\mathbf{w}_i \leftarrow \perp$ ;  $\mathbf{w}^{(i)} \leftarrow \perp$ ;  $I_{\text{ecc}} \leftarrow 0$ ;  $I_{\text{SI2}} \leftarrow 0$ ;  $I_1 \leftarrow 0$ ;  $I_2 \leftarrow 0$ ;  $I_3 \leftarrow 0$ 
2: upon receiving a non-empty message input  $\mathbf{w}$ , and if this node is the leader, i.e.,  $i = l$  do: // only for the leader node
3:    $[z_1, z_2, \dots, z_n] \leftarrow \text{ECCEnc}(n, k, \mathbf{w})$ 
4:   send ("LEAD", ID,  $z_j$ ) to  $S_j$ ,  $\forall j \in [n]$ 
5: upon receiving ("LEAD", ID,  $z_i$ ) from the leader for the first time do:
6:   send ("INITIAL", ID,  $z_i$ ) to all nodes // echo coded symbol
7: upon receiving message ("INITIAL", ID,  $z_j$ ) from  $S_j$  for the first time, and  $I_{\text{oec}} = 0$  do:
8:    $\mathbb{Z}_{\text{oec}} \leftarrow \mathbb{Z}_{\text{oec}} \cup \{j : z_j\}$ 
9:   if  $|\mathbb{Z}_{\text{oec}}| \geq k + t$  then // online error correcting (OEC)
10:     $\tilde{\mathbf{w}} \leftarrow \text{ECCDec}(n, k, \mathbb{Z}_{\text{oec}})$ 
11:     $[z'_1, z'_2, \dots, z'_n] \leftarrow \text{ECCEnc}(n, k, \tilde{\mathbf{w}})$ 
12:    if at least  $k + t$  symbols in  $[z'_1, z'_2, \dots, z'_n]$  match with those in  $\mathbb{Z}_{\text{oec}}$ , and  $\tilde{\mathbf{w}}$  is non-empty then
13:       $\mathbf{w}_i \leftarrow \tilde{\mathbf{w}}$ ,  $\mathbf{w}^{(i)} \leftarrow \tilde{\mathbf{w}}$ ;  $I_{\text{oec}} \leftarrow 1$ ;  $I_1 \leftarrow 1$ 

```

Phase 1

```

14: upon  $I_1 = 1$  do:
15:    $[y_1^{(i)}, y_2^{(i)}, \dots, y_n^{(i)}] \leftarrow \text{ECCEnc}(n, k, \mathbf{w}_i)$ 
16:   send ("SYMBOL", ID,  $(y_j^{(i)}, y_i^{(i)})$ ) to  $S_j$ ,  $\forall j \in [n]$ , and then set  $I_{\text{ecc}} \leftarrow 1$  // exchange coded symbols
17: upon receiving ("SYMBOL", ID,  $(y_j^{(j)}, y_j^{(j)})$ ) from  $S_j$  for the first time do:
18:   wait until  $I_{\text{ecc}} = 1$ 
19:   if  $(y_j^{(j)}, y_j^{(j)}) = (y_i^{(i)}, y_i^{(i)})$  then
20:      $\mathbb{U}_1 \leftarrow \mathbb{U}_1 \cup \{j\}$  //update the set of link indicators
21:   else
22:      $\mathbb{U}_0 \leftarrow \mathbb{U}_0 \cup \{j\}$ 
23: upon  $|\mathbb{U}_1| \geq n - t$ , and ("SI1", ID,  $*$ ) not yet sent do:
24:   set  $s_i^{[1]} \leftarrow 1$ , send ("SI1", ID,  $s_i^{[1]}$ ) to all nodes, and then set  $I_2 \leftarrow 1$  //set success indicator
25: upon  $|\mathbb{U}_0| \geq t + 1$ , and ("SI1", ID,  $*$ ) not yet sent do:
26:   set  $s_i^{[1]} \leftarrow 0$ , send ("SI1", ID,  $s_i^{[1]}$ ) to all nodes, and then set  $I_2 \leftarrow 1$ 
27: upon receiving ("SI1", ID,  $s_j^{[1]}$ ) from  $S_j$  for the first time do:
28:   if  $s_j^{[1]} = 1$  then
29:     wait until  $(j \in \mathbb{U}_1 \cup \mathbb{U}_0) \vee (|\mathbb{S}_1^{[1]}| \geq n - t) \vee (|\mathbb{S}_0^{[1]}| \geq t + 1)$ 
30:     if  $j \in \mathbb{U}_1$  then
31:        $\mathbb{S}_1^{[1]} \leftarrow \mathbb{S}_1^{[1]} \cup \{j\}$  //update the set of success indicator as ones
32:     else if  $j \in \mathbb{U}_0$  then
33:        $\mathbb{S}_0^{[1]} \leftarrow \mathbb{S}_0^{[1]} \cup \{j\}$  //mask identified errors (mismatched links)
34:   else
35:      $\mathbb{S}_0^{[1]} \leftarrow \mathbb{S}_0^{[1]} \cup \{j\}$  //mask identified errors (mismatched links)

```

Phase 2

```

36: upon  $(I_2 = 1) \wedge (s_i^{[1]} = 0)$ , and ("SI2", ID,  $s_i^{[2]}$ ) not yet sent do:
37:   set  $s_i^{[2]} \leftarrow 0$ , send ("SI2", ID,  $s_i^{[2]}$ ) to all nodes //update success indicator
38: upon  $(I_2 = 1) \wedge (s_i^{[1]} = 1) \wedge (|\mathbb{S}_1^{[1]}| \geq n - t)$ , and ("SI2", ID,  $s_i^{[2]}$ ) not yet sent do:
39:   set  $s_i^{[2]} \leftarrow 1$ ,  $I_{\text{SI2}} \leftarrow 1$ , and send ("SI2", ID,  $s_i^{[2]}$ ) to all nodes
40: upon  $|\mathbb{S}_0^{[1]}| \geq t + 1$ , and ("SI2", ID,  $s_i^{[2]}$ ) not yet sent do:
41:   set  $s_i^{[2]} \leftarrow 0$ , send ("SI2", ID,  $s_i^{[2]}$ ) to all nodes
42: upon receiving ("SI2", ID,  $s_j^{[2]}$ ) from  $S_j$  for the first time do:
43:   if  $s_j^{[2]} = 1$  then  $\mathbb{S}_1^{[2]} \leftarrow \mathbb{S}_1^{[2]} \cup \{j\}$  else  $\mathbb{S}_0^{[2]} \leftarrow \mathbb{S}_0^{[2]} \cup \{j\}$ 

```

```

44: upon  $|\mathbb{S}_v^{[2]}| \geq n - t$ , for a  $v \in \{1, 0\}$ , and ("READY", ID,  $v$ ) not yet sent do:
45:   send ("READY", ID,  $v$ ) to all nodes
46: upon receiving  $t + 1$  ("READY", ID,  $v$ ) messages from different nodes for the same  $v$  and ("READY", ID,  $v$ ) not yet sent do:
47:   send ("READY", ID,  $v$ ) to all nodes
48: upon receiving  $2t + 1$  ("READY", ID,  $v$ ) messages from different nodes for the same  $v$  do:
49:   if ("READY", ID,  $v$ ) not yet sent then
50:     send ("READY", ID,  $v$ ) to all nodes
51:   set  $v_o \leftarrow v$ 
52:   if  $v_o = 0$  then
53:     set  $w^{(i)} \leftarrow \perp$ , then output  $w^{(i)}$  and terminate                                //  $\perp$  is a default value
