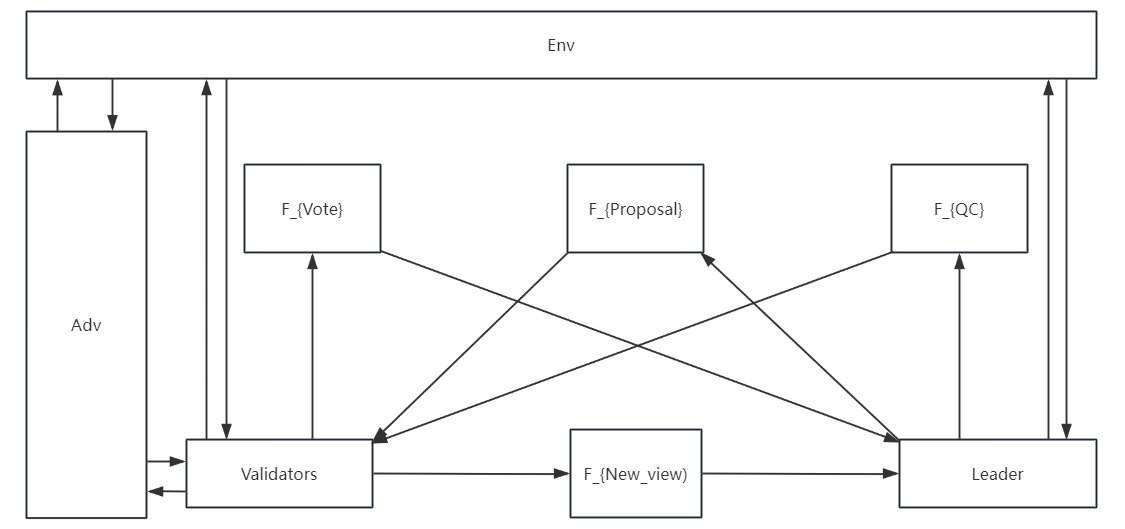
**Hotstuff建模进度**

**摘要：**

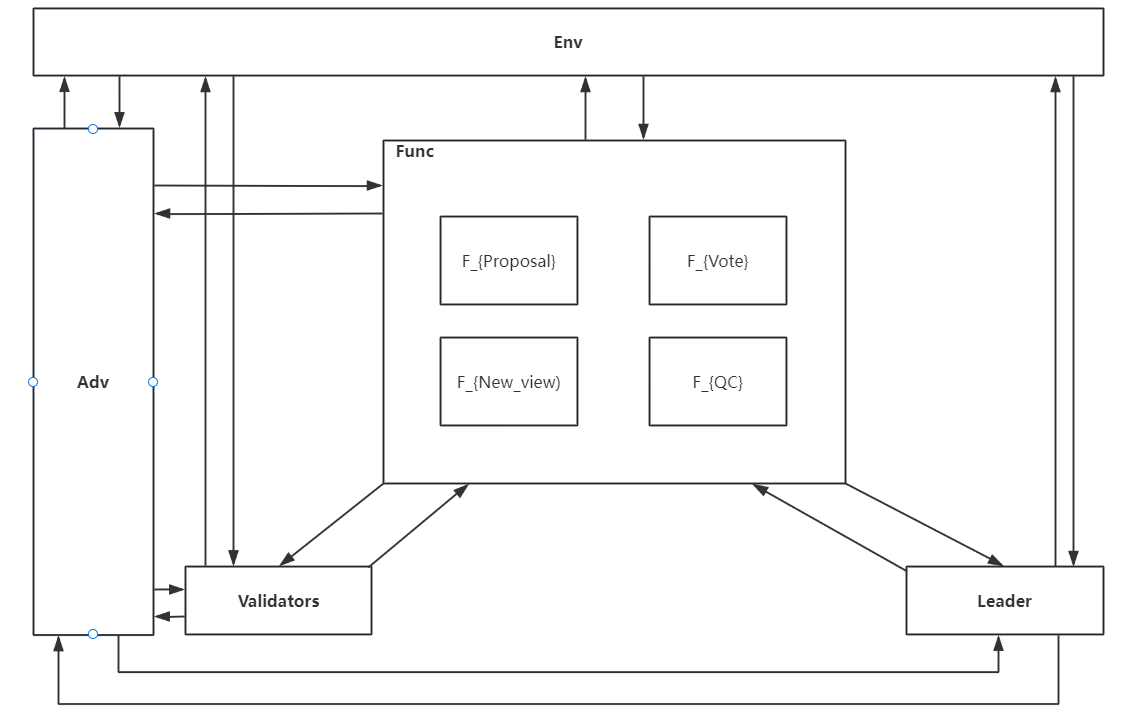
这篇文档是关于Hotstuff共识协议的11.3建模进度报告。首先是对整体框架图进行了进一步修改，将理想功能各小模块整合在理想功能中，各参与实体与理想功能进行交互。接着详细介绍了理想功能各个模块。

1. **整体框架图：**

**修改前：**



**修改后：**



1. **功能描述**

### 1. F\_Proposal

F\_Proposal功能负责生成并广播提案，供协议参与者进行投票。它确保只有合法的提案者（通常是当前领导者）可以创建并发送新的区块提案。具体要求如下：

* **授权领导者生成提案**：只有在获得授权的领导者才可以调用 F\_Proposal来生成新的区块提案。
* **验证区块的合法性**：F\_Proposal会检查区块是否符合协议的格式和内容要求，确保区块无误且未被篡改。
* **广播提案**：一旦提案生成， F\_Proposal会将其广播给所有参与者，使他们可以验证并准备投票。

