**A Stable Consensus Protocol in Wireless Blockchain System**

**Abstract**

# Introduction

随着无线通信技术和区块链技术的高速发展，许多研究者都探索将区块链技术用于无线网络的应用，如移动边缘计算[1]、智能5G[2]、车联网[3]等领域。区块链技术可以在分布式环境下提供可信赖和安全的资源共享服务，受到了学术界和工业界的高度关注。区块链的去中心化、持久性和可追溯等特性，使得将其与智能合约技术相结合时，能为克服无线网络应用中存在的安全和信任问题提供了新的解决方案。将区块链技术融入无线网络可以在无线环境中提供安全、可信、高效的资源共享、数据交互、安全接入控制、数据溯源和身份认证等服务。

目前，无线网络相关的区块链研究都直接将互联网中经典的区块链协议应用到无线网络环境中。这些区块链协议使用的共识算法通常依赖大量的资源消耗（比如，工作量证明[4]）、复杂的设计（比如，权益证明[5]）或者可靠的通信(比如PBFT[6])。尽管这些区块链协议在互联网上运行良好，但它们不适用于资源有限且信道不稳定的无线网络。无线网络的开放通信特性极大地受到环境的影响，信道带宽不稳定和易受干扰攻击是无线网络通信面临的瓶颈。这些都限制了传统区块链共识算法与无线网络相结合的应用，这也是我们研究无线网络上区块链协议的动机。

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，使之适用于无线网络通信环境。考虑移动无线自组织网络中节点的高动态特性，Z.Jiao 等人 [7] 设计了一种基于工作量证明算法的稳定感知区块链共识协议。该算法根据感知的节点信息和工作量证明选举首领达成共识，在确保协议稳定运行的同时提高性能。根据无线广播通信时延的特点，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。Q.Xu等人[9]利用无线网络广播通信的特点提出基于通信（Proof-of-Comunication）的无线区块链共识协议。Y.Zou等人[10]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[11]。考虑多跳无线网络的通信特征，M.Xu 等人采用最大独立集构建通信骨架的方式，提出了一种能够容错的快速区块链协议 wChain[12]。这些无线区块链共识协议要么依赖参与者的资源实现共识，要么以通信交互的方式实现共识。其中，出块节点的正确性将极大的影响共识过程，一旦出块节点发生故障将直接导致共识失败。

本文为了解决上述无线区块链共识协议面临的问题，提出了一个适用于单跳无线网络的稳定区块链共识协议，该协议是一个类权益证明共识协议。我们的协议采用可验证随机选择方案和门限签名方案实现共识。在该协议中，根据节点的剩余活动时间和生成区块数量定义的节点稳定度，采用随机、非交互的方式选举每一轮的出块节点。参与共识的节点只知道自身是否成为出块节点，并不知道具体是谁当选为出块节点。但是，每个节点都可以独立验证出块节点的合法性。这个设计可以降低选举出块节点的资源消耗和节点被敌手腐蚀的风险。此外，我们采用门限BLS签名方案对出块节点和区块验证共识过程解耦，区块确认终止过程可以由系统中所有正确的共识节点收集足够的投票来实现，并不单独依赖出块节点发出公告。这个设计可以提高系统共识过程的稳定性和处理交易效率，降低由于节点故障或者通信链路不稳定导致无法达成共识的风险。

我们的主要贡献总结如下：

* 我们提出一种融合随机可验证选举方案和阈值BLS签名方案的稳定共识协议，可以在不稳定的无线网络环境中稳定地生成区块并实现共识。
* 我们通过节点的生存期和生成的节点块的数量来定义节点的稳定性。我们确定了权重系数，并分析了节点稳定度函数的敏感性。
* 根据节点的稳定性采取随机可验证的方式选举优质出块节点生成区块；采用门限BLS签名机制解耦出块节点和共识协商过程，提高协议的鲁棒性。
* 区块最终确定过程只能通过一轮部分签名交换来完成。因此，我们共识协议的强一致性可以有效避免区块链分叉。
* 当对手控制的投票权不足50%时，我们的协议保证了正确节点的持久性和活跃性。
* 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。

# Related Work

## 2.1 Blockchain Consensus Protocols

我们将当前的区块链共识协议分为基于资源证明的共识协议和基于协商的共识协议，更详细全面的区块链分类的综述可详见[13]。

基于资源证明的共识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。其中，最典型的基于物理资源证明的共识算法是比特币和以太坊使用的工作量证明(Proof of Work[4])。共识节点通过消耗的计算资源证明来获得出块权限。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Burstcoin项目[15]中的空间证明 (Proof of Sapce, PoSpace)，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。物理资源证明共识算法需要通过消耗物理资源来获得生成区块的权限，这个过程会导致大量资源的浪费，对环境并不友好。为了降低共识的成本，提出了使用虚拟资源证明共识算法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)[5]，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GoChain项目中的信誉证明(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。

基于协商的区块链共识协议中，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Fabric v0.6.0[19]中实现的拜占庭容错算法(Practical Byzantine Fault Tolerant, PBFT[6])。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。典型的协商共识算法还有NEO项目[20]中提的出DBFT算法，授权的共识节点将获取共识权限并通过投票协商对区块达成共识。Zyzzyva[21]中的SBFT共识算法通过让指定节点收集签名降低BFT共识的通信开销，但是仍然需要两轮通信完成签名聚合。此外，Cosmos[22]中所采用的Tendermint共识算法实现了首领节点轮换机制和扩展交易传输，从而提高区块链共识协议的性能；Algorand共识算法[23]采用了拜占庭一致性协议(Byzantine Agreement, BA)与随机可验证选举方案，通过协商投票达成共识。大部分协商类共识算法的共识协商过程依赖可信的首领节点，一旦首领节点出现故障将直接导致共识无法达成，只能通过切换视图来确保共识算法的活性。此外，协商类共识算法通常需要可靠的消息传输模型，因此具有较高的通信时间复杂度。比如，拜占庭共识算法的时间复杂度为，当节点数量增加时，共识时延会急剧提升，平均吞吐量会迅速下降。因此，这类算法更适用于较少网络节点的场景。

## 2.2 Consensus Protocols for Wireless Networks

由于我们得分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，我们简单介绍一下无线网络中共识算法的研究进展。

（待补充）介绍当前有哪些无线网络相关的共识协议

C．Newport和 P.Robinson[]提出了无线系统中的分布式容错共识协议。该协议

目前，一些研究者利用无线网络通信特性提出适用于无线网络的共识协议。利用无线信道的干扰特性，文献[31~34]提出了在无线通信网络中快速达成共识的协议以及实现平均一致性和最大一致性的共识协议。文献[35]利用多径和频率选择性信道的网络模型，为无线传感器网络设计了一种分布式一致性算法。文献[37]提出了一个专为资源受限的无线自组织网络设计的异步拜占庭共识协议。为了提高在无线网络中提高区块链共识的效率，利用无线广播特性设计快速区块链共识协议[3,4,6]。文献[7]中，作者提出基于无线信道证明的单跳无线区块链网络共识协议，该协议可以有效的抵抗干扰攻击。文献[8]中，作者通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，能够快速地完成数据收集并且对提出的区块达成共识。

