**An Efficient Blockchain Consensus Protocol in Wireless Networks**

**Abstract**

1. **简介**

随着5G技术和物联网的快速发展，无线网络中通信和网络带来了巨大的发展。在5G技术的支持下，物联网可以实现人与人之间，人与物之间以及物与物之间的交互，从而更好的提供日常服务和商业支持。为了满足多样化应用的需求，物联网的设备将大规模增加。预计到2035年，物联网上连接的设备数量将达到1万亿，这将极大地影响社会经济的发展[1]。物联网的快速发展能为智能家居、公共安全、智慧城市、自动驾驶、远程医疗等应用提供有效的基础设施支撑[2,3，4]。然而，随着物联网设备数量地增加，采用集中式拓扑结构的传统大规模物联网面临着严重的数据安全和信任问题。

区块链技术被认为是可以很好地解决物联网应用中数据安全和信任问题的技术之一[5，6]。区块链技术具有去中心化、不可篡改性和可追溯性等优点，受到了学术界和工业界的广泛关注。区块链系统依赖分布式共识协议确保实现分布式账本的一致性。当不同的物联网设备（通常称为“节点”）以分布式方式通信时，可以实现完全自治的物联网 [2]。因此，物联网需要一种共识算法确保节点可以对通信数据的有效性达成一致。因此，区块链中使用的共识协议也可以应用于物联网。虽然将区块链与物联网结合具有良好的前景，但是也面临着一些新的挑战。大量无线设备的出现使得无线通信成为区块链和物联网结合的阻碍之一。不稳定的信道质量会影响节点之间的通信，从而影响系统中节点达成共识的效率，最终影响区块链的性能（低吞吐量和高时延）。

然而，大多数现有的研究都是在安全可靠的传输假设下设计了共识协议，忽略了无线设备资源是有限的且无线网络具有不稳定、不可预测甚至是易受干扰等特性。众所周知，大多数区块链系统都采用基于工作量证明共识算法[7]作为底层的共识协议。在工作量证明算法中，矿工通过解决一个复杂的数学谜题来竞争出块权限。但是，大部分的无线设备的计算能力都是有限，很难负担沉重的挖矿工作。同时，当对手控制的计算能力不足总计算能力的50%时，基于工作量证明的共识协议依赖“多区块确认”确保概率一致性。因此，由于具有较高的确认时延，基于工作量证明的共识协议的性能通常非常差。作为工作量证明的替代协议，权益证明共识算法[8]不需要较大的算力开销。但是，基于权益证明的共识协议总是需要一个复杂的设计，避免权益或资产集中的同时防止攻击者无成本或以较低成本篡改区块链。基于拜占庭容错的共识协议 (比如Practical Byzantine Fault Tolerance (PBFT) [9])需要频繁的消息传递。这在无线通信中会导致严重的信道竞争和巨大的通信开销。这些共识协议都不是专为无线网络环境设计的。

最近，一些研究工作利用无线网络的特性设计区块链共识协议，使之适用于单跳无线网络通信环境。根据无线广播通信的特点，Z.Jiang等人[10]提出基于女巫证明的拜占庭容错共识协议，该协议能够在无线网络中实现实时共识。针对无线设备的低功耗和无线传输的不稳定性，Q.Xu等人[11]提出一种适用于无线区块链系统的高效、公平的通信证明（Proof-of-Communication）一致性证明协议。Y.Zou等人[12]利用无线网络的广播通信特点提出了适用于许可无线区块链网络的快速共识协议。此外，考虑到无线信道易受干扰的特征，M.Xu 等人提出了可以抵抗干扰攻击的基于 Proof-of-Channel 共识算法的单跳无线区块链共识协议 BLOWN[13]。这些无线区块链共识协议都是考虑在单跳无线网络中快速实现共识。如果直接将这些共识协议用于多跳无线网络中，当网络范围变大时通信开销和功耗都将会急剧增加。因此，为多跳无线网络中的区块链系统设计一个安全、高效的共识协议具有重要意义。

在本文中，我们提出一个适用于多跳无线网络的区块链共识协议-稳定委员会共识协议。我们使用节点的稳定度作为构建共识委员会的阈值，这个委员会将会具有相同的任期，每个任期由多轮组成。在每一轮共识开始之前，使用可验证随机选择方案确保在每一轮随机并且不需要交互的选举出唯一的出块节点。这种方案使得敌手不能提前预测出块节点，可以确保出块节点的安全性。随后，采用门限BLS签名方案作为网络中认可区块有效的投票机制来降低共识过程的通信开销。为了缩短确认时间，每个区块只需要一轮签名就可以完成区块的确认。诚实的节点有较高的概率能够在有限时隙后收集足够的部分签名，并且将他们聚合成一个唯一的完整签名。最终，只需要一个签名验证就可以验证区块的有效性。我们的主要贡献总结如下：

* 我们提出了一个融合随机可验证选举方案和阈值BLS签名方案基于稳定委员会的无线区块链共识协议，该协议可以确保区块链系统在多跳无线网络中稳定地生成区块并达成一致。
* 我们提出一个新的共识算法可以确保选择更为稳定的节点作为委员会成员，并且通过随机的方式选举出块节点降低敌手腐蚀的风险。这种选举方式保障委员会成员有很大概率不会在共识过程中离开网络，提高共识过程的稳定性和安全性。当确定了
* 我们的共识算法将门限BLS签名与谣言通信协议相结合，确保区块确认过程可以通过一轮签名交互就完成。因此，参与共识的节点有较高概率在时隙之后收集足够的部分签名恢复完整签名，对区块的有效性达成共识。其中， 是委员会中任意两个共识节点的最大距离。因此，我们的共识算法不仅具有强一致性，还有较低通信开销。
* 最终，我们通过大量的仿真研究验证我们的理论分析。

本文剩余部分组成如下。下一节介绍区块链共识协议、无线共识算法和门限BLS签名方案等相关工作。第三节提出我们的模型和假设。第四节详细介绍稳定共识协议的细节。第五节讨论了协议的安全性和性能。第六节会给出共识协议相关的仿真结果。最后，在第七节给出了本文的结论。

1. **相关工作**

**2.1 区块链共识协议**

目前，有两种最常用于区块链系统的共识协议。第一种是基于资源证明的共识协议，该协议中共识节点通过物理资源（比如，算力，存储等）或虚拟资源(比如，股份和名誉等）证明方式来竞争获取每一轮的出块权限。最典型的基于物理资源证明的共识算法是比特币使用的工作量证明(Proof of Work， PoW[7])。在比特币中，共识节点通过消耗的计算资源找到一个复杂数学难题的解来证明获得出块权限。但是PoW共识算法并不支持即时共识最终性[14]，在敌手的算力小于总算力 时可以通过多个区块确认来保证协议的概率一致性。因此，采用这类共识算法的区块链协议在共识过程中区块确认的时延比较大，处理交易的吞吐量比较有限。此外，基于物理资源证明的共识算法还有空间证明 (Proof of Space, PoSpace)[15]，共识参与者通过占用的内存或磁盘空间来竞争出块权限；以及燃烧证明（Proof of Burn）[16]，共识参与节点通过烧毁另一种“货币”，比如比特币，来获得生成区块的权限。为了降低共识的能耗和提高效率，虚拟资源证明共识算法被提出代替物理资源证明共识算法。典型的虚拟资源证明的共识算法是的权益证明(Proof of Stake, PoS)[8]，参与共识的节点根据所持有的股份作为选举成为出块节点的权限证明，股份越多的节点成为出块节点的概率就越大。以太坊中的Casper[17]是混合PoW和PoS的共识协议，旨在用PoS取代PoW。此外，虚拟资源证明的共识算法还有信誉证明(Proof of Reputation, PoR)[18]，共识参与节点需要拥有足够的信誉才能获取出块权限。虚拟资源证明共识算法的共识过程并不需要消耗实际的资源，对于环境比较友好。因此，越来越多的区块链共识协议希望使用虚拟资源证明作为共识算法。

第二种是基于协商的共识协议，该协议中所有共识的节点通过执行本地计算和广播消息与其他节点通信协商对提出的区块达成共识。采用这种共识算法的区块链协议可以容忍协议中出现拜占庭故障，还能为区块链共识协议提供强一致性。典型的协商共识算法是HyperLedger Fabric [19]中实现的拜占庭容错算法(Practical Byzantine Fault Tolerant, PBFT[9])。该共识算法从全网节点中选举出块节点负责创建区块，通过与其他节点投票协商对区块达成全网共识。实际拜占庭容错协议无权益抵押或者资源的消耗会降低恶意节点的作恶成本，但是通过节点协作机制可以排除恶意行为对共识的影响。一些基于协商的区块链共识协议通过在确保安全性的情况下提升共识协议的性能。其他协商共识算法还有NEO项目[20]中采用的DBFT算法，授权的共识节点通过投票协商对区块达成共识。在Zyzzyva[21]中，改进的BFT一致性算法SBFT使用阈值签名来减少一致性过程的通信开销。然而，它仍然需要两轮通信来聚合签名和终止块确认。此外，Cosmos[22]的Tendermint 共识算法采用了领导者轮换机制来避免对手的腐败，并使用八卦协议来提高事务传播的可扩展性。Algorand共识协议[23]将拜占庭协议协议与VRF委员会选举方案相结合，以确保区块链共识过程的安全性和可扩展性。大多数基于通信的协商一致协议的协商一致安全性取决于首领的正确性。当首领故障导致共识过程中断时，共识节点只能通过改变视图重新开始共识过程。此外，基于通信的区块链共识协议通常需要可靠的消息传输模型和广播通信。因此，这些区块链共识协议更适合于网络规模小、通信模型可靠的场景。

