BAR Primer

Allen Clement[1*], Harry Li[1], Jeff Napper[1], Jean-Philippe Martin[2], Lorenzo Alvisi[1], Mike Dahlin[1]
[1]University of Texas at Austin, [2] Microsoft Research Cambridge
[1]{aclement, harry, jmn, lorenzo, dahlin}@cs.utexas.edu, [2]{jpmartin@microsoft.com}

Abstract

Byzantine and rational behaviors are increasingly recognized as unavoidable realities in today's cooperative services. Yet, how to design BAR-tolerant protocols and rigorously prove them strategy proof remains somewhat of a mystery: existing examples tend either to focus on unrealistically simple problems or to want in rigor. The goal of this paper is to demystify the process by presenting the full algorithmic development cycle that, starting from the classic synchronous Repeated Terminating Reliable Broadcast (R-TRB) problem statement, leads to a provably BAR-tolerant solution. We show i) how to express R-TRB as a game; ii) why the strategy corresponding to the optimal Byzantine Fault Tolerant algorithm of Dolev and Strong does not guarantee safety when non-Byzantine players behave rationally; iii) how to derive a BAR-tolerant R-TRB protocol: iv) how to prove rigorously that the protocol ensures safety in the presence of non-Byzantine rational players.

1 Introduction

Cooperative services are increasingly popular distributed systems in which nodes belong to multiple administrative domains (MADs). Examples of MAD distributed systems include Internet routing [8, 17], wireless mesh routing [13] and peer-to-peer services [2, 5].

Designing dependable MAD services turns out to be, quite appropriately, maddening. In these systems, nodes may deviate arbitrarily from their specification because they are broken (from bugs, hardware failures, configuration erros, or even malicious attacks). Nodes may also deviate because they are rational, i.e., selfishly intent on maximizing their own utility. Today, an administrator can turn his node(s) from obedient to rational with the few mouse clicks it takes to download and install a crafty modification. Any system deployed across multiple administrative domains needs to be designed for the possibility that *any* node may deviate for personal gain.

Under these circumstances, modeling rational nodes as Byzantine, though theoretically possible, may be pointless as the number of nodes classified as Byzantine quickly exceeds the threshold beyond which many distributed systems problems are unsolvable.

Aiyer et al. [3] propose the BAR model to explicitly distinguish rational from Byzantine behavior. Within this model they classify nodes as Byzantine, altruistic, or rational—hence BAR. Unlike Byzantine and rational nodes, altruistic ones always obey the protocol. Aiyer et al. use this classification to design and implement a replicated state-machine that is *BAR-tolerant*, i.e., resilient to both Byzantine faults and rational manipulation.

Subsequent works have also explicitly addressed rational behavior as distinct from Byzantine in epidemic broadcast [12], virus protection [15], and secret sharing [1]. Although each work has made advances in combining game theory with distributed systems, the entire process of designing protocols for MAD systems and rigorously proving them correct remains a dark art to the uninitiated. Existing works either focus on building MAD systems while only providing high-level proof sketches or emphasize the mathematical rigor for each proof but address over simplified problems.

Furthermore, despite claims that rational actions cause problems in existing Byzantine fault-tolerant protocols [3], there is no documented example of such a chain of events. It seems hasty to build a new class of protocols before understanding the shortcomings in current ones.

This work exposes the entire process of designing a BAR-tolerant protocol. We anchor our discussion to the classic terminating reliable broadcast (TRB) problem [11]. Specifically, we analyze protocols that solve a sequence of TRB instances as would be used for state-machine replication. This paper makes the following contributions:

- Formalizes the Repeated TRB (R-TRB) problem as a game and characterizes the classic Dolev-Strong protocol [6] as a strategy within that game;
- Demonstrates that Dolev-Strong is not an ex ante Nash equilibrium, i.e., a rational player expects to benefit by unilaterally deviating from the protocol;
- Shows that if more than one rational node follows the deviant strategy, then safety can be violated;

- Derives a new TRB protocol, Just TRB, based upon Dolev-Strong that demonstrates techniques to address the shortcomings of traditional Byzantine algorithms in a MAD environment;
- Proves that Just TRB solves R-TRB and is an ex ante Nash equilibrium.

We organize the rest of this paper as follows. In the next section, we frame our contributions within the existing literature. In Section 3, we cast R-TRB as a game and introduce notational conventions. Section 4 reviews the Dolev-Strong protocol and defines a deviant strategy that ultimately leads rational players to jeopardize safety. Sections 5 and 6 describe Just TRB and proves it is a Nash equilibrium, respectively. We conclude in Section 7.

2 Why a Primer?

Although a number of works combine Byzantine fault-tolerance with game theory, a detailed look at those works leaves one wanting more. The current literature is missing step-by-step examples of the *process* of creating a correct BAR-tolerant protocol. Existing approaches fall into two broad categories. The first focuses on developing a theory to formally reason about rational behavior in the presence of Byzantine activity. The second emphasizes building systems that tolerate Byzantine and rational participants. Although both make important contributions, neither shows the process in its entirety, namely, *i)* formalizing a protocol as a game, *ii)* showing how players can cheat and violate correctness conditions in the game, *iii)* designing a new BAR-tolerant protocol, and *iv)* analyzing the new protocol. In the rest of this section, we summarize related work, highlighting strengths and weaknesses of previous approaches to make clear where this paper fits.

To the best of our knowledge, Eliaz [7] is the first to address rational behavior in the presence of Byzantine activity. He defines a k-Fault Tolerant Nash Equilibrium (k-FTNE) as a situation in which no player benefits from unilaterally deviating despite up to k players behaving in arbitrary ways. Eliaz applies this concept to the constrained Walrasian function that is used in auctions. It is unclear whether the Walrasian function is applicable to a broader range of distributed problems.

More recently, Aiyer et al. [3, 14] introduce the BAR model to reason about systems with Byzantine and rational participants. They design a cooperative backup system based around a BAR-tolerant replicated state machine. Aiyer et al. recognize that in many situations, consuming bandwidth incurs cost, and so, design their protocols to curb rational deviations that may benefit from using less bandwidth. Li et al. [12] also use the BAR model, but design a peer to peer live streaming application based around a BAR-tolerant gossip protocol. Although both works prove game theoretic properties, the emphasis is on building a reasonably practical system. As such, the proofs contained in both papers remain at a high-level, sometimes appealing to intuition instead of exposing the mathematical foundation.

Moscibroda et al. [15] complement the BAR work by formalizing a Byzantine Nash equilibrium and rigorously analyzing a game in the context of that equilibrium. They examine a virus inoculation game in which Byzantine and rational players independently decide whether to inoculate against a virus that will eventually be inserted at random into the system. Inoculations are not free, but being infected is far worse. Moscibroda et al. show *i)* how to reason about what rational participants expect to happen when Byzantine participants seek to maximize infections and *ii)* how to quantify the increased cost because of Byzantine actions. The work makes valuable contributions in establishing a theory to reason about Byzantine and rational players. However, it is unclear how to extend the rigorous analysis of the inoculation game to situations, such as state machine replication protocols, that have correctness criteria.