54:   else
55:     set  $I_3 \leftarrow 1$ 

Phase 3
56: upon  $I_3 = 1$  do:                                                                // only after executing Line 56
57:   if  $I_{SI2} = 1$  then
58:     output  $w^{(i)}$  and terminate
59:   else
60:     wait until receiving  $t + 1$  ("SYMBOL", ID,  $(y_i^{(j)}, *)$ ) messages,  $\forall j \in \mathbb{S}_1^{[2]}$ , for the same  $y_i^{(j)} = y^*$ , for some  $y^*$ 
61:      $y_i^{(i)} \leftarrow y^*$                                                                 // update coded symbol based on majority rule
62:     send ("CORRECT", ID,  $y_i^{(i)}$ ) to all nodes
63:     wait until  $I_{\text{oecfinal}} = 1$ 
64:     output  $w^{(i)}$  and terminate
65: upon receiving ("CORRECT", ID,  $y_j^{(j)}$ ) from  $S_j$  for the first time,  $j \notin \mathbb{Y}_{\text{oec}}$ , and  $I_{\text{oecfinal}} = 0$  do:
66:    $\mathbb{Y}_{\text{oec}}[j] \leftarrow y_j^{(j)}$ 
67:   if  $|\mathbb{Y}_{\text{oec}}| \geq k + t$  then                                                        // online error correcting (OEC)
68:      $\hat{w} \leftarrow \text{ECCDec}(n, k, \mathbb{Y}_{\text{oec}})$ 
69:      $[y_1, y_2, \dots, y_n] \leftarrow \text{ECCEnc}(n, k, \hat{w})$ 
70:     if at least  $k + t$  symbols in  $[y_1, y_2, \dots, y_n]$  match with those in  $\mathbb{Y}_{\text{oec}}$  then
71:        $w^{(i)} \leftarrow \hat{w}$ ;  $I_{\text{oecfinal}} \leftarrow 1$ 
72: upon having received both ("SYMBOL", ID,  $(y_i^{(j)}, y_j^{(j)})$ ) and ("SI2", ID, 1) messages from  $S_j$ , and  $j \notin \mathbb{Y}_{\text{oec}}$ , and  $I_{\text{oecfinal}} = 0$  do:
73:    $\mathbb{Y}_{\text{oec}}[j] \leftarrow y_j^{(j)}$ 
74:   run the OEC steps as in Lines 68-72

```

III. OCiorRBC

This proposed OciorRBC is an asynchronous error-free Byzantine reliable broadcast protocol. OciorRBC doesn't rely on any cryptographic assumptions such as signatures or hashing. This proposed OciorRBC is an extension of OciorCOOL.

A. Overview of OciorRBC

The proposed OciorRBC is described in Algorithm 3. In the following, we provide an overview of the proposed protocol.

1) *Initial phase*: OciorRBC is a *balanced* reliable broadcast protocol where communication overhead is distributed evenly between the leader and the other nodes. In this initial phase, the goal is to multicast the leader's message to the distributed nodes with balanced communication. This initial phase guarantees that, if the leader is honest then every honest node eventually outputs the message sent from the leader.

Without considering balanced communication, the leader could simply send the entire initial message to each node during the initial phase. However, this simple multicasting would result in a heavy communication load for the leader. To reduce this load, the leader sends different coded symbols to different nodes in the initial phase, with these symbols being encoded from the initial message (Line 4). Each node then echoes the received symbol to all other nodes (Line 6). Upon receiving the coded symbols, each node conducts online error correction to decode the message sent by the leader (Lines 7-13).

2) *Phase 1*: The goal of Phase 1 is to exchange coded information symbols and mask inconsistent messages. This phase, together with Phase 2, guarantee that all honest nodes who set their success indicators to one in Phase 2 should have the same input message at the beginning of Phase 1.

In this phase, Node i encodes the message w_i delivered from the initial phase into coded symbols $[y_1^{(i)}, y_2^{(i)}, \dots, y_n^{(i)}]$ by using error correction code, for $i \in [n]$ (Line 15). Then, Node i sends (“SYMBOL”, ID, $(y_j^{(i)}, y_i^{(i)})$) to Node j , $\forall j \in [n]$ (Line 16). Upon receiving (“SYMBOL”, ID, $(y_j^{(j)}, y_j^{(j)})$) from Node j for the first time, Node i checks if the received observation $(y_i^{(j)}, y_j^{(j)})$ matches its available local observation $(y_i^{(i)}, y_j^{(i)})$ (Line 19). Node i includes the index j into the set $\mathbb{U}_1$ if $(y_i^{(j)}, y_j^{(j)}) = (y_i^{(i)}, y_j^{(i)})$, else puts the index j into the set $\mathbb{U}_0$ (Lines 20 and 22). If $(y_i^{(j)}, y_j^{(j)}) = (y_i^{(i)}, y_j^{(i)})$, it can be considered that the link indicator between Node i and Node j , denoted by $u_i(j)$, is $u_i(j) = 1$. On the other hand, $(y_i^{(j)}, y_j^{(j)}) \neq (y_i^{(i)}, y_j^{(i)})$ implies that $u_i(j) = 0$.

When $|\mathbb{U}_1| \geq n - t$, Node i sets the success indicator at Phase 1 as $s_i^{[1]} = 1$ and sends (“SI1”, ID, $s_i^{[1]}$) to all nodes (Line 24). On the other hand, when $|\mathbb{U}_0| \geq t + 1$, Node i sets $s_i^{[1]} = 0$, and sends (“SI1”, ID, $s_i^{[1]}$) to all nodes (Line 26). It is worth noting that the two conditions of $|\mathbb{U}_1| \geq n - t$ and $|\mathbb{U}_0| \geq t + 1$ cannot be satisfied at the same time.

