RBFT：基于 Raft 集群的拜占庭容错共识机制

黄冬艳，李浪，陈斌，王波

（桂林电子科技大学广西无线宽带通信与信号处理重点实验室，广西 桂林 541004）

摘 要：针对现有联盟链共识机制因可拓展性不足，无法在支持大规模网络的同时满足低时延、高吞吐量和安全性的问题，采用网络分片的思想，提出一种适用于联盟链的带有监督节点的两级共识机制——RBFT。首先对网络节点进行分组，组内采用改进的 Raft 机制进行共识，然后由每个组内选出的领导者组成网络委员会，网络委员会内部采用 PBFT 机制进行共识。研究结果表明，在大规模网络环境下，相比 PBFT 和 Raft，RBFT 在具备拜占庭容错能力的同时可以保证高共识效率，因而具有更高的扩展性。

关键词：联盟链；共识机制；网络分片；实用拜占庭容错；Raft

中图分类号：TP393

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021043

RBFT: a new Byzantine fault-tolerant consensus mechanism based on Raft cluster

HUANG Dongyan, LI Lang, CHEN Bin, WANG Bo

Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541004, China

Abstract: The existing consensus mechanisms of consortium blockchain are not scalable enough to provide low latency, high throughput and security while supporting large-scale network. A new consensus mechanism called RBFT was proposed to improve scalability, which was a two-level consensus mechanism with supervised nodes based on the idea of network fragmentation. In RBFT, the nodes were firstly divided into several groups. Each group adopted the improved Raft mechanism to reach consensus and select leader. Then, the leaders of each group formed the network committee, and the network committee adopted PBFT mechanism for consensus. Comparative experiments verify that RBFT can tolerant Byzantine fault while ensuring high consensus efficiency in large-scale network compared with PBFT and Raft.

Keywords: consortium blockchain, consensus mechanism, network fragmentation, PBFT, Raft

1 引言

区块链起源于比特币[1]，其实质是一个基于对等网络（P2P, peer to peer）和密码学的分布式数据库，具有去中心化、不可篡改和公开透明等特点。应用区块链可解决信息不对称问题，实现多个主体之间的协作信任与一致行动。

区块链的核心技术包括共识机制、分布式存储技术、密码学和智能合约[2]。其中，共识机制着重解决分布式系统的一致性问题，旨在保证所有节点维护的数据副本的一致性，避免数据不统一和信息不对称问题的发生。不同的场景对共识机制的扩展性、共识效率和隐私性有不同的需求[3-4]。现有的共识机制包括工作量证明（PoW, proof of work）[1]、权益证明（PoS, proof of stake）[5]、委托权益证明（DPoS, delegated proof of stake）[6]

收稿日期：2020−08−21；**修回日期**：2020−12−31

通信作者：王波，glbluewind@126.com

基金项目：广西科技基地和人才专项基金资助项目（桂科 No.AD19110042）；广西重点研发计划基金资助项目（桂科 No.AB20238026）

Foundation Items: Guangxi Science and Technology Base and Talent Special Project (No.AD19110042), Guangxi Key Research and Development Program (No.AB20238026)

等概率一致性共识机制，实用拜占庭容错（PBFT, practical Byzantine fault tolerance）[7]、Paxos[8]、Raft[9]等绝对一致性共识机制，以及多种混合共识机制，如 DPoS+PBFT 混合的 Tendermint [10]、迅雷链的 DPoA（DPoA, delegated proof of ability）+PBFT 的同构多链共识机制[11]等。

从去中心化的角度，可以将区块链分为公有链、私有链和联盟链。公有链的去中心化程度最高，任何节点都可加入，如比特币[1]、以太坊[5]等。私有链的去中心化程度最低，一般由个人或某个小团体创建，仅使用区块链技术进行记账。联盟链的去中心化程度介于公有链和私有链之间，一般由多个不同利益方的机构组成，节点必须经过联盟链节点成员管理服务进行身份确认和鉴权，获得准入证才可以加入联盟链网络[12]。联盟链支持节点管理、可插拔的框架和模块化的共识机制，因此在性能和隐私保护上比公有链更有优势，是区块链落地的主要趋势。