## 2.3 Threshold Signature Scheme

门限签名方案[40]可以帮助区块链共识协议在通信不稳定的无线网络中达成最终共识。节点利用自己的私钥对共识结果进行签名，通过将不同节点对共识结果的签名聚合成一个唯一的完整签名，最终利用聚合公钥对共识结果进行验证并完成共识。

BLS签名方案[38]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签名方案中，记 $G\_{1}$ 是阶为 $p$ 的循环群，且生成元是 $g\_{1}$, $H:\{0, 1\}^{\*}\rightarrow G\_{1}$ 是一个安全Hash函数，公开参数 $(G\_{1}, g\_{1}, p, H)$ 是全局信息。记节点 $i$ 的私钥为  $pri\_{key}^{i}$，计算得到公钥 $pub\_{key}^{i} = pri\_{key}^{i}\times G\_{1}$，Hash计算要签名的消息 $MSG$ 的确保消息的完整性 $H(MSG)^{i}$，用私钥签名得到 $sign\_{i} = pri\_{key}^{i}\times H(MSG)^{i}$。验证者可以通过签名者的公钥验证 $(g\_{1}, pub\_{key}^{i}, H(MSG), sign\_{i})$ 是否有效。

门限BLS签名方案源于BLS签名方案，由于聚合签名的生成与BLS签名方案是一致且最终恢复聚合签名的结果是一个无需交互的计算过程，因此门限BLS签名方案的工作方式是非交互式和分布式的。门限签名方案由密钥生成算法、签名生成算法和验证算法。密钥生成算法使用分布式密钥生成协议[39]向共识节点分发相应密钥。签名生成算法包含了一个最终签名生成协议，主要是生成节点的签名和根据多个节点的签名计算得到一个完整的签名，最后输出是集合签名的\*\*拉格朗日插值\*\*，签名验证算法使用分布式密钥生成协议生成的聚合公钥对最终签名结果进行验证。

为了确保在无线共识网络中能稳定达成共识，我们将一个 $t-n$ BLS门限签名方案与共识协议相结合。因此，我们的共识协议使得即使部分节点对区块共识结果的签名丢失，也能最终达成共识。由于聚合签名可以由任意共识节点执行，因此除了出块节点之外的正确共识节点都能发布最终共识结果。只要节点能够收集到足够的部分签名聚合形成共识完整的签名，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，减少了与其他的协商通信，从而极大地减少了共识过程中的通信成本。

# Models And Assumptions

## 3.1 Blockchain Basics

在无线区块链系统中，我们假设每个节点局部地维护一个由区块哈希链接形成的区块链，每个区块中会包含多个交易，分别记 $BC, B, tx$ 为区块链、区块和交易。交易由一组引用其他交易的输入和输出组成，以及由其发行者生成的签名以证明其有效性。区块的数据结构包括区块头和区块体，区块体主要包括原始的交易信息，区块头则记录区块的基本信息，包括区块ID，父区块Hash、区块Hash、区块最终签名 $sig\_{f}$ 和时间戳等信息。

## 3.2 Network Model

在本文中，我们考虑了一个由 $N$ 个任意部署在通信区域的节点集合 $V$ 组成的无线网络。在实际场景中，这样的网络可以是构建在一组无人机之间或者车联网之间。所有的网络节点之间是全连接的，即网络中任意一对节点都在彼此的无线通信范围之内。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。假设每个节点知道其他节点的ID、位置以及公钥。我们假设每个节点进入之后会分配得到一个密钥对，由分布式密钥生成协议分发给各个节点。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 Interference Model

无线网络中节点之间的通信会受到环境和干扰的影响，我们假设消息是在瑞利信道中传输。根据无线通信中小尺度衰落的特性，接收节点处的信噪比可以表示为

$$SNR = \frac{P hr^{-\alpha}}{\sigma^2}$$

其中 $P$ 是节点的发射功率； $h$ 表示瑞利衰落中非负功率增益随机变量，服从指数为 $1$ 的负指数分布；$r$ 是两节点之间的距离；$\alpha$ 是路径损耗指数；$是\sigma^2$ 是干扰噪声功率。设定无线网络的信噪比阈值 $\beta$ 是由节点的硬件设备决定的。我们假设每个节点都能够进行物理载波监听。在一个半径为 $R$ 的圆形网络区域中，发送节点到接收节点的距离 $r$ 的密度函数为 $f(r) = \frac{2r}{R^2}$，节点传输消息平均成功的概率为

$P\_s = \int\_0^R P\{SNR >\beta\}f(r)dr = \frac{2\pi\gamma}{N}\int\_0^{\sqrt{\frac{N}{\pi\cdot\gamma}}}\exp\{\frac{-\sigma^2\cdot r^\alpha\cdot \beta}{P}\}rdr,$

当 $\alpha = 2$ 时，节点传输成功的平均概率为 $P\_s = \frac{P\cdot \pi\cdot \gamma}{\sigma^2\cdot \beta\cdot N}\cdot(1 - \exp\{-\frac{\sigma^2\cdot \beta\cdot N}{P\cdot \pi\cdot \gamma}\})$。

## 3.4 Adversary

假设存在敌手可以自由地进出网络，并且最多能够控制网络中不超过 $50\%$ 的投票权，敌手的恶意行为如下：

\* 敌手节点可以发起女巫攻击，即伪造多个身份，在共识过程中并不传输任何对达成共识有效的消息或者生成有效区块；

\* 假设敌手可以在任意时刻制造噪声干扰其他诚实节点的消息传输。敌手可以发起阻塞攻击(Jamming Attack)，在长度为 $T$ 的时间区间内，敌手最多可以发起 $(1-\epsilon)T, 0<\epsilon\leq 1$ 次阻塞攻击，即在 $T$ 轮中敌手最多可以阻塞 $(1-\epsilon)T$ 轮。

在本文中，我们说一个事件 $E$ 有很高概率发生,如果对任意 $c\geq 1$，事件 $E$ 发生的概率为 $1 - \frac{1}{N^{c}}$。

# The Stable Consensus Protocol

在本小节，我们首先介绍稳定共识协议的概览，之后介绍协议的细节。

## 4.1 Overall Architecture

在本小节中，我们主要提出稳定共识协议的概述，并且通过描述节点功能更加简洁的表述稳定协议。

稳定共识协议是在无线网络环境下工作的，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，即质押金钱获得在系统中的活动时间。这笔押金会存放在一个虚拟账户中，任何人不能取出，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。这个质押机制可以有效的防止敌手发起女巫攻击。

稳定的无线区块链共识协议允许数百个节点参与共识协商过程，确保能够稳定的达成最终共识。共识协议是按顺序进行的。在每一轮中，随机选举一个出块节点，随后每个共识节点对区块进行一次投票。每个共识节点在一轮共识中只能对共识结果进行一次投票，记节点对区块Hash的签名被计数为对区块有效的一次投票。如下图所示，我们的共识协议主要包括：

![](Fig\_1.png)

