区块链通信漏洞

Andrei Lebedev[1] 和 Vincent Gramoli[1,2]

[1]悉尼大学 [2]Redbelly Network

arXiv:2603.02661v1 [cs.CR] 3 Mar 2026

摘要

区块链在处理节点间通信方面各不相同，这些通信用于传播信息、缓解攻击并就下一个区块达成一致。尽管已有研究发现安全漏洞，但这些漏洞均依赖于针对特定区块链通信协议量身定制的攻击。据我们所知，尚未有人比较过多种区块链通信协议在对抗性条件下的漏洞。

本文通过实证比较了五种现代、已投入生产的区块链——Algorand、Aptos、Avalanche、Redbelly 和 Solana——的通信协议在遭受五种不同攻击时的漏洞。我们得出结论：Algorand 易受丢包攻击；Aptos 易受定向负载攻击和领导者隔离攻击；Avalanche 易受瞬态故障攻击；Redbelly 的性能会受到丢包攻击的影响；Solana 易受停止攻击和领导者隔离攻击。我们的系统是开源的。

1 引言

众所周知，区块链容易受到网络攻击。它们通常隐含的 [20], 同步性假设 [33] 认为，所有消息都必须在已知的有限时间内送达，才能正常运行。由于这一假设在互联网这样的开放网络中并不现实，因此发生了各种攻击：互联网服务提供商攻击导致比特币资金被盗 [19], , 针对以太坊的余额攻击 [61,63,64] 都基于引入网络延迟，手段包括中间人攻击 [34]或 BGP 劫持 [8]。过去几年里，比特币损失了$70,000 [70] , 而以太坊经典损失了560万美元 [90]。这些攻击表明，通信协议会给经典区块链带来风险。然而，迄今为止，现代区块链技术的网络协议还从未被相互比较过。

本文比较了现代区块链通信协议的安全性。我们不考虑已知存在同步性不足问题的以太坊和比特币，而是将比较重点放在以下五种现代区块链上：*(1)* 据称，Algorand 在部分同步环境下不会违反安全性。 [38], *(2)*已有证明表明，Avalanche 可在同步环境下运行，并据称该证明可推广至非完全同步环境。 [71], *(3)*Aptos [9]和 *(4)* Redbelly [28] 是旨在应对意外消息延迟的区块链；*(5)*Solana [29] 是近期兴起的区块链，因推出迷因币而广受欢迎，其中包括美国总统家族推出的迷因币。它采用纠删码来应对丢包。这些区块链采用了各不相同的通信协议。Algorand 和 Aptos 的通信协议旨在通过将负责决定下一个区块的验证者节点分别置于中间节点 [26] 和验证者全节点 [10], 之后，从而隐藏这些验证者节点。Avalanche 和 Redbelly 的通信协议会使验证者节点暴露于拒绝服务（DoS）攻击，但分别通过限流和速率限制来保护它们，丢弃原本会消耗过多资源的特定请求。Solana 的通信协议利用分层覆盖网络来最大化传播 [77] , 有时会阻止对验证者节点的直接访问。 [9]

为了在统一标准下比较区块链通信协议的安全性，我们调查了针对特定区块链的攻击 [7, 22, 35, 44, 47, 57, 58,61, 63,64, 72–74,83]，并由此归纳出以下五种全新的攻击。这些攻击以与区块链无关的方式损害通信协议：

定向负载攻击。 一种受拒绝服务攻击 [22, 57, 73], 启发、向一个区块链节点发送流量的攻击。

瞬态故障攻击。 一种受持续 churn [47, 58], 启发、短暂影响少数节点的攻击。

丢包攻击。 一种受平衡攻击 [61, 63, 64], 启发、丢弃部分网络数据包的攻击。

停止攻击。 一种使部分区块链网络崩溃的攻击，受多数攻击 [7,35, 72]启发。

领导者隔离攻击。 一种受日蚀攻击 [44, 74, 83], 启发、隔离一个区块链节点的攻击。

我们的研究结果汇总于表 1，并在后续章节中详细阐述，由此得出四个有趣的结论。

表 1：不同区块链在遭受攻击时的漏洞情况，以及区块链团队声称的容错阈值（以区块链节点数量 n 为变量）。红色叉号表示存在显著漏洞，勾号表示在所测试的攻击条件下观察到其具有韧性。短横线'—'表示结论不确定或情形不适用。(*) 请注意，对于概率协议 Algorand 和 Avalanche，我们采用了其原始论文中选取的阈值 $n/5$，该阈值可将全局故障的概率降至 10^{-9} 以下。

区块链	参考	通信	容错（f）	Load	故障	Loss	停止	隔离
Algorand	[38]	Gossip	$n/5$*	✓	✓	✗	✓	✓
Aptos	[9]	分层	$n/3$	✗	—	—	✓	✗
Avalanche	[71]	启用限流	$n/5$*	—	✗	—	—	—
Redbelly	[28]	速率受限	$n/3$	✓	✓	—	✓	✓
Solana	[85]	分层	$n/3$	✓	—	✓	✗	✗

导致区块链崩溃。我们注意到，一些区块链（如 Avalanche 和 Solana）在遭受某些攻击时会停止处理请求。首先，Avalanche 容易受到瞬态故障攻击，因为限流与瞬态故障相结合，会使其无法提交交易。其次，Solana 容易受到停止攻击：如果足够多的节点经历瞬态中断，那么即使这些节点恢复，系统也可能陷入停滞。

传输协议影响。所选网络传输协议对信息传播质量起着关键作用。Redbelly 的性能尤其受到丢包攻击的影响，因其基于 TCP 的协议使其无法恢复丢失的交易。这通常是所有基于 TCP 的区块链的共性，包括 Algorand、Aptos 和 Avalanche。但 Solana 凭借 QUIC 与纠删码的结合，即使在极具对抗性的丢包条件下也能成功传播区块。

单节点隔离。Aptos 和 Solana 易受领导者隔离攻击影响，因为 Aptos 基于声誉的领导者选择与 Solana 的领导者调度都是确定性的，使攻击者能够预测并针对连续的领导者。因此，针对网络中的单个节点就可能转化为这些区块链网络的全局减速。

节点复杂性暴露。一些区块链隐藏了攻击者只需极少努力即可触发的 CPU 密集型任务。尤其是 Aptos 易受定向负载攻击影响，因为单个验证者在试图避免产生二次数量的签名时，可能因必须收集的签名而过载。

第 2 节介绍背景，第 3 节介绍用于比较区块链安全性的实验设置。第 4 节介绍在负载饱和攻击下响应请求的问题。第 5 节评估区块链对瞬态故障攻击的脆弱性，第 6 节评估其抵御丢包攻击的能力，第 7 节考察其对停止攻击的脆弱性，第 8 节分析领导者隔离攻击对它们的影响。第 9 节讨论 Avalanche 和 Solana 中存在的两个关键问题及其应对措施，第 10 节介绍相关工作，第 11 节作出总结。

负责任披露

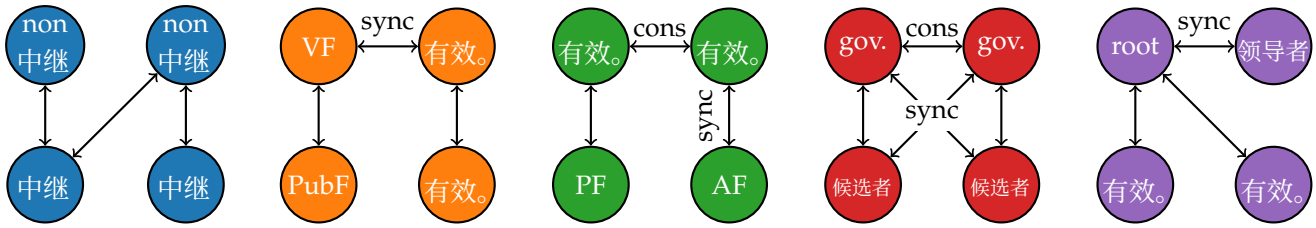
按照负责任披露原则，我们已在发表本文前，将文中讨论的漏洞和发现通知相关开发团队和安全团队。这包括与 Algorand、Aptos、Avalanche（Ava Labs）、Solana（Anza）和 Redbelly 相关的团队，使他们有机会调查并审查我们的结果。因此，Anza 确认 Solana 存在漏洞。

2 背景

本节将回顾本文评估的现代生产环境区块链的通信协议（参见表 1），这些区块链据称是安全的。

2.1 Algorand 通信协议

Algorand [38] 是一种区块链协议，它通过可验证随机函数（VRF）对共识参与者进行洗牌。随后，共识参与者运行 $BA\star$ 协议，以便在存在拜占庭故障的情况下就一个区块达成一致。每个节点在通过 Gossip 将每条消息至多向每个其他节点发送一次之前，都会先对其进行验证 [26]。为此，每个节点为其邻域中的每个节点维护一条 TCP 连接，并通过 HTTP 提供 WebSocket。如下一页的图 1a 所示，不同节点可以扮演不同角色：中继节点连接到（可能多个）非中继节点，这些非中继节点运行共识，但彼此并不相连。这些节点可以是归档节点和参与节点 [1]。



(a) Algorand (b) Aptos (c) Avalanche (d) Redbelly (e) Solana

图1：不同区块链所使用的不同拓扑，其中 valid.、VF、PubF、AF、PF、gov. 和 cand. 分别表示验证者、验证者全节点、公共全节点、归档全节点、裁剪全节点、治理者和候选节点，而 cons、sync 和 req 分别表示共识、同步和请求。

2.2 Aptos通信协议

Aptos [9] 是一种基于领导者的区块链，它构建于实用拜占庭容错 [21] 的一种变体之上，该变体的视图变更通信复杂度为二次，通信复杂度为三次。为应对领导者瓶颈 [81]和冗余验证 [78], ，Aptos 采用了 Quorum Store 优化。验证者并行重复以下步骤：*(1)* 从内存池拉取交易；*(2)*将交易整理为批次；*(3)* 向所有其他验证者广播批次；*(4)*持久化收到的批次，对其摘要进行签名，并回传签名；以及*(5)* 从超过 $2n/3$ 个节点收集签名，以形成存储证明。这使得验证者能够异步广播交易，从而在共识协议执行期间卸载领导者网络接口上的负载 [25]。Aptos 的通信模型基于 TCP，并且是分层的，节点在不同层级扮演不同角色 [10]。如图 1b 所示，验证者节点位于顶层并运行共识，其子节点是验证者全节点（VF），它们从验证者节点下载当前状态。叶节点是公共全节点（PubF），它们从 VF 节点下载区块，但不能与顶层（即验证者节点）直接通信。

