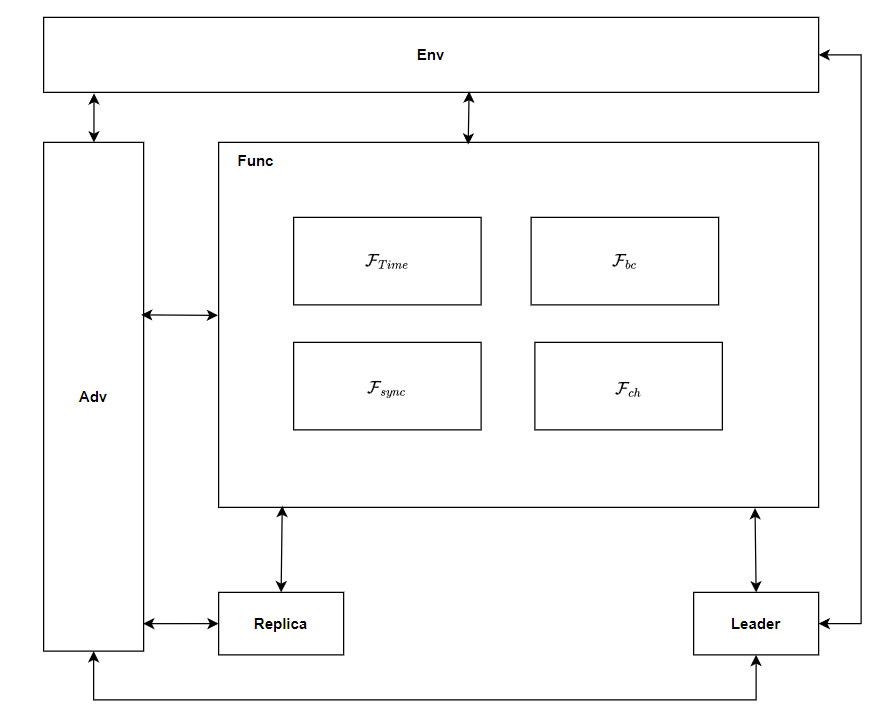
**Hotstuff建模进度**

**摘要：**

# 这篇文档是关于Hotstuff共识协议的v6建模进度报告：1、对框架图进行了修改，引入了新的通信通道功能F\_Ch满足网络级匿名性（例如，防止流量分析攻击和提取身份）2、对协议π\_hotstuff进行形式化建模，用功能F\_Proposal控制提案，F\_Time控制超时时间等对协议细节进行了封装处理。3、协议实现的uc建模完成度比例大概为70%

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. **F\_{Ch}**

我们定义一组参与方，其中 和 表示该集合中的两个参与方，分别是消息 的发送方和接收方。

是基于功能的参数进行定义的。消息标识符 由功能新选定。

在接收到来自 的输入 时，输出 给 。

在接收到来自 的 后，向发送 。

根据以下参数化函数设置 ：

– 对于 设置 , 。在接收到来自 的 后，向 发送 。

– 对于 设置 。

– 对于 设置 。

– 对于 设置 。

– 对于 设置 。在接收到来自 的 后，向 发送 。

– 对于 设置 。

在接收到来自 的 后，向 发送 。

在接收到来自 的 后，向 发送 。

在接收到来自 的 后，输出 给 。

在接收到来自 的 后，向 发送 。

1. **理想功能**

# **功能**

**网络延迟攻击**

**参数**：

* : 副本集
* ：最大超时持续时间
* : 当前视图中副本 的最大超时持续时间。
* : 理想时间功能。
* : 理想广播功能。
* : 理想同步功能。

**符号说明**：

### **接收消息** 来自 ：

1. 。
2. 如果 被破坏：

* 发送 给 。

1. 发送 给 ，等待响应 ：
   * 设置 ，其中 。

### **接收消息** 来自 ：

1. 发送 给 ，并等待响应 ：
   * 如果 ：
     + 设置
     + 发送 给 和 。
     + 发送 给 。
     + 发送 给 ，接收其响应 。
       - 如果 ，设置 。
       - 否则，重新执行此步骤。
   * 否则，发送 给 。
2. 如果 ：
   * 设置 。
   * 如果 ，发送 给 ，等待响应 ：
     + 创建 。
     + 如果没有收到来自 的 ：
       - 发送 给 。
       - 设置 。
     + 否则：
       - 设置 。
       - 发送 给 和 。
       - 发送 给 。
       - 发送 给 ，接收其响应 。
         * 如果 ，设置 。
         * 否则，重新执行此步骤。
3. 否则，忽略此消息。

### **接收消息** 来自 ：

1. 发送 给 ，并等待响应 ：
   * 如果 ：
     + 设置
     + 发送 给 和 。
     + 发送 给 。
     + 发送 给 ，接收其响应 。
       - 如果 ，设置 。
       - 否则，重新执行此步骤。
   * 否则，发送 给 。
2. 如果 ：
   * 如果 并且没有收到来自 的 ：
     + 发送 给 。
     + 发送 给 ，并等待响应 ：
       - 设置 。
   * 否则：
     + 设置 。
     + 发送 给 和 。
     + 发送 给 。
     + 发送 给 ，接收其响应 。
       - 如果 ，设置 。
       - 否则，重新执行此步骤。

### **收到消息** 来自 时：

1. 向 发送 ，并等待回复，回复形式为 ：
   * 如果 ：
     + 设置
     + 向 和 发送 。
     + 向 发送 。
     + 向 发送 ，接收其响应 。
       - 如果 ，设置 。
       - 否则重新执行此步骤。
   * 否则，向 发送 。
     + 如果 ：
       - 设置 。
       - 如果 ：
         * 创建 。
       - 如果没有收到来自 的 ：
         * 向 发送 。
         * 向 发送 ，并等待回复，回复形式为 ：