**2.2 无线网络中的共识协议**

我们的研究与无线网络密切相关且共识算法是区块链协议的核心，在本小节简单介绍一下无线网络中共识算法的研究进展。

为了提高无线网络中的一致性效率，当前已有许多关于无线共识算法的研究工作。M.Zheng等人[24]提出了一种用于集群无线传感器网络中的平均共识的八卦算法, 其中每个集群中的节点利用无线多址信道的自然叠加特性来实现快速平均共识。针对无线网络集群面临的共识问题，M. Goldenbau等人[25]提出了一种可以在集群无线网络中快速实现全局共识的迭代八卦算法。C．Newport和P.Robinson[26]提出了无线系统中的分布式容错共识算法，该算法适用于任何数量的节点故障且保证在多项式网络大小的系统中多次广播后能够高概率终止。F.Molinari等人[27]利用无线信道的广播特性，提出了一个使用广播授权代理的切换能够实现最大共识的共识协议。H. Moniz等人[28] 为资源受限的无线自组织网络提出了一种异步拜占庭共识协议，该协议确保即使系统中消息动态丢失也能快速的达成共识。

近年来，一些研究者将无线共识算法与区块链技术相结合，提出更加适用于无线网络环境的区块链共识协议。根据无线广播通信的特点，Z.Jiang等人[8]提出基于女巫证明的拜占庭容错共识协议，该协议能够在无线网络中实现实时共识。利用无线网络的通信特性，Q.Xu等人[9]提出基于通信（Proof-of-Comunication）的无线区块链共识协议；Y.Zou等人[10]提出了适用于许可无线区块链网络的快速共识协议。此外，考虑到无线信道易受干扰的特征，M.Xu 等人[11]提出了可以抵抗干扰攻击的基于 Proof-of-Channel 共识算法的单跳无线区块链共识协议 BLOWN。这些研究工作大多数是能够在不存在故障的单跳无线网络快速达成一致。针对多跳无线网络场景，M.Xu [23]等人过构建通信骨架的方式提出了适用于多跳无线网络的区块链共识协议wChain，该协议能够容忍故障且快速地完成数据收集。但是，该协议只能容忍节点崩溃故障，并不能在拜占庭故障出现时保证共识协议的安全性。同时，大多数无线区块链共识协议的安全性都依赖首领节点的正确性。首领节点除了生成区块之外，还要负责驱动消息传输过程以及公布最终共识结果的。因此，一旦首领节点发生故障，将直接导致共识失败。

**2.3 门限签名方案**

采用门限签名方案[29]的区块链共识协议可以确保在通信不稳定的无线网络中快速稳定地对区块达成最终共识。门限签名方案允许一组签名者在不知道其他私钥信息地情况下共同构建一个有效地签名。在门限签名方案中，个签名者持有不同的私钥，并且任意大于 个签名者的签名可以构建一个有效的完整签名。所有共识节点通过自己的私钥对共识结果进行签名并广播。同时，所有节点可以收集不同节点对共识结果的部分签名。当不同部分签名的数量达到阈值时，将聚合成一个唯一的完整签名。任何节点都可以通过聚合公钥快速地对完整签名进行验证，即对共识结果进行验证并实现共识。

BLS签名方案[30]是利用循环群和双线性映射的特性来构造聚合签名，实现多方签名和验证。在BLS签名方案包括了签名生成算法和签名验证算法。参与者通过私钥生成消息的签名，其他节点通过签名者的公钥可以验证签名的有效性。门限BLS签名方案[29]源于BLS签名方案，并以非交互的方式执行。其部分签名的生成与BLS签名方案是一致的，每个参与者收集足够地部分签名之后都能独立地恢复完整签名，即完整签名恢复地过程是非交互式和分布式的。门限BLS签名方案包括了密钥生成算法、签名生成算法和签名验证算法。密钥生成算法使用分布式密钥生成协议[31]向参与共识节点分发密钥 和一个聚合主公钥，其中聚合公钥用于验证生成的完整签名。签名生成算法包含了部分签名生成协议和一个完整签名恢复协议。其中，部分签名生成协议与BLS签名方案的相同，用于生成消息的部分签名；完整签名恢复算法将根据多个节点的部分签名计算得到用唯一主私钥签名消息形成的完整签名，即一个拉格朗日插值函数上的特殊值。部分签名验证算法则与BLS签名方案的签名算法一致，但是对于完整签名则需要使用分布式密钥生成协议生成聚合主公钥来进行验证。

-门限BLS签名方案可以保证无线区块链系统共识过程的安全性和高效性。由于聚合签名可以由任意共识节点执行，因此即使一些节点由于故障导致没有收集到足够的部分签名，最终系统也能够恢复完整签名。区块的共识结果并不依赖于出块节点公布。只要诚实的共识节点收集到足够的部分签名聚合形成完整的签名，就能完成区块确认并发布最终结果。这个设计极大地降低了由于出块节点故障或者链路不稳定导致无法达成共识的风险。同时,只需要一轮通信就能完成区块的共识过程，这减少了共识的通信开销；而对于投票结果也只需要进行一次验证，减少了验证共识结果的算力开销。因此，采用门限BLS签名作为投票机制的共识协议可以降低共识过程的系统开销和提升共识的效率。

1. **模型和假设**

**3.1 区块链系统的基本定义**

分布式系统中的经典的共识算法的目标是让系统中所有节点对于提交数据的值达成一致。因此，分布式共识协议通常需要满足以下性质：

* 一致性：所有节点的决策值都相同；
* 终止性：所有正确的节点都会最终做出决定（活性）；
* 有效性：决策的值是由一个节点提出的。

区块链系统也是一种分布式系统，节点将网络中的交易历史记录在区块中，并通过区块哈希值连接成区块链的形式。与分布式共识算法类似，区块链共识算法旨在让所有的区块链中的节点对于交易历史达成一致。我们的目标是在有故障的多跳无线区块链网络中，设计一个区块链共识协议满足下面的性质：

* 安全性：如果某个区块被写入一个正确节点的本地区块链中，该区块也会被写入其他正确节点的本地区块链中，即个诚实节点在相同的区块链高度上拥有相同的区块；
* 活性：对于提出的区块，所有正确的节点将最终决定是将区块丢弃或添加到正确节点的本地区块链上。

在本研究中，针对多跳无线区块链网络面临的共识问题，我们考虑设计一种安全、高效的拜占庭容错的区块链共识协议。

**3.2 网络模型**

我们考虑任意部署在二维欧式空间中由 个节点的集合 组成的多跳无线网络。每个节点具有一个半双工发射器可以发送和接收消息或者感知信道，但是不能同时收发消息或者同时发送和感应信道。网络中任意两个节点和 之间的欧式距离记作。我们假设两个节点之间最小距离为。记无线网络中任意两个节点之间的最大距离为。假设每个节点在进入网络后需要提交一个抗女巫攻击证明后才能获得身份。通过与系统中其他节点通信可以获知当前系统中的节点信息以及区块链历史信息。在我们的协议实现中，我们将时间分为同步的时隙，每个时隙是消息传输的最小时间。在每一个时隙中，任意节点可以决定是传输消息还是侦听信道。每一轮是由多个时隙组成的时间区间，而多个轮可以组成一个任期。在每一个任期结束的时候，将会确认新任期的委员会。一旦确定了委员会，则每个成员将执行一个分布式密钥生成协议获取公私密钥对和聚合主公钥为之后的共识做准备。我们的共识协议依赖可信的密码系统，假设签名和加密过程都是安全可信的。

**3.3 通信模型**

在本文中，我们采用信号干扰噪声(SINR)模型来描述节点之间消息的传输情况。标准信号干扰的信噪比模型为

其中， 是节点 接收到的节点 的信号的功率， 是平均信号发射功率， 为环境噪声，在节点 处的干扰为 。其中， 是在当前传输节点的集合，路径损耗指数为 ，阈值 取决于节点的硬件。我们假设每个节点可以进行物理载波监听，并且节点可以自行决定每次传输时的传输功率。

