互联网计算机的共识

简·卡梅尼什，马努车手，蒂莫·汉克，维克多·舒普，多米尼克·威廉姆斯

DFINITY 基础设施

2021 年 04 月 16 日

摘要

我们提出了原子广播的互联网计算机共识(ICC)协议家族（即共识），它支撑了拜占庭互联网计算机的容错复制状态机。ICC 协议是基于领导者的协议，它们假设有部分同步，并与区块链完全集成。领导者在每一轮比赛中都有可能发生变化。这些协议非常简单和健壮：在任何一轮领导人腐败（这本身发生的概率小于 1/3），每个 ICC 协议将有效地允许另一方接管领导这一轮，很少麻烦，将协议提前到下一轮。与许多其他协议不同，没有复杂的子协议（如 PBFT 中的“视图更改”）或未指定的子协议（如 HotStuff 中的“起搏器”）。此外，与许多其他协议(如 PBFT 和 HotStuff)不同，可靠地向各方传播块的任务是协议不可分割的部分，而不是留给其他未指定的子协议。国际刑事法院协议享有的额外属性(就像 PBFT 和 HotStuff，不像其他属性，如终端薄荷)是乐观的响应能力，这意味着当领导者是诚实的，协议将以实际网络延迟的速度进行，而不是网络延迟的一些上限。我们提出了三种不同的协议（每个协议还有不同的小变化）。其中一个协议(ICC1)被设计成与点对点八卦子层集成，这减少了在领导者为传播大块而造成的瓶颈，这是所有基于领导者的协议，如 PBFT 和 HotStuff，必须解决的问题，但通常不能解决。我们的协议 ICC2 通过用一个低通信可靠的广播子协议（这可能是独立的）来代替八卦子层来解决同样的问题。

1、产品简介

拜占庭容错(BFT)是计算系统在忍受部分部件的任意（即拜占庭）故障，同时仍能整体正常运行的能力。实现 BFT 的一种方法是通过状态机复制[Sch90]：系统的逻辑跨多个机器进行复制，每台机器都保持状态，并通过执行一系列命令来更新其状态。为了确保无故障的机器最终处于相同的状态，它们必须各自确定地执行相同的命令序列。这是通过使用一个针对原子广播的协议来实现的。

在一个原子广播协议中，我们有 n 个政党，其中一些是诚实的（并且是遵循协议的)，其中一些是腐败的(并且可能是任意行为的）。

粗略地说，这样的原子广播协议允许诚实的各方以一致的方式调度一系列命令，以便每个诚实的各方以相同的顺序调度相同的命令。

每个方都接收各种命令作为输入—这些输入随着时间的推移逐渐接收,而不是一次接收。可能要求命令满足某种类型的有效性条件，各方可以在本地验证。这些细节是特定于应用程序的，将不会进一步讨论。

每个方输出有序的命令顺序—这些输出是增量生成的，而不是一次生成。

任何安全的原子广播协议的一个关键的安全属性都是安全，这意味着各方输出相同的命令序列。注意，在任何给定的时间点，一方可以比其他协议进一步，因此这个条件意味着在任何时间点，如果一方输出一个序列 s，另一个输出一个序列 s，那么 s 必须是 s 的前缀，反之亦然。

任何安全的原子广播协议的另一个关键特性都是活力。人们可以考虑到一些不同的活力概念。在一个概念中，要求每个诚实方的输出队列都随着时间的推移以“合理的速度”（相对于网络的速度）增长。这种活力的概念相当弱，因为它不排除某些各方将其输入命令无限期地忽略的可能性。在另一个更强大的活力概念中，要求是，如果“足够多的”各方在某个时间点收到某个特定的命令作为输入，那么该命令将“不太晚”出现在所有诚实各方的输出队列中。当然，即使是这个定义，没有精确定义“足够多”和“不多”，也是不完整的。

互联网计算机共识(ICC)系列协议。在本文中，我们提出了一系列对应于互联网计算机中使用的原子广播协议。首先近似，因特网计算机是互通信的复制状态机的动态集合：一个复制状态机上的原子广播的命令要么来自其他复制状态机收到的消息，要么来自外部客户端。我们实际上提出了三个特定的协议，ICC0、ICC1 和 ICC2。协议 ICC0 是在互联网计算机上实际使用的协议的一个简化版本，但易于呈现和分析，是本文大部分的主要重点。协议 ICC1 最接近建模互联网计算机中使用的协议的版本，只比 ICC0 稍微多一点。协议 ICC2 有点超出了 ICC1，并使用了目前没有在互联网计算机中使用的技术。我们强调，ICC 协议是完全指定的（它们不依赖于未指定的、非标准的组件），非常简单(相当详细的描述很容易适合在单个页面上)，并且健壮(在面对拜占庭攻击时性能优雅地下降）。

在设计和分析任何原子广播协议时，对腐败方的性质和数量以及网络的可靠性的某些假设是至关重要的。在本文中，我们将假设双方最多是腐败的，可能行为任意，由对手完全协调。当然，这包括那些只是“崩溃”的政党。然而，我们确实假定对手在执行协议开始时选择了静态腐败的各方。

关于网络，通常有一些不同的假设：

- 在一个极端情况下，我们可以假设网络是同步的，这意味着从诚实方发送到诚实方的所有消息都在已知的时间内到达。
- 在另一个极端，人们可以假设网络是异步的，这意味着消息可以任意延迟。

在这两个极端之间，可以进行各种部分同步假设[DLS88]。对于我们这里的分析，我们需要的部分同步假设的类型是网络在相对较短的时间间隔内是同步的。

无论我们是假设是一个异步的还是部分同步的网络，我们都将假设从一个诚实的一方发送到另一方的每一条消息最终都将被传递。

就像许多原子广播协议一样，每个 ICC 协议都是基于区块链的。随着协议的进行，生长块树，从特殊的特殊“起源块”开始。树中的每个非生成块都包含（除其他外）一个有效负载，由一系列命令和树中块的父级的哈希组成。诚实的政党对这棵树有一个一致的观点：虽然每个政党可能对这棵树有一个不同的，部分的视图，但所有的政党都有一个对同一树的视图。此外，随着协议的进行，此树中总是有一个提交块的路径。同样，诚实的各方对这条路径有一致的观点：虽然各方可能对这条路径有不同的部分看法，但所有各方都有相同路径的观点。沿此路径的块的有效负载中的命令是各方输出的命令。

该协议分几轮形式进行。在协议的第 k 轮中，一个或多个深度 k 块被添加到树中。也就是说，在圆 k 中添加的块总是与根正好在 k 的距离上。在每一轮中，都使用一个随机信标来生成 n 个方的随机排列，以便给每个方分配一个秩。级别最低的政党是那一轮选举的领袖。当领导者是诚实的，网络是同步的，领导者会提出一个块将添加到树中。如果领导者不诚实或者网络异步的，其他级别较高的方也可以提出块，并将他们的块添加到树中。在任何情况下，协议的逻辑给予领导者提出的块最高优先级。

我们证明：

- 每一个 ICC 协议都在这种部分同步的假设下提供了活力。非常粗略地说，每当网络保持同步一段时间，无论当事人在当时，如果领导人是诚实的，只有领导人的块将被添加到树的块，和所有的节点从根块将承诺。
- 每个 ICC 协议都提供了安全的，即使是在异步设置中。

在 ICC 协议的最基本版本中，部分同步假设中的通信延迟绑定△bnd 是协议规范中的一个显式参数。与许多这样的协议一样，ICC 协议很容易被修改，以自适应地适应一个未知的通信延迟边界。然而，在这方面必须有所小心，我们要详细地讨论这个问题。

我们还分析了每个 ICC 协议的消息复杂的复杂性。消息复杂度被定义为在任何一轮——中所有诚实各方发送的消息的总数，因此一方广播消息对消息复杂度贡献了 n 的术语。在最坏的情况下，消息的复杂度是 O(n³)。然而，我们证明了在网络同步的任何一轮过程中，预期的消息复杂度都是 O(n²)-事实上，它是 O(n²)，可能性巨大。这里的概率是关于那一轮的随机信标而取的。

当然，消息的复杂性本身并不能说明整个通信复杂性的故事：消息的大小很重要，通信模式也很重要。在每个协议 ICC0 和 ICC1 中的每一个中，在任何一轮中，每个诚实的一方在最坏情况下广播 O(n)消息，其中每个消息都是签名、签名共享（用于阈值或多签名方案）或者块。签名和签名共享通常非常小（几十个字节），而块可能非常大（块的有效负载通常为几兆字节）。如果网络在这一轮是同步的，每个诚实的一方都以压倒性的可能性广播这样的消息（小大小大）。此外，所有诚实方广播的不同块的总数通常是 O（1)——也就是说，诚实方通常都广播相同的块(或少数不同块中的一个）。该属性与互联网计算机对这些广播的实现交互得很好，后者使用一个点对点的八卦子层来完成[DGH+87]。正如我们将讨论的，协议 ICC1 被明确地设计来很好地协调这个点对点的八卦子层（即使协议的逻辑可以很容易地独立于这个子层来理解）。

协议 ICC2 的结构与 ICC1 协议非常相同；然而，它不是依赖点对点八卦子层来有效地传播大块，而是利用基于擦除代码的子协议来实现这一点。假设块的大小为 S，并且 S=Q(nlogn 人)，其中签名（和哈列）的长度为 O（人），ICC2 中每一轮传输的比特总数为 O(S)（假设网络在这一轮中是同步的）。

我们还分析了国际通信法庭协议的交互吞吐量和延迟。在系统的稳定状态下，领导者诚实的，网络延迟受 5<△bnd 的限制，协议 ICC0 和 ICC1 将每 25 个单位的时间完成一轮一次。也就是说，倒数吞吐量是 25。这些协议的延迟，即，领导者提出块和各方提交块的经过时间为 35。对于协议 ICC2，互易吞吐量为 35，延迟为 45。绑定的 5 可能比网络延迟绑定的△要小得多无编号作为部分同步假设（用于确保活性）的基础。特别是，国际商会协议享有被称为乐观响应[PS18]的特性，这意味着协议将在领导者诚实的情况中以网络允许的速度运行。对于任意一轮，如果领导者不诚实或 5>△bnd，这一轮将以压倒性的时间完成 O(ˆbnd+5)。

1.1、各项相关工作

原子广播问题是所谓的共识问题的一个特例。面对任意失败达成共识被[LSP82]表述为拜占庭一般问题。[PSL80]给出了同步通信模型中的第一个解决方案。

在异步通信模型中，结果表明，没有一个确定性协议可以解决共识问题。尽管有这个负面的结果，这个问题还是可以通过概率协议来解决。第一个这样的协议是由[Ben83]给出的，他还证明了弹性约束 t<n/3 在异步设置中是最优的。如[CKS05、CKPS01]所示，最近在[MXC+16、DRZ18、GLT+20]中得到了显著的改进。

尽管最近在异步设置方面取得了进展,但在部分同步设置中可以获得更有效的共识协议。此设置的目标是在不做任何同步假设的情况下保证安全，并且只依赖网络同步的周期来保证活力。[DLS88]给出了部分同步设置中的第一个共识协议。在这个设置中，第一个真正实际的协议是众所周知的 PBFT 协议[CL99，CL02]，这是一个用于原子广播和状态机复制的协议。

PBFT 分轮进行。在每一轮比赛中，一个指定的领导通过向各方广播一批命令来提出一批命令。接下来是两个全面的通信步骤，以实际提交到批处理。在正常运作下，领导将继续担任其角色达多轮。然而，如果有足够多的各方确定该协议没有及时取得进展，他们将触发一个视图更改操作，这将安装一个新的领导人，并清除旧领导人留下的任何混乱。

尽管它对该领域产生了深远的影响，但它在几个方面留下了一些改进的空间。

1. 视图更改子协议是一个相对复杂和昂贵的过程。
2. 领导负责向各方分发该批次。这就造成了两个问题。
   (a) 首先，如果批量非常大，领导者就会成为通信复杂性方面的瓶颈。
   (b) 第二，如果领导人腐败，他可能无法向各方传播一批产品。事实上，一个腐败的领导人（在其他腐败政党的帮助下）可以很容易地推动协议推进任意数量的回合，并让诚实的政党的子集落后，没有任何与这些回合相对应的批次。没有对这些滞后方如何赶上的细节进行详细说明，除了说这样的一方可以从另一方获得任何丢失的批次。虽然这当然是真的，但这个想法的天真的实现使攻击者很容易进一步提高通信的复杂性，通过让许多腐败的方从许多诚实的方请求丢失批次——因此,人们可能不仅仅是广播批次的领导者,O(n)诚实的方每一轮传输 O(n)腐败方。
3. 每一轮最后两步的全对全通信模式也会导致较高的通信复杂度。然而，如果批次相对于 n 非常大，则不需要是这样——在这种情况下，批次的传播仍然是通信复杂性的主要因素。

协议的通信复杂性传统上被定义为所有诚实各方传输的比特总数。在像 PBFT 这样的协议中，这通常是在每轮的基础上测量的。

通过消除全面的通信步骤，大量降低了 PBFT 的通信复杂性[RC05，GAG+19，AMN+20]。然而，正如[SDV19]的经验所证明的那样，这种努力可能是错误的：在提高吞吐量和延迟方面，重要的不是通信复杂性，而是通信瓶颈。也就是说，相关度量不是各方传输的比特总数，而是任何一方传输的最大比特数。这样的经验发现当然对网络的特征很敏感。在[SDV19]中，该网络是一个全球广域网，这是我们对这项工作最感兴趣的设置。正如[SDV19]中所报道的那样，是大批量的传播在领导地位上造成了通信瓶颈，而不是只涉及较小对象的全面通信步骤。事实上，[SDV19]认为，诸如[RC05、GAG+19、AMN+20]中的方法只会加剧领导者的瓶颈。

最近也有关于消除 PBFT 的复杂的视图更改子协议的工作，例如 HotStuff[AMN+20]和帐篷薄荷[BKM18]。与 PBFT 一样，这两个协议都是基于领导者的；然而，它们并不依赖于一个复杂而昂贵的视图更改子协议，实际上可能每一轮轮换一个新的领导者。与 PBFT 不同的是，这两种协议都是基于区块链的协议(虽然 PBFT 可以在区块链的上下文中使用，但它不需要使用)。

HotStuff 消除了 PBFT 的全面通信步骤。此外，HotStuff（实际上是“链式”HotStuff，它是 HotSthff 的流水线版本）提高了 PBFT 的吞吐量，将互惠吞吐量从 35 减少到 25，其中 5 是网络延迟。和 PBFT 一样，HotStuff 反应乐观（当领导者诚实时，它以网络允许的最快）。但是，请注意，HotStuff 的延迟（领导者提出块和提交之间的经过时间）从 35 增加到 65。与 PBFT 一样，HotStuff 依赖领导者来传播块（即批），就像 PBFT 一样，这可能成为通信瓶颈，没有明确的机制来确保块在领导者腐败时可靠传播。此外，虽然 HotStuff 并不依赖于“视图更改”子协议，但它仍然依赖于一些被称为“起搏器”子协议的东西。虽然起搏器子协议的任务没有视图改变子协议那么繁重，但它仍然是不平凡的，在[AMN 中没有指定 20].ÃLibraBFT（即 DiemBFT）[Lib20]实现了一个起搏器子协议，但该子协议重新引入了 HotStuff 想要消除的全面通信模式。最近，已经提出了通信复杂性更好的起搏器协议[NBMS19、NK20、BCG20]。请注意，这些建议都没有涉及可靠和有效的块或批的分发，只涉及各方从一轮移动到下一轮时的同步。