联盟链在供应链管理[13]、产品溯源[14]、智能制造[15]等领域有着广阔的应用前景。但要将联盟链真正落地，迫切需要解决其扩展性差、安全性低和吞吐量小的问题。在节点数量上，实际商业应用的节点规模在几十到数百不等；在吞吐量上，诸如去中心化电商、车联网等领域要求节点规模在万级以上；在安全性方面，区块链金融、数字资产等金融领域对安全性要求相对较高。同时，为了保证用户体验，绝大多数的交易需要更低的时延。这些典型应用场景的要求远超现有区块链网络在时延、吞吐量和拓展性等方面所能达到的性能。共识机制是区块链的核心技术，决定了区块链的吞吐量、安全性和拓展性等性能。因此，改进共识机制是解决这些问题的有效手段。

联盟链的共识机制需要兼顾高效、安全与可拓展性，尤其是在大规模网络环境下仍需保持高吞吐量和低时延。目前，已有的共识机制尚不能同时满足这些需求，例如，公有链项目中常用的共识机制 PoW 效率太低，PoS/DPoS 以及类似算法瑞波协议[16]、Algorand 共识协议[17]等易导致“富豪统治”，且这些共识机制均依赖代币，而很多联盟链的实际应用场景并不需要发行代币。

Hyperledger Fabric 是业界最先发展联盟链的项目，其 0.6preview 版本采用了 PBFT 共识机制，1.0 版本使用了效率更高的 Kafka 集群共识机制[18]，最新的 Fabric 2.0 使用了更灵活的 Raft 共识机制，通常实测吞吐量在 300~500 笔/秒（PBFT 数据，采用 Raft 后有提升）。FISCO BCOS 联盟链项目支持并行计算的 PBFT 和标准 Raft 这 2 种模式，官方实测单链吞吐量可达 1 000 笔/秒以上。Coco 联盟链项目支持 Paxos 共识协议，官方吞吐量是 1 600 笔/秒。这些联盟链的吞吐量虽然远高于比特币、以太坊，但仍然无法满足现有联盟链实际应用的需求。

在共识机制的改进方面，文献[19]提出了一种基于 K-medoids 的改进 PBFT 共识机制，通过对大规模网络节点的特征进行聚类和层次划分，在 1 000 个节点的模拟中，单次共识耗时缩短了 20%；文献[20]将 PBFT 的三阶段共识简化为两阶段共识，从而减少了通信开销，提高了共识效率；文献[21]提出一种基于特征信任模型的优化 PBFT，提高了拜占庭容错能力；文献[22]通过建立信誉模型，根据信誉值赋予节点不同的话语权并在 PBFT 机制中加入预提交阶段来减少通信次数，提高了算法性能并可以有效防御女巫攻击；文献[23]基于 Kademlia 协议优化了 Raft 的领导者选举和共识机制，进一步提高了算法的领导者选举速度和交易吞吐量。对 PBFT 的改进并不能从根本上解决 PBFT 的算法瓶颈，节点间两两通信机制导致其扩展性低，不适合大规模网络。Raft 属于强领导者型共识机制，即使在节点规模扩大的情况下仍能保持算法的高共识效率，但也因此不具备拜占庭容错能力，仅适用于可信执行环境的联盟链。

本文的主要研究工作如下。

1) 针对联盟链共识机制 PBFT 在节点数量增多时共识效率下降的问题，结合 Raft 共识思想提出了一种基于 Raft 集群的拜占庭容错共识机制——RBFT（Raft cluster Byzantine fault tolerance）。通过将网络节点分组，组内采用改进的 Raft 共识机制，由 Raft 共识机制选举出的领导者组成委员会，委员会采用 PBFT 共识机制，大大降低了节点通信量，有效地提高了共识效率。同时，引入监督节点使 Raft 具有了拜占庭恶意行为容忍能力，提高了组内共识的安全性。

2) 讨论了 RBFT 的节点分组策略和监督节点的监督策略，论证了 RBFT 共识机制的一致性、安全性和灵活性。

3) 对 RBFT 的通信开销、共识时延、吞吐量和容错性进行了仿真与实验测试，以验证其有效性和可靠性。

2　背景知识

2.1　PBFT 共识机制简介

PBFT 机制由 Castro 等[7]提出，目的是改进 BFT 的算法效率，构建容忍拜占庭故障的高可用系统。它将算法的复杂度由指数级降低到多项式级，使算法能够处理大量事务并且保证一定的效率。PBFT 共识机制包含 2 个重要的流程：三阶段共识流程和视图更换流程。

