The Latency Costs of Optimism and of Strong Unanimity

MATTHIEU RAMBAUD, Telecom Paris, Institut polytechnique de Paris

This paper studies the classical problem of consensus in a partially synchronous message-passing system where some players may behave maliciously. We consider the validity requirement denoted as Weak Unanimity, which is that if all players are honest and have the same input, then this is the only output allowed. We enable furthermore players to sign their messages, so that they are authenticated even when forwarded. We finally place ourselves in the mainstream model where production of an output is guaranteed if a specific player (arbitrarily designated by a mechanism part of the model), is honest. We study the tradeoff between the time to produce an output in two restricted cases:

- the normal case, i.e., the network is synchronous;
- the favorable case, i.e., furthermore all players are honest and have input 0.

In this context we show that, if an algorithm produces an output in the favorable case after just one round of communication, then the normal case must take more than 3 rounds (there are algorithms that solve the normal case in 3 rounds). The same impossibility of output in 3 rounds applies to consensus with Strong Unanimity.

1 INTRODUCTION

We consider \( n = 3t + 1 \) players, linked together by pairwise message-passing authenticated channels. They have access to a digital signature scheme, that guarantees authentication of their messages even when forwarded (as in [23, 36]). We consider an external entity denoted as the Environment, which corrupts up to \( t \) players, sees their full internal state, and have them behave arbitrarily. The remaining \( n - t \) players are denoted "honest". They are initially given an input value in \( \{0, 1\} \) by the Environment. We assume that the network has the property defined as "partial synchrony" in [26], as follows. Players are given a global clock that ticks at regular intervals denoted rounds. Players send messages at the beginning of a round. By default, the Environment arbitrarily schedules the delivery of messages. At any moment in time, which we denote GST, the Environment can take the specific action of making the network synchronous forever. More precisely: all messages sent before GST are delivered at GST+1, and that messages sent after GST at the beginning of a round are delivered at the end of the same round. GST can possibly never happen in some executions.

The Environment makes public the identity of a player, which is denoted as the "leader". In our main result we assume for simplicity that this leader is the same throughout the execution.

Definition 1.1. A consensus with weak unanimity and (worst-case) normal-case latency in \( R \) rounds, is a protocol that has the following properties:

(Consistency) no two honest players output different values;

(Weak Unanimity) if all players are honest and have the same input, then this is the only possible output value;

(Normal case) if the leader is honest and GST happens at the beginning of the execution, i.e. GST = 0, then all honest players output by the end of the \( R \)th round;

(Backup) if GST = 0, then all honest players output after a finite number of rounds.

Let us make some comments on the model with respect to the literature. The popular protocols [13, 28, 45] guarantee output, in particular, under the event where there is a honest leader and that the network is synchronous forever. [13] has Normal-case termination in 3 messages delays. The protocols [13, 28, 45] do not specify Backup liveness as stated.
above. But, on the other hand, the leader in these protocols is regularly replaced. Plus, in these protocols players are guaranteed to output in finite time after GST after a honest leader is designated. So this implies Backup in our sense. We capture this implication more precisely in Lemma 9. Our Backup requirement is weaker, since it does not guarantee output is $GST > 0$, thus will makes our impossibility result stronger. This round-based model with a global clock, that we consider, gives more power to players than the network considered in [13, 28, 45], thus this will make our impossibility result stronger. Interest of the leader-based model that we consider in our Definition, is that the protocols [45] and [1, §6.1], which are in this model, are respectively used as building blocks in the leader-less state of the art randomized Consensus protocols of [7] and [1, §6]. Weak Unanimity is the only guarantee of most protocols called “BFT” or “state machine replication” [6, 13, 14, 16, 19, 45]. We now define two further guarantees.

Definition 1.2. A consensus with weak unanimity has:

(Fast Track) if, in every execution where all players are honest and have input 0, then, they output at the end of the 1st round;

(Strong Unanimity) if, if all honest players have the same input value, then this is the only possible output.

Notice that the following protocols are in our model but guarantee output in only 2 rounds under the favorable conditions of our Fast Track, instead of 1 as in our specification: [17, 28, 29, 31–33, 37]. The reason is that in their model, with respect to one instance of consensus, then the inputs may be seen as the values that players receive from the first leader in the first round (which is by convention the empty value if they do not receive any). The recent [28] has even a fast track in 3 rounds, because communications are centered around the leader. We observe in Lemma 7 that a Fast Track can be trivially enforced on the top of any given consensus with Strong Unanimity protocol, by imposing the following additional set of instructions. Require players to initially multicast their input value in a message denoted as “report” at the beginning, in addition to any other message they may send. Then, on reception $n$ such signed “report” messages containing the same value, any player outputs this value. Applying this Lemma to the consensus with Strong Unanimity of §6.1 of the long version of [1, Podc’19], gives a consensus with a Fast Track, and 4 rounds of Normal-case latency. So this is one more round of Normal case latency, than the 3 of PBFT. Our main Theorem below, in (B), shows that this is unavoidable.

Main Theorem 1. Under the previous model, then any deterministic binary consensus with weak unanimity:

(A) has a normal-case latency of at least 3 rounds as soon as $n \geq 7$. This is tight;

(B) but if one assumes furthermore either: a Fast Track, and/or Strong Unanimity, then the normal case latency is at least 4 rounds as soon as $n \geq 10$. This is tight.