2.3 Avalanche通信协议

Avalanche 是概率性的，并借助 Snowflake 二元共识协议 [71] 逐步趋向一致。节点通过 TCP 通信，并利用节流机制 [16] 来限制其他节点所消耗的 CPU、磁盘、带宽以及消息处理量。如图 1c 所示，Avalanche 设有归档全节点（AF）和裁剪全节点（PF），这些节点与验证者同步，而验证者则负责运行共识 [15]。引入动态费用以及逐步提高 Gas 目标值的升级将在后文第 9.1 节中讨论。动态提议者选择算法用于管理网络负载和区块生产。该算法基于由父区块高度和链 ID 派生的种子，为下一个区块高度伪随机地选择一系列潜在提议者。该序列表明提议者以何种顺序追加并签署新区块。

2.4 Redbelly通信协议

Redbelly 区块链 [28] 是一种基于 DBFT 共识算法 [27] 构建的可扩展区块链，该算法无领导者 （即并非基于领导者），且不需要同步性。DBFT 已借助参数化模型检验 [18], 完成形式化验证，结果表明它能在所有执行过程中、对任意系统规模解决共识问题。一种协作方法可保证追加包含尽可能多有效提议区块的超级区块 [28] 。节点使用 TCP 通信。如图 1d 所示，在所有被称为候选节点的节点中，会有一个随机子集被临时提升为治理者，并在运行某一共识实例时直接交换消息，以减少资源浪费 [40]。候选节点与多个治理者进行同步，从而（确定性地）应对拜占庭行为。每个候选节点、治理者或客户端均受到速率限制，以缓解拒绝服务攻击。Redbelly 具备可扩展以太坊虚拟机（SEVM），并运行以 Solidity [78] 编写的去中心化应用 （*dApps*） 。

2.5 Solana通信协议

Solana [77] 是一种基于领导者的区块链。其交易需要经过 30 个后续区块（至少 12.4 秒）才能提交。领导者按四个时隙进行调度，并根据验证者的质押情况，将验证者分配到一个纪元内的特定时隙。验证者会对分叉分支投票：每次在同一分支上连续投票，锁定期都会加倍，因此验证者必须等待锁定期结束后，才能对冲突分支投票。这种机制促使验证者选择最重的分支，但如果许多验证者同时被锁定而无法对新时隙投票，就可能导致活性停滞。节点通过 QUIC 网络协议 [46] 进行通信，以交换交易。节点将区块拆分为区块片段，并通过 UDP 按分层结构进行传播，这种结构称为 Turbine [6], 。如图 1e 所示，领导者通过将区块发送给根节点来传播区块，随后根节点再将其传播出去 [24]。交易按照由历史证明生成的时间戳进行全序排列。

3 区块链安全比较

本节列出我们的假设，定义对手，并介绍实验设置。

3.1 系统模型

我们将区块链网络视为由一组节点 构成，这些节点负责提交客户端发送的交易。为了对现实的开放网络进行建模（在这种网络中，节点可能相互串通并影响通信），我们定义一种对手：它要么是内部对手，控制部分节点（例如一个联盟）；要么是外部 对手，仅控制网络（例如伪造或丢弃消息），这与 Giuliari 等人 [39] 的做法类似。

3.2 威胁模型

为了对现实的开放网络进行建模（在这种网络中，节点可能相互串通并影响通信），我们定义一种对手：它要么是内部 对手，控制部分节点（例如一个联盟）；要么是外部 对手，仅控制网络（例如伪造或丢弃消息），这与 Giuliari 等人 [39] 的做法类似。更具体地说，对手可以执行以下五种与协议无关的攻击，它们构成了我们的“对抗性通信环境”的基础：*(i)*定向负载攻击：一种受拒绝服务（DoS）攻击 [22, 57, 73]，启发的攻击，以恒定速率向其区块链网络中的单个节点发送请求；*(ii)*瞬态故障攻击：一种受区块链网络中的持续 churn [47, 58], 启发的攻击，在短时间内使区块链网络中相对较小的一部分发生崩溃；*(iii)* 丢包攻击：一种受平衡攻击 [61, 63, 64], 启发的攻击，丢弃区块链网络两部分之间交换的数据包；*(iv)*停止攻击：一种受多数攻击 [7, 35, 72], 启发的攻击，使区块链网络中很大一部分发生崩溃，从而停止区块链服务；*(v)* 领导者隔离攻击：一种受日蚀攻击 [44, 74, 83], 启发的攻击，隔离区块链服务所用共识协议的领导者。

3.3 实验设置

我们部署了五种类型的区块链网络：*(i)*Algo- rand v3.27.0，其中节点是中继节点；*(ii)* Aptos v1.25.1、*(iii)* Avalanche C 链 v1.12.1 和 *(iv)* Solana Agave v2.0.20，其中节点是验证者；以及 *(v)* Redbelly v0.36.2，其中节点是治理者。每台机器都模拟区块链网络中由个人运行的普通计算机，因此，为了获得真实的结果，我们选择了每台配备 4 个虚拟中央处理器和 8 GB 内存的虚拟机（VM）。需要注意的是，这一配置低于部分区块链通常建议的配置，包括 Aptos [11], Avalanche [59] 或 Redbelly [69], ；然而，远程节点的严格硬件要求仍然难以强制执行，而独特的配置并非

```
1 ethtool -K eth2 tso off gso off gro off sg
      off
2 tc qdisc add dev eth2 root handle 1: prio
3 tc filter add dev eth2 protocol ip parent 1:0
       prio 3 u32 match ip dst 10.40.10.1
      flowid 1:3
4 tc qdisc add dev eth2 parent 1:3 handle 30:
      netem loss 50%
5 tc qdisc del dev eth2 root
6 ethtool -K eth2 tso on gso on gro on sg on
```

清单 1：注入瞬时丢包的命令。

这对于我们的比较是必要的。评估安全性只需一个小型网络。因此，我们在由物理服务器组成的 Proxmox 集群上部署了包含 25 台虚拟机的分布式系统，这些虚拟机运行 Ubuntu 24.04.1 LTS。每台服务器均配备 4 颗 AMD Opteron 6378 16 核 CPU，运行频率为 2.40 吉赫，拥有 256 GB 随机存取存储器和 10 千兆以太网网卡。该设置包含 5 个基准测试节点和 20 个区块链节点，另有最多 20 个客户端，每个客户端都向单个区块链节点发送原生转账交易。为了自动化操作，我们使用 Diablo [41] 基准测试，其中每个客户端都是独立的 Diablo 副节点。

3.4 注入瞬时丢包

清单 1 展示了我们如何配置网络接口上的流量控制设置，以引入临时丢包。这一系列命令会修改网络接口 eth2，使发往10.40.10.1的数据包产生 50% 的丢包率，然后恢复原始配置。具体来说，我们禁用了硬件卸载功能（第 1 行），添加了prio 队列规则（第 2 行），将发往特定 IP 地址的流量（第 3 行）过滤至应用丢包的netem 队列规则（第 4 行）。随后，我们清除了规则（第 5 行），并重新启用了卸载功能（第 6 行）。此方法可以临时引入丢包。通过动态移除配置，丢包影响是暂时的，无需重启系统。

3.5 通过日志检测领导者

我们开发了一套实时日志监控系统，用于跟踪区块链网络中的领导者。该系统识别日志模式，以判断基于领导者的区块链中的节点何时成为或不再担任领导者（例如，我现在是领导者、ProposerElection或协议特定的区块构建消息）。该监控系统使用基于inotify的文件监视器持续解析节点日志，并维护领导者状态的状态模型。检测到节点开始担任领导者后，监控器会触发一条 tc 规则，对与该节点关联的网络接口施加丢包（例如，loss 75%），并在节点不再担任领导者时停止丢包。这样，我们便能在不修改协议的情况下，在节点担任领导者期间隔离其连接，暂时以有针对性的方式降低其参与共识的能力。

3.6 测量点对点带宽

我们还实现了一套细粒度带宽监控系统，用于捕获攻击对网络层面的影响。我们的方法可以测量网络中每个节点之间交换的流量，并提供成对的测量结果。该系统依托 Linux iptables 防火墙基础设施，以非侵入方式进行数据包记账。对于每个受监控节点，我们会以编程方式安装一组 iptables 规则。这些规则会创建自定义记账链，其中针对实验中的每个其他对等节点都设有一条特定规则。每条规则都会根据数据包的源 IP 地址（针对入站流量）或目的 IP 地址（针对出站流量）进行匹配。随后，监控脚本会定期查询与各个对等节点规则关联的字节计数器，并立即将其重置为零。此过程会生成一个时间序列，其中每个数据点表示前一时间间隔内的平均发送（TX）和接收（RX）速率，使我们能够精确分析网络行为。

3.7 发送速率

受 Avalanche 每秒 357 笔交易的吞吐能力限制（该能力由区块间隔 2 秒及其 15M gas 上限决定），我们将实验的发送速率固定为每秒 200 笔交易。虽然这一速率可作为公平的基线，但初步测试发现，它会触发 Avalanche 动态费用机制中的负载诱发漏洞。在持续负载下，网络的 gas 消耗始终超过目标，导致基础费用不断上涨，直至交易处理完全停止。为公平比较通信协议，我们在所有 Avalanche 实验的创世配置中禁用了这一费用上涨机制。第 9.1 节将进一步讨论该费用机制的具体细节及我们的应对措施。

4 易受定向负载攻击

在本节中，我们通过向一部分特定的区块链节点发送每秒 200 笔交易的有效原生转账请求，评估区块链遭受定向负载攻击的漏洞。与此前受到广泛研究的区块链拒绝服务攻击类似， [22,57,73], 这些攻击旨在增加延迟并造成交易丢失。有趣的是，此攻击揭示了 Aptos 中的瓶颈效应，其服务延迟比其他区块链高出一个数量级。此外，此攻击也会影响 Avalanche；不过，我们找到了应对措施，并按照第 9.1 节所述，仅需禁用 Avalanche 的新基础费用机制即可实施。