Upon receiving (“SI1”, ID, 1) from Node j , Node i puts the index j into the set $\mathbb{S}_1^{[1]}$ once Node i has received matched observation $(y_i^{(j)}, y_j^{(j)})$ from Node j , i.e., $j \in \mathbb{U}_1$ (Line 32). If Node i has received unmatched observation from Node j , i.e., $j \in \mathbb{U}_0$, then Node i puts the index j into the set $\mathbb{S}_0^{[1]}$ (Line 34). Upon receiving (“SI1”, ID, 0) from Node j , Node i directly puts the index j into the set $\mathbb{S}_0^{[1]}$ (Line 36).

3) *Phase 2*: One goal of Phase 2 is to mask the remaining inconsistent messages so that all honest nodes who set their success indicators to one in this phase should have the same input message at the beginning of Phase 1. Another goal of Phase 2 is to reach a consensus on whether to proceed to the next phase or terminate at this phase, together.

If the success indicator was set as $s_i^{[1]} = 0$ at Phase 1, or if $|\mathbb{S}_0^{[1]}| \geq t + 1$, then Node i sets the success indicator at Phase 2 as $s_i^{[2]} = 0$ (Lines 38 and 42). If the success indicator was set as $s_i^{[1]} = 1$ at Phase 1 and given $|\mathbb{S}_1^{[1]}| \geq n - t$, then Node i sets the success indicator at Phase 2 as $s_i^{[2]} = 1$ and sets a ready indicator as $I_{SI2} = 1$ (Line 40). After setting the value of $s_i^{[2]}$, Node i sends (“SI2”, ID, $s_i^{[2]}$) to all nodes.

Upon receiving (“SI2”, ID, $s_j^{[2]}$) from S_j , Node i conducts a process to decide whether to include the index j in $\mathbb{S}_1^{[2]}$ or $\mathbb{S}_0^{[2]}$ (Lines 43-44), similarly to the process in Phase 1 upon receiving (“SI1”, ID, $s_j^{[1]}$) (Lines 27-36).

In this phase, the distributed honest nodes also conduct a process to reach a consensus on whether to proceed to the next phase or terminate at this phase, together (Lines 45-56).

4) *Phase 3*: Phase 3 is initiated only after the distributed nodes have decided to proceed to this phase, together (Line 56). The goal of Phase 3 is to calibrate the coded symbols based on the majority rule to ensure consistent consensus outputs from honest nodes. In this phase, if Node i has set the success indicator at Phase 2 as $s_i^{[2]} = 1$ (or has set $I_{SI2} = 1$), then Node i outputs the message updated in the initial phase (Line 13) and then terminates (Line 59).

In this phase, if Node i hasn't set the success indicator at Phase 2 as $s_i^{[2]} = 1$ yet, then Node i waits until receiving $t + 1$ (“SYMBOL”, ID, $(y_j^{(j)}, *)$) messages, $\forall j \in \mathbb{S}_1^{[2]}$, for the same $y_i^{(j)} = y^*$, for some y^* (Line 61) and then updated its coded symbol $y_i^{(i)}$ as $y_i^{(i)} = y^*$ based on the majority rule (Line 62). Then Node i sends the message (“CORRECT”, ID, $y_i^{(i)}$) with updated symbol to all nodes.

In this phase, Node i conducts the online error correction to decode the message (Lines 66-75), based on the received updated symbols (Lines 66-72) and symbols from nodes that have set their success indicators to ones at Phase 2 (Lines 73-75). After the completion of online error correction, Node i outputs the decoded message and terminates (Line 65).

B. Analysis of OciorRBC

The analysis of OciorRBC follows closely that of OciorCOOL shown in Section II-B. In the analysis here we will use similar notations previously used for OciorCOOL. Similarly to the analysis for OciorCOOL, without loss of generality we just focus on the case with $|\mathcal{F}| = t$, where $\mathcal{F}$ is defined as the set of dishonest nodes. Here we use $\mathbf{w}_i^{[0]}$ to denote the value of $\mathbf{w}_i$ updated at Phase 0. If Node i never updates the value of $\mathbf{w}_i$ before termination, then $\mathbf{w}_i^{[0]}$ is considered to be a default value $\mathbf{w}_i^{[0]} = \perp$. We define some groups of honest nodes as

$$\mathcal{A}_l \triangleq \{i : \mathbf{w}_i^{[0]} = \bar{\mathbf{w}}_l, i \notin \mathcal{F}, i \in [1 : n]\}, \quad l \in [1 : \eta] \quad (53)$$

$$\mathcal{A}_l^{[p]} \triangleq \{i : s_i^{[p]} = 1, \mathbf{w}_i^{[0]} = \bar{\mathbf{w}}_l, i \notin \mathcal{F}, i \in [1 : n]\}, \quad l \in [1 : \eta^{[p]}], \quad p \in \{1, 2\} \quad (54)$$

$$\mathcal{B}^{[p]} \triangleq \{i : s_i^{[p]} = 0 \text{ or } s_i^{[p]} \text{ has never been set}, i \notin \mathcal{F}, i \in [1 : n]\}, \quad p \in \{1, 2\} \quad (55)$$

for some different ℓ -bit values $\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \dots, \bar{\mathbf{w}}_{\eta-1}$, where only one of them could be a default value $\perp$; and for some non-negative integers $\eta, \eta^{[1]}, \eta^{[2]}$ such that $\eta^{[2]} \leq \eta^{[1]} \leq \eta$. We use the same notions of $\mathcal{A}_{l,j}, \mathcal{A}_{l,l}, \mathcal{A}_{l,j}^{[p]}, \mathcal{A}_{l,l}^{[p]}$ defined in (4)-(7). We use $u_i(j), u_i^{[1]}(j), u_i^{[2]}(j) \in \{0, 1\}$ to denote the link indicator between Node i and Node j , its value at Phase 1, and its value at Phase 2, respectively, from the view of Node i , defined by

$$u_i^{[1]}(j) = u_i(j) = \begin{cases} 1 & \text{if } (y_i^{(j)}, y_j^{(j)}) = (y_i^{(i)}, y_j^{(i)}) \\ 0 & \text{else} \end{cases} \quad (56)$$

and

$$u_i^{[2]}(j) = \begin{cases} u_i^{[1]}(j) & \text{if } s_i^{[1]} = s_j^{[1]} = 1 \\ 0 & \text{else} \end{cases} \quad (57)$$

It is worth mentioning that if $(y_i^{(i)}, y_j^{(i)})$ are never sent by Node i , or $(y_i^{(j)}, y_j^{(j)})$ are never received at Node i , then $u_i^{[1]}(j) = u_i(j) = 0$. Similarly, if $s_i^{[1]}$ is never sent by Node i , or $s_j^{[1]}$ is never received at Node i , then $u_i^{[2]}(j) = 0$. In our setting, for any honest Node i and Node j , eventually they will have the same view on $(y_i^{(j)}, y_j^{(j)}), (y_i^{(i)}, y_j^{(i)}), s_i^{[1]}, s_j^{[1]}$. Therefore, it holds true that eventually $u_i(j) = u_j(i)$, $u_i^{[1]}(j) = u_j^{[1]}(i)$, and $u_i^{[2]}(j) = u_j^{[2]}(i)$, for any $i, j \notin \mathcal{F}$. In the analysis here we focus on the final values of the link indicators.