Abraham et al. [1] propose (k, t)-robustness, extending Eliaz's work to accommodate rational players who collude. They design a secret sharing protocol and prove that it is (k, t)-robust, meaning that it is correct despite up to k colluding rational players and t Byzantine players. Although correct, their proof assumes that communication costs are zero, an assumption at odds with the fact that in real life bandwidth is not free. More broadly, Clement et al. [4] show that if communication is not free, then only trivial fault-tolerant distributed systems can be (k, t)-robust.

3 R-TRB Meets Game Theory

In the rest of this paper, we consider the Repeating Terminating Reliable Broadcast (R-TRB) problem. The R-TRB problem consists of an infinite sequence of TRB instances in which a non-Byzantine node is leader an infinite number of times. We say that a protocol fulfills the *functionality* $\mathcal{F}_{TRB}$ of R-TRB if the protocol guarantees the safety and liveness properties below for every TRB instance.

TRB1 Validity. If a non-Byzantine leader broadcasts value v, no non-Byzantine process delivers $v' \neq v$.

TRB2 Integrity. Each non-Byzantine process delivers at most one value, and if it delivers $v \neq$ *sender faulty* (**SF**) then the leader broadcast v.

TRB3 Agreement. No two non-Byzantine processes deliver different values.

TRB4 Termination. Each non-Byzantine process eventually delivers a value.

We address the R-TRB problem using authenticated messages [10] and synchronous and reliable communication channels. Further, we assume at most f processes are

Byzantine and act arbitrarily. The remaining processes are either rational or altruistic; that is, they act to maximize personal gain or obey the prescribed protocol, respectively.

We describe a distributed system as a game $\Gamma = (\mathcal{N}, \mathcal{S}_\mathcal{N}, \mathcal{U})$. The players $\mathcal{N}$ correspond to the processes in the system; we let $\mathcal{B} \subset \mathcal{N}$ represent the set of Byzantine players. The *strategy space* $\mathcal{S}_\mathcal{N}$ denotes the set of protocols or *strategies* available to each player $p \in \mathcal{N}$. *Strategy profile* $\vec{\sigma}_\mathcal{B}$ assigns a strategy σ_p to each player $p \in \mathcal{B}$. For notational simplicity $\vec{\sigma} = \vec{\sigma}_\mathcal{N}$. Note that σ_p and σ_q are not necessarily identical. The set $\mathcal{U}$ contains a *utility function* $u_p(\vec{\sigma})$ for each $p \in \mathcal{N}$ that maps a strategy profile[1] to the utility p receives when every player plays their component of $\vec{\sigma}$. The utility p receives is $\textit{benefits}_p(\vec{\sigma})$ minus $\textit{costs}_p(\vec{\sigma})$ where the game's outcome decides the benefits and p's actions determine the costs.

We assume that rational players incur costs for sending protocol messages. Player p's cost in TRB instance k is

$$\textit{costs}_p^k(\vec{\sigma}) = \sum_{m \in \text{sent}_p^k(\vec{\sigma})} c_{snd}(m)$$

where $\text{sent}_p^k(\vec{\sigma})$ is the set of messages player p sent and $c_{snd}(m)$ is the cost of sending message m. We assume that large messages cost more to send than smaller ones. The costs incurred by p when playing R-TRB are the sum of the costs of each individual instance.

The benefits received by rational player p depend on which TRB properties hold and whether or not p is the leader. When p is the leader, if **TRB2–4** hold then p benefits by ϖ from delivering the appropriate value. If **TRB1** also holds, then p receives an additional β benefit for proposing the delivered value. When p is not the leader and **TRB2–4** hold, then p benefits by ϖ as before. In any other case, p does not benefit. Similar to costs, p's benefit in playing R-TRB is the sum of the benefits received in each TRB instance.

We assume that rational players follow protocols if there is no *expected* benefit from unilaterally deviating; that is, the protocol is an *ex ante* Nash equilibrium [16]. Formally,

$$\forall p \in \mathcal{N}, \forall \phi_i \in \mathcal{S}_p : \bar{u}_p(\vec{\sigma}) \geq \bar{u}_p(\vec{\sigma}_{\mathcal{N}-\{p\}}, \phi_i)$$

where $\bar{u}_p$ is a function that models the utility p expects when playing a given strategy.

In this work, we choose to model risk averse players who act to maximize the worst case utility. Formally,

$$\bar{u}_p(\vec{\sigma}) = \min_{\mathcal{B} \subseteq \mathcal{N} : |\mathcal{B}| \leq f} \circ \min_{\vec{\tau}_\mathcal{B} \in \mathcal{S}_\mathcal{B}} \circ u_p(\vec{\sigma}_{\mathcal{N}-\mathcal{B}}, \vec{\tau}_\mathcal{B})$$

[1] Note that the strategy profile argument to the utility function is a set of strategies for all players that may be specified by a single argument ($\vec{\sigma}$) or as multiple sets of strategies such as ($\vec{\sigma}_{\mathcal{N}-\{p\}}$,$\phi_p$), which specifies ϕ_i for player p, and $\vec{\sigma}$ for everyone else.

```
1   Initialization for process p in instance k > 0:
2     leader := k mod |N|                          // Leader in this instance
3     extracted := ∅                               // Values extracted this instance
4     ∀i : 1 < i ≤ f + 1 : relay_i := ∅            // Messages to send in round i

7   Round 1, for p = leader, and value v:
8     extracted := {v}
9     R := N − {p}                                 [|R| = f + 1]
10    send ⟨VALUE, k, v⟩_p to q ∈ R

12  Round 1, for p ≠ leader:
13    when receive m = ⟨VALUE, k, v⟩_leader
14      if v ∉ extracted ∧ |extracted| < 2 then
15        relay_2 ∪= {m}
16        sigs_v := {leader}
17      extracted ∪= {v}

20  Round i, 2 ≤ i ≤ f for p:
21    foreach m = ⟨VALUE, k, v⟩_leader,...,s_{i−1} ∈ relay_i
22      R := N − sigs_v − {p}                      [|R| = min(n − 1, f + 1) − |sigs_v|]
23      send ⟨m⟩_p to q ∈ R
24    when receive m = ⟨VALUE, k, v⟩_leader,...,s_i
25      if v ∉ extracted ∧ |extracted| < 2 then
26        relay_{i+1} ∪= {m}
27        sigs_v := {leader, ..., s_i}
28      else if v ∈ extracted then
29        sigs_v ∪= {leader, ..., s_i}
30      extracted ∪= {v}

32  Round f + 1 for p:
33    foreach m = ⟨VALUE, k, v⟩_leader,...,s_f ∈ relay_{f+1}
34      send ⟨m⟩_p to q ∈ N − sigs_v − {p}
35    when receive m = ⟨VALUE, k, v⟩_leader,...,s_{f+1}
36      if v ∉ extracted ∧ |extracted| < 2 then
37        extracted ∪= {v}
38    if |extracted| = 1 then
39      deliver v ∈ extracted
40    else
41      deliver SF
```

Figure 1. Dolev-Strong protocol for instance $k > 0$. The lazy strategy is derived by further constraining the size of R at Lines 8 and 21 using the boxes on the right.