4.1 展示延迟劣化

为展示延迟劣化，我们针对五种区块链进行了无故障实验，并实施定向负载攻击。单个客户端以每秒 200 笔交易的持续速率，向单个区块链节点发送交易。结果显示，Aptos 存在显著的性能漏洞。虽然大多数区块链都能高效处理负载，但 Aptos 的速度慢了一个数量级。要提交所有交易，Avalanche（已禁用基础费用）和 Redbelly 用时不到 3 秒，Algorand 用时 10 秒，Solana 用时 27 秒。相比之下，Aptos 则需要 4 分 15 秒。中位数（第50百分位）延迟的差异也十分明显：Aptos 超过 2 分 30 秒，而其他系统的延迟仍然很低。需要指出的是，此攻击针对的是静态节点；针对 Aptos 轮换领导者的攻击将在第 8 节讨论。

Aptos 缺乏响应能力显然属于异常行为。尽管 Aptos 建议为每个节点配置更多资源，包括 32 核、2.8 吉赫和 64 GB 内存， [11], 比我们在本实验中配置的资源更多，但我们认为实验所用资源接近普通商用计算机的配置。此外，我们将在第 4.3 节展示，在没有遭受此攻击的情况下，我们的资源足以让 Aptos 快速提交交易（中位延迟为 2.27 秒）。由于每个验证者都知道其他所有验证者的身份，这一问题揭示了 Aptos 在定向负载下的一个重要漏洞：对手可能通过持续向单个节点发送请求，使 Aptos 服务停止运行。

4.2 Aptos 验证者瓶颈

为了更好地理解 Aptos 为何容易受到定向负载的影响，我们测量了发送请求期间及之后的吞吐量。图 2 描述了无故障情况下 Aptos 的吞吐量随时间的变化：在 20 台服务器中，有一台服务器接收来自单个客户端以每秒 200 笔交易的速率发送的请求。800 秒后，客户端停止发送请求，但我们仍继续监测吞吐量 200 秒，如条纹区域所示。我们观察到，800 秒后吞吐量突然跃升，这表明 Aptos 在攻击期间因每秒 200 笔交易的负载而发生了拥塞，原因似乎是内部瓶颈。

导致该漏洞的 Aptos 瓶颈由 Quorum Store 协议引起，该协议将在第 2.2 节讨论。Quorum Store 协议消除了领导者瓶颈，却造成了验证者瓶颈。

为验证这一假设，我们分析了验证者之间的网络流量分布。数据显示，在 Aptos 中，遭受攻击的节点承载了大量交易流量。如图 3 所示，该单一节点贡献了各节点总出站流量的 25.7%，与其他区块链形成鲜明对比。这凸显了 Aptos 架构中显著的流量失衡。

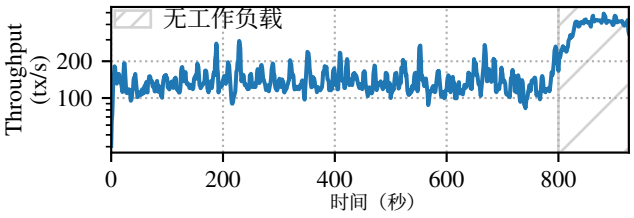


图 2：在持续 800 秒的负载攻击期间，Aptos 的拥塞使其无法像攻击停止后（800 秒后）那样快速提交交易。

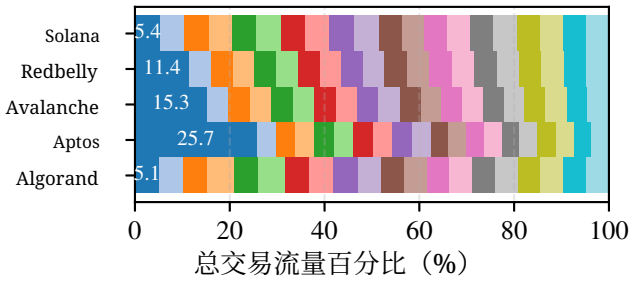


图 3：在实验期间，接收交易的 Aptos 节点产生的出站流量占总量的四分之一，最高可达其他区块链的 5 倍。

对验证者节点进行日志检查后发现，只有接收方验证者节点经历了ProofOfStoreInit 和ProofOfStoreReady 事件，这些事件属于ProofCoordinator 模块。这表明只有接收方节点会启动并完成存储证明流程；该节点必须：*(1)* 处理所有传入交易；*(2)*生成所有批次；*(3)* 将批次发送给所有其他验证者；以及*(4)* 为每个批次收集来自 $2n/3+1$ 个验证者的签名。由于存储证明处理必须在共识排序开始前完成，因此该节点的网络和处理能力成为主要限制因素。因此，攻击者可以让单个 Aptos 验证者节点负载达到最大，从而最大化这种瓶颈效应。分散负载看似可以缓解问题，但正如我们接下来将展示的，这会引发其他问题。

4.3 平衡负载未能缓解攻击

为缓解上述识别出的定向攻击漏洞（第 4.2 节），我们在不注入故障的情况下，改变了受到 200 TPS 负载定向影响的验证者节点数量。图 4 仅展示了负载分配至 5%、25%、30%、55% 和 60% 的验证者时的结果，因为当负载分配至 60% 至 100% 的验证者时，Aptos 会丢失交易。这实际上意味着，即使完美地平衡负载，也无法解决该问题。从图 4 可以看出，当 30% 的验证者接收交易时，中位延迟最低，为 2.27 秒。然而，当 200 TPS 的负载分配至 60% 的验证者时，Aptos 开始丢失交易。当客户端向所有节点发送交

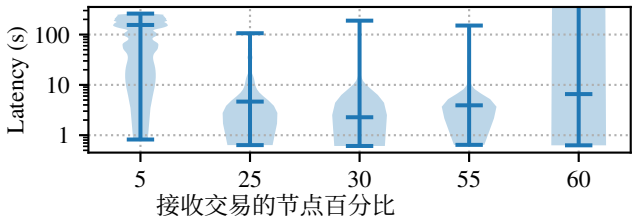


图 4：在持续 800 秒、恒定且均匀的 200 每秒交易数工作负载下，Aptos 网络的交易延迟分布。

易时，只有 50% 的已提交交易最终得到提交；而当负载分配至 70% 的节点时，90% 的已提交交易最终得到提交。这表明，即使分散定向负载攻击，也可能导致拒绝服务攻击。

我们推测，系统在接收交易的节点过少或过多时负载过重，原因分别是密码解密和密码加密：一方面，当只有一个节点接收所有交易时，它必须为自己创建的每个交易批次验证 $\lfloor 2n/3 \rfloor + 1$ 个签名。由此产生的 CPU 消耗会拖慢该节点，使其成为瓶颈。另一方面，当有很多节点（例如 $\Omega(n)$ 个）接收所有交易时，所需签名总数会呈二次增长，即 $\Omega(n^2)$，因为不同的接收节点会创建不同的批次，因此这些 $\Omega(n)$ 个节点中的每一个都需要 $\lfloor 2n/3 \rfloor + 1$ 个不同的签名。

5 瞬态故障攻击漏洞

本节评估区块链易受瞬态故障攻击的程度。这些攻击的动机来自区块链网络中观察到的 *churn* 或参与动态变化 [47]，也来自对其影响的模拟 [58]。我们通过使节点崩溃并恢复来实施此类攻击，并在后续实验中逐步增加受影响节点的数量。我们的观察表明，Avalanche 易受此类瞬态故障攻击的影响，因为故障与其限流机制共同作用，导致交易丢失。尽管 Aptos 似乎也受到此类瞬态故障攻击的影响，但这一漏洞似乎是由第 4 节提到的拥塞所致。我们没有观察到这些攻击对 Algorand 和 Redbelly 产生任何影响；而候选领导者发生瞬时故障，则会阻碍 Solana 快速趋于稳定。

5.1 Avalanche 交易损失

我们预计，即使一小部分节点出现短暂的瞬态故障，也可能降低区块链服务的质量。为了以实证方式验证这一点，我们实施了一次攻击，最初涉及 10% 的瞬态故障，使 10% 的

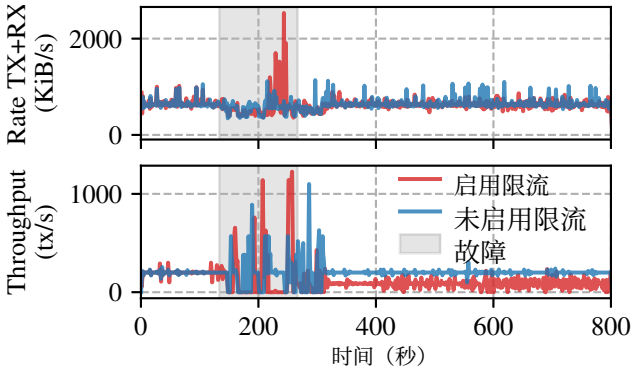


图 5：瞬态故障攻击下 Avalanche 的网络流量（TX+RX 速率）和交易吞吐量。启用限流（红色）时，在故障窗口（133-266 秒）内尝试对故障节点进行采样以参与共识，会导致大量无效网络流量激增。这种活动会触发限流机制，进而抑制网络活动，并阻止吞吐量在故障后恢复，从而造成持续的交易损失。相比之下，未启用限流的系统（蓝色）没有流量激增，并且能够快速恢复。

区块链节点在 133 秒时崩溃，同时仅向正常节点发送交易，然后在 266 秒时恢复故障节点。鉴于区块链需要共识 [4]，并且在一般设定下，只要“确定性”故障少于三分之一，共识问题便可得到解决 [49]，这一发生瞬态故障的节点所占比例相对较低。

然而，我们观察到 Avalanche 无法承受瞬态故障攻击，并开始丢失交易。在我们的实验中，Avalanche 永久丢失了约 60% 的交易；即使故障节点恢复后很久，也只有 40% 的交易得到提交。短暂故障仅影响了 10% 的质押，却产生如此持久的影响，这令人惊讶，原因有二。首先，预计 Avalanche 能够容忍最多 20% 质押的所有者离线。其次，我们注入的故障仅为瞬时崩溃，其危害性低于拜占庭式（或恶意）故障。

5.2 Avalanche 限流放大效应

为缓解拒绝服务攻击，Avalanche 采用了一种限流机制，根据资源消耗对消息处理实施速率限制。尽管该机制旨在提供防御，但它似乎是不稳定性在瞬态故障下出现的根本原因。如图 5 所示，当我们在禁用限流的情况下（蓝线）重复进行 10% 瞬态故障实验时，Avalanche 成功恢复，并已提交 100% 的交易。然而，在启用限流后（红线），网络未能恢复，导致大量交易丢失。