We now state variations of this lower bounds, in order to capture more protocols in the litterature. First, we enlarge the model and allow the Environment to designate a second leader at any point in time. This enrichment of the model is motivated by that it is part of the model of [45], and that such mechanism for designation of a new leader is implemented in [11, 13]. We observe that in [13], such a leader which is not the first one, needs 4 rounds instead of 3 to enforce an output. This is due to an initial round where players report their status to him. The next result (C) will show this one more round unavoidable. Next, we consider the subclass of protocols with a message complexity which is linear in the number of players. In this setting, a Fast track takes necessarily two round because otherwise it would require a round of all to all communication (cf the bounds of [25]), which is quadratic. The next theorem addresses those two variations and is proven by the same techniques.

Theorem 2. (C) [late honest leader] In the same model, consider a consensus with weak unanimity. Assume furthermore: that messages sent before GST can be deleted forever by the Environment before being received (as assumed in the “basic
round” model [26, §3.1]). Assume the following intermediate requirement between weak and Strong Unanimity: in executions where all honest players have input 1 and dishonest players send no message, then 1 is the only possible output. Then consider executions in which the Environment designates a second leader, which is honest, after 3 rounds, and such that GST happens before or at this point, i.e. GST ≤ 3. Then the worst case latency from round 3, until before every player outputs, is at least of 4 rounds as soon as n ≥ 22. This is tight.

Restricting to protocols with a worst case subquadratic number of messages (that is: in o(n^2)) sent by honest players, this number of messages being measured in the timeframe: from the point where we have both that GST happened and a honest leader is designated; until all honest players have output. Then (A) becomes (A’): 5 (for t ≥ 4) for plain consensus with Weak Unanimity. We consider protocols with a fast track in an optimal two rounds, then (B) becomes (B’): 6 (for t ≥ 3). (C) becomes (C’): 6 (for t ≥ 4).

The technical validity condition on 1 in (C) is actually a particular loose case of external validity, as guaranteed in [12] and [7, Lemma 15 of full paper]. For the last results for subquadratic protocols, we prove first the lower bounds for the subcase of protocols with “star-shaped” communications, i.e., between players and the current leader. We then sketch in §4 how to deduce the general result.

The easy Lemma 7 of §3 shows that Strong Unanimity compiles itself into Optimistically fast output in one round, without modifying the Initial honest leader latency. This reduces the proof of (B), to the case of Optimistically fast output in one round. We have the analogous Lemma 8 for the case of leader-centric protocols with a linear message complexity.

1.1 Other related works and upper bounds

Upper bounds (now proven tight). In our setting of malicious corruptions, the only works known to us considering a Fast Track in one round, as in our Definition, are [27, 39, 44, 46]. This scarcity contrasts with the much more investigated crash-fault setting of Fast Paxos, see below. A consensus with Strong Unanimity with normal case latency of 4 rounds is given in [1, §6.1 of long version]. A Fast Track can then be deduced from Lemma 7. (A’) Plain consensus with Weak Unanimity with linear communication complexity in 5 rounds: the two-phase Hotstuff ([3],[45, 4.4]) achieves this, since the first leader needs not waiting for view change reports before he can propose his value. SBFT [28, DSN’19] achieves the same. It has in addition fast output in 3 rounds instead of 2, thus escapes the lower bound (B’).

Other Consensus with weak Unanimity with optimistically fast output in an underoptimal number of rounds only. [25] allows a Fast Track in two rounds under favorable conditions, hence it escapes our lower bound (B) that deals with a Fast Track in 1 round. This fast output in one more round than the optimal happens also in [17, 29, 31, 32, 37]. We have the same underoptimality for [28] (3 fast rounds instead of 2) in the subclass of protocols with linear message complexity. Adding one more round for the Fast Track makes things easier for the normal case. But the technique used by [28] precludes Strong Unanimity, since in some sense the leader imposes his input to players. It is thus not surprising that they escape the lower bounds (B) and (B’). In detail the technique, examplified on both [28, 31], consists in having the dealer propose his input v to all players right in the first round. If they all acknowledge reception, then a fast output of v is enforced. Otherwise, the leader anyway has to collect a quorum of 2t + 1 acknowledgements for a value proposed in the fast track, in order for this value to be eligible to the normal track. This makes safety of the normal track much easier. Despite this easier setting, the preprints [4, 5, 42, 43] exhibit safety or liveness violations in such published protocols.
The crash fault setting is what motivated this work. Consensus in one round was initiated by [10, 41]. Guerroui-Raynal [30, §5.5.3] then Dobre-Suri [20, §5], and independently Charron-Bost & Schiper [15] were the first to our knowledge to ask if being optimistic unavoidably harms latency in the normal case. Indeed, Lamport observed in his [34] that “If collisions are too frequent [here: if not all players have the same input], then classic Paxos might be better than Fast Paxos.” The three of them answered that degradation can be circumvented, at least when Fast Track is conditioned to honesty of all the n players, as in our Definition. Our lower bound (B) shows that, by contrast, this degradation of latency cannot be circumvented in the partially synchronous Byzantine setting.