**3.4 故障模型**

在无线网络中，故障节点的恶意行为的假设如下：

* 故障节点的目标是为了阻止共识的实现而不仅仅是让自己获利；
* 为了实现上述目标，故障节点可能的恶意行为包括不遵循协议并且传输任意消息（错误消息或者垃圾消息等）；
* 在无线网络中，故障节点不会一直阻止或者干扰其他节点的传输和通信。如果一个故障节点拥有较大的传输功率并且不遵循通信协议将很容易被检查到。

1. **稳定委员会无线区块链协议**

在本节，我们首先介绍我们协议的概览。随后，我们详细介绍了稳定委员会共识协议的共识过程。

**4.1 概览**

在多跳无线区块链网络中，参与共识的节点通过提交一个女巫攻击抵抗证明加入系统，即质押一笔金钱注册身份。这笔押金会存放在一个虚拟账户中，除非本人通过一个解质押的方式才能取出属于自己的质押金额；否则任何人不能取出。这个质押机制可以有效的防止敌手发起女巫攻击。每个节点进入系统之后会获取当前的系统中其他节点的公钥和最新的区块链历史，并且将自己的公钥也广播给其他节点。每个节点都需要维护一个相同的节点列表，每个节点使用公钥作为身份。

我们旨在设计一个能够在多跳无线网络中安全、快速、稳定达成共识的区块链共识协议。我们的共识协议将时间分为不同的任期，每个任期开始的时候将选举出一个共识委员会，成员在任期时间内参与共识过程。其他非委员会成员只会在接收到共识结果后更新本地区块链。委员会成员将在每个任期结束之后更新。每个任期是由多个轮组成，每轮中委员会将运行稳定共识协议将一个新的区块被添加到区块链。在每一轮共识中，先随机选举一个出块节点生成区块并广播到全网。随后每个共识节点对区块的有效性进行投票，并随后将区块和签名的集合广播到网络。特别地，节点在一轮共识中只能对共识结果进行一次投票。稳定共识协议中节点对区块Hash的签名被记为对区块有效的一次投票。因此，我们的共识协议正常运行主要分为以下几个阶段：

* 出块节点选举：委员会中的节点独立地运行一个分布式随机生成协议生成一个在 区间的随机数。节点将自己私钥和当前轮的随机数作为可验证出块节点选举协议的输入，查看自己是否成为当前轮出块节点。同时，节点可以根据出块节点的公钥验证出块节点的合法性；
* 区块的生成：出块节点确定自己获得出块权限后，从自己的未处理交易池中取出部分交易打包生成区块，并广播区块到网络中的所有其他节点；
* 区块的验证：当节点接收到区块之后，将执行区块验证过程。首先，节点根据出块节点的公钥和随机数以及证明验证出块节点的合法性；其次，节点还需要验证区块的有效性。一旦验证成功，节点生成该区块Hash的部分签名，并将部分签名放入签名集合中。随后将区块和签名集合传输给其他节点；
* 区块确认：节点收集到足够的部分签名之后，运行一个完整签名恢复方案将一定数量的区块Hash签名聚合成一个完整签名作为区块确认的标志，并广播完整签名给网络中的其他节点。区块确认的条件是：1）收集到足够的部分签名并聚合成完整签名；或2）接收到区块Hash的完整签名并使用聚合主公钥验证有效。当节点生成或者接收到当前区块Hash的完整签名之后，节点将新区块添加到本地主链上，并生成新一轮的随机数，开始新的一轮共识。

**4.2 稳定委员会无线区块链协议细节**

**4.2.1选举委员会成员**

节点加入系统之后，将维护一个共同的节点列表。该列表是将节点的公钥与节点的押金和剩余活动时间对应并排序。因此，我们将根据节点的剩余活动时间选举每个任期的委员会。为了确保委员会的稳定性，剩余活动时间小于某一阈值的节点将不会被选作委员会成员。该阈值取决于委员会任期的大小，即如果剩余活动时间小于任期的大小节点将不会被选为委员会成员。

当系统中所有节点的剩余活动时间都超过任期阈值，委员会成员可以从系统中的节点随机选举得到。假设系统中节点数量为时，恶意节点的数量。记是指定的委员会平均大小。令 是网络中节点的集合且有。我们也可以记集合，其中是诚实节点的集合，是故障节点集合；满足。记集合是委员会成员集合，且有。假设是从中随机均匀选举出的 个节点组成的集合。那么，委员会成员可能的组合数量为，选择的成员中有个诚实节点和个故障节点的概率为。因此，委员会中有超过的节点是故障节点的概率为

为了确保共识协议的活性，我们需要足够数量的诚实节点确保能够达到门限BLS签名方案的安全阈值。

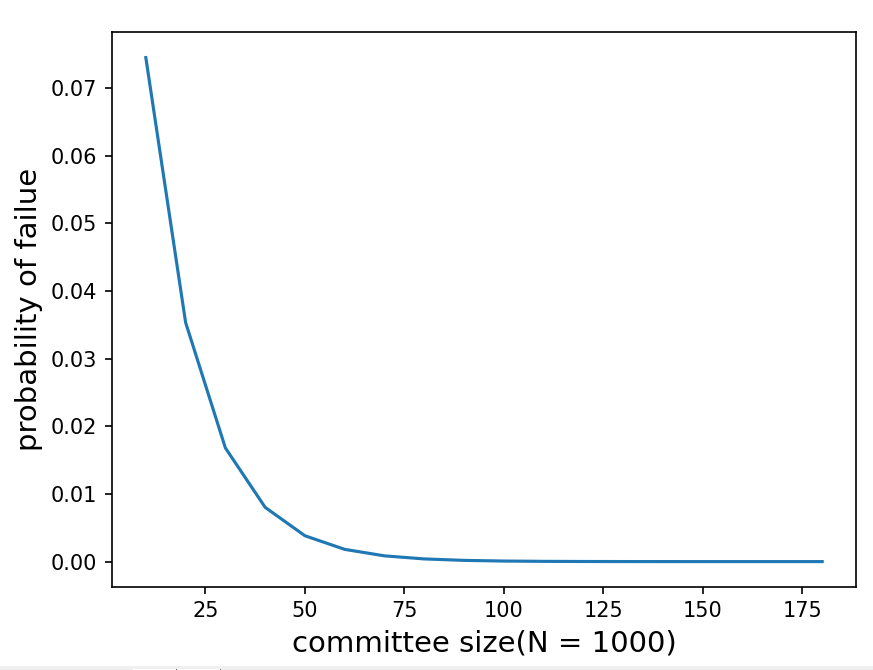
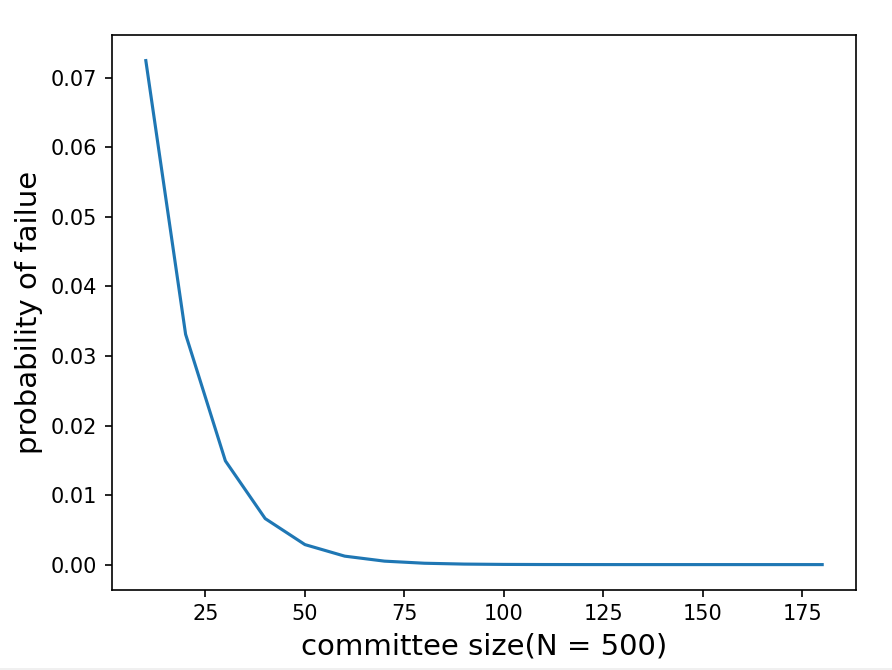
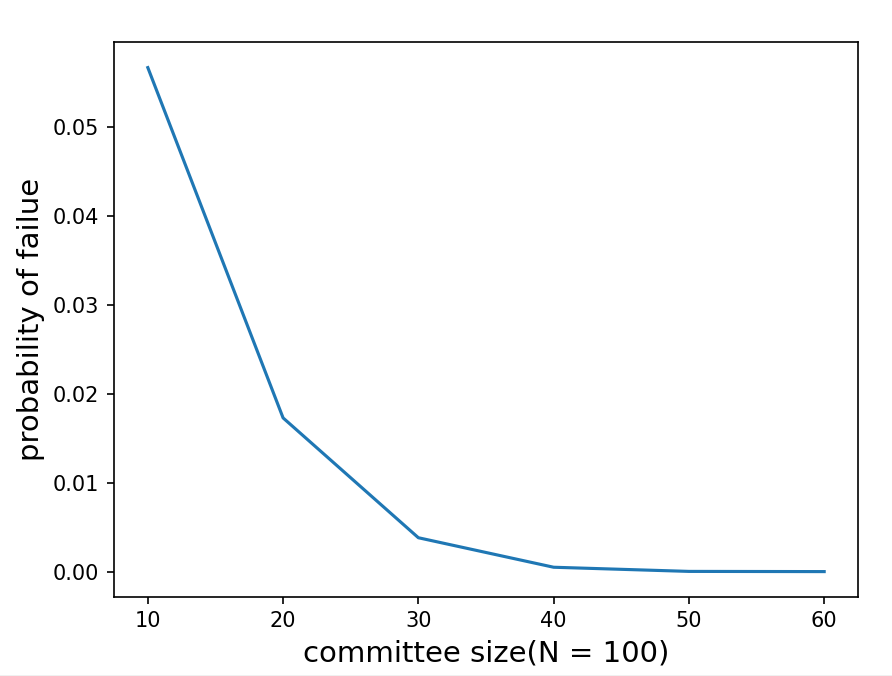