该功能的目标是保证每个区块提案的唯一性和合法性，并防止恶意领导者提议无效区块。

### 2. F\_{Vote} F\_{Vote}功能用于管理参与者对区块提案的投票过程，确保每个参与者在一个特定轮次中只能投票一次。具体要求如下：

* **接收投票**：每个参与者在接收到提案后，可以调用 F\_{Vote}对提案进行投票。
* **确保唯一性**： F\_{Vote}确保每个参与者在同一轮次中只能提交一票，避免重复投票的发生。
* **投票有效性检查**：F\_{Vote}会验证投票是否合法（如是否属于授权参与者的有效投票）。
* **记录投票结果**：F\_{Vote}记录所有投票，作为后续达成共识的依据。

该功能确保了投票过程的完整性和准确性，防止恶意参与者在一个轮次中多次投票或提交无效投票。

### 3. F\_{New\_view} F\_{New\_view}功能用于协调领导者选举和视图更替，确保当当前领导者失效时能够平稳地进入下一视图。具体要求如下：

* **触发视图更替**：当检测到当前领导者失效（如无法生成提案或未能及时达成共识）时，F\_{New\_view}会自动触发视图更替流程。
* **选择新领导者**：F\_{New\_view}确定新的领导者，使其成为下一个视图中的提案人。
* **视图同步**：确保所有参与者在视图更替后能够一致地同步到新的视图，避免分叉的发生。

该功能的目标是确保协议在领导者失效时可以自动、有效地切换视图，保持系统的正常运行和共识进度。

### 4. F\_{QC} F\_{QC}功能负责管理法定证书的生成和验证，确保只有在达成足够多的投票同意时才会形成证书。具体要求如下：

* **收集投票**：F\_{QC}收集并验证参与者的投票，以确认其数量达到法定人数要求。
* **生成法定证书**：一旦达成法定人数， F\_{QC}会生成一个合法的证书，并将其广播给所有参与者。
* **验证证书的有效性**：每次新的区块提案都需要包含上一个区块的法定证书。F\_{QC}会检查证书是否有效，以确保新提案的合法性。

该功能确保了系统在每个区块提案中都包含足够的参与者支持，防止篡改和伪造证书的攻击，从而维持共识协议的安全性和正确性。

1. **实体定义**

* **领导者（Leader）：** 在每一轮视图中有唯一的领导者负责提议区块并发起共识过程。
* **验证者（Validator）：** 验证者对领导者提议的区块进行验证并投票。每个诚实的验证者在满足协议条件的情况下参与共识投票。
* **拜占庭节点（Byzantine Nodes）：** 由敌手控制的恶意节点，可能会发送错误的区块提议或投票，试图破坏共识过程。

1. **定理**

定理一：如果两个区块B1和B2冲突，那么它们不能同时被不同的诚实节点提交。

证明：通过引入HotStuff中的锁机制，每个诚实节点在提交一个区块之前必须验证其投票的QC，并且只有在满足安全条件的情况下才能提交该区块。我们可以证明，在两个冲突的区块同时被提交的情况下，必然会导致矛盾，因为一个节点必须看到前一视图中的锁定状态，并避免与之冲突。

定理二：在拜占庭节点数量不超过f的情况下，诚实的领导者最终能够达成共识，并推动系统前进。

证明：HotStuff协议通过视图切换（view change）机制保证活性。当当前视图中的领导者无法达成共识时，协议会通过视图切换来选择下一个领导者。我们可以证明，只要网络在全局稳定时间（GST）后保持同步，正确的领导者将能够在有限时间内收集足够的投票并达成共识。

定理三：若敌手仅能访问 f<k个拜占庭节点，则其生成的部分签名不足以组合成完整签名 σ，伪造签名的概率可以忽略不计。

证明：阈值签名方案要求 k 个签名节点的部分签名才能生成 σ，在系统总共 n个节点中，假设有至多 f拜占庭节点（其中 f<k ），敌手所能利用的部分签名数量最多为 f，小于阈值 k。因此，即使敌手能够访问这些拜占庭节点的签名权限，其仍然无法获得足够的部分签名来生成有效的完整签名 σ。这意味着，敌手生成一个伪造签名的概率几乎为零。

定理四：任意节点只有在共识达成后才能知道最终输出的数据值。

证明：在 HotStuff 协议中，只有在决定阶段（decide phase）形成有效的 commitQC 后，领导者才会将最终的共识结果发送给所有其他节点。节点在收到包含 commitQC 的决定消息后才会将该数据值作为共识结果存储。由于协议的每个阶段均依赖于 (n - f) 个节点的投票，拜占庭节点无法在协议的决定阶段之前推断出最终数据值。此外，协议的安全性依赖于锁定机制（lockedQC），保证诚实节点在之前阶段无法在未达成共识的情况下确定数据值。因此，任意节点在共识达成之前无法知道最终的输出数据值，满足不可篡改性。

1. **下阶段安排**
2. 功能细节与伪代码结合起来
3. 进一步引入新的功能如clock等