Bounds on the number of corruptions under which a Fast Track can be guaranteed. n minus this number is denoted as a “fast quorum”. Our Definition of a Fast Track holds only if a fast quorum is n. Lamport [35] gives tight bounds on this number in the crash fault setting. In our setting it is folklore (and straightforward) that a fast quorum cannot be of smaller size than n = 3t + 1 (otherwise in a backup quorum of 2t + 1 players, we might have that t of them report input 0, while t others report input 1, so both could have been fast-decided by an isolated honest player). See [25] for close bounds. Two recent works [8, 38] recently established tight bounds on the fast quorum sizes in the Byzantine synchronous setting, with respect to the overall resilience and latency. However their upper bounds confirm that, also in this setting, degradation of the overall latency can be circumvented, with e.g. a fast quorum larger than 3n/4. Our result (B) shows that this is hopeless in the partially synchronous setting.

Early decision in the unauthenticated deterministic synchronous setting, for broadcast. Dolev-Reischuk-Strong [22, §9] (FOCS’82) asked if Byzantine broadcast in which players output in \( \leq f + 2 \) rounds when there are \( f < t \) corrupted players, can always output within the tight general \( t + 1 \) number of rounds with optimal \( t < n/3 \) resiliency. The problem was solved in [9], then in Abraham-Dolev [2] (STOC’15) with optimal communication complexity. Notice that, in the unauthenticated leaderless setting, Dolev-Lenzen [21, Podc’13] show a strict communication cost for earlier output, even in the crash fault setting (cubic instead of quadratic).

Other related models and randomized algorithms. In the blockchain setting we have the optimistic [40]. In the leaderless synchronous randomized setting, Cohen [18] showed that under synchrony, then 3 rounds is the minimum to have a non negligible probability that every player outputs, which is matched by the baseline of [16] provided a common coin. The question is open whether this degrades over partial synchrony: for instance recall that in the deterministic setting, partial synchrony is shown in [24] (PODC’02) to require one more round before output. Anyway the expected latency may be longer: for instance in the partially synchronous [1, §6] (PODC’19 long version), which uses leader sortition using a trusted setup, the expected number of rounds before output is more than 5 "phases", so 20 rounds.

2 PROOF OF MAIN THEOREM 1

We follow the classical strategy consisting in exhibiting two executions in which players output different values, then prove that the values should actually be the same in both executions, whence a contradiction. The goal to prove (B) is to show this for the executions depicted on the top of Figure 1. In \( E^\text{fast}_0 \) all players are honest, have input 0, and the network is synchronous. Thus by the fast track assumption, player \( P_{10} \) outputs 0 at the end of the 1st round. Whereas in \( E_1 \), all players are honest, in particular the leader, and the network is synchronous. So by the normal case assumption they all output at the end of the 3rd round. By Weak Unanimity, they must output 1. In §2.1 we introduce the formalism. In §2.2 we recall the structure of the proof of [23]. Then highlight why the proofs of their intermediary results do not apply to our setting. We address first the main difficulty, which is Lemma 4. We prove it by proving first in §2.3 a more
general case of Theorem 1 (A). Then deduce the Lemma in 2.4. This modular approach may be more friendly, but gives a suboptimal condition \( t \geq 9 \), so in §2.6 we prove the Lemma directly. In §2.5 we finally formalize the proof of 1 (B), which should be clear from the §2.2.

2.1 Formalism

A (partial) execution is a finite sequence of rounds that starts from when the Environment initializes players, gives the honest ones their inputs and corrupts up to \( t \) of them. When \( GST = 0 \), then all messages sent are delivered within one round. We depict this as “GST” in a green circle when \( GST = 0 \). Notice that by determinism, the full state of a player in a round boils down: his input, the leader, and the chronology of all messages received so far. Likewise the state of a group \( BQ_h \) of honest players (e.g. as in Lemma 9) is, accordingly, their inputs, the leader, and the chronology of all messages received by players in \( BQ_h \) so far. We assume without loss of generality a protocol which instructs players to multicast, at the beginning of each every round, their full internal state to all other players. Then to emulate any actual protocol, we simply have receivers discard the information that they are not supposed to process. We prove (B) for \( 3t + 1 = 10 \) players: \( P_1, \ldots, P_{10} \) (and (A) for \( n = 7 \)), then it generalizes to any \( n \) by standard methods [30, 36].

We say that two (partial) executions \( E \) and \( E' \) up to some finite round \( r \) are equivalent from the point of view of some honest player \( P \), or some group of honest players \( BQ_h \), and we note \( E \equiv E' \), if and only if: \( P \) outputs in at least one of the two executions at the end of round \( r \), and if the state of \( P \) at the end of round \( r \) is the same in both executions (that is, both executions are indistinguishable for \( P \) up to \( r \)). Notice that by determinism, this automatically implies that \( P \) (or \( BQ_h \)) output in both executions, at the end of round \( r \), the same values in both executions. We say that two executions \( E \) and \( E' \) are linked, which we note \( E \equiv E' \), if and only if there is a chain of equivalent executions from \( E \) to \( E' \). It is then a straightforward consequence of Consistency that, if some player outputs in \( E \), then no player in \( E' \) can output a different value.

2.2 Roadmap

Dolev-Strong [23] prove that any leaderless deterministic synchronous consensus protocol has at least an execution in \( t + 1 \) rounds. The pivot of their proof is that, assuming instead that players always output within \( t \) rounds, then, starting from any execution in which all players are honest, considering any target player, it is possible to link the execution with one in which this target player is completely silent. From here they can change his input silently. Then apply the pivot in the other direction to make him honest again with this changed input. We are going to discuss how we can prove this pivot in our context, except that we cannot target the leader. This will be stated as Proposition 5 in §2.5, from which the proof of Theorem (B) will follow with no technicality. Except for the observation that \( E_0^{fast} \) is equivalent for \( P_{10} \), up to round 1, to the alternative execution \( E_1^{fast} \), in which: he would be isolated from honest players in subsequent rounds, while corrupted players would have equivocated and behaved towards other honest players as if they had input 1.