薄荷依赖点对点八卦子层进行沟通。这样做的一个优点是，与 PBFT 和 HotStuff 等协议不同，由领导者提出的块的可靠传播被内置于协议中。此外，一个设计良好的八卦子层可以显著减少领导者的通信瓶颈——当然，这可能以增加互惠吞吐量和延迟为代价，因为通过八卦子层传播消息可能会通过底层物理网络跳几跳。此外，与 HotStuff 不同，肌腱薄荷不依赖任何未指定的组件（如 HotStuff 中的“起搏器”）。薄荷的一个缺点是，不像 PBFT 和 HotStuff，它反应乐观。这可能是一个问题，因为为了保证活力，一个人通常必须选择一个可能明显大于实际网络延迟 5 的网络延迟上限，并且在网络薄荷中，每一轮都需要时间 O(Abnd)，即使领导者是诚实的。

MirBFT[SDV19]是 PBFT 的一个有趣的变体，其中 PBFT 的许多实例同时运行。这样做的动机是为了缓解在普通 PBFT 中在领导者身上观察到的瓶颈。由于 MirBFT 依赖于 PBFT，它也使用了同样复杂而昂贵的视图更改子协议——，然而，正如在[SDV19]中所指出的那样，除了 PBFT 之外的其他协议也可以在他们的框架中使用。让许多各方同时提出批处理会带来新的挑战，其中之一是防止命令的重复，这可以否定吞吐量上的任何改进。在[SDV19]中给出了这个问题的一个解决方案。

Algorand[GHM+17]是一个可证明区块链共识的系统，但其核心是一个原子广播协议。和天德薄荷一样，它是基于一个八卦子层，块的传播被内置在协议中。就像嫩薄荷一样，它的反应并不乐观。与这里讨论的所有其他协议不同，它依赖于一个（非常弱的）同步假设来保证安全性。和国际商会的协议一样，阿尔戈兰德也使用了一些类似于随机信标的方法来对当事人进行排名，但如何使用这些排名的基本逻辑是完全不同的。

我们现在强调了 ICC 协议系列的主要特性，以及它们与上面讨论的一些协议的关联。

- 国际商会的协议是非常简单的，而且完全是独立的。没有复杂的子协议（类似于 PBFT 中的视图更改），也没有未指明的子协议（如 HotStuff 中的起搏器，或可靠的批/块传播，如 PBFT 和 HotStuff 中）。

- 如前所述，国际商会的协议明确地处理了块的传播问题。与帐篷薄荷和阿尔戈兰德一样，协议 ICC1 也被设计成与点对点的八卦子层集成。如上所述，这样的八卦子层可以减少领导者的沟通瓶颈。

  协议 ICC2 而不是一个八卦子层，而是依赖于一个可靠广播的子协议，该协议使用擦除代码来降低整体通信复杂性和领导者的通信瓶颈。这种可靠的广播协议在[CT05]中被引入，并且以前在[MXC+16]中的原子广播环境中使用。我们提出了一种新的擦除编码可靠的广播子协议，具有比[CT05]更好的延迟，并在与 ICC2 协议的集成中利用了更强大的特性。

- 就像 PBFT 和 HotStuff 一样，与帐篷薄荷和奥尔戈兰德不同，所有的 ICC 协议都有乐观的响应。协议 ICC0 和 ICC1 实现了一个 25 的互惠吞吐量和一个 35 的延迟（当领导者是诚实的且网络是同步的）。对于协议 ICC2，这些数字分别增加到 35 和 45。

- 与 PBFT 一样，但与 HotStuff 不同的是，ICC 协议使用了签名和签名共享的全面传输。然而，国际通信法庭协议面向数据块相当大的设置，因此对全对全传输的通信复杂性的贡献通常不是瓶颈。相反，通信瓶颈是块本身的传播，ICC1 协议通过使用八卦子层来缓解，而 ICC2 协议通过使用擦除编码的可靠广播子协议来缓解。

- 与上面讨论的所有协议不同，对于 ICC 协议，每一轮，至少有一个块被添加到块树中，其中一个块最终将成为提交块链的一部分。这确保了总体吞吐量保持相当稳定，即使是在异步时期或领导者腐败的回合中也是如此。也就是说，在与腐败领导人的一轮谈判中，领导人提出的障碍可能没有领导人诚实那么有用；例如，在一个极端，一个腐败的领导人总是可以提出一个空块。然而，如果一个领导者在这方面一直表现不佳，互联网计算机提供了重新配置协议参与者集的机制（这里没有讨论），通过这个机制可以删除这样的领导者。

强有力的共识。我们注意到，国际商会协议的简单设计也确保了它们在拜占庭式故障实际发生时可以相当优雅地降解。正如[CWA+09]中指出的，最近的共识工作大多集中在在没有失败的“乐观情况”中提高性能，因此产生的协议非常脆弱，并且在发生故障时可能实际上无法使用。例如，[CWA+09]表明，在某些类型的（相当简单的）拜占庭行为下，现有的 PBFT 实现的吞吐量会下降到零。本文[CWA+09]主张鲁棒共识，在最优条件下，为了确保当某些方面实际损坏（但仍然假设网络是同步的）时，合理的性能，会部分牺牲峰值性能。国际刑事法院协议确实是健壮的[CWA+09]：在任何一轮的领导者是腐败的(这本身发生的概率小于 1/3)，每个国际刑事法院协议将有效地允许另一方接管这一轮的领导者，几乎没有麻烦，将协议及时推进到下一轮。在这种情况下，唯一的性能下降是，而不是完成这一轮的时间 O（5），其中 5 是实际的网络延迟，这一轮将完成(以压倒性的概率）的时间 O(Abnd)，其中△bnd 之 5 是部分同步假设（用于确保活力）所基于的网络延迟边界。

国际刑事法庭协议的初步版本。请注意，这里提出的协议与[HMW18]或[AMNR18]中所讨论的协议非常不同。特别是，与这里提出的协议不同，在[HMW18，AMNR18]（1）中的初步协议

只有在同步设置中保证安全，（2)没有乐观地响应，而(3）有潜在的无限的通信复杂性。

1.2 论文其余部分的概述

在第 2 节中，我们回顾了协议所需要的加密原语。在第 3 节中，我们提出了协议 ICCO。在第 4 节中，我们对 ICCO 进行了详细分析。第 5 节介绍了 ICCO 的几种变体。1.和 2ICC 协议在第 5.2 和 5.3 节中详细讨论。

2 个加密原语

2.1 抗碰撞的散列功能

我们的协议使用一个哈希函数 H，这个哈希函数被假设是抗碰撞的，这意味着找到两个哈希到相同值的不同的输入是不可行的；也就是说，用 x=x 找到输入 x，X，但是 H(x)=H(x 是不可行的[2]).

2.2 数字签名

我们的协议使用的数字签名方案是一个标准意义上的安全方案，即在自适应选择的消息攻击中创建一个存在的伪造是不可行的。

2.3 阈值签名

A(t、h、n)阈值签名方案是指 n 个方使用公钥/密钥对、公钥以及所有 n 个方的公钥以及全局公钥进行初始化的方案。

- 有一个签名算法，给定一方的密钥和消息 m，在 m 上生成一个签名共享。
- 还有一种签名共享验证算法，给定当事方的公钥，以及消息 m 和签名共享 ss，确定 ss 在给定的公钥下在 m 上是否是有效的签名共享。

  要求正确生成的签名共享始终有效。
- 有一个签名共享组合算法，给给定消息 m 上不同各方的有效签名共享，组合这些签名共享在 m 上形成签名。
- 有一个签名验证算法，给定全局验证密钥，以及签名 a 和消息 m，确定 a 是否是 m 上的有效签名。

  如果组合算法组合了来自不同各方的有效签名共享，那么（概率巨大）结果是一个有效的签名

  在 m。

我们说，如果一个有效的对手是不可行的，以赢得下面的游戏，这样的方案是安全的。

- 对手首先选择 t“腐败”政党的子集。让我们称其余的政党为“诚实”。
- 然后，挑战者生成所有的关键材料，给对手所有的公钥，以及腐败政党的密钥。
- 对手会进行一系列的签名查询。在每个这样的查询中，对手都会指定一个消息和一个

诚实的一方。挑战者在指定的消息上响应该方的签名共享。

- 在游戏结束时，对手输出一个消息 m 和一个签名 a。
- 我们说，如果 a 是 m 上的有效签名，对手将赢得游戏，但对手从少于 h-t 诚实的各方在 m 上获得签名股份。

此类型的阈值签名可以通过多种方式来实现。

(i) 一种方法是简单地使用一个普通的签名方案来生成单个的签名共享，而组合算法只输出一组签名共享。

(ii) 第二种方法是使用多签名，例如 BLS 多签名[BDN18]，其中签名共享是普通的 BLS 签名[BLS01]，可以在单个公钥的集合上合并为新的描述符。

(iii) 第三种方法是使用普通的签名方案，如 BLS，但双方共享密钥（通过沙密尔秘密共享[Sha79]）。

这些方法之间有各种不同的权衡：

- 与(iii)不同，方法(i)和(ii)的优点是不需要任何受信任的设置或分布式密钥生成协议。
- 与(iii)不同，方法(i)和(ii)中标识签名（可以是“bug”，也可以是“特征”）。
- 类型(iii)的签名是唯一的（如果底层非阈值方案的签名是唯一的，则这是 BLS 签名的情况）。类型(i)和(ii)的签名不是唯一的。
- (iii)中的签名（例如，对于 BLS）通常比(i)或(ii)中的签名更紧凑。
- 最后，对于 h>t+1，方法(iii)的安全性可能依赖于一些更强的（尽管仍然是合理的）安全假设。

为了保证原子广播协议的安全性，我们使用这两种方法(ii)和(iii）。

我们使用使用=的方法(ii)用于授权目的：当一方希望授权给定的消息时，它会在消息上广播签名共享。假设该方案是安全的，在一个消息上存在一个有效的签名意味着至少 n-2t 诚实的一方必须已经授权了该消息。

我们使用方法(iii)来构建一个随机信标。为此，我们需要唯一的签名(这是 BLS 提供的)。随机信标是一系列 Rq、Ri，R2 的值。值 Ro 是所有各方都已知的固定初始值。对于 k=1,2,……，值 Rk 是 Rk_i 中的阈值签名。当一方有 Rk-1 并希望生成 Rk 时，它会在消息 Rk_i 上广播其签名共享。如果 t+1 诚实的各方总共做同样的事情，他们可以每个人构建值 Rk。然而，假设阈值签名方案是安全的，除非至少有一个诚实的一方贡献了一个签名共享，否则值 Rk 不能被构造，实际上是 R 的散列 $_k$ 将与随机字符串无法区分（如果我们将哈希函数建模为“随机谕”[BR93]）。

3 协议 ICC0

在本节中，我们详细介绍了我们关于原子广播的协议 ICC0。

3.1 前期

时间间隔符号。在本文中，我们使用符号来表示集合，…。

我们有一个聚会，派，…，P_n。我们假定最多没有一个腐败的政党。我们将假设一个静态的腐败模型，其中对手在协议执行一开始就决定哪些方腐败。我们通常会假设拜占庭式的失败，即一个腐败的一方可以任意行事，而所有腐败的一方都是由对手协调的。然而，我们有时会考虑较弱的腐败形式，如崩溃失败，其中腐败的政党根本没有反应。我们也将有机会考虑一种称为一致失败的中间腐败形式，这有点具体的协议，但通常意味着腐败一方的行为并非明显不正确（见第 4.7 节）。

我们的协议执行的唯一通信类型是广播，其中一方向各方发送相同的消息(这将适用于协议 ICC0 和 ICC1，但不适用于 ICC2)。这不是一个安全的广播：如果发送人腐败，不能保证诚实的方会收到相同的消息或任何消息；如果发送人是诚实的，那么所有诚实的方最终都会收到消息。我们通常假设消息传递的精确调度是由对手决定的。

各方都有一个池，其中包含从所有各方（包括其自身）接收到的所有消息的集合。正如我们所描述的协议一样，从来没有从池中删除任何消息。虽然可以优化协议，以便丢弃不再相关的消息，但我们在这里不讨论这些细节。此外，复制状态机的实际实现通常会包含某种检查点和垃圾收集机制，类似于 PBFT[CL99]中的机制。同样，我们也不讨论这些细节。虽然互联网计算机实现使用“八卦网络”在各方之间传输消息，但除上述那些假设外，我们不会对底层网络做出任何假设。

各方将使用一些密钥以及自己和其他各方的公钥。对于某些加密原语，双方的密钥相互关联，必须由受信任的密方或安全的分布式密钥生成协议来设置。

其中一些密钥是用于数字签名的，用于进行身份验证消息。不需要其他消息身份验证机制。

3.2 组件

我们的协议使用的是：

- 一个抗碰撞的散列函数 H；
- 签名方案，其中每个诚实方都有一个密钥，并提供各方的公钥；
- （t、n-n、n）阈值签名方案的实例公证人，其中每个诚实方都有一个密钥，并提供该实例的所有公钥材料；
- （t、n-n、n）阈值签名方案的 Sfinai 实例，其中每个诚实方都有一个密钥，并提供该实例的所有公钥材料；
- (t、t+1，n)阈值签名方案的一个实例信标，其中每个诚实方都有一个密钥，并提供该实例的所有公钥材料；这用于实现随机信标，该随机信标如第 2 章所述；因此，该方

案需要提供唯一的签名。

3.3 对协议的高级描述

该协议分几轮形式进行。在每一轮会议中，双方都可以建议将一个块添加到一个块树中。在这里，一个块树是一个定向的有根的树。除了根节点外，树中的每个节点都是一个块，它由

- 圆数（也是树中的块的深度），
- 提出这个街区的政党的指数，
- 块树中块的父级的哈列（使用抗碰撞哈列函数 H），
- 数据块的有效载荷。

根本身是一个特殊的块，用根表示。

块的有效负载的详细信息依赖于应用程序。在原子广播的环境中，如第 1 节所述，有效载荷自然会由一个有效载荷组成

或更多已输入给提议该块的一方的命令。此外，在构造所提议的块的有效负载时，一方总是在扩展块树中的特定路径，并且可以考虑该路径中已经存在的块中的有效负载（例如，为了避免重复命令）。这是一个重要的特性。

要提出一个块，一方必须用数字签名对块签名(使用 $S_{自动}h$)。要向块树添加建议块，必须使用阈值签名方案公证程序进行公证。此外，经公证的区块可以使用 SfinaiSfinai。