设置 。

* + - 否则：
      * 设置 。
      * 向 和 发送 。
      * 向 发送 。
      * 向 发送 ，接收其响应 。
        + 如果 ，设置 。
        + 否则重新执行此步骤。

1. 否则，忽略此消息。

### **收到消息** 来自 时：

1. 向 发送 ，并等待回复，回复形式为 ：
   * 如果 ：
     + 设置 。
     + 向 和 发送 。
   * 否则，向 发送 。
2. 如果 ：
   * 如果没有收到来自 的 ：
     + 设置 。
     + 向 发送 。
     + 向 发送 ，并等待回复，回复形式为 ：
       - 设置 。
   * 否则：
     + 设置 。
     + 向 和 发送 。
     + 向 发送 。
     + 向 发送 ，接收其响应 。
       - 如果 ，设置 。
       - 否则重新执行此步骤。
3. 否则，忽略此消息。

### **收到消息** 来自 时：

1. 向 发送 ，并等待回复，回复形式为 ：
   * 如果 ：
     + 设置
     + 向 和 发送 。
     + 向 发送 。
     + 向 发送 ，接收其响应 。
       - 如果 ，设置 。
       - 否则重新执行此步骤。
   * 否则，向 发送 。
     + 如果 ：
       - 设置 。
       - 如果 ：
         * 创建 。
       - 如果没有收到来自 的 ：
         * 向 发送 。
         * 向 发送 ，并等待回复，回复形式为 ：

设置 。

* + - 否则：
      * 设置 。
      * 向 和 发送 。
      * 向 发送 。
      * 向 发送 ，接收其响应 。
        + 如果 ，设置 。
        + 否则重新执行此步骤。

1. 否则，忽略此消息。

### **收到消息** 来自 时：

* 1. 向 发送 ，并等待回复，回复形式为 ：
     + 如果 ：
       - 设置 。
       - 向 和 发送 。
     + 否则，向 发送 。
  2. 如果 ：
     + 如果没有收到来自 的 ：
       - 设置 。
       - 向 发送 。
       - 向 发送 ，并等待回复，回复形式为 ：
         * 设置 。
     + 否则：
       - 设置 。
       - 向 和 发送 。
       - 向 发送 。
       - 向 发送 ，接收其响应 。
         * 如果 ，设置 。
         * 否则重新执行此步骤。
  3. 否则，忽略此消息。

### **收到消息** 来自 时：

* 1. 向 发送 ，并等待回复，回复形式为 ：
     + 如果 ：
       - 设置 。
       - 向 和 发送 。
       - 向 发送 。
       - 向 发送 ，接收其响应 。
         * 如果 ，设置 。
         * 否则重新执行此步骤。
     + 否则，向 发送 。
       - 如果 ：
       - 设置 。
       - 如果 ：
         * 创建 。
       - 如果没有收到来自 的 ：
         * 向 发送 。
         * 向 发送 ，并等待回复，回复形式为 ：

设置 。

* + - * 否则：
        + 设置 。
        + 向 和 发送 。
        + 向 发送 。
        + 向 发送 ，接收其响应 。

如果 ，设置 。

否则重新执行此步骤。

* 1. 否则，忽略此消息。

### **收到消息** 来自 时：

* 1. 向 发送 ，并等待回复，回复形式为 ：
     + 如果 ：
       - 设置 。
       - 向 和 发送 。
     + 否则，向 发送 。
  2. 如果 ：
     + 如果没有收到来自 的 ：
       - 通过 执行新命令。
       - 向 发送 。
       - 向 发送 ，接收其响应 。
         * 如果 ，设置 ，并设置 。
         * 否则重新执行此步骤。
     + 否则：
       - 设置 。
       - 向 和 发送 。
       - 向 发送 。
       - 向 发送 ，接收其响应 。
         * 如果 ，设置 。
         * 否则重新执行此步骤。
  3. 否则，忽略此消息。

1. **协议描述**

# 协议

|  | Leader |  | Replica |  |
| --- | --- | --- | --- | --- |
|  | 1: 发送 到 |  |  |  |
|  | 2: 发送 到 ，接收 |  |  |  |
|  | 3: 发送 到 |  |  |  |
|  | 4: 如果收到来自 的消息 ，发送 到 |  |  |  |
|  | 5: 否则，广播 给 Replicas |  |  |  |
|  | 6: 发送 到 |  |  |  |
|  |  |  | 7: 发送 到 |  |
|  |  |  | 如果收到来自 的消息 ，发送 到 |  |
|  |  |  | 8: 否则，如果 ，调用 |  |
|  | 9: 发送 到 |  |  |  |
|  | 10: 调用 |  |  |  |
|  |  |  | 11: 发送 到 |  |
|  |  |  | 12: 设置 |  |
|  |  |  | 13: 调用 |  |
|  | 14: 发送 到 |  |  |  |
|  | 15: 调用 |  |  |  |
|  |  |  | 16:发送 到 |  |
|  |  |  | 17: 设置 |  |
|  |  |  | 18: 调用 |  |
|  | 19: 发送 到 |  |  |  |
|  | 20: 调用 |  |  |  |
|  | 21: 发送 到 |  |  |  |
|  |  |  | 22: 发送 到 |  |
|  |  |  | 23: 如果收到来自 的消息 ，发送 到 |  |
|  |  |  | 24: 否则，通过 执行新命令 |  |
|  |  |  | 25: 发送 到 |  |

相关数据结构：