Fig. 不同节点数量时委员会大小Vs. 委员会中故障概率

当给定节点数量后，随着委员会大小的增加，委员会中故障节点数量过半的概率会下降，直到最后趋近于零。如果委员会中在当前轮无法确认区块，则需要重新选举委员会成员。在我们的协议中，如果在一轮结束后没有生成新一轮的随机数，即完整签名，则需要更换整个委员会成员。在正常情况下，这样的情况几乎不会发生。因为委员会中故障节点数量超过 的概率比较低。一旦大多数节点的都是诚实的，并且运行足够长的时间，必然能够生成完整签名完成区块的确认。在委员会任期结束之前将不会该边委员会。

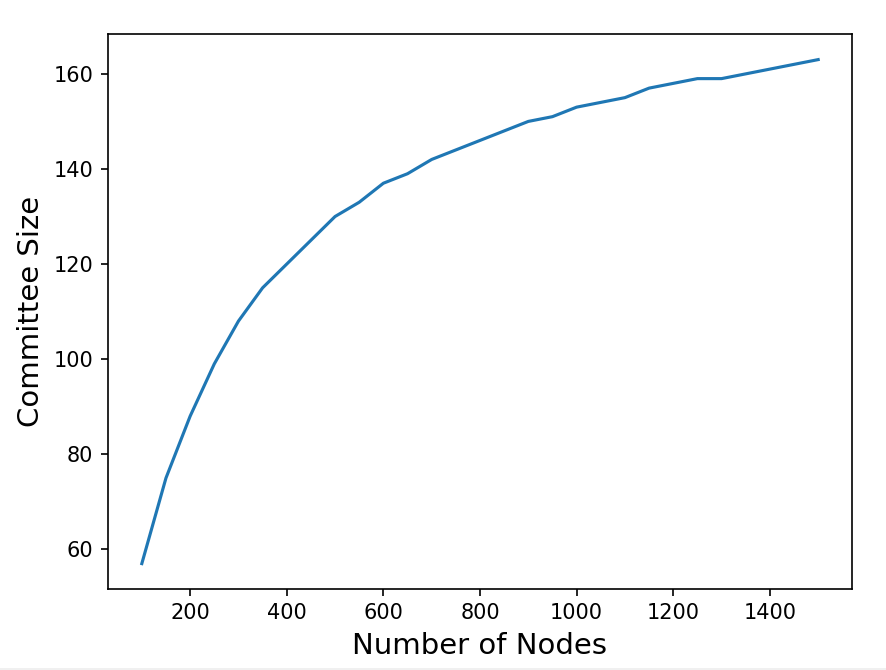


Fig 节点数量Vs.委员会大小

我们的协议具有比较好的扩展性。通过在适当的范围内观察委员会大小与节点数量可以证明。根据图Fig 2可知，委员会大小与节点总数呈次线性增长。当 非常大时，委员会大小与相比可以忽略不记。因此，委员会大小的增长速度比网络总数增长缓慢时，也可以减少共识过程中对协议扩展性和吞吐量的影响。

在选举委员会成功之后，委员会成员需要通过分布式密钥生成算法生成BLS密钥对、获取其他节点的公钥和和聚合主公钥。我们的共识协议采用门限BLS签名方案的三个相关函数：签名生成函数用于生成部分签名；聚合签名恢复函数通过一定数量的部分签名重构完整的签名；签名验证函数通过签名者的公钥验证每个部分签名的有效性和聚合主公钥验证完整签名的有效性。如果当前任期结束，则需要重新选举一个新的委员会。如果委员会任期终止之前有正在确认的区块，则在新委员会选举之后，重新对区块执行共识过程。如果委员会任期终止时，刚好有节点生成区块Hash的完整签名，则认为当前轮完整区块的确认。新的委员会将不需要重新对这个区块执行共识进程。

**4.2.1 出块节点选举**

每个节点在进入系统之后通过分布式密钥生成协议获取自己的密钥对  和聚合主公钥 用于签名的生成和验证。在开始执行共识协议之前，节点需要和其他共识节点交换公钥，最终获得一个公钥列表。节点根据公钥的Hash值对所有的共识节点排序，确保参与共识的节点拥有相同的节点公钥列表。

在稳定区块链共识协议中，主要根据节点的稳定度来决定每个节点被选举为出块节点的概率，我们首先给出节点稳定度的定义。记节点 在区块链系统中的剩余活动时间为 ，则所有共识节点的剩余活动时间之和为 。定义节点 的剩余活动时间比值为

假设在最近的 个确认区块中，由节点 生成的区块数量为 。定义节点 的共识比值为

定义节点 的稳定度为

其中， 为剩余活动时间比权重系数， 为共识比权重系数，可根据偏好设置。在系统运行初期，当确认区块数量不足 个时，记节点的共识比为 。此时，节点的稳定度主要受节点的剩余活动时间的影响。根据定义的节点稳定度可以得到节点 被选为出块节点的概率为

根据基于公钥Hash值排序的节点列表和所有节点的被选中的概率可以构建一个轮盘。记 是节点 （）的稳定度，所有节点的稳定度之和为 ，那么节点 被选中的概率为 且有 。为了确定被选中的节点，将区间 分为连续的多个区间

在每一轮共识中根据轮随机数 和构建的轮盘来选举当前轮的出块节点。因此稳定共识协议需要一个分布式随机数生成协议来确保每一轮出块节点的选举是唯一的、随机的、可验证的。

在我们的协议中，当节点接收到当前轮区块Hash的完整签名 之后，将执行分布式随机数生成协议来产生下一轮的随机数 。为了确保分布式随机数生成协议输入的统一和即时性，我们使用当前轮的轮数 、区块Hash 与完整签名 结合作为随机输入。最后得到下一轮的随机数:

分布式随机数生成协议的输出是不可预测且唯一的。虽然区块Hash 和当前的轮数 是预先知道的信息，但是完整签名 在聚合产生之前是不可以事先预估得到。即使存在节点首先恢复出完整签名，但是不可能篡改完整签名。因此，新一轮随机数 的生成是基于相同的、可验证的输入计算的，确保最终输出是随机且唯一的。分布式随机数生成协议是非交互式的，参与共识的节点不需要通信便能够进入新的一轮共识。一旦节点生成或接收到完整签名 ，就会立刻确认区块并将区块添加到本地区块链上。同时，节点可以自主生成新一轮的随机数开始新一轮共识进程。

可验证出块节点选举协议确保出块节点的选举过程是随机的、可验证的且安全的。所有的共识节点通过输入相同的当前轮的随机数和自己的私钥，可以确定自己是否当选为出块节点。如果随机值  是在拥有私钥的节点所在的区间则该节点当选为出块节点，计算如下：

我们采用的可验证出块节点选举协议是建立在安全可靠的BLS签名方案之上的，即使 个共识节点中存在 个恶意节点也能够正常运行。当绝大多数的诚实共识节点都能正确运行时时，门限签名方案的阈值总是能达到，即在每一轮中最终总是能够恢复完整签名。

当节点被选中为当前轮的出块节点之后，将打包交易生成区块。区块分为区块头和区块体。区块体主要是存储的交易元数据，区块头可以用一个元组表示 ，其中 是当前的轮数， 是前一个区块的Hash， 是存储在区块体中交易的信息， 是出块节点的id， 是区块的Hash。随后，节点将收集足够数量的区块Hash部分签名聚合成完整签名 ，确认区块后将完整签名添加到区块中并完成区块上链操作。