还使用随机信标，使用阈值签名方案信信标实现，以便在每一轮中，显示随机信标的下一个值。给定一轮中随机信标的值决定了双方上的排列 n，它分配一个唯一的秩 0，…给每个方，n-1。在密码假设下，每一轮的排列 n 有效上是一个随机排列，独立于前一轮使用的排列，并且不受腐败政党的选择影响（这假设一个静态地破坏政党的对手）。

第 0 级的政党是那一轮选举的领袖。虽然协议优先考虑回合领导人提出的块，但其他方也可以提出块。特别是，如果领导人腐败或暂时切断网络，其他方提出的区块将经过公证并可能最终确定。

正如我们将看到的，在某些密码假设下，但没有任何同步假设，它可以保证在每一轮的 k>1：

P1：至少添加一个公证的深度 k，

P2：如果确定公证的深度 k，则没有其他公证的深度 k。

此外，我们还有：

P3：如果网络在任何诚实方首次进入 k 轮的短时间间隔内同步，并且 k 轮的领导是诚实的，那么 k 轮的领导提出的块将最终确定。

属性 P1 确保协议不会死锁，因为树每一轮都在增长。

属性 P2 的使用方式如下。假设某一方看到一个最终的深度 k 块 B，并让块树中从根到 B

的路径。假设某个政党（相同或另一个）看到一个最终的深度 k 块 B，其中 k'>k，让块树中从根到 B 的路径。那么属性 P2 意味着路径 p 必须是路径 p‘的前缀：如果不是，那么在深度 k 处将会有两个不同的公证块，与属性 P2 相矛盾。

在原子广播的上下文中，上述参数表明，当一方看到一个最终确定的块 B 时，它可以安全地按照该顺序将从根到 B 的路径上的块的有效载荷中的命令附加到其输出队列中。

属性 P3 保证了在部分同步假设下的强活性概念。实际上，如果命令至少以第 k 轮输入到 n-t 方，那么至少 n-2t>n/3 诚实方将收到该命令作为输入，因此，概率为>1/3，第 k 方的领导者可以确保该命令在其提议的块中，如果同步假设为第 k 轮，每个诚实方将以第 k 轮输出该命令（一旦传递所有相关消息就行）。[1]

此外，由于属性 P1，即使网络在许多轮中保持异步，即使在短时间内同步，所有中间轮的有效负载的命令也将由所有诚实的各方输出。因此，即使网络只是间歇性同步的，系统也将保持恒定的吞吐量。然而，如果干预回合中的障碍是只有由腐败的政党提出的，那么来自这些干预回合的命令可能并没有多大用处。

3.4 块块

我们现在给出更多关于块的细节。有一个特殊的圆 0 块根。

对于 k>1，圆 k 块 B 是该形式的一个元组

（块、k、a、phash、有效载荷）。 (1)

在这里，a 代表 P 党的索引，谁提出了这个块，帕希是哈希函数 H 的输出，而有效负载是特定于应用程序的内容。

根据诚实方 Q 池中的其他数据，我们将一个块分类为真实的、有效的、公证的或最终确定的(Q)，这取决于 Q 池中的其他数据。特殊的词根总是存在于 Q 的池中，并且总是被认为是真实、有效、公证和最终确定的（对于 Q）。

让 k>1，让 B 成为一个圆 k 块 B，就像 Q 池中的（1）一样。

- *如果 Q 池中存在 B 的身份验证器，则 B 称为真实（对于 Q）。*

  B 的身份验证器是元组（身份验证器、k、a、H(B)、a），其中 a 是 P*方提供的有效（身份验证器、k、a、H(B)）签名

- *如果 B 为真实(针对 Q)，如果对 Q 池中公证(针对 Q)的某轮(k-1)块进行 phash=H(BP)，则称为有效（对于 Q）。Bp 被称为 B 的父级，我们说 B 扩展了 Bp。*

  请注意，根据 H 的抗碰撞特性，我们可以假设 B 的父级是唯一的。

  还要注意，可能存在特定于应用程序的属性才能认为 B 有效。

- *如果 B 有效，并且在 Q 的池中有 B 有公证，则称为公证(Q)。*

  B 的公证是元组（公证、k、a、H(B)、a），其中 a 是有效的公证签名（公证、k、a、H(B)）。B 的公证份额是元组（公证份额、k、a、H(B)、ns、月），其中 ns 是 P^方有效的公证签名份额（公证、k、a、H（B））。

[1] 这就假定有效载荷的大小没有限制。如果有一个限制，但诚实的政党优先考虑旧的命令，一个相当强烈的活力概念仍然会得到满足。

如果 B 有效(Q)并且 Q 池中存在 B 的最终确定，则称为最终确定(Q)。

B 的定型是一个元组（定型、k、a、H(B)、a），其中 a 是有效的 Sfinai 签名（定型、k、a、H(B)）。B 的最终共享是元组（最终共享、k、a、H(B)、fs、月），其中 fs 是一方有效的 Sfinai 签名共享（最终共享、k、a、H（B））。

在下面，root 作为自己的身份验证、公证和最终确定。

注意，如果一方在其池中有一个有效的圆 k 块 B，那么还有块根=Bq，Bi，…Bk=B 形成区块链，意味着 B《是 B《+的父级，对于 i=0，…，k-1，以及 Bi 的身份验证，…，Bk 和公证 ^{B}i，$^{B}k-i$.

3.5 本协议的详细信息

该协议由两个同时运行的子协议组成：树形构建子协议和最终化子协议。

P 方的树形构建子协议 。如图 1 所示。树形构建子协议使用了两个延迟功能：

- Ap_r操作：TR>q 是用来根据提出者的排名来延迟提出一个块的。它应该是一个非递减的函数。
- 摘要：>q 用于基于提案人的排名延迟产生公证份额。它应该是一个非递减的函数。

我们对协议的表示和分析将基于这些一般的延迟函数。展望未来，对于活力，唯一的要求是 25+Aprop（0）<Antry（1），其中 5 是在那轮过程中网络延迟的绑定。但是，为了更好地控制协议的通信复杂性，建议这些功能的实现如下：

$$\begin{matrix} {}^{A}\text{支柱(r)}:=2\ \text{编号；} \\ \text{安特里(r)：=2Abndr+£。} \end{matrix} \qquad (2)$$

网络延迟受 5<限制的轮将满足上述活动要求，参数 e 为“调速器”一，可以设置为零，但设置为非零值将防止协议运行“太快”。

我们提醒读者，由我们的协议执行的唯一通信类型是广播，其中一方向各方发送相同的消息。此外，这个广播被假定是不安全的：从腐败方收到消息的一方不能确定其他方是否会收到相同的消息。

在此协议描述中，一方等待其池包含满足特定条件的消息。如前所述，此池包含从任何一方接收到的所有消息集（包括本身广播的消息），并且不会从池中删除任何消息（尽管协议的适当优化版本会这样做）。

广播第一轮随机信标 对于每一轮的 k=1,2,3.。。：

等待 t+1 共享的圆 k 随机信标计算圆 k 随机信标(它定义了圆 k 的排列 n)广播圆 k 共享的随机信标+1

让 r_e 是 P 的等级 。

N—0　　*P° 已经播放了公证股票的一组区块* D—0　　*被 P 取消参赛资格的队伍集* 。 *做的，假的提议，假的到时钟的()*

```
重复操作
    等待以下一个：
        (a)  经过公证的圆形 B 街区，
                或一套完整的公证股份为部分有效
                但未经公证的圆形 B 区块：
            //完成这轮将公证股份合并成公证 B，必要时广播公证 B
            已真
            如果 NC，那么广播 B 的最终共享
        (b)  未提出和时钟()>到+△道具(rme)：
            //提出一个提议的块-true 选择一个经过公证的圆（k-1）块 Bp 有效载荷-获取有效载荷(Bp)创建一个新的圆 k 块 B=（块、k、a、H、Bp）、有效载荷）广播 B、B 的身份验证器，并为 B 的父母进行公证
        (c)  一个有效的秩级块 B，这样
                B 牛 N，r 屯 D，时钟()>到+△ntry(r)，没有有效排名 R**的 r*e[r]\DB*：
            //回声方块 B
            //和广播公证份额，或取消资格，如果 r=rme
                广播 B，B 的认证人，和公证 B 的父母，如果 N 中的某些块有排名 r
                然后，为 B 广播一个公证份额，直到完成
```

图 1：ICC0：P 方的树状结构构建子协议。

在每一轮的树构建子协议中，作为初步步骤，P。将从等待 t+1 共享的阈值签名，用于计算该一轮的随机信标开始。之后，它将计算圆 k+1 的随机信标，并立即广播圆 k+1 的随机信标份额。这是一种用于最小化延迟的“流水线”逻辑——因此，在任何诚实的一方完成 k 回合之前，对手可能就已经知道了 k 轮+1 的随机信标，但这不是问题。

如前所述，圆 k 的随机信标决定了各方的排列 n，它分配了一个唯一的秩 -各各一方。排名为 0 的政党是
称为圆 k 块的领袖。对于圆 k 块 B，我们定义等级ˆ(B)为提出 B 的政党的等级。

现在第 k 轮就正式开始了。对于这一轮，这是 P 党。将保留一组已经播放过公证股份的区块，以及一组被取消资格的排名。如果一个级别被取消资格，这意味着该级别的政党已经被发现为这一轮提出两个不同的区块。

这轮比赛将结束给 P 党。一旦它在其池中发现一个经过公证的轮块 B，或者在其池中找到一套有效但未经公证的轮块 B 的全套非公证股票。在后一种情况下，P 方认为。将公证股份合并为 B 的公证，在任何一种情况下，都将在 B 播出公证。此外，如果是 P 方。除了 B 以外，它自己还没有在任何街区播放公证份额，它将在 B 上播放最终的份额。

甲方。当△道具(rme)时间单位从一轮开始时（更准确的说，当它执行步骤时钟()），将提出自己的块。这种延迟对安全或活力并不重要，而是为了防止所有诚实的各方用他们自己的建议淹没网络。特别是，当领导诚实时，适当选择延迟功能，网络同步时，只有领导才能播放自己的区块。在提出它自己的区块时，P。必须首先在其池中选择一个经过公证的圆(k-1)块以进行扩展。总会有这样的块，因为前一轮只有在有这样的块（或 k=1 和 Bp=根）时才会结束。可能有不止一个这样的公证区块，在这种情况下，选择哪一个并不重要。接下来，P。必须计算一个有效负载。在图 1 中，这是通过调用函数 getPayload(Bp)来实现的，该函数的细节与应用程序有关，但请注意，它可能依赖于 Bp 和以 Bp 结束的整个块链（例如，为了

避免重复命令）。最后，通过构建了一个块 B 来提出，P 方 广播 B，B 的认证人，以及为 B 的父母 Bp 的公证。

最后，也是 P 方 将响应其池中排名 r 的有效轮 B 块(i)还没有为 B 广播公证份额，(ii)没有取消排名 r，(iii)自一轮开始以来至少通过△ntry(t)时间单位，(iv)池中没有“更好的”块。在这里，一个“更好的”块将是一个有效的圆 k 块，其排名小于 r，但尚未被取消资格。如果这些条件成立，则 P 方表示 是否要执行以下操作：

它“呼应”方块 B，即它广播 B、B 的认证人和 B 的父母的公证；

kmax-由 P 最终确定的 0 个最大回合 。
重复操作
请等待：
(i) 使用 k>kmax 完成的圆 k 块 B，或
(ii) 一些有效但未最终确定的圆>块 B 的完整最终共享最大值：
提交以 B 结束的链中最后一个 k-kmax 块，将最终共享合并为 B 的最终确定，如有必要，广播 B 的最终确定
输出链中以 B 结束的最后一个 k-kmax 块的有效载荷 ^{k}最大值：$<^{k}$
永远永远的

图 2：P 方的最终化子协议 。

此外，它广播 B 的公证份额，除非它已经为同一级别 r 的不同区块广播了公证份额，在这种情况下，它取消了 r 级的资格。

请注意，P 将回声 B，即使它已经广播了同一级别的另一个街区的公证份额。这是为了确保所有其他诚实的政党也有机会取消 r 级排名的资格。但是，请注意，P 将最多响应任何给定秩的 2 个块。

P 方的最终化子协议 如图 2 所示。甲方 跟踪它看到最终确定块的最后一轮 kmax。每当它看到(i)其池中确定的圆块 B，或者(ii)其池中一些有效但未确定的圆块 B 的 n-t 最终共享的完整集合，其中 k>kkax，它进行如下。如果(ii)，它将定型股份合并为 B 上的定型，如果两种情况(i)或(ii)，它将在 B 上广播定型。此外，它将按顺序输出区块链中最后一个 k-kmax 块的有效负载。

我们的正式执行模型是，当执行“等待”语句时，执行将暂停（如果必要），直到消息到达或出现时间条件，从而满足“等待”中的条件之一。发生这种情况时，将执行相应的子句（如果满足多个条件，则任意选择一个）。我们将假设，在执行此条款时，该方的池没有被修改。

4 协议 ICCO 的分析

4.1、初步试验结果

引理 1（终止）。在每一轮的过程中，每个诚实的一方都将执行主循环 O(n)次。

证明文件。当触发主循环中“等待”语句中的条件(a)时，循环终止。条件(b)最多可执行一

次。所以，要考虑一下条件(c)。这最多可以触发 2n 次。要看到这一点，注意每次触发，我们要么

- 在 N 中添加一个块，不同于 N 中最多可能发生的任何秩，或者
- 或者我们在集合 D 中添加一个新的等级，这最多也可能发生在 n 次时间(最多是 t 次，假设最多不是腐败的政党和安全的签名，因为每个这样的等级都属于一个腐败的政党)。

□

引理 1 给出了每轮 O 的消息复杂度的最坏情况限制(n^3)。然后，在部分同步假设下，我们将证明每轮 O(n 的期望消息复杂度的一个界 ^{2}).