\* \*\*随机数生成\*\*：节点通过一个分布式随机生成方案根据前一轮确认的区块独立地生成一个随机数 $[0, 1)$ 区间的随机数；

\* \*\*出块节点选举\*\*：对当前所有参与共识的节点进行统一排序，根据节点的稳定度构建轮盘。节点将自己的私钥和当前轮的随机数作为可验证随机函数的输入确定自己是否成为出块节点，并且可以通过输入出块节点的公钥验证出块节点的合法性；

\* \*\*区块生成\*\*：出块节点将打包交易生成一个区块，并将区块广播给网络中其他共识节点；

\* \*\*区块验证\*\*：接收到新区块之后，共识节点会验证区块的有效性。如果验证区块成功节点将会对区块Hash签名，并将签名广播， 否则将不会对区块Hash签名；

\* \*\*区块确认\*\*：节点在以下两种情况下将确认区块：

  \* 节点收集到足够的部分签名之后，通过一个聚合签名恢复方案将一定数量的区块Hash签名并验证成功后，聚合成一个完整签名；

  \* 节点接收到区块Hash的完整签名并验证成功。

\* \*\*链更新阶段\*\*：当节点生成或接收到一个区块Hash的完整签名并验证成功后，就完成区块共识的最终性。节点可以将新区块添加到本地主链上，并开始新的一轮共识。

Algorithm 1表示了稳定共识协议每一轮共识所需要执行的流程。每个阶段距离的细节将会在之后的小节中详细介绍。我们解决了出块节点的选举、区块验证和最终确认面临的挑战，确保了系统的安全性。我们的协议确保了即使存在敌手，也能顺利的运行。

![](./consensus.png)

## 4.2 The Stable Consensus Protocol

在这一小节，我们将详细介绍我们的共识算法的设计细节。

### 4.2.1 门限签名方案

在系统运行共识协议之前，系统通过一个安全的分布式密钥生成算法，为每个参与共识的节点生成和分配密钥对。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签名内容的部分签名；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和完整签名的有效性。在我们的区块链协议中，每个节点在验证区块成功之后，将通过签名生成函数利用自身密钥生成区块Hash的部分签名并广播。其他接收到该签名的节点可以签名节点的公钥验证部分签名的有效性。当节点收集到超过门限的部分签名并验证成功后，会通过聚合签名恢复函数生成一个完整签名并广播。接收到完整签名的节点都可以通过聚合公钥对完整签名进行验证。

### 4.2.2 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 $<sk\_{i}, pk\_{i}>$ 和其他共识节点的公钥 $\{pk\_{1},\cdots, pk\_{i-1}, pk\_{i+1}, \cdots, pk\_{n}\}$ 以及聚合公钥 $PKs$，用于签名和验证签名。当得到所有共识节点的公钥之后，节点可以通过对所有的共识节点采用基于公钥Hash值进行排序。这种方式可以使得全部参与共识的节点拥有相同的序列。

在稳定区块链共识协议中，主要根据节点的稳定度来选举出块节点。新节点加入系统时，通过保证金质押获得有限的活动时间，活动的时长与交付的保证金成正比。记节点 ${\rm Node}\_{v}$ 在区块链系统中的<font color=red>剩余活动时间</font>为 $T\_{v}$，则所有共识节点的剩余活动时间之和为 $\sum\_{v}T\_{v}$。定义节点 ${\rm Node}\_{v}$ 的 \*\*<font color=red>剩余活动时间比值</font>\*\* 为

$$\rho\_{v} = \frac{T\_{v}}{\sum\_{v}T\_{v}}$$

假设在最近的 $K$ 个确认区块中，由节点 ${\rm Node}\_{v}$  生成的区块数量为 $N\_{v}$。定义节点 ${\rm Node}\_{v}$ 的 \*\*<font color=red>共识比值</font>\*\* 为

$$r\_{v}=\frac{N\_{v}}{K}$$

定义节点 ${\rm Node}\_{v}$ 的 \*\*<font color=red>稳定度</font>\*\* 为

$$S\_{v}=\alpha\times \rho\_{v}+\beta\times r\_{v}\qquad(\alpha+\beta=1,\ \alpha\geq 0,\ \beta\geq 0)$$

其中，$\alpha$ 为剩余活动时间比权重系数，$\beta$ 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 $K$ 个时，记节点的共识比为 $r\_{v}=0$。此时，节点的稳定度主要受节点的剩余活动时间的影响。

在稳定区块链共识协议中，节点 ${\rm Node}\_{v}$ 被选为出块节点的概率为

$$p\_{v}=\frac{S\_v}{\sum\_{v}S\_v}$$

根据节点的稳定度决定节点被选中的概率，稳定度越高的节点越容易被选中。通过节点的稳定度和统一排序，可以构建一个轮盘。合法的出块节点将会从节点轮盘中选出。共识节点根据轮随机数 $Rds^{r}$ 从构建的轮盘中决定出块节点的索引BPI。记 $S\_{i}$ 是节点 $Node\_{i}$（$i=0,\dots,N-1$）的稳定度，所有节点的稳定度之和为 $S =\sum\_{i=1}^{N}S\_{i}$ ，那么节点 $i$ 被选中的概率为 $p\_{i}=\frac{S\_{i}}{S}$ 且有 $\sum\_{i=1}^{N}p\_{i}=1$。为了确定被选中的节点，将区间 $[0, 1)$ 分为连续的多个区间

$$[\sum\_{k=1}^{i}p\_{k}, \sum\_{k=1}^{i+1}p\_{k}),\ i=0,\dots,N-1.$$

在稳定共识协议中，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、可验证的输出。

在我们的协议中，当共识节点接收到当前轮区块Hash的完整签名 $sig\_{f}^{r}$ 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 $Rds^{r+1} \in [0, 1)$。为了确保输入值的统一和即时性，我们使用当前轮的区块Hash $B\_{Hash}^{r}$ 与完整签名 $sig\_{f}^{r}$ 结合作为随机输入。最后得到下一轮的随机数:

$$Rds^{r+1} = \frac{Hash(B\_{Hash}^{r}||sig\_{f}^{r})}{2^{len(Hash(B\_{Hash}^{r}||sig\_{f}^{r}))}}$$

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 $B\_{Hash}^{r}$ 是一个预先知道的信息，但是完整签名 $sig\_{f}^{r}$ 在聚合产生之前是不可以事先预估得到的。即使存在节点首先恢复出完整签名，但是节点是不可能篡改完整签名的。因此，新一轮随机数 $Rds^{r+1}$ 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 $sig\_{f}^{r}$，就会立刻确认区块并将区块添加到本地区块链上。同时，节点可以根据确认的区块Hash $B\_{Hash}^{r}$ 和完整签名 $sig\_{f}^{r}$ 自主生成新一轮的随机数 $Rds^{r+ 1}$，直接开始新一轮共识进程。