这种放大效应源于共识采样故障，而非交易积压。在故障期间，Avalanche 的动态提议者选择协议继续从整个验证者集合中采样，其中包括 10% 的离线节点。当健康节点试图与这些无响应的宕机提议者通信时，会产生大量无效网络流量，表现为图 5 上方面板中陡峭的红色尖峰。

限流机制将这次失败共识通信的峰值误判为恶意过载，并开始抑制消息队列。因此，即使故障节点在 266 秒时恢复，限流防御机制仍保持启用状态，对恢复共识所需的合法通信实施速率限制。如图 5 下方面板所示，这使系统无法恢复其吞吐量，并导致在攻击期间及之后提交的交易永久丢失。我们的发现阐明了此前关于限流可能造成吞吐量不稳定的观察结果 [42]；然而，该结果是在默认配置下获得的，正如我们在第 9 节中所解释的，动态费用机制也可能导致停顿。我们的工作表明，限流是一种独立的漏洞，即使费用问题得到纠正，也会造成交易丢失。

5.3 Aptos 交易损失

Aptos 同样易受瞬态故障攻击影响，但需要高得多的故障比例才会受到冲击。我们对 Aptos 实施了更严重的攻击，在 133 秒时使其 35% 的区块链节点崩溃，并在 266 秒时将其恢复。由于交易需要更长时间才能被提交，这可能是因为第 4.2 节所述的验证器瓶颈，我们不得不延长实验时长。即便在这一更为严苛的 35% 故障场景下，Aptos 仍比 Avalanche 更具韧性，最终提交了其 89% 的交易。然而，性能显著下降，处理该负载需要超过 600 秒，并且最终仍损失了所提交交易的 11%。请注意，这是在所有交易都发送至正常节点的情况下注入瞬时故障的结果。

5.4 Avalanche 多重故障的影响

有趣的是，随着瞬时故障攻击针对的节点增多，Avalanche 的吞吐量会下降。在 Algorand、Redbelly 和 Solana 中，这种逐步恶化的现象并不那么明显。图 6 展示了一个 Avalanche 网络在 800 秒内以每秒 200 笔交易运行时的吞吐量随时间的变化，数据按 2 秒滑动窗口取平均值。蓝色曲线（红色曲线）表示在 133 秒时有 10%（25%）的节点发生故障，并于 266 秒时恢复的实验。故障发生前，吞吐量相对稳定。在 133 秒时，红色曲线的降幅比蓝色曲线更大。故障节点恢复后，吞吐量仍意外地处于低位：发生 2 次故障时，仅有 36% 的交易

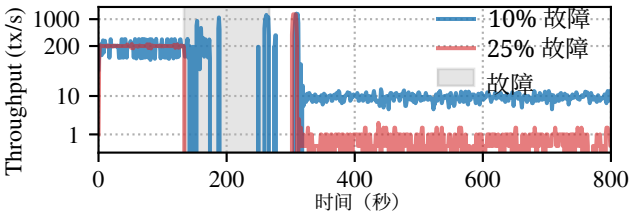


图 6：随着瞬态故障数量增加，Avalanche 吞吐量下降。

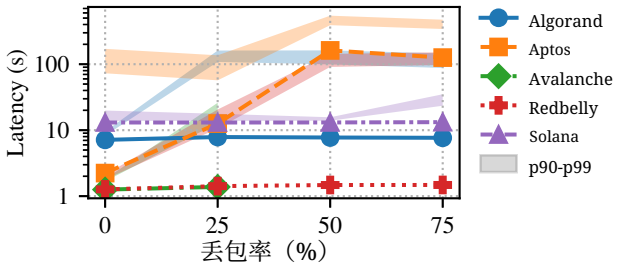


图 7：在持续 800 秒、恒定工作负载为 200TPS 的压力测试下，受测 5 个区块链的交易延迟百分位数。曲线表示中位延迟。

已提交；而发生 5 次故障时，仅有 22% 的交易已提交。

这一漏洞的原因是内存池已满时产生的连锁效应：拒绝交易会导致随后具有更大随机数的交易也被拒绝。攻击期间约可提交 26,600 笔交易（200 TPS × 133s），而 Avalanche 交易池最多只能存储 6,144 笔交易。由于空间不足，交易遭到拒绝后，具有更大随机数的新交易也会因 Er-rUnderpriced错误而被拒绝，尽管它们的定价相同，这是由pool.priced.Underpriced(tx)=true导致的。此外，恢复中的节点内存池为空，必须从其他节点接收交易。故障节点越多，交易丢失就越严重，从而加剧对吞吐量的影响。当攻击针对网络中 25% 的节点时，Avalanche 的性能会进一步下降，这可由表 1 中所示的其$n/5$ 故障阈值解释。

6 丢包攻击漏洞

本节评估区块链对丢包攻击的脆弱性。在这类攻击中，对手会故意丢弃节点之间或网络分段之间的网络消息。与旨在将网络划分为均衡子网络的平衡攻击 [61,63,64] 类似，此类攻击会扰乱两个子网络之间的通信。我们通过注入不同比例的丢包来实施这些攻击。尽管在整个网络范围内造成高比例丢包颇具挑战，但这些测试揭示了协议本身的敏感性。我们发现，传输协议（TCP 与 QUIC）的选择会显著影响对丢包的容忍度，而表 2 所示的点对点带宽测量结果有力地支持了这一结论。数据揭示了对基于 TCP 的协议造成的巨大影响：攻击期间，Algorand 和 Redbelly 等链的平均对等带宽下降了 95% 以上。与之形成鲜明对比的是，使用 QUIC 的 Solana 在整个攻击期间都保持了完整的发送（TX）速率，接收（RX）带宽也仅部分下降。这种网络层面的韧性避免了连锁故障，例如基于 TCP 的系统（如 Algorand）所遭遇的广播队列完全堵塞。

6.1 利用丢包进行攻击

为评估对丢包攻击的脆弱性，我们在一组 $\lfloor f+1 \rfloor$ 个节点与网络其余部分之间丢弃 25% 至 75% 的数据包；这组节点的规模为各区块链在表 1 中报告的故障容忍比例 f。此操作在 133 秒后进行。最后，我们在 266 秒后停止丢包。为避免第 4.3 节中观察到的 Aptos 拥塞，本节所有实验均选择 25% 的区块链节点。

6.2 基于TCP的区块链的漏洞

图 7 展示了丢包增加对 5 个区块链延迟的影响。横轴表示引入系统的丢包百分比，纵轴采用对数刻度，表示以秒为单位的延迟。每个区块链的第50百分位（即中位延迟）以实线表示，阴影区域则体现第90百分位（即 90th 百分位数）与第99百分位（即 99th 百分位数）延迟之间的变化范围。在较少（25%）丢包情况下，Aptos 和 Algorand 的漏洞更为明显，其第90百分位和第99百分位延迟显著上升。然而，在更多（50% 和 75%）丢包情况下，出现了更为明显的分化。最极端的漏洞出现在 Avalanche 中，其在 50% 和 75% 损失下的第50百分位、第90百分位和第99百分位数值缺失。同样，Aptos 和 Redbelly 在这些损失水平下延迟急剧上升，表明在不利网络条件下共识停滞，导致最终确认出现严重延迟。

与其他基于 TCP 的区块链一样，Algorand 在遭受 50% 和 75% 丢包攻击时受到的影响更大，这意味着攻击期间交易无法提交。因此会出现明显的空档期，没有交易得到最终确认；丢包停止后，积压的交易开始提交，延迟随之大幅增加。值得注意的是，与其他区块链相比，Algorand 的恢复速度也更慢；丢包停止后，它需要更长时间才能恢复到先前的吞吐量水平。恢复延迟进一步加剧了观察到的延迟波动性：在丢包期间提交的交易在最终确认前会累积额外延迟。

表 2: $\lfloor f+1 \rfloor$ 组中一个节点在 50% 丢包攻击前、期间和之后的平均点对点带宽比较

	平均节点发送速率（KiB/s）			平均节点接收速率（KiB/s）		
区块链	之前	期间（% Δ）	之后（% Δ）	之前	期间（% Δ）	之后（% Δ）
Algorand	1,480	67（-95.5%）	1,373（-7.3%）	1,550	59（-96.2%）	1,483（-4.4%）
Aptos	112	18 (-83.8%)	68 (-39.0%)	162	16 (-90.1%)	247 (+51.9%)
Avalanche	152	24 (-84.2%)	134（-12.1%）	172	20（-88.5%）	155（-9.5%）
Redbelly	547	24（-95.6%）	571（+4.3%）	534	23（-95.7%）	617（+15.5%）
Solana	840	840（-0.0%）	824（-1.8%）	872	595（-31.8%）	865（-0.7%）

6.3 Algorand 交易丢失

即使丢包率只有 25%，Algorand 也会受到特别大的影响。原因有三。首先，Algorand 通过Remember 函数将交易存储在内存池中，并将待发送的交易存储在广播队列中：丢包会导致交易滞留在广播队列中无法处理，并产生如下类型的日志消息：HTTP 400: "message":"broadcast queue full"。广播队列满时，无法入队的新消息会被丢弃。其次，我们观察到，连接恢复后，Algorand 还需要额外 99 秒才能恢复交易处理，这证实了先前的观察结果 [42]。第三，Algorand 或许能够找回遗漏的交易；然而，其TxSyncer同步协议基于布隆过滤器构建，而布隆过滤器的误报可能导致某些交易未被请求，进而造成前文所述的交易“卡住” [43]。这三种现象共同导致部分交易最终无法提交。

6.4 Solana 的多层弹性

与基于 TCP 的协议相比，Solana 对丢包的抵抗力显著更强（图 7）。这种弹性源于多层策略。首先，它使用 QUIC 而非 TCP，这一选择既提供了可靠性和流量控制，又避免了队头阻塞；在发生丢包时，队头阻塞可能导致数据流停滞。

其次，其 Turbine 区块传播协议直接应对这一问题。Turbine 通过分层树传播区块，并且关键性地采用纠删码将区块拆分为带有额外冗余的片段（“碎片”）。这样，即使传输过程中有相当比例的碎片丢失，验证者仍能重建完整区块。最后，权益加权服务质量（QoS）机制按照验证者质押的比例分配网络带宽，防止垃圾流量造成拥塞，并确保公平地访问资源。总体而言，这些机制使 Solana 天生更能抵御网络中断。