The main results of OciorRBC are summarized in Theorems 5-7. Theorems 5-6 reveal that, given $n \geq 3t + 1$, the totality, validity and consistency conditions are all satisfied in all executions (*error-free*). Theorem 7 shows that OciorRBC is optimal in terms of communication complexity, round complexity and resilience.

Theorem 5 (Totality and Consistency). *In OciorRBC, given $n \geq 3t + 1$, if one honest node outputs a value $\mathbf{w}^*$, then every honest node eventually outputs a value $\mathbf{w}^*$, for some $\mathbf{w}^*$.*

Proof. Lemma 9 reveals that, if one honest node sets the value of v_o in Line 52 as $v_o = v^*$ for a binary value $v^* \in \{1, 0\}$, then every honest node eventually sets $v_o = v^*$. Based on this result, if one honest node sets $v_o = 0$ then every honest node eventually sets $v_o = 0$. In this case, every honest node eventually outputs a default value $\perp$ (see Line 54).

Based on the result of Lemma 9, if one honest node sets $v_o = 1$ (see Line 56), then every honest node eventually sets $v_o = 1$. What remains to be proved is that in this case all honest nodes will eventually output the same value at Phase 3.

If an honest node sets $v_o = 1$ and $I_{\text{SI}2} = 1$, then this node outputs the value of $\mathbf{w}^{(i)}$ (see Line 59). Lemma 11 reveals that all of the honest nodes who set $I_{\text{SI}2} = 1$ at Phase 2 should have the same input message $\mathbf{w}^{(i)} = \mathbf{w}^*$ at the beginning of Phase 1, for some $\mathbf{w}^*$. Thus, all of the honest node who set $v_o = 1$ and $I_{\text{SI}2} = 1$ eventually output the same value $\mathbf{w}^*$.

If an honest node sets $v_o = 1$ and $I_{SI2} = 0$, it can be shown that this node will eventually output the same value w^* in Line 65. If an honest node sets $v_o = 1$, it means that this node has received at least $2t + 1$ (“READY”, ID, 1) messages from different nodes, which also implies that at least one honest node has received $n - t$ (“SI2”, ID, 1) messages from different nodes at Phase 2 (see Line 45). In other words, if an honest node sets $v_o = 1$, then at least $n - 2t$ honest nodes have sent out the message (“SI2”, ID, 1) at Phase 2. It is worth noting that if an honest Node i sends out a message (“SI2”, ID, 1) at Phase 2, this node should have sent (“SYMBOL”, ID, $(y_j^{(i)}, y_i^{(i)})$) to S_j , $\forall j \in [n]$ at Phase 1. On the other hand, Lemma 11 reveals that the honest nodes who send out (“SI2”, ID, 1) in Phase 2 should have the same input message w^* at the beginning of Phase 1, for some w^* . Thus, if an honest Node i sets $v_o = 1$, and $I_{SI2} = 0$, then it will eventually receives at least $n - 2t \geq t + 1$ matching (“SYMBOL”, ID, $(y_i^{(j)}, *)$) messages from the honest nodes within $S_1^{[2]}$, for one and only one value $y_i^{(j)} = \text{ECCEnc}_i(n, k, w^*)$, where $\text{ECCEnc}_i(n, k, w^*)$ denotes the i th symbol encoded from message w^* . In this case, Node i will set $y_i^{(i)} \leftarrow \text{ECCEnc}_i(n, k, w^*)$ in Line 62, and send (“CORRECT”, ID, $y_i^{(i)}$) to all nodes in Line 63. At this point, every symbol $y_j^{(j)}$ collected in $\mathbb{Y}_{\text{oec}}$ for $j \notin \mathcal{F}$ should be the symbol encoded from the same message w^* , where $\mathcal{F}$ denotes the set of dishonest nodes. Therefore, every honest node who sets $v_o = 1$ and $I_{SI2} = 0$ will eventually decode the message w^* with OEC decoding and output w^* in Line 65. $\square$

Theorem 6 (Validity). *Given $n \geq 3t + 1$, if the leader is honest and inputs a value w , then every honest node eventually outputs w in OciorRBC.*

Proof. If the leader is honest and inputs a value w , then each symbol z_j in (“LEAD”, ID, z_j) sent from the leader (see Line 4) or in (“INITIAL”, ID, z_j) echoed by the honest node (see Line 6) should be encoded from w . Thus, at Phase 0 every honest node will eventually decode the same message w with OEC decoding (see Lines 9-13), if this node hasn’t output a value yet.

Based on the above conclusion, if an honest node starts Phase 1, it should have already set the value of w_i and $w^{(i)}$ as $w_i = w^{(i)} = w$ (see Line 13). Therefore, all symbols $(y_i^{(j)}, y_j^{(j)})$ sent in the messages (“SYMBOL”, ID, $(y_i^{(j)}, y_j^{(j)})$) by any honest nodes should be encoded from the message w , which implies that the condition of $(y_i^{(j)}, y_j^{(j)}) = (y_i^{(i)}, y_j^{(i)})$ should be satisfied for any $i, j \notin \mathcal{F}$ (see Line 19). This suggests that the condition of $|\mathbb{U}_0| \geq t + 1$ should not be satisfied at any honest node (see Line 25) and that no honest node will set $s_i^{[1]} = 0$ or send out (“SI1”, ID, 0) at Phase 1 (see Line 26). Similarly, no honest node will set $s_i^{[2]} = 0$ or send out (“SI2”, ID, 0) at Phase 2. In this case, at least one honest node eventually receives at least $2t + 1$ (“READY”, ID, 1) messages and sets $v_o \leftarrow 1$, and then outputs $w^{(i)} = w$ at Line 59. From Theorem 5, if one honest node outputs a value w , then every honest node eventually outputs a value w . $\square$

Lemma 9. *In OciorRBC, if one honest node sets the value of v_o in Line 52 as $v_o = v^*$ for a binary value $v^* \in \{1, 0\}$, then every honest node eventually sets $v_o = v^*$.*

Proof. Let us consider the case that one honest node sets the value of v_o in Line 52 as $v_o = v^*$ for a binary value $v^* \in \{1, 0\}$. In this case, the node setting $v_o = v^*$ should have received at least $2t + 1$ (“READY”, ID, v^*) messages (see Line 49). It means that at least $t + 1$ (“READY”, ID, v^*) messages have been sent out from honest nodes. On the other hand, if two honest nodes send out messages (“READY”, ID, v^*) and (“READY”, ID, v'), respectively, then $v^* = v'$ (see Lemma 10). Therefore, in this case, each honest node eventually sends out a message (“READY”, ID, v^*) (see Lines 47-48). Thus, each honest node eventually receives at least $2t + 1$ (“READY”, ID, v^*) messages, which suggests that each honest node eventually goes to Line 49 and then set $v_o = v^*$ in Line 52. $\square$

Lemma 10. *In OciorRBC, given $n \geq 3t + 1$, if two honest nodes send out messages (“READY”, ID, v^*) and (“READY”, ID, v'), respectively, then $v^* = v'$.*

Proof. If one honest node sends out a message (“READY”, ID, v^*) (see Lines 46, 48 and 51), it means that at least one honest node has received at least $n - t$ (“SI2”, ID, v^*) messages from different nodes

(see Line 45), for a binary value $v^* \in \{1, 0\}$. In this case, at least $n - 2t$ honest nodes have sent out the same message (“SI2”, ID, v^*).