**4.2.2 区块的验证与确认**

我们通过无线网络广播的方式将出块节点生成的区块和区块Hash的部分签名传输给其他共识节点。当节点接收到新的区块消息时会检查区块的有效性和验证出块节点对区块Hash签名。共识节点通过验证以下组件的有效性来验证区块的有效性：

* 出块节点id ：节点通过出块节点的公钥和当前轮的随机数验证出块节点的合法性；
* 区块父Hash ：新区块的父Hash必须与节点维护的当前最新区块链的最后一个区块的Hash相同；
* 交易信息 ：验证交易列表的基本信息是否有效；
* 交易：检查区块中的所有交易是否都是有效的，如果存在无效交易，则认为区块是无效的；

如果以上条件都满足后，节点验证区块的有效性都成功，生成区块Hash的部分签名并广播部分签名给其他共识节点。当共识节点收集到 个有效的部分签名时，可以聚合恢复成一个区块Hash的完整签名，作为区块被确认的证明。当一个完整签名通过聚合主公钥快速验证有效，说明已经有足够多的节点认为这个区块是有效的。因此，完整签名作为区块确认的标志是可行的。由于诚实的节点在一轮中最多只会为一个有效区块的区块Hash签名，最终只有一个区块完成验证和确认过程。这意味着共识协议确保系统在同一轮中不会出现多个区块同时被确认，防止了链分叉的出现。

如前所述，完整签名的恢复可以由任意节点独立完成，这极大地提升了共识过程的稳定性。当节点聚合多个部分签名时，就可以生成完整签名。即使恶意节点拒绝生成完整签名并广播给其他节点，诚实的节点也会宣布区块被确认的结果。这样的设计使得我们的协议在部分节点发生故障或者无线网络通信不可靠时也能稳定的实现共识完成区块的确认。

**4.3 奖惩机制**

我们的共识协议通过一个激励机制可以提高节点愿意签名区块Hash的积极性。无论是验证区块的有效性还是生成签名都需要消耗节点的算力。当没有成为出块节点时，理性的节点可能不愿意浪费算力验证区块生成签名，而是更愿意等待生成完整签名结束当前轮之后开始新的一轮共识。因此，为了确保协议能够高效运行，需要一个奖励机制促使节点愿意验证区块生成签名。系统中所产生的主要费用是打包交易所产生的部分交易费用。每一轮中区块的交易费用除了分配给出块节点以外，剩下的一部分交易费用将分配给为生成完整签名提供部分签名的节点。

针对可能存在节点接收到多个最终签名是由不同部分签名组成的问题，签名奖金将发放给聚合完整签名的部分签名集合中平均时间戳最小的节点集合。记聚合完整签名的部分签名集合为 ，每个签名对应的时间戳分别为 。假设系统存在两个聚合完整签名的部分签名集合， 每个签名的时间戳分别对应为  和 。两个签名集合中平均时间戳分别为 。如果 ，则签名奖励将会被均分给 的所有节点。这种奖励机制不仅会提高节点签名的积极性，还能提高区块签名的效率，从而提高系统的性能。这个机制不仅可以激励节点参与签名，更可以激励节点产生完整签名之后尽快将结果广播给其他节点，从而提高共识的效率。

此外，协议中设置了惩罚机制降低理性节点作恶的机会。一旦发现合法出块节点生成无效区块或者超时不生成区块导致在当前轮最终确认空区块来判定节点的恶意行为。针对这类出块节点，我们将会选择减少节点在系统中的活动时间作为惩罚。这个惩罚措施会降低节点的稳定度，进而会降低节点被选为出块节点的概率，最终降低节点当选为出块节点获得奖励的概率。因此，理性的出块节点为了确保自己的收益，会减少生成无效区块的情况，从而提高系统的安全性和效率。

**4.3 同步机制**

在我们的协议中，我们考虑了一种更加高效合理的区块链节点自启机制。当新节点加入系统之后，在开始参与共识之前需要获取区块链的历史。为了尽量确保节点能够获取真实的区块链信息，节点会向稳定度更高的 个邻居节点请求部分区块链信息。稳定度更高的节点更愿意传输正确的、最新的区块链历史。当需要历史区块链的区块数量为时，可以向每个邻居节点请求个区块。假设节点接收一个区块所需要的时间为,并且在一轮通信中接收成功的概率为时，则节点有很高概率能够在后接收区块链历史成功。这个设计可以减少单个节点长期传输区块链历史的负担，同时也可以降低节点发送错误区块链历史的概率。一旦有节点传输错误的区块链历史，接收节点也可以根据其他节点传输的部分区块链历史检查到。

1. **协议分析**

在本小节，我们详细分析稳定无线区块链协议的安全性、稳定性和共识过程的高效性。我们通过证明协议满足持续性（persistence）和活性来证明协议的安全性。我们详细分析了协议共识过程的开销来证明协议能够节能且高效的生成有效区块实现共识。

**5.1 安全性分析**

在这一小节，我们证明稳定无线区块链协议能够确保无线区块链系统在存在部分节点故障的情况下安全的运行。当绝大多数节点是诚实时，我们的协议就可以保证满足持续性和活性：

* 持续性：如果诚实节点宣布区块是有效的, 则其他诚实诚实节点也认为 是有效的；
* 活性：所有诚实的节点将最终确认一个区块并且生成新一轮的随机数。

根据故障的类型可以分为节点崩溃故障和拜占庭故障。节点发生崩溃故障时将出现消息丢失或者节点不响应，进而影响共识过程。而节点发生拜占庭故障时，将会发送一些错误消息，误导其他节点来影响共识的实现。节点发生故障可能的原因多种多样，敌手攻击是比较常见的导致节点故障的原因。

我们共识协议的安全性并不依赖出块节点的正确性，而是由安全的门限签名方案来确保的。出块节点发生故障时并不会影响系统的安全性。出块节点在稳定无线区块链协议中只承担生成区块的功能。当出块节点生成一个无效区块或者不生成区块时，我们的共识协议将会通过门限签名机制强制更换出块节点开确保协议的运行。因此，协议中一轮共识进程结束之后，会出现两种输出：有效的被确认的区块和空区块（与一般的区块数据结构相同，但是区块中没有任何交易）。一个有效区块被确认需要满足以下条件：1）出块节点创建一个有效区块；2）该有效区块被足够多的共识节点接受并最终为其Hash聚合生成完整签名。稳定共识协议需要至少 个诚实节点来满足门限签名的需求。门限签名机制的输出是不可伪造确保了这个机制的安全性。门限签名机制具有两个性质：

* 唯一性：门限签名方案中任意超过 个部分签名都能恢复相同的完整签名；
* 可验证性：任意节点都可以通过聚合主公钥验证最终恢复的完整签名的有效性。聚合主公钥是在节点加入系统的时候由密钥生成协议生成和分配的。

我们将在之后详细讨论稳定无线区块链共识协议能够满足持续性和活性，确保无线区块链系统能够稳定和安全地运行。

**5.1.1 持续性分析**

只要诚实节点的数量能够满足门限签名机制的安全阈值，稳定无线区块链协议就能满足持续性：

**定理一（持续性Persistence）**：记 是共识节点的数量。当诚实节点的数量超过且每一轮中每个节点只能对一个区块进行一次投票时，所有诚实的节点将维护相同的区块链。换而言之，对任意的诚实节点和，如果节点 声明区块 在区块链第 个位置上, 则诚实节点区块链相同位置上的区块一定是 。

**证明**：要证明持续性，我们需要证明任意两个诚实的节点维护的区块链相同。采用反证法证明，假设 分别是诚实节点 维护的区块链上相同位置的区块，且 。那么会有两种情况出现：

* + C1： 在同一轮中分别被添加到节点的本地区块链 上的两个不同的区块。由于区块的确认是聚合任何至少个部分签名恢复唯一的完整签名。如果在同一轮两个不同的区块都最终被确认，则表明在同一轮中门限签名机制最终恢复了两个不同的完整签名，这与门限签名机制的唯一性冲突了，因此假设不成立。
  + C2:是在不同的轮被添加到节点的区块链上的区块。假设分别是在 轮中被添加到节点的区块链 上的第个区块。根据我们的协议，在 轮中添加 到区块链上时至少有 个节点认为区块是有效的，这些节点认可区块 在 轮首次被添加到的区块链的第 个区块。由于在一轮中不会最终确认两个不同的区块，因此添加区块 的节点在 轮中有相同的区块链视图，即这些节点维护的区块链中的第 个区块是相同的。由于 ，且 。当节点 在 轮之前故障了并且在 轮的时候恢复，那么将会有至少 个节点在第 轮的时候认可区块链的第 个区块是 。由于之前在 轮中也至少有 个节点认可区块链在第 个区块中是 。如果两个区块都认为是有效的，则网络大小为 ，矛盾了。

