arXiv:2601.03862v1 [cs.DC] 7 Jan 2026

Majorum：采用动态法定人数的涨落式共识

Francesco D'Amato Roberto Saltini 以太坊基金会
francesco.damato@ethereum.org roberto.saltini@ethereum.org
Thanh-Hai Tran Yann Vonlanthen 独立研究者 以太坊基金会
thanhhai1302@gmail.com yann.vonlanthen@ethereum.orgLuca
Zanolini 以太坊基金会luca.zanolini@ethereum.org

摘要

动态可用性是指即使诚实参与者离线后又重新加入，共识协议仍能保持活性的能力。一个众所周知的局限是，动态可用协议单独无法在网络分区或长时间异步期间提供强有力的安全性保证。涨落式协议 [SP21] 通过将动态可用协议与部分同步最终性协议相结合来解决这一问题，后者会不可逆地确定一个前缀。

我们提出 Majorum，这是一种涨落式构造，其动态可用组件基于基于法定人数的协议（TOB-SVD）。在乐观条件下，Majorum 最快三个时隙即可最终确定区块，并且每个时隙只需一个 *single* 投票阶段。具体而言，当条件持续有利时，每个时隙都会最终确定扩展先前已最终确定区块的下一个区块。

1 引言与相关工作

传统的拜占庭共识协议，例如 PBFT [4]和 HotStuff [19]，通常假设验证者集合固定且始终在线；在这种情况下，离线会被视为故障。

一些无需许可系统要求动态参与：诚实验证者可能离线，之后再重新加入。这一概念隐含于比特币 [14] 之中，并由 Pass 和 Shi 通过休眠模型 [18]将其形式化。如果某个协议在此类动态变化下仍能保持安全性和活性，我们就称其为动态可用协议。

一个局限是，动态可用协议单独使用时，无法在网络分区或长时间异步期间提供强安全性保证，同时又维持活性 [10]。涨落式协议 [15]通过结合动态可用的链增长协议与部分同步的最终性层来解决这一问题；后者不可撤销地确定一个前缀。以太坊采用了这种模式：LMD-GHOST [20]提供动态可用的链增长，而 Casper FFG [2] 则确定一个前缀（如 Gasper [3]所示）。

Momose 和 Ren [13]开启了通过适用于动态参与的法定人数技术研究<strong?deterministically safe 具有确定性安全保障的动态可用共识。这一研究方向促成了多种基于法定人数的动态可用协议，它们在不同方面进行了权衡 [11, 12, 9, 5]。

具体而言，D'Amato 等 [7] 提出了 TOB-SVD，这是一种以实用性和可扩展性为目标的基于多数法定人数的协议。TOB-SVD 可容忍高达 50% 的拜占庭故障，并且在最佳情况下只需一个投票阶段即可完成决策，相较于早期的多阶段设计有所改进。

在这项工作中，我们提出了 Majorum，这是一种涨落协议，其动态可用组件基于 TOB-SVD 构建。在乐观条件下，Majorum 只需每个时隙进行一轮投票，即可实现三个时隙的最终性。与我们所知唯一已在生产环境中部署的涨落协议——以太坊的协议——相比，这将乐观情况下达到最终性的时间从 64 个时隙缩短至 3 个时隙。此外，据我们所知，我们首次在涨落协议中以有原则的方式整合了两种互补的确认规则：传统的 κ 深度确认和快速确认。

1.1 技术概述

在大规模共识系统中，每一轮投票的成本都很高。当验证者数量达到数十万时，投票通常需要先经过聚合并重新广播，才能影响全局视图；因此，即使只是一个逻辑上的 投票阶段，端到端延迟也可能更接近 2Δ 而非 Δ。以太坊便体现了这种情况：证明在参与分叉选择之前需要先聚合并传播，因此每轮投票都会产生有效的2Δ延迟。这促使我们确立了首要设计目标：一种动态可用协议，能够在同步条件下以尽可能少的投票阶段取得进展。

我们的第二个目标是获得涨落式协议的典型保证 [16]。在同步条件下（即使参与度降低），系统仍应保持活性，并生成同步安全链。在网络分区或长时间异步的情况下，已最终确定 的前缀应始终保持安全，而只有未最终确定后缀可能被回滚。要实现这两个目标，需要一种动态可用协议，它具有高效的投票结构，并能与部分同步最终确定组件无缝配合。

基于这些原因，我们以 TOB-SVD 为起点。TOB-SVD 是一种基于多数派法定人数的动态可用共识协议；在同步模型中，它在这一设计空间内实现了最优的拜占庭容错能力，而且在最佳情况下，每次决策只需一个投票阶段。在本概述的其余部分，我们将介绍对 TOB-SVD 所做的修改，以及如何将由此得到的动态可用协议与部分同步最终性层配合使用。该层旨在最终确定 ——也就是说，为所生成链的一个前缀提供异步安全性——从而构成 Majorum。

Majorum 以及更广泛意义上的涨落式协议的吸引力在于，即使在低参与度场景下，只要处于同步状态，它们仍能保持活性——例如，当很大一部分验证者暂时离线时。以太坊是这种设计模式中一个值得关注的已部署实例：即使参与度下降，在同步状态下区块生产仍可继续（尽管最终性可能会停滞）。此外，以太坊的激励机制鼓励验证者保持在线并及时参与；然而，动态参与在实践中仍然具有现实意义，例如，共识客户端漏洞可能导致崩溃故障 [17]。

TOB-SVD 按时隙运行：在时隙 t开始时，当选提议者广播一个区块（提议阶段）； Δ之后，验证者进行投票（投票阶段）；再经过 Δ后，协议根据观察到的多数投票作出决策（决策阶段）。在原始协议中，在时隙 t 提出的提案会在时隙 $t+1$的决策阶段作出决策。所有安全性保证都是完全确定性的，不使用概率论证。此外，在 TOB-SVD 中，诚实提议者提出的提案总会得到所有在线诚实验证者的投票。

一个关键洞见是，确定性的决策阶段不会影响验证者的行为；相反，它充当一种显式的确认机制。因此，我们移除决策阶段，并在投票阶段通过 κ深度确认规则直接确认链。其中， κ 的取值确保在 κ 个时隙内，以压倒性概率选出一位诚实提议者。由此得到的变体提供的是概率安全性，而非确定性安全性（第 3.2 节）。然而，仅这一修改只能确认其提案至少在 κ 个时隙之前提出的链。为解决这一限制，我们引入一种用于快速确认的乐观机制（第 3.3 节）。在有利条件下，链可以在其提案提出的同一时隙内得到确认：具体而言，如果所有验证者都在线，且超过 $\frac{2}{3}n$ 的验证者为同一条链投票，那么该链便会快速确认，并可立即取代 κ 深度前缀而被采用。

这一新协议构成了潮汐式构造 Majorum 中的动态可用组件：一旦某个区块在动态可用层得到确认，该确认便可传递到部分同步层，作为最终确定的候选项。

整合这两个层十分简单。我们通过添加一个额外组件来扩展 TOB-SVD 的投票消息格式，该组件与 Casper FFG 的机制非常相似——Casper FFG 是以太坊共识协议中使用的最终性工具。具体而言，时隙 t 的投票阶段有两个目的：(1) 为时隙 t提出的提案投票；(2) 推动 TOB-SVD 中此前已确认链的最终确定。

按照这一设计，诚实提议者在时隙 t提出的提案，会在同一时隙内乐观地快速确认。随后，它将在时隙 $t+1$ 成为第一（共两轮）投票轮次的目标 。如果它在第一轮中获得超过 $\frac{2}{3}n$ 张投票，我们就称其为已获正当化。最后，它会在时隙 $t+2$ 的第二轮投票中最终确定。图1展示了组合协议的演进过程。尽管这比同槽最终性大约增加 $3\times$ ，但它避免了在单个时隙内进行多轮投票的需要——这种做法在规模扩大时会要求更长的时隙时长——同时保留了每个时隙一轮投票，从而简化了协议的时序假设和整体设计。

本文其余部分的安排如下。第 2 节介绍模型和背景。第3节介绍动态可用组件，回顾 TOB-SVD（第 3.1 节），并将其扩展至概率安全性和快速确认（第 3.2 节；证明见附录 A）。第 4 节介绍最终性工具（证明见附录B）。第5节给出完整的涨落协议及其安全性分析（见附录 C）。最后，我们在第6节总结全文。

2 模型与初步概念

2.1 系统模型

验证者。 我们考虑一组 n 验证者 v_1，……， v_n，每个验证者都具有唯一的密码学身份和已知的公钥。验证者执行共同协议。自适应概率多项式时间对手$\mathcal{A}$ 最多可以腐化 f 个验证者，并获取其状态。一旦被腐化，验证者即为对手控制的；否则为诚实。诚实验证者遵循协议，而对手控制的验证者则可任意行事。所有验证者拥有相等的权益，被证明存在不当行为的验证者将被罚没 （损失部分权益）。

网络。 我们假设部分同步：验证者拥有同步时钟，消息延迟起初没有上界，但在未知的全局稳定时间 （GST）之后，延迟上界为 Δ。

时间。 时间被划分为离散的轮次。一个时隙 包含 4Δ 轮次。对于轮次 r，其时隙为 $\mathsf{slot}(r) := \lfloor r/4\Delta \rfloor$。

提议者选举机制。 在每个时隙 t中，选择一个验证者 v_p 作为提议者，记为 v_p^t。如果 t明确，则省略上标。提议者选举机制必须满足：唯一性 （每个时隙恰好有一个提议者）、不可预测性 （身份在揭示之前保持隐藏）以及公平性 （每个验证者当选的概率为 $1/n$）。

交易池。 我们假设存在一个不断增长的交易集合，称为交易池，记作交易池，且所有验证者均可读取该集合。与之前的记法一致，我们用交易池^r 表示交易池 在时间 r的内容。

区块与链。 区块 是 $B = (b, p)$，其中 b 是区块体（一批交易）， $p \geq 0$是提出 B 时所在的时隙。创世区块 是 $B_{\text{genesis}} = (b_{-1}, -1)$，且没有父区块。每个区块都引用其父区块，子区块的时隙必须更大。一个区块确定一条链，即通往创世区块的路径。链以其末端区块标识。我们将 $\mathsf{ch}_1 \prec \mathsf{ch}_2$ 记作 ch_1 是 ch_2, 的严格前缀；如果两条链互不扩展，则称它们冲突。

在时隙 t，链 ch 的 κ深前缀记作 $\mathsf{ch}^{\lceil \kappa, t}$，是满足 $\mathsf{ch}'.p \leq t - \kappa$的最长前缀 ch'。当 t 明确时，我们记作 $\mathsf{ch}^{\lceil \kappa}$。如果 $\mathsf{ch}_1.p \leq \mathsf{ch}_2.p$，则定义 $\mathsf{ch}_1 \leq \mathsf{ch}_2$，其中相等仅当两条链完全相同。我们称 $\mathsf{ch}.p$ 为 ch 的高度 （或长度）。如果交易tx 出现在 ch 的某个区块中，则称其属于 ch，记作$tx \in \mathsf{ch}$。同理，$TX \subseteq \mathsf{ch}$ 表示TX 中的所有交易都包含在 ch 中。

提议与投票。 验证者可以提议 链，从而将区块提议推广到链。提案在每个时隙的第一轮中提出。为便于阅读，定义 $\mathsf{propose}(t) := 4\Delta t$。每个时隙还包含一个投票轮次，验证者在该轮次对提议链进行投票 。定义 $\mathsf{vote}(t) := 4\Delta t + \Delta$。

Gossip协议。 我们假设存在一种尽力而为的 gossip 原语，最终会将消息传递给所有验证者。诚实验证者之间的消息始终会被传递，且无法伪造。诚实验证者 v_i收到对抗性消息后，也会将其通过 v_i进行 Gossip传播。无论投票和区块是直接收到的，还是包含在其他消息中收到的，都会始终进行 Gossip传播。验证者收到投票后，也会传播相应的区块；收到提案后，则会传播其中的区块和投票。时隙 t 的提案仅在时隙 t的前 Δ 轮中进行 Gossip传播。

视图。 第 r轮的视图 记作 $\mathcal{V}^r$，是验证者截至第 r轮收到的所有消息的集合。我们用 $\mathcal{V}_i^r$ 表示验证者 v_i的视图。

休眠性。 在每一轮 r，对手可以指定哪些诚实验证者处于唤醒 或 休眠状态。休眠的验证者会暂停协议执行；消息会排队等待，直到它们醒来。在第 r 轮醒来后，验证者必须先完成加入协议 ，才能变为活跃 [6] （见下文段落）。对抗性验证者始终处于唤醒状态。因此，唤醒、休眠和活跃 这些状态仅适用于诚实验证者。休眠计划可由对手自适应地安排。在休眠模型中 [18], ，唤醒与活跃状态重合。最终，在某个未知的全局唤醒时间 （GAT）之后，所有验证者都将永远保持唤醒。

令 H_r、 A_r 和 $H_{r,\ r'}$ 分别表示第 r 轮的活跃验证者集合、第 r 轮的对抗性验证者集合，以及在第 $[r, r']$ 轮期间某个时刻处于活跃状态的验证者集合，即 $H_{r,\ r'} =$

$\bigcup_{i=r}^{r'} H_i$，其中 $H_i := \emptyset$ 若 $i < 0$. 我们令 $A_\infty = \lim_{t\to\infty} A_{\mathsf{vote}(t)}$，并注意到 $f = |A_\infty|$。除非另有说明，以下全文均假设 $f < \frac{n}{3}$。对于 GST 之后的每个时隙 t，我们要求

$$|H_{\mathsf{vote}(t-1)} \setminus A_{\mathsf{vote}(t)}| > |A_{\mathsf{vote}(t)}| \tag{1}$$

换言之，在 vote(t) 时尚未被腐化、且在 vote($t-1$) 时处于活跃状态的验证者数量，必须大于 vote(t) 时对抗性验证者的数量。

我们施加这一限制，是为了限制对手在时隙 $t-1$ 的投票阶段之后、但在后续时隙统计这些投票之前，能够腐化的诚实验证者数量。如果没有这样的上界，对手就可以在投票阶段结束后立即腐化过多的 诚实验证者，迫使它们进行矛盾投票。

与原始休眠模型的主要区别在于，我们的约束仅适用于活跃的 验证者，即那些已完成加入协议、因此在 vote(t) 之前至少已保持唤醒状态 Δ 轮的验证者。这反映了 TOB-SVD 为保证安全性所假定的 2Δ 稳定期；形式化论述请参阅原论文 [7]。在实践中，验证者刚刚唤醒时不太可能立即享有缓冲消息的即时传递：它必须请求恢复，并等待其他节点返还它错过的决策相关消息。我们只有在收到这些回复后才将验证者视为活跃，这至少需要 2Δ （发送需一个 Δ，接收需一个 Δ），而当需要获取大量状态时，耗时可能会长得多。因此，显式的 2Δ稳定窗口相较于实际恢复时间通常可以忽略不计。

我们将上述定义的对手模型称为广义部分同步休眠模型。上下文明确时，我们简称为休眠模型。当且仅当该模型中的一次执行满足休眠条件 (1) 时，我们称其为合规。

加入协议。 诚实验证者在第 r 轮醒来时，必须先运行加入协议。如果 v_i 在第 $r \in$ 轮醒来（vote($t-2$) $+\Delta$，vote($t-1$) $+\Delta$]），它会收到所有排队的消息并恢复执行，但在 vote(t) 轮之前不会发送任何消息；到该轮时，它便会变为活跃状态（除非它被腐化或再次进入休眠状态）。如果 v_i 被选为时隙 t 的领导者，但在提案时处于非活跃状态，则不会在该时隙提出提案。

2.2 安全性

我们分别将 λ 和 κ 视为密码学原语和协议的安全参数，并假设 $\kappa > 1$。我们考虑一个有限时间范围 T_{conf}，其长度关于 κ 为多项式级。若某事件发生失败的概率为 negl(κ)+negl(λ)，则称该事件以压倒性概率（w.o.p.）发生。密码学保证成立的概率为 negl(λ)，不过在后文中我们不再明确说明这一点。

定义 1（安全协议）。 当且仅当对于执行 $\mathcal{E}$ 中的任意一次执行、任意两个轮次 r、$r' \geq T_{\mathsf{sec}}$，以及分别处于轮次 r 和 r' 的任意两个诚实验证者 v_i 和 v_j （可能 $i = j$），若协议输出已确认链 Ch，则称该协议在执行 $\mathcal{E}$ 中的时间 T_{sec} 之后是安全的，当且仅当$\mathsf{Ch}_i^r \preceq \mathsf{Ch}_j^{r'}$ 或$\mathsf{Ch}_j^{r'} \preceq \mathsf{Ch}_i^r$。如果 $T_{\mathsf{sec}} = 0$ 和 $\mathcal{E}$ 包含所有部分同步执行，则称该协议始终安全。

定义 2（活性协议）。 当且仅当对于执行 $\mathcal{E}$ 中的任意一次执行、任意轮次 $r \geq T_{\mathsf{sec}}$ 和 $r_i \geq r + T_{\mathsf{conf}}$、时间 r 位于交易池中的任意交易 tx，以及在轮次 r_i 中活跃的任意验证者 v_i，$tx \in \mathsf{Ch}_i^{r_i}$，若协议输出已确认链 Ch，则称该协议在执行 $\mathcal{E}$ 中的时间 T_{sec} 之后具有活性，且确认时间为 T_{conf}。如果 $T_{\mathsf{sec}} = 0$ 和 $\mathcal{E}$ 包含所有部分同步执行，则称该协议始终具有活性。

定义 3 (安全协议 [6])。 如果某协议输出一条已确认链Ch，并且在时间 T_{sec}之后是安全的，且确认时间为 T_{conf}，那么我们称该协议在时间 T_{sec} 之后是安全的，并在时间 T_{sec} 之后保持活性，确认时间为 T_{conf}。如果一个协议始终既安全又保持活性，则称其始终安全。

现在我们回顾 [8] 中的动态可用性和重组韧性概念。

定义 4 (动态可用性)。 当且仅当某协议在时间 T_{dyn} 之后具有动态可用性，也就是说，它在时间 T_{dyn} 之后是安全的，且确认时间为 $T_{\mathsf{conf}} = O(\kappa)$。如果一个协议在GST $= 0$时始终安全，我们就称该协议具有动态可用性。