As for the pivot, it is proven by induction in [23]. First, they show equivalence of any two executions such as, in the former, all players are honest except one “target” player, in the last round only, where he does not send at least one message \( m \). And such as the latter execution differs by having this player send this message. We notice in Lemma 10 that this equivalence easily carries over our setting, even if the receiver is the leader. From this equivalence, they can iterate and consider a target player who was honest up to the end of the first round, then did not send at least a message \( m \) in the \( 2^{nd} \) round and was silent in the \( 3^{rd} \). They show how to link this with the same execution, with
Fig. 1. Main Theorem 1 (B): $E_0^\text{fast} \equiv E_1^{iso} \equiv E^{(10)}$
the difference that honest message $m$ is now sent. For this, [23] corrupt the receiver, then change one at a time the messages that the receiver sends in the last round, using the previous iteration. We adapt this strategy in Lemma 11, stated in the hardest case where the receiver is the leader, thanks to a quicker move of the Environment. However, the strategy would not work at all if the sender was the leader. The reason being that, otherwise, the proof would involve at least three consecutive executions in which the leader is malicious. And thus no output would be guaranteed within three rounds in the execution in the middle. This is the technical hurdle that makes direct application of [23] impossible: in their leaderless setting, players always output within a fixed number ($i$) of rounds. Then remains the last step of the recursion. It consists in creating a message sent in the first round by some malicious sender, as long as he is not the leader. Lemma 4 in §2.4 states the hardest case by far, which is when the receiver is the leader. This is achieved with a 6 moves strategy of the Environment, that leverages partial synchrony and the backup termination condition. We use this strategy to prove a formal generalization of (A) in §2.3, then show how to deduce directly Lemma 4 in §2.4. From which we deduce Prop 5.

2.3 Core technique, illustrated on Main Theorem 1 (A)

We want to disprove the assumption of worst case Initial honest leader termination in 2 rounds. Here we need only a total $t = 2$, so 7 players. The theorem will be proven if we can link an execution in which all players are honest and have input 0, with one in which they are honest with input 1. This follows from the following proposition, that enables to change one by one the inputs. For purpose of later use in the proofs of (B) and (C), we state it with: input values of arbitrary nature, and with a sharp corruption ratio of 2/7 (to leave room for other corruptions).

**Proposition 3.** Consider $n$ players among which $t$ are corrupted, with $t/n \geq 2/7$. Let $E$ any execution of a consensus with Weak Unanimity with normal case latency in 2 rounds, in which all players are honest and $GST = 0$, with fixed inputs $v_1, v_2, v_3, v_4, v_5, v, v_7$, and thus in which they all output within 2 rounds. Then for every fixed value $w$, it is linked to an execution $E'$ in which only the input $v$ of $P_6$ has been flipped to $w$, and in which all players are honest and in which $GST = 0$.

**Proof.** It is enough to prove it for $t = 2$ and $n = 7$. We deal with the hardest case, which is where the player $P_6$, whose input is to be flipped, is the leader. The statement is illustrated at the top of Figure 2.

In $E$, the green diamond around $P_6$ denotes both: that $P_6$ is the leader, and that he is honest.

**$E^{(i)}$** We have still $GST = 0$ (circled in green below the diagram). Corrupted players $P_1$ and $P_2$ do not send any message to $P_6$ in round 2. So this is still equivalent to $E$ from the point of view of $P_5$ up to the end of round 2. Since $GST = 0$ and the leader $P_6$ is still honest, then $P_3$ outputs at the end of round 2. This event where $P_3$ outputs is denoted by the small green circle around him.

**$E^{(ii)}$** GST never happens. $P_5$ and $P_6$ are corrupted (the red squares around them) although $P_6$ is still the leader (thus the plain line for the square). $P_6$ equivocates in the first round: he reports input $w$ to $P_1$ and $P_2$ (dashed red arrows), while still reporting input $v$ (faded blue lines) to the other players, as in execution $E^{(i)}$. Also, he provides $P_5$ with two conflicting messages: one (plain blue) as in execution $E^{(i)}$, reporting input $v$, and the other one (red dotted) reporting that he has input $w$. These two parallel messages enable $P_5$ to also equivocate along with $P_6$ in the second round. Precisely, $P_5$ and $P_6$:

- towards the group $BQ = P_1, P_2, P_3, P_4, P_5$ "backup quorum", they send messages (the dotted red lines) compatible with an execution in which $P_6$ is honest with input $w$ from the beginning. Of course players in $BQ$ "see" that $P_6$...
Fig. 2. Proposition \( \mathcal{E} \equiv \mathcal{E}' \), beginning of proof: \( \mathcal{E} \equiv \mathcal{E}^{(i)} \equiv \mathcal{E}^{(ii)} \ldots \).
is equivocating, since they still receive conflicting messages from \( P_6 \) in the first round reporting input \( v \) (those messages are in faded blue, since they are unchanged from \( E^{(i)} \)).

- towards \( P_7 \) they still behave as in \( E^{(i)} \) in which \( P_6 \) is honest with input \( v \).