随机可验证函数结合了分布式随机数生成协议和安全可靠的BLS签名方案，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随机数和自己的私钥，可以确定自己是否当选为出块节点。同时，可以利用出块节点的公钥和随机数，验证出块节点的合法性。我们采用的分布式随机生成协议是建立在安全可靠的BLS签名方案之上的，在 $N$ 个共识节点中即使存在 $\lfloor\frac{n}{2}\rfloor$ 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知道其他节点的活动时间和维护相同的区块链时，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。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。最先聚合生成完整签名的部分签名集合的所有节点将获得部分奖励。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。

出块节点的选举是每个共识节点将分布式随机数生成协议的输出和私钥作为可验证随机函数的输入，确定自己是否当选。如果随机值 $Rds^{r}$ 是在拥有私钥的节点所在的区间则当选为出块节点，计算如下：

$$ BPI = \{i | Rds^{r} \in [\sum\_{k=1}^{i}p\_{k}, \sum\_{k=1}^{i+1}p\_{k})\}$$

出块节点的选择是基于 $Rds^{r}$，这个数是随机、唯一和可验证的。因此，出块节点的选择也是安全随机的，并且出块节点的确定也是每个节点独立完成，不需要进行任何的消息交互。由于所有共识节点计算BPI的输入的相同的，因此最终计算结果也是唯一。通过出块节点在区块上的签名和出块节点的公钥可以验证出块节点的合法性。

当节点维护的区块链或者剩余活动存在差异时，最终的出块节点的选举并不会出现较大的偏差，即敏感度并不高。

记 $S\_{v} = \alpha\times\frac{T\_{v}}{\sum\_{i \in N}T\_{i}} + \beta\times\frac{N\_{v}}{K}$，记 $S\_{v}' = \alpha\times\frac{T\_{v}+ \Delta T\_{v}}{\sum\_{i \in N}T\_{i} + \Delta T\_{v}} + \beta\times\frac{N\_{v} + \Delta N\_{v}}{K}$, 那么 $||S\_{v}' - S\_{v}|| = ||\alpha\frac{\sum\_{i \in N}T\_{i}\Delta T\_{v} - T\_{v}\Delta T\_{v}}{\sum\_{i \in N}T\_{i}(\sum\_{i \in N}T\_{i}+\Delta T\_{v})} + \beta\frac{\Delta N\_{v}}{K}||\leq ||\frac{\Delta T\_{v} }{\sum\_{i \in N}T\_{i}+\Delta T\_{v}} + \frac{\Delta N\_{v}}{K}|| \ll \Omega(1)$

因此，当节点记录的剩余活动时间和共识区块数量存在些许误差时，对于最终稳定性的影响不会太大，节点被选中的概率是对于这两个度量指标的敏感度也不会特别大。因此，诚实节点选中不同出块节点的概率是非常低的。为了尽可能使系统中节点维护相同的区块链，在每次同时开始之前，节点会执行一次区块链同步操作。节点会随机请求几个邻节点的区块链信息，最终同步拥有共同链前缀且有最长有效区块链。

当节点被选中为当前轮的出块节点之后，出块节点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 $(R, Hash\_{pre}, inf\_{trans}, B\_{Hash}, L\_{id},sig\_{f})$，其中 $R$ 是当前的轮数，$Hash\_{pre}$ 是前一个区块的Hash，$inf\_{trans}$ 是存储在区块体中交易的信息，$L\_{id}$ 是出块节点的id，$B\_{Hash}$ 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 $sig\_{f}$，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。

### 4.2.3 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\*\* $L\_{id}$：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\*\* $Hash\_{pre}$：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\*\* $inf\_{trans}$：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；

\* \*\*部分签名\*\*：验证区块Hash的签名是否有效。验证区块Hash的有效性可以确保区块完整性，验证区块Hash的签名可以确定出块节点的合法性，所有节点可以通过签名节点的公钥来确定部分签名的有效性。

如果以上条件都满足后，节点验证区块的有效性都成功，会对区块Hash签名并广播部分签名给其他共识节点。当共识节点收集到 $\lceil\frac{N}{2}\rceil$ 个有效的区块Hash的部分签名时，可以聚合恢复成一个区块Hash的完整签名就可以证明区块被确认，并不需要更多的消息通信。当一个有效的完整签名出现，说明已经有足够多的节点认为这个区块是有效的，这个签名可以通过聚合公钥快速验证。当有一个节点广播了完整签名，所有诚实的节点都会在 $\log N$ 轮通信中接收到完整签名。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，节点只需要单向传输部分签名，并不需要其他节点的回复，这极大地减少了节点之间的通信。此外，聚合多个部分签名生成完整签名可以由任意共识节点完成，即区块的确认过程是完全去中心化的、无首领的。这样的设计使得即使出块节点故障或者无线网络通信不稳定，我们的共识协议也能确保最终确认有效区块。

### 4.2.4 节点故障时协议的操作

\* \*\*节点故障\*\*：在我们的共识协议中，出块节点只承担生成区块的功能，出块节点的恶意行为只能是生成一个无效区块或者不生成区块。为了确保共识协议的持续运行，如果被选中的出块节点出现故障，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。协议中一轮共识进程结束之后，会出现两种输出：输出一个有效的被确认的区块和一个空区块（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：

  \* 出块节点创建一个有效区块；

  \* 该有效区块被足够多的共识节点接受并且为其Hash签名。

  如果一个合法的出块节点创建一个无效区块，则其他节点会为一个空区块的Hash签名并广播表明当前出块节点的无效行为，最终确定当前轮是无效的。如果出块节点不生成区块或生成的有效区块没有被足够多的共识节点接受并对区块Hash签名，要么出现了网络问题，要么部分共识节点拒绝对区块Hash签名。这种情况，我们设置一个超时机制，所有诚实的共识节点会在一个空区块上签名，最终确认该区块并开始新一轮的共识进程。只有出现超时或者出块节点创建一个无效区块时，共识节点才在一个空区块上签名。我们的协议可以确保为空块或者有效块收集到足够多的区块Hash部分签名，之后将所有的部分签名聚合恢复成一个完整签名，将确认区块的Hash和完整签名作为输入，选出下一轮的出块节点。从而，协议可以确保敌手无法干扰新区块的出块节点的选举过程 ，提高系统的安全性。

#### (1) Sibil Attack

当女巫节点出现恶意行为时，将发送一些错误的消息阻碍协议达成共识。如果女巫节点当选为出块节点生成一个无效区块，其他共识节点将会验证区块失败，最终生成和确认一个空区块。当生成一个有效区块，敌手控制的女巫节点可以不对区块Hash签名来影响共识过程。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。只要有诚实节点收集到足够的签名确认消息，将会生成完整签名确认区块。

#### (2) Jamming Attack

在无线网络中，节点发起阻塞攻击（Jamming Attack）时，会阻塞其他节点传输消息。可能会导致其他诚实节点即使对有效区块Hash签名也无法成功广播部分签名。我们的协议中根据估计的敌手时间窗口动态调整节点收集签名的超时时间，确保节点能够在敌手发起阻塞攻击时也达成共识。我们的共识协议中，每个节点根据感应信道的情况维护一个敌手时间窗口估计 $T$。每个节点会根据共识过程中信道的情况，动态调整 $T$ 的大小，当 $T$ 达到某一阈值时需要更改超时的阈值。延长节点接收签名消息的超时时间，最终确保敌手发起阻塞攻击时也能对有效区块达成共识，降低敌手发起干扰攻击共识过程的影响。