请注意，动态可用性以一定程度的对抗韧性为前提。在第 5 节的协议中，只要约束（1）成立，就能实现动态可用性。

定义 5 （重组韧性）。 如果在执行 $\mathcal{E}$ 中的时隙 t_{reorg} 和时间 T_{reorg}[1] 之后，对于任意 $\mathcal{E}$-执行、轮次 $r \geq T_{\mathsf{reorg}}$ 以及在 r 中诚实的验证者 v_i，由在提议(t) 中诚实的验证者于时隙 $t \geq t_{\mathsf{reorg}}$ 提议的任何链都不会与Ch_i^r冲突，则称协议在这些执行中具备重组韧性 。如果协议在时隙0 和时间 0之后满足这一条件，则称该协议具备重组韧性 。

可问责安全性。 除动态可用性外，我们还要求在发生安全性违规时具备问责能力。

定义 6 (可问责安全性)。 如果任何安全性违规都能通过基于消息记录构造的密码学证明，识别出至少 f^{acc} 个作恶验证者，且绝不错误指控遵循协议的诚实参与者，则该协议在弹性参数为 $f^{\mathrm{acc}} > 0$ 时具有可问责安全性 。

根据 CAP 定理 [10], ，任何共识协议都无法同时确保动态参与下的活性和网络分区情况下的安全性。因此，Neu、Tas 和 Tse [15] 提出了涨落 协议，该协议输出两条链：一条在同步性和动态参与下保持活性，另一条在异步性下保持安全性。

定义 7 (安全涨落协议)。 安全涨落协议 会输出一条chAva 可用链，具备动态可用性[2]，；还会输出一条chFin 最终确定链，该链始终安全，并在$\max(\mathsf{GST}, \mathsf{GAT})+O(\Delta)$之后，在 $T_{\mathsf{conf}} = O(\kappa)$条件下保持活性。因此，chFin 在$\max(\mathsf{GST},\mathsf{GAT})+O(\Delta)$之后，在$T_{\mathsf{conf}} = O(\kappa)$条件下是安全的。此外，对于每个诚实验证者 v_i 和每一轮 r，chFin_i^r 都是chAva_i^r的前缀。

我们的协议采用涨落方法 [16]: chAva 来自动态可用组件（第 3 节），而 chFin 则由类似 Casper FFG[2] 的 PBFT 风格协议最终确定（第 4 节）。

分叉选择函数。 分叉选择函数 FC 是确定性的：给定视图、链和时隙 t，它会输出一条链 ch。我们使用多数派分叉选择函数 （算法1），该函数会选择获得多数投票权重支持的最长链。

3 动态可用组件

我们首先介绍协议中的动态可用组件。在第 3.1 节中，我们回顾 TOB-SVD，并概述其运行方式和保证。在第 3.2 节中，我们介绍所做的修改，这些修改构成了一个概率安全、动态可用的共识协议。随后，我们引入快速确认的概念，并证明其动态可用性（附录 A）；这使其能够集成到第 5 节介绍、并在第 C 节中完整详述的涨落协议中。

3.1 回顾 TOB-SVD

我们回顾 D'Amato等人 [7]提出的确定性安全、动态可用的共识协议。完整的技术细节请参阅原始工作；此处我们将总结其运行方式、安全性保证，以及与我们修改为概率安全变体相关的方面。在回顾原始协议后，我们将展示如何引入快速确认并证明动态可用性，从而使其能够用于我们的涨落协议。

TOB-SVD [7] 的运行方式如下。在每个时隙 t中，第一轮通过一条$[\textsc{propose}, B, t, v_i]$ 消息提出一个提案（B），并在第三轮作出决策。在第二轮中，每个活跃验证者 v_i 都会为链 ch 投出一条 $[\textsc{vote}, \mathsf{ch}\ \cdot,\ t, v_i]$ 消息（投票消息的第三个组成部分在此省略；稍后介绍涨落协议时将引入该组成部分）。重要的是，在时隙 t 中提出的提案，会在时隙 $t+1$的决策阶段作出决策。所有安全性保证都是完全确定性的。

令 $\mathcal{V}$ 为时隙 t 的验证者视图。我们定义：

$$\mathcal{V}^{\mathsf{ch},t} := \left\{[\textsc{vote}, \mathsf{ch}', \cdot, \cdot, v_k] \in \mathcal{V}' : \mathcal{V}' = \mathsf{FIL}_{\mathrm{lmd}}(\mathsf{FIL}_{\text{1-exp}}(\mathsf{FIL}_{\mathrm{eq}}(\mathcal{V}), t)) \ \wedge\ \mathsf{ch} \preceq \mathsf{ch}'\right\},$$

其中，$\mathsf{FIL}_{\mathrm{eq}}$ 移除矛盾行为，$\mathsf{FIL}_{\text{1-exp}}$ 丢弃已过期的投票（即早于一个时隙的投票，这些投票不再被考虑），$\mathsf{FIL}_{\mathrm{lmd}}$ 仅保留每个验证者的最新投票。

[1]在本文中，我们始终有 $T_{\mathsf{reorg}} \geq 4\Delta t_{\mathsf{reorg}}$。[2]当我们说协议 Π 生成的链满足性质 P 时，我们的意思是协议 Π 本身满足 P。

因此，$\mathcal{V}^{\mathsf{ch},\,t}$ 是 支持扩展 ch 的链的最新、未过期且不存在双重签名的投票集合 .

该协议采用多数派分叉选择函数 MFC（在算法1中给出形式化定义）：给定视图 $\mathcal{V}$ 和 $\mathcal{V}'$、基础链 ch^C 以及时隙 t，它会返回满足以下条件的最长链 $\mathsf{ch} \succeq \mathsf{ch}^C$：$|\mathcal{V}^{\mathsf{ch},\,t} \cap (\mathcal{V}')^{\mathsf{ch},\,t}| > \frac{1}{2}|\mathsf{S}(\mathcal{V}',\ t)|$，其中，$\mathsf{S}(\mathcal{V}',\ t)$ 是在时隙 t于 $\mathcal{V}'$ 中出现且投票未过期的验证者集合。若不存在这样的链，则 MFC 返回 ch^C。

算法 1 MFC，多数派分叉选择函数

1：函数 $\mathrm{MFC}(\mathcal{V}, \mathcal{V}', \mathsf{ch}^C, t)$ 2：返回满足以下条件的最长链 ch $\succeq \mathsf{ch}^C$：3：$\mathsf{ch} = \mathsf{ch}^C \vee \left|\mathcal{V}^{\mathsf{ch},t} \cap (\mathcal{V}')^{\mathsf{ch},t}\right| > \frac{|\mathsf{S}(\mathcal{V}',t)|}{2}$ | 4：函数 $\mathsf{S}(\mathcal{V}, t)$ 5：返回 $\{v_k : [\mathrm{VOTE}, \cdot, \cdot, \cdot, v_k] \in \mathcal{V}' \nprec \mathcal{V}' \geq \mathrm{FIL}_{1\text{-exp}}(\mathcal{V}, t)\}$

验证者 v_i 执行的协议分为四个阶段：

1. **提议 ($4\Delta t$)：** 如果 v_p 是时隙 t 的提议者，它会通过 Gossip协议广播 $[\mathrm{PROPOSE}, \mathsf{ch}_p, t, v_p]$，其中 $\mathsf{ch}_p \succeq \mathrm{MFC}(\mathcal{V}_p, \mathcal{V}_p, B_{\mathrm{genesis}}, t)$，且 $\mathsf{ch}_p.p = t$。实际中，$\mathrm{MFC}(\mathcal{V}_i, \mathcal{V}_i, B_{\mathrm{genesis}}, t)$ 是 ch_p 的父区块，但我们允许更一般的情况：ch_p 是该父区块的任意扩展，并带有 $\mathsf{ch}_p.p = t$。

2. **投票 ($4\Delta t + \Delta$)：** 每个验证者计算 $\mathsf{ch}^{\mathsf{MFC}} := \mathrm{MFC}(\mathcal{V}_i^{\mathrm{frozen}}, \mathcal{V}_i, B_{\mathrm{genesis}}, t)$，然后泛洪传播投票消息 $[\mathrm{VOTE}, \mathsf{ch}', \cdot, t, v_i]$，其中 ch' 是在时隙 t 提出的链，并在 $\mathsf{ch}^{\mathsf{MFC}}$ 的基础上扩展出 $\mathsf{ch}'.p = t$；如果不存在有效提案，则为 $\mathsf{ch}' = \mathsf{ch}^{\mathsf{MFC}}$。

3. **决策（$4\Delta t + 2\Delta$）：** 设置 $\mathsf{Ch}_i = \mathrm{MFC}(\mathcal{V}_i'', \mathcal{V}_i, B_{\mathrm{genesis}}, t)$，更新 $\mathcal{V}_i'' = \mathcal{V}_i$，并存储 $\mathcal{V}'_i$.

4. **冻结（$4\Delta t + 3\Delta$）：** 设置 $\mathcal{V}_i^{\mathrm{frozen}} = \mathcal{V}_i$ 并将其存储以供下一个时隙使用。

3.2 概率安全变体

刚才介绍的协议的一种概率安全变体，是通过注意到决策不会影响验证者的行为这一点得到的：决策仅作为确定性安全性的确认机制。在时隙t中，决策阶段会存储 $\mathcal{V}''$ 以供下一个时隙使用，但除此之外没有任何影响。移除这一阶段可以简化协议，同时保持概率安全性。主要区别在于引入了新的概率确认规则——除了移除决策阶段，从而将时隙时长从 4Δ 缩短为 3Δ 之外——在投票阶段，协议将 MFC输出的链中深度为 κ的前缀指定为已确认。参数 κ 的选择应确保在 κ 个时隙内，以压倒性概率选出诚实提议者，从而确保共识可靠推进，并且该前缀之后不会被回滚。

1. 提议（$3\Delta t$）：如果 v_i 是时隙 t的提议者，则通过 Gossip协议泛洪传播 $[\mathrm{PROPOSE}, \mathsf{ch}_p, t, v_i]$，其中包含 $\mathsf{ch}_p \succeq \mathrm{MFC}(\mathcal{V}_i, \mathcal{V}_i, B_{\mathrm{genesis}}, t)$ 和 $\mathsf{ch}_p.p = t$。2. 投票（$3\Delta t+\Delta$）：计算 $\mathsf{ch}^{\mathsf{MFC}} := \mathrm{MFC}(\mathcal{V}_i^{\mathrm{frozen}}, \mathcal{V}_i, B_{\mathrm{genesis}}, t)$，然后通过 Gossip协议泛洪传播投票消息 $[\mathrm{VOTE}, \mathsf{ch}', \cdot, t, v_i]$，其中 ch' 是时隙 t中提出的、通过 $\mathsf{ch}'.p = t$扩展 $\mathsf{ch}^{\mathsf{MFC}}$ 的链；如果不存在有效提案，则为 $\mathsf{ch}' = \mathsf{ch}^{\mathsf{MFC}}$。将确认链设为 κ深度的前缀 $\mathsf{Ch}_i = \left(\mathsf{ch}^{\mathsf{MFC}}\right)^{\lceil\kappa}$。

3. **冻结（$3\Delta t + 2\Delta$）：** 设置 $\mathcal{V}_i^{\mathrm{frozen}} = \mathcal{V}_i$，将其存储起来，并推进到时隙 $t + 1$。

在同步性条件下，约束（1）（调整为时长为 3Δ 的时隙）成立。在时隙 t 中有诚实提议者时，提议链 ch_p 会扩展锁定（即 $\mathsf{ch}_i^{3\Delta t+\Delta}$），确保所有诚实验证者都为 ch_p 投票。投票达成一致可带来重组韧性，进而在 κ深规则下提供概率安全性。详情见附录中对最终协议变体的证明。

3.3 概率安全变体与快速确认

上述协议仅确认至少在过去 κ 个时隙提出的链。现在，我们通过一种机制对其进行扩展，实现**快速确认**：在某些有利条件下，链可以在其提出的同一时隙内得到确认。具体而言，如果所有验证者都处于活跃状态，且超过 $\frac{2}{3}n$ 的验证者为同一条链投票，那么该链就会**快速确认**，并可立即取代 κ深前缀。

与原有规则相比，这一确认规则的优势在于，现在在时隙 t 提议的链可在时隙 t 就得到确认，而不必像原始设计那样等到时隙 $t+1$。其缺点是，与原有规则不同，这种快速确认要求投票阶段全员参与。不过，由于协议仍保留 κ 深度确认机制，未能全员参与只会导致协议退回到确认由 MFC 输出的链的 κ 深前缀。

这种能力 （在算法 2 中形式化）是我们涨落协议的一个构建模块。
请注意，确认 κ 深的内容就足够了，但这会导致延迟较高；而快速确认虽已足够，却会在参与度较低时停滞不前。

协议。算法 2 定义了 fastconfirmsimple($\mathcal{V}$, t) $\mapsto$ (ch^C, Q^C)，其中，ch^C 是由超过 $\frac{2}{3}n$ 个验证者根据视图 $\mathcal{V}$ 在时隙 t 投票支持的链，而 Q^C 是由相应 VOTE 消息组成的**仲裁证书**。如果不存在这样的链，则返回 (B_{genesis}, $\emptyset$)。

该协议还依赖函数 Extend(ch t)：如果 $\mathsf{ch}.p < t$，则 Extend(ch t) 会生成一条长度为 t 的链 ch′，该链扩展 ch 并包含**交易池** $\mathsf{propose}(t)$ 中的所有交易。如果 $\mathsf{ch}.p \geq t$，则该函数的行为未作规定。

每个诚实验证者 v_i 都会维护三个本地对象。首先是**消息集** $\mathcal{V}_i^{\text{frozen}}$，即每个时隙中在时间 3Δ 对 $\mathcal{V}_i$ 进行快照得到的结果。其次是**链变量** $\mathsf{ch}_i^{\text{frozen}}$，定义为时隙 t 中时间 3Δ 时快速确认链中最长的一条；该变量在时隙 $t+1$ 的 0 到 Δ 之间，使用该时隙 $t+1$ 的 PROPOSE 消息进行更新。最后是**已确认链** Ch_i，它会在每个时隙的时间 Δ 更新，也可能在 2Δ 再次更新。

为便于记法，若 var 是一个状态变量，我们用 var^r 表示其在第 r轮之后的值；如果它已有上标，则写作 $\mathcal{V}_i^{\text{frozen},\,r}$。为便于阅读，我们将 fconf($t$) := $4\Delta t + 2\Delta$ 和 merge(t) := $4\Delta t + 3\Delta$分别简写为相应形式。

算法 2 的流程如下：

1. **提议**（$4\Delta t$）：提议者 v_p 广播 $[\text{PROPOSE}, \mathsf{ch}p, \mathsf{ch}^C, Q^C, \cdot, t, v_p]$，其中 $\mathsf{ch}p$ =通过 Extend($\mathsf{ch}^{\mathsf{MFC}}$, t) 得到，而 $\mathsf{ch}^{\mathsf{MFC}}$ = MFC为($\mathcal{V}_p$, $\mathcal{V}_p$, B_{genesis}, t)。元组 (ch^C, Q^C) 通过快速确认简单算法($\mathcal{V}_p$, t)计算得出。

2. **投票**（$4\Delta t + \Delta$）：收到有效提案（其中 Q^C 为 ch^C 和 $\mathsf{ch}_p.p = t$提供证明）后，验证者v_i 更新 $\mathsf{ch}_i^{\text{frozen}}$。随后，它使用以下内容计算 MFC: (i) $\mathcal{V}_{i}^{\text{frozen},\,\mathsf{vote}(t)}$，(ii) $\mathcal{V}_i^{\mathsf{vote}(t)}$，以及 (iii) $\mathsf{ch}_i^{\text{frozen}}$。根据构造，投票阶段的输出始终是提议者链的前缀（分级交付性质 [7]）。如果提案扩展了这条链，v_i 就为其投票；否则，它会为本地分叉选择投票。它还会将 Ch_i 更新为以下两者中较长者：(i) 其 MFC输出的 κ深前缀，以及 (ii) 先前的 Ch_i，除非后者不是前者的前缀。

3. **快速确认**（fconf(t)）：如果至少有 $2/3n$ 个验证者为链 $\mathsf{ch}^{\text{cand}}$投票，那么 v_i 会通过设置 Ch_i =$\mathsf{ch}^{\text{cand}}$ 来快速确认该链。

4. **合并**（merge(t)）：在轮次 $4\Delta t + 3\Delta$，v_i 会更新 $\mathcal{V}_i^{\text{frozen}}$和 $\mathsf{ch}_i^{\text{frozen}}$，以供时隙 $t+1$使用。

在附录 A 中，我们证明算法 2 满足重组韧性（定理 1）和动态可用性（定理 2）。

我们首先证明：如果时隙 t 中的所有活跃验证者都为一条扩展 ch 的链发送 VOTE，那么时隙 t+1 中的每个活跃验证者也会为一条扩展 ch 的链投票，因为 MFC 必然会输出这样的扩展（引理 2）。由此可知，如果任何验证者在时隙 t 快速确认了一条链 ch，那么后续时隙中的所有活跃验证者都将通过 MFC 输出 ch 的扩展（引理 3）。对于诚实验证者提议的链，也有类似的保证（引理 4）。

这些结果意味着，在第 r_i 轮，任何诚实验证者的已确认链——即使该验证者在 r_i 处处于非活跃状态——始终是后续轮次中处于活跃状态的任意验证者的 MFC 输出的前缀（引理 5）。这一性质是证明重组韧性的关键要素。