引理 2（限制于公证数量）。假设最多包含 3 个损坏方、安全签名和防碰撞。在每一轮，对于每一方，最多可以有一个街区的提议进行公证。特别是，在任何一轮中，最多 n 个公证块出现在系统中的任何地方。

证明文件。在每一轮选举中，一个诚实的政党最多会为任何特定等级的一个街区产生一个公证份额。由于公证的阈值是 n-t 和 t<n/3，因此引理遵循一个标准的法定人数交叉参数。

口

引理 3（到处都是块链）。假设最多包含 3 个损坏方、安全签名和防碰撞。假设在某个时间点，一些诚实的政党 P 已经用区块链块 B 的根=Bq，Bi，　　=B.然后，我们有：

(i)块 Bi，…，Bk，以及块 Bi 的认证器，…，B^和块 Bi 的公证…，B^-x 已经被诚实的方播出了。

(ii)特别是，当(i)中的所有区块、认证人和公证都已交付给任何诚实方时，Q，B 是 Q 的有效区块。

证明文件。每当一个诚实的政党广播一个街区 B=Bk，它也广播 Bk 的认证器和 Bk-i 的公证。此外，由于该党有 Bk-i 的公证，一些诚实的一方一定已经为 Bk-i 播放了相应的公证份额，因此一定已经播放了区块 Bk_i。(i)随后进行归纳法。(ii)如下定义。　　□

4.2、主要研究结果

现在我们就证明了对应于第 3.3 节中讨论的属性 P1、P2 和 P3 的主要结果。以下引理对应于属性 P1。

引理 4（死锁自由度）。假设最多包含 3 个损坏方、安全签名和防碰撞。在每一轮中，如果诚实的方通过这一轮广播的所有信息已经传递给所有诚实的方，并且所有的延迟已经过期，那么所有诚实的方都将用一个公证的区块来完成这一轮。

证明文件。假设这个定理是假的。然后有一些有限的执行让系统在一个状态，所有消息由诚实的政党通过这一轮已经传递给所有诚实的政党，所有延迟已经过期，但有一些诚实的政党

没有完成一轮公证块。选择第一轮这样的比赛。

所以我们知道每个诚实的政党都进入了这一轮。此外，由于所有诚实方都完成了前一轮，他们都广播了这一轮的随机信标的共享，因此所有诚实方都收到了足够多的共享来重建这一轮的随机信标。因此，所有诚实的政党不仅都进入了这一轮，而且也进入了这一轮的主要循环。

我们也知道，没有一个诚实的政党通过一个公证的区块来完成这一轮，否则，如果一些诚实的政党这样做，它会广播公证，所有诚实的政党都会通过公证来完成这一轮。

让我们用政党的等级来命名，所以 P 党(r)是 r 级的政党。对于任何一个诚实的聚会，P(r)，让我们来说，一个排名 s 对 P 有好处(s)如果是 P(r)P 是否完全提出了一个有效的数据块(s)对于这一轮的池；否则，我们说排名 s 对 P 不好(r)。如果有什么好的排名为 P(r)，将 s*(r)定义为它们中最小的；否则，定义 s*(r)：=n。

*权利要求 1。让 P(r)*做一个诚实的聚会。对于每个<*(r)，或者

- P(r)没有 P 提出的有效块(s)在这一轮的游泳池里，或者
- P(r)P 提出了多个有效的块(s)在这一轮的游泳池中，P 侦)已经播放了两个这样的街区，和 P(r)已取消等级资格。

证明文件。这从协议的逻辑中是很清楚的。回想一下，我们假设 P(r)尚未完成这一轮比赛。对于每个<*(r)，如果 P 侦)有多个由 P 提出的有效块(s)在它的池中，然后（通过归纳）对于任何秩的 t<，P 侦)都没有 P 提出的块(t)或者会取消 t 级资格，因此 P 侦)将广播 P 提出的两个块(s)，从而取消了不符合资格的等级 s。

权利要求 2。让我们成为最不平等的诚实政党的地位吧。然后是 s*(h)<h。

证明文件。我们想证明 s*(h)<h 或 s*(h)=h，或等价地，s*(h)>h 意味着 s*(h)=h。所以假设 s*(h)>h。然后根据协议和权利要求 1 的逻辑，并假设 P(h)还没有完成这一轮比赛，P(h)会提出并播放自己的块，这意味着排名 h 对 P 有利吗(h)，意思是 s*(h)=h。

权利要求 3。假设 P(7)是一个诚实的一方，并假设 s：=*(r)<n。那么 P(7)有一个由 P 提出的唯一有效块(s)在其池中，P(r)广播该区块和相应的公证份额。

证明文件。这来自权利要求 1、协议的逻辑以及 P(7)尚未完成这一轮的假设。

权利要求 4。对于任何两个诚实的政党，P(7)和 P，我们有 s*(r)=*(r).

证明文件。让我们来说：=s*(r)和 s：=s*(r ‘)。为了自矛盾，假设，比如说，s< ‘。根据权利要求 3，P 侦)具有由 P 提出的唯一有效方块 B(s)在其池中和 P 中(s)广播方块 B。因此（根据引理 3），B 也是 P（*）池中的有效块。自

对于<，我们看到秩 s 对 P（*）不好，因此，根据定义，P（”）必须有 P 提出的另一个有效的块 B(s)在其存储池中。此外，根据权利要求 1，P（*）将广播 P(1)提出的两个有效块，并且（同样，通过引则 3)这两个块将在 P 侦池中作为有效块出现）。这与秩 s 对 P(7)有利的假设相矛盾。

让我们成为所有诚实的各方中 s(r)的共同价值(r)，如先前的索赔所保证的那样。根据权利要求 2，s*<h。

权利要求 5。有一个排名的 B*，这样对于每个诚实的一方来说，B*在 P 的池中作为一个有效的块，至少 B*的公证份额也在 P 的池中。

证明文件。根据权利要求 3,每个诚实方 P(7)在其池中具有唯一的有效等级 s*的块 B*(r)，并广播 B*(r)，以及相应的公证份额。此外（通过引理 3），每个 B*(r)在每个诚实的参与方的池中都是一个有效的方块。因此，B*(r)=B*（时）为所有诚实的各方 P(7)和 P 印)。设 B 是 B(r)的共同值。此外，每个诚实的一方至少收到了 B 的公证份额*.

最后，根据权利要求 5，我们看到每个诚实的一方都已经完成了协议。一个矛盾。 □

以下引理对应于第 3.3 节中讨论的属性 P2。

引理 5（安全性）。假设最多包含 3 个损坏方、安全签名和防碰撞。在每一轮，如果某个区块最终确定，那么那一轮没有其他区块可以公证。

证明文件。假设</3 方损坏。如果一个区块 B 最终定稿，那么至少 n-t-f 诚实方为 B 发行最终定稿股份，这意味着这些诚实方没有为除了 B 之外的任何区块发行公证股份。如果另一个街区进行公证，这将需要一群不同的诚实方公证股份，这是不可能的。 口

如第 3.3 节中所讨论的，该特性意味着该协议满足安全特性。

接下来，我们将讨论活力。为此，我们正式地陈述了我们的部分同步假设：

定义 1。假设在当时 T，所有诚实的政党都已经开始了。我们说，如果在时间 T+6 之前 T 发送的诚实的消息到达他们的目的地，那么通信网络在时间 T 时是 6 同步的。

以下引理对应于第 3.3 节中讨论的属性 P3。

引理 6（活性假设为部分同步）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) k>1，第一个进入 k 轮的诚实方 P 在 T 启动，所有诚实方已在 T 启动；

(iii) k 轮的领导 Q 是诚实的；

(iv) 通信网络在+和++时有时是+同步的$_{r}$操作（0）；

(*) 2d+^{A}支柱$^{(0)}$V^{A}n 试(1)。

然后，当来自诚实的方的所有信息传递给所有诚实的方时，每个诚实的方都将有 Q 的建议块作为最终的块。

证明文件。当时 T，P 已经在第一轮播出了区块公证…，k-1。这意味着至少 n-2t 不诚实的政党已经播放了所有这些块，因此也分享了第一轮的随机信标，……，k。

因此，在时间之前，根据同步假设和引理 3，对于 i=1，…，k-1，Q 的池中有一个经过公证的 r+i 块，对于 i=1，…k，Q 的池中的 riundi 随机信标超过 t 共享。

因此，Q 将在时间+之前进入+的主循环，并在时间+之前播放自己的+块+$_{r}$操作中的（0）。[2]同样，根据同步假设和引理 3，B 出现在每个诚实方的池中作为一个有效的块，T+2d+Aprop（0）<T+A$_{ntr}$y（1）。这意味着每一个诚实的一方都会为 B 和只为 B 广播公证股票。

由此可见，每当来自诚实方的所有圆形信息被传递给所有诚实方时，每个诚实方都将为 B 广播最终定稿的份额，因此每个诚实方都将有 B 作为一个最终定稿的区块。 口

[2]在这里，我们假设计算时间相对于网络延迟可以忽略不计。

注意，引理 6 的条件(v)函数将满足（2）中定义的延迟函数。

4.3 预期的消息复杂性和延迟

在给定的 k 轮中，我们将主要的诚实党定义为最低的诚实党。 *(k)* .

在这一轮中排名，我们将 N 定义为在这一轮中被呼应的任何块的最高排名，我们将 nnecho 定义为在这一轮中任何诚实的一方呼应的任何块中的最高排名。请注意，任何一个 hones 十参与方广播的块数为/0(“ˆ在部分同步下的翼翼
其中一个诚实的政党是ˆˆ(inecho)。丄 he 遵循引理说，在部分同步假设下，我们有 $N^{(k)}$<h，其中 h 是 assu 丄丄丄 ption 轮中主要诚实党的排名，我们有 nn 回声<h，其中 h 是 k 轮中主要诚实党的排名。这意味着 k 轮中的信息复杂性是 0((h+1)n^2)。稍后，我们将使用这个结果来证明每轮的预期消息复杂度是 0(n^2)-事实上，它是 0(n^2)，可能性巨大。

引理 7（假设是部分同步的消息复杂度）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) k>1，第一个进入 k 轮的诚实方已在时间 T 时启动，且所有诚实方均已在时间 T 时启动；

(iii) k 轮主要诚实党 Q 排名 h；

(iv) 通信网络是+同步时 T+和 T+支持(h)；

(^{r}*)* 2d+Ap_{r}操作（^{h})V△ntry(^{h}+ 1)-

然后， ^{h}

证明文件。根据与引理 6 相同的推理，Q 将进入 k 轮，并在时间 T+5+△道具(h)之前播放它自己提出的 k 轮块 B。同样，根据同步假设和引理 3，B 出现在每个诚实方的池中，T+25+△道具(h)<T+△ntry(h+1)。这意味着每一个诚实的政党都不会回应任何排名大于 h 的政党。 □

请注意，当 5<△bnd 时，（2）中定义的延迟函数将始终满足引理 7 的条件(v)。

接下来，我们根据这一轮诚实党的地位，证明了一轮延迟的界限。假设网络在一定时间间隔内是 5 同步，证明了这个结果。然而，与 Lemmas6 和 7 不同，假设 5 和延迟函数△道具和△ntry 之间没有关系。

引理 8（延迟）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) 此时 T，所有诚实方都已经启动，k 是所有诚实方进入的数量最多的一方；

(iii) k 轮主要诚实党 Q 排名 h；

(iv) 通信网络在整个期间始终 5 同步

[T，T+△o(h，5)+2h5]，

其中：

△（h，5）：=最大值（25+△支柱(h)，5+△试(h)）。 (3)

然后所有诚实的政党按时间完成 k 轮

$$\triangle o(h,\ 5)+(2h+1)5. \tag{4}$$

证明文件。根据与引理 6 相同的推理，所有诚实的各方都将进入 k 轮，并通过时间 T+5 到达 k 轮的主循环。[3]此外，Q 将通过 T+5+^道具(h)播放自己的提议块 B。同样，根据同步假设和引理 3，B 作为每个诚实的方的池出现在时间 T+25+△道具(h)。因此，每一个诚实的方都会在 T+△(h，5)之前播放 B 的公证股份，这将使各方在 T+△(h，5)+5 之前完成这一轮，除非一些诚实的方在 T+++△(h，5)之前收到并回应了排名较低的区块。

所以我们假设从现在开始，当 T+△o(h，5)一些诚实的政党已经收到并回应了一个等级的<h。

我们将考虑 i=0 的 T+△o(h，5)+id 的时间点，……，2 小时。在每个时间点，我们将为每个秩 r<h 分配一个状态，即未使用、已使用或已损坏。每个这样的等级将一开始是未使用的。在每个这样的时间点+△+△(h，5)+，我们将改变最多一个秩 r 的状态，使以下条件成立：

- 排名 r 可能从未使用改为使用，如果发生这种情况，至少一个块排名 r 已经收到和呼应一些诚实的政党 T+^(h，5)+i5，所有诚实的政党将有一个块排名 r 池时间 T++^(h，5)+(i+1)5。

- 等级 r 可能从习惯改为被宠坏，如果发生这种情况，至少两个不同的等级 r 块已经被诚实的方通过时间 T+△(h，5)+id，每个诚实的方在时间 T+△(h，5)+池中有两个不同的等级 r 块(i+1)5。

此外，正如我们将看到的，如果没有排名改变状态，那么所有诚实的政党都将在 T+△o(h，5)+(i+1)5 之前完成这一轮。由于状态变化不得超过 2 小时，因此各方将在 T+△o(h，5)+(2h+1)5 之前完成这一轮比赛。

我们假设在时间点 T+△(h，5)，一些诚实的政党已经收到并响应了一个等级<。选择最小的等级，并将其状态从未使用改为使用。

反过来，我们考虑时间点 T+^o(h，5)+i5，对于 i=1，……，2 小时。设 r 是在时间 T+△o(h，5)+(i-1)5 时使用的最小秩，如果没有这样的秩，则设置 r：=h。

现在，假设到 T+^(h，5)+i5，一些诚实的政党已经收到并响应了一个小于 r 的未使用等级；在这种情况下，我们选择最小的秩，并将其状态改为使用。

否则，到 T+△o(h，5)+i5 时，所有诚实的政党已经取消了任何目前小于 r 的等级的资格，并收到并呼应了至少一个等级的 r。

- 如果各方在同一街区播放公证份额，所有诚实的各方将在 T+△o(h，5)+(i+1)5 之前完成这一轮交易。

- 否则，至少有两个不同的等级 r 块进行了公证，我们将 r 的状态改为破坏。

[3]与引理 6 中不同的是，在这个引理中，我们也允许 k=1，但同样的结果在这种情况下也成立。

请注意，引理 8 中的延迟绑定（4）测量第一个诚实方从 k 开始到最后一个诚实方结束 k 之间的时间。如果发生这样的话，它不考虑实际完成块的额外通信延迟。这将在绑定的（4）中添加一个名为 5 的项。绑定（4）更恰当地视为控制协议的吞吐量——块的公证速率，与协议输出有效负载的摊销速率成比例（考虑到有效负载可能包含许多命令）。

我们注意到引理 8 本质上是紧的，因为有一个攻击会导致一个本质上等于引理中的界的延迟。

4.3.1 概率分析

假设 h 是 k 轮中主要的诚实党的排名，我们可以把它看作是一个随机变量。引理 7 和 8 中消息复杂度和延迟的结果取决于 h 的值。假设对手静态地破坏双方（即在协议执行的最开始），或者至少在 k 轮的随机信标显示给对手之前。在这种情况下，对手选择哪些腐败的政党和在第 k 轮的排名函数是独立的。也假设 t<n/3，那么对于 i=1，……，t，

$$\Pr[h>i=-\blacksquare \text{七}—^{1}\ldots\ldots\ldots\ldots\text{七}—^{(i}—D<\text{丄}。 \qquad (5)$$
$$—n\ n—1n—(i—1) \qquad 3\text{厂}$$