当敌手想要篡改区块链数据时，即使篡改最后一个区块的数据，也需要获得至少 $\lceil\frac{N}{2}\rceil$ 个节点的私钥。节点获取私钥要么通过分布式密钥生成协议获取，要么贿赂节点获取。节点通过分布式密钥生成协议只能获得属于自己的密钥，因此，敌手很大概率只能通过贿赂节点获取多个节点的密钥。要贿赂 $\lceil\frac{N}{2}\rceil$ 个节点需要掌控整个共识协议超过 $50\%$ 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，并且也没有任何的收益。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，稳定共识协议能够确保数据的安全性。

## 4.3 Reward and Punishment Mechanism

为了提高节点愿意参与共识的积极性，出块节点在区块被确认之后会获得来自系统的区块奖励。为了提高节点愿意对区块Hash签名的积极性，我们的协议将为提供部分签名的节点发放签名奖励。

记聚合完整签名的部分签名集合为 $Sigs = \{sig\_{1}, sig{2}, \cdots, sig\_{m}\}$，每个签名对应的时间戳分别为 $Ts = \{t\_{1}, t\_{2}, \cdots, t\_{m}\}$。针对可能存在节点接收到多个最终签名是由不同签名组成，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。假设系统存在两个聚合完整签名的部分签名集合 $Sigs\_{1} = \{sig\_{11}, sig{12}, \cdots, sig\_{1m\_{1}}\}, signs\_{2} = \{sig\_{21}, sig{22}, \cdots, sig\_{2m\_{2}}\}$， 每个签名的时间戳分别对应为 $Ts\_{1} = \{t\_{11}, t\_{12}, \cdots, t\_{1m\_{1}}\}, Ts\_{2} = \{t\_{21}, t\_{22}, \cdots, t\_{2m\_{2}}\}$。两个签名集合中平均时间戳分别为 $T\_{avg}^{1} = \frac{\sum\_{i = 1}^{m\_1} t\_i}{m\_1}，T\_{avg}^{2} = \frac{\sum\_{i = 1}^{m\_2} t\_i}{m\_2}$。如果 $T\_{avg}^{1} < T\_{avg}^{2}$，则签名奖励将会被均分给 $Sigs\_{1}$ 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，进一步提高系统的性能。

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作出块节点的概率，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，会减少生成无线区块的情况，从而提高系统的安全性和效率。

# Protocol Analysis

在本小节，我们主要分析系统的安全性和性能。通过分析系统的安全性来证明我们的协议能够为无线区块链系统提供持续性和活性，通过对系统达成共识的开销分析系统的性能，可以证明我们的协议的高效性和适用于无线区块链网络环境。

## 5.1 Security Analysis

区块链的共识算法需要确保确保所有的共识节点对系统中的交易历史达成共识。当系统中绝大多数节点是诚实的，我们的协议可以保证持续性和活性。

为了确保安全性，稳定共识协议需要至少 $\lceil\frac{n+1}{2}\rceil$ 个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和鲁棒的，这个机制具有两个重要的性质：

\* \*\*唯一性\*\*：门限签名机制中任意超过 $\lceil\frac{n+1}{2}\rceil$ 个部分签名恢复的完整签名都是相同的。

\* \*\*可验证性\*\*：最终恢复的完整签名可以被持有聚合公钥的任意节点验证。聚合公钥是在节点加入系统的时候由网络的密钥生成协议创建和分配的。

### 5.1.1 Persistence Analysis

我们的共识协议并以依赖出块节点确保安全性。恶意的节点可能会提出一个冲突区块来对共识发起攻击。由于在每一轮中，诚实节点只会对认可的首领节点生成的有效区块的区块Hash签名确认一次。因此，在一轮共识结束后，要么确认一个有效区块，要么确认一个空块，并且所有的诚实节点将会有相同的操作。我们的共识协议满足持续性：

\* \*\*定理一（持续性）\*\*：所有诚实的节点将维护相同的区块链，即诚实节点 ${\rm Node}\_{v}$ 声明交易 $tx$ 在区块链第 $i$ 个区块的第 $j$ 个位置上, 则最终所有诚实节点的区块链的第 $i$ 个区块的第 $j$ 个位置的交易一定是 $tx$。

\*\*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 $tx\_{u}\in BC\_{u}, tx\_{v}\in BC\_{v}$ 分别是诚实节点 ${rm Node}\_{u}, {rm Node}\_{v}$ 维护的区块链上相同位置的交易，且 $tx\_{u}\neq tx\_{v}$。那么会有两种情况出现：

  \* $tx\_{u}, tx\_{v}$ 在同一轮中分别被添加到节点${rm Node}\_{u}, {rm Node}\_{v}$ 的本地区块链 $BC\_{u}, BC\_{v}$ 上。这表明首领节点在同一轮中生成并确认了两个不同的区块，这违反了我们协议中的门限签名机制的唯一性，因此与假设相矛盾。

  \* $tx\_{u}, tx\_{v}$ 在不同的轮被添加到两个节点的区块链上。假设 $tx\_{u}, tx\_{v}$ 分别在 $r\_{m}, r\_{n}(m < n)$ 轮中被添加到两个节点的区块链 $BC\_{u}, BC\_{v}$。根据我们的协议，在 $r\_{m}$ 轮中添加 $tx\_{u}$ 到区块链上的节点至少有 $\lceil\frac{n+1}{2}\rceil$ 个，这些节点认可交易 $tx\_{u}$ 是在 $r\_{u}$ 轮首次被添加到的区块链的第 $i$ 个区块中第 $j$ 个交易。由于在一轮中不会最终确认两个不同的区块，因此添加交易 $tx\_{u}$ 的节点在 $r\_{m}$ 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 $i$ 个区块中第 $j$ 个交易 $tx\_{i}^{j}$ 是相同的。我们假设 $tx\_{v} \neq tx\_{u}$，且 $r\_{m} < r\_{n}$。当节点 $v$ 在 $r\_{m}$ 之前故障了并且在 $r\_{n}$ 的时候恢复，此时节点将会同步区块链的最新信息。那么将会有至少 $\lceil\frac{n+1}{2}\rceil$ 个节点在第 $r\_{n}$ 轮的时候认可区块链的第 $i$ 个区块的第 $j$ 个交易是 $tx\_{v}$。由于之前在 $r\_{m}$ 轮中也至少有 $\lceil\frac{n+1}{2}\rceil$ 个节点认可区块链在第 $i$ 个区块的第 $j$ 个交易是 $tx\_{u}$。此时，网络大小为 $n > \lceil\frac{n+1}{2}\rceil + \lceil \frac{n+1}{2}\rceil = \lceil n+ 1 \rceil$，矛盾了。因此 $tx\_{u} = tx\_{v}$，与假设 $tx\_{u} \neq tx\_{v}$ 相矛盾。

因此得出结论，在每一轮中，即使存在矛盾区块，所有的诚实节点将会确认相同的有效区块。最终，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。