算法 2 TOB-SVD 的概率安全变体，支持快速确认 - v_i 的代码

输出
    $\mathsf{Ch}_i \leftarrow B_{\text{创世}}$ ：验证者 v 的已确认链 $_i$
状态
    $\mathcal{V}_i^{\text{frozen}} \leftarrow \{B_{\text{genesis}}\}$：$\mathcal{V}$ 在时间 $4\Delta t + 3\Delta$ 的快照
    $\mathsf{ch}_i^{\text{frozen}} \leftarrow B_{\text{genesis}}$：快速确认链在时间 $4\Delta t + 3\Delta$ 的快照
函数 快速确认简单算法($\mathcal{V}$, t)
    令 $\mathsf{fast}^{\text{cands}} := \{\max |\{v_j \mathsf{fast} \exists$: $|\{v_j$: $\exists \mathsf{ch}' \succeq \mathsf{ch}$: $[\text{VOTE}, \mathsf{ch}', \cdot, t, \cdot] \in \mathcal{V}\}| \geq \frac{2}{3}n\}$
    如果 $\mathsf{fast}^{\text{cands}} \neq \emptyset$ **则**
        令 $\mathsf{fast}^{\text{cand}} := \max\left(\mathsf{fast}^{\text{cands}}\right)$
        令 $Q := \{[\text{VOTE}, \mathsf{ch}', \cdot, t, \cdot] \in \mathcal{V} : \mathsf{ch}' \succeq \mathsf{fast}^{\text{cand}}\}$
        返回 (快速$^{\text{cand}}$, Q)
    else
        返回 (B_{genesis}, $\emptyset$)
PROPOSE
在第 $4\Delta t$ **轮执行**
    如果 $v_i = v_p^t$ **then**
        令 (ch^C, Q^C) := 快速确认简单算法($\mathcal{V}_i$, $t-1$)
        let $\mathsf{ch}^{\mathsf{MFC}} := \text{MFC}(\mathcal{V}_i, \mathcal{V}_i, \mathsf{ch}^C, t)$
        令 ch_p := 扩展($\mathsf{ch}^{\mathsf{MFC}}, t$)
        通过 Gossip协议发送消息 $[\text{PROPOSE}, \mathsf{ch}_p, \mathsf{ch}^C, Q^C, \cdot, t, v_i]$
VOTE
在第 $4\Delta t + \Delta$ **轮执行**
    let 通道$^{\mathsf{MFC}} := \text{MFC}$ ($\mathcal{V}_i^{\text{frozen}}$, $\mathcal{V}_i$, 通道$_i^{\text{frozen}}$, t)
    $\mathsf{Ch}_i \leftarrow \max(\{\mathsf{ch} \in \{\mathsf{Ch}_i, (\mathsf{ch}^{\mathsf{MFC}})^{\lceil \kappa}\} : \mathsf{ch} \preceq \mathsf{ch}^{\mathsf{MFC}}\})$
    令 ch := PROPOSEd 从时隙 t 扩展 $\mathsf{ch}^{\mathsf{MFC}}$ 和
        若存在，则使用 $\mathsf{ch}.p = t$，否则使用 $\mathsf{ch}^{\mathsf{MFC}}$ 否则
    通过 Gossip协议发送消息 $[\text{VOTE}, \mathsf{ch}, \cdot, t, v_i]$
FAST CONFIRM
在轮次 $4\Delta t + 2\Delta$ **执行**
    令 (快速确认$^{\text{cand}}, Q$) := 快速确认简单算法($\mathcal{V}_i, t$)
    若 $Q \neq \emptyset$ **则**
        $\mathsf{Ch}_i \leftarrow$快速确认$^{\text{cand}}$
MERGE
在第 $4\Delta t + 3\Delta$ **轮执行**
    $\mathcal{V}_i^{\text{frozen}} \leftarrow \mathcal{V}_i$
    ($\mathsf{ch}_i^{\text{frozen}}, \cdot$) $\leftarrow$ 快速确认简单算法($\mathcal{V}_i, t$)

收到消息 $[\text{PROPOSE}, \mathsf{ch}_p, \mathsf{ch}_p^C, Q_p^C, \cdot, t, v_p]$
        在 $[4\Delta t, 4\Delta t + \Delta]$ **中的任意轮次**
    若 有效($[\text{PROPOSE}, \mathsf{ch}_p, \mathsf{ch}_p^C, Q_p^C, \cdot, t, v_p]$) $\wedge\, \mathsf{ch}_i^{\text{frozen}} \preceq \mathsf{ch}_p^C$ **then**
        $\mathsf{ch}_i^{\text{frozen}} \leftarrow \mathsf{ch}_p^C$

我们进一步证明，诚实提议者可以在其提案中纳入池中的所有交易（引理 6）。结合上述引理，可以得到以压倒性概率成立的动态可用性（定理 2）。

我们证明了快速确认是可以实现的（引理 7）。尽管这并非正式的安全性属性，但这一结果对于协议在实践中的响应性至关重要。

算法 3 证明与最终确定

```
1:  函数 有效(C_1 → C_2)
2:      返回 ∧ C_1.ch ⪯ C_2.ch
3:           ∧ C_1.c < C_2.c

4:  函数 J(C, V)
5:      返回 ∨ C = (B_genesis, 0)
6:           ∨ ∃S, M ⊆ V :  ∧ J(S, V)
7:                          ∧ 有效(S → C)
8:                          ∧ |{v_k :  [VOTE, ·, S → C, ·, v_k] ∈ M}| ≥ (2/3)n

9:  函数 F(C, V)
10:     返回 ∨ C = (B_genesis, 0)
11:          ∨ ∧ J(C,  V)
12:            ∧ ∃T, M ⊆ V :  ∧ T.c = C.c + 1
13:                           ∧ 有效(C → T)
14:                           ∧ |{v_k :  [VOTE, ·, C → T, ·, v_k] ∈ M}| ≥ (2/3)n
```

4 最终性组件

我们的协议中的最终性组件会在每个时隙最终确定一条链，该链扩展了前一个时隙中已最终确定的链。我们引入以下术语。

检查点。 检查点 是一个元组 $\mathcal{C} = (\mathsf{ch}\ c)$，其中 ch 是一条链， c 是 ch 被提议以获得证明的时隙。根据构造，$c \geq \mathsf{ch}.p$。我们用 c 对检查点进行排序： $\mathcal{C} \leq \mathcal{C}'$ 当且仅当 $\mathcal{C}.c \leq \mathcal{C}'.c$，且只有检查点完全相同时才相等。

证明。 一组 FIN-VOTE构成超多数链接 $\mathcal{C} \to \mathcal{C}'$ ，条件是至少有 $2n/3$ 个验证者从 $\mathcal{C}$ 向 $\mathcal{C}'$发出FIN-VOTE。如果检查点满足以下任一条件，则称其已获正当化： (i) $\mathcal{C} = (B_{\text{genesis}},\ 0)$；或 (ii) 可以通过一系列超多数链接从 ($B_{\text{genesis}},\ 0$) 到达该检查点。当且仅当存在某个已获正当化的检查点 $\mathcal{C}$ 满足$\mathcal{C}.\mathsf{ch} = \mathsf{ch}$ 时，链 ch 才是已获正当化的。给定视图 $\mathcal{V}$，若 $\mathcal{V}$ 包含相应的超多数链接，则 $\mathcal{C}$ 在 $\mathcal{V}$ 中是已获正当化的。 $\mathcal{V}$ 中时隙最大的已获正当化检查点是最大已论证检查点 $\mathcal{GJ}(\mathcal{V})$，其链是最大已证明链。若有并列，则任意打破平局。

最终确定。 如果检查点 $\mathcal{C}$ 已获正当化，且存在超多数链接 $\mathcal{C} \to \mathcal{C}'$ 与$\mathcal{C}'.c = \mathcal{C}.c + 1$，则该检查点已最终确定 。如果某条链属于某个已最终确定的检查点，则该链已最终确定。根据定义，(B_{genesis},0) 已最终确定。函数 F($\mathcal{V}$,$\mathcal{C}$)（算法 3）当且仅当 $\mathcal{C}$ 在 $\mathcal{V}$ 中已最终确定时返回 true。时隙最高的已最终化检查点是 $\mathcal{GF}(\mathcal{V})$。链 ch 在 $\mathcal{V}$ 中已最终确定，当且仅当 $\mathsf{ch} \preceq \mathcal{GF}(\mathcal{V})$。ch。

最终性投票。 验证者会发出两种类型的投票。标准 VOTE消息嵌入一个 FIN-VOTE，即将源 $\mathcal{C}_1$ 与目标 $\mathcal{C}_2$相连的元组 $[\text{FIN-VOTE}, \mathcal{C}_1, \mathcal{C}_2, v_i]$ 。如果 $\mathcal{C}_1.c < \mathcal{C}_2.c$ 且$\mathcal{C}_1.\mathsf{ch} \preceq \mathcal{C}_2.\mathsf{ch}$，则 FIN-VOTE 有效 。我们将此类投票简记为 $\mathcal{C}_1 \to \mathcal{C}_2$。

矛盾行为。 如果验证者 v_i 在同一时隙 t 针对不同的链 $\mathsf{ch} \neq \mathsf{ch}'$发送了两条 VOTE消息，则该验证者存在双重签名 。

验证者 v_i 若满足以下任一条件，就会因两个不同的 FIN-VOTES $\mathcal{C}_1 \to \mathcal{C}_2$和 $\mathcal{C}_3 \to \mathcal{C}_4$ 而受到罚没（如第 2 节$_1$ 所述）： $\mathbf{E_1}$ （双重投票），如果 $\mathcal{C}_2.c = \mathcal{C}_4.c$，这意味着验证者不得针对同一个检查点时隙发送不同的 FIN-VOTES；或者$\mathbf{E_2}$ （环绕投票），如果$\mathcal{C}_3.c < \mathcal{C}_1.c < \mathcal{C}_2.c < \mathcal{C}_4.c$。

在附录 B 中，我们证明：如果检查点 ($\mathsf{ch}j,\ c_j$) 已获正当化，而另一个检查点 ($\mathsf{ch}f,\ c_f$) 已在 $c_j \geq c_f$ 时最终确定，那么 $\mathsf{ch}j$ 要么扩展 $\mathsf{ch}f$，要么至少有 $n/3$ 名验证者违反了两项罚没条件之一（引理 8）。特别是，我们证明，如果两条冲突链最终都被最终确定，那么至少有 $n/3$ 名验证者行为不端（引理 9）。

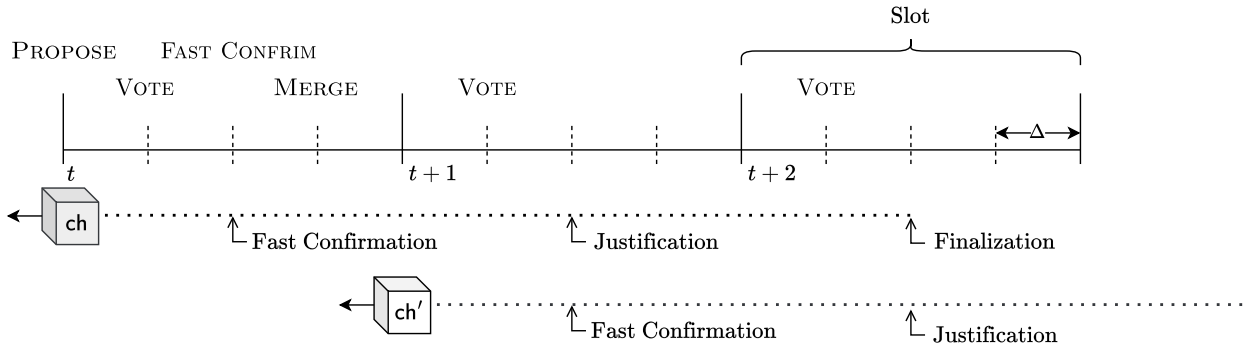


图1：图中概述了时隙结构，并突出显示了与提案的快速确认和最终确定相关的阶段。提案 ch 在第一个时隙 t（左上方）中提出，并在同一时隙内得到快速确认。随后，它在时隙 $t+1$ 中获得正当化，并在时隙 $t+2$ 中最终确定。同样，下一个提案 ch′ 在时隙 $t+1$ 中提出；一旦 ch 获得正当化，该提案便得到快速确认；一旦 ch 最终确定，该提案便获得正当化。这说明了确认和最终确定过程如何在连续时隙之间以流水线方式推进。

算法 4 Majorum 的状态与输出变量（算法 5）

1：**输出**2：$\mathsf{chAva}_i \leftarrow B_{\text{genesis}}$：可用链3：$\mathsf{chFin}_i \leftarrow B_{\text{genesis}}$：最终确定链4：**状态**5：$\mathcal{V}_i^{\text{frozen}} \leftarrow \{B_{\text{genesis}}\}$：$\mathcal{V}$ 在时间 $4\Delta t + 3\Delta$ 的快照6：$\mathsf{ch}_i^{\text{frozen}} \leftarrow B_{\text{genesis}}$：时间 $4\Delta t + 3\Delta$ 时快速确认链的快照7：$\mathcal{GJ}_{\text{冻结}_i} \leftarrow (B_{\text{genesis}}, 0)$：最新冻结的最大已论证检查点

5 Majorum

现在我们介绍 Majorum（算法 5），它将算法 2 中动态可用且具备重组韧性的协议与第 4 节中的最终性组件相结合，从而构成一种安全涨落协议。我们通过强调 Majorum 与算法 2 的主要区别来进行说明。图1概述了正常路径执行及相应的时隙结构。

函数快速确认($\mathcal{V}$, t)在算法 2 的快速确认简单算法($\mathcal{V}$, t)基础上进行了扩展：如果快速确认链扩展了 $\mathcal{GJ}(\mathcal{V})$。ch，则返回相同的输出；否则返回 ($\mathcal{GJ}(\mathcal{V})$。ch $\emptyset$)。

与算法 2 相比，验证者会维护一个额外变量：检查点变量 $\mathcal{GJ}_i^{\text{frozen}}$，其定义为在 v_i的视图中，时隙 $t-1$的时间 3Δ 所对应的最大已证明合理的检查点。该变量会在时隙 t 的 $[0, \Delta]$ 期间根据 PROPOSE 消息更新。

Majorum 会输出两条链：

1. **可用链** chAva_i**：** 对应于算法 2 中的已确认链 Ch_i 。

2. **最终确定链** chFin_i**：** 在快速确认轮次中设为 $\mathcal{GF}(\mathcal{V}_i)$。ch；在投票轮次中，设为满足 $\mathsf{chFin}_i \preceq \mathsf{ch} \preceq \mathcal{GF}(\mathcal{V}_i)$。ch 且 $\mathsf{chFin}_i \preceq \mathsf{chAva}_i$的最长链 ch。

与算法 2 相比，主要变化如下：

1. **提议：** 时隙 t 的提议者会在其当前视图中，将最大已论证检查点纳入初始化消息 PROPOSE 。

2. **投票：** 仅当提案 $[\text{PROPOSE}, \mathsf{ch}_p, \mathsf{ch}^C, Q^C, \mathcal{GJ}_p, t, v_p]$ 在 v_i 的视图中已获正当化时，该提案才有效。为允许 FIN-VOTE 累积，验证者会延迟处理此类提案，直到投票(t)。如果有效且 $\mathcal{GJ}_p \succ \mathcal{GJ}_i^{\text{frozen}}$，则 v_i 同时更新 $\mathcal{GJ}_{\text{冻结}_i}$ 和 $\mathsf{ch}_i^{\text{frozen}}$（确保 $\mathsf{ch}_i^{\text{frozen}} \succeq \mathcal{GJ}_i^{\text{frozen}}$）。随后，可用链 chAva_i 被设置为以下各项中的最大值：(i) 其先前的值；(ii) 深度为 κ 的前缀

算法 5 Majorum——验证者代码 v_i1：函数 快速确认($\mathcal{V}, t$) 2：令 (ch^C, Q) := 快速确认简单算法($\mathcal{V}, t$) 3：如果 $\mathsf{ch}^C \succeq \mathcal{GJ}(\mathcal{V})$.ch 则返回 ($\mathsf{ch}^C, Q$) 4：否则返回 ($\mathcal{GJ}(\mathcal{V})$.ch ∅)PROPOSE5：在轮次 $4\Delta t$ 执行 6：如果 $v_i = v_p^t$ 则 7：令 (ch^C, Q^C) := 快速确认($\mathcal{V}_i, t-1$) 8：令 $\mathsf{ch}^{\mathsf{MFC}} := \mathrm{MFC}(\mathcal{V}_i, \mathcal{V}_i, \mathsf{ch}^C, t)$，并令 ch_p := 扩展($\mathsf{ch}^{\mathsf{MFC}}, t$) 9：通过Gossip协议发送消息 [PROPOSE, ch_p、ch^C、 Q^C、 $\mathcal{GJ}(\mathcal{V}_i)$、 t、 v_i] 10：在轮次 $4\Delta t + \Delta$ 执行 11：令 $\mathsf{ch}^{\mathsf{MFC}} := \mathrm{MFC}(\mathcal{V}_i^{\text{frozen}}, \mathcal{V}_i, \mathsf{ch}_i^{\text{frozen}}, t)$ 12：chAva $_i \leftarrow$ 取最大值({ch ∈ {chAva$_i$、($\mathsf{ch}^{\mathsf{MFC}}$)⌈κ冻结$_i$ ·ch} : ch $\preceq \mathsf{ch}^{\mathsf{MFC}}$}) 13：chFin$_i \leftarrow$ 取最大值({ch：ch $\preceq$ chAva$_i \wedge$ ch $\preceq \mathcal{GF}(\mathcal{V}_i)$.ch}) 14：如果 $\mathcal{GJ}_i^{\text{frozen}}.c = t-1$15：令 $\mathcal{T}$:= (chAva$_i$, t) 16：否则 17：令 $\mathcal{T}$:= ($\mathcal{GJ}$冻结$_i$ · ch t) 18：令 ch：= PROPOSE从时隙 t 开始、扩展 $\mathsf{ch}^{\mathsf{MFC}}$ 的链 19：并令其包含 ch.$p = t$（如果存在），否则包含 $\mathsf{ch}^{\mathsf{MFC}}$ 20：通过Gossip协议发送消息 [VOTE, ch、 $\mathcal{GJ}_i^{\text{frozen}} \to \mathcal{T}$、 t、 v_i] 21：在轮次 $4\Delta t + 2\Delta$ 执行 22：令 (快速$^{\text{cand}}$, ·) := 快速确认($\mathcal{V}_i, t$) 23：如果 chAva$_i \not\succeq$快速$^{\text{cand}}$ ，则 chAva$_i \leftarrow$ 快速$^{\text{cand}}$24：chFin $_i \leftarrow \mathcal{GF}(\mathcal{V}_i)$.chMERGE25：在轮次 $4\Delta t + 3\Delta$ 执行 26：$\mathcal{V}_i^{\text{frozen}} \leftarrow \mathcal{V}_i$27：($\mathsf{ch}_i^{\text{frozen}}$, ·) ← 快速确认 ($\mathcal{V}_i, t$) 28：$\mathcal{GJ}$ 冻结 $_i$ ← $\mathcal{GJ}(\mathcal{V}_i)$ 29：收到消息 [PROPOSE, ch_p 、ch_p^C、 Q_p^C、 $\mathcal{GJ}_p$、 t时 30：在 $[4\Delta t, 4\Delta t + \Delta]$ 的任意轮次 31：当轮次 $4\Delta t + \Delta$ 执行32 [PROPOSE, $_p$ ^C_p Q_p^C $\mathcal{GJ}_p$ t v_p] ∧ $\mathcal{GJ}_p$ $\mathcal{V}_i$ ∧：如果有效33(ch 、ch 、 、 、 、) J(、)：$\mathcal{GJ}_p \geq \mathcal{GJ}$冻结$_i$ 则 34：$\mathcal{GJ}_i^{\text{frozen}} \leftarrow \mathcal{GJ}_p$ 35：如果 $\mathsf{ch}_i^{\text{frozen}} \not\succeq \mathcal{GJ}_p$.ch，则 36：$\mathsf{ch}_i^{\text{frozen}} \leftarrow \mathcal{GJ}_p$.ch 37：如果 $\mathsf{ch}_i^{\text{frozen}} \preceq$ch ^C_p ，则 38：$\mathsf{ch}_i^{\text{frozen}} \leftarrow \mathsf{ch}_p^C$

