**TBFT建模进度**

**摘要**

版本7：本文完善了TBFT共识机制的建模框架。1、完善了广播理想功能，确保了信息在指定集合中的可靠传递以及对手的可见性。2、完善功能描述，加入了随机交易剔除功能，使得系统在处理交易时更加灵活和安全。3、细化协议描述，加入了对各子功能的描述，为协议的实现提供了更加明确的指导。4、协议实现的UC建模完成度比例为80%左右。

1. **整体框架**

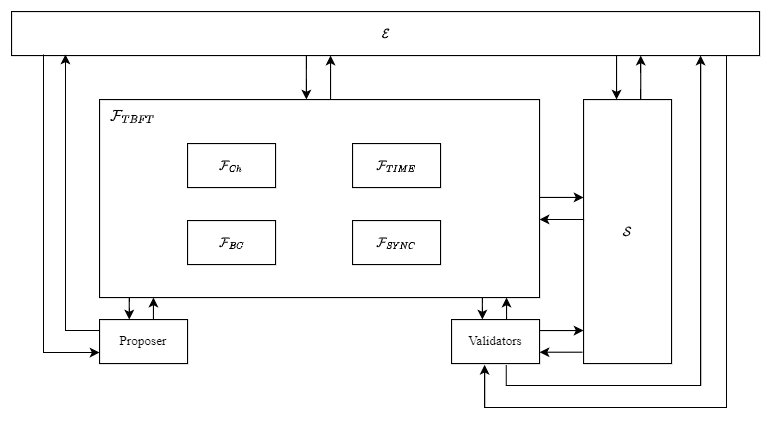


图 1 TBFT协议整体框架

**二、功能描述**

（一）功能

初始化：定义一组参与方，其中 和 分别表示该组中的两个参与方，作为消息 的发送方和接收方。

根据功能参数定义如下。消息标识符 由功能随机选择。

1. 当从 接收到输入 时，向 输出 。
2. 当从 接收到 时，向 发送 。

根据以下参数化函数设置 ：

– 对于 ，设置 , 。当从 接收到 时，向 发送 。

– 对于 ，设置 。

– 对于 ，设置 。

– 对于 ，设置 。

– 对于 ，设置 。当从 接收到 时，向 发送 。

– 对于 ，设置 。

1. 当从 接收到 时，向 发送 。当从 接收到 时，向 发送 。
2. 当从 接收到 时，向 输出 。当从 接收到 时，向 发送 。

a 这赋予了对手 更多的权力，因为 UC 模型中需要顺序发送消息， 决定发送方何时可以继续。

（二）功能

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 H，H为V中诚实者的集合，
  + 初始化Validator的votingPower为其质押资金：
  + 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：
  + 被选中的Validator更新为：

–（超时处理）：当从敌手A接收到(timeout, T)消息时，如果T有效，设置Round = Round + 1，并选择新的提议者。

（三）功能

初始化：向发送(timeStart,)命令。若在任何阶段从收到(timeOver)消息，直接投票给nil块。

–当从验证者传入(Prevote, Proposal)消息时，

* 若收到Proposal，则向发送(, queryState)，获取PoLC。
  + 查询PoLC，若锁定在上一轮Proposal，则签名并广播上一轮区块。
  + 否则，签名并广播当前轮区块。
* 否则，则签名并广播。

–当从验证者传入(Precommit, Proposal)消息时，

* 若收到超过2f+1的prevote投票，
  + 签名并广播 ，向发送(, unlock, B')解锁上一轮区块，然后向发送(, lock, B)锁定当前区块。
* 若收到超过2f+1的空prevote投票，
  + 签名并广播 ，向发送(, unlock, ALL)释放所有锁定的区块。
* 否则，不锁定任何区块。

（四）功能

初始化：对于，设置，。表示Proposal是否已Commit。向发送(timeStart,)命令。若在任何阶段从收到(timeOver)消息，向发送(newRound)。

–当收到从验证者传入(Commit,Proposal)消息时，

* 若收到超过2f+1的precommit投票，
  + 签名并广播，同时收集全网的commit投票。
  + 若已为区块B广播commit投票且收集到超过2f+1的commit投票，则设置，向验证者发送(allowCommit,Proposal)消息，向发送(newHeight)。
  + 否则，向验证者发送(denyCommit,Proposal)消息，向发送(newRound)。
* 否则，向发送(newRound)，开启下一轮。