### 5.1.2 Liveness Analysis

对于活性，当至少 $\lceil\frac{n+1}{2}\rceil$ 个诚实节点能够对有效区块或者空块签名，并且最终只要有一个节点聚合足够的部分签名形成完整签名，就能够完成共识过程。

\* \*\*定理二（活性）\*\*：当诚实节点数量超过 $\lceil\frac{n+1}{2}\rceil$ 时，即使系统中存在 $\lfloor\frac{n}{2}\rfloor$ 个故障节点，诚实节点也能够完成共识并获得新一轮的随机源。

\*\*证明\*\*：

  \* 出块节点故障：由于我们的共识协议并不依赖出块节点来确保安全性，因此也不依赖出块节点的正确性来确保活性。如果出块节点出现故障未创建有效区块或者创建一个无效区块，诚实的共识节点在超时未收到有效区块或者验证区块无效之后，会对一个空块的区块Hash签名，最终收集到至少 $\lceil\frac{n+1}{2}\rceil$ 个部分签名的节点会生成完整签名，完成当前轮的共识过程并获得新一轮的随机源——完整签名。

  \* 系统中存在 $\lfloor\frac{n}{2}\rfloor$ 个非出块节点故障：由于出块节点未发生故障，因此会生成一个有效的区块。其他 $n - \lfloor\frac{n}{2}\rfloor \geq \lceil\frac{n+1}{2}\rceil$ 个诚实的节点在验证区块成功后，会生成有效区块Hash的部分签名并广播给其他节点，故障节点会拒绝为区块Hash签名。但是由于诚实节点的数量超过安全阈值，因此会有至少 $\lceil\frac{n+1}{2}\rceil$ 个节点会生成有效部分签名。诚实节点会收集到足够的部分签名生成有效区块Hash的完整签名并广播。因此最终会完成当前轮的共识过程并获得新一轮的随机源。当由于敌手发起网络阻塞攻击导致最终诚实节点没有生成有效区块Hash的最终签名时，最终所有诚实的节点会因为超时而确认一个空区块。并开始新一轮的共识过程并获得下一轮的随机源。

我们的协议确保当节点故障时也能够完成一轮共识过程，并获得下一轮的随机源。因此，我们的共识协议是能够确保活性的。

### 5.1.3 Sybil Attack Analysis

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 $p\_{A}$，其余 $m$ 个节点的概率分别为 $\{p\_1, p\_2, \cdots, p\_m\}$。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 $\frac{1}{n}\sum\_{k = 1}^n k\cdot p\_k, (\sum\_{k = 1}^n p\_k = p\_{A})$。此时有 $\frac{1}{n}\sum\_{k = 1}^n k\cdot p\_k < p\_A$。

\*\*证明\*\*：令 $p\_k = \frac{p\_A}{n}$，则有 $\frac{1}{n}\sum\_{k = 1}^n k\cdot p\_k = \frac{1}{n}\sum\_{k = 1}^n k\cdot \frac{p\_A}{n} = p\_A\sum\_{k=1}^n\frac{k}{n^2} = p\_A\frac{\sum\_{k=1}^n k}{n^2}$。当 $n > 1$ 是有 $\sum\_{k = 1}^n < n^2$ ，因此 $p\_A\frac{\sum\_{k=1}^n k}{n^2} < p\_A, n>1$。最终得出结论 $\frac{1}{n}\sum\_{k = 1}^n k\cdot p\_k < p\_A, n > 1$。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，则成为出块节点的概率将会越低。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，因此我们的协议能够较好的抵抗女巫攻击。

### 5.1.4 Jamming Attack Analysis

我们的协议能够较好的抵抗节点发起阻塞攻击影响共识过程。假设节点在 $T$ 轮通信中最多能够发起 $(1 - \epsilon)T$ 轮阻塞攻击。在共识节点开始新一轮的共识过程时，节点会根据检测信道的情况来动态调整收集部分签名的超时阈值。

当没有敌手发起阻塞攻击时，假设每个节点成功传输消息的概率为 $p$，可以在 $K$ 轮中以较高的概率传输部分签名成功。计算 $1 - (1 - p)^{K} \geq 1 - \frac{1}{N^{2}}$，从而就有 $1 - (1 - p)^{K} \geq 1 - e^{-pK} \geq 1 - \frac{1}{N^{2}}$，可知共识节点在 $\frac{2}{p}\log n$ 轮中有很高的概率可以传输部分签名成功。因此，在 $\lceil\frac{n+1}{2}\rceil \frac{2}{p}\log n$ 轮后，共识节点有很高的概率可以生成一个完整签名，最终确认区块。

当敌手发起阻塞攻击时，诚实节点能够成功传输的轮数为 $\frac{2\epsilon}{p}\log n$，成功生成完整签名的轮数为 $\lceil\frac{n+1}{2}\rceil \frac{2\epsilon}{p}\log n$。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。我们的协议通过根据节点对于敌手攻击窗口的估计来动态调整超时机制的阈值，降低阻塞攻击对于共识过程的影响。

## 5.2 Performance Analysis

在协议的共识过程中，只有在出块节点生成有效区块，并有节点接收到至少 $t = \lceil\frac{N+1}{2}\rceil$ 个区块Hash的部分签名时，才会生成完整签名确认有效区块。节点接收到至少 $t$ 个区块Hash的部分签名的概率为 $P\_{rs} = \sum\_{k = t}^{N-1}C\_{N-1}^{k}P\_{s}^{k}(1 - P\_{s})^{N-1-k}$，共识协议生成有效区块的概率为

$$P\_{valid} = P\{L \geq t\} = \sum\_{L=t}^{N-1}C\_{N-1}^L P\_{rs}^L(1 - P\_{rs})^{N-1-L}$$

其中 $L$ 为成功接收到 $t$ 以上个区块签名份额节点的数量，当 $L\geq t$ 时说明系统中大部分节点都能将有效区块添加到本地区块链上。因此，生成空块的的概率为

$$P\_{null} = 1 - P\_{valid}.$$

## 5.2.1 Computational Cost Analysis

在我们的共识协议中，主要的算力开销来自共识过程中出块节点的选择过程的计算，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区块Hash签名的计算，以及生成和验证完整签名时的计算。

记协议中每个节点的算力为 $r$，令区块中交易的打包时间为 $T\_{c}$，区块哈希和签名的计算时间都为 $T\_{s}$。

\* 生成一个空块的哈希次数为 $NrT\_{s} + (N-1)NrT\_{s} + NrT\_{s} = (N+1)NrT\_{s} \approx O(N^2rT\_{s})$；

\* 生成一个有效区块的哈希次数为 $NrT\_{s} + Nr(T\_{s}+T\_{c}) + (N-1)NrT\_{s} + NrT\_{s} = Nr(T\_{s}+T\_{c}) + (N+1)NrT\_{s} \approx O(N^{2}r(T\_{s}+T\_{c})$。

假设在生成一个有效区块之前已经生成过 $m$ 次空块，当生成空块的概率 $0\leq P\_{null} < 1$时，生成一个有效区块所需的哈希次数为

$H\_{Stable} = \sum\_{m=0}^{\infty}P\_{null}^mP\_{valid}(m\cdot(N+1)NrT\_{s} + Nr(T\_{s}+T\_{c}) + (N+1)NrT\_{s})$。最终计算得到生成一个有效区块的平均哈希次数为

$$H\_{Stable} = \left\{

      \begin{aligned}

      & \frac{(N+1)NrT\_{s}}{1 - P\_{null}} + Nr(T\_{s}+T\_{c}), \text{if } P\_{null}\neq 1, \\

      & \infty, \text{ if } P\_{null} = 1.