当前分叉选择的；以及 (iii) $\mathcal{GJ}$冻结$_i$ ·ch，丢弃任何未扩展分叉选择的选项。每个验证者还会在其 VOTE 消息中包含一个 FIN-VOTE，其源为 $\mathcal{GJ}$冻结$_i$，而目标则取决于其时隙：如果 $\mathcal{GJ}$冻结$_i$ 来自时隙 $t-1$，目标为 chAva$_i$；否则，目标为提升至时隙 t的 $\mathcal{GJ}_i^{\text{frozen}}$.ch。

3. **快速确认：** 如果快速确认($\mathcal{V}_i, t$)返回候选 $\mathsf{ch}^{\text{cand}}$，则 v_i 将 chAva$_i := \mathsf{ch}^{\text{cand}}$。

4. **合并：** 在第 $4\Delta t + 3\Delta$ 轮， v_i 将 $\mathcal{GJ}_i^{\text{frozen}}$更新为当前视图中的最大已论证检查点。

在 GST $= 0$ 之后，附录 C 表明，Majorum 中的 chAva 以压倒性概率动态可用（定理 2），且具备重组韧性（定理 1）。

从总体上看，在假设 GST $= 0$、 $f < n/3$ 和约束（1）成立的情况下，附录 C.1 表明，集成最终性不会改变算法 2 的核心行为。具体而言，我们证明：（i）除最终性组件外，诚实验证者在两种协议中发送相同的消息；（ii）它们的分叉选择输出一致；（iii）Majorum 中的每条 chAva 都是算法 2 的一条已确认链。因此，附录 A 中针对算法 2 证明的所有引理和定理也适用于 Majorum。

最后，在附录 C.2 中，我们证明，在时间 max(GST，GAT) $+ 4\Delta$ 之后，Majorum 可保证可问责安全性和活性。具体而言，它满足三项关键性质。第一，可问责安全性：诚实验证者绝不会被罚没（引理 15）；因此，若存在两条相互冲突的已最终确定链，则至少有 $\frac{n}{3}$ 个验证者违反了罚没条件（定理 5）。第二，单调性：任何诚实验证者的最终确定链 $\overline{\text{chFin}}_i$ 都会单调增长（引理16）。第三，活性：在 max(GST， GAT)$+4\Delta$ 之后，如果有两名诚实提议者在连续时隙中当选，那么第一名提议者的链会在 2–3 个时隙内最终确定（定理 6），具体取决于是否需要重新证明。我们在定理 7 中得出结论：Majorum 以压倒性概率满足安全涨落协议的性质。

通信复杂度 Majorum 的预期通信复杂度为 $\mathcal{O}(Ln^3)$，其中 L 为区块大小。这与底层动态可用协议 TOB-SVD 的界限相符，因为诚实验证者会在每一轮转发收到的所有消息，产生 $\mathcal{O}(Ln^3)$ 的开销。最终性工具不会引入额外开销，因为它嵌入在轮次 $4\Delta t + \Delta$的 VOTE 消息中。因此，总体复杂度与 TOB-SVD 相同。

6 结论

我们提出了 Majorum，这是一种安全的涨落式共识协议，每个时隙仅需一个投票阶段，即可实现安全性和效率。在 Momose 和 Ren [13] 及后续工作的基础上，我们将 TOB-SVD 改造为动态可用的基础协议，并将其简化为概率安全性，从而得到一种更简单、更实用且保留原有保证的协议。结合最终性协议，Majorum 可在乐观条件下于两到三个时隙内确定链的最终性，同时保留其底层动态可用协议的通信复杂度。

参考文献

[1]Aditya Asgaonkar、Francesco D'Amato、Roberto Saltini 和 Luca Zanolini。以太坊共识性质列表。网址：https://docs.google.com/document/d/1Q_iDOODIq-glLRPSMnSf3HAKDQGHXsENHzRG9iwok7g.[2] Vitalik Buterin 和 Virgil Griffith。友好的 Casper 最终性装置。*CoRR*，abs/1710.09437，2017。网址：http://arxiv.org/abs/1710.09437，arXiv:1710.09437。[3] Vitalik Buterin、Diego Hernandez、Thor Kamphefner、Khiem Pham、Zhi Qiao、Danny Ryan、Juhyeok Sin、Ying Wang 和 Yan X Zhang。结合 GHOST 与 Casper。2020。[4] Miguel Castro 和 Barbara Liskov。实用拜占庭容错。载于 Margo I. Seltzer 和 Paul J. Leach 编，第三届 *USENIX* 操作系统设计与实现研讨会论文集（*OSDI*），美国路易斯安那州新奥尔良，*1999* 年 *2* 月 *22—25* 日，第 173–186 页。USENIX 协会，1999。[5] Francesco D'Amato、Giuliano Losa 和 Luca Zanolini。异步容忍型休眠总序广播协议。载于 Ran Gelles、Dennis Olivetti 和 Petr Kuznetsov 编，第 *43* 届 *ACM* 分布式计算原理研讨会论文集（*PODC 2024*），法国南特，*2024* 年 *6* 月 *17—21* 日，第 247–256 页。ACM，2024。doi:10.1145/3662158.3662779。

[6] Francesco D'Amato、Joachim Neu、Ertem Nusret Tas 和 David Tse。Goldfish：以太坊不再遭受攻击？！载于 Jeremy Clark 和 Elaine Shi 编，金融密码学与数据安全——第 *28* 届国际会议，*FC 2024*，威廉斯塔德，库拉.索，*2024* 年 *3* 月 *4—8* 日，修订版精选论文，第一部分，计算机科学讲义第 14744 卷，第 3–23 页。Springer，2024。doi: 10.1007/978-3-031-78676-1_1。[7] Francesco D'Amato、Roberto Saltini、Thanh-Hai Tran 和 Luca Zanolini。TOB-SVD：休眠模型中采用单票决策的全序广播。载于第 *45* 届 *IEEE* 分布式计算系统国际会议（*ICDCS 2025*），英国格拉斯哥，*2025* 年 *7* 月 *21—23* 日，第 1033–1043 页。IEEE，2025。doi:10.1109/ICDCS63083.2025.00104。[8] Francesco D'Amato 和 Luca Zanolini。近期的最新消息驱动 GHOST：在动态可用性与异步性韧性之间取得平衡。载于第 *37* 届 *IEEE* 计算机安全基础研讨会（*CSF 2024*），荷兰恩斯赫德，*2024* 年 *7* 月 *8—12* 日，第 127–142 页。IEEE，2024。doi:10.1109/CSF61375. 2024.00001。[9] Eli Gafni 和 Giuliano Losa。简短公告：在行为良好的多数参与下实现动态参与的拜占庭共识。载于 Rotem Oshman 编，第 *37* 届分布式计算国际研讨会（*DISC 2023*），意大利拉奎拉，*2023* 年 *10* 月 *10—12* 日，LIPIcs 第 281 卷，第 41:1–41:7 页。Schloss Dagstuhl 信息学莱布尼茨中心，2023。网址：https://doi.org/10.4230/LIPIcs. DISC.2023.41，doi:10.4230/LIPICS.DISC.2023.41。[10]Andrew Lewis-Pye 和 Tim Roughgarden。无需许可的共识。*CoRR*，abs/2304.14701，2023。网址：https://doi.org/10.48550/arXiv.2304.14701，arXiv:2304.14701，doi:10.48550/ ARXIV.2304.14701。[11] Dahlia Malkhi、Atsuki Momose 和 Ling Ren。完全涨落参与下的拜占庭共识。2022。网址：https://eprint.iacr.org/archive/2022/1448/20221024:011919。[12] Dahlia Malkhi、Atsuki Momose 和 Ling Ren。迈向实用的休眠 BFT。载于 Weizhi Meng、Christian Damsgaard Jensen、Cas Cremers 和 Engin Kirda 编，*2023* 年 *ACM SIGSAC*计算机与通信安全会议论文集（*CCS 2023*），丹麦哥本哈根，*2023* 年 *11* 月 *26—30* 日，第 490–503 页。ACM，2023。doi:10.1145/3576915.3623073。[13] Atsuki Momose 和 Ling Ren。休眠共识中的恒定延迟。载于 Heng Yin、Angelos Stavrou、Cas Cremers 和 Elaine Shi 编，*2022* 年 *ACM SIGSAC*计算机与通信安全会议论文集（*CCS 2022*），美国加利福尼亚州洛杉矶，*2022* 年 *11* 月 *7—11* 日，第 2295–2308 页。ACM，2022。doi:10.1145/3548606.3559347。[14] 中本聪。比特币：一种点对点电子现金系统，2008。[15] Joachim Neu、Ertem Nusret Tas 和 David Tse。涨落式协议：解决可用性与最终性困境。载于第 *42* 届 *IEEE* 安全与隐私研讨会（*SP 2021*），美国加利福尼亚州旧金山，*2021* 年 *5* 月 *24—27* 日，第 446–465 页。IEEE，2021。doi:10.1109/SP40001.2021.00045。[16] Joachim Neu、Ertem Nusret Tas 和 David Tse。涨落式协议：解决可用性与最终性困境。载于第 *42* 届 *IEEE* 安全与隐私研讨会，2021。即将出版。网址：https: //arxiv.org/abs/2009.04987。[17] Margaux Nijkerk。以太坊曾短暂停止终结交易。发生了什么？https://www.coindesk.com/tech/2023/05/17/ethereums-loss-of-finality-what-happened，2023 年 5 月。[18] Rafael Pass 和 Elaine Shi。共识的休眠模型。载于*ASIACRYPT*（第 *2* 部分），计算机科学讲义第 10625 卷，第 380–409 页。Springer，2017。[19] Maofan Yin、Dahlia Malkhi、Michael K. Reiter、Guy Golan-Gueta 和 Ittai Abraham。HotStuff：兼具线性性和响应性的 BFT 共识。载于 Peter Robinson 和 Faith Ellen 编，*2019* 年 *ACM* 分布式计算原理研讨会论文集（*PODC 2019*），加拿大安大略省多伦多，*2019* 年 *7* 月 *29* 日至 *8* 月 *2* 日，第 347–356 页。ACM，2019。

[20] Vlad Zamfir。Casper 友好幽灵。一种通过构造确保正确的区块链共识协议。网址：
https://github.com/ethereum/research/blob/master/papers/cbc-consensus/AbstractCBC.pdf。

A 算法 2 分析

引理 1。 考虑一个由 κ 个独立时隙组成的序列，在每个时隙中，一名诚实验证者当选提议者的概率至少为 $p = \frac{h_0}{n}$，其中 h_0 是活跃诚实验证者的数量， n 是验证者总数。对于有限时间范围 T_{conf}（关于 κ为多项式），如果 p 不可忽略，那么以压倒性概率（即概率至少为 $1-\mathrm{negl}(\kappa)-\mathrm{negl}(\lambda)$），至少存在一对连续时隙，其中当选了诚实提议者。

证明。 令 X_i 表示第 i 个时隙有诚实提议者这一事件。根据假设，每个 X_i 都独立发生，且概率至少为 p。令 E 表示不存在两个连续时隙都有诚实提议者这一事件。我们通过将该序列划分为 $\lfloor \kappa/2 \rfloor$ 个互不相交的、每组包含两个连续时隙的组，对 $P(E)$ 给出保守的上界。在每一组中，两对相邻时隙（i， $i+1$）都不同时包含诚实提议者的概率至多为 $1-p^2$。

T因此，事件 E 发生的概率至多为 st:

$$P(E) \leq (1-p^2)^{\lfloor \kappa/2 \rfloor}.$$

如果 p 非可忽略，且将 κ 视为安全参数，那么该上界在 κ 中变为可忽略；即， $P(E) \leq \mathsf{negl}(\kappa)$。

由于密码学失效发生的概率为 $\mathsf{negl}(\lambda)$（第 2.2 节），因此未能观察到两个连续的诚实时隙提议者的概率至多为 $\mathsf{negl}(\kappa) + \mathsf{negl}(\lambda)$。

因此，以压倒性概率，至少存在一对连续时隙，在这两个时隙中均选出了诚实提议者。

□

对于任意时隙 t，我们定义：

- $\mathrm{MFC}_i^{\mathsf{propose}(t)} := \mathrm{MFC}(\mathcal{V}_i^{\mathsf{propose}(t)}, \mathcal{V}_i^{\mathsf{propose}(t)}, \mathsf{ch}^C, t)$， 其中 $(\mathsf{ch}^C, \cdot) = \mathtt{fastconfirmsimple}(\mathcal{V}_i^{\mathsf{propose}(t)}, t-1)$;
- $\mathrm{MFC}_i^{\mathsf{vote}(t)} := \mathrm{MFC}(\mathcal{V}_i^{\mathrm{frozen},\mathsf{vote}(t)}, \mathcal{V}_i^{\mathsf{vote}(t)}, \mathsf{ch}_i^{\mathrm{frozen},\mathsf{vote}(t)}, t)$.

引理 2. 如果在时隙 t 中， $H_{\mathsf{vote}(t)}$ 中的所有验证者都为扩展链 ch 的链发送 VOTE 条消息，那么对于任意验证者 $v_i \in H_{\mathsf{vote}(t+1)}$， $\mathrm{MFC}_i^{\mathsf{propose}(t+1)} \succeq \mathsf{ch}$ 和 $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}$，这意味着在时隙 $t+1$ 中， $H_{\mathsf{vote}(t+1)}$ 中的所有验证者都会为扩展 ch 的链发送 VOTE 条消息。

证明。 设 v_i 是 $H_{\mathsf{vote}(t+1)}$ 中任意一个诚实验证者。根据加入协议，这意味着 v_i 在轮次 $\mathsf{merge}(t) = 4\Delta(t+1) - \Delta$ 处于唤醒状态。显然，在时间 $\mathsf{propose}(t+1)$ 也同样如此：$\mathsf{ch}_i^{\mathrm{frozen\ propose}(t+1)} = B_{\mathrm{genesis}} \vee \mathsf{ch}_i^{\mathrm{frozen\ propose}(t+1)} \succeq \mathsf{ch}$。现在我们分别考虑每种情况，并证明对于每种情况都有 $\mathrm{MFC}_i^{\mathsf{propose}(t+1)} \succeq \mathsf{ch}$。

情况 1： $\mathsf{ch}_i^{\mathrm{frozen\ propose}(t+1)} = B_{\mathrm{genesis}}$。 注意，由于引理和同步性假设， $H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}$ 中的验证者发送的所有 VOTE 消息都包含在 $\mathcal{V}_i^{\mathsf{propose}(t+1)}$ 中。因此，在时间 $\mathsf{propose}(t+1)$， $\left|\mathcal{V}^{\mathsf{ch}}{}_i^{t+1}\right| \geq \left|H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}\right|$。类似地，$H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)} \subseteq \mathsf{S}(\mathcal{V}_i, t+1)$。此外，$\mathsf{S}(\mathcal{V}_i, t+1) \setminus \left(H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}\right) \subseteq A_{\mathsf{vote}(t+1)} \cup \left(H_{\mathsf{vote}(t-1+1),\mathsf{vote}(t-1)} \setminus H_{\mathsf{vote}(t)}\right)$ 因为 $\mathsf{S}(\mathcal{V}_i, t+1)$ 中任何不来自 $H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}$ 中验证者的 VOTE，要么来自在时间 $\mathsf{vote}(t+1)$处于拜占庭状态的验证者，要么来自在 $\mathsf{vote}(t-1+1)$ 与 $\mathsf{vote}(t-1)$ 之间的某个时刻处于活跃状态、但在时间 $\mathsf{vote}(t)$并未处于活跃状态的验证者。于是，我们有

$$\begin{aligned} |\mathsf{S}(\mathcal{V}_i, t+1)| &= \left|H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}\right| + \left|\mathsf{S}(\mathcal{V}_i, t+1) \setminus \left(H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}\right)\right| \\ &\leq \left|H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}\right| + \left|A_{\mathsf{vote}(t+1)} \cup \left(H_{\mathsf{vote}(t-1+1),\mathsf{vote}(t-1)} \setminus H_{\mathsf{vote}(t)}\right)\right| \end{aligned}$$

$$< 2\left|H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}\right|$$

其中，最后一个不等式来自方程（1）。

Hence, at time $\mathsf{propose}(t+1)$, $\left|\mathcal{V}_i^{\mathsf{ch},t+1}\right| \geq \left|H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}\right| > \frac{|\mathsf{S}(\mathcal{V}_i,t+1)|}{2}$.