7 易受停止攻击

在本节中，我们研究区块链对“停止攻击”的脆弱性。与多数攻击中验证者可以施加其区块 [72], 、组建大型联盟 [35] 或拥有大部分算力 [7],的情况类似，这些攻击依赖对手控制多数节点，但目的是停止网络。即使故障节点重新加入网络，系统仍无法恢复正常运行，这一点使此类攻击区别于第 5 节所述的瞬态故障攻击。我们观察到，大规模攻击会导致 Solana 在所有节点恢复后仍无限期停止，而 Avalanche 则陷入近停机状态，至少无法恢复瞬态故障期间发起的交易。

7.1 Avalanche 接近停止

我们从第 5.4 节得知，交易nonce管理与限流相结合，会阻止 Avalanche 提交某些交易。图 8 对比了网络中 133rd 至 266th 秒期间，遭受 10% 和 30% 瞬时故障攻击时 Avalanche 的吞吐量随时间的变化；吞吐量是按 2 秒滑动窗口计算的。发生 10% 瞬时故障时，吞吐量显著降低，但并未完全停止。然而，发生 30% 瞬时故障时，吞吐量急剧下降，即使故障节点重启后也未能恢复。这是丢失的交易、nonce间隙和限流共同作用的结果，它们共同阻止了交易执行。重要的是，这个问题并非 Avalanche 共识机制的根本缺陷，而是其交易池配置造成的。我们发现，增大交易池容量并在恢复期间禁用限流，可以缓解这一特定的近停止漏洞，使系统恢复到此前的吞吐量。这表明，这些因素正是导致观察到的停止现象的原因。

7.2 Solana 停止攻击漏洞

Solana 易受停止攻击影响：其很大一部分节点（例如 90%）同时发生瞬时故障时，会导致交易处理完全且不可恢复地停止，如图 8 所示。这种停滞表现

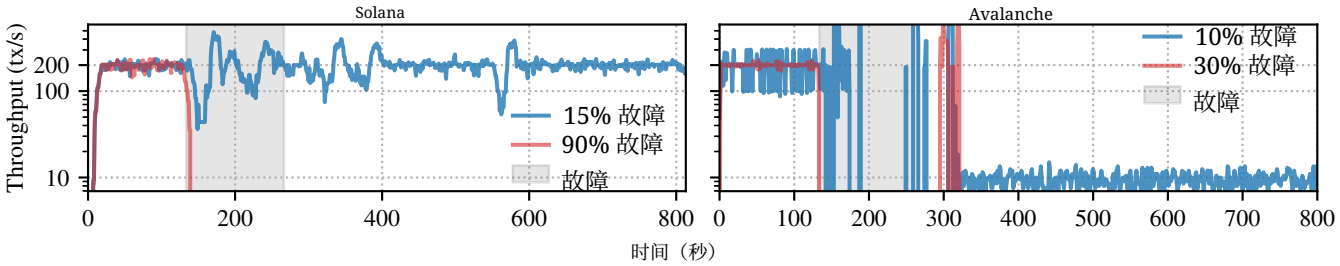


图 8：Solana 在 15% 和 90% 瞬态故障下的吞吐量变化，与 Avalanche 在 10% 和 30% 瞬态故障下的吞吐量变化之间的差异。Solana 会因瞬态故障而崩溃，而 Avalanche 在较少的瞬态故障发生后也无法恢复其初始吞吐量。

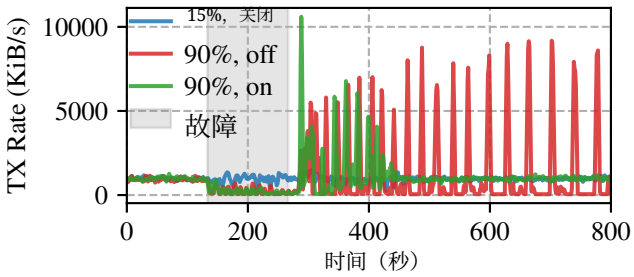


图 9：从客户端接收交易的节点的出站带宽，展现了交易传播模式。未启用该标志时，尚未提交的交易会通过无内存池转发机制持续重新传播，造成持续的带宽峰值。启用该标志后，随着交易积压得到处理，带宽会在恢复后恢复正常。故障率为 15% 时，系统运行正常。

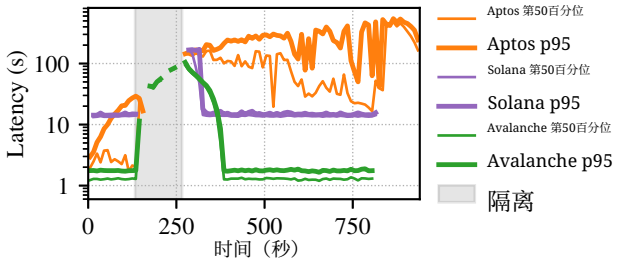


图 10：在领导者隔离期间，根据 Aptos、Solana 和 Avalanche 的交易提交时间，以 10 秒为间隔计算的延迟百分位数（第50百分位为细实线，p95 为粗实线）。线条中的间隙对应没有已提交交易（吞吐量为零）的时间间隔。阴影区域表示领导者隔离时段（133 秒至 266 秒）。

不仅体现为零吞吐量，还体现在特有的网络行为中：图 9 显示，未提交的交易如何通过 Solana 交易转发组件持续循环，造成持续的带宽峰值，只有在恢复之后才会消退。

根本原因是与其投票锁定机制相关的活性停滞。大规模重启后，我们的日志分析发现，新当选的领导者被设定为等待超级多数的“已确立根的投票”，因此拒绝生成区块。然而，由于超级多数节点处于离线状态，少数幸存节点无法独自生成已确立根的区块，网络因此陷入死锁。重启后的节点等待着一个无法满足的条件，导致系统无限期停滞。

我们确认，该漏洞并非共识协议的根本问题，而是源于默认配置。启用--no-wait-for-vote-to-start-leader 标志后，领导者无需观察到近期已确立根的投票即可开始生产区块，网络在攻击后成功恢复。这表明，在大规模恢复事件期间，严格依赖已确立根的投票会造成单点故障。

7.3 抵御停止攻击

此外，我们在 Algorand、Aptos 和 Redbelly 上运行了相同的停止攻击，并测试了网络中遭受瞬态故障攻击的节点比例不同的情况。我们有两点观察：

1. 在我们的测试条件下，Algorand、Aptos 和 Redbelly 均证明能够抵御停止攻击：即使遭受 95% 的停止攻击，这些区块链也能在节点重启后恢复活性，并提交新发起的交易。

2. 经历至少 25% 的瞬态故障后，Avalanche 几乎会停止运行；而 Solana 在经历至少 85% 的瞬态故障后会停止运行。在我们的实验中，Solana 能够容忍比例从 15%、20%……到 80% 的瞬态故障。不过，在某些执行情况下，经历 85% 的瞬态故障时它会停止运行。90% 和 95% 的瞬态故障情况也一样。

8 对领导者隔离攻击的漏洞

我们提出一组新的攻击，称为领导者隔离攻击，即每次将一个单独的节点与网络的其余部分隔离开来。它类似于隔

离针对比特币 [44],以太坊 [83]和 Monero [74]实施的日蚀攻击，但不同的是，它以底层共识协议的潜在领导者为目标。这些攻击的目的在于确定：延迟执行共识协议的单个节点的通信，是否能够在无需大规模网络分区的情况下扰乱区块链的进展。

我们观察到，这种定向隔离会限制 Aptos 和 Solana 的进展，并降低 Avalanche 的性能。我们省略了 Algorand 和 Redbelly 的结果，因为这两个区块链分别采用随机化共识协议和无领导者共识协议。这些共识协议不依赖单个节点或“领导者”的及时通信来协调进展，而是在多个节点上复制这一任务，从而自然地应对单点故障。

8.1 执行领导者隔离攻击

对于每个区块链，我们在统一的实验配置下评估领导者隔离攻击的影响。在总计 800 秒的时间内，以每秒 200 笔交易的持续速率向网络提交交易。

为模拟攻击，我们使用实时日志监控（第 3 节）识别当前的共识领导者。一旦检测到某个节点担任领导者角色，我们便通过注入丢包来实施定向网络性能受损，使用 Linuxtc子系统。具体而言，我们配置一条 netem 规则，在领导者任期内丢弃领导者节点 75% 的入站和出站数据包。

攻击窗口从实验开始后的第 133 秒起，至第 266 秒结束。在此窗口之外，网络条件保持不变。攻击的选择性确保网络的其余部分不受影响，使我们能够评估领导者连接受损对共识进展和交易处理的影响。

8.2 Aptos 停止运行并难以恢复

如图 10 所示，只要领导者与 Aptos 隔离，Aptos 就会停止运行。更准确地说，攻击发生前，Aptos 的中位数和 p95（即第 95th 百分位数）延迟相对稳定。然而，已提交交易的数量在攻击开始后不久便降至零，并一直维持为零，直到攻击停止。这从 Aptos 的第50百分位和 p95 延迟曲线在第 133 秒后几秒出现中断、并持续至第 266 秒可以看出。图 11 清楚地展现了此次停摆及随后恢复困难所造成的网络层面的影响。虽然

领导者被隔离时，网络活动很少。然而，攻击在 266 秒结束后，节点的传输（TX）速率急剧飙升，并在实验剩余时间内保持高位。我们对输出日志的调查证实，Aptos 已停止运行。首先，我们发现一条 INFO消息，指出 Quorum Store 的背压已恢复，这表明交易池接收

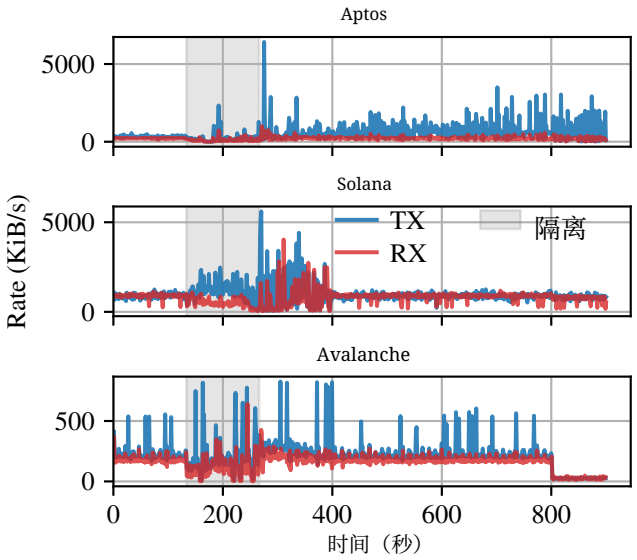


图 11：单个节点从客户端接收交易时的 TX 与 RX 速率。攻击后，Aptos 的 TX 速率出现大幅且持续的峰值，这与图 10 中较高的恢复延迟相吻合。Solana 在迅速恢复正常水平前出现短暂的活动激增，表明其能高效处理积压。攻击期间，Avalanche 的带宽仅受到中等程度的影响，并平稳恢复，这与其不停机的特性一致。

