**TBFT建模进度**

**摘要**

本文档介绍了Tendermint-BFT（TBFT）共识机制的初步框架、核心功能以及协议描述。本文首先展示了协议框架，修改了某些功能描述，加入了对于终止性、有效性、不可篡改性和数据完整性的保障条件和措施，完善了协议描述，提出了一些目前问题。

1. **初步框架**

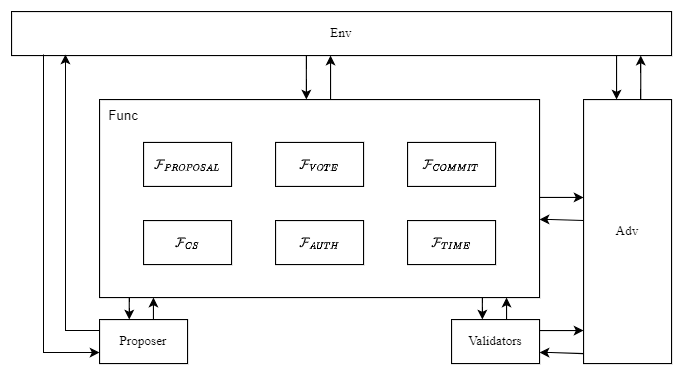


图 1 TBFT协议初步框架

**二、功能描述**

（一）功能

初始化：变量和。

–当从接收到输入时，执行以下操作：

设置和。存储并将发送给敌手。向发送命令。

–当从收到消息，设置，执行以下操作：

如果，则对 进行哈希处理，并将哈希值和签名一起广播给，。否则向发送命令。

–当首次接收到来自敌手的时，执行以下操作：

如果是一个以一元表示法编码的自然数且，则设置。否则忽略该消息。

–当消息送达时，执行以下操作：

使用哈希值验证区块内容的完整性和签名，如果数据未被修改，则将交付给。如果数据被修改则进行重发操作。

（二）功能

初始化：设置和。

–当收到消息时，

* 通过Round-robin规则选定提议者，为中诚实者的集合：

-初始化Validator的为其质押资金：

-按Round-robin规则依次选举，更新。

* 更新：

-未被选中的Validator更新为：

-被选中为的Validator更新为：

–（超时处理）：当从接收到消息时，如果有效，增加，并选择新的提议者。

（三）功能

初始化：设置和。向发送命令。如果在任何阶段从收到消息，直接投票给块。

–当从验证者传入消息时，

* 当收到时，向发送，获取：

-若锁定在上一轮，则签名并广播，为上一轮投票区块。

-否则广播当前轮，即签名并广播。

* 若未收到：

-若未收到任何，则签名并广播。

–当从验证者传入消息时，

持续接收网络中的投票，

* 若收到超过 的投票：

签名并广播 ，向发送解锁上一轮区块，然后向发送锁定当前轮区块。

* 若收到超过 的空投票：

向发送释放所有锁定的区块

* 否则，不会锁定任何区块。

（四）功能

初始化:对每个验证者，设置，表示验证者是否已提交区块。向发送命令。如果在任何阶段从收到消息，直接提交块。

–当从验证者传入消息时，

* 持续接收网络中的投票，判断是否可以进入阶段：

-若收到超过 的投票，进入阶段。

-否则，进入下一轮阶段。

* 阶段并行步骤：

-为区块投票 。

-为区块收集全网的投票。

* 阶段结束：

-若已为区块广播投票且收集到超过的投票，则设置，设置为当前时间，向发送()。

-若没有为区块收集到超过的投票，向发送。

-否则过程仍未完成。

* 提前进入阶段：

若在任何阶段收到超过的投票，立即进入阶段。

–收到来自任意方的消息时：

返回集合并指示区块是否已完成。

（五）功能

初始化：设置，和。

–当从任意验证者接收到消息时，

更新并将重置为。

–当从任意验证者接收到消息时，

将重置为，向发送消息

–当从接收到消息时，将加入到中 对应的中。

–当从接收到消息时，将在对应的中 的中删除。

–当从接收到消息时，则设置。

–当从接收到消息时，返回。

（六）功能

初始化：设置，。

–当从任意验证者接收到请求时，将更新为 ← ，向验证者返回一个消息, 然后开始倒计时。

–当从任意验证者接收到请求时，它会将当前的返回给请求方。

–当从某一个时，它会向对应的验证者发送一个消息。

–当从任意验证者接收到请求时，将重置为 ，向验证者返回一个消息。

**三、协议描述**

Tendermint-BFT协议通过轮次机制和投票阶段确保多个验证者之间就区块达成一致，并最终提交区块。该协议支持容忍少量恶意节点，依赖于消息广播、延迟处理和投票收集来实现共识。

–Party Environment:

**StartProposal**：调用，选择并激活一个提议者Proposer，开始共识。

–Party Proposer:

**Initialize**: 向发送命令。若从收到消息，则直接跳转执行**RoundOK**部分。

**Input**: 从功能中接收提案请求，从中选择一个区块，确定该区块有效后将其作为提议区块。

**Output**: 将提议信息发送给对手，然后签名并广播给所有验证者。

**RoundOK**: Proposer将等待共识完成，然后调用更新轮次，重新选择提议者。

–Party Validator:

**Initialize**: 向发送自己的提案。

**Input**: 在收到来自Proposer的后，验证的完整性和有效性。

**Prevote**: 根据收到的，调用。若共识失败则直接跳转执行**RoundOK**部分。

**Precommit**: 根据收到的，调用。若共识失败则直接跳转执行**RoundOK**部分。

**Commit**: 根据收到的，调用。若共识失败则直接跳转执行**RoundOK**部分。

**RoundOK**: Validator将等待中的提议轮次更新，开始新的轮次。

|  |  |
| --- | --- |
| 图 2 Proposal数据结构 | 图 3 Vote数据结构 |

**四、当前问题**

1. 对TBFT协议共识失败情况了解不多

对于TBFT协议，我觉得需要了解一些共识失败的情况以及敌手攻击操作的情况，以便我对于敌手的操作有更多的了解，更好的了解TBFT协议的运行。

2. TBFT协议有效性

对于有效性，应该如何建模并验证只有由合法Proposer提出的数据能够通过共识过程呢？我认为一种可能的方法是确保非法Proposer的Proposal不会被接受，同时保证合法Proposer的Proposal能够顺利通过共识机制。或者是直接规定非法Proposer不能提议，直接从诚实的Validator中选取Proposer。

3. TBFT协议不可篡改性

对于不可篡改性，我对于“如何确保在共识达成之前，所有节点都无法提前得知最终的输出值”理解不够。在TBFT协议中，Proposer会在Propose阶段会全网广播Proposal，而Proposal中包含Block字段。Validator在接收到Proposal后会不会读取其中的信息呢？如果读取其中信息，那么节点应该是可以提前得知最终的输出值。