Intuitively, we define a rational player p's expected utility $\bar{u}_p$ by considering the worst configuration of Byzantine players and the worst set of strategies that those Byzantine players could take, assuming that all other non-Byzantine participants obey the specified strategy profile.

4 BFT $\not\Longrightarrow$ Incentive-Compatible

We now demonstrate that Byzantine fault-tolerance does not necessarily imply a natural resilience to rational deviations. In particular, we show in the classic Dolev-Strong (D-S) TRB protocol [6] how a rational player can benefit by shirking its responsibility of forwarding messages onto other players. Interestingly, such a deviation preserves the safety and liveness properties of TRB despite Dolev-Strong being a message optimal protocol. However, if more than one rational player takes that deviation, then safety is lost. We now describe the D-S TRB protocol for a TRB instance k and provide Figure 1 for reference.

A D-S TRB instance proceeds through $f + 1$ rounds. In round 1, the leader broadcasts a signed message m containing a value v to all players. A message m is *valid* for player

p_j in round i if m has the form $\langle \text{VALUE}, v, k \rangle_{p_1,\ldots,p_i}$ where v is a value, k is the instance number, p_1 is *leader*^{k}, the players' signatures $p_1, \ldots, p_i$ are unique, and p_j's signature is not in $\{p_1, \ldots, p_i\}$. In every round i, upon receiving a valid message m containing value v, player p adds v to its $extracted$ set that represents values that the leader sent. In round $1 < i \leq f$, a player considers each value v added to $extracted$ in the previous round, appends its signature to m where m is the message containing v, and relays m to all players who have not yet signed a message containing v. Note that a player is allowed to relay at most two messages in each TRB instance. In the last round, each player delivers v if v is the only value in $extracted$ and delivers **SF** otherwise. We denote $\vec{\delta}$ as the strategy profile in which each player obeys the D-S TRB protocol; δ_p denotes player p's strategy in $\vec{\delta}$.

We now consider a *lazy* strategy λ_r that rational player r can use as an alternative to δ_r. Strategy λ_r is similar to δ_r. The difference is that in round $i \leq f$, rather than relaying message m to *all* players as in δ_r, r sends m to a subset of $f + 1 - s$ players who to r's knowledge have not yet extracted v, where s is the number of players whom r believes to have extracted v.

By following λ_r, r shirks the responsibility of relaying messages onto other players. By forwarding a message to $f + 1 - s$ players instead of all other players who have not yet extracted v, r pushes some of the relaying work onto at least one other non-Byzantine player. That player then finishes the relay for r and thereby guarantees safety for all.

4.1 No Harm in Being Lazy

We now prove that a rational player r expects the same benefit from following the lazy strategy λ_r as from obeying the Dolev-Strong protocol δ_r. In the next section, we combine this proof with the observation that λ_r is no more costly than δ_r, and in some cases even cheaper, implying that r should expect greater utility from λ_r.

Remember that r's benefit is tied to the properties **TRB1–4**. We show that those properties continue to hold, thereby preserving r's benefit, despite r following λ_r while the remaining non-Byzantine players obey δ_r. Formally, we prove the following:

Theorem 1 (Lazy Safety and Liveness)**.** *For all $\mathcal{B} \subseteq \mathcal{N}$, $|\mathcal{B}| \leq f$, and $\forall \vec{\tau}_{\mathcal{B}} \in \mathcal{S}_{\mathcal{B}}$, if $\vec{\sigma} = (\vec{\delta}_{\mathcal{N}-\mathcal{B}-\{r\}}, \lambda_r, \vec{\tau}_{\mathcal{B}})$ is played for Γ_{TRB} then* **TRB1-4** *hold.*

We structure the proof of Theorem 1 into four parts corresponding to TRB1–4. Validity (**TRB1**) and Integrity (**TRB2**) are simple to prove, whereas Agreement (**TRB3**) is more involved. Termination (**TRB4**) is also easy to prove. For the lemmas in this section, we assume that non-Byzantine players follow strategy profile $(\vec{\delta}_{\mathcal{N}-\mathcal{B}-\{r\}}, \lambda_r)$ and $|\mathcal{B}| \leq f$. We present Validity and Integrity first.

Lemma 1 (Lazy Validity—**TRB1**)**.** *If a non-Byzantine leader broadcasts v, then no non-Byzantine process delivers $v' \neq v$.*

Proof. A valid message requires a signature by the leader. If a non-Byzantine leader broadcasts at most one value v, then no valid message can contain value $v' \neq v$. Therefore, $v' \notin extracted$ for any non-Byzantine player. □

Lemma 2 (Lazy Integrity—**TRB2**)**.** *Each non-Byzantine process delivers at most one value, and if it delivers $v \neq$* **SF** *then the leader broadcast v.*

Proof. Both strategies specify to deliver a single value only during round $f+1$. If $v \neq$ **SF**, then v was extracted in round $i \leq f + 1$. Extracted values come from valid messages, and a message is valid only if it contains the signature of the leader. As signatures are unforgeable, the leader broadcast a message containing v. □

The following four lemmas ensure a property called *relay*—essentially if any non-Byzantine player extracts a value v then all non-Byzantine players extract that same value (or two distinct values) by round $f + 1$.

Lemma 3. *If a player p following the Dolev-Strong protocol extracts v in round $i \leq f$ then all non-Byzantine players extract v or 2 values $v' \neq v''$ by round $i + 1$.*

Proof. Assume v is the first or second value extracted by p. Since p follows δ_p, p signs and forwards a valid message containing v during round $i + 1$ (guaranteed to exist since $i \leq f$) to all other players that p has not observed to sign a message containing v. Each of those non-Byzantine players extracts v in round $i + 1$. Finally, all non-Byzantine players that signed the message containing v followed the Dolev-Strong protocol and thus extracted v prior to signing the message.