因此， $\mathrm{MFC}_i^{\mathsf{propose}(t+1)} \succeq \mathsf{ch}$。

情况 2： $\mathsf{ch}_i^{\text{frozen propose}(t+1)} \succeq \mathsf{ch}$。鉴于根据定义， $\mathrm{MFC}_i^{\mathsf{propose}(t+1)} \succeq \mathsf{ch}_i^{\text{frozen propose}(t+1)}$，，我们有 $\mathrm{MFC}_i^{\mathsf{propose}(t+1)} \succeq \mathsf{ch}$。

现在我们来证明 $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}$。注意，如果一个 $[\text{PROPOSE}, B, \mathsf{ch}_p^C, Q_p^C, t+1, v_p]$消息是有效的，那么 $\mathsf{ch}_p^C = B_{\text{genesis}} \vee \mathsf{ch}_p^C \succeq \mathsf{ch}$。第 36 行意味着 $\mathsf{ch}_i^{\text{frozen vote}(t+1)} =, \mathsf{ch}_i^{\text{frozen merge}(t)} \vee, \mathsf{ch}_i^{\text{frozen vote}(t+1)} =, \mathsf{ch}_p^C$。然后，根据 $\mathsf{ch}_i^{\text{frozen merge}(t)} = B_{\text{genesis}} \vee, \mathsf{ch}_i^{\text{frozen merge}(t)} \succeq, \mathsf{ch}$ 和第 36 行可知，在时间 $\mathsf{vote}(t+1)$，$\mathsf{ch}_i^{\text{frozen vote}(t+1)} = B_{\text{genesis}} \vee, \mathsf{ch}_i^{\text{frozen vote}(t+1)} \succeq, \mathsf{ch}$。现在我们分别考虑每种情况。

情况 1： $\mathsf{ch}_i^{\text{frozen vote}(t+1)} = B_{\text{genesis}},$。注意，由于引理和同步性假设，验证者在 $H_{\mathsf{vote}(t)} \setminus A_{\mathsf{vote}(t+1)}$ 中发送的所有 VOTE消息都包含在 $\mathcal{V}^{\text{frozen}}{}_{,i}^{\mathsf{vote}(t+1)} \cap \mathcal{V}_i^{\mathsf{vote}(t+1)}$ 中。从这里开始，证明实际上与 $\mathrm{MFC}_i^{\mathsf{propose}(t+1)}$ 的情况 1 相同，只需将 $\mathcal{V}_i^{\mathsf{propose}(t+1)}$ 替换为 $\mathcal{V}^{\text{frozen}}{}_{,i}^{\mathsf{vote}(t+1)} \cap \mathcal{V}_i^{\mathsf{vote}(t+1)}$。

情况 2： $\mathsf{ch}_i^{\text{frozen vote}(t+1)} \succeq \mathsf{ch}$。与 $\mathrm{MFC}_i^{\mathsf{propose}(t+1)}$ 的情况 2 的证明类似，鉴于根据定义，$\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}_i^{\text{frozen vote}(t+1)}$，，我们有 $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch.l}$

由 $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}$ 可知，在时隙 $t+1$， v_i 会发送一条针对扩展 ch 的链的 VOTE 消息。 □

引理 3. 如果一位诚实验证者在时隙 t 快速确认了一条链 ch （即，存在 Q 使得$(\mathsf{ch}\ Q) = \mathit{fastconfirmsimple}(\mathcal{V}_i^{\mathsf{fconf}(t)}, t) \wedge Q \neq \emptyset$），那么，对于任意时隙 $t' > t$ 和验证者 $v_i \in H_{\mathsf{vote}(t')}$，$\mathrm{MFC}^{\mathsf{propose}(t')}{}_i \succeq \mathsf{ch}$ 且 $\mathrm{MFC}^{\mathsf{vote}(t')}{}_i \succeq \mathsf{ch}$，这意味着 $H_{\mathsf{vote}(t')}$ 中的所有验证者都会发送一条针对扩展 ch 的链的 VOTE 消息。

证明。 通过对 t' 进行归纳法证明。

基础情形： $t' = t+1$. 假设在轮次 $\mathsf{fconf}(t)$， $H_{\mathsf{vote}(t)}$ 中的一位诚实验证者快速确认了一条链 ch。鉴于我们假设 $f < \frac{n}{3}$，同一时隙内无法形成相互冲突的仲裁证书。于是，由于加入协议和同步性假设，在时间 $\mathsf{merge}(t)$ 时， $= 4\Delta(t+1) - \Delta$，对于任意验证者 $v_i \in H_{\mathsf{vote}(t+1)}$， $\mathsf{ch}_{\text{frozen, 合并}(t)\,i} = \mathsf{ch}$。

同样，如果 $[\text{PROPOSE}, \mathsf{ch}_p, \mathsf{ch}_p^C, Q_p^C, t+1, v_p]$ 消息有效，那么 $\mathsf{ch}_p^C = B_{\text{genesis}} \vee \mathsf{ch}_p^C = \mathsf{ch}$。这一点与第 35 至 36 行意味着，在时间 $\mathsf{vote}(t+1)$，$\mathsf{ch}_i^{\text{frozen vote}(t+1)} =, \mathsf{ch}$。

因此，由于 $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}_i^{\text{frozen vote}(t+1)}$，， $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}$，这意味着 $H_{\mathsf{vote}(t+1)}$ 中的所有验证者都会发送扩展链 ch 的 VOTE 消息。

In归纳步骤： $t' > t+1$. 这里我们只需应用引理 2 即可得出证明。 □

□

引理 4。 设 t 是一个时隙，其提议者 v_p 诚实，并假设 v_p 发送了一条 $[\text{PROPOSE}, \mathsf{ch}_p, \mathsf{ch}_p^C, Q_p^C, t, v_p]$消息。那么，对于任意时隙 $t' \geq t$，$H_{\mathsf{vote}(t')}$ 中的所有验证者都会发送一条针对扩展 ch_p 的链的 VOTE 消息。此外，对于任意时隙 $t'' > t$ 和任意验证者$v_i \in H_{\mathsf{vote}(t'')}$， $\mathrm{MFC}^{\mathsf{propose}(t'')}{}_i \succeq \mathsf{ch}_p$，且 $\mathrm{MFC}^{\mathsf{vote}(t'')}{}_i \succeq \mathsf{ch}_p$。

证明。 通过对 t' 进行归纳法来证明。

基础情形： $t' = t$。假设在时隙 t 中，诚实提议者 v_p 发送了一条 $[\text{PROPOSE}, \mathsf{ch}_p, \mathsf{ch}_p^C, Q_p^C t, v_p]$ 消息。考虑一名诚实验证者 $v_i \in H_{\mathsf{vote}(t)}$。注意，由于同步性假设，$\mathcal{V}_i^{\mathsf{merge}(t-1)} \subseteq \mathcal{V}_p^{\mathsf{propose}(t)}$。鉴于我们假设 $f < \frac{n}{3}$，这进一步意味着 $\mathsf{ch}_i^{\text{frozen merge}(t-1)} \neq B_{\text{genesis}} \Longrightarrow \mathsf{ch}_p^C = \mathsf{ch}_i^{\text{frozen merge}(t-1)}$。因此，显然 $\mathsf{ch}_p^C \succeq \mathsf{ch}_i^{\text{frozen merge}(t-1)}$。所以，根据代码行 35 至 36，$\mathsf{ch}_i^{\text{frozen vote}(t)} = \mathsf{ch}_p^C$。

我们知道，要么 $\left|\left(\mathcal{V}_i^{\text{frozen}}\right)^{\text{MFC}_i^{\mathsf{vote}(t)}} \cap \mathcal{V}^{\text{MFC}_i^{\mathsf{vote}(t)}}\right| > \frac{|\mathsf{S}(\mathcal{V}_i, t)|}{2}$，要么 $\text{MFC}_i^{\mathsf{vote}(t)} = \mathsf{ch}_i^{\text{frozen}}$。下面分别讨论这两种情况。

情况 1： $\left|\left(\mathcal{V}^{\text{frozen}}\right)^{\mathbf{MFC}_i^{\mathsf{vote}(t)}} \cap \mathcal{V}^{\mathbf{MFC}_i^{\mathsf{vote}(t)}}\right| > \frac{|\mathsf{S}(\mathcal{V}_i, t)|}{2}$。根据分级交付性质，[7]，这意味着在时间 propose(t)，$\left|\mathcal{V}_p^{\text{MFC}_i^{\mathsf{vote}(t)}}\right| > \frac{|\mathsf{S}(\mathcal{V}_p, t)|}{2}$；也就是说，根据代码行 18，$\mathsf{ch}_p \succeq \text{MFC}_i^{\mathsf{vote}(t)}$，因此，根据代码行 24 和 25，在时隙 t，v_i 会为 ch_p 发送一条 VOTE 消息。**情况 2：** $\mathbf{MFC}_i^{\mathsf{vote}(t)} = \mathsf{ch}_i^{\mathbf{frozen}}$。根据代码行 24 和 25，$v_i$ 仍会为 ch_p 发送一条 VOTE，因为 $\mathsf{ch}_p \succeq \mathsf{ch}_p^C = \mathsf{ch}_i^{\text{frozen}} = \text{MFC}_i^{\mathsf{vote}(t)}$。

归纳步骤： $t' > t$。这里我们只需应用引理 2 即可完成证明。 □

□

引理 5。 设 r_i 为任意轮次，r_j 为满足以下条件的任意轮次：$r_j \geq r_i$ 且 $r_j \in \{\text{提议}(\text{slot}(r_j))，\text{投票}(\text{slot}(r_j))\}$。那么，对于在轮次 r_i 中诚实的任意验证者 v_i 以及任意验证者 $v_j \in H_{r_j}$，$\mathsf{Ch}_i^{r_i} \preceq \text{MFC}_j^{r_j}$。

证明。我们通过反证法进行证明。设 r_i 是满足以下条件的最小轮次：存在两个诚实验证者 v_i 和 v_j，以及轮次 r_j，使得 $r_j \geq r_i$ 且 $r_j \in \{\text{在提议轮次或投票轮次 } r_j \geq r_i \text{提议}(\text{slot}(r_j))，\text{投票}(\text{slot}(r_j))\}$ 且 $\mathsf{Ch}_i^{r_i} \not\preceq \text{MFC}_j^{r_j}$；也就是说，$r_i$ 是诚实验证者确认的链与另一位诚实验证者在提议轮次或投票轮次 $r_j \geq r_i$ 的输出 MFC 冲突的第一轮次。由 r_i 的最小性可知，$\mathsf{Ch}_i^{r_i - 1} \neq \mathsf{Ch}_i^{r_i}$，进而推出 $v_i \in H_{r_i}$。这只有在 r_i 是投票轮次或快速确认轮次时才可能发生。令 $t_i = \text{slot}(r_i)$，并分别分析每种情况。

情况 1： r_i **是一个投票轮次。** 由于第 22 行，$\mathsf{Ch}_i^{r_i} \succeq \left(\text{MFC}^{\mathsf{vote}(t_i)}\right)^{\lceil \kappa}$。现在我们来考虑两种子情况。

情况 1.1： $\mathsf{Ch}_i^{r_i} = \left(\mathbf{MFC}_i^{\mathsf{vote}(t_i)}\right)^{\lceil \kappa})$。以压倒性概率（引理 1），区间 $[t_i - \kappa, t_i)$ 中至少存在一个时隙 t_p，其提议者 v_p 是诚实的。令 ch_p 为时隙 t_p 中由 v_p 生成的链 PROPOSEd。鉴于 $t_p < t_i$，引理 4 蕴含 $\text{MFC}^{\mathsf{vote}(t_i)} \succeq \mathsf{ch}_p$。然后，因为 $t_p \geq t_i - \kappa$，我们有 $\mathsf{ch}_p \succeq \left(\text{MFC}^{\mathsf{vote}(t_i)}\right)^{\lceil \kappa} = \mathsf{Ch}_i^{r_i}$。因为 $t_p < \text{slot}(r_j)$，引理 4 还蕴含 $\mathsf{ch}_j^{r_j} \succeq \mathsf{ch}_p \succeq \mathsf{Ch}_i^{r_i}$，从而导致矛盾。

情况 1.2： $\mathsf{Ch}_i^{r_i} \succ \left(\mathbf{MFC}_i^{\mathsf{vote}(t_i)}\right)^{\lceil \kappa})$。此情况意味着 $\mathsf{Ch}_i^{r_i} = \mathsf{Ch}_i^{r_i - 1}$。根据 r_i 的最小性，我们得出矛盾。

情况 2： r_i **是一个快速确认轮次。** 注意，这意味着 $t_i <$ 时隙(r_j)。由于 r_i 的最小性，我们只需考虑 $(\mathsf{Ch}_i^{r_i}, Q) =$ 快速确认简单算法$(\mathcal{V}_i^{r_i}, t_i) \wedge Q \neq \emptyset$ 的情况。因此，我们可以应用引理 3 得出 $\mathsf{ch}_j^{r_j} \succeq \mathsf{Ch}_i^{r_i}$，从而得到矛盾。 □

□

定理 1（重组韧性）。*算法 2 具备重组韧性。*

证明。取一个时隙 t_p，其诚实提议者 v_p 为链 ch_p 发送了一条 PROPOSE 消息。再取任意轮次 r_i 以及在轮次 r_i 中诚实的验证者 v_i。现在令 t_j 为满足 $t_j > \max(t_p, \text{slot}(r_i))$ 的任意时隙。选取任意验证者 $v_j \in H_{\mathsf{vote}(t_j)}$。由引理 4 可知 $\text{MFC}^{\mathsf{vote}(t_j)} \succeq \mathsf{ch}_p$。由引理 5 可知 $\text{MFC}^{\mathsf{vote}(t_j)} \succeq \mathsf{Ch}_i^{r_i}$。因此，$\mathsf{ch}_p$ 与 $\mathsf{Ch}_i^{r_i}$ 不冲突。 □

引理 6. 对于任意时隙 t 和验证者 $v_i \in H_{\mathsf{propose}(t)}$，$\mathrm{MFC}_i^{\mathsf{propose}(t)}.p < t$。

证明。 由于分级一致性的有效性属性 [7] 以及 $B_{\mathrm{genesis}}.p < 0$这一事实，只需证明：在轮次 propose($t$) 中诚实的验证者从未为满足 $\mathsf{ch}.p \geq t$ 的链 ch 发送过 VOTE 消息。证明采用归纳法。

基础情形： $t = 0$。 显然，因为 $H_{\mathsf{propose}(t)}$ 中没有验证者曾发送过任何消息。

归纳步骤： $t > 0$。 根据归纳假设，以及诚实节点在轮次 propose($t-1$) 中不会发送任何VOTE 消息这一事实，由分级一致性的有效性属性可知，$\mathrm{MFC}^{\mathsf{vote}(t-1)}.p < t-1$。然后，证明可由第 24 行得出。

□

□

定理 2。 算法 *2* 以压倒性概率动态可用。

证明。我们首先证明该协议在确认时间为 $T_{\mathrm{conf}} = 8\kappa\Delta + \Delta$ 时具有活性。我们先仅考虑深度为 κ 的规则，以此证明活性。取时隙 $t = \mathsf{slot}(r)$ 中的轮次 r 、另一轮次$r_i \geq r + 8\kappa\Delta + \Delta \geq 4\Delta(t + 2\kappa) + \Delta = \mathsf{vote}(t + 2\kappa)$，以及一名诚实验证者 $v_i \in H_{r_i}$。令 $t_i = \mathsf{slot}(r_i)$。根据加入协议可知，对于 v_i，在 $\mathsf{vote}(t + 2\kappa)$ 轮次当轮或之后的第一个活跃轮次是投票轮次。以很高的概率，可以找到一个由诚实提议者主持的时隙 $t_p \in [t+1, t+\kappa]$ （引理 1）。令 ch_p为诚实提议者 v_p 在时隙 t_p中 PROPOSEd 的链。根据引理 6 和性质 1 可知，ch_p包含 txpool propose(t_p) 中的所有交易。鉴于 $\mathsf{propose}(t_p) \geq r$，$\mathsf{txpool}^r \subseteq \mathsf{txpool\ propose}(t_p)$，这意味着 ch_p 包含 txpool^r中的所有交易。鉴于 $\mathsf{slot}(r_i) > t_p$，根据引理 4，$\mathrm{MFC}^{\mathsf{vote}(t_i)}{}_i \succeq \mathsf{ch}_p$。注意，第 22 行意味着 $\mathsf{Ch}_i^{r_i} \succeq \left(\mathrm{MFC}^{\mathsf{vote}(t_i)}{}_i\right)^{\lceil \kappa, t_i}$。然后，由于$t_p \leq t + \kappa \leq t_i - \kappa$，$\mathsf{ch}_p \preceq \left(\mathrm{MFC}^{\mathsf{vote}(t_i)}{}_i\right)^{\lceil \kappa, t_i}$，因此 $\mathsf{ch}_p \preceq \mathsf{Ch}_i^{r_i}$，这意味着 $\mathsf{Ch}_i^{r_i}$ 包含 txpool^r 中的任何交易。

现在我们要证明快速确认不会造成干扰。已知 $t_i \geq t_j$，根据引理 4，我们知道诚实验证者在时隙 t_i 发送的任何 VOTE 都是针对扩展 ch_p 的链；鉴于我们假设$f < \frac{n}{3}$，如果 r_i 是快速确认轮次，且 v_i 设置了 $\mathsf{Ch}_i^{r_i}$，那么 $\mathsf{Ch}_i^{r_i} \succeq \mathsf{ch}_p$ 仍然成立。

–

现在我们证明安全性。任取两个轮次 r_i 和 r_j，以及分别在轮次 r_i 和 r_j中诚实的验证者 v_i 和 v_j。现在令 t_k 为任意一个时隙，使得 $t_k > \max(\mathsf{slot}(r_j), \mathsf{slot}(r_i))$。根据约束（1），我们知道$H_{\mathsf{vote}(t_k)}$ 非空。因此，任取一个验证者 $v_k \in H_{\mathsf{vote}(t_k)}$。引理 5 表明，$\mathrm{MFC}^{\mathsf{vote}(t_k)}{}_k \succeq \mathsf{Ch}_i^{r_i}$且$\mathrm{MFC}^{\mathsf{vote}(t_k)}{}_k \succeq \mathsf{Ch}_j^{r_j}$。因此，$\mathsf{Ch}_i^{r_i}$与 $\mathsf{Ch}_j^{r_j}$ 不冲突。 □

引理 7 (快速确认的活性)。 取一个满足 $|H_{\mathsf{vote}(t)}| \geq \frac{2}{3}n$ 的时隙 t 。如果在时隙 t，一名诚实验证者发送了一条针对链 ch_p 的 PROPOSE 消息，那么对于任意验证者 $v_i \in H_{\mathsf{fconf}(t)}$，$\mathsf{Ch}_i^{\mathsf{fconf}(t)} \succeq \mathsf{ch}_p$。

证明。 根据假设，我们有 $|H_{\mathsf{vote}(t)}| \geq \frac{2}{3}n$，由引理 4 可知，$H_{\mathsf{vote}(t)}$ 中的每个诚实验证者都会为 ch_p发送一条 VOTE 消息。因此，任何验证者 $v_i \in H_{\mathsf{fconf}(t)}$ 都会收到针对 ch_p的法定人数消息。于是，根据第 28 行和第 29 行，$\mathsf{Ch}_i^{\mathsf{fconf}(t)} \succeq \mathsf{ch}_p$。 □

B 最终性组件分析

引理 8。 假设 $f \in [0, n]$，并令 $\mathcal{C}_\mathsf{f} = (\mathsf{ch_f}, c_\mathsf{f})$ 为一个已最终确定的检查点，$\mathcal{C}_\mathsf{j} = (\mathsf{ch_j}, c_\mathsf{j})$ 为一个已证明合理的检查点，且 $c_\mathsf{j} \geq c_\mathsf{f}$ 分别对应任意两个视图。要么 $\mathsf{ch_j} \succeq \mathsf{ch_f}$，要么可以检测到至少 $\frac{n}{3}$ 个验证者违反了 $\mathbf{E_1}$ 或 $\mathbf{E_2}$。