Similarly, if one honest node sends out a message (“READY”, ID, v'), it means that at least one honest node has received at least $n - t$ (“SI2”, ID, v') messages from different nodes, for a binary value $v' \in \{1, 0\}$.

In OciorRBC, each honest node sends out at most one message (“SI2”, ID, $*$). Thus, if $n - 2t$ honest nodes have sent out the same message (“SI2”, ID, v^*), it is impossible to have $n - t$ nodes sending out a different message (“SI2”, ID, v'), for $v' \neq v^*$, because $n - (n - 2t) < n - t$ given $n \geq 3t + 1$. Therefore, if two honest nodes send out messages (“READY”, ID, v^*) and (“READY”, ID, v'), respectively, then $v^* = v'$. $\square$

Lemma 11. *In OciorRBC, given $n \geq 3t + 1$, all of the honest nodes who set $I_{SI2} = 1$ or send out (“SI2”, ID, 1) in Phase 2 should have the same input message w^* at the beginning of Phase 1, for some w^* , i.e., it holds true that $\eta^{[2]} \leq 1$.*

Proof. The proof of this lemma is similar to that of Lemma 3. At first, from Lemma 14 it is concluded that $\eta^{[2]} \leq 2$. Next, we argue that the case of $\eta^{[2]} = 2$ does not exist. Let us assume that $\eta^{[2]} = 2$. Under the assumption of $\eta^{[2]} = 2$, it holds true that $\eta^{[1]} = 2$ (see Lemma 15). However, if $\eta^{[1]} = 2$ then it implies that $\eta^{[2]} \leq 1$ (see Lemma 16), which contradicts the assumption of $\eta^{[2]} = 2$. Therefore, the case of $\eta^{[2]} = 2$ does not exist, which, together with the result $\eta^{[2]} \leq 2$, concludes that $\eta^{[2]} \leq 1$. $\square$

Theorem 7 (Communication, Round, and Resilience). *For the consensus on an ℓ -bit message, and given $n \geq 3t + 1$, the total communication complexity of OciorRBC is $O(\max\{n\ell, n^2 \log n\})$ bits, while the communication per node is $O(\max\{\ell, n \log n\})$ bits. Additionally, the round complexity of OciorRBC is 7 asynchronous rounds. Without considering balance communication, the round complexity of OciorRBC is 6 rounds.*

Proof. The proposed OciorRBC satisfies the totality, validity and consistency conditions in all executions, given $n \geq 3t + 1$ (see Theorems 5-6). For the proposed OciorRBC, the communication is involved in Lines 4, 6, 16, 24, 26, 38, 40, 42, 46, 48, 51, 63. Specifically, in each communication, the node sends coded symbols or binary information to other nodes, where each symbol carries only c bits, for $c = \lceil \frac{\max\{\ell, k \cdot \log(n+1)\}}{k} \rceil$ and $k = \lfloor \frac{t}{5} \rfloor + 1$. Also, the total number of communication steps for each node is finite. Therefore, the communication per node is $O(\max\{\ell, n \log n\})$ bits, while the total communication complexity of OciorRBC is $O(\max\{n\ell, n^2 \log n\})$ bits.

OciorRBC consists of an initial phase and Phases 1-3. The number of asynchronous rounds in these phases are: 2 rounds (see Lines 4, 6), 2 rounds (see Lines 16, 24, 26), 2 round (see Lines 38, 40, 42, 46, 48, 51), and 1 round (see Line 63), respectively. Therefore, the round complexity of OciorRBC is 7 rounds in the worst case. $\square$

C. Lemmas used in the proof of Lemma 11

The proof of Lemma 11 will use the result of [4, Lemma 8]. This result considers a graph $G = (\mathcal{P}, \mathcal{E})$, where $\mathcal{P}$ is a set of $n - t$ vertices, for $\mathcal{P} = [1 : n - t]$ without loss of generality, and $\mathcal{E}$ is a set of edges. In this graph, there is a given vertex i^* for $i^* \in \mathcal{P}$, and a set of vertices $\mathcal{C}$ for $\mathcal{C} \subseteq \mathcal{P} \setminus \{i^*\}$ and $|\mathcal{C}| \geq n - 2t - 1$, such that each vertex in $\mathcal{C}$ is connected with at least $n - 2t$ edges and that one of the edges is connected to vertex i^* . Let $E_{i,j} = 1$ (respectively, $E_{i,j} = 0$) denote the presence (respectively, absence) of an edge between vertex i and vertex j , for $E_{i,j} = E_{j,i}, \forall i, j \in \mathcal{P}$. Mathematically, for this graph $G = (\mathcal{P}, \mathcal{E})$, there exists a set $\mathcal{C} \subseteq \mathcal{P} \setminus \{i^*\}$ satisfying the following conditions:

$$E_{i,i^*} = 1 \quad \forall i \in \mathcal{C} \quad (58)$$

$$\sum_{j \in \mathcal{P}} E_{i,j} \geq n - 2t \quad \forall i \in \mathcal{C} \quad (59)$$

$$|\mathcal{C}| \geq n - 2t - 1 \quad (60)$$

for a given $i^* \in \mathcal{P} = [1 : n - t]$. For this graph, we use $\mathcal{D} \subseteq \mathcal{P}$ to define a set of vertices such that each vertex in $\mathcal{D}$ is connected with at least k vertices in $\mathcal{C}$, i.e.,

$$\mathcal{D} \triangleq \left\{ i : \sum_{j \in \mathcal{C}} E_{i,j} \geq k, i \in \mathcal{P} \setminus \{i^*\} \right\} \quad (61)$$

where k is a parameter of (n, k) error correction code, which is set here as $k = \lfloor \frac{t}{5} \rfloor + 1$. For this graph $G = (\mathcal{P}, \mathcal{E})$, the size of $\mathcal{D}$ can be bounded, based on the result of [4, Lemma 8] that is restated below.