最终得出结论，所有的诚实节点对于区块链历史应该有相同的视图，即所有诚实的节点将维护相同的区块链。因此，我们的协议是满足持续性的。

**5.1.2 活性分析**

只要诚实节点的数量能够满足门限签名机制的安全阈值，稳定无线区块链协议就能满足活性：

**定理二（活性）**：记是参与共识的节点总数，且诚实节点的数量满足门限签名的安全阈值。即使系统存在个故障节点，协议最终也一定会终止并获得新一轮的随机种子。

**证明：**运行稳定无线区块链共识协议时，故障节点可以相互勾结发起两种攻击：不生成有效区块和拒绝为有效区块的Hash签名。

* + 出块节点故障：如果恶意的出块节点不生成任何区块，则诚实的节点在超时未接收到区块后会最终确认一个空区块。如果恶意的出块节点生成一个无效区块，该区块将无法通过区块有效性检查。诚实节点将会拒绝为无效区块的区块Hash签名，最终导致无法聚合生成证明该区块有效的完整签名。因此，最终诚实节点将在当前轮聚合恢复一个空区块的完整签名，得到新一轮的随机种子。
  + 如果恶意的节点拒绝为有效区块的Hash签名，则剩下的个诚实节点会正确地生成有效区块Hash地部分签名并广播给其他节点。任意诚实节点在收集到足够的部分签名都会聚合恢复有效区块Hash的完整签名，并广播给网络中的其他节点。因此，最终会完成当前轮的共识过程，获得新一轮的随机源—完整签名。

我们的协议确保当系统中存在故障节点时也能够完成一轮共识过程，并获得下一轮的随机种子。因此，我们的共识协议是能够确保活性的。

**5.1.3 女巫攻击分析**

我们的协议能够比较好的防止理性的节点发起女巫攻击。假设攻击者被选中成为出块节点的概率为 ，其余 个节点的概率分别为 。若攻击者发起女巫攻击，则攻击者控制的节点被选中的概率的期望 。此时有 。

**证明**：令 ，则有 $。当 时有 ，因此 。最终得出结论 。因此，我们的协议中，攻击者发起女巫攻击时，成为出块节点的概率会降低。攻击者的伪造节点越多，则成为出块节点概率期望将会越低。通常理性的节点不会愿意伪造多个节点降低成为出块节点的概率。

即使敌手发起女巫攻击，我们的协议也能够安全的运行。当敌手发起女巫攻击时，会控制节点不发送消息或者发送一些错误的消息阻碍协议达成共识。如果女巫节点成为出块节点，生成一个无效区块或者不生成区块，其他诚实节点在超时未接受到有效区块就会对一个空区块达成共识。当女巫节点没有成为出块节点时，将不会对当前轮的有效区块Hash签名或者发送错误的签名。由于区块验证和确认的过程是节点独立完成，因此只要有足够多的诚实节点对区块Hash签名就能确保共识过程的最终性。诚实节点接收到错误的签名之后会验证得知该签名无效，最终丢弃无效签名。只要有诚实节点收集到足够多的有效部分签名就能聚合生成完整签名，完成区块最终确认达成共识的过程。因此，只要敌手不控制超过安全阈值的投票权，我们的协议可以抵抗女巫攻击。

**5.1.4 阻塞攻击分析**

我们的协议能够较好的抵抗敌手发起阻塞攻击，确保系统安全的运行。敌手可以控制节点发起阻塞攻击（Jamming Attack）阻止其他节点传输消息，最终导致诚实节点无法收集到足够的部分签名完成对有效区块或者空区块的确认。攻击节点通过发送大量的消息，阻止节点接收其他消息。如果敌手攻击足够大且同时阻塞大量节点时，可以使得无法在超时范围之内收集到足够的部分签名，最终导致无法完成区块的确认协议失去活性。在我们的协议中，每个共识节点将会维护一个对敌手攻击能力的时间窗口估计。根据对敌手攻击时间的估计，节点动态调整节点收集部分签名的超时时间和发送消息的概率。最终，稳定共识协议在敌手发起阻塞攻击时也能最终对区块达成共识。

**定理三：**当敌手攻击能力为时，稳定无线区块链协议有很高概率在 轮后恢复完整签名确认区块。其中，是节点之间传输消息成功的概率。

**证明**：假设敌手在 轮通信中最多能够发起  轮阻塞攻击，每个节点成功传输消息的概率为 。当敌手不发起攻击时，稳定无线区块链协议有很高概率在  轮后恢复完整签名确认区块。根据节点传输成功的概率计算 ，可知共识节点在 轮后有很高的概率可以成传输部分签名给另一个节点。因此，在  轮后，共识节点有很高的概率可以收集到足够的部分签名且聚合恢复完整签名，最终确认区块。当敌手发起阻塞攻击时，每轮中只有轮可以让诚实节点传输消息。因此，诚实节点有很高概率能够在 轮后成功传输消息到另一个节点的，节点成功收集到足够的部分签名恢复完整签名的轮数为。(证毕)

稳定无线区块链共识协议通过动态调整超时时间能够很好的抵抗阻塞攻击，确保系统运行的安全性。敌手发起阻塞攻击会使得诚实节点的部分签名不能成功传输，最终导致系统无法在超时阈值之内接收到足够的部分签名而无法完成共识。因此，超时时间阈值固定不变将会导致在一轮结束后无法得到新一轮的随机种子，使得协议失去活性。在稳定共识协议运行时，每个节点会根据感知的信道情况来估计敌手攻击的时间窗口上界。之后，根据敌手攻击时间窗口估计来动态调整超时机制的阈值，降低由于敌手阻塞攻击导致最终无法恢复完整签名达成共识的概率。因此，这个机制可以保证稳定共识协议在敌手发起阻塞攻击时的活性，进一步确保无线区块链系统的安全性。

**5.1.5长程攻击**

敌手发起长程攻击时可以伪造区块链历史使得自己获益来。根据区块链的最长链原则，敌手可以伪造一条更长的区块链，使之成为被其他节点认可的主链来使自己获益。这种攻击通常发生在生成区块代价较小的区块链系统中。

稳定无线区块链协议融合了安全的门限BLS签名方案，具有较好的长程攻击抗性。当敌手想要篡改区块链上的区块时，需要获得至少 个节点的私钥生成区块确认的完整签名。节点获取其他节点的私钥只能通过贿赂节点获取。要贿赂 个节点需要掌控整个共识协议超过 的投票权。敌手要获取足够的私钥需要花费大量的金钱，篡改区块链数据付出的代价将非常的大。当节点想要篡改多个区块链历史数据时，需要的密钥数量将增加，敌手的代价将增加。理性的敌手将不愿意花费代价篡改数据，不理性的敌手也很难付出足够的代价篡改数据。因此，稳定共识协议能够很好的抵抗敌手发起长程攻击，确保数据的安全性。

**5.2 性能分析**

在本小节，我们将详细分析协议共识过程的算力开销和通信开销，证明稳定无线区块链共识协议是比较节能和高效的，适用于无线网络环境。

在稳定无线区块链协议的共识过程中，有效区块获得至少 个区块Hash的部分签名时，可以恢复完整签名确认有效区块。记节点成功接收到另一个节点的消息的概率为,则节点成功接收到至少 个区块Hash的部分签名的概率为 ，只要有一个节点恢复了有效区块Hash的完整签名，就可以确认有效区块。因此，共识协议在一轮中生成有效区块的概率为

其中 为成功生成接收到完整签名的节点的数量。当 时说明只要有一个节点恢复完整签名，就可以确认有效区块并将其添加到本地区块链上。因此，生成空块的概率为。

**5.2.1 算力开销分析**

算力开销主要是共识过程系统进行计算时的消耗。在我们的共识协议中，算力开销来自共识过程中出块节点选举过程的计算，生成和验证区块时的交易Hash计算、区块Hash计算，确认区块时生成和验证部分签名的计算，以及生成和验证完整签名时的计算。

记每个节点的算力为 ，令出块节点选举时间为，区块中交易的打包时间为 ，区块哈希和签名的计算时间都为 。

* 生成一个空块的哈希次数为 ；
* 生成一个有效区块的哈希次数为 。

假设在生成一个有效区块之前已经生成过 次空块，当生成空块的概率 时，生成一个有效区块所需的哈希次数为

。最终计算得到生成一个有效区块的平均哈希次数为

**5.2.2 通信开销分析**

在我们的共识协议中生成一个有效区块的时间包括生成和验证区块（毫秒级）、广播区块（秒级）；生成和验证部分签名（毫秒级）、广播部分签名（秒级）、生成和验证完整签名（毫秒级）、广播完整签名（秒级）。

记节点广播一次的时延为 ，打包有效区块的时间为 ，生成签名的时间为 。

* 生成一个有效区块的时间为 ；
* 生成一个空块的时间为 。

假设生成有效区块的通信次数为 ；生成空块的通信次数为 。假设生成一个有效区块之前连续 次生成空块。

* + 生成有效区块：假设交易到达速率为 ，协议在 时间内生成一个有效区块，则交易到达数量为 。生成一个有效区块的通信次数为 ；
  + 生成空块：出块节点在 时间内未生成一个有效区块。所有节点会在 时间内生成一个空块。这段时间内的交易到达数为 ，增加的通信数是 。生成空块的通信数量为 。