证明。 首先，我们证明任何检查点 $\mathcal{C}' = (\mathsf{ch}', c_\mathsf{f})$ 且 $\mathsf{ch}' \neq \mathsf{ch_f}$ 都不可能被正当化。否则，显然有 $\geq \frac{n}{3}$验证者因违反 $\mathbf{E_1}$ 而可被罚没：要使（$\mathsf{ch_f}$，c_f）获得正当化，必须有 FIN-VOTE个投票者构成超多数，且目标检查点的链为 $\mathsf{ch_f}$。同样，要使（ch'，c_f）获得正当化，必须有 FIN-VOTE个投票者构成超多数，且目标检查点的链为 ch'。鉴于正当化需要一组超多数链接，参与证明

$\mathcal{C}_\mathsf{f}$ 和 $\mathcal{C}'$ 的投票者集合的交集至少包括 $\frac{n}{3}$ 个验证者；这些验证者发送了两个 FIN-VOTES$\mathcal{S}_\mathsf{f} \to \mathcal{T}_\mathsf{f}$ 和 $\mathcal{S}' \to \mathcal{T}'$，其中 $\mathcal{T}_\mathsf{f}.c = \mathcal{T}'.c = c_\mathsf{f}$ 和 $\mathcal{T}_\mathsf{f}.\mathsf{ch} \neq \mathcal{T}'.\mathsf{ch}$，从而违反了条件 $\mathbf{E_1}$。

现在，反设 $\mathsf{ch_j} \not\succeq \mathsf{ch_f}$，并且不存在一组至少由 $\frac{n}{3}$ 个验证者组成的集合，可检测出他们违反了 $\mathbf{E_1}$ 或 $\mathbf{E_2}$。令 $c'_\mathsf{j} > c_\mathsf{f}$ 为满足以下条件的最小时隙：检查点 $\mathcal{C}'_\mathsf{j} = (\mathsf{ch}'_\mathsf{j}, c'_\mathsf{j})$ 已获正当化，且 $\mathsf{ch}'_\mathsf{j} \not\succeq \mathsf{ch_f}$，即，$\mathsf{ch}'_\mathsf{j}$ 与 $\mathsf{ch_f}$ 矛盾，或 ch'_j 是 $\mathsf{ch_f}$ 的严格前缀。根据我们的假设，我们知道存在这样一个检查点。

令 $(A_i, c_i) \to (\mathsf{ch}'_\mathsf{j}, c'_\mathsf{j})$ 和 $(\mathsf{ch_f}, c_\mathsf{f}) \to (C, c_\mathsf{f}+1)$ 分别为参与证明 $(\mathsf{ch}'_\mathsf{j}, c'_\mathsf{j})$ 和最终确定 $\mathsf{ch_f}$ 的 FIN-VOTES，它们由验证者 v_i 发出。我们观察到两种情况：

情况 1： $c'_\mathsf{j} = c_\mathsf{f} + 1$。如果 ch'_j 与 $\mathsf{ch_f}$ 矛盾，那么 $A_i \preceq \mathsf{ch}'_\mathsf{j}$ 意味着 $A_i \neq \mathsf{ch_f}$，因此这两张投票不同。反之，如果 ch'_j 是 $\mathsf{ch_f}$ 的严格前缀，那么 $A_i \preceq \mathsf{ch}'_\mathsf{j} \prec \mathsf{ch_f}$，因此 $A_i \neq \mathsf{ch_f}$，这也意味着这两张投票不同。因此，在这两种情况下，验证者 v_i 都违反了条件$\mathbf{E_1}$，所以可被罚没。

情况 2： $c'_\mathsf{j} > c_\mathsf{f} + 1$。我们需要考虑三种情况：

> **情况 2.1：** $c_i > c_\mathsf{f}$。根据证明的定义，如果检查点 $(\mathsf{ch}'_\mathsf{j}, c'_\mathsf{j})$ 已获正当化，那么检查点 (A_i, c_i) 也必须已获正当化。然而，鉴于 $c_i < c'_\mathsf{j}$，在假设 c'_j 具有最小性的前提下，我们有 $\mathsf{ch_f} \preceq A_i$。鉴于 $A_i \preceq \mathsf{ch}'_\mathsf{j}$，这会导致矛盾 $\mathsf{ch_f} \preceq \mathsf{ch}'_\mathsf{j}$。
>
> **情况 2.2：** $c_i = c_\mathsf{f}$。如上文情况的证明所述，检查点 (A_i, c_i) 已获正当化。正如本证明开头所证，要么 $A_i = \mathsf{ch_f}$，要么至少有 $\frac{n}{3}$ 个验证者违反了条件 $\mathbf{E_1}$。因此 $\mathsf{ch_f} = A_i \preceq \mathsf{ch}'_\mathsf{j}$，这导致矛盾。
>
> **情况 2.3：** $c_i < c_\mathsf{f}$。考虑到 $c_i < c_\mathsf{f} < c_\mathsf{f} + 1 < c'_\mathsf{j}$，由于存在环绕投票，这种情况违反了 $\mathbf{E_2}$。

鉴于证明需要超多数链接，参与最终确定 $\mathcal{C}_\mathsf{f}$ 和证明 $\mathcal{C}'_\mathsf{j}$ 的投票者集合的交集至少包含 $\frac{n}{3}$ 名验证者。根据上述推理，这些验证者违反了 $\mathbf{E_1}$ 或 $\mathbf{E_2}$。这与我们的假设相矛盾，因此证明得证。

□

引理 9. 假设 $f \in [0, n]$。如果根据任意两个相应的视图最终确定了两条冲突链，则可以检测出至少 $\frac{n}{3}$ 名验证者违反了 $\mathbf{E_1}$ 或 $\mathbf{E_2}$。

证明。 设 ch_1 和 ch_2 分别是根据视图 $\mathcal{V}_1$ 和 $\mathcal{V}_2$ 确定的两条冲突链。根据最终确定链的定义，这意味着存在两个检查点 $\mathcal{C}_1 = (\mathsf{ch}'_1, c_1)$ 和 $\mathcal{C}_2 = (\mathsf{ch}'_2, c_2)$，它们分别根据视图 $\mathcal{V}_1$ 和 $\mathcal{V}_2$ 最终确定，且 $\mathsf{ch}'_1 \succeq \mathsf{ch}_1$ 与 $\mathsf{ch}'_2 \succeq \mathsf{ch}_2$冲突。不失一般性，假设 $c_2 \geq c_1$。鉴于最终确定意味着已获得证明，我们可以应用引理 8 得出结论：如果 $\mathsf{ch}'_2 \not\succeq \mathsf{ch}'_1$，则可以检测出至少 $\frac{n}{3}$ 名验证者违反了 $\mathbf{E_1}$ 或 $\mathbf{E_2}$。 □

C Majorum 分析

C.1 同步性

在本部分分析中，我们始终假设 $\mathsf{GST} = 0$，整个验证者集合中受对手控制的成员始终少于三分之一（即，$f < \frac{n}{3}$），并且约束（1）成立。

我们首先形式化验证者发送的消息彼此相同、仅最终性组件不同这一概念。此外，我们定义算法 2 中的一次执行与算法 5 中的一次执行之间的等价关系。这些定义将在后续的引理和定理中使用。

定义 8. 当且仅当两条消息至多在其最终性相关组件上有所不同，我们称它们为动态等价。即，它们要么是 PROPOSE 消息，且至多在最大已论证检查点组件上有所不同；要么是 VOTE 消息，且至多在 FIN-VOTE 组件上有所不同。

当且仅当两组消息在类型等价下具有相同的等价类集合时，我们称它们动态等价；也就是说，对于任一集合中的任意消息 m，另一个集合中都存在一条消息 m'，使得 m 和 m' 动态等价。

给定两次执行 e 和 e' 以及第 r 轮，当且仅当对于任意轮次 $r_j < r$，在两次执行中均于第 r_j 轮诚实的验证者 v_j，执行 e 中 v_j 在第 r_j 轮发送的消息集合与执行 e' 中 v_j 在第 r_j 轮发送的消息集合动态等价时，我们称这两次执行在第 r 轮之前（含该轮）满足诚实输出动态等价。

$_e$在本节余下部分中，我们使用记号 $\mathcal{X}$，其中 $\mathcal{X}$ 是任意变量或定义，以明确表示执行 e中 $\mathcal{X}$ 的值。由于这两种算法采用的快速确认函数不同，我们确实需要明确说明 $\overset{e}{\mathrm{MFC}}_i^{\mathsf{propose}(t)}$ 的含义，其中 t 是任意时隙。若 e 是算法 2 的一次执行，则 $\overset{e}{\mathrm{MFC}}_i^{\mathsf{propose}(t)}$ 对应于 A 节中给出的定义。若e 是算法 5 的一次执行，则 $\overset{e}{\mathrm{MFC}}_i^{\mathsf{propose}(t)} := \overset{e}{\mathrm{MFC}}(\mathcal{V}_i^{\mathsf{propose}(t)}, \mathcal{V}_i^{\mathsf{propose}(t)}, \mathsf{ch}^C, t)$ 与 $(\mathsf{ch}^C, \cdot)$ =快速确认$(\mathcal{V}_i^{\mathsf{propose}(t)}, t-1)$。

定义9。 设 e_{FIN} 是算法 *5* 的一次合规执行， e_{FIN} 是算法 *2* 的一次合规执行。当且仅当以下约束成立时，我们称 e_{FIN} 和 e_{FIN} 是 MFC-等价：

1. e_{FIN} 和 e_{FIN} 在任何轮次之前都满足诚实输出动态等价。*2.* 对于任意时隙 t_i，*2.1* $\overset{e_{\mathrm{FIN}}}{\mathrm{MFC}}{}^{\mathsf{propose}(t_i)}_i = \overset{e_{\mathrm{FIN}}}{\mathrm{MFC}}{}^{\mathsf{propose}(t_i)}_j$ *2.2* $\overset{e_{\mathrm{FIN}}}{\mathrm{MFC}}{}^{\mathsf{vote}(t_i)}_i = \overset{e_{\mathrm{FIN}}}{\mathrm{MFC}}{}^{\mathsf{vote}(t_i)}_j$ *2.3* 对于任意时隙 $t_j \le t_i$ 和验证者 $v_j \in H_{\mathsf{vote}(t_j)}$，存在一个时隙 $t_k \le t_j$ 和一个验证者$v_k \in H_{\mathsf{vote}(t_k)}$，使得 $Ch^{\mathsf{vote}(t_k)}_k \succeq chAva^{\mathsf{vote}(t_j)}_j$。*2.4* 对于任意时隙 $t_j \le t_i$ 和验证者 $v_j \in H_{\mathsf{vote}(t_j)}$，存在一个时隙 $t_k \le t_j$ 和一个验证者 $v_k \in H_{\mathsf{vote}(t_k)}$，使得 $Ch^{\mathsf{vote}(t_k)}_k \succeq chAva^{\mathsf{fconf}(t_j)}_j$。其中，链 $chAva_j$ 属于 e_{FIN}，链 Ch_i 属于 e_{FIN}。

引理 10. 设 e_{FIN} 是算法 *5* 的任意一次合规执行。存在算法 *2* 的一次合规执行 e_{FIN}，使得 e_{FIN} 和 e_{FIN} 动态等价。

证明。 设 e_{FIN} 是算法 5 的任意一次合规执行，并设 $A^{e_{\mathrm{FIN}}}$ 是该执行中的对手。下文将说明对手 $A^{e_{\mathrm{FIN}}}$ 在算法 2 的一次执行 e_{FIN} 中所作的决策。稍后我们将证明，这会使 e_{FIN} 满足引理中的所有条件。

(i) 对于任意轮次 r， $A^{e_{\mathrm{FIN}}}$ 在轮次 r 中腐化的验证者集合，对应于 $A^{e_{\mathrm{FIN}}}$ 在同一轮次 r中腐化的验证者集合。(ii) 对于任意轮次 r， $A^{e_{\mathrm{FIN}}}$ 在轮次 r 中使其休眠的验证者集合，对应于 $A^{e_{\mathrm{FIN}}}$ 在同一轮次 r中使其休眠的验证者集合。(iii) 对于任意时隙 t，如果一名诚实验证者是 e_{FIN}中时隙 t 的提议者，那么它也是 e_{FIN} 中时隙 t 的提议者。(iv) 对于任意轮次 r，如果 e_{FIN} 和 e_{FIN} 在轮次 r之前诚实输出动态等价，那么 $A^{e_{\mathrm{FIN}}}$ 会安排向在轮次 r 中诚实的验证者发送消息，使得对于在轮次 r_i中诚实的任意验证者 v_i， v_i 在执行 e_{FIN} 中收到的消息集合与 v_i 在执行 e_{FIN}中收到的消息集合动态等价。

现在，我们通过对 t_i 进行归纳法，证明由 $A^{e_{\mathrm{FIN}}}$ 导出的执行 e_{FIN} 满足定义9的所有条件。为此，我们在归纳假设中加入以下条件。

2.5 For any $\mathcal{J}$ such that $\mathsf{J}(\mathcal{J}, \overset{e_{\mathrm{FIN}}}{\mathcal{V}}{}^{\mathsf{propose}(t_i)}_i), \overset{e_{\mathrm{FIN}}}{\mathrm{MFC}}{}^{\mathsf{propose}(t_i)}_i \succeq \mathcal{J}.\mathsf{ch}$.

2.6 对于任意 $\mathcal{J}$，使得 $\mathsf{J}(\mathcal{J}, \overset{e_{\mathrm{FIN}}}{\mathcal{V}}{}^{\text{投票}(t_i)}_i), \overset{e_{\mathrm{FIN}}}{\mathrm{MFC}}{}^{\mathsf{vote}(t_i)}_i \succeq \mathcal{J}.\mathsf{ch}$。

并将条件 1 改述如下

2.7 e_{FIN} 和 e_{FIN} 在第 4Δ 轮（t_i+1）之前是诚实输出动态等价的。

注意，对于任意 t_i，条件 2.7 成立都意味着条件 1 成立。

为减少证明中的重复内容，我们在归纳步骤中处理归纳法的基础情形。此外，令 $\mathcal{T}^{\mathsf{vote}(t_\ell)}{}_j$ 为验证者 v_j 在时隙 t_ℓ发送 FIN-VOTE 时使用的目标检查点。

接着，任取 $t_i \geq 0$，并假设若 $t_i > 0$，则引理以及附加条件 2.5 至 2.7 对时隙 $t_i - 1$成立。我们证明它们对时隙 t_i也成立。首先证明条件 2.5 和 2.6 对时隙 t_i成立，然后依次证明条件 2.1 至 2.4，最后证明条件 2.7。

条件 2.5 和 2.6。令 $\mathcal{J}$为任意检查点，使得 $\mathsf{J}(\mathcal{J}, \mathcal{V}^{e_{\text{FIN}}}(t_i)_i)$。由于诚实验证者 v_ℓ 在第 r_ℓ 轮发送的 FIN-VOTE 的目标检查点对应于 $(\mathcal{T}^{e_{\text{FIN}}}_{\ell}{}^{r_\ell}.\mathsf{ch}\ \cdot)$，且我们假设$f < \frac{n}{3}$，因此存在时隙 $t_k \in [0, t_i)$ 和验证者 $v_k \in H_{\mathsf{vote}(t_k)}$，使得 $\mathcal{T}$对$(t_k)_k \succeq \mathcal{J}.\mathsf{ch}$。根据归纳假设，条件 2.3 意味着存在时隙 $t_m \in [0, t_k]$ 和验证者 $v_m \in H_{\mathsf{vote}(t_m)}$，使得 $\mathsf{Ch}^{\mathsf{vote}(t_m)}{}_m \succeq \mathsf{chAva}^{\mathsf{vote}(t_k)}{}_k$。此外，由于代码行 12、15 和 17，显然 $\mathsf{chAva}^{\mathsf{vote}(t_k)}{}_k \succeq \mathcal{T}^{\mathsf{vote}(t_k)}{}_k$。鉴于 $t_m \leq t_k < t_i$，根据引理 5，我们知道 $\mathrm{MFC}^{e_{\text{FIN}}\,\mathsf{propose}(t_i)} \succeq \mathsf{Ch}^{\mathsf{vote}(t_m)}{}_m \succeq \mathsf{chAva}^{\mathsf{vote}(t_k)}{}_k \succeq \mathcal{T}$对$(t_k)_k \succeq \mathcal{J}.\mathsf{ch}$，且 $\mathrm{MFC}^{e_{\text{FIN}}\,\mathsf{vote}(t_i)} \succeq \mathsf{Ch}^{\mathsf{vote}(t_m)}{}_m \succeq \mathsf{chAva}^{\mathsf{vote}(t_k)}{}_k \succeq \mathcal{T}$对$(t_k)_k \succeq \mathcal{J}.\mathsf{ch}$。鉴于 $\mathcal{V}^{e_{\text{FIN}}\,\mathsf{vote}(t_i)}{}_i$ 中的已获正当化检查点集合是 $\mathcal{V}^{e_{\text{FIN}}\,\mathsf{propose}(t_i)}{}_i$ 中已获正当化检查点集合的超集，条件 2.5 和条件 2.6 均得证。

条件 2.1 和 2.2。我们知道，e_{FIN} 和 e_{FIN} 在提议(t_i)轮次之前是诚实输出动态等价的。如果 $t_i > 0$，那么根据归纳假设，条件 2.7 蕴含这一点；如果 $t = 0$，那么这一点显然成立。因此，由于 $A^{e_{\text{FIN}}}$ 的决策集合中的第 (iv) 项，$\mathcal{V}^{e_{\text{FIN}}\,\mathsf{propose}(t_i)}{}_i$ 和 $\mathcal{V}^{e_{\text{FIN}}\,\mathsf{propose}(t_i)}{}_i$是动态等价的。由此可知，$e_{\text{FIN}}$ 和 e_{FIN} 在投票(t_i)轮次之前也同样是诚实输出动态等价的，因此，$\mathcal{V}^{e_{\text{FIN}}\,\mathsf{vote}(t_i)}{}_i$ 和 $\mathcal{V}^{e_{\text{FIN}}\,\mathsf{vote}(t_i)}{}_i$ 也动态等价。这一点以及条件 2.5 和 2.6 蕴含条件 2.1 和 2.2。