当然，对于我来说，我们有 Pr[h>i]=0。特别地，通过尾和公式，我们可以将 h 的期望值限定为

$$E[h]=\pounds \Pr[h>i]<=\pounds 3? = 2.$$
$$i>1 \qquad i>1^{3} \qquad ^{2}$$

由此可见，如果网络在整个 k 轮中保持ˆ同步，我们就有 E[MCho]<2，因此这轮的预期消息复杂度是 O(n^2)。然而，尾部绑定的（5）是更重要的事实。

现在让我们考虑引理 8 的假设下考虑预期延迟。让我们假设延迟的函数为 $A_{ntr}y$ 和 Ap_r 操作被定义为（2）。在这种情况下，我们有

$$\triangle o(h，5)=2A_{b}ndh+6+\text{最大值（e，5），}$$

因此，延迟的（4）是由

$$2（Abnd+5）h \text{ 最大}+25+（e、5），$$

由于 E[h]=1/2，预期的延迟最多是

$$\triangle bnd+\text{最大 } 35+（e，5）。$$

4.4 时间间隔同步下的活动

如上所述，如果网络在第一诚实方进入该一轮的时间点开始的短时间间隔内是 5 同步的，则引理 6 保证了给定一轮的活力。最好能够说，只要网络在足够长的时间间隔内保持 5 同步，那么协议所达到的任何一轮的活动都将保持不变。我们可以通过将引理 6 和引理 8 结合起来来得到这样的结果：

引理 9（假设间隔同步）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) 此时 T，所有诚实方都已经启动，k 是所有诚实方进入的数量最多的一方；

(iii) k 轮的主要诚实党排名 h；

(iv) 通信网络在整个期间始终 5 同步

$$|T，\ T+Ag(^{k}，\ ^{5})\ +\ (2^{h}+\ 2)^{5}+Ap_{r}op（0）]，$$

其中，Ao（h、5）的定义为（3）；

(v) k 轮+1 的领导很诚实：

（vi）25 +Ap$_{r}$操作（0）<$A_{n}t_{r}$y（1）。

然后，当来自诚实的各方的所有轮(k+1)信息被传递给所有诚实的各方时，每个诚实的各方将由 k 轮+1 的领导人在其池中作为一个最终的块。

如果我们进一步扩展同步间隔的长度，我们可以保证以压倒性的概率，在该间隔内的一些圆将最终确定一个诚实的领导提出的块。

4.5 方案的复杂性

在给定的 k 轮中,我们将 $NPROp^{(k)}$ 定义为在 k 轮中提出自己的一个块的诚实方的最高级别。NpOp 的大小决定了在 k 轮中通过网络循环的不同块的数量（至少是由诚实方提出的）。如第 1 节中简要讨论的，边界 NPOP 将有助于控制整体通信的复杂性，特别是当底层广播使用点对点八卦子层实现时。

在这一节中，我们研究在什么假设下 NpOp 仍然是有界的。

引理 10（假设部分同步的提案复杂性）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) k>1，第一个进入 k 轮的诚实方已在时间 T 时启动，且所有诚实方均已在时间 T 时启动；

(iii) k 轮主要诚实党 Q 排名 h；

(iv) 通信网络在整个期间始终 5 同步

$$[T，\ T+Ao(h，\ 5)+2h5]，$$

其中，Ao（h、5）的定义为（3）；

(v) />1 是这样的：

$$\triangle o^{(h，\ 5)}\ +\ ^{(2h+1)5}\ <\ ^{A}{}_{支柱}{}^{(h+}\ /).\qquad(6)$$

然后是 NpOp<h+1。

这个引理很容易从引理 8 跟踪到延迟。假设延迟函数在（2）中被定义为，并且 d<Abnd

和 e<Abnd，这样

$$\triangle o(^{h,}\ d)<2(^{h}+1)A_{\text{无编号}}.$$

因此，如果网络在整个 k 轮过程中都保持 d 同步，那么我们就有了

4.6　对碰撞故障的分析

在只有崩溃故障的常见情况下，我们重新回顾上述一些分析。在这种情况下，如果领先的诚实党有 h 级，那么排名 0 的政党，…，h-1 崩溃。

引理 11（假设部分同步和崩溃故障的活动和消息复杂性）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) k>1，第一个进入 k 轮的诚实方会在时间 T 时这样做，并且所有诚实方都已在时间 T 时启动；

(iii) k 轮主要诚实党 Q 排名 h；

(iv) 通信网络是+同步时 T+和 T+支持(h)；

$$^{(v)2d} + {}^{A}_{\text{支柱}}{}^{(h)} < {}^{A}_{\text{n 试}}{}^{(h\ +\ 1)}-$$

然后，

(a) 当来自诚实方的所有回合信息都传递给所有诚实方时，每个诚实方都将有 Q 的回合提议块作为最终块，以及

(b) 事实上，唯一被诚实的一方回应的是 Q 提出的区块。

请注意，引理 11 的条件(v)将由在（2）中定义的延迟函数来满足。

引理 12（假设崩溃故障的延迟）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) 此时 T，所有诚实方都已经启动，k 是所有诚实方进入的数量最多的一方；

(iii) k 轮主要诚实党 Q 排名 h；

(iv) 通信网络在整个时间间隔内始终是 d 同步的

$$[T，\ T+Ag(h，\ 5)]，$$

其中，Ao（h、5）的定义为（3）；

然后所有诚实的政党按时间完成 k 轮

Ao（h、5)+5- (7)

引理 13（假设部分同步和崩溃故障的方案复杂度）。假设：

(i) 最多有 3 个损坏方，签名安全，哈希函数抗碰撞；

(ii) k>1，第一个进入 k 轮的诚实方会在时间 T 时这样做，并且所有诚实方都已在时间 T 时启动；

(iii) k 轮主要诚实党 Q 排名 h；

(iv) 通信网络在整个期间始终 5 同步

[T，T+Ao(h，5)]，

其中，Ao（h、5）的定义为（3）；

(v) />1 是这样的：

$^{A}0^{(h,\ 5)}$ ＋ $^{5}V^{A}$支柱$^{(h}$+ /). (8)

然后是 NpOp<h+1。

让我们再次检验引理 13 的结果，假设延迟函数被定义为（2），以及 5<Abnd 和 e<Abnd，因此

$^{A}0^{(h,\ 5)}$ ＜ $^{2(h}$ ＋ $^{1)A}$编号。

因此，如果网络在整个 k 轮中保持 5 同步，那么我们就有了

^{N}打印的$^{<h}$+^{1}.

特别地，最多会有两个 k 圆块在网络中循环，即一个 h 块和（可能）一个 h+1 块。

4.7 对一致性故障的分析

考虑一个政党可能腐败，但行为一致的环境，这意味着他们不会发出不一致的建议，从而取消他们的资格。这在协议的变更中特别相关，如第 5.1 节中详细讨论。有了这种变化，最多有 t 轮，任何一方都可以被取消资格。

一个可以改进的结果是在延迟上的引理 8。特别地，延迟绑定的（4）可以简化为的

△o$^{(h,}$ d)+$^{(h}$ ＋ $^{1)}$5, (9)

其中，△□(h，d)被定义为在（3）中。

另一个可以改进的结果是提案的复杂性。具体地说，引理 10)中的条件（6）可以被稍微较弱的条件所代替

△o$^{(h,\ d)+(h+1)d}$V 型△支柱$^{(h+}$ /). (10)

4.8 局部延迟功能

到目前为止，为了简单起见，我们已经假设所有诚实的各方在每一轮都使用相同的延迟函数△prop 和△ntry。这并是不完全现实的，至少有两个原因。

首先，诚实各方的时钟可能以略微不同的速率运行（时钟漂移）。这可以通过对各方具有不同的延迟函数来建模。

第二，如果网络延迟可能随时间而变化，则诚实的各方可能希望根据系统的感知性能，动态地调整其延迟功能，以为此进行调整。

例如，如果一方看到也可能没有最终确定，或者意外地看到几轮的高消息复杂度，它可能会增加其△ntry 延迟函数。在△ntry 的增加之后，一方可能会试图在以后的一轮减少它。对于在（2）中定义的延迟函数，可以通过简单地调整在定义函数△ntry 时使用的△bnd 的值(而不一定要改变在定义函数△道具时使用的△bnd 的值)来进行这些调整。在另一种情况中，如果协议相对于系统中的其他元素运行得太快，一方可以增加（2）中的附加项 e 以减缓协议速度。无论如何，每一方都是在本地做出这些决定的，因此它们最终总是会产生不同的延迟函数。

因此，我们引入了局部延迟函数。具体地说，让我们假设在圆 k 方/Q 使用十 he 延迟函数△和△XATe 阿苏ˆ我十 ha 十为每 个人 k
在圆 k 回合中，党使用延迟函数 z-Aprop 和ˆ△ntry♦vve 假设的 那个 为每个人 k

和/Q 功能△和△不递减，kcQ，功能ˆ-△道具和ˆ△ntry 不递减。

对于每个 k，我们定义

△(ᵏ)-一 T△- 诚实问题） 以及 *一 • 诚实问题）

/、♦ 丄丄丄丄丄丄 i\道具 ♦丄丄丄丄 ddu j a 丄丄 n /\支柱♦ 丄丄丄\prop♦丄丄丄丄 dduj，

同样地，

/\⁽ᵏ⁾：一皿访[八(*⁹)*Q，诚实的，和 ：一 △⁽ᵏ,ᑫ⁾：问：诚实的人
ˆ△ntry*— 每分钟一次* ‘Q 诚实 以及 ˆ△次 *—inax{ˆJntry *
‘Q 诚实的}。

我们可以看到，对于每个 k，每个函数

△%、爲 2p、驾 y 和膏 y

是非减少的。

我们现在重新讨论依赖于这些延迟函数的主要结果。显而易见，为了保持我们的活力和信息复杂性，重要的是，虽然双方在当地可以增加他们的 A_{ntr} 在使用 y 函数时，它们不应该在本地调整其 Aprop 函数。然而，我们仍然可以在本地建模，如果只是适当地考虑时钟漂移。

充满活力。引理 6 仍然正确，但已将条件(v)替换为

2 日五；爲（0）<AnSy(i)。 (11)

同样地，在碰撞故障的设置中，引理 11 的活性结果（部分(a)）仍然正确，并将该引理的条件(v)替换为

$$25+Apx(h)<和？\ y(h+1)。 \tag{12}$$

在保持活力方面，对 Aprop 功能进行任何局部调整都没有任何好处，事实上，局部增加 Aprop 的值只会使（11)和(12）不太可能得到满足。因此，让我们从现在开始假设，没有对代理功能进行本地调整——事实上，从来没有理由使用一侬）
Aprop（0）的值不是 0，所以我们可以假设 Ap_{r} 操作（0）=0。

现在，如果有几轮没有结束，双方 Q 启发式增加到足以弥补大于预期的网络延迟，那么结束将在 k 轮恢复。此外，就活力而言，如果音频太大，因为（11)和(12）仍然会保持。也就是说，如果当地对安特里进行的调整彼此不一致，只要它们都足够大，就不会受到惩罚。

消息的复杂性。引理 7 仍然正确，并将条件(v)替换为

$$25+Ap\&p(h)<\qquad(h+1)。 \tag{13}$$

因此，适用于活动的相同评论也适用于消息的复杂性。就保持消息的复杂性而言，对函数道具进行任何本地调整都没有任何好处，而且我们继续假设没有进行这样的调整。此外，如果各 Q 局部增加函数 Antry，足以补偿大于预期的网络延迟，则消息复杂度应在 k 轮中再次受到限制。此外，就消息复杂度而言，如果对函数的局部调整彼此不一致，只要都足够大，就没有惩罚。

延迟时间。引理 8 仍然正确，△o(h，5)始终替换为

$$\text{压“(W)}^{\cdot}= \text{最大值}(25+\triangle \text{px}(^{\text{h}}\text{，}\ ^{5}+\triangle \text{i}_{1}\text{Sy}(^{\text{h}))}- \tag{14}$$

正是在这里，如果△Itry(h)太大，我们就会受到惩罚。然而，我们强调，无论△<：，，，该协议仍然是有响应性的，即，以网络速度运行，假设

（0）和△Itry(O)是（接近）零（推荐），并且假设 k 轮中的领导者是诚实的。

然而，我们确实注意到，在区间同步假设下，我们在引理 9 中也遭受了同样的惩罚——同步必须保持的时间间隔的长度随着△Itry(h)的增加而增加。

提案的复杂性。引理 10 仍然正确，△o(h，5)的值替换了上面由△碧出版社(h，5)出版，在（14)中定义的路径，不等式(6）替换为

$$\triangle \text{o)}^{(\text{h},\ 5)}\ +\ ^{(2\text{h}\ +\ 1)5}\text{V}\triangle \text{PtOpS+，)。} \tag{15}$$

如上所述，为了保持活力和信息的复杂性，我们不应该在本地进行调整 十 he △ 函数，所以 — △ 对于所有的 k 和 /Q 十 he 条件
局部调整 的 ˆ△支柱 函数，所以假设是ˆ△道具 — ˆ△支柱 对于所有的 k 和 KCQ，该条件
然后，（15）就会变成

$$\text{最大值（5+△p）}_{\text{r}}\text{op(h)，驾\(“+(2h+2)5<△p}_{\text{r}}\textit{操作(h+£)。} \tag{16}$$

然而，这揭示了维护信息复杂性的目标和提案复杂性之间的紧张关系。如果网络意外变慢，则为△1_{1}：，(h)意外大，任何合理小的£可能无法满足条件（16）。

请注意，即使我们只考虑一致的故障或崩溃故障，也存在同样的问题。在崩溃故障的设置中，引理 13 中的条件（16）变为

$$\text{最大值（5+△p）}_{\text{r}}\text{操作(h)，}\qquad\qquad\text{(h))+25<△道具+<△)，} \tag{17}$$

这也可能是不可能满足的任何合理小的 I。

解决这个问题的一个可能的缓解措施是定义超线性增长的延迟函数，而不是像（2)那样线性增长——这将使(17）更有可能满足较小的 t 值。

5 个协议的变更

在本节中，我们将考虑了一些协议的变化及其后果。

5.1.意见不一致的当事人被永久取消资格

正如我们所提出的，如果另一方发现第二一方在一轮中提出了两个不同的区块，那么另一方可能会取消另一方的资格。然而，这种取消资格并不会延续到随后的几轮比赛中。

(c)秩 r 的有效的圆 k 块 B，使 r 牛排名(Dp)，时钟()>到+$A_{\text{n}}\text{t,y(r)}$，并且没有等级(Dp)

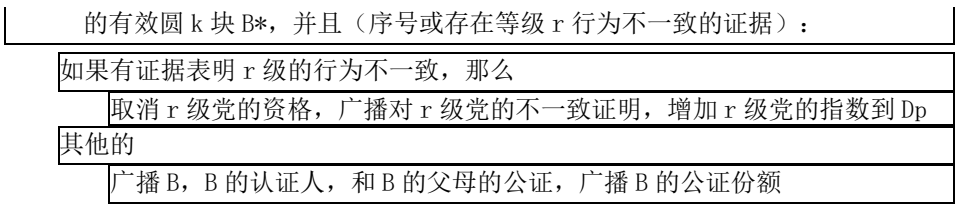

图 3：永久不合格的逻辑

我们可以修改协议，以便如果一方在某一轮中取消一方的资格，而不是广播导致取消资格的第二个块（如我们当前的协议），而是广播一个称为“不一致证明”的特殊消息，证明一方在同一轮中验证了两个不同的块。