Lemma 12. [4, Lemma 8] For any graph $G = (\mathcal{P}, \mathcal{E})$ specified by (58)-(60) and for the set $\mathcal{D} \subseteq \mathcal{P}$ defined by (61), and given $n \geq 3t + 1$, the size of $\mathcal{D}$ is bounded as:

$$|\mathcal{D}| \geq n - 9t/4 - 1. \quad (62)$$

The proof of Lemma 11 will also use the following lemma, which is obtained by following the proof of [4, Lemma 9].

Lemma 13. When $\eta^{[2]} \geq 1$, it holds true that $|\mathcal{A}_l| \geq n - 9t/4$, for any $l \in [1 : \eta^{[2]}]$.

Proof. By following the proof of [4, Lemma 9], the proof here includes the following key steps:

- Step (a): Transform the network into a graph that is within the family of graphs satisfying (58)-(60), for a fixed i^* in $\mathcal{A}_{l^*}^{[2]}$ and $l^* \in [1 : \eta^{[2]}]$.
- Step (b): Bound the size of a group of honest nodes, denoted by $\mathcal{D}'$ (with the same form as in (61)), using the result of Lemma 12, i.e., $|\mathcal{D}'| \geq n - 9t/4 - 1$.
- Step (c): Argue that every processor in $\mathcal{D}'$ has the same initial message as Processor i^* .
- Step (d): Conclude from Step (c) that $\mathcal{D}'$ is a subset of $\mathcal{A}_{l^*}$, i.e., $\mathcal{D}' \cup \{i^*\} \subseteq \mathcal{A}_{l^*}$ and conclude that the size of $\mathcal{A}_{l^*}$ is bounded by the number determined in Step (b), i.e., $|\mathcal{A}_{l^*}| \geq |\mathcal{D}'| + 1 \geq n - 9t/4 - 1 + 1$, for $l^* \in [1 : \eta^{[2]}]$.

Step (a): The first step of the proof is to transform the network into a graph that is within the family of graphs defined by (58)-(60). We will consider the case of $\eta^{[2]} \geq 1$. Let us consider a fixed i^* for $i^* \in \mathcal{A}_{l^*}^{[2]}$ and $l^* \in [1 : \eta^{[2]}]$, and given $\eta^{[2]} \geq 1$. Based on the definition in (54), in this setting the honest Node i^* sets the success indicator $s_{i^*}^{[2]} = 1$ at Phase 2, under the condition of

$$|\mathbb{S}_1^{[1]}| \geq n - t \quad (63)$$

(see Lines 39 and 40). The condition in (63) implies the following inequalities:

$$|\mathbb{S}_1^{[1]} \cap \{\mathcal{F} \cup \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\}\}| \geq n - t \quad (64)$$

$$|\mathbb{S}_1^{[1]} \cap \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\}| \geq n - t - t \quad (65)$$

$$|\mathbb{U}_1 \cap \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\}| \geq n - t - t \quad (66)$$

where $\mathbb{S}_1^{[1]}$ and $\mathbb{U}_1$ are viewed from Node i^* ; (64) follows from the facts that $s_j^{[1]} = 0$ and that $j \notin \mathbb{S}_1^{[1]}$, $\forall j \in [1 : n] \setminus \{\mathcal{F} \cup \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\}\}$; (65) stems from the assumption that $|\mathcal{F}| \leq t$; and (66) is true due to the identity that $\mathbb{S}_1^{[1]} \subseteq \mathbb{U}_1$ (see Lines 31 and 32). The condition in (66) also implies that

$$\sum_{j \in \{\mathbb{U}_1 \cap \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\} \setminus \{i^*\}\}} u_{i^*}^{[1]}(j) \geq n - t - t - 1. \quad (67)$$

where $u_{i^*}^{[1]}(j)$ is the link indicator at Phase 1 defined in (56). Based on the definition in (56), it is true that $u_{i^*}^{[1]}(j) = 1, \forall j \in \mathbb{U}_1$ (see Lines 19 and 20) and that $u_{i^*}^{[1]}(j) = 1, \forall j \in \mathbb{U}_1 \cap \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\}$.

For $i^* \in \mathcal{A}_{l^*}^{[2]}$ and $l^* \in [1 : \eta^{[2]}]$, let us define a subset of $\{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\} \setminus \{i^*\}$ of honest nodes as

$$\mathcal{C}' \triangleq \{j : u_{i^*}^{[1]}(j) = 1, j \in \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\} \setminus \{i^*\}\}. \quad (68)$$

We can understand $\mathcal{C}'$ as a subset of $\{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\} \setminus \{i^*\}$ of honest nodes, in which each node sends a matched observation to Node i^* . The observation sent from Node j to Node i^* is defined by $(y_{i^*}^{(j)}, y_j^{(j)})$ (see Line 16). This observation is said to be matched if $(y_{i^*}^{(j)}, y_j^{(j)}) = (y_{i^*}^{(i^*)}, y_j^{(i^*)})$. One can see that $\{\mathbb{U}_1 \cap \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\}\} \setminus \{i^*\}$ is a subset of $\mathcal{C}'$ due to the fact that $u_{i^*}^{[1]}(j) = 1, \forall j \in \mathbb{U}_1 \cap \{\cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}\}$. Based on (67) and (68), the following conclusions are true

$$u_j^{[1]}(i^*) = 1, \quad \forall j \in \mathcal{C}' \quad (69)$$

$$|\mathcal{C}'| \geq n - 2t - 1. \quad (70)$$

Note that in our setting it holds true that $u_i^{[1]}(j) = u_j^{[1]}(i), \forall i, j \in \cup_{l=1}^{\eta} \mathcal{A}_l$ (see (56)).

Due to the fact that $\mathcal{C}' \subseteq \cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}$ and that $s_j^{[1]} = 1, \forall j \in \cup_{p=1}^{\eta^{[1]}} \mathcal{A}_p^{[1]}$ (see (54)), the following conclusion is true:

$$s_j^{[1]} = 1, \quad \forall j \in \mathcal{C}'. \quad (71)$$

The conclusion in (71) also implies that

$$\sum_{p=1}^n u_j^{[1]}(p) \geq n - t, \quad \forall j \in \mathcal{C}' \quad (72)$$

(see Line 23 and Line 24) and that

$$\sum_{p \in \cup_{l=1}^{\eta} \mathcal{A}_l} u_j^{[1]}(p) \geq n - 2t, \quad \forall j \in \mathcal{C}' \quad (73)$$

where $\cup_{l=1}^{\eta} \mathcal{A}_l = [1 : n] \setminus \mathcal{F}$. Intuitively, for any $j \in \mathcal{C}'$, Node j receives at least $n - 2t$ number of matched observations from the honest nodes within $\cup_{l=1}^{\eta} \mathcal{A}_l$. Let us define a subset of $\{\cup_{l=1}^{\eta} \mathcal{A}_l\} \setminus \{i^*\}$ of honest processors as

$$\mathcal{D}' \triangleq \left\{ p : \sum_{j \in \mathcal{C}'} u_j^{[1]}(p) \geq k, p \in \{\cup_{l=1}^{\eta} \mathcal{A}_l\} \setminus \{i^*\} \right\} \quad (74)$$

where k is set as $k = \lfloor \frac{t}{5} \rfloor + 1$. We can understand $\mathcal{D}'$ as a set of honest nodes in which each node sends at least k matched observations to the nodes in $\mathcal{C}'$.

