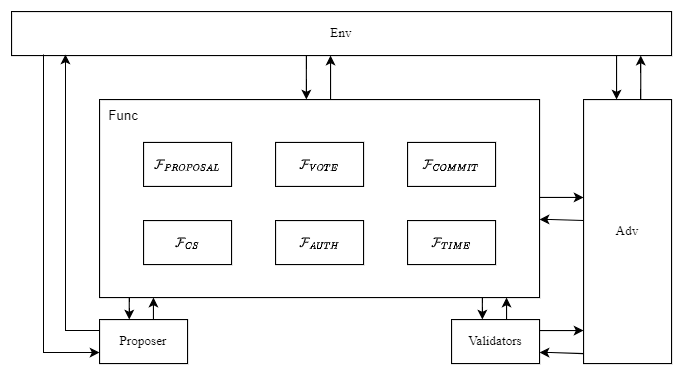
**TBFT建模进度**

**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，然后完善了各功能的描述步骤，加入了对于超时机制的描述，并提出了目前亟待解决的关键问题。

1. **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。将发送给敌手。将广播给，

–当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，则设置。否则忽略该消息。

–当从敌手接收到消息时，执行以下操作：

如果且是有效的延迟值，则设置并将。否则忽略该消息。

（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：

-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：

-若锁定上一轮，则广播。

-否则广播当前轮，即广播。

* 若未收到：

-若未收到任何，则广播。

–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 ，向发送释放之前锁定的区块，然后向发送锁定该区块。

* 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：

-为区块广播投票 。

-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:

**Proposal**：调用，更新轮次，选择一个提议者Proposer并激活。

–Party Proposer:

**Input**: 在每一轮开始时，Proposer首先向发送命令，然后从功能中接收提案请求，并从中选择一个区块作为提议区块。如果从收到消息，直接跳转执行**RoundOK**部分。

**BroadcastProposal**: 当Proposer选择了一个区块并确定该区块有效时，将提议信息发送给对手，并广播给所有验证者。

**RoundOK**: Proposer将调用更新轮次，并重新选择一个新的提议者，等待下一个提议轮次。

–Party Validator:

**Input**: 在收到来自Proposer的提议消息后，Validator验证区块的有效性，并准备参与投票。

**Prevote**: 根据收到的区块，调用。

**Precommit**: 根据收到的区块，调用。

**Commit**: 根据收到的区块，调用。

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

**四、当前问题**

1. 终止性

在 UC 框架下，如何确保在存在网络延迟和恶意节点干扰的情况下，协议能够在有限时间内完成共识？具体包括超时机制设计的有效性、恶意节点拖延行为的应对，以及视图切换过程是否始终能够保证系统最终达成一致。

2. 有效性

如何建模并验证只有由合法提议者提出的数据值能够通过共识过程？当前的挑战在于确保提议值在传播过程中不被篡改，同时如何有效排除恶意提议者的非法数据仍需深入研究。

3. 不可篡改性

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？在建模过程中，需解决投票隐私性如何被保证，协议是否能抵御恶意节点通过协作提前泄露结果，以及如何证明最终输出值不可被修改。

4. 数据完整性

在消息的传输和存储过程中，如何在不增加协议开销的前提下，确保数据不被恶意节点篡改？当前仍需研究如何通过消息认证和抗篡改验证机制，有效抵御拜占庭节点对共识过程中中间值和通信完整性的攻击。