最后计算得到在生成个空块之后生成一个有效区块时平均通信数量为

1. **仿真结果**

在这一节中，我们仿真研究变量参数是如何影响稳定协议的性能。为了评估协议的性能，我们主要采用两个指标：吞吐量和共识时延，具体定义如下：

* 共识时延：区块从产生到最终链接到区块的时间（区块确认时间）；
* 吞吐量：单位时间内处理交易的数量。

吞吐量是单位时间内确认交易的数量，这个指标受到确认交易的数量和共识时延的影响。

我们仿真代码是用Python语言写的，采用的 python 版本是 Python 3.10.0。所有的实验都是在64 位操作系统, 基于 x64 的处理器，Intel(R) Core(TM) i7-10700F 2.90GHz CPU 的处理器，8.00 GB RAM 中进行的。

在正式分析协议性能之前，我们先要确定协议中节点稳定度的权重系数。

**6.1 权重系数**

在稳定区块链协议中，出块节点的稳定度是由节点活动时间比和节点成功参与共识的共识比决定。这两个因素的权重会对选取出块节点的可靠程度产生影响。虽然我们协议的安全性并不依靠出块节点的正确性，但是我们还是尽可能的期望在每一轮都能生成有效出块节点。当无线节点的活动时间比较低时，节点作恶的成本会降低，有更大的概率会生成无效区块或者直接不生成区块。当节点在最近一段时间内参与共识的次数比较多时，节点更愿意维护当前的区块链并且生成有效区块获得奖励。只有生成有效区块才能确保系统处理交易的效率，因此需要尽可能选出一个更加值得信任的节点作为出块节点。为了确定稳定度的权重系数，我们实验分析了不同的权重系数时优质节点被选中的概率。实验表明，当所有节点的活动时间比和共识比服从均匀分布时，两个因素对于优质节点选举的影响都比较大。

实验考虑的场景是在网络大小为 的区块链系统中，有  的节点活动时间比和共识比都比较低，有 的节点活动时间比高但共识比低，有 的节点活动时间比高和共识比低，有 的节点活动时间比和共识比都比较高，如下图所示：

我们将四种节点均匀分布在网络区域中，且每个 都会测试 100 次共识的情况。除了 变化以外，其余所有参数都一样。实验最后确认不同的 下，优质稳定点被选为出块节点的比率。从测试结果图可以看出，当 附近时，优质节点被选中的比率更大。实验结果表明，节点活动时间比和共识比都会影响优质节点被选中的概率，且重要性相差无几。因此，两者的权重系数应该尽可能地接近。在之后的性能测试中，稳定度的权重系数都设置为 ，确保在共识过程中尽量选举出优质稳定的出块节点。

定义节点 的稳定度为 。我们认为活动时间长且历史产生区块数量多的节点更值得信任。质押金投入比较多且已经创建部分区块链历史的节点，有更大概率被选中成为出块节点。为了以后也能有更高的概率继续生成区块，这类节点更愿意遵循协议生成有效区块，从而获得奖励。

**6.2 区块大小**

共识时延是用来衡量共识协议性能的重要度量，主要都到区块大小和参与共识的节点数量的影响。根据使用区块的大小和共识时延，我们可以计算最终确认的交易的数量来测量协议的吞吐量。

我们首先测试了不同区块大小对于共识协议中共识时延的影响。我们分别测试了区块大小从 到 时的共识时延和平均吞吐量，并且用四种不同的节点数量重复了相同的实验。我们测试了10次共识的平均时延和平均吞吐量，结果如下图所示：



根据之前的实验可知，随着区块大小的增加，共识时延随着区块大小的增加呈线性增长，在有700个节点的网络中区块大小为1MB的时候的最终确认时延也在0.3秒左右。由于所有网络节点的带宽限制为35Mbps，传输较大的区块需要更长的时间使得共识时延增大。详细的实验结果表明，区块传输占共识完成时间的90%以上说明区块传输是共识协议的主要瓶颈。此外，随着区块大小的提升，平均吞吐量先是缓慢增长，最终在区块大大于1MB之后平均吞吐量趋于平缓。结果表明，只增加区块的大小并不能无限制的提升平均吞吐量。区块大小为1MB时协议能够在较小的平均共识时延中实现较大的平均吞吐量。最终结果表明，即使区块大小比较大的时候，我们的协议能够保证在较低的时延时拥有较高的吞吐量。

**6.3 节点数量**

其次，我们实验分析共识协议中节点的数量对于共识时延和吞吐量的影响。我们分别测试了节点数量从 100 增加到 1000 时的共识时延和平均吞吐量，并且用四种不同的区块大小重复了相同的实验。我们测试了10次共识的平均时延和平均吞吐量，结果如下图所示：



根据上面的实验结果可知，随着节点数量的增加，共识时延也以比较缓慢的速度增加。这是因为在单跳无线网络中，节点采用广播的形式传输消息。因此，当节点拥有相同的网络带宽限制时，节点传输区块的时间是非常接近的。但是，随着节点数量的增加，用于确认区块的签名数量会成比例增加。因此，所有验证区块的节点需要验证更多的部分签名，并且聚合成完整签名的也需要更高的阈值。最终结果表明，随着节点数量的增加，验证节点需要更多的计算时间验证签名和收集更多的部分签名来聚合形成完整签名。当区块大小固定且共识时延增加时，我们的平均吞吐量会降低。结果显示，即使节点数量增加到1000时，平均吞吐量也是大于4000TPS，因此，我们的协议仍然具有较好的扩展性。

**6.4 带宽**

我们实验分析了带宽对于共识时延和平均吞吐量的影响。我们分别测试了带宽从 增加到 时的共识时延和平均吞吐量，并且用四种不同的区块大小重复了相同的实验。我们测试了10次共识的平均时延和平均吞吐量，结果如下图所示：



从上图可知，共识时延随着带宽的增加而下降最后趋于平缓。随着带宽的增加，节点在网络中传输消息的速度变快使得共识时延会降低。同时，随着共识时延降低，协议的平均吞吐量线性增长。最终结果表明，带宽会影响共识协议的性能，带宽越大协议处理交易的效率就越高。

无线网络中敌手可以通过发起攻击影响共识过程，进而影响共识的效率。测试不同敌手能力下我们协议的性能，进一步分析我们协议的抗攻击性。

**6.5 女巫攻击**

敌手发起女巫攻击，攻击者控制的节点会竞争成为出块节点但并不打包区块或者打包一个无效区块。在我们的协议中，最终会生成并确认一个空块。即使出块节点故障，我们的协议依旧能够继续运行并不依赖出块节点确保协议的安全性。当敌手控制的恶意节点没有成为出块节点时，在共识过程中将不对验证有效的区块进行签名确认。在之前的分析中，我们已经说明了只要敌手控制的投票权不超过安全阈值，最终有效区块一定会被确认，并且添加到区块链上。

我们实验分析了共识协议中，女巫节点百分比对于共识时延和吞吐量的影响。我们分别测试了女巫节点占比从 5% 增加到 49% 时的共识时延和平均吞吐量，并且用四种不同的节点数量重复了相同的实验。当女巫节点数量为总数量的 50% 时，会导致没有足够数量的部分签名而无法确认任何区块，最终导致协议瘫痪。因此，我们测试最大的女巫节点占比为 49%。我们测试了10次共识的平均时延和平均吞吐量，结果如下图所示：



上图实验结果显示，共识时延不会随着女巫节点的百分比的增加而明显增加。当女巫节点当选为出块节点，在当前轮会最终确认一个空块。由于空块比较小传输比较快，因此生成空块轮的共识时延会比较小。结果表明，生成空块会使得平均共识时延降低。但是，吞吐量不会随着女巫节点数量的增加呈现较大波动。这是因为随着攻击者投入越多，女巫节点当选为出块节点的概率会增加，生成空块的次数会增加。由于空块中并不包含任何交易且共识时延会比较低，因此平均交易吞吐量不会发生较大的变化。最终结果表明，我们的协议面临女巫攻击具有较好的活性和性能。

**6.6 阻塞攻击**

敌手可以通过发起阻塞攻击影响无线网络中消息的传输，进而影响区块的传输和确认过程，导致最终无法实现区块确认。通过之前的分析可知，我们的协议具有较好的阻塞攻击抗性。每个节点会维护一个敌手攻击窗口的估计，从而保证最终有很高的概率能够实现区块的传输、验证和最终确认的过程。

我们实验分析了共识协议中，攻击能力为 的攻击者发起阻塞攻击时对共识时延和平均吞吐量的影响。假设攻击者发起攻击时将在任意 轮中阻塞连续 轮。我们分别测试了在敌手具有不同攻击能力 到 时的共识时延和平均吞吐量，并且用四种不同的节点数量重复了相同的实验。我们测试了10次共识的平均时延和平均吞吐量，结果如下图所示：