      \end{aligned}

      \right.$$

## 5.2.2 Communication Cost Analysis

在我们的共识协议中生成一个有效区块主要的时间包括打包生成和验证区块（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 $T\_{b}$ ，打包有效区块的交易时间为 $T\_{c}$，生成区块Hash和签名的时间为 $T\_{s}$。

\* 生成一个有效区块的时间为 $T\_{1} = 2T\_{c} + 4T\_{s} + 3T\_{b}$；

\* 生成一个空块的时间为 $T\_{2} = 3T\_b + 4T\_s$。

假设生成有效区块的通信次数为 $C\_{normal}$；生成空块的通信次数为 $C\_{null}$。假设生成一个有效区块之前连续 $m$ 次生成空块。

  $$C\_{Stable} = \left\{

      \begin{aligned}

      & \sum\_{m=0}^\infty P\_{null}^m\cdot P\_{valid}\cdot (mC\_{null} + C\_{normal}), \text{if } P\_{null}\neq 1, \\

      & \infty, \text{ if } P\_{null} = 1.

      \end{aligned}

      \right.$$

  \* \*\*生成有效区块\*\*：假设交易到达速率为 $\lambda$，协议在 $T\_{1}$ 时间内生成一个有效区块，则交易到达数量为 $\lambda T\_{1}$。生成一个有效区块的通信次数为 $C\_{normal} = \lambda T\_{1}(N-1) + (N-1) + （N-1) + (N-1) = (N-1)(\lambda T\_{1} + 3)$；

  \* \*\*生成空块\*\*：出块节点在 $T\_1$ 时间内生成一个区块但是确认失败。所有节点会在 $T\_{2}$ 时间内生成一个空块，并最终确认该空块。这段时间内的交易到达数为 $\lambda T\_{2}$，增加的通信数是 $\lambda T\_{2}(N-1) + (N-1) + (N-1) + (N-1) = (N-1)(\lambda T\_{2} + 3)$。生成空块的通信数量为 $C\_{null} = C\_{normal} +(N-1)(\lambda T\_{2} + 3) = (N-1)[\lambda(T\_1 + T\_2) + 6]$。

最后计算得到在我们的协议中平均通信数量为

  $$C\_{Stable} = \left\{

      \begin{aligned}

      & \frac{P\_{null}\cdot(N-1)\cdot(\lambda T\_{2} + 3)}{1 - P\_{null}} + \frac{(N-1)\cdot(\lambda T\_{1} + 3)}{1 - P\_{null}}, \text{if } P\_{null}\neq 1, \\

      & \infty, \text{ if } P\_{null} = 1.

      \end{aligned}

      \right.$$

## 5.2.3 Stability Analysis

每个节点在注册进入系统时，通过支付金钱来获取在系统中的活动时间。支付的金钱将通过购买基础设施或者做一些投资来获取盈利。因此，关于活动时间的价格可以根据价格在一段时间内的波动现象和规律来对活动时间的价格进行预测定价。

节点可以通过增加金额来延长活动时间，也可以重新注册新的节点来获得活动时间。记 $N$ 为共识节点集合，$T\_{i}$ 是节点 $i\in N$ 的活动时间，用户增加活动时间有两种方式：一种是通过充值金钱延长节点 $i$ 的活动时间；一种是通过注册新节点延长用户的活动时间。

\* 第一种情况时用户的稳定度概率计算如下：

  \* 节点的活动时间比为 $\rho\_{i} = \frac{T\_{i} + \Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} $

  \* 节点的共识比 $r\_i = \frac{N\_{i} + \Delta N\_i}{K}$

  \* 节点的稳定度为 $w\_{i} = \alpha \rho\_{i} + \beta r\_{i} = \alpha \frac{T\_{i} + \Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{N\_{i} + \Delta N\_i}{K}$

  \* 节点被选中的概率为 $p\_{i} = \frac{w\_{i}}{\sum\_{j\in N} w\_{j} + \alpha \Delta \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K}}$

\* 第二种情况时用户稳定度概率计算如下：

  \* 用户所有节点的活动时间比为 $\rho\_{i}' = \frac{T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} = \rho\_{i}'' + \Delta \rho\_{i}$

  \* 节点的共识比 $r\_{i}'= \frac{N\_{i}}{K} +  \frac{\Delta N\_i}{K} = r\_{i}'' + \Delta r\_{i}$

  \* 两个节点的稳定度分别为：$w\_i'' = \alpha \frac{T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{N\_{i}}{K} = \alpha \rho\_{i}'' + \beta r\_{i}'', \Delta w\_{i} = \alpha \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K} =\alpha \Delta\rho\_{i} + \beta \Delta r\_{i}$

  \* 两个节点的被选中的概率分别为 $p\_{i}'' = \frac{w\_{i}''}{\sum\_{j\in N} w\_{j} + \alpha \Delta \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K}}, \Delta p\_{i} = \frac{\Delta w\_{i}}{\sum\_{j\in N} w\_{j} + \alpha \Delta \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K}}$。

通过计算可知 $p\_{i} = \frac{\alpha \frac{T\_{i} + \Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{N\_{i} + \Delta N\_i}{K}}{\sum\_{j\in N} w\_{j} + \alpha \Delta \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K}} = \frac{\alpha \frac{T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{N\_{i}}{K}}{\sum\_{j\in N} w\_{j} + \alpha \Delta \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K}}  + \frac{\alpha \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K}}{\sum\_{j\in N} w\_{j} + \alpha \Delta \frac{\Delta T\_{i}}{\sum\_{j\in N}T\_j + \Delta T\_{i}} + \beta \frac{\Delta N\_i}{K}} = \alpha\rho\_{i}'' + \beta r\_{i}'' + \alpha \Delta\rho\_{i} + \beta \Delta r\_{i}  = p\_{i}''$。因此，不管是直接注资延长节点的活动时间还是注册新节点增加活动时间，最终用户控制的节点被选中成为出块节点的概率是相等的。

# Simulation Result

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。

为了估计协议的性能，我们主要采用两个指标：吞吐量和共识时延。区块链的共识时延和吞吐量的定义如下：

\* \*\*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，表示为

$$Throughput = \frac{\text{number of transactions}}{\text{consensus latency}}$$

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 Weight Coefficient

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。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线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提高共识效率，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。为了确定权重系数，我们对不同的权重系数进行了多次对比实验。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印象都比较大。