条件 2.3。根据第 12 行，$\mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i \in \{\mathsf{chAva}^{\mathsf{fconf}(t_i-1)}{}_i, \mathcal{GJ}_{\text{冻结,投票}(t_i)\,i}.\mathsf{ch}, (\mathrm{MFC}^{e_{\text{FIN}}\,\mathsf{vote}(t_i)}{}_i)^{\lceil\kappa}\}$。我们来逐一考察每种情况。

情况 1： $\mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i = \mathsf{chAva}^{\mathsf{fconf}(t_i-1)}{}_i$。该情况意味着 $t_i > 0$。因此，我们可以应用归纳假设。具体而言，时隙 $t_i - 1$ 的条件 2.4 意味着时隙 t_i 的条件 2.3。

情况 2： $\mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i = \mathcal{GJ}_{\text{冻结}_i}{}^{t_i}.\mathsf{ch}$ 由于第 27、33 和 34 行，$\mathsf{J}(\mathcal{GJ}_{\text{冻结}_i}{}^{t_i}, \mathcal{V}^{\mathsf{vote}(t_i)})$。因此，按照讨论条件 2.5 和 2.6 时所采用的推理，存在一个时隙 $t_k \in [0, t_i)$ 和验证者 $v_k \in H_{\mathsf{vote}(t_k)}$，使得 $\mathcal{T}_{\text{投票}(t_k)}{}_k \succeq \mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i$，由此我们可以得出 $\mathsf{chAva}^{\mathsf{vote}(t_k)}{}_k \succeq \mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i$。因此，我们可以应用归纳假设。具体而言，时隙 t_k 的条件 2.3 意味着时隙 t_i 的条件 2.3。

情况 3： $\mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i = (\mathrm{MFC}^{e_{\text{FIN}}\,\mathsf{vote}(t_i)}{}_i)^{\lceil\kappa}$。算法 2 的第 22 行意味着 $\mathsf{Ch}_i^{\mathsf{vote}(t_i} \succeq \mathrm{MFC}_i^{e_{\text{FIN}}\,\mathsf{vote}(t_i)})$。随后，条件 2.2 意味着 $\mathsf{Ch}^{\mathsf{vote}(t_i)}{}_i \succeq \mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i$。

条件 2.4。根据第 21 至 23 行，$\mathsf{chAva}^{\mathsf{fconf}(t_i)}{}_i \in \{\mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i, \mathcal{GJ}(\mathcal{V}^{e_{\text{FIN}}}{}_{\mathsf{fconf}(t_i)}).\mathsf{ch}, \mathsf{ch}^C\}$ with $(\mathsf{ch}^C, Q) =$快速确认$(\mathcal{V}^{e_{\text{FIN}}}{}_{\mathsf{fconf}(t_i)}{}_i, t_i) \wedge Q \neq \emptyset$。下面分别考虑每种情况。

情况 1： $\mathsf{chAva}^{\mathsf{fconf}(t_i)}{}_i = \mathsf{chAva}^{\mathsf{vote}(t_i)}{}_i$。在这种情况下，条件 2.3 意味着条件 2.4 .

情况 2： $\mathsf{chAva}^{\mathsf{fconf}(t_i)}{}_i = \mathcal{GJ}(\mathcal{V}^{e_{\text{FIN}}}{}_{\mathsf{fconf}(t_i)}{}_i).\mathsf{ch}$。按照讨论条件 2.5 和 2.6 时所采用的推理，这意味着存在一个时隙 $t_k \in [0, t_i]$ 和一个验证者 $v_k \in H_{\mathsf{vote}(t_k)}$，使得 $\mathsf{chAva}^{\mathsf{vote}(t_k)}{}_k \succeq \mathsf{chAva}^{\mathsf{fconf}(t_i)}{}_i$。那么，时隙 t_k 的条件 2.3 意味着时隙 t_i满足条件 2.4。

情况 3： $\mathsf{chAva}^{\mathsf{fconf}(t_{i_i})\ i} = \mathsf{ch}^C$ 与$(\mathsf{ch}^C,\ Q) =$ 快速确认$(\mathcal{V}^{e_{\mathsf{FIN}}}_{\mathsf{fconf}(t_i)\ i},\ t_i) \wedge Q \neq \emptyset$。这意味着 $\mathcal{V}^{e_{\mathsf{FIN}}}_{\mathsf{fconf}(t_i)\ i}$ 包含链 ch^C的 VOTE 消息的一个法定人数。根据条件 2.2，$\mathcal{V}^{e_{\mathsf{FIN}}}_{\mathsf{fconf}(t_{i_i})\ i}$ 也包含链 ch^C的 VOTE 消息的一个法定人数。因此，$\mathsf{Ch}^{\mathsf{fconf}(t_{i_i})\ i} = \mathsf{chAva}^{\mathsf{fconf}(t_{i_i})\ i}$。

条件 2.7。正如条件 2.1 和 2.2 的证明中所论证的，我们知道，e_{FIN} 和 e_{FIN} 在提议轮次 $\mathsf{propose}(t_i)$ 之前是诚实输出动态等价的。这一点、条件 2.1 和 2.2，以及$A^{e_{\mathrm{FIN}}}$的决策集合中的第 (iv) 项，显然意味着条件 2.7 在提议轮次 $\mathsf{propose}(t_i + 1)$ 之前成立。 □

□

在接 下来的引理和定理中，除非另有说明，我们讨论算法 5 的执行 .

引理 11（引理 2 的类似结论）。如果在时隙 t，$H_{\mathsf{vote}(t)}$ 中的所有验证者都发送针对扩展链 ch的链的 VOTE 消息，那么，对于任意验证者 $v_i \in H_{\mathsf{vote}(t+1)}$，$\mathrm{MFC}_i^{\mathsf{propose}(t+1)} \succeq \mathsf{ch}$ 且 $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}$，这意味着，在时隙 $t+1$，$H_{\mathsf{vote}(t+1)}$ 中的所有验证者都会发送针对扩展ch的链的 VOTE 消息。

证明。取算法 5 的任意一次合规执行 e_{FIN}，其中在时隙 t，$H_{\mathsf{vote}(t)}$ 中的所有验证者都发送了针对扩展链 ch 的链的 VOTE消息。设 e_{FIN} 是算法 2 的一次 MFC等价执行；引理 10 证明了这次执行确实存在。根据 MFC等价诚实输出动态等价执行的定义，显然，在执行 e_{FIN} 的时隙 t，$H_{\mathsf{vote}(t)}$ 中的所有验证者也都发送了针对扩展链 ch 的链的 VOTE 消息。因此，根据引理 2，在执行 e_{FIN}中，对于任意验证者 $v_i \in H_{\mathsf{vote}(t+1)}$，$\mathrm{MFC}_i^{\mathsf{propose}(t+1)} \succeq \mathsf{ch}$ 且 $\mathrm{MFC}_i^{\mathsf{vote}(t+1)} \succeq \mathsf{ch}$。由引理 10 的条件 2.1 和 2.2 可知，这对执行 e_{FIN} 同样成立，这意味着在时隙 $t+1$，$H_{\mathsf{vote}(t+1)}$ 中的所有验证者都发送VOTE消息，针对的是扩展链 ch 的链。

□

引理 12（引理 4 的类似结果）。设 t 是一个时隙，其提议者 v_p 诚实，并假设 v_p 发送了一条 [PROPOSE, ch_p、ch_p^C、Q_p^C、t、v_p] 消息。那么，对于任意时隙 $t' \geq t$，$H_{\mathsf{vote}(t')}$ 中的所有验证者都会发送一条消息，其链扩展自 ch_p。此外，对于任意时隙 $t'' > t$ 以及任意验证者$v_i \in H_{\mathsf{vote}(t'')}$，$\mathrm{MFC}^{\mathsf{propose}(t'')\ i} \succeq \mathsf{ch}_p$，且 $\mathrm{MFC}^{\mathsf{vote}(t'')\ i} \succeq \mathsf{ch}_p$。

证明。取算法 5 的任意一次合规执行 e_{FIN}，其中 v_p（时隙 t 的提议者）是诚实的，并发送了一条 [PROPOSE, ch_p、ch_p^C、$Q_p^C t$、v_p] 消息。令 e_{FIN} 为算法 2 的一次 MFC-等价执行；引理 10 证明了该执行存在。根据 MFC-等价的诚实输出动态等价执行的定义，显然 v_p 在执行 e_{FIN} 的时隙 t 中发送相同的 PROPOSE 消息。因此，根据引理 4，在执行 e_{FIN} 中，对于任意时隙 $t' \geq t$，$H_{\mathsf{vote}(t')}$ 中的所有验证者都会发送一条消息，其链扩展自 ch_p，

并且，对于任意时隙 $t'' > t$ 和任意验证者 $v_i \in H_{\mathsf{vote}(t'')}$，$\mathrm{MFC}^{\mathsf{propose}(t'')\ i} \succeq \mathsf{ch}_p$ 以及 $\mathrm{MFC}^{\mathsf{vote}(t'')\ i} \succeq \mathsf{ch}_p$。由引理 10 的条件 1、2.1 和 2.2 可知，这对执行 e_{FIN} 也成立。 □

引理 13（引理 5 的类似结果）。设 r_i 为任意轮次，r_j 为满足 $r_j \geq r_i$ 且$r_j \in$ {提议（r_j）、投票（r_j）}的任意轮次。那么，对于在轮次 r_i 中诚实的任意验证者 v_i 以及任意验证者 $v_j \in H_{r_j}$，$\mathsf{chAva}_i^{r_i} \preceq \mathrm{MFC}_j^{r_j}$。

证明。取算法 5 的任意合规执行 e_{FIN}。令 e_{FIN} 为算法 2 的一个等价执行；引理 10 证明这样的执行存在。设 r_i 为任意轮次，r_j 为满足 $r_j \geq r_i$且 $r_j \in$ {提议(r_j)，投票(r_j)}的任意轮次。我们知道，$\mathsf{chAva}_i^{r_i}$首先由 v_i 在满足$r_i' \in$ {投票(slot(r_i'))，fconf(slot(r_i'))}的轮次 $r_i' \leq r_i$ 中设定。由引理 10 的条件 2.3 和 2.4 可知，存在一个时隙 $t_m \leq r_k$ 和一个验证者 $v_m \in H_{\mathsf{vote}(t_m)}$，使得 $\mathsf{Ch}^{\mathsf{vote}(t_m)}{}_m \succeq \mathsf{chAva}_i^{r_i'} = \mathsf{chAva}_i^{r_i}$。引理 5 推出 $\mathrm{MFC}_j^{r_j} \succeq \mathsf{Ch}^{\mathsf{vote}(t_m)}{}_m$。由引理 10 可知，$\mathrm{MFC}_j^{r_j} = \mathrm{MFC}_j^{r_j}$。因此，$\mathrm{MFC}_j^{r_j} = \mathrm{MFC}_j^{r_j} \succeq \mathsf{Ch}^{\mathsf{vote}(t_m)}{}_m \succeq \mathsf{chAva}_i^{r_i'} = \mathsf{chAva}_i^{r_i}$。

□

定理 3（重组韧性——定理 1 的类似结论）。 算法 *5* 具备重组韧性。

证明。 证明过程与定理 1 的证明相同，仅有以下几点改动。

1. 将 Ch 替换为 chAva,

2. 将引理 4 替换为引理 12，并且 3. 将引理 5 替换为引理 13。 □

□

定理 4（动态可用性——定理 2 的类似结果）。 *算法 5 以压倒性概率具有动态可用性。*

证明。 注意，引理 6 对算法 5 同样成立。因此，证明与定理 2 的证明相同，仅有以下几点变化。

1. 将 Ch 替换为 chAva，

2. 将对算法 2 第 22 行的引用替换为算法 5 第 12 行，

3. 将引理 4 替换为引理 12，

4. 将引理 5 替换为引理 13；5. 在活性证明中，考虑以下额外情形。假设 $r_i \geq$ 投票$(t+2\kappa)$ 是一个快速确认轮次，并且 $\mathsf{chAva}_i^{r_i}$通过第 23 行被设为 $\mathcal{GJ}(\mathcal{V}_i^{r_i})$.ch。由于诚实验证者 v_ℓ 在轮次 r_ℓ 发送的 FIN-VOTE 的目标检查点所对应的链为 $(\overset{e_{\text{FIN}}}{\mathcal{T}}{}_\ell^{r_\ell}.\mathsf{ch}\ \cdot)$，其中 $e_{\text{FIN}}\mathsf{chAva}_\ell^{r_\ell}.\mathsf{ch} \succeq \overset{e_{\text{FIN}}}{\mathcal{T}}{}_\ell^{r_\ell}.\mathsf{ch}$，且我们假设 $f < \frac{n}{3}$，因此存在一个时隙 $t_k \in [0,\mathsf{slot}(r_i)]$ 和一个验证者$v_k \in H_{\mathsf{vote}(t_k)}$，使得 $\mathsf{chAva}^{\mathsf{vote}(t_k)}{}_k \succeq \mathcal{GJ}(\mathcal{V}_i^{r_i}).\mathsf{ch} = \mathsf{chAva}_i^{r_i}$。我们已经证明 $\mathsf{chAva}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i \succeq \mathsf{ch}_p$。于是，引理 13 意味着 $\mathrm{MFC}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i \succeq \mathcal{GJ}(\mathcal{V}_i^{r_i}).\mathsf{ch} = \mathsf{chAva}_i^{r_i}$，并且$\mathrm{MFC}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i \succeq \mathsf{chAva}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i \succeq \mathsf{ch}_p$。这意味着 $\mathsf{chAva}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i$ 与 $\mathcal{GJ}(\mathcal{V}_i^{r_i}).\mathsf{ch}$ 不冲突。因此，鉴于我们假设第 23 行已执行，$\mathsf{chAva}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i \prec \mathcal{GJ}(\mathcal{V}_i^{r_i}).\mathsf{ch}$。由于 $\mathsf{chAva}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i \succeq \mathsf{ch}_p$，我们可以得出结论：$\mathsf{chAva}_i^{r_i} = \mathcal{GJ}(\mathcal{V}_i^{r_i}).\mathsf{ch} \succ \mathsf{chAva}^{\mathsf{vote}(\mathsf{slot}(r_i))}{}_i \succeq \mathsf{ch}_p$。

□

□

引理 14（引理 7 的类似结论）。 *取一个时隙 t，其中 $|H_{\mathsf{vote}(t)}| \geq \frac{2}{3}n$。若在时隙 t，一名诚实验证者为链 ch_p发送一条 PROPOSE 消息，则对于任意验证者 $v_i \in H_{\mathsf{fconf}(t)}$，$\mathsf{chAva}_i^{\mathsf{fconf}(t)} \succeq \mathsf{ch}_p$。*

证明。 证明过程与引理 7 的证明相同，唯一的变化是将 Ch_i 替换为 chAva_i。 □

C.2 部分同步

本节将证明，算法 5 可确保链 chFin 始终是可问责安全的，并且在时间 max(GST，GAT) $+\Delta$ 之后保持活性；这意味着，在假设 $f < \frac{n}{3}$ 成立的情况下，算法 5 是一种安全涨落协议。

我们首先证明，遵循算法 5 的诚实验证者绝不会被罚没 d.

引理 15。 *遵循算法 5 的诚实验证者绝不会被罚没。*

证明。 首先，注意到

1. 根据第 20 行，诚实的活跃验证者每个时隙只发送一个 FIN-VOTE 。

2. 根据第 15 行，因此，若 $\mathcal{T}$是时隙 t 中发送的 FIN-VOTE 里的目标检查点，则 $\mathcal{T}.c = t.$

3. 任取两个时隙 t 和 t'，满足 $t < t'$，且 $v_i \in H_{\mathsf{vote}(t)}$ 和 $v_i \in H_{\mathsf{vote}(t')}$。令 $\mathcal{S}^\ell$ 为验证者 v_i 在任意时隙 ℓ 发送的 FIN-VOTE 的源检查点。那么 $\mathcal{S}^t.c \leq \mathcal{S}^{t'}.c$。具体而言，任取两个时隙 k 和 k'，满足 $t \leq k < k' \leq t'$、$v_i \in H_{\mathsf{vote}(k)}$，且 k' 是 k 之后满足$v_i \in H_{\mathsf{vote}(k')}$的第一个时隙。根据第20行，我们知道 $\mathcal{S}^k = \mathcal{GJ}^{\text{frozen}}{}_i^{\mathsf{vote}(k)}$；根据第27行和第33行，我们有 $\mathsf{J}(\mathcal{GJ}_{\text{冻结，投票}(k)\,i}, \mathcal{V}_i^{\mathsf{vote}(k)})$。然后注意到 $k \leq k'-1$，并且由于加入协议，我们知道

在合并$(k'-1)$轮次，v_i处于活跃状态。因此，我们有$\mathcal{GJ}_{\text{冻结}i}$，$.c \leq \mathcal{GJ}_{\text{投票}(k)}(\mathcal{V}^{\mathsf{merge}(k'-1)}{}_i)$ $.c = \mathcal{GJ}_{\text{冻结，合并}(k'-1)\,i}$ $.c$。最后，根据第33行和第34行，$\mathcal{GJ}_{\text{冻结}}$ $_i$，$k'-1$ $.c \leq \mathcal{GJ}^{\text{frozen合并}(),\,\mathsf{vote}(k')\,i}$ $.c = \mathcal{S}^{k'}.c$和$\mathcal{S}^{k}.c \leq \mathcal{S}^{k'}.c$。

现在，任取轮次 r 以及在轮次 r中 v_i 诚实的任意验证者。再任取 FIN-VOTE个不同的{s}$\mathcal{C}_1 \to \mathcal{C}_2$ 和 $\mathcal{C}_3 \to \mathcal{C}_4$，它们在轮次 r之前已发送。如果不存在这样的 FIN-VOTEs，那么显然 v_i 不能被罚没。根据上述第1点和第2点，我们有 $\mathcal{C}_2.c \neq \mathcal{C}_4.c$。因此，$\mathbf{E_1}$ 未被违反。然后，不失一般性地，假设 $\mathcal{C}_2.c < \mathcal{C}_4.c$。这意味着 FIN-VOTE $\mathcal{C}_1 \to \mathcal{C}_2$ 已在 FIN-VOTE $\mathcal{C}_3 \to \mathcal{C}_4$之前发送。接着，根据上述第3点，我们知道 $\mathcal{C}_1.c \leq \mathcal{C}_3.c$。鉴于 $\mathcal{C}_2.c < \mathcal{C}_4.c$，$\mathbf{E_2}$ 不可能被违反。因此，在v_i 保持诚实之前，它不会被罚没。 □

令 $\mathcal{V}^r_{\mathsf{G}}$ 表示时间 r时的全局视图，即截至并包括时间 r为止发送的所有消息的集合。算法 5中，诚实验证者 v_i 在轮次 r 输出的链 chFin_i^r，是根据视图 $\mathcal{V}^r_{\mathsf{G}}$已最终确定的任意链。这直接由第3行、第13行和第24行以及 $\mathcal{GF}$的定义得出。

Theo引理 5 (可问责安全性)。 设 $f \in [0, n]$。最终确定链 *chFin </style}*是 $\frac{n}{3}$-可问责的 .

