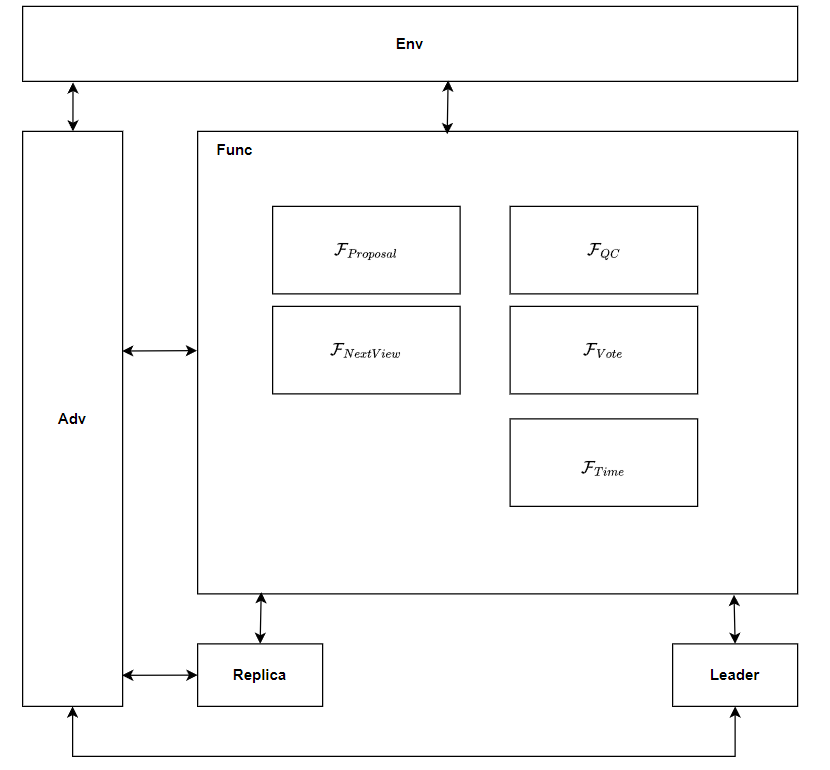
**Hotstuff建模进度**

**摘要：**

针对上周发现的超时处理不够细化的问题，引入了新的理想功能F\_time,对之前不够细化的理想功能重新进行了描述，提出了一个简单的F\_ideal草稿。

1. **整体框架图：**



1. **功能描述**

### F\_{Proposal}

初始化：设置。

–当收到消息时，

* 选取这些消息中最高的prepareQC最为highQC：
* 在highQC的节点的叶子上写入客户指令，提出新的提案B：

-将提案B、highQC封装在MSG中广播给replica：

### F\_{Vote} 初始化：设置。

–当收到来自的消息时，

* 先检查m是否与自己状态匹配：
* 检查叶子节点是否是本地lockedQC对应节点后继以及QC是否比本地lockedQC对应节点的视图更高：
* 在highQC的节点的叶子上写入客户指令，提出新的提案B：

-将投票信息结点m、自己的部分签名封装在VOTEMSG中发送给leader。

-当收到来自的消息时，

* 先检查QC是否与自己状态匹配：
* 如果决定投票且type是PREPARE阶段，更新本地状态：
* 如果决定投票且type是PRE-COMMIT阶段，更新本地状态：

-将投票信息m.justify.node、自己的部分签名封装在VOTEMSG中发送给leader：

### F\_{QC}

初始化：设置。

-当收到2f+1条投票消息时：

* 先检查m是否与自己状态匹配：
* 收集replica的投票，把部分签名组合：

-将QC封装在MSG中广播给replica：

### F\_{TIME}

初始化：设置，。

–当从任意replica 接收到请求时，将更新为 ← ，向replica 返回一个消息, 然后开始倒计时。

–当从某一个时，它会向对应的replica 发送一个消息。

1. **F\_{Next\_view}**

初始化：设置，。

–当从任意replica m收到 next\_view请求MSG(⊥，m,prepareQC)时，将更新为，将prepareQC更新为。

–将、封装在NEW-VIEW message中发送给下一视图的leader。

1. **协议描述**

–Party Environment:

调用更新轮次，选取某个副本作为本轮的leader。

–Party Leader:

**New\_view**：新领导者从功能收集来自(n - f)个副本的"new-view"消息。这些消息包含每个副本在上一轮的（prepareQC）。

**Proposal**: 领导者调用从这些New\_view消息中选择具有最高视图编号prepareQC，（如果没有的话，为⊥)并基于此创建一个新的提案（Proposal）

**Broadcast MSG**: 领导者向所有副本广播这个提案，并附带其选择的最高prepareQC作为安全证明。

**QC**:领导者调用，对来自replica的部分签名进行组合生成QC。并且将其广播给replica。

–Party Replica:

**safeNode**: 在收到来自Leader的提议消息m后，它首先调用功能检查提案消息m，m携带QC的正确性参数（justification）m.justify，检查后确定m.node是否可以安全接受。

**Prepare**: 根据收到的Proposal消息m，调用，将投票发送给leader。

**Pre-commit**: 根据收到的PrepareQC，更新自身PrepareQC，调用。将投票发送给leader。

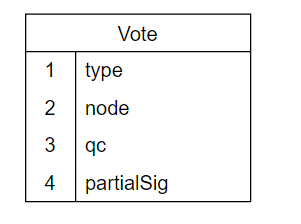
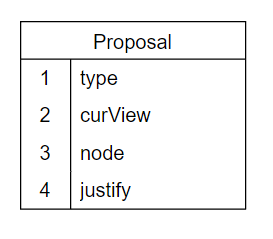
**Commit**: 根据收到的Pre-commitQC，更新lockedQC，调用。将投票发送给leader。

**Next\_view**:

在所有阶段中，副本在视图viewNumber处等待消息的超时时间，超时时间由辅助的确定。如果nextView（viewNumber）中断等待，副本调用,增加viewNumber并开始下一个视图。

**RoundOK**: 副本将等待轮次更新，开始新的轮次。

相关数据结构：



1. **理想功能**

–Hotstuff的理想功能::

### 初始化

初始化以下变量：

* 视图编号
* 初次超时时间T0:=300 ms
* 各副本当前prepareQC
* 各副本当前lockedQC

### Prepare 阶段

* 当节点收到提议块后，进入 Prepare 阶段，并广播 voteMsg(prepare, n.node, ⊥) 消息，表示自己对提议块的初步投票。
* 节点等待其他节点的投票消息。如果收到 n-f 个有效的 voteMsg(prepare, m.node, ⊥) 消息，且这些消息来自不同节点，且没有冲突，节点进入 Pre-commit 阶段。

**条件检查：**

* 如果收到 n-f 个有效的 voteMsg(prepare, m.node, ⊥) 消息，且这些消息有效且一致，节点认为提议块有效，进入 Pre-commit 阶段。

### Pre-commit 阶段

* 当节点从其他节点收到 n-f 个有效的 voteMsg(prepare, m.node, ⊥) 消息后，进入 Pre-commit 阶段，并广播 preCommitMsg(prepare, n.node, curView) 消息，表示自己对提议块的预提交。
* 节点继续等待 n-f 个有效的 new-view 消息，其中每个消息必须匹配当前视图的提议块，且视图号为 curView - 1。
* 如果节点收到 n-f 个有效的 matchingMsg(m, new-view, curView − 1) 消息，且这些消息匹配当前视图的提议块，节点进入 Commit 阶段。

**条件检查：**

* 如果节点收到 n-f 个有效的 matchingMsg(m, new-view, curView − 1) 消息，且这些消息有效，节点认为当前视图的提议块有效，进入 Commit 阶段。

**Commit 阶段**

* 当节点从其他节点收到 n-f 个有效的 preCommitMsg(prepare, m.node, curView) 消息后，进入 Commit 阶段，并广播 commitMsg(prepare, n.node, curView) 消息，表示自己确认该提议块。
* 节点继续等待其他节点的 commitMsg 消息，直到收到 n-f 个有效的 commitMsg 消息。如果这些消息来自不同节点，且确认的提议块一致，则节点将区块提升到更高的高度，进入下一轮的共识过程。