At this point, we map the network into a graph $G = (\mathcal{P}', \mathcal{E}')$, where $\mathcal{P}'$ is a set of $n - t$ vertices defined by $\mathcal{P}' = \cup_{l=1}^{\eta} \mathcal{A}_l$, and $\mathcal{E}'$ is a set of edges defined by $E_{i,j} = u_i^{[1]}(j), \forall i, j \in \mathcal{P}'$. For this graph $G = (\mathcal{P}', \mathcal{E}')$, there exists a set $\mathcal{C}' \subseteq \mathcal{P}' \setminus \{i^*\}$ such that the conditions in (69), (70) and (73) are satisfied, for a given $i^* \in \mathcal{A}_{l^*}^{[2]} \subseteq \mathcal{P}'$. Since conditions in (69), (70) and (73) are similar to the conditions in (58), (60) and (59), respectively, this graph $G = (\mathcal{P}', \mathcal{E}')$ falls into a family of graphs satisfying (58)-(60).

Steps (b)-(d): The remaining steps of this proof are similar to the steps (b)-(d) of the proof of [4, Lemma 9]. $\square$

Lemma 14. *For the proposed OciorRBC with $n \geq 3t + 1$, it holds true that $\eta^{[2]} \leq 2$.*

Proof. This proof is based on the result of Lemma 13. The proof of this lemma is similar to that of [4, Lemma 11]. The details are omitted here. $\square$

Lemma 15. *For the proposed OciorRBC with $n \geq 3t + 1$, if $\eta^{[2]} = 2$, then it holds true that $\eta^{[1]} = 2$.*

Proof. This proof is based on the result of Lemma 4 and Lemma 13. The proof of this lemma is similar to that of [4, Lemma 13]. The details are omitted here. $\square$

Lemma 16. *For the proposed OciorRBC with $n \geq 3t + 1$, if $\eta^{[1]} = 2$ then it holds true that $\eta^{[2]} \leq 1$.*

Proof. The proof of this lemma is similar to that of Lemma 7 of OciorCOOL. We will consider the assumption of $\eta^{[1]} = 2$. Under this assumption, the definition in (53)-(55) implies that

$$\mathcal{A}_1^{[1]} = \{i : s_i^{[1]} = 1, \mathbf{w}_i^{[0]} = \bar{\mathbf{w}}_1, i \notin \mathcal{F}, i \in [1 : n]\} \quad (75)$$

$$\mathcal{A}_2^{[1]} = \{i : s_i^{[1]} = 1, \mathbf{w}_i^{[0]} = \bar{\mathbf{w}}_2, i \notin \mathcal{F}, i \in [1 : n]\} \quad (76)$$

$$\mathcal{A}_{1,2}^{[1]} = \{i : i \in \mathcal{A}_1^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_1 = \mathbf{h}_i^\top \bar{\mathbf{w}}_2\} \quad (77)$$

$$\mathcal{A}_{1,1}^{[1]} = \mathcal{A}_1^{[1]} \setminus \mathcal{A}_{1,2}^{[1]} = \{i : i \in \mathcal{A}_1^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_1 \neq \mathbf{h}_i^\top \bar{\mathbf{w}}_2\} \quad (78)$$

$$\mathcal{A}_{2,1}^{[1]} = \{i : i \in \mathcal{A}_2^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_2 = \mathbf{h}_i^\top \bar{\mathbf{w}}_1\} \quad (79)$$

$$\mathcal{A}_{2,2}^{[1]} = \mathcal{A}_2^{[1]} \setminus \mathcal{A}_{2,1}^{[1]} = \{i : i \in \mathcal{A}_2^{[1]}, \mathbf{h}_i^\top \bar{\mathbf{w}}_2 \neq \mathbf{h}_i^\top \bar{\mathbf{w}}_1\} \quad (80)$$

$$\mathcal{B}^{[1]} = \{i : s_i^{[1]} = 0 \text{ or } s_i^{[1]} \text{ has never been set, } i \notin \mathcal{F}, i \in [1 : n]\} \quad (81)$$

where $\mathbf{w}_i^{[0]}$ denotes the value of $\mathbf{w}_i$ updated at Phase 0. If Node i never updates the value of $\mathbf{w}_i$ before termination, then $\mathbf{w}_i^{[0]}$ is considered to be a default value $\mathbf{w}_i^{[0]} = \perp$. Since $|\mathcal{A}_1| + |\mathcal{A}_2| = n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|$, it is true that at least one of the following cases is satisfied:

$$\text{Case 1: } |\mathcal{A}_2| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2} \quad (82)$$

$$\text{Case 2: } |\mathcal{A}_1| \leq \frac{n - |\mathcal{F}| - \sum_{l=3}^{\eta} |\mathcal{A}_l|}{2}. \quad (83)$$

1) *Analysis for Case 1:* We will first consider Case 1 and prove that $|\mathcal{A}_2^{[2]}| = 0$ under this case. Let us define $\mathcal{U}_i^{[p]}$ as a set of links that are matched with Node i at Phase p , that is,

$$\mathcal{U}_i^{[p]} := \{j : u_i^{[p]}(j) = 1, j \in [1 : n]\}, \quad \text{for } i \in [1 : n], p \in \{1, 2\}, \quad (84)$$

where $u_i^{[p]}(j)$ is defined in (56) and (57). Then, for any $i \in \mathcal{A}_2^{[1]}$, the size of $\mathcal{U}_i^{[2]}$ can be bounded as

$$|\mathcal{U}_i^{[2]}| = \sum_{j \in [1:n]} u_i^{[2]}(j) \quad (85)$$

$$= \sum_{j \in [1:n] \setminus \mathcal{B}^{[1]}} u_i^{[2]}(j) \quad (86)$$

$$= \sum_{j \in [1:n] \setminus \{\mathcal{B}^{[1]} \cup \mathcal{A}_{1,1}^{[1]}\}} u_i^{[2]}(j) \quad (87)$$

$$= \sum_{j \in \{\mathcal{A}_{1,2}^{[1]} \cup \mathcal{A}_{2,1}^{[1]} \cup \mathcal{A}_{2,2}^{[1]} \cup \mathcal{F}\}} u_i^{[2]}(j) \quad (88)$$

$$\leq |\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}| + |\mathcal{A}_{2,2}^{[1]}| + |\mathcal{F}| \quad (89)$$

where (86) stems from the fact that $s_j^{[1]} = 0$ for any $j \in \mathcal{B}^{[1]}$ (see (81)), which suggests that $u_i^{[2]}(j) = 0$ (see (57)); (87) results from the identity that $\mathbf{h}_j^\top \bar{\mathbf{w}}_1 \neq \mathbf{h}_j^\top \bar{\mathbf{w}}_2$ for any $j \in \mathcal{A}_{1,1}^{[1]}$ (see (78)), which implies that $u_i^{[1]}(j) = 0$; (88) is from the fact that $[1 : n] = \mathcal{A}_{1,1}^{[1]} \cup \mathcal{A}_{1,2}^{[1]} \cup \mathcal{A}_{2,1}^{[1]} \cup \mathcal{A}_{2,2}^{[1]} \cup \mathcal{B}^{[1]} \cup \mathcal{F}$.