If v is the third or higher value extracted by p, then it follows from the previous discussion that all non-Byzantine players extract at least two distinct values by round $i + 1$, completing the proof. □

Lemma 4. *If lazy player r extracts v in round $i < f$ then all non-Byzantine players extract v or 2 values $v' \neq v''$ by round $i + 2$.*

Lemma 5. *If lazy player r extracts v in round f then all non-Byzantine players extract v or 2 values $v' \neq v''$ by round $f + 1$.*

Lemma 6 (Lazy Relay)**.** *If a non-Byzantine player extracts v in round $i \leq f$, then all non-Byzantine players extract v or 2 values $v' \neq v''$ by round $f + 1$.*

Lemma 7 (Lazy Agreement—-**TRB3**)**.** *If non-Byzantine player p delivers v and non-Byzantine player q delivers v' then $v = v'$.*

Proof. Without loss of generality, assume that $v \neq$ **SF**. It follows from the protocol definition that p's $extracted$ set contains exactly one value, v, at the end of round $f+1$. Since p is non-Byzantine, p extracted v in round $i \leq f$. It follows from Lemma 6 that q extracted v or two distinct valued by round $f+1$. Since q delivers v' there are two cases to consider, either $v = v'$ or $v' =$ **SF**. In the former case the proof is complete. In the latter case, q delivered **SF** because $|extracted| \geq 2$ implying that q extracted some value $u \neq v$. Again, since there are at most f Byzantine players and non-Byzantine players only extract values contained in valid messages, some non-Byzantine player r' extracted u in round $j \leq f$. It then follows from Lemma 6 that p extracted u or 2 distinct values by round $f+1$. Since p extracted v, p extracted 2 distinct values by round $f+1$ and delivers $v =$ **SF** at the end of round $f+1$. This contradicts our initial assumption that $v \neq$ **SF**, so $v' = v$ completing the proof. □

Lemma 8 (Lazy Termination—**TRB4**)**.** *Each non-Byzantine process eventually delivers a value.*

Proof. Trivial. □

4.2 Rationality

In this section, we use the lazy strategy to demonstrate that D-S TRB is not a Nash Equilibrium. We also prove that the lazy strategy is not guaranteed to fulfill the functionality of TRB when it is played by all non-Byzantine players. We focus our attention on the setting in which $n > f+2$, $f > 1$, and $|\mathcal{B}| \leq f$—the conditions under which D-S TRB and the lazy strategy differ.

Theorem 2. *D-S TRB is not a Nash Equilibrium.*

Proof. It follows from Theorem 1 that **TRB1–4** hold in all instances when non-Byzantine players utilize the strategy profile $(\vec{\delta}_{\mathcal{N}-\mathcal{B}-\{r\}}, \lambda_r)$. Player r thus expects the same benefit from playing either λ_r or δ_r.

We now show that player r expects less cost from playing λ_r than from playing δ_r. When following either strategy in instances with a Byzantine leader, player r sends at most two messages to $n-2$ other players, resulting in identical expected costs.

When following δ_r in instances with a non-Byzantine leader, player r sends one message to $n-2$ other players (or one message to $n-1$ players if r is the leader). Meanwhile, when following λ_r in instances where the leader is non-Byzantine, player r sends one message to f other players (or one message to $f+1$ other players when r is the leader). Since $n > f+2$ it follows that r sends more messages by following δ_r than by following λ_r when the leader is non-Byzantine.

Since the leader is non-Byzantine infinitely often, player r expects less cost and thus higher utility from following λ_r. Hence $\bar{u}_r(\vec{\delta}) < \bar{u}_r(\vec{\delta}_{\mathcal{N}-\{r\}}, \lambda_r)$ and D-S TRB is not a Nash Equilibrium. □

The consequence of D-S TRB not being a Nash Equilibrium is that all rational players will choose to follow the lazy strategy instead. Unfortunately, this results in a tragedy of the commons scenario [9] as Agreement is not guaranteed.

Theorem 3 (Failed Agreement)**.** *If all non-Byzantine players follow strategy profile $\vec{\lambda}_{\mathcal{N}-\mathcal{B}}$ then* **TRB3** *can be violated.*

Proof. It is sufficient to show a scenario in which **TRB3** does not hold. Since $n > f+2$, there are at least three non-Byzantine players p, q, and r. Suppose r is the leader. In the first round, r sends the broadcast value $m = \langle \text{VALUE}, v, k \rangle_r$ to $f+1$ other players. Without loss of generality, assume this set contains player p and f other Byzantine players that never forward v. Since $f > 1$, there are guaranteed to be at least three rounds. In the second round, p should send $m' = \langle \text{VALUE}, v, k \rangle_{r,p}$ to $f+1$ other players. Because $sigs_m = \{r\}$, p need only send m' to f other players, which can include the set of Byzantine players chosen by r. These f players continue to not forward v in the third round, and thus non-Byzantine player q does not receive or extract v. In round $f+1$, q delivers **SF** while r and p both deliver v, violating **TRB3**. □

5 Just TRB

This section presents Just TRB, a R-TRB protocol based on D-S TRB and resilient to both Byzantine and rational players. For clarity, we structure our presentation into five parts.

1. Underscore the weaknesses in D-S TRB that a rational player exploits.
2. Discuss design principles that we use to control rational players.
3. Describe Just TRB and explain the mechanisms that curb rational deviations.
4. Show that Just TRB solves the R-TRB problem if all non-Byzantine players obey the protocol.
5. Prove that Just TRB is an ex ante Nash equilibrium.

The D-S TRB protocol has three shortcomings that a rational player r abuses by playing the lazy strategy. First, it is impossible for a player to determine whether r's silence in a round is because r has not recently extracted a message or because r is deviating from the protocol. Second, r increases its long-term utility by relaying messages to fewer players than prescribed. Third, there is no consequence for failing to send a message.

Design principles from earlier BAR works [3, 12] guide our solution. We impose a *predictable communication pattern* to aid players in detecting when another player has failed to send a message. We also *balance costs* across protocol messages to eliminate any long-term utility a player can gain by failing to send prescribed messages. Finally, we enforce *accountability* through a shunning mechanism that punishes players for deviating from the predictable communication pattern.

5.1 Just TRB Protocol

The Just TRB protocol is similar in structure to the D-S TRB protocol. Differences between the two reflect changes made when appyling the above principles. Each player in Just TRB maintains a status with other players. All players begin as *friends* with one another. If a player p observes q to have deviated from the predictable communication pattern, then p considers q an *ex-friend* and henceforth shuns q by not sending messages to q. Ex-friend q can be lowered further to *enemy* status if q causes p to do more work in any Just TRB instance. Ex-friends are tracked in the set $shun_p$ and enemies are tracked in the set $penDuring$. A player cannot make amends for past actions once it damages a relationship.

Each Just TRB instance proceeds through $f + 1$ rounds. In round 1, the leader broadcasts a signed message m containing a value v to its friends. A *valid* message has the same definition as in D-S TRB. In each round $i \leq f$, upon receiving a valid message m containing value v, player p adds v to its $extracted$ set only if $i \neq 1$ or the leader is not an enemy. The leader immediately becomes $p's$ enemy if p does not receive a valid message from the leader in round 1 because this forces p to send extra *penance* messages, to be discussed shortly, in the last round. In round $1 < i \leq f$, a player considers each value v added to $extracted$ in the previous round, appends its signature to m where m is the message containing v, and relays m to all friends. Note that a player relays at most two values in each TRB instance. We say that a message is *meaningful* if it contains a value.

In the last round, players may send *dummy* value messages, which contain no value. A player sends enough dummy messages to its friends so that each friend receives exactly two value messages (meaningful or dummy) in each Just TRB instance. A player also sends *penance* messages to all friends if the leader is an enemy. A player reduces a friend to an ex-friend if that friend fails to send the appropriate messages by the end of the instance. Finally, each player examines its $extracted$ set. If $extracted = \{v\}$ then a player delivers v and delivers **SF** otherwise. We provide Figure 2 for reference.

