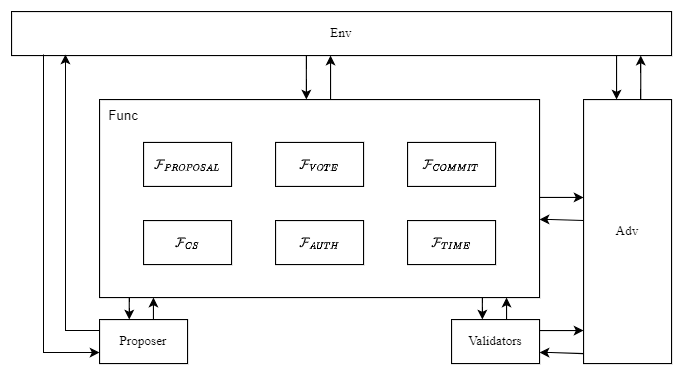
**TBFT建模进度**

**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，并且加入了协议的整体初步描述。

1. **初步框架**



**二、功能描述**

（一）功能F\_{AUTH}

初始化：变量M:=⊥和D:=0。

–当从v\_i∈V接收到输入m时，执行以下操作：

设置D:=1和M:=m。将L(M)发送给敌手A。将M广播给v\_j，(v\_j∈V, j!=i)

–当接收到来自敌手A的(delay, T)时，执行以下操作：

如果T是一个以一元表示法编码的自然数，则设置D:=D+T。否则忽略该消息。

–当从敌手A接收到消息(corrupt, v\_i, m’, T’)时，执行以下操作：

如果D > 0且T’是有效的延迟值，则设置D:=T’并将M:=m’。否则忽略该消息。

（二）功能F\_{PROPOSAL}

初始化：设置Proposal:=⊥和Round:=0。

–当收到消息(startProposal)时，

* 通过Round-robin规则选定提议者Proposer∈V：

-初始化Validator的votingPower为其质押资金：

{votingPower}\_i={stake}\_i, \quad \foralli \in \{1,\dots,N\}

-按Round-robin规则依次选举Proposer，更新Round:=Round+1。

* 更新votingPower：

-未被选中的Validator更新为：

{votingPower}\_i\leftarrow{votingPower}\_i+{stake}\_i

-被选中为Proposer的Validatori更新为：

{votingPower}\_i\leftarrow{votingPower}\_i-\sum\_{j\neqi}{stake}\_j

–（超时处理）：当从A接收到(timeout,T)消息时，如果T有效，增加Round，并选择新的提议者。

（三）功能F\_{VOTE}

初始化：设置Prevotes:=⊥和Precommits:=⊥。

–当从验证者v∈V传入(Prevote, B)消息时，

* 当收到Proposal时，向F\_{CS}发送(v, queryState)，获取PoLC：

-若锁定上一轮Proposal，则广播(v, prevote, B’)。

-否则广播当前轮Proposal，即广播(v, prevote, B)。

* 若未收到Proposal：

-若未收到任何Proposal，则广播(v, prevote, \perp)。

–当从验证者v∈V传入(Precommit, B)消息时，持续接收网络中的prevote投票，

* 若收到超过\frac{2}{3}的prevote投票：

签名并广播 (v, precommit, B)，向F\_{CS}发送(v, unlock, B’)释放之前锁定的区块，然后向F\_{CS}发送(v, lock, B)锁定该区块。

* 若收到超过\frac{2}{3}的空Prevote投票：

向F\_{CS}发送(v, unlock, ALL)释放所有锁定的区块

* 否则不锁定任何区块。

（四）功能F\_{COMMIT}

初始化:对每个验证者v∈V，设置C\_i∈C，C\_i:=0表示验证者是否已提交区块。

–当从验证者v∈V传入(Commit, B)消息时，

* 持续接收网络中的precommit投票，判断是否可以进入Commit阶段：

-若收到超过 \frac{2}{3} 的precommit投票，进入Commit阶段。

-否则，进入下一轮Proposal阶段。

* Commit阶段并行步骤：

-为区块B广播commit投票 (v, commit, B)。

-为区块B收集全网的commit投票。

* Commit阶段结束：

-若v已为区块B广播commit投票且收集到超过\frac{2}{3}的commit投票，则设置C\_i:=1，设置commitTime为当前时间，向F\_{CS}发送(newHeight)。

-若v没有为区块B收集到超过\frac{2}{3}的commit投票，向F\_{CS}发送(newRound)。

-否则Commit过程仍未完成。

* 提前进入Commit阶段：

若在任何阶段收到超过\frac{2}{3}的commit投票，立即进入Commit阶段。

–收到来自任意方v\_k的消息(request\_status)时：

返回集合C并指示区块B是否已完成。

（五）功能F\_{CS}

初始化：设置Height:=0，Round:=0和PoLC:=⊥。

–当从V中的任意验证者v接收到(newHeight)消息时，

更新Height:= Height+1并将Round重置为0。

–当从V中的任意验证者v接收到(newRound)消息时，

将Round重置为0，向F\_{PROPOSAL}发送(startProposal)消息

–当从v接收到(v, lock, B)消息时，将v加入到PoLC中 (Height, Round, B)对应的ValidatorSet中。

–当从v接收到(v, unlock, B)消息时，将v在对应的PoLC中 (Height, Round, B)的ValidatorSet中删除。

–当从v接收到(v, unlock, ALL)消息时，则设置PoLC:=⊥。

–当从v接收到(v, queryState)消息时，返回PoLC。

**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F\_{PROPOSAL}，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从功能F\_{CS}中接收一个新的提案请求，并选择一个区块B作为提议区块。

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

**RoundOK**: 在轮次超时的情况下，Proposer将调用F\_{PROPOSAL}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`(Proposal,B)`后，Validator验证区块B的有效性，并准备参与投票。

**Prevote**: 根据收到的Proposal区块B，调用F\_{VOTE}(Prevote,B)。

**Precommit**: 根据收到的Proposal区块B，调用F\_{VOTE}(Precommit,B)。

**Commit**: 根据收到的Proposal区块B，调用F\_{COMMIT}(Commit,B)。

**RoundOK**: 在验证者确认区块后，Validator将等待F\_{PROPOSAL}中的提议轮次更新，开始新的轮次。

**四、当前问题**

1，目前对于协议中各方交互逻辑进行了初步完善，下一步将对其主要性质进行尝试证明。

2．对于协议整体实现的各功能角色的交互逻辑仍有欠缺，需要继续学习共识协议，阅读论文来完善协议描述。