From Lemma 4, the summation of $|\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}|$ in (89) can be bounded as

$$|\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}| \leq k - 1. \quad (90)$$

From Lemma 8, the term $|\mathcal{A}_{2,2}^{[1]}|$ in (89) can be upper bounded by

$$|\mathcal{A}_{2,2}^{[1]}| \leq 2(k - 1). \quad (91)$$

At this point, for any $i \in \mathcal{A}_2^{[1]}$, the size of $\mathcal{U}_i^{[2]}$ can be bounded as

$$|\mathcal{U}_i^{[2]}| \leq |\mathcal{A}_{1,2}^{[1]}| + |\mathcal{A}_{2,1}^{[1]}| + |\mathcal{A}_{2,2}^{[1]}| + |\mathcal{F}| \quad (92)$$

$$\leq k - 1 + 2(k - 1) + |\mathcal{F}| \quad (93)$$

$$\leq 3(k - 1) + t$$

$$= 3(\lfloor t/5 \rfloor + 1 - 1) + t \quad (94)$$

$$\leq 2t$$

$$< n - t \quad (95)$$

where (92) is from (89); (93) is from Lemma 4 and Lemma 8; (94) uses the value of the parameter k , i.e., $k = \lfloor t/5 \rfloor + 1$; and the last inequality uses the condition of $n \geq 3t + 1 > 3t$. With the result in (95), it is concluded that $s_i^{[2]}$ cannot be 1 for any $i \in \mathcal{A}_2^{[1]}$, due to the derived result $|\mathcal{U}_i^{[2]}| < n - t$ (see Lines 39 and 40). Therefore, it can be concluded that

$$|\mathcal{A}_2^{[2]}| = 0 \quad (96)$$

for Case 1.

2) *Analysis for Case 2:* By interchange the roles of $\mathcal{A}_1$ and $\mathcal{A}_2$, one can easily follow the proof for Case 1 and show that

$$|\mathcal{A}_1^{[2]}| = 0 \quad (97)$$

for Case 2. Then it completes the proof of this lemma. $\square$

REFERENCES

- [1] M. Pease, R. Shostak, and L. Lamport, “Reaching agreement in the presence of faults,” *Journal of the ACM*, vol. 27, no. 2, pp. 228–234, Apr. 1980.
- [2] L. Lamport, R. Shostak, and M. Pease, “The Byzantine generals problem,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, Jul. 1982.
- [3] J. Chen, “Optimal error-free multi-valued Byzantine agreement,” in *International Symposium on Distributed Computing (DISC)*, Oct. 2021.
- [4] —, “Fundamental limits of Byzantine agreement,” 2020, available on arXiv: <https://arxiv.org/pdf/2009.10965.pdf>.
- [5] F. Li and J. Chen, “Communication-efficient signature-free asynchronous Byzantine agreement,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021.
- [6] J. Zhu, F. Li, and J. Chen, “Communication-efficient and error-free gradecast with optimal resilience,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2023, pp. 108–113.
- [7] M. Fitzi and M. Hirt, “Optimally efficient multi-valued Byzantine agreement,” in *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, Jul. 2006, pp. 163–168.
- [8] G. Liang and N. Vaidya, “Error-free multi-valued consensus with Byzantine failures,” in *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, Jun. 2011, pp. 11–20.
- [9] C. Ganesh and A. Patra, “Optimal extension protocols for Byzantine broadcast and agreement,” in *Distributed Computing*, Jul. 2020.
- [10] A. Loveless, R. Dreslinski, and B. Kasikci, “Optimal and error-free multi-valued Byzantine consensus through parallel execution,” 2020, available on: <https://eprint.iacr.org/2020/322>.
- [11] K. Nayak, L. Ren, E. Shi, N. Vaidya, and Z. Xiang, “Improved extension protocols for Byzantine broadcast and agreement,” in *International Symposium on Distributed Computing (DISC)*, Oct. 2020.
- [12] A. Patra, “Error-free multi-valued broadcast and Byzantine agreement with optimal communication complexity,” in *International Conference on Principles of Distributed Systems (OPODIS)*, 2011, pp. 34–49.
- [13] C. Cachin and S. Tessaro, “Asynchronous verifiable information dispersal,” in *IEEE Symposium on Reliable Distributed Systems (SRDS)*, Oct. 2005.
- [14] P. Civit, M. A. Dzulfikar, S. Gilbert, R. Guerraoui, J. Komatovic, M. Vidigueira, and I. Zabolotchi, “Efficient signature-free validated agreement,” in *International Symposium on Distributed Computing (DISC)*, vol. 319, Oct. 2024, pp. 14:1–14:23.
- [15] N. Alhaddad, S. Das, S. Duan, L. Ren, M. Varia, Z. Xiang, and H. Zhang, “Balanced Byzantine reliable broadcast with near-optimal communication and improved computation,” in *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, Jul. 2022, pp. 399–417.
- [16] P. Berman, J. Garay, and K. Perry, “Bit optimal distributed consensus,” *Computer Science*, pp. 313–321, 1992.
- [17] B. Coan and J. Welch, “Modular construction of a Byzantine agreement protocol with optimal message bit complexity,” *Information and Computation*, vol. 97, no. 1, pp. 61–85, Mar. 1992.
- [18] G. Bracha, “Asynchronous Byzantine agreement protocols,” *Information and Computation*, vol. 75, no. 2, pp. 130–143, Nov. 1987.

- [19] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, Jun. 1960.
- [20] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.
- [21] E. Berlekamp, "Nonbinary BCH decoding (abstr.)," *IEEE Trans. Inf. Theory*, vol. 14, no. 2, pp. 242–242, Mar. 1968.
- [22] M. Ben-Or, R. Canetti, and O. Goldreich, "Asynchronous secure computation," in *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, 1993, pp. 52–61.