Just TRB shows the repeated application of our earlier principles. We impose a predictable communication pattern by requiring players to send exactly two value messages to friends in each TRB instance. Players hold their friends accountable for adhering to this pattern; if a player p receives fewer than two such messages from a friend q, then p considers q an ex-friend and shuns q.

We introduce penance messages to balance costs. Without penance messages in Just TRB, a rational player p could, without just cause, turn some friends into ex-friends to save costs by not sending them messages. This would expose Just TRB to the same weakness that plagued D-S TRB. Penances inoculate Just TRB: p will not frivolously turn q into an ex-friend because q, as the future leader of an infinite number of TRB instances, can force p to incur additional costs (via penances) during the last round of each of those instances. Section 6 establishes the inequalities nec-

```
1   Protocol initialization for process p:
2     shun_p := ∅                                   // Set of players that p shuns
3     foreach a ∈ N
4       penDuring[a] := ∅                           // Leaders where p expects penance from a
5       recvdSeq[a] := ∅                            // Messages p received from a

7   Initialization for process p in instance k > 0:
8     leader := k mod |N|
9     extracted := ∅ ; penance := ∅
10    ∀i : 1 < i ≤ f + 1 : relay_i := ∅


13  Round 1, for p = leader, and value v:
14    extracted := {v}
15    send ⟨VALUE, k, v⟩_p to q ∈ N − shun_p − {p}

17  Round 1, for p ≠ leader:
18    when receive m = ⟨VALUE, k, v⟩_leader
19      if leader ∉ penDuring[p] then
20        if v ∉ extracted ∧ |extracted| < 2 then
21          relay_2 ∪= {m}
22          extracted ∪= {v}
23      recvdSeq[leader] ∪= {m}
24    if extracted = ∅ then
25      penDuring[p] ∪= {leader} ; shun_p ∪= {leader}
26      penance := {⟨PENANCE, k, |penDuring[p]|, filler⟩_p}


29  Round i, 2 ≤ i ≤ f for p:
30    foreach m = ⟨VALUE, k, v⟩_leader,...,s_{i−1} ∈ relay_i
31      send ⟨m⟩_p to q ∈ N − shun_p − {p}
32    when receive m = ⟨VALUE, k, v⟩_leader,...,s_i
33      if v ∉ extracted ∧ |extracted| < 2 then
34        relay_{i+1} ∪= {m}
35        extracted ∪= {v}
36      recvdSeq[s_i] ∪= {m}

38  Round f + 1, for p:
39    if f > 0 then
40      foreach m = ⟨VALUE, k, v⟩_leader,...,s_f ∈ relay_{f+1}
41        send ⟨m⟩_p to q ∈ N − shun_p − {p}
42      if |extracted| < 2 then
43        send ⟨VALUE, k, ⊥_1⟩_p to q ∈ N − shun_p − {p}
44      if |extracted| < 1 then
45        send ⟨VALUE, k, ⊥_2⟩_p to q ∈ N − shun_p − {p}
46      if penance ≠ ∅ then
47        send m ∈ penance to q ∈ N − shun_p − {p}
48      when receive m = ⟨VALUE, k, v⟩_leader,...,s_{f+1}
49        if v ∉ extracted ∪ {⊥_1, ⊥_2} then
50          extracted ∪= {v}
51        recvdSeq[s_{f+1}] ∪= {m}
52      when receive m = ⟨PENANCE, k, t, filler⟩_q
53        penDuring[q] ∪= {leader}
54        if t = |penDuring[q]| then
55          recvdSeq[q] ∪= {m}
56      foreach q ∈ N − shun_p − {p}
57        if recvdSeq[q] ∉ M^k_{q→p} then
58          shun_p ∪= {q}
59    if |extracted| = 1 then
60      deliver v ∈ extracted
61    else
62      deliver SF
63      shun_p ∪= {leader}
```

Figure 2. Just TRB for instance $k > 0$.

essary to balance appropriately the cost of penances against the possible savings from frivolously losing friends.

The next subsection formalizes the predictable communication pattern and friend, ex-friend, and enemy relationships. Afterwards, we prove that Just TRB is safe and live if rational players obey the protocol.

5.2 Definitions & Lemmas

Let $\vec{\rho}$ denote the strategy profile in which each player obeys the Just TRB protocol. We formally define the *message sequence* sent from player r to player p through instance k when $\vec{\sigma}$ is played as $\text{seq}^k_{r\to p}(\vec{\sigma}) = \bigcup_{h\in[1,k]} \text{sent}^h_{r\to p}(\vec{\sigma})$. A message sequence is *acceptable* if it could have been sent by a player r following ρ_r. Formally,

Definition 1 (Acceptable Message Sequence)**.** *A message sequence from r to p through instance k is* acceptable *if and only if that sequence is in the set:*

$$\mathcal{M}^k_{r\to p} = \bigcup_{\substack{\forall \mathcal{C} \subseteq \mathcal{N} - \{r,p\},\\ \forall \vec{\sigma}_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}}} \text{seq}^k_{r\to p}(\vec{\rho}_{\mathcal{N}-\mathcal{C}}, \vec{\sigma}_{\mathcal{C}})$$

For simplicity, $\mathcal{M}_{r\to p} \equiv \mathcal{M}^\infty_{r\to p}$.

In Just TRB, a message sequence is acceptable for an instance k if it contains two value messages and, when necessary, a penance message—that is, if either no value is forwarded in round 2 of k or a penance message was sent during a previous instance led by k's leader.

Definition 2 (Friends)**.** *The* friends *of a player r at instance k when $\vec{\sigma}$ is played are* $\text{F}^k_r(\vec{\sigma}) = \{p \in \mathcal{N} - \{r\} : \text{seq}^k_{r\to p}(\vec{\sigma}) \in \mathcal{M}^k_{r\to p} \wedge \text{seq}^k_{p\to r}(\vec{\sigma}) \in \mathcal{M}^k_{p\to r}\}$.

Definition 3 (Ex-friends)**.** *The* ex-friends *of a player r at instance k when $\vec{\sigma}$ is played are* $\text{X}^k_r(\vec{\sigma}) = \mathcal{N} - \text{F}^k_r(\vec{\sigma}) - \{r\}$.

Definition 4 (Enemies)**.** *The enemies of player r at instance k when $\vec{\sigma}$ is played are* $\text{E}^k_r(\vec{\sigma}) = \text{E}^{k-1}_r(\vec{\sigma}) \cup \{p = \text{leader}^k : \langle \textit{VALUE}, k, v\rangle_p \notin \text{sent}^k_{p\to r}(\vec{\sigma})\}$ *where trivially,* $\text{E}^0_r(\vec{\sigma}) = \emptyset$.