Finally, the messages from \( P_1 \) and \( P_2 \) to \( P_7 \) are delayed forever from the beginning of round 2. In particular, the whole is thus equivalent for \( P_7 \) to \( E^{(i)} \).

In \( E^{(iii)} \) on top left of Figure 3: Now GST = 0, so the messages from \( P_1 \) and \( P_2 \) to \( P_7 \) in the second round are timely delivered (thick red dotted arrows). on the one hand \( P_3 \) is now honest and receives only message from \( P_6 \), which is the one reporting \( w \). Conversely, \( P_7 \) is now corrupted, such that now he simply does not send any message from the second round, forever. All other messages are unchanged. Recall that these messages (in faded colours) are equivalent to \( E^{(ii)} \) up to \( \infty \) from the point of view of the subgroup of players in the "backup quorum" \( BQ \) which are honest in both \( E^{(iii)} \) and \( E^{(iv)} \), that is: \( \{ P_1, P_2, P_3, P_5 \} \). For later purpose, we actually remove \( P_3 \) from this subgroup and define instead the smaller:

\[
BQ_h := \{ P_1, P_2, P_5 \} \ .
\]

since GST = 0, these honest players \( BQ_h \) ultimately output. This execution, along with the next, is the core of the argument.

\( E^{(iv)} \) parallels \( E^{(i)} \), where know \( P_4 \) is the corrupted "accomplice" of \( P_6 \). GST never happens. \( P_5 \) is still the leader (thus the plain line for the square). \( P_6 \) equivocates in the first round: he still reports input \( v \) to \( P_3 \) (faded blue line), but apart from this, reports \( w \) to other players (faded dashed red arrows), in particular to the honest \( P_7 \), whose messages are delayed forever since the second round. Also, he provides \( P_4 \) with two conflicting messages: one (plain blue) as in execution \( E^{(i)} \), reporting input \( v \), and the other one (red dotted) reporting that he has input \( w \). This two parallel messages enable \( P_3 \) to also equivocate along with \( P_5 \) in the second round:

- towards the subgroup \( \{ P_1, P_2, P_3, P_5 \} \) of honest players in \( BQ \), they send messages (the dotted red lines) identical to the ones in the previous execution \( E^{(iii)} \). In particular, the whole is thus equivalent \( E^{(iv)} \) up to infinity for \( BQ_h := \{ P_1, P_2, P_4 \} \).
- towards \( P_7 \) they send messages compatible to an execution in which \( P_5 \) is honest with input \( w \).

Finally, the messages from \( P_3 \) to \( P_7 \) are delayed forever from the second round. [Thus in particular, notice that now \( P_7 \) receives only messages compatible with a history in which \( P_5 \) is honest].

\( E^{(v)} \) GST = 0, only \( P_3 \) is corrupted and does not send any message to \( P_7 \) in the second round. This is thus equivalent to \( E^{(iv)} \) for \( P_7 \). Leader \( P_6 \) is now honest (the green diamond) and has input \( w \). In particular, \( P_7 \) outputs at the end of the second round (the small green circle).

\( E^{(vi)} \) GST = 0 and \( P_3 \) now sends honest messages also to \( P_7 \), and is overall honest. This is equivalent to \( E^{(v)} \) for \( P_1 \). The whole execution is actually equal to \( E' \) by definition.

\( \Box \)
Fig. 3. End of proof of Proposition 3: $E(iii) \equiv E(iii) \equiv E(iii) \equiv E(iii) \equiv E'(i) (= E')$
2.4 Deducing Lemma 4

**Lemma 4.** (depicted at the bottom of Figure 4) Execution $E := \mathcal{H}(P)$ is linked to execution $E'$ where the corrupted player $P$ sends one correct message in the first round, to leader (denoted $P_8$), as if it were honest and had input $v$, and sends no other message otherwise.

**Proof.** Consider the honest players of executions $E$ and $E'$ at the beginning of round 2, then the result for $t \geq 9$ follows from applying Proposition 3 to the set of input values defined as their states at the beginning of round 2. Let us justify why:

- The state of all honest players except the leader is unchanged between both executions.
- Furthermore the state of the leader is only changed by a quantity: the message that he received from $P$, which is independent of the leader’s status in the first round (equal to his input, which remains unchanged). This quantity is instead fully controlled by the Environment, which can indeed issue instantaneously a message from $P$ to the leader at the beginning of round 2.

The Environment uses one extra corruption for $P$, so that the remaining corruption rate available to apply Proposition 3 is greater or equal to $(9 - 1)/(3 \times 9 + 1) = 8/28 = 2/7$, which is enough □

For sake of optimality we give a direct proof of Lemma 4 in §2.6, which enables us to include cases as small as $t = 3$ and $n = 10$, as stated in Proposition 5.

2.5 End of proof of Main Theorem (B)

The following proposition states that the intermediary results of [23] do hold (although with different proof techniques), except to the leader. It is depicted on Figure 5.

**Proposition 5.** Consider any plain consensus with Weak Unanimity protocol for $n = 3t + 1 \geq 10$ players, with normal case latency in three rounds. Let $P$ be any player, for instance $P := P_{10}$, and consider any execution $\mathcal{H}(P)$ in which $P_{10}$ is corrupted and completely silent, but where otherwise the other players have arbitrarily fixed initial input values: $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9$, are honest, including the leader $P_9$, where GST = 0, and thus in which all honest players output within three rounds. Then for any input value $v$, $\mathcal{H}(P)$ is linked to execution $\mathcal{H}(P')$ in which $P = P_{10}$ is also honest with input $v$.