证明。 由引理9和引理15得出。 □

现在我们证明，诚实验证者的最终确定链会单调增长。虽然第 2.2 节没有明确界定这一性质，但以太坊协议实现者 [1] 经常需要它。

Lem引理 16。 对于任意两个轮次 $r' \geq r$，以及在轮次 r'中诚实的验证者 v_i，$\mathsf{chFin}_i^{r'} \succeq \mathsf{chFin}_i^r$.

证明。 我们用反证法证明这个引理。设 r 是满足以下条件的最小轮次：存在轮次 $r' \geq r$ 和一名在轮次 r' 中诚实的验证者 v_i，使得 $\mathsf{chFin}_i^{r'} \not\succeq \mathsf{chFin}_i^r$。显然 $r' > r$。假设 r' 是满足条件的最小轮次。首先，我们要证明存在一个检查点 $\mathcal{J}$ $_i^{r'} \succeq$，使得 $\mathcal{J}.$ ∧ $\mathcal{J}$ $\mathcal{V}_i^{r'} \wedge \mathcal{J} \geq \mathcal{GF}$ $\mathcal{V}_i^r$ r r' chAva ch J(,) ( )。注意，两者要么是投票轮次，要么是快速确认轮次。这是由 r 和 r' 的最小性，以及 chFin_i 仅在这些类型的轮次中设置这一事实所致。接下来，我们分情况讨论。

情况 1： r' **是投票轮次**。 在这种情况下，$\mathcal{J} = \mathcal{GJ}^{\text{frozen}}{}_{,\,i}$ $^{r'}$。请注意，由于代码行 27、28、35 和 36，$\mathsf{ch}_i^{\text{frozen}\,r'} \succeq \mathcal{GJ}^{\text{frozen}}{}_{、,\,i}$ $^{r'}$.ch。然后，第 12 行意味着 $\mathsf{chAva}_i^{r'} \succeq \mathcal{GJ}_{\text{冻结}_i}$ $^{r'}$.ch。此外，代码行 28、33 和 34 意味着 J($\mathcal{GJ}_{\text{冻结}_i}$ $^{r'}$，$\mathcal{V}_i^{r'}$)。最后，代码行 28、33 和 34，以及 $r' > r$ 意味着 $\mathcal{GJ}_{\text{冻结}}$ $_i^{r'} \geq \mathcal{GJ}(\mathcal{V}_i^r)$。

情况 2： r' **是快速确认轮次**。 在这种情况下，$\mathcal{J} = \mathcal{GJ}(\mathcal{V}_i^{r'})$。显然，J($\mathcal{GJ}(\mathcal{V}_i^{r'})$ $\mathcal{V}_i^{r'}$)。代码行 22 至 23 确保 $\mathsf{chAva}_i^{r'} \succeq \mathcal{GJ}(\mathcal{V}_i^{r'})$。ch。然后，根据算法 3 以及 $\mathcal{V}_i^r \subseteq \mathcal{V}_i^{r'}$ 这一事实，我们可以得出结论：$\mathcal{GJ}(\mathcal{V}_i^{r'})$ $\geq \mathcal{GF}(\mathcal{V}_i^{r'})$ $\geq \mathcal{GF}(\mathcal{V}_i^r).c$。

根据算法 3 和 $\mathcal{V}_i^r \subseteq \mathcal{V}_i^{r'}$，我们还有 $\mathcal{GF}(\mathcal{V}_i^{r'})$ $\geq \mathcal{GF}(\mathcal{V}_i^r)$。接着，引理 8 和 $f < \frac{n}{3}$ 意味着 chAva $_{-}$ $_i^{r'} \succeq \mathcal{J}$.ch $\succeq \mathcal{GF}(\mathcal{V}_i^r)$。ch $\succeq \mathsf{chFin}_i^r$ 以及 $\mathcal{GF}(\mathcal{V}_i^{r'})$。ch $\succeq \mathcal{GF}(\mathcal{V}_i^r)$。ch，这进一步意味着 $\mathsf{chFin}_i^{r'} \succeq \mathcal{GF}(\mathcal{V}_i^r)$。ch。因此，$\mathsf{chFin}_i^{r'} \succeq \mathcal{GF}(\mathcal{V}_i^r)$。ch $\succeq \mathsf{chFin}_i^r$，从而得出矛盾。 □

现在我们来讨论活性性质。

引理 17. 设 t 为一个时隙，其诚实提议者为 v_p，且 提议$(t) \geq \mathsf{GST} + \Delta$，并假设 v_p发送了一条 $[\text{PROPOSE}, \mathsf{ch}_p$、$\mathsf{ch}_p^C$、$Q_p^C$、$\mathcal{GJ}_p$、$t$、$v_p]$ 消息。那么，对于任意验证者 $v_i \in H_{\mathsf{vote}(t)}$，$v_i$ 都会为链 ch_p 发送一条 VOTE消息，并且 $\mathcal{GJ}_{\text{冻结}i}$ $, = \mathcal{GJ}_{p\text{投票}(t)}$。

证明。 假设在时隙 t 中，一名诚实提议者发送了一条 $[\text{PROPOSE}, \mathsf{ch}_p$、$\mathsf{ch}_p^C$、$Q_p^C$、$\mathcal{GJ}_p$、$t$、$v_p]$ 消息。考虑 v_i 中的一名诚实验证者。注意，由于同步性假设，$\mathcal{V}_i^{\mathsf{merge}(t-1)} \subseteq \mathcal{V}_p^{\mathsf{propose}(t)}$。因此，$\mathcal{GJ}_p \geq \mathcal{GJ}_{\text{冻结、合并}(t-1)\,i}$。现在，我们要证明 $\mathsf{ch}_i^{\text{frozen}\ \mathsf{vote}(t)} =$、$\mathsf{ch}_p^C$。让我们考虑两种情况。

情况 1： $\mathsf{ch}_i^{\text{frozen merge}(t-1)} \succ \mathcal{GJ}^{\text{frozen}}$、、$_i$ $^{\text{merge}(t-1)}$.ch。这意味着在时间 merge($t-1$)，v_i 已收到针对 $\mathsf{ch}_i^{\text{frozen merge}(t-1)}$、的一组法定人数的 VOTE 条消息。让我们考虑三种子情况。

情况 1.1： $\mathsf{ch}_i^{\text{frozen merge}(t-1)} = \mathcal{GJ}_{p}$.、ch。鉴于 $\mathsf{ch}_p^C \succeq \mathcal{GJ}_p$.ch，第 37 和 38 行将 $\mathsf{ch}_i^{\text{frozen vote}(t)} =$、$\mathsf{ch}_p^C$设为。

情况 1.2： $\mathsf{ch}_i^{\text{frozen merge}(t-1)} \succ \mathcal{GJ}_{p}$.、ch。这意味着 $\mathcal{V}_p^{\text{propose}(t)}$包含 ch_p^C的 VOTE 条消息法定人数。根据同步性假设，$\mathsf{ch}_i^{\text{frozen merge}(t-1)}$ 的 VOTE 条消息法定人数，在时间 propose(t)时位于验证者 v_p 的视图中。鉴于 $f < \frac{n}{3}$，此情况意味着 $\mathsf{ch}_i^{\text{frozen merge}(t-1)} =$、$\mathsf{ch}_p^C$，因此 $\mathsf{ch}_i^{\text{frozen vote}(t)} =$、$\mathsf{ch}_p^C$。

情况 1.3： $\mathsf{ch}_i^{\text{frozen merge}(t-1)} \not\succeq \mathcal{GJ}_{p}$.、ch。在此情况下，第 35 至 38 行将 $\mathsf{ch}_i^{\text{frozen vote}(t)} =$、$\mathsf{ch}_p^C$设为。

情况 2： $\mathsf{ch}_i^{\text{frozen merge}(t-1)} = \mathcal{GJ}_{\text{冻结，合并}(t-1)\,i}$ ·ch。鉴于 $\mathsf{ch}_p^C \succeq \mathcal{GJ}_p$ 以及 $\mathcal{GJ}_p \geq \mathcal{GJ}_{\text{冻结，合并}(t-1)\,i}$，代码行 33 至 38 表明 $\mathsf{ch}_i^{\text{frozen vote}(t)} =$、$\mathsf{ch}_p^C$。

现在，首先假设在轮次 vote(t)，存在一条链 ch $\succeq \mathsf{ch}_i^{\text{frozen}} = \mathsf{ch}_p^C$，满足$\left|\left(\mathcal{V}^{\text{frozen}}\right)^{\mathsf{ch},\ t} \cap \mathcal{V}^{\mathsf{ch},\ \ t}_{\ i}\right| > |\mathbb{S}(\mathcal{V}_{i,\ \ t\mid})$。根据分级交付性质[7]，，这意味着在时间 propose(t)，$\left|\mathcal{V}^{\mathsf{ch},\ \ t}_{\ p}\right| > |\mathbb{S}(\mathcal{V}_{p,\ \ t\mid})$，也就是说 $\mathsf{ch}_p \succeq$ ch，因此在时隙 t，v_i 会为 ch_p发送一条 VOTE 消息。

如果不存在这样的链，v_i 仍会为 ch_p 发送一个 VOTE，因为 $\mathsf{ch}_p \succeq \mathsf{ch}_p^C = \mathsf{ch}_i^{\text{frozen}}$。

此外，鉴于 $\mathcal{GJ}_p \geq \mathcal{GJ}_{\text{冻结，合并}(t-1)\,i}$，根据第33行和第34行，$\mathcal{GJ}^{\text{frozen},\ \text{vote}(t)} = \mathcal{GJ}_p$。 □

引理 18。假设 $f < \frac{n}{3}$。设 v_p 是在其为时隙 t 的链 ch_p 提议提议(t) $\geq \max(\mathsf{GST},\mathsf{GAT}) + 4\Delta$ 时诚实的任意验证者，并假设 $\mathcal{GJ}_{\text{提议}(t)p}$.$c < t-1$。那么，链 ch_p 在时隙 $t+1$ 已获正当化，并在时隙 $t+2$ 已最终确定。

证明。设 ch_p 是诚实提议者在时隙 t 提议的链，其中 propose(t) $\geq \max$(GST，GAT)+4Δ，并同时扩展分叉选择函数 MFC 的输出和最大已论证检查点 $\mathcal{GJ}_p$，且带有 $\mathcal{GJ}_p^{\text{frozen}}.c < t-1$。

根据第 17 行和引理 17，每个诚实验证者都会将目标检查点设为同一个检查点$\mathcal{T}$:= ($\mathcal{GJ}_{\text{提议}(t)p}$ ·ch t)。

然后，同样根据引理 17，在时隙 t 中，所有诚实验证者 v_i 都会发送形式为 $[\text{VOTE}, \mathsf{ch}_p, \mathcal{GJ}_p^{\text{propose}(t)} \to \mathcal{T}, t, v_i]$ 的消息 VOTE，这些消息会证明 $\mathcal{T}$（因为 $f < n/3$ 确保了足够数量的诚实法定人数）。

因此，在时隙 $t+1$ 中，$\mathcal{T}$是所有诚实验证者视图中的最大已论证检查点，并被用作 FIN-VOTE 的源。引理 17 还意味着 $\mathsf{chAva}_i^{\text{vote}(t+1)} = \mathsf{ch}_p$。

因此，根据第 14 行和第 15 行，在时隙 $t+1$ 中，任何诚实验证者都会发送源为 $\mathcal{T}$、目标为 (ch_p, $t+1$) 的 FIN-VOTE，从而证明 (ch_p, $t+1$)。在时隙 $t+2$ 中，所有诚实验证者都以 (ch_p, $t+1$) 为源进行投票；到轮次 $4\Delta(t+2) + 2\Delta$ 时，所有诚实验证者都能看到这些投票，从而在时隙 $t+2$ 最终确定 ch_p。□

定理 6 (活性)。假设 $f < \frac{n}{3}$，且令 t和 $t+1$ 为晚于 $\max(\mathsf{GST},\mathsf{GAT}) + 4\Delta$的时隙。那么，时隙 t 的提议者所提议的链 ch_p 会在时隙 $t+2$ 或时隙 $t+3$被最终确定。

特别地，对于任意验证者 $v_i \in H_r$，其中 $r \geq 4\Delta(t+3)+2\Delta$，有 $\mathsf{ch}_p \preceq \mathsf{chFin}_i^r$且交易池$^{\text{propose}(t)} \subseteq \mathsf{chFin}_i^r$.

证明。令 v_{p_1} 和 v_{p_2} 分别为时隙 t 和 $t+1$中的诚实提议者。首先假设$\mathcal{GJ}_p^{\text{propose}(t)}.c = t-1$。我们区分两种情况：

1. 情况 1：在时隙 $t+1$ 的提议时间，提议者 v_{p_2} 的视图中最大已论证检查点的时隙为 t，即对于某条链 ch′，$\mathcal{GJ}$ propose($t+1$)$_{p_2}$ $= \mathcal{C} = (\mathsf{ch}', t)$。根据引理 17 以及第 14 行和第 15 行，在时隙 $t+1$ 的所有诚实验证者 v_i 都有 $\mathcal{GJ}^{\text{frozen}}$,$_i$ $^{\text{vote}(t+1)} = \mathcal{GJ}$propose($t+1$)$_{p_2}$ $= \mathcal{C}$，并发送源为 $\mathcal{C}$、目标为 ($\mathsf{ch}_p, t+1$) 的 FIN-VOTEs。由于 $f < \frac{n}{3}$，这些投票会在时隙 $t+1$ 证明 ($\mathsf{ch}_p, t+1$) 合理。在时隙 $t+2$，诚实验证者以 ($\mathsf{ch}_p, t+1$) 为源进行投票。到轮次 $4\Delta(t+$

2) $+ 2\Delta$，这些投票会出现在每个诚实验证者的视图中，从而在时隙 $t+2$ 最终确定 ch_p。特别是，$\mathsf{ch}_p \preceq \mathcal{GF}(\mathcal{V}^{4\Delta(t+2)+2\Delta})$。ch，并且根据第 24 行，$\mathsf{ch}_p \preceq \mathsf{chFin}_i^{4\Delta(t+2)+2\Delta}$。根据引理 6，可得 交易池$^{\mathsf{propose}(t)} \subseteq \mathsf{chFin}_i^{4\Delta(t+2)+2\Delta}$。根据引理16，对于每个 $r \geq 4\Delta(t+3) + 2\Delta$，$\mathsf{chFin}_i^{4\Delta(t+2)+2\Delta} \preceq \mathsf{chFin}_i^{4\Delta(t+3)+2\Delta} \preceq \mathsf{chFin}_i^r$；由此即可得出结果。

2. **情况 2：时隙 t 的证明未被 v_{p_2} 在时隙 $t+1$ 的提议时间观察到**。在这种情况下，我们对时隙 $t+1$ 应用引理 18，这意味着 v_{p_2} 在时隙 $t+1$ 提议的链 PROPOSEd 在时隙 $t+2$ 获得正当化，并在时隙 $t+3$ 已最终确定。由于这条链扩展了 v_{p_1} 提议的 ch_p（依据引理 12），因此 ch_p 在时隙 $t+3$ 作为最终确定链的前缀而已最终确定。特别地，$\mathsf{ch}_p \preceq \mathcal{GF}(\mathcal{V}^{4\Delta(t+3)+2\Delta})$。ch，并且根据第 24 行，$\mathsf{ch}_p \preceq \mathsf{chFin}_i^{4\Delta(t+3)+2\Delta}$。根据引理 6，可得 交易池$^{\mathsf{propose}(t)} \subseteq \mathsf{chFin}_i^{4\Delta(t+3)+2\Delta}$

注意，如果 $\mathcal{GJ}_p^{\mathsf{propose}(t)}.c < t-1$，那么我们可以立即应用引理 18，使 v_p^t 在时隙 $t+2$ 提议的链 PROPOSE 已最终确定。 □

定理 7. 算法 *5* 是一个安全涨落协议，且以压倒性概率成立。

证明。 chAva 和 chFin 的安全性，以及 chAva 的活性，可由此、定理4和定理5，以及引理 15 得出。

接下来，我们需要证明，在 $\max(\mathsf{GST},\mathsf{GAT}) + O(\Delta)$ 之后，chFin 具有活性，其确认时间为 $O(\kappa)$。令 t 为满足 $\mathsf{propose}(t) \geq \max(\mathsf{GST},\mathsf{GAT}) + \Delta$ 的任意时隙。以很高的概率，存在一个时隙 $t_p \in [t, t+\kappa)$，使得 t_p 和 $t_p + 1$ 都由诚实提议者提出（引理 1）。根据定理 6，交易池$^{\mathsf{propose}(t)} \subseteq$交易池$^{\text{提议}(t_p)} \subseteq \mathsf{chFin}^{\mathsf{fconf}(t_p+3)_i}$，对于任何在 $\mathsf{fconf}(t_p+3)$ 中诚实的验证者均成立；显然，$\mathsf{fconf}(t_p+3) - \mathsf{propose}(t) \in O(\kappa)$。

最后，我们需要证明，对于任意轮次 r 和验证者 $v_i \in H_r$，$\mathsf{chFin}_i^r \preceq \mathsf{chAva}_i^r$。代码行 13 显然确保了当 r 是投票轮次时这一点成立。注意，chAva 或 chFin 可能被修改的其他唯一时机是快速确认轮次。接下来，令 r 为任意一个快速确认轮次。代码行 22 至 23 确保 $\mathcal{GJ}(\mathcal{V}_i^r)$。ch $\preceq \mathsf{chAva}_i^r$。根据算法 3，$\mathcal{GJ}(\mathcal{V}_i^r) \geq \mathcal{GF}(\mathcal{V}_i^r)$，结合引理 8 和假设 $f < \frac{n}{3}$，可推出 $\mathsf{chFin} \preceq \mathcal{GJ}(\mathcal{V}_i^r)$。ch $\preceq \mathsf{chAva}_i^r$。 □