For notational simplicity, $\text{F}_r(\vec{\sigma}) \equiv \text{F}^\infty_r(\vec{\sigma})$, $\text{X}_r(\vec{\sigma}) \equiv \text{X}^\infty_r(\vec{\sigma})$, and $\text{E}_r(\vec{\sigma}) \equiv \text{E}^\infty_r(\vec{\sigma})$. Let $\text{X}^{\text{BYZ}}_r(\vec{\sigma})$ be the set of Byzantine ex-friends and $\text{X}^{\text{NON}}_r(\vec{\sigma})$ be the remaining ex-friends. Of course, enemies are also considered ex-friends. The following two lemmas characterize friend and enemy relationships within Just TRB; they will be useful in later proofs.

Lemma 9. *If players p and q follow Just TRB, then p and q are friends.*

Proof. By definition, for all $k > 0$, $\text{seq}^k_{p\to q}(\vec{\sigma}) \in \mathcal{M}^k_{p\to q}$, implying $p \in \text{F}_q(\vec{\sigma})$. □

Lemma 10. *Suppose player p follows the protocol by playing ρ_p. If p considers q to be an ex-friend, then p eventually is $q's$ enemy.*

Proof. Since p follows ρ_p, p does not send messages to players in *shun*$_p$. Since $q \in \text{X}^{\text{NON}}_p(\vec{\sigma})$, we infer that $q \in$ *shun*$_p$, and thus p does not send a *VAL*$_1$ message to q when p is leader. Hence, $p \in \text{E}_q(\vec{\sigma})$. □

5.3 Safety

When non-Byzantine players follow the protocol, Just TRB solves the R-TRB problem in a nearly identical way as D-S TRB. Lemma 9 states that players following the protocol are friends, so any value message sent from one non-Byzantine player to another in D-S TRB would also be sent following Just TRB. While Just TRB uses additional messages—penances and fillers—these messages do not change the value of the $extracted$ sets used to actually deliver a value. Theorem 4 formalizes the above intuition.

Theorem 4. *If all non-Byzantine players follow the protocol by playing ρ then* **TRB1-4** *hold.*

Proof. Since D-S TRB maintains **TRB1–4** when all non-Byzantine players play D-S TRB, it is sufficient to show that if all non-Byzantine players play Just TRB they deliver the same value as they would have delivered if playing D-S TRB. It follows from Lemma 9 that non-Byzantine players are friends and thus do not shun each other.

A non-Byzantine leader sends exactly one value v to all non-Byzantine players in round 1 of both protocols. Every non-Byzantine player thus receives only v in both protocols and delivers v in round $f + 1$.

With a Byzantine leader, if any non-Byzantine player delivers v broadcast by the leader, then some non-Byzantine player received any value broadcast by the sender by round f at the latest (since there are at most f Byzantine players). Both protocols specify that any non-Byzantine player forward the first two values received in an instance to other non-Byzantine players. So at the end of round $f+1$ all non-Byzantine players will have received either the same unique value v, or at least two values, or no value. Non-Byzantine players all deliver **SF** in the latter two cases or the unique value v in the first case. □

6 Rationality Analysis

We now prove that the Just TRB protocol is an ex ante Nash equilibrium. In this section, the level of technical detail increases sharply. Though at times painful, these details provide the necessary closure to revealing the complete process of BAR-tolerant protocol design.

The key insight is that each player p's utility in the protocol's steady state dominates p's overall utility. Therefore, the rational strategy is to maximize the steady state utility.

message	content	c_{snd}
VAL_i	$\langle \text{VALUE}, k, v \rangle_{leader^k, s_2, \ldots, s_{i-1}, r}$	γ
$VAL_\perp$	$\langle \text{VALUE}, k, \perp_{\{1,2\}} \rangle_r$	γ
PNC_t	$\langle \text{PENANCE}, k, t, \text{filler} \rangle_r$	κ_t

Table 1. Costs and contents of specific messages sent by player r in instance k.

Definition 5 (Steady State)**.** *A game execution with strategy profile $\vec{\sigma}$ is in the* steady state *at instance k if and only if every player makes no more enemies and loses no more friends in future instances.*

Every R-TRB game eventually reaches the steady state since the set of each player's enemies is non-decreasing and the set of each player's friends is non-increasing. Because the steady state condition holds for an infinite suffix of TRB instances, the average utility in the steady state dominates the average expected utility. We define a player's average utility in the steady state as its utility across n consecutive steady state instances, thereby accounting for the increased utility that a player receives when it is leader—exactly once every n instances.

We start the proof by establishing a lower bound on the utility a player p expects when p obeys Just TRB. We then consider strategy profiles in which p deviates. Since there are a large number of possible deviations, we group deviations into equivalence classes; two deviations are equivalent if they expect to produce the same number of friends and enemies in the steady state. Next, we establish upper bounds on the utility p expects for strategies in each class. Finally, we show that the lower bound for obeying Just TRB is at least the upper bound for every equivalence class, thereby proving that p expects no benefit from unilaterally deviating and that Just TRB is an ex ante Nash equilibrium.

Similar to our discussion in Section 4.2, we consider only $n > f + 1$ and $f > 0$. We address the corner cases in a technical report [4].

6.1 Proof Preliminaries

We now define the costs of sending value and penance messages in Just TRB. Using these costs, we determine the cost required to maintain a set of friends in the steady state, which we call the *cost of friendship*.

Message Costs. Just TRB uses three kinds of messages: meaningful value (VAL_i sent in round i), dummy value ($VAL_\perp$), and penance messages (PNC_t where the sender has t enemies). Table 1 shows the content and costs of these messages.

Both meaningful and dummy value messages have identical costs (γ) so as to eliminate incentives for sending one kind of value message over another. Penance messsages are more complicated; their cost varies to balance the possible savings from frivolously losing friends. Players verify the size of fillers in each penance to ensure that each penance has the appropriate cost.

A penance message PNC_t costs κ_t, where

$$\kappa_t = \begin{cases} \frac{(n-t)(t-1)\kappa_{t-1}+2n\gamma}{t(n-t-1)}, & t \in [1, n-2] \\ 0, & \text{otherwise} \end{cases}$$

By making an additional enemy, a player r saves the cost of sending $2n$ value messages and trades the $(n-t)(t-1)$ PNC_{t-1} messages for $t(n-t-1)$ PNC_t messages.

Cost of Friendship. There are exactly 3 message patterns that a player p following ρ_p could send to another player r: (a) no messages, (b) two *VAL* messages, including at least one during round 2, and (c) no *VAL* messages during round 2, two *VAL* messages total, and one *PNC* message.

The *cost of friendship*, $C(x, y)$, to a player r with x friends and y enemies in following ρ_r is

$$\mathrm{C}(x, y) = x(y\kappa_y + 2n\gamma)$$

In n instances of TRB, Just TRB specifies that a player r sends the following messages only to its x friends: (a) two value messages (costing $2n\gamma$) and (b) a penance message each time one of the y enemies is leader (costing $y\kappa_y$).