Like in [23], the proof proceeds by linking $\mathcal{H}(P)$ to executions in which one adds, one by one, correct outgoing messages from $P = P_{10}$ to every other player, starting from the first round, to the third round. Thanks to our previous technical results, we can now carry out this program. We apply first the Lemma 4, which links an execution with: no message sent, to: an execution with one message sent in the first round to the leader. To be sure, the Lemma 4 still holds (with the proof unchanged) for a starting point execution where $P_{10}$ sends messages (to other players than the leader) in the first round.

Coming back to the proof of Theorem 1 (B). On Figure 1: all players are honest with input 0, GST = 0, so they all output 0 at the end of the first round by Optimistically fast output. The goal is to show equivalence with the execution $E_1$ on the right with all inputs equal to 1, and thus output 1.

$E_{iso}^{last}$ on the bottom left of Figure 1: player $P_{10}$ is honest but isolated: messages sent by $P_{10}$ are never delivered. This is possible since GST does not happen anymore. Still, we have equivalence with $E_{iso}^{last}$ from the point of view of $P_{10}$ up to the first round, in which he thus outputs. Players $P_8$, $P_7$ and $P_6$ are dishonest, while $P_5$ is still designated as leader.
We note $\mathcal{P}_E$ which is the end of phase B). The proof consists in linking execution while as having $1$

Throughout, we denote the subset of messages sent before GST which are furthermore delivered, even after GST, as $\mathcal{E}_0$.

The Environment follows exactly the same strategy as in Prop 3, where now the input of the leader $P_0$, which is his state at the end of round 1, is flipped by whether or not he received a message from $P_{10}$. We go over the strategy again in Figures 6,7 and 8.

2.6 Direct proof of Lemma 4 for a tight $n = 10$

The Environment follows exactly the same strategy as in Prop 3, where now the input of the leader $P_0$, which is his state at the end of round 1, is flipped by whether or not he received a message from $P_{10}$. We go over the strategy again in Figures 6,7 and 8.

2.7 Proof of Theorem 2 (C)

Throughout, we denote the subset of messages sent before GST which are furthermore delivered, even after GST, as lost forever. Recall that this is made possible by the new assumption that we made for Theorem 2 (the model of [26, §3.1]).

We note $\mathcal{P}$ the set of players, "phase A" the first three rounds and "phase B" the next three rounds (from 4 to 6). We suppose by contradiction that in all executions in which GST $\leq 4$, i.e., in which GST happens before the end of round 3, i.e., before the beginning of phase B, and in which a “late” honest leader $L_B$ is designated from the end of round 3, i.e., from the beginning of round 4; then all players output within three rounds later (that is: by the end of round $3 + 3 = 6$, which is the end of phase B). The proof consists in linking execution $\mathcal{E}_0^{late}$ (on top of Figure 9) where all players output 0 by the end of round 6, with execution $\mathcal{E}_1^{late}$ (on top of Figure 9) where all players output 1 by the end of round 3. Let us detail them.

$\mathcal{E}_0^{late}$: all players are honest and have input 0, thus 0 is the only value that can be output. From the beginning of round 3 we have GST and a honest leader: $L_B$ so by assumption every player outputs by the end of round 6. Let us specify more $\mathcal{E}_0^{late}$ for later use in the proof. Let us “$L_A$” the leader during phase A. We assume that no message is delivered during phase A. We call "phase B" the next three rounds.

$\mathcal{E}_1^{late}$: in phase A all players including the leader $L_A$ are honest, every message sent by players except $L_B$ ($\mathcal{P}\backslash L_B$) is delivered to $\mathcal{P}\backslash L_B$, and all players except $L_B$ ($\mathcal{P}\backslash L_B$) have input 1. Then GST holds from the beginning of round 4 (phase B). In particular, from the point of view of $\mathcal{P}\backslash L_B$, the execution is equivalent up to round three from the alternative one $\mathcal{E}_1^{late}_{corr}$ in which: GST holds from the beginning and $L_B$ is corrupted and never sends any message. So every honest player must output by the end of round 3. Furthermore we have in this alternative execution $\mathcal{E}_1^{late}_{corr}$ that every player that sends messages is honest and has input 1, so by the additional assumption 1 is the only allowed output value.

Equivalence of $\mathcal{E}_0^{late}$ and $\mathcal{E}_1^{late}$ follows from the possibility to flip the input of each player, except $L_B$, such that furthermore his messages are delivered in phase A (like in Proposition 3). Precisely we have

**Proposition 6.** For any player $P$ except $L_B$, consider any execution $\mathcal{E}^{(P)}$ ($\mathcal{E}^{(2)}$ on Figure 10) such that: all players are honest, GST holds from the beginning of round 4 and $L_B$ is leader from this point, some players in $\mathcal{P}\backslash \{P, L_B\}$ have input 1 ($\{P_1\}$ on Figure 10), the messages of these players are delivered to all except to $L_B$ during the first three rounds (phase A),
Fig. 4. Main Theorem (B) (end): $E^{(10)} \equiv \cdots \equiv E^{(1)}$ (as in Lemma 4) and Lemma 4: $E \equiv E'$.
Fig. 5. Proposition 5: $H_{\text{top}} \equiv H$. 