实验考虑的场景是在 $N = 500$ 个区块链节点中，有 $\frac{N}{4}$ 的节点活动时间比和共识比都比较低，有 $\frac{N}{4}$ 的节点活动时间比高但共识比低，有 $\frac{N}{4}$ 的节点活动时间比高和共识比低，有 $\frac{N}{4}$ 的节点活动时间比和共识比都比较高，如下图所示：

![](Weight.png)

这些节点的都是事先排序的，为了更具有说服力，我们将四种节点的序号均匀的分布。共识的次数我们设置为 $100$ 次，即每个 $\alpha$ 都会测试 100 次共识的情况。除了 $\alpha$ 变化以外，其余所有参数都一样。实验最后确认不同的 $\alpha$ 下，优质稳定点被选为出块节点的比率。

![](weight1.png)

从上图可以看出，当 $alpha = 0.5$ 左右时，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 $\alpha = \beta = 0.5$，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 ${\rm Node}\_v$ 的稳定度为 $S\_{v}=\alpha\times \rho\_{v}+\beta\times r\_{v}$。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 Block Size

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 $0.5MB$ 到 $5MB$ 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：

![](blocksize\_latency.png)

![](blocksize\_throughput.png)

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 Number of Nodes

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：

![](num\_nodes\_latency.png)

![](num\_nodes\_throughput.png)

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 Bandwidth

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 $10Mbps$ 增加到 $130Mbps$ 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：

![](bandwidth\_latency.png)

![](bandwidth\_throughput.png)

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 Sybil Attack

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：

![](Sybil\_Attack.png)

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。最终结果表明，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 Jamming Attack

敌手可以通过发起阻塞攻击影响无线网络中消息的传输，最终影响区块的传输和确认过程，导致最终无法实现区块最终性。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 $(1-\epsilon)\*T$ 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 $T$ 轮中阻塞连续 $(1-\epsilon)\*T$ 轮。我们分别测试了在敌手具有不同攻击能力 $\epsilon = 0.1$ 到 $\epsilon = 0.5$ 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：

<font color = red>测试的结果绘图</font>

从上图可知，随着 $\epsilon$ 的增加共识时延降低，这是因为随着 $\epsilon$ 的增加，敌手发起攻击的频率会降低。当$\epsilon$ 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$\epsilon$ 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恒昊的抵抗连续阻塞攻击。

# Conclusion and Future Research

在本文中，我们提出一个适用于无线网络的区块链协议——稳定共识协议。这个协议可以建立在可信的门限签名机制和安全高效的出块节点选举机制上。此外，我们的协议可以有效的抵抗双花攻击、女巫攻击和阻塞攻击。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，并且探索更加安全高效的抵抗区块链协议面临的各种攻击。

# References:

[1] J. Xu, S. Wang, A. Zhou and F. Yang, "Edgence: A blockchain-enabled edge-computing platform for intelligent IoT-based dApps," in China Communications, vol. 17, no. 4, pp. 78-87, April 2020, doi: 10.23919/JCC.2020.04.008.

[2] T. Maksymyuk, J. Gazda, L. Han and M. Jo, "Blockchain-Based Intelligent Network Management for 5G and Beyond," 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), 2019, pp. 36-39, doi: 10.1109/AIACT.2019.8847762.

[3] R. Jabbar, N. Fetais, M. Kharbeche, M. Krichen, K. Barkaoui and M. Shinoy, "Blockchain for the Internet of Vehicles: How to Use Blockchain to Secure Vehicle-to-Everything (V2X) Communication and Payment?," in IEEE Sensors Journal, vol. 21, no. 14, pp. 15807-15823, 15 July15, 2021, doi: 10.1109/JSEN.2021.3062219.

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

[5] 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.

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

[7] Z. Jiao, B. Zhang, L. Zhang, M. Liu, W. Gong and C. Li. A Blockchain-Based Computing Architecture for Mobile Ad Hoc Cloud, in IEEE Network, vol. 34, no. 4, pp. 140-149, July/August 2020.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, “A survey of distributed consensus protocols for blockchain networks,” IEEE Commun.Surv. Tutorials, vol. 22, no. 2, pp. 1432–1465, 2020.

[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] 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.

[21] L.Luu, V. Narayanan, C. Zheng, et al. A secure sharding protocol for open Blockchains[C]. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 17–30.

[22] E. Kokoris- Kogias, P. Jovanovic, L. Gasser, et al. OmniLedger: A secure, scale-out, decentralized ledger via sharding[C]. In: Proceedings of 2018 IEEE Symposium on Security and Privacy (SP 2018). IEEE, 2018: 583–598.

[23] M. Zamani, M. Movahedi, M. Raykova. RapidChain: Scaling Blockchain via full sharding[C]. In: Proceed ings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS 2018). Toronto, ON, Canada, October 15–19, 2018: 931–948.

[24] S. Yu, L. Kun, S. Zhou, Y. Guo, J. Zhou and B. Zhang, “A High Performance Blockchain Platform for Intelligent Devices,” In Proc.IEEE International Conference on Hot Information-Centric Networking (HotICN’18), pp.260-261, 2018.

[25] P. Kumar, M. Chen and J. Park, “A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT,” IEEE Access, vol.6, pp.115-124, 2018.

[26] O. Novo, “Blockchain meets iot: An architecture for scalable access management in iot,” IEEE Internet of Things Journal, vol. 5, no. 2,

pp. 1184–1195, 2018.

[27] T. Hardjono, A. Pentland, “Verifiable Anonymous Identities and Access Control in Permissioned Blockchains,” pp. 9, 2016.

[28] M. Conoscenti, A. Vetro and J. Martin, “Peer to Peer for Privacy

and Decentralization in the Internet of Things,” In Proc. IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C’17), pp.288-290, 2017.

[29] X. Zhu and Y. Badr, “Identity Management Systems for the Internet of Things: A Survey Towards Blockchain Solutions,” Sensors,

vol.18, no.12, pp.4215-4215, 2018.

[30] H. Halpin, “NEXTLEAP: Decentralizing Identity with Privacy for Secure Messaging,” In Proc.International Conference on Availability,Reliability and Security (ARES’17), 2017.

[31]M. Zheng, M. Goldenbaum, S. Stańczak and H. Yu, "Fast average consensus in clustered wireless sensor networks by superposition gossiping", Proc. IEEE Wireless Commun. Netw. Conf., pp. 1982-1987, 2012.

[32] 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.

[33] F. Molinari, S. Stańczak and J. Raisch, "Exploiting the superposition property of wireless communication for average consensus problems in multi-agent systems", Proc. Eur. Control Conf., pp. 1766-1772, 2018.

[34] 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.

[35] G. Scutari and S. Barbarossa, “Distributed consensus over wireless sensor networks affected by multipath fading,” IEEE Transactions on Signal Processing, vol. 56, no. 8, pp. 4100–4106, 2008.

[36] T. C. Aysal, A. D. Sarwate, and A. G. Dimakis, “Reaching consensus in wireless networks with probabilistic broadcast,” in 2009 47th Annual Allerton Conference on Communication, Control, and

Computing (Allerton). IEEE, 2009, pp. 732–739.

[37] 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.

[38] 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.

[39] 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.

[40] A. Boldyreva. "Threshold signatures, multisignatures 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.