From the definitions of $\mathrm{C}(x, y)$ and κ_y, we derive the following properties of the cost of friendship: (a) it costs more to keep the same set of friends while making more enemies; (b) it costs more to have a player as an enemy than it does to keep him as a friend; and (c) costs are trivially minimized by having no friends:

Lemma 11. *Let $x \in [0, n-1], y \in [0, n-x-2]$.*
(a) $x > 0 \Rightarrow \mathrm{C}(x, y) \leq \mathrm{C}(x, y+1)$.
(b) $x > 0 \Rightarrow \mathrm{C}(x, y) \leq \mathrm{C}(x-1, y+1)$.
(c) $x = 0 \Rightarrow \mathrm{C}(x, y) = 0$.

One interesting implication of the cost of friendship is that Byzantine players increase the costs paid by rational players when they are enemies rather than friends.

6.2 Utility of Playing ρ

To prove that $\vec{\rho}$ is a Nash equilibrium for risk-averse players, we first place a lower bound on the utility that a player expects from playing the recommended strategy $\vec{\rho}$; in the next section, we show that the lower bound of $\vec{\rho}$ is no less than an upper bound on the utility a player expects from unilaterally deviating from $\vec{\rho}$.

The utility $\bar{u}_r(\vec{\rho})$ identifies a rational player r's worst-case utility when every non-Byzantine player follows $\vec{\rho}_{\mathcal{N}-\mathcal{B}}$ and the Byzantine players follow arbitrary strategies $\vec{\tau}_{\mathcal{B}} \in \mathcal{S}_{\mathcal{B}}$. To calculate the worst-case utility of following $\vec{\rho}$, we establish a lower bound on r's benefit and an upper bound on r's cost as a function of the friends and enemies of r in the steady state.

Benefits. By proving Just TRB is a Byzantine fault-tolerant TRB protocol in Theorem 4, we can easily assert in the following Lemma that r receives full benefit when $(\vec{\rho}_{\mathcal{N}-\mathcal{B}}, \vec{\tau}_{\mathcal{B}})$ is played.

Lemma 12. *If all non-Byzantine players follow the protocol by playing ρ and at most f Byzantine players deviate in an arbitrary fashion, then non-Byzantine player r receives benefit $\beta + n\varpi$ during n consecutive steady state instances.*

Costs. The next Lemma bounds the maximum cost with respect to the cost of friendship in the steady state when rational players follow $\vec{\rho}$. It is important to note that Byzantine players maximize costs by being enemies, rather than friends, of rational players.

Lemma 13. *Let $n > f+1$ and $f > 0$. If all non-Byzantine players follow the protocol by playing ρ and the Byzantine players deviate arbitrarily, then the worst case expected cost for non-Byzantine player r is at most $C(n-f-1, f)$.*

Proof. It follows from Lemma 9 that $|\mathrm{F}_r(\vec{\varphi})| \geq n-f-1$ and $|\mathrm{E}_r(\vec{\varphi})| \leq f$. It follows from Lemma 11 that $\mathrm{C}(x,y)$ is maximized when $y = f$. Since $\mathrm{C}(x,y)$ is defined for ρ, $costs_r(\vec{\varphi}) \leq \mathrm{C}(n-f-1, f)$. □

Utility. Using the bounds on steady state benefit and cost we provide a lower bound on utility.

Lemma 14. *Let $n > f+1$ and $f > 0$. If all non-Byzantine players follow the protocol by playing ρ and there are at most f Byzantine players, then the expected utility for risk averse player r is at least $\frac{(\beta+n\varpi)-\mathrm{C}(n-f-1,f)}{n}$*

Proof. The utility under the risk-averse rational model depends upon the worst-case average expected utility. Let $\vec{\varphi} = (\vec{\rho}_{\mathcal{N}-\mathcal{B}}, \vec{\tau}_{\mathcal{B}})$, $\forall \mathcal{B} \subseteq \mathcal{N}$, $|\mathcal{B}| \leq f$. The average expected utility for any $r \in \mathcal{N} - \mathcal{B}$ is determined by the costs and benefits of the steady state, leading to $\hat{u}_r(\vec{\varphi}) = \frac{benefits_r(\vec{\varphi}) - costs_r(\vec{\varphi})}{n}$. Substituting according to Lemmas 12 and 13, we obtain $\bar{u}_r(\vec{\varphi}) \geq \frac{(\beta+n\varpi)-\mathrm{C}(n-f-1,f)}{n}$ for any Byzantine behavior. □

6.3 Utility of Deviating

We now show that there exists a *spiteful strategy* for Byzantine players to follow that places an upper bound on a rational player r's average expected utility, irrespective of r's unilateral deviation. This upper bound matches the lower bound for $\bar{u}_r(\vec{\rho})$ and demonstrates that Just TRB is a Nash equilibrium for risk-averse players.

We define the spiteful strategy $\vec{\varsigma}^r_{\mathcal{B}}$ such that Byzantine players follow $\vec{\rho}_{\mathcal{B}}$, but collude against r by inserting r into $shun_q$ for all $q \in \mathcal{B}$. Since spiteful players shun r, they are by definition enemies of r:

Lemma 15. *If all Byzantine players follow the spiteful strategy against non-Byzantine player r then all Byzantine players are in the enemy set of r.*

Benefits. We derive an upper bound on the benefit of any unilateral deviation by r using the benefits for Γ_{TRB}:

Lemma 16. *For any strategy followed by non-Byzantine player r, if there are at most f Byzantine players then the benefits received by r are at most $\beta + n\varpi$.*

A tighter bound can be obtained for the special case of deviations that result in r having no friends.

Lemma 17. *Let $n \geq f+1$ and $f > 0$. If non-Byzantine player r has no friends and all Byzantine players play the spiteful strategy against r, then r obtains benefit at most ϖ.*

Proof. Without friends, r cannot learn the values proposed by other players and must deliver **SF**, which clearly can violate **TRB3**, resulting in no benefit to r. However, when r is leader, r can trivially guarantee **TRB3** and obtain ϖ by delivering **SF**. All other non-Byzantine players will also deliver **SF** because they receive no value messages from r because r shuns all players since $|\mathrm{F}_r(\vec{o})| = 0$. □

Costs. We next derive a lower bound on r's cost when f Byzantine players follow the spiteful strategy and r pursues any unilateral deviation. For deviations that maintain a non-zero number of friends, the following Lemma bounds the minimum cost of deviation:

Lemma 18. *Let $n \geq f+1$ and $f > 0$. If there are at most f Byzantine players and r has $e < n-1$ enemies in the steady state, then r expects costs of at least $C(n-f-1, f)$.*

Proof. Lemmas 15 and 10 imply that $\mathrm{X}_r(\vec{o}) = \mathrm{E}_r(\vec{o})$ so that a player is either a friend or enemy. If every player is counted by x or y, Lemma 11 rule (b) then states that $\mathrm{C}(x,y)$ is minimized for $\min(x)$. Given the lower bound of $|\mathrm{E}_r(\vec{o})|$ determined by $|\mathcal{B}| = f$, minimal costs are attained for $\mathrm{C}(n-f-1, f)$. □

For deviations by player r described by $\vec{o}$ that maintain no friends ($\mathrm{F}_r(\vec{o}) = \emptyset$), we note that r is not required to send any messages so that trivially, $costs_r(\vec{o}) \geq 0$.