从上图可知，随着 的增加共识时延降低，这是因为随着 的增加，敌手发起攻击的频率会降低。当 的增加时，节点收集部分签名聚合成完整签名的所需要的时间会降低，从而使得共识时延降低了。此外，平均吞吐量会随着 的增加而增加。由于区块大小固定且共识时延降低，会使得平均吞吐量增加。针对敌手发起的连续阻塞攻击，我们协议中每个节点都会维护一个敌手攻击时间窗口的估计，因此可以动态调整确认超时阈值。这个机制确保了即使敌手发起阻塞攻击，我们的协议也会最终确认区块。最终结果表明，节点维护的敌手窗口估计可以很好的抵抗连续阻塞攻击。

1. **结论与展望**

在本文中，我们提出一个适用于单跳无线网络的稳定五西安区块链共识协议。这个协议的安全性是建立在安全的门限签名机制和可信高效的出块节点选举机制上。此外，我们的协议具有强一致性，可以避免区块链出现分叉。同时，稳定五西安区块链协议对敌手发起的女巫攻击和阻塞攻击都有比较好的抵抗能力。协议在节点存在故障时，也能够稳定、安全且高效的运行。实验分析和仿真结果显示了稳定共识协议的安全性和高效性。在未来，我们将研究多跳无线网络下的稳定共识协议，并且改进实现更加安全高效的无线区块链协议，使得协议能够更好的抵抗各种区块链攻击。

**参考文献:**

[1] A. Salam. "Internet of things for sustainable community development: introduction and overview[M]." Internet of Things for Sustainable Community Development. Springer, Cham, 2020: 1-31.

[2] O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1184-1195, April 2018, doi: 10.1109/JIOT.2018.2812239.

[3] A. K. Gupta and R. Johari, "IOT based Electrical Device Surveillance and Control System," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1-5, doi: 10.1109/IoT-SIU.2019.8777342.

[4] N. Dlodlo, O. Gcaba and A. Smith, "Internet of things technologies in smart cities," 2016 IST-Africa Week Conference, 2016, pp. 1-7, doi: 10.1109/ISTAFRICA.2016.7530575.

[5] H. -N. Dai, Z. Zheng and Y. Zhang, "Blockchain for Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8076-8094, Oct. 2019, doi: 10.1109/JIOT.2019.2920987.

[6] O. Onireti, L. Zhang and M. A. Imran, "On the Viable Area of Wireless Practical Byzantine Fault Tolerance (PBFT) Blockchain Networks," 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9013778.

[7] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf, 2008.

[8] A. Kiayias, A. Russell, B. David, and R. Oliynykov, “Ouroboros: A provably secure proof-of-stake blockchain protocol,” in *Annual* *International Cryptology Conference*. Springer, 2017, pp. 357–388.

[9] M. Castro, B. Liskov. Practical Byzantine fault tolerance[C]. Proceedings of the 3rd Symposium on Operating Systems Design and Implementation(OSDI), 1999: 173-186.

[10] Z. Jiang, Z. Cao, B. Krishnamachari, S. Zhou and Z. Niu, "SENATE: A Permissionless Byzantine Consensus Protocol in Wireless Networks for Real-Time Internet-of-Things Applications," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6576-6588, July 2020.

[11] Q. Xu, Y. Zou, D. Yu, M. Xu, S. Shen, F. Li. Consensus in Wireless Blockchain System, in WASA, 2020.

[12] Y. Zou, M. Xu, J. Yu, F. Zhao and X. Cheng, "A Fast Consensus for Permissioned Wireless Blockchains," in IEEE Internet of Things Journal, 2021.

[13] M. Xu, F. Zhao, Y. Zou, C. Liu, X. Cheng, F. Dressler. BLOWN:A Blockchain Protocol for Single-Hop Wireless Networks under Adversarial SINR, in CoRR abs/2103.08361, 2021.

[14] M. Vukolic, ‘‘The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication,’’ in Proc. Int. Workshop Open Problems Netw. Secur., 2015, pp. 112–125.

[15] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, “Proofs of space,” in Annual Cryptology Conference. Springer, 2015, pp. 585–605.

[16] K. Karantias, A. Kiayias, and D. Zindros. "Proof-of-Burn," in Financial Cryptography and Data Security, 2020, pp. 523-540, doi: 10.1007/978-3-030-51280-4\_28.

[17] V. Buterin and V. Griffith, ‘‘Casper the friendly finality gadget,’’ 2017, arXiv:1710.09437. [Online]. Available: https://arxiv.org/abs/1710.09437

[18] F. Gai, B. Wang, W. Deng and W. Peng. "Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-to-Peer Network." In Database Systems for Advanced Applications, 2018, pp. 666-681, doi: 10.1007/978-3-319-91458-9\_41.

[19] Hyperledger Fabric. [Online]. Available: https://cn.hyperledger. org/projects/fabric. 2019.

[20] Q. Wang, R. Li, S. Chen, and Y. Xiang. "Formal Security Analysis on dBFT Protocol of NEO."  [arXiv:2105.07459](https://arxiv.org/abs/2105.07459) [cs.CR], 2021

[21] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative byzantine fault tolerance," ACM Trans. Comput. Syst., vol. 27, no. 4, pp. 1–39, 2010.

[22] E. Buchman. "Tendermint: Byzantine fault tolerance in the age of blockchains." Ph.D. thesis, The University of Guelph, Guelph, Ontario, Canada, June 2016.

[23] M. Xu, C. Liu, Y. Zou, F. Zhao, J. Yu and X. Cheng, "wChain: A Fast Fault-Tolerant Blockchain Protocol for Multihop Wireless Networks," in IEEE Transactions on Wireless Communications, vol. 20, no. 10, pp. 6915-6926, Oct. 2021, doi: 10.1109/TWC.2021.3078639.

[15] Burstcoin official website. https://www.burst-coin.org/. May. 2019.

[16] B. Wiki. Proof of burn. [Online]. Available: https://en.bitcoin.it/wiki/Proof\_of\_burn

[17] V. Buterin and V. Griffith, ‘‘Casper the friendly finality gadget,’’ 2017, arXiv:1710.09437. [Online]. Available: https://arxiv.org/abs/1710.09437

[18] Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-to-Peer Network. https://link.springer.com/content/pdf/10.1007%2F978-3-319-91458-9\_41.pdf. Jan. 2019.

[19] Fabric official website. https://get.fabric.io/. Jan. 2019.

[20] Open Network for the Smart Economy. Accessed: Mar. 20, 2018. [Online]. Available: <https://neo.org/>

[21] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, ‘‘Zyzzyva: Speculative byzantine fault tolerance,’’ ACM Trans. Comput. Syst., vol. 27, no. 4, pp. 1–39, 2010.

[22] J. Kwon. Tendermint: Consensus without mining.

https://tendermint.com/static/docs/tendermint.pdf (21 August 2021, date last accessed).

[23] Y.Gilad, R. Hemo, S. Micali, et al. Algorand: Scaling Byzantine agreements for cryptocurrencies[C]. In: Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28–31, 2017: 51–68.

[24] M. Zheng, M. Goldenbaum, S. Stańczak and H. Yu, "Fast average consensus in clustered wireless sensor networks by superposition gossiping," 2012 IEEE Wireless Communications and Networking Conference (WCNC), 2012, pp. 1982-1987, doi: 10.1109/WCNC.2012.6214113.

[25] M. Goldenbaum, H. Boche and S. Stańczak, "Nomographic gossiping for f-consensus", Proc. 10th Int. Symp. Model. Optimiz. Mobile Ad Hoc Wireless Netw., pp. 130-137, 2012.

[26] C. Newport and P. Robinson, “Fault-tolerant consensus with an abstract mac layer,” arXiv preprint arXiv:1810.02848, 2018.

[27] F. Molinari, N. Agrawal, S. Stańczak and J. Raisch, "Max-Consensus Over Fading Wireless Channels," in IEEE Transactions on Control of Network Systems, vol. 8, no. 2, pp. 791-802, June 2021.

[28] H. Moniz, N. F. Neves and M. Correia, "Byzantine Fault-Tolerant Consensus in Wireless Ad Hoc Networks," in IEEE Transactions on Mobile Computing, vol. 12, no. 12, pp. 2441-2454, Dec. 2013, doi: 10.1109/TMC.2012.225.

[29] A. Boldyreva. "Threshold signatures, multi signatures and blind signatures

based on the gap-Diffie-Hellman-group signature scheme," in Proc. 6th Int. Workshop Theory Pract. Public Key Cryptogr., 2003, pp. 31–46.

[30] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing[C]". International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2001:514-532.

[31] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. "Secure distributed key generation for discrete-log based cryptosystems," in Proc.Int. Conf. Theory Appl. Cryptograph. Techn., vol. 1592, Aug. 2010,pp. 295–310.