$H_{\text{top}}$ to all $\{\text{all}\}$
The Latency Costs of Optimism and of Strong Unanimity

Fig. 6. Proof of Lemma 4 (first)
Fig. 7. Proof of Lemma 4 (second)
Fig. 8. Proof of Lemma 4 (third and last)
while the other players, including $L_B$, still have input 0 and all their messages are lost in phase A. Then $E^{(P)}$ is equivalent to $E^{(P')}$ (bottom of Figure 10), in which messages sent by $P$ in phase A are now also delivered, except those to $L_B$ which are still lost forever.

The Proposition, in turn, follows from the (backwards) recursion of [23]. The new technical difficulty being that in our case, replacing a message sent to $L_B$ in round 4 requires the new technique used in Lemma 4 (which we recall is the same strategy as for the case described in the proof of Proposition 3). For completeness let us recall the general strategy of the recursion, then follow it step by step. The goal it to, first, corrupt $P$ and suppress all the messages that he sends (so in our case: sent in the last three rounds). Then for every $i$ from 1 to 6: create one by one outgoing messages from $P$ in round $i$, such that $P$ has, up to round $i$, the behavior of a honest player which would have input 1 and all his messages delivered (except those sent to $L_B$ in phase A). Let us now follow the recursion step by step.

- Suppressing or creating a message sent in round 3 from any player $R \in P \setminus \{L_B\}$ to any player $S \in P \setminus \{L_B\}$ requires modifying all messages sent by $R$ in round 4, and by all players in subsequent rounds. This is handled by Proposition 3 and requires 3 corruptions (including $S$), in addition to the corruption of $R$ (and of $P$ which is always assumed).
- Suppressing or creating a message sent in round 2 from any player $Q \in P \setminus \{L_B\}$ to any player $R \in P \setminus \{L_B\}$ requires modifying all messages sent by $Q$ in round 3, and by all players in subsequent rounds. This is handled by the previous case of the recursion and requires 4 corruptions (including $R$), in addition to the corruption of $Q$ (and of $P$ which is always assumed).
- Suppressing or creating a message sent in round 1 from player $P$ to any player $Q \in P \setminus \{L_B\}$ requires modifying all messages sent by $Q$ in round 2, and by all players in subsequent rounds. This is handled by the previous case of the recursion and requires 5 corruptions (including $Q$), in addition to the corruption of $P$ which is always assumed).

In addition to the $6 = 5 + 1$ corruptions needed by the recursion, we also corrupt $L_B$ in the equivalent execution $E^{late corr}_1$. This explains the sufficient condition $t \geq 7$ in the statement.

3 EASY LEMMAS

The following compiles Strong Unanimity into a fast track with no latency overhead

**Lemma 7.** Consider a consensus protocol with Strong unanimity. Then the following additional instructions enable to obtain a Fast Track in 1 round without harming the overall latency:

Players multicast their inputs in the first round. Upon receiving $n = 3t + 1$ such reports for the same value $v$, a player outputs $v$.

**Proof.** Safety: if some player fast outputs $w$, then he must have received $n = 3t + 1$ reports for this value $w$, which proves that all honest players have input $w$. Thus by Strong Unanimity, no honest player can output a value different from $w$. Liveness (Optimistic output in one round) is obvious. □

In the subclass of protocols with linear number of messages complexity, then we have the following variant of Lemma 7:

**Lemma 8.** Consider a protocol for consensus with Strong unanimity. Then the following additional instructions enable, in addition, Optimistically fast output in 2 rounds:

Players send their inputs in the first round to the leader. Upon receiving reports for the same value $v$ from all the $n = 3t + 1$ players, the leader forwards this set of signed messages to the players. On reception of $3t + 1$ such signed messages for an identical value, a player outputs $v$. 

The Latency Costs of Optimism and of Strong Unanimity

Fig. 9. Statement of Main Theorem 1 (C) $\epsilon_0^{\text{late}} \equiv \epsilon_1^{\text{late}}$
Fig. 10. Statement of Proposition 6 $E^{(2)} \equiv E^{(2)'}$.

Messages from those with input 1 (e.g., $P_1$) are delivered, $L_B$ still isolated.

$P_2$ has now input 1 and his messages are now delivered, still excepted to $L_B$.
The following shows that any isolated "Backup Quorum" of $2t+1$ honest players ultimately output in some finite extension of any finite execution.

**Lemma 9.** Consider a round-by-round consensus with weak unanimity under partial synchrony. Suppose that we have the Liveness condition that, if synchrony holds from some point in an execution, then, players output in finite time as soon as a honest leader is designated. Then for any (partial) execution $E$, and any set $BQ$ of $2t + 1$ (or more) honest players, there exists an execution extending $E$ (a "suffix"), in which all players in $BQ$ receive no further incoming message from the other $t$ players $P \setminus BQ$, and where they all output.

**Proof.** Either synchrony (from GST) already holds before the end of $E$, which then means that we have all the $t$ remaining players which are corrupted and did not send any further message to $BQ$. Or GST does not hold yet. Then consider all messages sent to $BQ$ in $E$ but not yet received. The common view in $E$ of all players in $BQ$, is undistinguishable from $E'$. Now, there exists an extension $Ext(E')$ in which all players in $BQ$ output. This can be achieved by setting GST from the end of $E'$ (if it was not already set before this point), then having a honest leader elected (so, among $BQ$). Coming back to $E$ (if not equal to $E'$), let us extend it into $Ext(E)$ by: delaying these messages to $BQ$ for a sufficiently long time, and, within $BQ$, extend it as in $Ext(E')$. Then we have that the two extensions coincide for players in $BQ$. Thus they also output in $Ext(E)$. □

In the setting of Theorem 1, the following allows to replace any message sent in the last round. As in §2, we consider a consensus with Weak Unanimity protocol where players send their full state in each round, and such that output is guaranteed in 3 rounds under normal conditions.