回想一下，在第 3.4 节中，我们为一个圆 k 块定义了一个身份验证器

B=（块、k、a、phash、有效载荷）

作为元组（验证器、k、a、H(B)、a），其中 a 是 P 方在（验证器、k、a、H(B)）。

来自 P 的一对不一致的身份验证器。对于圆 k 是一对元组

（身份验证器、k、a、hashi、ai）、（身份验证器、k、a、a 希 2、a2）、

这样 ai 是 P 方有效的(身份验证器、k、a、hash 签名。对于 i=1、2 和哈希 1=哈希 2。

对 P 的不一致证明。因为圆 k 是这个形式的一个元组

（不一致，k、a、hashi、ai、哈希 2、a2），

其中 a 是 P 方有效的(身份验证器、k、a、hashi)签名。对于 i=1、2 和哈希 1=哈希 2。假设索斯签名是安全的，就不能构建针对诚实一方的不一致证据。请注意，使用这个新的身份验证器语法，可以在不知道块本身的情况下验证这种不一致性验证。

对树构建子协议进行了如下修改。首先，不是保持一组不合格的 D，每轮，每个党 P。保持一套被永久取消资格的当事人的民主党人。该集的 Dp 在协议的最开头被初始化为 0。

其次，修改了图 1 的主回路中的“等待”条件(c)的逻辑，如图 3 所示。

这里，排名(Dp)表示当前一轮政党的排名组。此外，如果鸟是当前轮中排名 r 的一方，那么排名 r 的行为不一致的证据也意味着

(i) 来自当前一轮的一对不一致的身份验证器，或

(ii) 当前一轮或前一轮针对乌的不一致证明；

如果是情况(ii)，P。广播其不一致的证据，而以防万一(i)，P。构造和广播一个新的不一致证明对 P^，源自不一致的身份验证器。

对于这种变化，我们在上面证明的所有引理都没有变化，并且没有显著的缺点。在这种变化中，最多有 t 轮任何一方可能被取消资格，因此，就长期制度的执行而言，只有协议的履行，各方的行为一致才重要。我们已经在第 4.7 节中讨论了在这个设置中如何提高延迟和

建议的复杂性。

5.1.1 临时取消资格的替代实施方案

即使一个人不希望永久取消各方的资格，使用不一致证明和这个替代语法可以更实际，因为它不需要广播整个第二块来说服其他方一方的行为不一致。事实上，我们最初的临时取消资格的原始协议可以通过简单地改变不一致行为的证据的定义来更有效地实现

(i) 对于当前一轮，来自 P° 的一对不一致的身份验证器，或

(ii) 当前一轮对 P° 的不一致证明，

5.2 协议 ICC1：收紧提案条件

在提出这种变化，我们称之为 ICC1 议定书时，我们将假设我们已经纳入了第 5.1 节中引入的“永久取消资格”规则。但是，也可以使用“临时取消资格”，进行一些非常小的变化（下文讨论）。

回顾那个方 P„会提出自己的区块吗 r操作（rme)的时间单位从这轮比赛一开始就已经过去了（实际上，从它获得当前这轮比赛的随机信标的时间开始）。在这种变化中，我们将收紧 P 党所处的条件 „将提出自己的块，这样，如果其池中有一个明显的“更好的”块，它将“阻止”这样做。这里，“更好”的块是指比 P 低的块 „这还没有被取消比赛资格。

但是，请注意，如果某一方因为在自己的池中看到了一个“更好的”区块，那么就不能保证任何其他诚实的一方在短期内看到了这个“更好的”区块。因此，该党不会简单地“退缩”，而是会呼应这个“更好”的障碍。更准确地说，在回合结束之前，它将和最小排名 r 的任何“更好的”块相呼应，但只有在自回合开始以来已经经过了道具(r)时间单位。

这种变化的动机是为了在底层通信层是点对点流言层的设置（如在互联网计算机的实现中）中尽量减少带宽利用率。在这种设置中，真正想要最小化的是通过网络循环的不同块的总数（可能非常大），而不是尽量减少传统定义的消息复杂性(或通信复杂性）。此外，这种变化对崩溃故障和一致的故障设置进行了优化（如果协议永久取消了不一致方的资格，或者是使用协议栈中更高的其他机制，这是最相关的）。这个协议的变化最类似于目前为互联网计算机设计的协议。我们将简要讨论此协议变化与以下点对点八卦层之间的更多联系（见第 5.2.1 节）。

此变化的细节见图 4。注意，为了简化协议的逻辑，我们在本节中假定△支持(r)<A_{ntr}对于所有的资源。在设置中，如果局部延迟函数（如第 4.8 节），这个不等式只需要局部保持(ie △(k,Q)(r) V △(k,Q)(r) 对于每个 k 和 Q) 此处为^) 是永久的集合吗
本地的（即，/△道具（/） _< ^△ntry（‘）对于每个 k 和 KQ). 丄 丄 ere， *Dp* 是 per 丄丄丄 anently 的集合吗
被取消资格的政党，和排名(Dp)是这些政党的排名的集合，由当前一轮的随机信标决定。“不一致行为的证据”和“不一致证据”概念的细节可参见第 5.1 节。请注意，如果我们希望实施临时取消资格，而不是永久取消资格，我们可以简单地(a)在每一轮开始时将 DP 重置为 0，以及(b)定义第 5.1.1 节中所述的“不一致行为的证据”和“不一致证明”的概念。

5.2.1 更多关于点对点的八卦层

如果不讨论太多的细节，我们就可以进一步谈谈这个协议的变化和底层的点对点八卦层之间的关系。

八卦沟通有时被描述为满足以下特性：

- 如果任何诚实的一方发送消息，那么最终（或在部分同步假设下），每个诚实的一方都会收到消息。
- 如果任何诚实的一方收到了一个“有趣的”的消息，那么最终（或在部分同步的假设下），每个诚实的一方都收到了这个消息。

显然，要使这种通信完全保持限制，构成“有趣”信息的标准必须有相当的限制。事实上，如果所有消息都“有趣”，那么通信的复杂性可能会变得完全无限。

在本节的协议变更中，就点对点八卦层而言，一方认为一个块很“有趣”，如果它是未公证的块，并愿意告诉其他各方。因此，每轮通过八卦层循环的“有趣”块数量将是有限的，正如我们下面讨论的，在部分同步假设下将非常小（至少对于崩溃和一致的失败是这样）。除了区块之外，还有其他类型的信息，如公证共享、公证等。一般来说，这些类型的消息是

广播第一轮随机信标
Dp—0 *被 P 取消资格的当事人组。*
对于每一轮的 k=1,2,3...。:
等待 t+1 共享的圆形 k 随机信标
计算圆 k 随机信标（它定义了圆 k 的排列 n），为圆 k+1 广播随机信标的份额

让 r 我的名字是 P 的等级。
B—0 *已由 P 广播的一组数据块。*
N—0 *P 已播放公证股票的一组区块。*
做了，假了，提议了，假了，到时钟()
重复操作
等待以下一个：
(a) 经过公证的圆形 B 街区，
或一套完整的公证股份为部分有效
但未经公证的圆形 B 区块：
完成整轮将公证股份合并为 B 公证，必要时广播 B 公证
已真
如果 NC，那么广播 B 的最终共享
(b) 秩为 r 的一个有效的圆 k 块 B，这样
排名(Dp)，时钟()>到+Ap,操作(r)，对于 r*G[r]\级(Dp)没有有效的圆 k 块 B*，并且（BGB
或存在 r 级行为不一致的证据）：
如果有证据表明 r 级的行为不一致，那么
取消 r 级党的资格，广播对 r 级党的不一致证明，增加 r 级党的指数到 Dp
其他的
回声块 B 广播 B，B 的认证器，以及 B 的父母 B-BU 的公证
(c) 未提议，时钟()>到+△道具(rme)，和
等级(Dp)中没有有效的圆 k 块 B：
建议选择一个块，选择经过公证的圆（k-1）块 Bp 有效载荷 get 有效载荷(Bp)

	创建一个新的圆块 B=（块、k、a、H(Bp)、有效载荷）广播 B、B 的身份验证器，以及 B 的父 B-BU 提议的公证-true
(d)	等级 r(DP)的块 BGB\N，这样
	时钟()>到+△ntry(r)， 等级 r*G[r]\等级(DP)没有有效的圆 k 块 B*，也没有对等级 r 的行为不一致的证据：
	为 B 产生公证份额 ，广播 B 的公证份额
直到完成为止	

图 4：ICC1：P 方的树状结构构建子协议。如果该块本身是有趣的，那么它将被认为是“有趣的^”。

除了可能提出和公证数据块的各方之外，该网络还可能包含额外的“中继”节点，以帮助传播“有趣的”消息。这样的中继节点本质上运行协议就好像它们有排名 oo——特别是，它们从不提出自己的块，也不会生成任何公证、终结或随机信标共享。然而，这些节点可以选择回声块，使用与普通参与者相同的规则。此外，这种“中继”节点也可以运行复制状态机副本。

对于大型消息，如果一个节点有一个它认为“有趣”的消息，它会向网络中的直接节点发送一个“广告”。这样的“广告”是对实际信息的一个非常紧凑的描述。例如，对于一个块，这样的“广告”可能包含关于块的所有信息，除了有效载荷本身，它通常相当大。如果对等看到“广告”，并认为相应消息会“有趣”（例如，它自身回声的块），则对等会请求消息本身。请求对等方也可以对这些请求进行优先级，基于这些信息似乎更“有趣”（在给定的一轮中，排名较低的块通常更“有趣”）。这种启发式方法通常会产生良好的带宽利用率，同时仍然在合理的假设下保持活力（当然，安全性永远不会受到威胁）。

5.2.2 分析方法

回顾一下我们所定义的内容

- $N_{prop}^{(k)}$ 是排名最高的诚实党，在 k 轮提出自己的区块，和
- N 窯。是所有政党中排名最高的街区。

让我们来定义一下

$$N\hat{}x：=最大值（NPkOp，N/小时。）。$$

这只是在任何情况下由任何诚实的政党在 k 轮播放的最高排名块。

让我们来探讨 ICC1 提出的更严格的区块提出条件的后果。Lemmas1-6（包括安全性和活性）仍然有效。此外，就崩溃故障设置中的活力（但不是消息复杂性）而言，部分(a)引理 11 的更强的结果成立。请注意，这些关于活性的结果也适用于本地延迟函数设置中，如第 4.8 节所述。就像我们在那个部分讨论的，如果每个方局部增加函数 $A_{n}t_{r}y$ 足够（但不局部调整△支柱），将实现活力。此外，局部延迟函数 Antry 是否不同步并不重要——最重要的是它们每个函数都足够大。

如上所述，在带宽利用率方面，我们主要感兴趣的是在崩溃故障和一致的故障设置中限制通过网络循环的不同块的数量。所以在这个设置中，这相当于边界 N^nlx 的值。

妨碍了碰撞和一致的故障设置。在崩溃故障或一致的失败设置中，如果 h 是 k 轮的排名，(k)在这轮网络中是^同步的，然后我们将提供 Nmax<h+1

$$\&+\triangle\text{道具}(^{h})\text{V}\triangle\text{道具}(\text{h+£})。 \tag{18}$$

证明性的草图。如果第一个诚实的政党在时间 T 进入这一轮，那么 h 排名的政党 Q 将在 T+5 之前进入，并将在时间之前发布自己的提案或在 T+5+△之前发布较低排名的提案(h)。这将到达所有诚实的派对之前，T+25+△道具(h)。由于没有被取消资格，该提案将阻止任何诚实的政党播放任何排名高于 h 的提案。 口

如果我们按照（2）和 5<△bnd 中线性定义延迟函数，这个条件将符合/：=1。在碰撞故障设置中，这意味着只有一个块在循环中。在一致故障设置中，循环可能有高达+1 块。

如果网络延迟绑定了$\triangle_{\text{无编号}}$在（2)中太小了，那么我们将需要选择一个更大的值/来满足(18），因此 N^x 可能会更大。这可以通过超线性增长，而不是线性增长的延迟函数来缓解。但是，请注意，由于△支柱出现在不等式（18）的两侧，如第 4.8 节中的局部调整延迟函数对减轻这一点并没有多大帮助。

让我们将 N 慝 x 上的这个绑定与原始协议进行比较。对于该协议，请按顺序执行
为了确保 NM/x<$h^{(k)}$+1，我们需要

$$\text{25+}\triangle\text{道具(h)+}\triangle\triangle\text{ntry（h+£），最大值（25+}\triangle\text{道具(h)，5+}\triangle\text{ntry(h)）+（h+1）5+}\triangle\triangle\text{propS+。} \tag{19}$$

在一致的故障设置中，以及

$$\text{25+\^{}道具(h)<+\^{}ntry(h+<+\^{})，最大值(25+}\triangle\text{道具(h)，5+}\triangle\text{ntry(h)）+5+<+\^{}+}\triangle\text{propS+。} \tag{20}$$

在崩溃故障设置中。我们可以看到，条件（18)严格弱于条件(19)和(20）。此外，如果我们局部调整延迟函数△ntry，条件（19)和(20）就会变得更难以满足。因此，就边界 N^x 而言，在一致的故障和崩溃故障设置中，这个变化更优越。

边界 N（拜占庭故障设置中的/x。让我们也考虑一下拜占庭式的背景。在这个设置中，虽然数量 N(/x 调节该协议变化的消息复杂复杂性，但它不再调节通过网络循环的不同块的数量(因为现在可能有给定秩的几个块）。在这种情况下，如果 h 是在 k 轮中主要的诚实党的排名，并且网络是 5 同步的(k)
在这一轮过程中，然后我们将提供 NM/x<h+/

$$^{\text{最大值（25 个）}}\text{+}\triangle\text{支柱}^{(\text{h}),\ 5}\text{+}\triangle\text{+}\triangle^{-1))}\ +\ ^{2\text{h5}}\text{<}\triangle\text{提出+，)。} \tag{21}$$

*证明性的草图。*如果第一个诚实的政党在时间 T 进入一轮，那么 h 排名的政党 Q 将在时间 T+5 之前进入，并将在时间之前发布自己的提案或在 T+5+△道具(h)发布较低排名的提案(h)。

有两个情况需要考虑。在第一种情况下，Q 在时间前广播它自己的建议，T+5+△道具(h)，其论证如下。

在第二种情况下，Q 广播一个较低级别的提议，并阻止播放自己的建议。我们想提前争论一下这一点

△0：=
/---------- ----------
最大（25+△支柱(h)，5+△（h-1））+2h5，

要么所有诚实的政党都已经完成了这一轮，要么已经收到了 Q 的提案，这将阻止任何诚实的政党播放任何排名高于 h 的提案。

现在，在 T+△之前，所有诚实的各方都收到了 Q 的排名较低的提案，并准备发行公证份额。我们提出了一个改变状态的论点，如在引理 8 的证明中，其中每个等级 0，…，h-1 被分配了一个未使用、被使用或被破坏的状态。然而，在我们在 T+△0 开始状态变化参数之前，我们可能已经初始化 Q 广播的第一个提案的排名使用的状态，而不是未使用的状态。根据在引理 8 中使用的相同的推理，在最多 2h-1 的状态改变后，所以根据时间 T+△0+(2h-1)5，也可以