Utility. Using the bounds on benefit and cost in the steady state, we prove an upper bound on r's utility.

Lemma 19. *Let $n > f+1$ and $f > 0$. If there are at most f Byzantine players, then the expected utility for risk averse player r is at most $\max\{\frac{(\beta+n\varpi)-\mathrm{C}(n-f-1,f)}{n}, \frac{\varpi}{n}\}$.*

Proof. To find the utility under the risk-averse rational model, we find the worst-case average expected utility. For all $\mathcal{B} \subseteq \mathcal{N} - \{r\}$, $|\mathcal{B}| = f$, let $\vec{o} = (\vec{\rho}_{\mathcal{N}-\mathcal{B}-\{r\}}, \sigma_r, \vec{\varsigma}^r_{\mathcal{B}})$.

The average expected utility for any $r \in \mathcal{N} - \mathcal{B}$ is determined by the costs and benefits of the steady state, leading to $\bar{u}_r(\vec{o}) = \frac{\textit{benefits}_r(\vec{o}) - \textit{costs}_r(\vec{o})}{n}$.

Consider first the case where $|\mathrm{E}_r(\vec{o})| < n-1$. It follows from Lemma 18 that $\textit{costs}_r(\vec{o}) \geq \mathrm{C}(n-f-1, f)$. Finally, $\bar{u}_r(\vec{o}) \leq \frac{(\beta + n\varpi) - \mathrm{C}(n-f-1,f)}{n}$ using the upper bound on benefits provided by Lemma 16.

Assume $|\mathrm{E}_r(\vec{o})| = n-1$. It follows from Lemma 17 that $\textit{benefits}_r(\vec{o}) \leq \varpi$ and as argued above, that $\textit{costs}_r(\vec{o}) \geq 0$. Hence, $\bar{u}_r(\vec{o}) \leq \frac{\varpi}{n}$. □

6.4 Just TRB Is a Nash Equilibrium

We prove that $\vec{\rho}$ is a Nash Equilibrium using the bounds on utility proved in the previous sections. In the presence of Byzantine behavior, we show that the minimum expected utility of executing Just TRB is the maximum expected utility of any unilateral deviation strategy profile.

Theorem 5. *Let $n > f+1$ and $f > 0$. The Just TRB protocol is a Nash equilibrium for risk averse players if $\beta + (n-1)\varpi \geq \mathrm{C}(n-f-1, f)$.*

Proof. It suffices to show $\forall r \in \mathcal{N}, \forall \sigma_r \in \mathcal{S}_r$, $\bar{u}_r(\vec{\rho}) \geq \bar{u}_r(\vec{\rho}_{\mathcal{N}-\{r\}}, \sigma_r)$. It follows from Lemma 14 that $\bar{u}_r(\vec{\rho}) \geq \frac{\beta + n\varpi - \mathrm{C}(n-f-1,f)}{n}$ and from Lemma 19 that $\bar{u}_r(\vec{\rho}_{\mathcal{N}-\{r\}}, \sigma_r) \leq \max\{\frac{\beta + n\varpi - \mathrm{C}(n-f-1,f)}{n}, \frac{\varpi}{n}\}$. By our assumption that $\beta + (n-1)\varpi \geq \mathrm{C}(n-f-1, f)$, $\bar{u}_r(\vec{\rho}) \geq \bar{u}_r(\vec{\rho}_{\mathcal{N}-\{r\}}, \sigma_r)$, completing the proof. □

Theorem 5 is useful to discuss the assumption of game theory that the game should be worth playing when everyone cooperates. A sufficient condition for playing the Just TRB game is that a player expects the benefits of running the protocol (successful agreements and proposals) to exceed the cost of doing so (messages). To meet this condition for risk-averse players, Theorem 5 provides a precise requirement on the costs and benefits: $\beta + (n-1)\varpi \geq \mathrm{C}(n-f-1, f)$.

7 Conclusion

We have taken the classic synchronous R-TRB problem and examined it in the presence of rational players. We have seen that BFT protocols, even when optimal, are not necessarily resilient to rational behavior. We introduce a novel protocol that solves the synchronous R-TRB problem in the presence of both Byzantine and rational players and presented a detailed analysis of that protocol. The Just TRB protocol and analysis is limited by our assumptions on rational players. Changes in the sources of cost and/or benefits or the expectations rational players place on Byzantine players will fundamentally change the analysis and may leave our protocol short of tolerating both Byzantine and rational participants.

8 Acknowledgements

This work was supported by NSF grant CNS 0509338. We would like to thank the reviewers for their insisghtful comments.

References

[1] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *Proc. 25th PODC*, July 2006.

[2] E. Adar and B. A. Huberman. Free riding on Gnutella. *First Monday*, 5(10):2–13, Oct. 2000.

[3] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth. BAR fault tolerance for cooperative services. In *Proc. 20th SOSP*, Oct. 2005.

[4] A. Clement, J. Napper, H. Li, J. Martin, L. Alvisi, and M. Dahlin. Theory of bar games. Technical report, University of Texas Department of Computer Sciences Technical Report R-06-63, December 2005.

[5] B. Cohen. Incentives build robustness in BitTorrent. In *First Workshop on the Economics of Peer-to-Peer Systems*, June 2003.

[6] D. Dolev and H. R. Strong. Authenticated algorithms for Byzantine agreement. *Siam Journal Computing*, 12(4):656–666, Nov. 1983.

[7] K. Eliaz. Fault tolerant implementation. *Review of Economic Studies*, 69:589–610, Aug 2002.

[8] J. Feigenbaum, R. Sami, and S. Shenker. Mechanism design for policy routing. In *Proc. 23rd PODC*, pages 11–20. ACM Press, 2004.

[9] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[10] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 1978.

[11] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 1982.

[12] H. C. Li, A. Clement, E. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. Bar Gossip. In *Proc. 7th OSDI*, 2006.

[13] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining cooperation in multi-hop wireless networks. In *NSDI*, May 2005.

[14] J.-P. Martin. *Byzantine Fault-Tolerance and Beyond*. PhD thesis, The University of Texas at Austin, Dec. 2006. TR-06-66.

[15] T. Moscibroda, S. Schmid, and R. Wattenhofer. When selfish meets evil: Byzantine players in a virus inoculation game. In *Proc. 25th PODC*, 2006.

[16] J. Nash. Non-cooperative games. *The Annals of Mathematics*, 54:286–295, Sept 1951.

[17] J. Shneidman and D. C. Parkes. Specification faithfulness in networks with rational nodes. In *Proc. 23rd PODC*, pages 88–97. ACM Press, 2004.