交易的速度超过了其执行速度。其次，消息正在执行区块，交易数量 在攻击前后都有记录，但攻击期间停止记录，证明相应的进展已停滞。总之，这些实验证实，只需隔离领导者就足以使 Aptos 服务停止运行。

请注意，Aptos基于声誉的领导者选举无法提供帮助，因为该攻击可以预测并针对每一位后续领导者。具体而言，隔离当前领导者会形成恶性循环：由于被隔离的领导者无法正确提案，其提案失败次数会增加，从而提高其失败率。当其失败率达到10%时，该节点在即将到来的轮次中被新选为领导者的概率会除以1000。由于该选举是确定性的，对新当选领导者的持续攻击会通过逐一降低每位潜在领导者的声誉来抵消基于声誉的选举，从而促使整个Aptos网络停止。

8.3 Solana停止但迅速恢复

与 Aptos 类似，图 10 表明，当其领导者被隔离时，Solana 会停止运行。这体现为在攻击期间（阴影区域内），第50百分位和 p95 延迟曲线完全消失。该攻击按照 Solana 的领导者调度表（参见第 2.5 节）依次针对每位领导者，并阻止 Turbine 协议（参见第 6.4 节）传播区块以及接收 Tower

BFT投票。我们在日志中观察到这一问题，表现为新根消息消失，这表明缺少来自超级多数的投票，而这些投票是最终确定区块并推进区块链根所必需的。此外，正在等待将投票切换至... 日志消息的出现表明，验证者观察到更重分叉，却无法对其做出承诺或取得进展。

领导者隔离攻击停止后，Solana 的延迟曲线重新出现。调度表中的下一位指定领导者能够成功创建区块，Tower BFT 共识协议也随之恢复。由于需要处理之前积压的交易，延迟，尤其是 p95，会急剧上升。我们对交易时间戳的观察表明，攻击期间提交的交易会经历约等于攻击持续时间加上处理时间的延迟，即 $\simeq 150$ 秒，这也促成了这一峰值。清空积压后，Solana 迅速恢复，表现为 p95 接近 p50。这种快速恢复也反映在 Solana 的网络带宽概况中，如图 11 所示。隔离期结束后，随着新领导者传播区块并清空积压，TX与RX速率均会激增。重要的是，这种活动激增只持续很短时间，网络流量很快便恢复到攻击前的水平。总而言之，与 Aptos 类似，Solana 的领导者一旦被隔离便会停止运行。然而，与 Aptos 以及 Quorum Store 引发的负载（参见第 4.2 节）不同，Solana 在领导者隔离攻击停止后能够迅速恢复，延迟也会接近正常水平。

8.4 Avalanche 在延迟增加的情况下持续推进

与 Aptos 和 Solana 不同，Avalanche 在隔离攻击期间并不会完全停止。事实上，图 10 显示，尽管延迟曲线受到中断，但仍出现在攻击时段的阴影区域内。这是由于 Avalanche 的“软”提议者机制（参见第 2.3 节）：指定提议者因攻击而未能在各自的集体窗口内生成区块时，协议允许任何活跃验证者在窗口结束后提出区块。由于我们针对 Avalanche 的隔离攻击会在检测到某个验证者成为提议者后立即对其实施网络性能受损，并持续 5 秒（这与 Avalanche 的默认 WindowDuration 相对应），因此其他验证者有机会在未受网络性能受损的情况下成功传播区块。这就是性能受到影响、但进程并未停止的原因。图 11 中的网络带宽数据进一步支持了这一观察：尽管进展有所退化，但仍在继续。与 Aptos 和 Solana 不同，Avalanche 不会停止通信；相反，其 TX与RX速率仅在隔离期内适度下降。攻击结束后，流量并未出现明显峰值。网络平稳恢复正常运行状态，这表明即使指定提议者离线，Avalanche 协议仍能继续处理工作，从而成功避免交易积压大量累积。

攻击生效时，任何被识别为正尝试提出区块的节点——无论是在其窗口内行动的指定提议者，还是 任何 在窗口到期后尝试构建区块的验证者——都会遭受第 8.1 节所述的显著网络性能受损。这种性能受损会妨碍受影响节点成功传播其区块并收集必要确认。攻击期间的日志证实了通信困难，其中的消息表明请求被 启用限流 或 超时，这很可能是诱发的丢包扰乱网络交互所致。我们观察到，攻击期间第50百分位和 p95 延迟均有所增加，并且在攻击结束后仍保持较高水平。这表明，在区块生产效率低下且受到影响的时段，可能产生了持续的网络影响或积压。

9 讨论与应对措施

本节讨论 Avalanche 的燃气费管理以及 Solana 预热的有趣之处和应对措施；当故障发生时，后者会直接影响服务。

9.1 燃气费对 Avalanche 性能的影响

Avalanche C 链的 Gas 费用是动态的 [14]。最初的动态费用（Apricot 阶段 3 [67]）随着 Snowman++和基于区块的费用（阶段 4 [66]）不断演进，最终将燃气目标值提高到每 10 秒 15M Gas（阶段 5 [65]），以提升可扩展性 [12, 87]。虽然这对主网有益，但在我们持续进行的每秒 200 笔交易基准测试中，这种动态费用机制引发了问题：目标燃气消耗量持续超过 15M，导致基础费用不断上涨，直至无法再提交交易 [13]。如第 3.7 节所述，我们找到了一种简单的应对措施，即绕过这项递增费用。我们还在第 6 至 8 节中通过实验验证了这一应对措施。因此，对 Avalanche 在攻击下的性能应结合这一特定创世配置来解读；该配置对于深入探究超越初始费用相关停顿的协议行为是必要的。

9.2 避免 Solana 预热期间发生崩溃

我们注意到，我们在第 7.2 节发现的 Solana 全局中断问题会影响当前可用源代码最新版本的预热阶段。不过，我们也注意到，生产环境中的版本并不存在这个问题。

更具体地说，Solana 默认的本地网络部署脚本启用了预热，这使网络以较小的纪元大小启动，并逐渐增至正常大小。这导致网络

因引入故障而崩溃，这是由于 [42] 中详细说明的哈希计算所致。

尽管我们在第 7.2 节发现的停机问题依然存在，导致 Solana 在瞬时故障下无法保持活性，但一个简单的应对措施是禁用预热。具体而言，我们通过实证验证，采用该应对措施后，第 7.2 节所述的 Solana 全局中断不再发生。

10 相关工作

我们的丢包攻击与针对比特币 [44],以太坊 [61]和 ZLB [68], 专门设计的各种分区攻击类似，但我们的方法可以精确控制丢包比例。我们的领导者隔离攻击基于一项已知观察：基于领导者的共识协议容易受到单个节点故障的影响 [5], ，而这种中心化已知会使区块链面临风险 [80]。这种领导者隔离可视为针对比特币的日蚀攻击的一种变体 [44]，但适用于基于领导者的区块链。我们的定向负载攻击与某些拒绝服务攻击有相似之处 [45], ；不过，前者会使 Aptos 过载，而后者会分割以太坊。其他拒绝服务攻击针对交易执行逻辑和手续费市场 [84], 、RPC 服务 [51], 、以太坊的交易池 [53], 放大攻击 [79], ，以及特定协议漏洞，例如 TON 的 ADNL [36]。我们的研究与这些工作互为补充，重点关注更广泛的网络层攻击，以比较多种区块链的基础通信韧性。

网络拓扑在区块链韧性中发挥着至关重要的作用。研究 [30, 31, 52, 89] 已深入揭示了各种区块链的网络结构。尽管我们的研究不涉及拓扑发现，但这些工作凸显了可能影响我们所研究攻击效果的网络特征。

直接在隔离的区块链网络上开展漏洞实验正变得越来越普遍。研究人员已在运行采用工作量证明的原始以太坊协议的机器网络上验证了区块链异常攻击 [62] 和余额攻击 [63] 。STABL 框架 [42]会向区块链网络注入故障，以评估其“敏感性”；然而，该敏感性指标不足以用于我们的比较：(i) 无论故障相较于基线是改善还是恶化性能，该指标都会以相同幅度增加；(ii) 区块链一旦损失一笔交易，该指标就会变为无穷大。

复制状态机的可靠性此前通过注入崩溃 [81] 或拜占庭故障 [2, 17, 50, 75] 进行评估，但未考虑整体区块链协议。区块链协议的可靠性此前通过注入拜占庭故障 [28, 68], 、注入系统调用故障 [88], 、使用模糊测试 [23, 54, 55, 82, 86, 91] 或注入网络延迟 [37, 68]进行评估。它们无法在相同标准下比较不同区块链的容错能力。

大多数区块链协议的比较通常侧重于评估无攻击 [32, 37, 41, 48, 56, 60]情况下的性能。它们并未衡量区块链协议在对抗性条件下的漏洞。近期研究结果发现了 Avalanche [3] 和 Solana [76]的局限性。这些观察结果与我们的结果不同。在 [3] 中，作者对 Avalanche 进行了概率分析，但未明确说明某种攻击或其 Gas 费用会如何影响其进展。在 [76]中，作者演示了 Solana 中的单个恶意领导者如何使协议停滞，但未衡量消息丢失或瞬态故障的影响。在 [42] 中，作者展示了如何使 Solana 崩溃；然而，这种崩溃直接源于公开可用的脚本，无法推广到生产环境中的版本，正如我们在第 9.2 节中所解释的那样。

11 结论

本文首次对区块链通信协议进行了比较性漏洞分析。我们研究了五种现代区块链：Algorand、Aptos、Avalanche、Redbelly 和 Solana，考察了定向负载、瞬态故障、丢包、停止和领导者隔离攻击，这些研究结果形成了第 2 页表 1 的概览。更具体地说，我们的研究发现：由于验证者瓶颈，Aptos 的延迟在定向负载下会严重增加；Avalanche 和 Aptos 在瞬态故障下会损失交易；与采用纠删码的基于 QUIC 的协议相比，基于 TCP 的协议更容易受到丢包攻击；大量瞬态故障可能导致 Solana 崩溃，而 Avalanche 也表现出接近停止的漏洞；领导者隔离攻击会使 Aptos 和 Solana 停止运行，并降低 Avalanche 的性能，凸显了基于领导者的设计所面临的风险。我们还提出了避免已识别中断的对策：为应对 Avalanche 的中断，可以限制基础费用的增长；为应对 Solana 的中断，可以禁用其预热机制。我们计划将框架开源，并测试其他区块链。