(a) 所有诚实的当事人都在同一街区发行了公证份额，或

(b) Q 将会播放它自己的提案。

所以在 T+△0+2h5 之前，所有诚实的方都已经完成了这一轮，要么收到了 Q 的建议。 口

让我们将这个边界与我们原来的原始协议进行比较。对于该协议，然后在(k)中为了确保 Nmax<h+&我们需要

25+ˆ道具(h)+ˆˆntry(h+和最大值（25+） △支柱(h)，5+△次(h)) +
(2h + 1)5<△提出+，)。 (22)

条件（22)意味着条件(21）。因此，就边界 Nˆx 而言，在拜占庭的设置中，这种变化也是优越的。但是，请注意，如果函数△ntry 被局部调整，原始协议和这个变体在拜占庭失败设置中的边界 NˆX 都会有困难。

延迟时间。我们所有的延迟结果保持没有变化，在崩溃，一致，和拜占庭失败设置。

证明性的草图。对于崩溃故障，没有任何更改。

对于拜占庭式的失败，我们争论如下。如果第一个诚实的政党进入按时间 T，则 h 级的政党 Q 将在时间 T+5 之前进入，并将在时间之前播放自己的提案或在 T+5+△道具之前播放排名较低的提案(h)。

有两个情况需要考虑。在第一种情况下，Q 通过时间 T+5+广告来播放它自己的提案 ₜop(h)，参数如下对原始协议的分析。

正如我们在上面分析 Nˆx 时所争论的，按时间计算

△0：=
/---------- ---------- S
T+最大值（25+支柱(h)，5+A_n试试（h-1））+5，

在<2h-1，Q 要么播放自己的提案，要么所有诚实的方将在同一区块播放公证份额，我们将更改+1 状态。所以，按时间计算

T+最大（（s+1）5+A0.5+安距(h)）

所有诚实的方要么完成这一轮，要么准备好对 Q 的提议进行公证。在后一种情况下，他们将对 Q 的提案进行公证，除非有更多的状态变更，其中最多 2 小时（+1）。所以，按时间计算

T+最大（+1）5+A0,5+试验(h)）+（2 小时—（+1））5，

各方将对 Q 的提案进行公证，各方将按时完成一轮

最大值（（s+1）5+A0、T+5+间距(h)）+（2h-s）5，

哪一个最多可以验证的是延迟绑定的（4）。

对于持续的失败，我们论证如下。如果第一个诚实的政党进入时间 T，则 h 级的政党 Q 将通过时间 T+5 进入，并将播放自己的提案或通过时间 T+5+Aprop(h)播放排名较低的提案。

有两个情况需要考虑。在第一种情况下，Q 通过时间 T+5+Aprop(h)广播其自己的建议，其论证如下，与对原始协议的分析一样。

使用类似于上面使用的论点，并使用当事人按时间一致的事实

△o：=

/---------------------- ------------------------ s

T + 最大值（25 个） + A 支柱(h)，5 + A n 试(h—1)) +(h—1)5,

所有诚实的政党都将在同一街区播放公证份额。所以所有的政党都按时完成了这一轮比赛

T+A0+h5，

可以验证哪一个小于延迟绑定的（9）。 □

5.3 协议 ICC2：减少通信瓶颈

在第 5.2 节的协议 ICC1 中，大部分协议，特别是主循环中“等待”的状态(b)逻辑，可以看作是并行运行许多简单的广播子协议，但启动时间交错——r 级的一方 m 只有在从一轮开始以来<（<）时间单位通过时，e 才愿意参加<级的广播子协议。

然而，所使用的底层广播子协议本身并不能保证一致性，这就是为什么原子广播协议本身需要有额外的逻辑来取消不一致各方的资格。此外，底层的广播子协议在通信复杂性方面并不是最优的——至少在传统的通信复杂性度量下是这样，我们简单地计算所有诚实各方发送的位数。如果块的大小为 S，则忽略签名和签名共享贡献的通信复杂度（我们可能对于非常大的块），通信复杂度为 O(n2S)。

在本节中，我们概述了如何用可靠的广播协议替换底层广播子协议，这将消除取消不一致各方的资格。此外，我们使用的特定可靠的广播协议的通信复杂度只有 O(NS)——这假设 S=Q(nlogn 人)，并且签名和哈希的长度为 O（人）。

这种方法的一个缺点是，每轮数据的最佳情况下的延迟会有所增加，但只有一个网络延迟 5。这种方法唯一的另一个缺点是，该协议的计算复杂度要高一些。

5.3.1 可靠的广播

在一个可靠的广播协议中，我们有当事人 Pi，P_n，其中 t<n/3 可能已损坏，我们假设有一个异步通信网络。在每轮 k 中，各有 4 方 P_a 可以启动协议以可靠地向各方广播一个消息 m：我

们说 P„*从 P 启动可靠的广播„在 k 轮与消息 m*。在每一轮 k 中，每个方 Pˆ可以启动协议以可靠地接收来自 P 的一些消息„，*对于=月，在 k 轮中：我们说 Pˆ从 P 启动可靠的广播„在 k 轮中*。在每一轮 k 中，从 P 启动可靠广播的 3„在 k 轮（我们可能有一个=月）以后可能会输出一个单个消息 m：我们说 Pˆ可靠地从 P 接收 m„*在 k 轮中*。

该协议的关键属性如下，它应适用于每轮 k：

有效性：如果是一个诚实的一方 P„从 P 启动可靠的广播„在 k 轮与消息 m，然后每个诚实的一方最终可靠地从 P 接收 m„在 k 轮中。

一致性：如果两个诚实的一方分别可靠地从一方收到 m 和 m ‘„在 k 轮中，然后是 m=m ‘。

诚信：如果一个诚实的一方可靠地收到了来自一个诚实的一方 P 的信息„在 k 轮中，然后是 P„以前开始的可靠广播„在第 k 轮对 m。

总数：如果一个诚实的一方可靠地收到了来自 P 的消息„在 k 轮中，最后每个诚实的政党都这样做了。

对于有效性和整体属性，“最终”指所有诚实各方已启动协议（作为广播员或接收器），以便从 P 进行可靠广播„在 k 轮中，由与这个可靠的广播实例相关联的诚实的各方产生的所有消息都已经传递。

‘由于我们的应用程序，我们认为 k 是一个圆数。更一般地说，k 可以是任何类型的“会话标识符”。

5.3.2 使用擦除码降低通信复杂性

我们草拟了一个基于擦除代码的可靠的广播协议，这是[CT05]中协议的一个变体(在[AKK+02，AKK+07]中使用了类似的技术)。虽然[CT05]中的协议可以看作是来自布拉 cha 的广播协议[Bra87]，但我们在这里呈现的协议可以看作是来自图中提供的更简单的协议。，其延迟比[CT05]中的延迟稍少一些。我们的协议也有一个额外的特性，这是在我们的原子广播协议中至关重要的，我们称之为强整体，并在下面讨论。原则上，我们可以在我们的原子广播协议中使用任何可靠的广播协议来满足这一额外的特性。

当然，我们希望可靠地广播的信息是块，所以我们将限制自己在这个设置中。

我们需要一个（n，n-2t）擦除代码（参见[Riz97]，简要介绍擦除代码、进一步引用以及在网络协议中的应用）。这样的代码可以获取大小为 S 的块 B，并将其分解成 n 个片段，每个大小为 rS/（n-2t），任何 n-2t 片段都可以用来重建块 B。注意，假设 t<n/3 时，我们有 S/(n-2t)<3S/n，因此所有 n 个片段最多大小为 r3S。

我们还需要一棵汞树。知道 B 块的一方可以获取 B 的所有 n 个片段，并构建一个以这些 n 个片段作为叶子的汞树，并只发布汞树的根。一方可以通过沿着 Merkle 树中的相应路径发布片段和节点（及其兄弟）来可验证地发布 B 的一个片段。我们称这些节点为验证路径。

最后，我们需要一个实例 S_r一个（t、n-t、n）阈值签名方案的基本参数。

当 P 出现时„可靠地用 k 轮广播它自己提出的块 B，它首先计算

- *散希 H(B)，*
- 碎片 Fi，……，F„的 B，和

- 一棵有网络的汞树。

然后，它为B构造了一个身份验证器，它现在有一个与第3.4节不同的语法：它是一个元组(身份验证器、k、a、哈希、a)，其中哈希是计算出的Merkle树的根，而a是一个有效的S自动由ˆ签名（验证器、k、a、哈希）。更一般地说：

> *在我们的原子广播协议中，我们不会使用块的散列作为块的“句柄”，而是使用相应的Merkle树的根；这不仅适用于身份验证者，还适用于公证和最终化，以及在块内标识块的父级。*

甲方。也将为每个参与方的Pˆ构造一个可验证的片段，其形式是

(rbc片段，k，a，哈希，月，path，Fˆ)，

其中，路径是Merkle树中的叶Fˆ的验证路径。最后，也是P方。

- 发送到每个它相应的片段，
- 向各方广播B的认证人和B的父母的公证，
- *可靠地接收B。*

现在考虑一下已经从P启动可靠广播的一方P°的逻辑。在k轮中，其中包括一个=月的情况。

1. 如果和何时，P°同时接收到一个身份验证器

   （身份验证器、k、a、哈希、a）

   以及相应的可验证的片段

   (rbc片段，k，a，哈希，月，路径，　　)，

   它将向所有各方广播该可验证的片段，以及具有该形式的rbc验证共享

   (rbc-验证共享、k、a、哈希、房车、月)，

   其中房车是有效的Pˆ签名共享

   （rbc验证，k、a、哈希）。

   *在下面，对P的rbc验证。*在k圆中的形式是

   （rbc-验证，k、a、哈希、a），

   其中，a是一个有效的Rbc签名

   （rbc验证，k、a、哈希）。

党 P^只会做一次(即，如果它看到第二个身份验证器从 P„在 k 轮中，它将忽略它)，如果它已经执行了步骤 2，它也不会这样做。

2. 如果和当 P^接收到

   - 与某个身份验证器对应的可验证的片段

     （身份验证器，k、a、哈希、a），

     不一定是它在步骤 1 中收到的相同的身份验证器，并且
   - 无论是没有相应的 rbc 验证共享或相应的 rbc 验证，一方 P^将
   - 重建相应的方块 B，
   - 计算所有 B 的片段，验证哈希是否正确
   - 验证 B 是否是有效的块（为此，可能需要等到收到 B 父母的公证）。

   如果上述验证通过，乌将构建 rbc 验证（如有必要），并进行广播

   - 对 rbc 的验证，
   - B 的认证人和 B 的父母的公证（如果是=月）。

   最后，与将可靠地接收 B（如果是=月）。

初步的观察结果。我们可以验证签名方案是否安全，哈希函数是否具有抗碰撞性，并满足了可靠广播的所有基本特性。事实上，有效性和完整性是清楚的。现在假设某个诚实的政党可靠地从腐败的政党那里一个区块 B„在 k 轮中。因为有一个 rbc 验证，所以这意味着至少有 n-2t 诚实的各方播放相应的片段和 rbc 验证共享。由于 t<n/3，和一个诚实的一方只发布一个 rbc 验证共享的 P„在 k 轮中，P 的任何其他身份验证器都不能有 rbc 验证 „这意味着一致性。它也意味着整体，因为至少 n-2t 方已经广播了相应的片段，并且方鸟已经广播了对方完成协议所需要的所有剩余数据。

事实上，在我们的原子广播协议中，我们利用了一个更强的特性，我们称之为强整体：

如果诚实的当事人已经可靠地从一方那里收到了一个区块 B„在 k 轮中，已经广播了足够的数据，以便任何诚实的一方恢复块 B。

对于上述协议，如果 n-2t 诚实方可靠地从 P 方接收了块 B„在 k 轮中，然后是一个诚实的聚会，除了 P„已经在步骤 2 中可靠地接收到 B，并且通过上述参数，所需的所有数据已经被广播。

正如我们将看到的，在我们的原子广播协议中，只有在它可靠地接收到这个块时，一方才会在一个块上广播公证共享。因此，如果任何诚实方在其池中有公证区块 B，那么一旦所有当前传输的消息被交付，所有诚实方也将 B 视为公证区块（以及区块链中 B 的所有祖先以 B 结束）。

假设该网络是^同步的。

如果到时间 T，所有诚实方都从诚实方发起了一个可靠的广播实例，那么在时间 T+25 之前，所有诚实方都将可靠地收到该消息。

此外，如果到时间 T，所有诚实方都从一个腐败方发起了一个可靠的广播实例，并且一些诚实方 Q 可靠地收到了来自该方的消息，那么在时间 T+5 之前，所有诚实方都将可靠地收到该消息。

就通信复杂性而言，因为每个可靠的广播实例最多广播一个可验证的片段。正如我们已经看到的，n 这样的片段的大小最多为 r3S，所以 n^2 这些片段（是所有诚实各方发送的片段总数）最多大小为 r3Sn。重要的是，不仅整个通信复杂性都有限制，而且在所有诚实的各方上分布良好——特别是，发送方贡献的通信复杂性只有其他各方的两倍左右。

5.3.3 使用可靠广播的一种可变的原子广播协议

我们现在提出了在第 5.2 节中提出的协议的一个变体，我们称之为协议 ICC2，使用上述可靠的广播协议作为子协议。具体详见图 5。然而，我们详细说明了一些额外的细节。

1. 集合 I 和 R 在每一轮的开始时被初始化为 0。

   - 当 P 时通过可靠的广播协议添加到 I 中 。从 r 级方（作为发送方或接收方）启动可靠的广播实例。
   - 当 P 时，通过可靠的广播协议添加到 R 中一个块 $B_{。}$ *可靠地接收 k 方的一个块。*特别是，当 $P_{。}$ 提出自己的块 B，块 B 立即添加到 R。

2. 很有可能是一个政党的 $P_{。}$ 接收公证（或定稿）或全套公证（或定稿）股份，即使没有可靠地接收（甚至根本接收）相应区块 B，甚至在相应一轮中从相应方发起可靠的广播实例。在这里，我们利用了上面所讨论的强整体性质。事实上，这种公证（或最终定稿)或一套公证(或最终定稿）股份意味着至少诚实的一方可靠地收到了区块，因此 $P_{。}$ 最终可以重建相应的块 B(通过归纳，区块链中以 B 结尾的 B 的所有祖先)。

3. 原则上，一旦是 P 的一方 。从某方 P 启动一个可靠的广播实例(在给定的 k 中，它将执行到完成，无论排名较低的块后来是否可靠地接收。但是，一次 $P_{。}$ 已经完成了这一轮，它不需要进行任何未完成的可靠的广播实例。这是由强大的整体属性来证明的。

5.3.4 分析方法

假设网络是^同步的，第一个诚实方 P 在时间 T 进入轮，然后在时间 T+5 之前，所有诚实方进入轮，包括 h 级的主要诚实方 Q。这对于可靠广播协议的强大整体属性是合理的：如果 P 以公证块 B 结束了前一轮，那么所有其他诚实方看到块 B 作为公证块所需的所有数据已经在 T 时飞行。因此，在 T+5+△道具(h)之前，所有诚实方都准备从 Q 和所有排名较低的方启动广播。