–收到来自任意方的消息(request\_status)时：

* 返回集合C并指示区块B是否已完成。

（五）功能

初始化：设置Height := 0，Round := 0和PoLC := ⊥。

–当从任意验证者接收到(newHeight)消息时，

更新Height := Height+1并将Round重置为0。

–当从任意验证者接收到(newRound)消息时，

更新Round := Round +1。

–当从出块人Proposer接收到(getProposal, sid, , )消息时，

从配置文件中获取Proposals，然后将其返回给调用者。

–当从接收到 (updateProposal, sid, , Proposals)消息时，

将Proposals更新到配置文件中。

–当从接收到(,lock,B)消息时，

将加入到PoLC中 (Height,Round,B)对应的ValidatorSet中。

–当从接收到(,unlock,B)消息时，

将在对应的PoLC中 (Height,Round,B)的ValidatorSet中删除。

–当从接收到(,unlock,ALL)消息时，设置PoLC := ⊥。

–当从接收到(,queryState)消息时，返回PoLC。

（六）功能

初始化：设置， := ⊥。

–当收到(GetTime)请求时，将当前的返回给请求方。

–当收到(ResetTime)请求时，

将重置为 ，向调用者返回一个(timeOK)消息。

–当收到(timeStart, sid, , )请求时，

将更新为 ，向理想功能返回一个(timeOK)消息, 然后开始倒计时。

–当从时，

会向对应的调用者发送一个(timeOver, sid, , )消息。

（七）功能

初始化：由集合 参数化，其执行过程如下：

–当从参与方 接收到 时，向集合 中的所有实体以及 发送 。

1. **理想功能**

# Functionality

**Parameters**:

**Symbol Explanation:**

* : Total number of nodes in the validator set.

### Upon receiving message from , while :

1. Send to and wait for a response of the form .
2. If :
   * Return to step 1.
3. Otherwise:
   * Send to , and suspend execution.
   * Upon receiving from , resume execution.
   * Send to .
   * Send to and wait for a response of the form .
   * Send to and wait for a response of the form .
   * If is corrupted,
     + Send to .
   * If and no has been received from :
     + Broadcast .
   * Otherwise:
     + Return to step 1.
   * Update .

### Upon receiving message from , while :

1. Send to and wait for a response of the form .
2. If :
   * Broadcast .
3. Otherwise:
   * Send to , and suspend execution.
   * Upon receiving from , resume execution.
   * Send to .
   * If and no has been received from :
     + Broadcast and wait for a response .
     + If :
       - Broadcast and receive .
       - Broadcast .
     + Otherwise:
       - If :  
         Broadcast .
       - Otherwise:  
         Broadcast .
   * Otherwise:
     + Broadcast .
4. Send to .
5. Update .

### Upon receiving message from , while :

1. For each :
   * Set .
2. If there exists a such that :
   * Broadcast t to remove the transaction from the transaction pool.
   * Reset .

### Upon receiving message from , while :

1. Set .
2. If :
   * Broadcast .
3. Otherwise:
   * Broadcast .

### Upon receiving message from , while :

1. Set .
2. Send to and wait for a response of the form .
3. If :
   * Broadcast .
4. Otherwise:
   * Send to , and suspend execution.
   * Upon receiving from , resume execution.
   * Send to .
   * If and no has been received from :
     + Set , .
     + Broadcast .
     + Set , .
   * Otherwise:
     + Broadcast .
5. Update .

### Upon receiving message from , while :

1. Set .
2. Send to and wait for a response of the form .
3. If :
   * Broadcast , and set .
4. Otherwise:
   * Send to , and suspend execution.
   * Upon receiving from , resume execution.
   * Send to .
   * If and no has been received from :
     + Broadcast , and set .
   * Otherwise:
     + Update and .

### Upon receiving message from , while :

1. Set .
2. If :
   * Set , and update , .
3. Send to .
4. Send to , receive its response :
   * If :
     + Update and reset
     + Send to .
   * Otherwise re-execute this step.

### Upon receiving message :

1. Set .
2. If :
   * Send to .

**四、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 Z:

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

–Party Proposer:

**Initialize**: 向发送(timeStart,)命令。若从收到(timeOver)消息，则直接跳转执行RoundOK部分。

**Input**: 从功能中接收并选择一个提案，确定其区块B有效后将其作为提议区块。

**Propose**: 将提议信息L(|Proposal|)发送给敌手A，然后签名并广播(Proposal)给验证者。

**RoundOK**: 调用更新轮次，重新选择提议者，开始新的轮次。

–Party Validator:

**Initialize**: 向发送自己的提案。

**Input**: 在收到来自Proposer的Proposal后，验证Proposal的完整性和有效性。

**Prevote**: 根据收到Proposal的，调用。

**Precommit**: 根据收到的Proposal，调用。若共识失败跳转执行RoundOK部分。

**Commit**: 根据收到的Proposal，调用。若共识失败跳转执行RoundOK部分。

**RoundOK**: 调用更新轮次，重新选择提议者，开始新的轮次。

## **The Protocol**

|  | Proposer |  | Validator |  |
| --- | --- | --- | --- | --- |
|  | 1: Send to |  |  |  |
|  | 2: Send to |  |  |  |
|  | 3 : Get from |  |  |  |
|  | 4: Select a Proposal value from the Proposals. |  |  |  |
|  | 5: Send to |  |  |  |
|  | 6: If and get from , then call |  |  |  |
|  | 7: Otherwise:  broadcast |  |  |  |
|  |  |  | 8: Send to |  |
|  |  |  | 9: If , then call |  |
|  |  |  | 10: Send to |  |
|  |  |  | 11: If , then call |  |
|  |  |  | 12: Send to |  |
|  |  |  | 13: If , then call |  |
|  | Output to |  | 14 : If get from  Call |  |
|  | Output to |  | 15 : Otherwise: Call |  |

## **The Functionality**

| or Validators |  |  |
| --- | --- | --- |
| 1: Send to |  |  |
|  |  | 2: Send to |
|  |  | 3 : Get from |
|  |  | 4 : Each validator initializes their : |
|  |  | 5 : The validator is chosen as the proposer: |
|  |  | 6 : Send to |
|  |  | 7 : For unselected validators : |
|  |  | 8 : For the selected proposer : |
|  |  | 9 : Update based on |
|  |  | 10 : Send to |

## **The Functionality**

|  | Validator |  |  |
| --- | --- | --- | --- |
| **Prevote** | 1: Send to |  |  |
|  |  |  | 2 : Send to |
|  |  |  | 3 : If no is received from :  Sign and broadcast |
|  |  |  | 4 : Otherwise, If :  Send to to get PoLC |
|  |  |  | 5 : If is locked on Proposal from the previous round:  Sign and broadcast |
|  |  |  | 6 : Otherwise:  Sign and broadcast |
|  |  |  | 7 : If no :  Sign and broadcast |
| **Precommit** | 1: Send to |  |  |
|  |  |  | 2 : Send to |
|  |  |  | 3 : If no is received from :  Sign and broadcast |
|  |  |  | 4 : Otherwise, If :  Upon receiving more than prevote votes:  Sign and broadcast |
|  |  |  | 5 : Send to |
|  |  |  | 6 : Send to |
|  |  |  | 7 : Upon receiving more than null prevote votes:  Sign and broadcast |
|  |  |  | 8 : Send to |
|  |  |  | 9 : If no , Do not lock any blocks. |

## **The Functionality**

|  | Validator |  |  |
| --- | --- | --- | --- |
| **Commit** | 1: Send to |  |  |
|  |  |  | 2 : Send to |
|  |  |  | 3 : If no is received from :  Send to |
|  |  |  | 4 : Otherwise, If : If more than precommit votes are received:  Sign and broadcast |
|  |  |  | 5 : Collect commit votes from the network. |
|  |  |  | 6 : If node has already broadcast a commit vote for and collected more than commit votes:  Send to Validator |
|  |  |  | 7 : Send to |
|  |  |  | 8 : Otherwise: Send to Validator |
|  |  |  | 9 : Send to |

## **The Functionality**

|  | Proposer |  |  |
| --- | --- | --- | --- |
| **Get Proposals** | 1: Send to |  |  |
|  |  |  | 2: Get from Configuration File |
|  |  |  | 3: Send to Proposer |
|  |  |  |  |
| **Set Proposals** | 1 : Send to |  |  |
|  |  |  | 2: Update to Configuration File |
|  |  |  |  |
| **GetPoLC** | 1 : Send to |  |  |
|  |  |  | 2 : Return PoLC to the caller |
|  |  |  |  |
| **UnLock** | 1 : Send to |  |  |
|  |  |  | 2 : Delete from the ValidatorSet corresponding to in PoLC |
|  | 1 : Send to |  |  |
|  |  |  | 2 : Set PoLC := ⊥ |
| **Lock** | 1 : Send to |  |  |
|  |  |  | 2 : Add to the ValidatorSet corresponding to in PoLC. |
|  |  |  |  |
| **NewHeight** | 1 : Send to |  |  |
|  |  |  | 2 : Set , |
|  |  |  |  |
| **NewRound** | 1 : Send to |  |  |
|  |  |  | 2 : Set |

## **The Functionality**

|  | Proposer or Validator |  |  |
| --- | --- | --- | --- |
| **Countdown** | 1: Send to |  |  |
|  |  |  | 2: Set , start the countdown. Return |
|  |  |  | 3 : When and :  to the caller |
| **ResetTime** | 1: Send to |  |  |
|  |  |  | 2 : Set , stop the countdown Return |
| **GetTime** | 1: Send to |  |  |
|  |  |  | 2 : Return to the caller |