致谢

本研究获得澳大利亚研究理事会探索项目资助计划的支持（项目编号 250101739），项目名称为“资源去中心化访问的公平排序”。

参考文献

[1] Algorand。Algorand 节点类型。访问日期：2025年3月7日。网址：https://developer.algorand.org/docs/run-a-node/setup/types/.[2] Mohammad Javad Amiri、Chenyuan Wu、Divyakant Agrawal、Amr El Abbadi、Boon Thau Loo 和 Mohammad Sadoghi。拜占庭故障

容错：用于 BFT 协议分析、实现和实验的统一平台。载于 *USENIX NSDI* 会议论文集，第 371–400 页，2024 年。

[3]Ignacio Amores-Sesar、Christian Cachin 和 Philipp Schneider。Avalanche 共识分析。载于*SIROCCO* 会议论文集，第 27–44 页，2024。

[4] Emmanuelle Anceaume、Antonella Del Pozzo、Romaric Ludinard、Maria Potop-Butucaru 和 Sara Tucci Piergiovanni。区块链抽象数据类型。载于*ACM SPAA* 会议论文集，第349–358页，2019。

[5] KarolosAntoniadis、JulienBenhaim、Antoine Desjardins、Elias Poroma、Vincent Gramoli、Rachid Guerraoui、Gauthier Voron 和 Igor Zablotchi。无领导者共识。《并行与分布式计算杂志》，176:95–113，2023。

[6] Anza。Turbine 区块传播。访问日期：2025年3月7日。URL：https://docs.anza.xyz/consensus/turbine-block-propagation。

[7] Fredy Andres Aponte-Novoa、Ana Lucila Sandoval Orozco、Ricardo Villanueva-Polanco 和 Pedro Wightman。区块链上的 51%攻击：一项挖矿行为研究。*IEEEAccess*，9:140549–140564，2021。

[8] Maria Apostolaki、Aviv Zohar 和 Laurent Vanbever。劫持比特币：针对加密货币的路由攻击。载于 *IEEE S&P*会议论文集，第 375–392 页，2017。

[9] Aptos。Aptos 区块链：安全、可扩展且可升级的 Web3 基础设施，2022。访问日期：2024年8月31日。网址：https://aptosfoundation. org/whitepaper/aptos-whitepaper_en.pdf。

[10]Aptos。节点网络与同步，2025。访问日期：2025年3月7日。网址：https://aptos.dev/en/network/blockchain/node-networks-sync。

[11] Aptos. 节点要求，2025。访问日期：2025年3月7日。网址：https://aptos.dev/en/network/nodes/validator-node/node-requirements.[12] Avalanche。Avalanche L1 区块链浏览器。访问日期：2025年3月7日。网址：https://subnets.avax.network/stats/.[13]Avalanche。C 链。访问日期：2025年3月7日。网址：https://docs.avax.network/nodes/chain-configs/c-chain#rpc-tx-fee-cap.[14]Avalanche。Gas 费用如何计算？访问日期：2025年3月7日。网址：https: //support.avax.network/en/articles/6169826-how-are-gas-fees-calculated.[15] Avalanche。节点与验证者。访问日期：2025年3月7日。网址：https://build.avax.network/docs/ nodes.

[16] Avalanche。AvalancheGo 配置和标志，2024。访问日期：2024年8月31日。网址：https://docs.avax.network/nodes/configure/configs-flags.

[17]Shehar Bano、Alberto Sonnino、Andrey Chursin、Dmitri Perelman、Zekun Li、Avery Ching 和 Dahlia Malkhi。Twins：增强拜占庭容错系统的鲁棒性。载于*OPODIS* 论文集，第 7:1–7:29 页，2022。

[18] NathalieBertrand、VincentGramoli、Marijana Lazi´c、Igor Konnov、Pierre Tholoniat 和 Josef Widder。区块链共识的整体验证。载于*DISC* 论文集，2022。

[19] Tom Brewster。黑客劫持比特币矿池，获利 8.4 万美元，2014。https://www.theguardian.com/technology/2014/aug/07/hacker-bitcoin-mining-pool-internet-service-providers- canada-dell。

[20]Vitalik Buterin。对余额攻击的回应，2017。访问日期：2025 年 6 月 2 日。URL：https://www.reddit.com/r/ethereum/comments/5rcm4o/comment/dd6b4je/。

[21] Miguel Castro 和 Barbara Liskov。实用拜占庭容错。载于 *OSDI* 会议论文集，第 173–186 页，1999 年。

[22]Rajasekhar Chaganti、Rajendra V. Boppana、Vinayakumar Ravi、Kashif Munir、Mubarak Al-mutairi、Furqan Rustam、Ernesto Lee 和 I mran Ashraf。区块链生态系统中拒绝服务攻击的综合评述及未解决的挑战。*IEEEAccess*，第 10 卷，2022 年。

[23]Yuanliang Chen、Fuchen Ma、Yuanhang Zhou、Yu Jiang、Ting Chen 和 Jiaguang Sun。Tyr：利用行为分歧模型发现区块链系统中的共识失效漏洞。载于 *IEEE S&P* 会议论文集，第 2517–2532 页，2023 年。

[24]Ryan Chern。Turbine：Solana 上的区块传播，2023 年。访问日期：2025年3月7日。网址：https://www.helius.dev/blog/turbine-block-propagation-on-solana。[25] Brian Cho 和 Alexander Spiegelman。Quorum Store：共识如何在 Aptos 区块链上实现横向扩展，2023年。访问日期：2025年3月7日。网址：https://medium.com/aptoslabs/quorum-store-how-consensus-horizontally-scales-on-the-aptos-blockchain-988866f6d5b0。[26] Mauro Conti、Ankit Gangwal 和 Michele Todero。区块链三难问题的解决方案 Algorand 在不可判定消息方面面临困境。发表于*ARES* 会议论文集，2019年。[27]Tyler Crain、Vincent Gramoli、Mikel Larrea 和 Michel Raynal。DBFT：高效的无领导者拜占庭共识及其在区块链中的应用。发表于 *IEEE NCA* 会议论文集，第 1–8 页，2018年。

[28]Tyler Crain、Christopher Natoli 和 Vincent Gramoli。Red Belly：一种安全、公平且可扩展的开放式区块链。发表于 *IEEE S&P* 会议论文集，2021年。

[29] Cryptodaily。Solana 网络中断之际，$TRUMP、$ MELANIA 及其他基于 Solana 的模因币飙升，2025。访问日期：2025年1月20日。网址：https://cryptodaily.co.uk/2025/01/solana-network-outages-as-trump-melania-and-other-solana-based-memecoins-surge。

[30] Vincenzo Di Perna、Marco Bernardo、Francesco Fabris、Sebastiao Amaro、Miguel Matos 和 Vale- rio Schiavoni。网络拓扑对区块链性能的影响。发表于*ACM DEBS* 论文集，2025。

[31]Vincenzo Di Perna、Valerio Schiavoni、Francesco Fabris 和 Marco Bernardo。区块链能耗：揭示网络拓扑的影响。发表于*IEEE ICBC* 论文集，2025。

[32] Tien Tuan Anh Dinh、Ji Wang、Gang Chen、Rui Liu、Beng Chin Ooi 和 Kian-Lee Tan。BLOCKBENCH：用于分析私有区块链的框架。发表于*ACM SIGMOD* 论文集，第 1085–1100 页，2017。

[33]Cynthia Dwork、Nancy Lynch 与 Larry Stockmeyer。部分同步条件下的共识。*J. ACM*，35(2):288–323，1988。

[34] Parinya Ekparinya、Vincent Gramoli和Guillaume Jourjon。中间人攻击对以太坊的影响。载于*Proc. IEEE SRDS*，第11–20页，2018。[35] Ittay Eyal和Emin G¨un Sirer。多数并不足够：比特币挖矿是脆弱的。*Commun.ACM*，61(7):95–102，2018。[36] Aviv Frenkel和Dmitry Kogan。针对TON的ADNL安全信道协议的攻击。载于*Proc.IEEE S&P*，2025。[37]Frank Christian Geyer、Hans-Arno Jacobsen、Ruben Mayer和Peter Mandl。七种许可型区块链系统的端到端性能比较。载于*Proc.Middleware*，第71–84页，2023。[38] Yossi Gilad、Rotem Hemo、Silvio Micali、Georgios Vlachos和Nickolai Zeldovich。Algorand：面向加密货币扩展拜占庭协议。载于*Proc.ACM SOSP*，第51–68页，2017。[39] Giacomo Giuliari、Alberto Sonnino、Marc Frei、Fabio Streun、Lefteris Kokoris-Kogias和Adrian Perrig。共识协议拒绝服务攻击韧性的实证研究。载于*Proc. ACMASIACCS*，第1345–1360页，2024。[40] Vincent Gramoli。区块链可扩展性及其在分布式系统中的基础。Springer，2022。

[41] Vincent Gramoli、Rachid Guerraoui、Andrei Lebedev、Chris Natoli和Gauthier Voron。Diablo：面向区块链的基准测试套件。载于*Proc. EuroSys*，第540–556页，2023。

[42] Vincent Gramoli、Rachid Guerraoui、Andrei Lebedev 与 Gauthier Voron。Stabl：区块链对故障的敏感性。载于*Middleware* 会议论文集，第 202–214 页，2025。

[43] JuanGranados。网络：处理交易广播失败 · 议题 #5641 · algorand/go-algorand，2023。访问日期：2025年3月7日。URL：https:// github.com/algorand/go-algorand/issues/5641。

[44] Ethan Heilman、Alison Kendler、Aviv Zohar 与 Sharon Goldberg。对比特币点对点网络的日蚀攻击。载于*USENIX* 安全会议论文集，第 129–144 页，2015。

[45]Hwanjo Heo、Seungwon Woo、Taeung Yoon、Min Suk Kang 和 Seungwon Shin。以太坊分区而不使其陷入隔离。在*NDSS* 会议录，2023 年。

[46] 互联网工程任务组。RFC 9000 – QUIC：一种基于 UDP 的多路复用安全传输协议，2021 年。

[47]Muhammad Anas Imtiaz、David Starobinski、Ari Trachtenberg 和 Nabeel Younis。比特币网络中的 churn。*IEEE* 网络与服务管理汇刊，18(2):1598–1615，2021 年。