索赔要求。在 T+35+△道具(h)之前，所有诚实方都可靠地收到了一些<块。

一方面，假设到 T+5+△道具(h)，没有诚实的一方可靠地收到任何等级<h。这意味着到那时，所有诚实的一方都会

对于每一轮的 k=1,2,3.。。：

等待 t+1 共享的圆 k 随机信标计算圆 k 随机信标(它定义了圆 k 的排列 n)广播圆 k 共享的随机信标+1

让 r 我的名字是 P 的等级。

Z—0 *由 P 初始化可靠广播的秩组。*

R—0 *P 已可靠地接收到的数据块集。*

N—0 *P 已播放公证股票的一组区块。*

做完了-假的，到了-时钟的()

重复操作

等待以下一个：

(a) 经过公证的圆形 B 街区，

或一套完整的公证股份为部分有效
但未经公证的圆形 B 区块：

完成整轮将公证股份合并为 B 公证，必要时广播 B 公证
已真
如果 NC，那么广播 B 的最终共享

(b) 一个等级，是这样的

时钟()>到+△道具(r)，没有等级块：

在 k 轮为 r 级党发起可靠的广播

(c) rmeeZ，时钟()>到+△道具(rme)，和 R=0：

建议选择一个块，选择经过公证的圆（k-1）块 Bp 有效载荷 get 有效载荷(Bp)
创建一个新的圆 k 块 B=（块、k、a、H(BP)、有效负载），从 P 启动可靠的广播。在 B 块的 k 轮中

(d) 等级 r 的块

时钟()>到+△ntry(r)，并且没有秩的块：

为 B 产生公证份额
，广播 B 的公证份额

直到完成为止

图 5：ICC2：P 方的树状结构构建子协议。从 Q 开始可靠的广播，包括 Q 本身，在不到 25 个时间单位后，所有诚实的各方都将可靠地收到 Q 的建议。

另一方面，假设到时间 T+5+Ap，操作(h)，一些诚实的政党已经可靠地收到了任何等级的<h。让 B 组是当时任何诚实的一方都能可靠地收到的排名最小的块。因此，所有诚实方都将可靠地收到 B 少于 5 次单位。

这一主张允许我们保证活力，前提是 h=0（即，领导者是诚实的）和

$$^{35} + {}^{A}\text{支柱}^{(0)}\hat{M}\text{n 试(1)。} \tag{23}$$

在碰撞故障设置中，保证提供活力

$$^{35} + {}^{A}\text{支柱}^{(h)}\text{M}\,{}^{A}\text{n 试}^{(h} + \text{】).} \tag{24}$$

如果延迟函数被线性选择，如在（2）中，但斜率为 3Abnd，且网络保持不同步，则将满足不等式（23)和(24）。重要的是，正如在第 4.8 节中一样，如果 Abnd 太小，双方可以自适应地调整安特里函数来进行补偿，因此即使在这种情况下也能保持活力。

我们也可以使用这个主张来约束协议的通信复杂性。让我们成为任何诚实的政党在 k 轮开始广播的最高排名。然后，为(k)
我们已经提供了 Nmax<h+1

$$^{35} + {}^{A}\text{支柱}^{(h)\mathrm{M}\,A}\text{支柱}^{(h} + /). \tag{25}$$

如果延迟函数是线性选择的，如在（2）中，但斜率为 3A $_{\text{无编号}}$，并且网络确实保持不同步，我们已经绑定了 NSxMh，我们知道 E[h]=O（1)(见第 4.3.1 节）。与活力不同的是，如果 Abnd 太小，我们不能轻易地自适应地调整延迟函数来进行补偿。

至于延迟，对应于（4）的绑定就变成了

$$\text{最大值（35+支柱(h)，5+Antry(h)）+（h+1）5。} \tag{26}$$

5.4 限制数据块提案人的设置

我们可以将提出块的各方组限制在第 0 级的各方，…到目前为止，我们证明的所有修改都没有改变（结合上面给出的任何变化）。事实上，引理 2 中关于每轮公证块数的界限从 n 下降到 t+1。

参考文献

[AKK+02]N.阿隆，H.卡普兰，M.克里维维奇，D.马尔希和 J.P.斯特恩。一半系统故障时可扩展的安全存储。国际。兼容的。，174（2）：203-213,2002 年。

[AKK+07]N.阿隆，H.卡普兰，M.克里维维奇，D.马尔希和 J.P.斯特恩。“一半系统出现故障时可扩展安全存储”的附录。配置表 174（2)(2002）203-213]。国际。兼容性。，205（7）：1114-1116,2007 年。

[amn）20] I.亚伯拉罕，D.马尔基，K.Nayak，L.Ren，和M.Yin。同步热点：简单而实用的同步状态机复制。2020年IEEE安全与隐私研讨会，SP2020，美国旧金山，2020年5月18-21日，第106-118页。我，2020年。

【amnr18】 I.亚伯拉罕，D.马尔基，K.Nayak，和L.Ren。缺陷共识，探索。密码学电子打印档案，报告2018/1153,2018.https://eprint.iacr。组织结构表/2018/1153。

[anrx21] I.亚伯拉罕，K.Nayak，L.任，和Z.翔。拜占庭广播的良好案例延迟：一个完整的分类，2021年。arXiv：2102.07240，http://arxiv.org/abs/2102.07240。

[bcg20] M.Bravo，G.乔克勒，和A.戈斯博曼。让拜占庭共识现场直播（扩展版），2020年。arXiv：2008.04167，http://arxiv.org/abs/2008. 04167.

[bdn18] D. *博尼，德里弗斯先生和G.尼文先生。针对较小的块链的紧凑型多签名。在T.Peyrin和S.D.加尔布雷斯，编辑，密码学进展-ASIACRYPT2018-24密码学理论与应用国际会议，布里斯班，QLD，澳大利亚，2018年12月2-6日，论文集，第二部分，第11273卷计算机科学课堂讲稿，第435-464页。施普林格出版社，2018年。*

【本，83】 M. 本，或。自由选择的另一个优点：完全异步协议（扩展摘要）。在R.L.普罗伯特，N.A.林奇，和
N. *桑托罗，编辑，第二届ACM SIGACT-SIGOPS分布式计算原理研讨会论文集，蒙特利尔，魁北克，加拿大，1983年8月17-19日，第27-30页。acm，1983年。*

【bkm18】 E. 布赫曼、跆拳道和米洛舍维奇。关于BFT共识的最新八卦，2018年。arXiv：1807.04938，http://arxiv.org/abs/1807.04938。

D. 波， B. 林恩和沙罕。韦尔对的简短签名 C. 博伊德，编辑，密码学进展-ASIACRYPT2001，第七次密码学与信息安全理论与应用国际会议，澳大利亚黄金海岸，2001年12月9-13日，论文集，计算机科学课堂讲稿第2248卷，第514-532页。施普林格，2001年。

[bls01] M.Bellare和P.Rogaway。随机神谕是实用的：一个设计高效协议的范例。在D.E.丹宁，R.派尔，R.甘尼森，R.S.桑德胡，和V.阿什比，编辑，中国化学会‘93，第一次ACM计算机和通信安全会议会议记录，费尔法克斯，美国弗吉尼亚州，1993年11月3-5日，第62-73页。acm，1993年。

[br93] G. 布拉查语。异步拜占庭协议的协议。国际。投入。，75(2):130-143，1987.

【陈罗07】

[ckps01] C. 卡钦，库尔萨，佩佐德和舒普。安全和高效的异步广播协议。在 J.Kilian，编辑，密码学进展 2001，第 21 届国际密码学会议，美国加利福尼亚州圣巴巴拉，2001 年 8 月 19-23 日，论文集，计算机科学课堂讲稿第 2139 卷，第 524-541 页。施普林格，2001 年。

[ccs05] C. 卡钦，K.库尔萨维，和 V.舒普。君士坦丁堡的随机神谕：使用密码学的实用异步拜占庭协议。J.密码。，18（3）：219-246,2005 年。

[cl99] M.Castro 和 B.Liskov。实用的拜占庭式容错能力。在 M.I.Seltzer 和 P.J.Leach，编辑，第三次 USENIX 操作系统设计和实现研讨会(OSDI)，新奥尔良，路易斯安那州，1999 年 2 月 22-25 日，第 173-186 页。USENIX 协会，1999 年。

[cl02] M.Castro 和 B.Liskov。实用的拜占庭容错和主动恢复。ACM 横向分析。投入。系统节。，20(4):398-461，2002.

[ct05] *C. 卡钦和特塞萨罗。异步可验证的信息分散。在 P.弗雷伊诺，编辑，分布式计算，第 19 届国际会议，光盘 2005，波兰，2005 年 9 月 26-29 日，论文集，计算机科学课堂讲稿第 3724 卷，第 503-504 页。施普林格，2005 年。*

[cwa+09] A. 克莱门特、王家菲、阿尔维西先生、达林先生和马尔凯蒂先生。使拜占庭容错系统容忍拜占庭故障。在 J.Rex 福德和 E.G.Sirer，编辑，第 6 届 USENIX 网络系统设计和实现研讨会论文集，NSDI2009,2009 年 4 月 22-24 日，波士顿，马，美国，第 153-168 页。USENIX 协会，2009 年。http://www.usenix.org/events/nsdi09/tech/full_papers/clement/clement.pdf.

[dfi20] 精细。互联网计算机的技术概览，2020.https://medium.com/dfinity/a-technical-overview-of-the-internetcomputer-f57c62abc20f.

[dgh+87] A. J.需求， D. H.格林石油天然气股份有限公司， C. 豪泽，W.爱尔兰，J.拉尔森，S.申克，H.E.斯特吉斯， D.Ã C. 斯文，和 D.Ã B. 复制数据库维护的流行病算法。B. 施耐德，编辑，第六届 ACM 分布式计算原理年度研讨会论文集，温哥华，不列颠哥伦比亚省，加拿大，1987 年 8 月 10-12 日，第 1-12.ACM 页，1987 年。

[dls88] C. 工程，N。A. 林奇和斯托克迈耶。在部分同步存在下的共识。J.ACM，35（2）：288-323,1988。

[dyg+19] *S.Duan，灵气和张志志。异步 BFT 非常实用。D.李，曼南，巴克斯和王志志，编辑，2018ACM SIGSAC 计算机和通信安全会议记录，中国化学会 2018，加拿大多伦多，2018 年 10 月 15-19 日，2028-2041 页。2018 年上午上午。*

[GAG+19]G.戈兰-圭塔，I.亚伯拉罕，S.格罗斯曼， D. 马尔克希语， B. 皮克斯，赖特， D.塞雷迪斯奇，奥。塔米尔，和 A. Tomescu.SBFT：可扩展和分散的信托基础设施。在第 49 届 IEEE/IFIP 可靠系统和网络国际会议上，DSN2019，美国波特兰，6 月

24-27 日，
2019, 第 568-580 页。我，2019 年。

[GHM+17]Y. 吉拉德，R. Hemo，S. Micali，G. 弗拉克霍斯，和 N. 泽尔多维奇。阿尔哥兰：扩大拜占庭加密货币协议。密码学电子打印档案，报告 2017/454,2017 年。https://eprint.iacr.org/2017/454.

[GLT+20]郭、陆、唐、徐、张。Dumbo：更快的异步 BFT 协议。在 J. Ligatti，X. Ou，J. Katz 和 G. Vigna，编辑，中国化学会‘20：2020ACM SIGSAC 计算机和通信安全会议，虚拟事件，美国，2020 年 11 月 9-13 日，第 803-818 页。2020 年上午会议。

汉克，莫瓦赫迪和威廉姆斯。DFINITY 技术概述系列，共识系统，2018 年。arXiv：1805.04548，http://arxiv.org/abs/1805.04548。

自由银行团队。天秤座区块链中的状态机复制，
2020. https://diem-developers-components.netlify.app/papers/diemconsensus-state-machine-replication-in-the-diem-blockchain/202005-26.pdf.

L. 兰波特，R. E. 肖斯塔克和 M. C. 皮斯。拜占庭将军的问题。ACM 横向分析。该程序。朗先生。系统。，4（3）：382-401,1982 年。

默克尔。针对公钥加密系统的协议。《1980 年 IEEE 安全与隐私研讨会论文集》，美国加利福尼亚州奥克兰，1980 年 4 月 14-16 日，第 122-134 页。IEEE 计算机学会，1980 年。

默克尔。一个经过认证的数字签名。1989 年，第 9 届国际密码学年度会议，美国圣巴巴拉，1989 年 8 月 20-24 日，论文集，计算机科学课堂讲稿第 435 卷，第 218-238 页。施普林格，1989 年。

[mxc+16] A. 米勒，约夏，克罗曼，东石，和 D. 歌曲。BFT 协议中的蜂蜜獾。在韦普尔，卡岑贝瑟尔， C. 克鲁格尔石油公司股份有限公司， A.Ã C. 迈尔斯和 S. 哈雷维，编辑，2016 年 ACM SIGSAC 计算机和通信安全会议论文集，奥地利维也纳，2016 年 10 月 24-28 日，第 31-42.ACM 页，2016 年。

O. Naor，鲍德特先生， D. 马尔克语，和 A. 明镜周刊。戈斯沃斯：拜占庭观点同步，2019 年。arXiv：1909.05204，http://arxiv.org/abs/1909. 05204.

O. Naor 和 I. Keidar。预期的线性圆同步：线性拜占庭 SMR 缺少的链接，2020。arXiv：2002.07539，http://arxiv.org/abs/2002.07539。

R. 帕斯和 E. 施。带有最优的即时确认的区块链。在 J. 中。B. 尼尔森和 V. Rijmen，编辑，密码学的进展-EUROCRYPT2018-第 37 届密码技术理论和应用国际会议，特拉维夫，以色列，2018 年 4 月 29 日-5 月 3 日，论文集，第二部分，计算机科学第 10821 卷课堂讲稿，第 3-33 页。施普林格出版社，2018 年。

皮斯，R. E. 肖斯塔克和 L. 兰波特。在存在故障时达成协议。J. acm，27（2）：228-234,1980 年。

拉马萨米和 C.卡钦。吝啬的异步拜占庭容错原子广播。在 J.H.安德森，G.普伦西比和 R.瓦滕霍弗，编辑，分布式系统原理，第 9 届国际会议，OPODIS2005，意大利比萨，2005 年 12 月 12-12-14 日，修订精选论文，第 3974 卷计算机科学课堂讲稿，第 88-102 页。施普林格，2005 年。

L.Rizzo。针对可靠的计算机通信协议的有效擦除代码。
投入。请提交电子邮件。修订版，27（2）：1997 年 24-36 日。

施耐德。使用状态机方法实现容错服务：一个教程。ACM 组合。冲浪。，22(4):299-319，1990.

斯塔塔科普卢，大卫和武库科利奇。Mir-BFT：区块链的高通量 BFT，2019 年。arXiv：1906.05552，http://arxiv.org/abs/1906.05552。

A.沙米尔。如何分享一个秘密。请提交电子邮件。acm，22（11）：612-613,1979 年。