**Lemma 10.** Consider $4 = 3 + 1$ players playing the protocol, and an execution $H$ which is synchronous from the beginning (GST= 0), and all players are honest, except some player $P_4$ which behaves dishonestly in the last (3rd) round only, by not sending at least some message $m$. Then the execution is equivalent, for some honest player, to the execution $H'$ which differs only by $P_4$ sending $m$ in the 3rd round.

**Proof.** In $H$ the leader is honest and GST holds from the beginning so the honest player $P_4$ outputs by the end ot the 3rd round. At this point, his state is unaffected by the sending or not of $m$ (also in the 3rd round). So the execution is equivalent to $H'$ for $P$ up to the point where he outputs (end of 3rd round), which was to be proven. □

**Lemma 11.** Consider $7 = 5 + 2$ players playing the protocol, and an execution $G$ which is synchronous from the beginning (GST= 0), and all players are honest, except some player $P_7$ which behaves dishonestly in the 2nd round, by not sending at least some message $m$. Then in the 3rd round, by being completely silent. Then, the execution is equivalent, for some honest player, to the execution $G'$ which differs only by $P_7$ sending $m$ in the 2nd round.

**Proof.** We deal with the hardest case where the receiver of $m$, say $P_6$, is the leader. In $G$ the leader is honest and which is synchronous from the beginning (GST= 0) so the honest player $P_1$ outputs by the end of the 3rd round. We now consider:

$G_1$: Is still is synchronous from the beginning (GST= 0), which differs by $P_1$ sending $m$ to $P_6$. Then in the 3rd round, $P_6$ behaves honestly towards all players except $P_1$, to which he sends a message as if he did not receive $m$ at the end of the 2nd round. Both executions are equivalent from the point of view of $P_1$, who thus still outputs. Also, we have $P_2$ who outputs in $G_1$ since from his perspective the leader $P_6$ is honest.
$G'$: Differs only from the previous by the leader sending a honest message to $P_1$, thus behaving honestly towards every player. We have even the leader is honest, by definition of $G'$, so all players output in it. In particular $P_2$, for whom $G_1$ and $G'$ are equivalent. □

4 FOR PROTOCOLS WITH SUBQUADRATIC COMMUNICATION COMPLEXITY

Consider star-shaped communication pattern. Those where players send and receive messages only from the leader. Then in short, the proof in this setting follows from emulating the previous strategy of Prop 3. Then, the analogous of Lem 4 will follow as in §2.4, or, can be proven directly with the same method as in §2.6. For instance, the additional +1 to the corruption bound comes from the fact that, for instance in execution $E_i$ of Figure 2, messages of $P_1$ and $P_2$ to $P_7$ are now supposed to go through the leader. So to emulate their silence towards $P_7$, we now have the leader equivocate and act towards $P_7$ as if he did not hear of $P_1$ and $P_2$, while acting towards other players as if he did.

Let us give the details for emulating Prop 3. We are considering a protocol that terminates in four rounds under normal conditions, in which instructs players to send their full state to the leader in each round, and the leader to send his full state to players. Consider every execution used in the proof of Prop 3. We emulate communications in each round of these executions as follows (and likewise for the second round). Each round maps to a block of two rounds in the star shaped protocol (either the first two, or the last two). In a block of two rounds, we have two threads of messages in parallel.

- The first thread is leader-to-all in the 1st round, then-all-to leader in the 2nd. We use this thread to have the leader send in the 1st round, to players, exactly the same state as in the proof of Prop 3. Like in this proof, the leader may thus in, particular equivocate by running in parallel two honest executions, and thus report two conflicting internal states to different honest players. Then in the 2nd round, honest players simply send their state to the leader, updated by reception of the message in the 1st round. Informally, this plays no role since the leader “knows” what he sent to them.
- The second thread is then-all-to leader in the 1st round, and leader-to-all in the 2nd round. As exemplified above with $P_1$ and $P_2$ to $P_7$, we have the leader relay in this thread exactly the messages sent directly from players to players in the proof of Prop 3

Consider general patterns, with linear message complexity: the claim is that the proof falls back to the star shaped case, provided a number of players large enough. Indeed, only $o(n)$ players receive $\Omega(n)$ messages throughout the protocol. We can thus corrupt them, except the leader. The rest of players, except the leader, receive $o(n)$ messages. Then, notice that the argument, e.g. of Proposition 3 carries over the general $n$ setting with the same fixed number of consecutive executions, in which it is only important that two specific “pivot” players output (in Prop 3: $P_3$ then $P_7$ then $P_1$). We can thus choose this fixed set of pivots among the players receiving $o(n)$ messages, then silence the $o(n)^R = o(n)$ players from which they hear of (that is: receive signatures) before they output, except of course the leader in executions where he is assumed honest. By “silence players” we mean either corrupting them, or delaying forever their messages in executions in which synchrony does not hold.

REFERENCES

The Latency Costs of Optimism and of Strong Unanimity