[48]Dave Kelsey。Hyperledger Caliper，2024。访问日期：2024年8月31日。网址：https://hyperledger.github.io/caliper/.[49] Leslie Lamport、Robert Shostak 和 Marshall Pease。拜占庭将军问题。*ACM Trans. Pro-gram.Lang. Syst.*，4(3):382–401，1982。[50] Hyojeong Lee、Jeff Seibert、Endadul Hoque、Charles Killian 和 Cristina Nita-Rotaru。Turret：一个用于在未经修改的分布式系统实现中进行自动化攻击发现的平台。载于*Proc. IEEEICDCS*，第660–669页，2014年。[51] Kai Li、Jiaqi Chen、Xianghong Liu、Yuzhe Tang、Xi- aoFeng Wang 和 Xiapu Luo。坚如其最薄弱的一环：如何通过 RPC 服务攻破区块链去中心化应用。载于*Proc. NDSS*，2021年。[52] Kai Li、Yuzhe Tang、Jiaqi Chen、Yibo Wang 和 Xi- anghong Liu。Toposhot：利用替换交易揭示以太坊网络拓扑。载于*Proc. ACM互*联网测量会议，第302–319页，2021年。[53] Kai Li、Yibo Wang 和 Yuzhe Tang。Deter：以太坊交易池服务拒绝。载于*Proc. ACM CCS*，第1645–1667页，2021年。[54] Fuchen Ma、Yuanliang Chen、Meng Ren、Yuanhang Zhou、Yu Jiang、Ting Chen、Huizhong Li 和 Ji- aguang Sun。LOKI：一种状态感知模糊测试框架

用于区块链共识协议的实现。载于*Proc. NDSS*，2023年。

[55]Fuchen Ma、Yuanliang Chen、Yuanhang Zhou、Jingxuan Sun、Zhuo Su、Yu Jiang、Jiaguang Sun 和 Huizhong Li。Phoenix：通过上下文敏感混沌检测并定位区块链韧性问题。载于*ACM CCS* 会议论文集，第 1182–1196 页，2023 年。

[56] Liyuan Ma、Xiulong Liu、Yuhan Li、Chenyu Zhang、Gaowei Shi 和 Keqiu Li。GFBE：一种通用且细粒度的区块链评估框架。*IEEE* 计算机汇刊，73(3):942–955，2024 年。

[57]Michael Mirkin、Yan Ji、Jonathan Pang、Ariah Klages-Mundt、Ittay Eyal 和 Ari Juels。Bdos：区块链拒绝服务攻击。载于*ACM CCS* 会议论文集，第 601–619 页，2020 年。

[58] Saeideh G. Motlagh、Jelena Mi˘si´c 和 Vojislav B. Mi˘si´c。节点 churn 对比特币网络的影响。*IEEE* 网络科学与工程汇刊，7(3):2104–2113，2020 年。

[59]Michael Nadeau。《Avalanche 基础》，2023年。访问日期：2025年3月7日。网址：https://tokenterminal.com/crypto-research/avalanche#decentralization。

[60] Bulat Nasrulin、Martijn De Vos、Georgy Ishmaev 和 Johan Pouwelse。《Gromit：区块链系统性能与可扩展性的基准测试》。载于*IEEE DAPPS* 会议论文集，第 56–63 页，2022年。

[61] Chris Natoli、ParinyaEkparinya、Guillaume Jourjon 和 Vincent Gramoli。《低挖矿算力和网络延迟下的区块链双重支付》。《*ACM* 分布式账本技术》，3(4)，2024年。

[62]Christopher Natoli 和 Vincent Gramoli。《区块链异常攻击》。载于*IEEE NCA* 会议论文集，第 310–317 页，2016年。

[63] Christopher Natoli 和 Vincent Gramoli。《平衡攻击：可分叉区块链为何不适用于联盟》。载于*IEEE DSN* 会议论文集，第 579–590 页，2017年。

[64] Joachim Neu、Ertem Nusret Tas 和 David Tse。《潮汐协议：解决可用性与最终性困境》。载于*IEEE S&P* 会议论文集，第 446–465 页，2021 年。[65] Patrick O'Grady。《Apricot第五阶段：P<>C 原子转账、原子交易批处理和 C 链费用算法》……，2021 年。访问日期：2025年3月7日。网址：https://medium.com/avalancheavax/apricot-phase-five-p-c-atomic-transfers-atomic-transaction-batching-and-c-chain-fee-algorithm-912507489ecd.[66]Patrick O'Grady。《Apricot 第四阶段：Snow- man++和降低 C 链交易费用》，2021 年。访问日期：2025年3月7日。网址：https://medium.com/avalancheavax/apricot-phase-four-snowman-and-reduced-c-chain-transaction-fees-1e1f67b42ecf.

[67] Patrick O'Grady。Apricot 第三阶段：C 链动态费用，2021。访问日期：2025年3月7日。网址：https://medium.com/avalancheavax/apricot-phase-three-c-chain-dynamic-fees- 432d32d67b60。

[68]Alejandro Ranchal-Pedrosa 和 Vincent Gramoli。ZLB：一种能够容忍共谋多数的区块链。载于*IEEE DSN* 会议论文集，第 209–222 页，2024。

[69]Redbelly。节点硬件规格要求 — Vine（Redbelly 开发者门户）。访问日期：2025年3月7日。网址：https://vine.redbelly.network/nds-node-hardware-specification-requirements。

[70]Jamie Redman。Bitcoin Gold 遭遇 51%攻击——网络因双重支付损失 70,000 美元，2020。访问日期：2025年8月26日。网址：https://news.bitcoin.com/bitcoin-gold-51-attacked-network-loses-70000-in-double- spends/。

[71] Team Rocket、Maofan Yin、Kevin Sekniqi、Robbert van Renesse 和 Emin G¨un Sirer。通过亚稳态实现可扩展且概率化的无领导者 BFT 共识。技术报告，arXiv，2020 年。

[72] Meni Rosenfeld。基于哈希率的双重支付分析，2012 年。

[73] Muhammad Saad、My T Thai 和 Aziz Mohaisen。海报：通过内存池优化遏制区块链加密货币上的 DDoS 攻击。载于*ACM ASIACCS* 会议论文集，第 809–811 页，2018 年。

[74] Ruisheng Shi、Zhiyuan Peng、Lina Lan、Yulian Ge、Peng Liu、Qin Wang 和 Juan Wang。针对 Monero 点对点网络的日蚀攻击。载于*NDSS* 会议论文集，2025 年。

[75] Atul Singh、Tathagata Das、Petros Maniatis、Peter Druschel 和 Timothy Roscoe。遭受攻击的 BFT 协议。载于*USENIX NSDI* 会议论文集，第 189–204 页，2008 年。[76] Jakub Sliwinski、Quentin Kniep、Roger Wattenhofer 和 Fabian Schaich。以 epsilon 质押使 Solana 区块链停止运行。载于 *ICDCN* 会议论文集，第 45–54 页，2024 年。[77] Solana。Solana 领导者轮换，2024 年。访问日期：2024年8月31日。网址：https://docs.solanalabs.com/consensus/leader-rotation。[78]Deepal Tennakoon、Yiding Hua 和 Vincent Gramoli。Smart Redbelly 区块链：降低 Web3 的拥塞。载于 *IEEE IPDPS* 会议论文集，第 940–950 页，2023 年。

[79] Taro Tsuchiya、Liyi Zhou、Kaihua Qin、Arthur Gervais 和 Nicolas Christin。区块链放大攻击。载于*ACM SIGMETRICS* 会议论文集，2025 年。[80]Richard von Seck、Filip Rezabek、Sebastian Gallenmüller 和 Georg Carle。网络传输协议对基于领导者的共识通信的影响。载于*ACM BSCI*，第 1–11 页，2025 年。[81]Gauthier Voron 和 Vincent Gramoli。Dispel：采用分布式流水线的拜占庭状态机复制。技术报告 1912.10367，arXiv，2020 年。[82]Levin N. Winter、Florena Buse、Daan De Graaf、Klaus Von Gleissenthall 和 Burcu Kulahcioglu Ozkan。拜占庭容错算法的随机化测试。*ACM 程序语言*，7：757–788，2023 年。[83] Karl Wust 和 Arthur Gervais。以太坊日蚀攻击。技术报告，苏黎世联邦理工学院，2016 年。[84] Aviv Yaish、Kaihua Qin、Liyi Zhou、Aviv Zohar 和 Arthur Gervais。以太坊中的投机性拒绝服务攻击。载于*USENIX* 安全会议，2024 年。[85] Anatoly Yakovenko。Solana：高性能区块链的新架构 v0.8.13，2021 年。访问日期：2024年8月31日。网址：https://solana.com/solana-whitepaper.pdf。[86] Youngseok Yang、Taesoo Kim 和 Byung-Gon Chun。通过多交易差分模糊测试发现以太坊中的共识漏洞。载于 *USENIX OSDI* 会议论文集，第 349–365 页，2021 年。[87]Charles Yu。RL1：深入了解 Avalanche，2022 年。访问日期：2025年3月7日。网址：
h
ttps://
www.galaxy.com/insights/research/ready-layer-one-avalanche/。[88] Long Zhang、Javier Ron、Benoit Baudry 和 Martin Monperrus。以太坊区块链客户端的混沌工程。*ACM* 分布式账本技术研究与实践，2(3)，2023 年。[89] Chonghe Zhao、Yipeng Zhou、Shengli Zhang、Taotao Wang、Quan Z. Sheng 和 Song Guo。Dethna：通过标记交易精确发现以太坊网络拓扑。载于*IEEE INFOCOM* 会议论文集，第 1711–1720 页，2024 年。[90]Terence Zimwara。560万美元双重支付：ETC 团队终于承认网络遭受 51%攻击，2020 年。访问日期：2025年8月26日。网址：
h
t
t
p
s://
n
e
w
s.
b
i
t
c
o
i
n.
c
o
m/
5-
6-
m
i
l
l
i
on-stolen-as-etc-tea
m-finally-acknowledge-the-51-attack-on-network/。[91] Yonghao Zou、Jia-Ju Bai、Zu-Ming Jiang、Ming Zhao 和 Diyu Zhou。采用多维输入和基于对称性的反馈剪枝对分布式系统进行黑盒模糊测试。载于*NDSS* 会议论文集，2025 年。