三阶段共识流程用于对候选区块形成共识，分为预准备、准备和提交 3 个阶段。预准备阶段消息由主节点广播给副本节点，副本节点收到预准备消息后会广播准备消息。当某一节点收到不少于 $2f$ 条来自不同节点的准备消息时，该节点进入提交阶段并广播提交消息。同样，当节点收到的提交消息数不少于 $2f+1$ 时，该节点三阶段共识完成，将区块记入本地账本。因此，PBFT 的通信复杂度为 $O(N^2)$，在不超过 $1/3$ 的节点为拜占庭节点时仍能保持一致，即 $f \leqslant (N-1)/3$，其中，N 为网络总节点数。

同时，PBFT 通过视图更换流程替换失效的主节点以保证共识服务的持续进行，给算法提供活性。当副本节点认为当前主节点无法在给定时间内完成共识或收到 $f+1$ 个视图切换消息时，进入视图更换，切换主节点进入下一个视图，开始新一轮的三阶段共识流程。

2.2　Raft 共识机制简介

Raft 机制是由 Ongaro 等[9]提出的一种分布式共识机制，目标是实现账本复制的一致性。Raft 将一致性算法分解成 2 个关键的模块：领导者选举和日志复制。

在 Raft 中，节点有 3 种可能的状态：跟随者、候选者和领导者。在任一时刻，每个节点都处于这 3 种状态中的一种。Raft 机制将时间划分为若干个任意长度的时间段，每个时间段称之为一个任期。任期的编号是连续递增的整数。每个任期开始于一次领导者选举。所有节点的初始状态都是跟随者，如果跟随者没能感知到领导者的存在，那么跟随者就跳转为候选者。候选者将会发起投票，要求其他节点选择自己为领导者。其他节点将会回应候选者的请求。如果候选者获得多数节点赞同票，那么它就成为领导者。这个过程称为领导者选举。正常情况下，网络中仅有一个领导者，其余节点均为跟随者。领导者通过周期性地向跟随者发送心跳包以维持其领导地位。所有经过验证的交易在该任期内都由领导者打包生成区块。在收到超过半数节点回复后，领导者发送确认信息给跟随者，并将该区块作为账本上的下一个区块。跟随者收到领导者的确认信息后，将该区块作为本地账本上的下一个区块。这个过程称为账本复制。

Raft 是一种强领导者算法，领导者选举是该共识协议的重要组成部分。在基于 Raft 的区块链系统中，数据仅能从领导者到其他服务器单向流动，因而架构简单且易于理解。Raft 具有线性的复杂度，因而共识效率很高。在容错性方面，Raft 可以容忍不超过 50%的停机故障，但不具备拜占庭容错能力。

3　RBFT 共识机制

为了提供一种具备高扩展性、高安全性、低时延高吞吐量的共识算法，本文结合 PBFT 的拜占庭容错安全性与 Raft 高共识效率的优点，提出一种基于 Raft 集群的拜占庭容错共识机制——RBFT。RBFT 共识机制流程如图 1 所示。

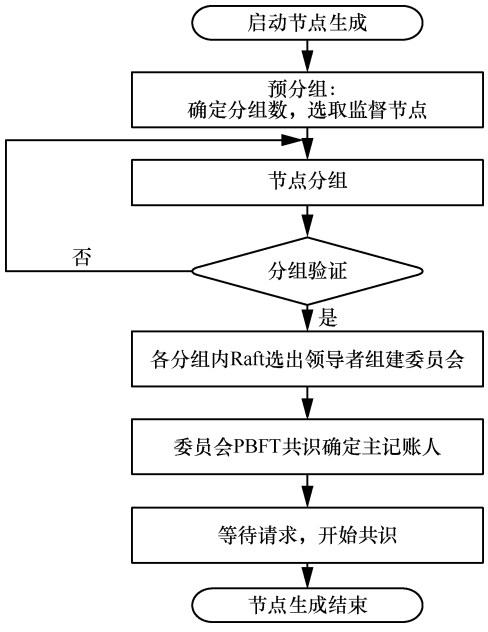


图 1　RBFT 共识机制流程

RBFT 共识机制采用分片的思想将区块链网络节点进行分组，各分组内采用 Raft 机制共识并选出领导者组建委员会，委员会内部采用 PBFT 机制进行共识。RBFT 同时引入监督节点，解决 Raft 共识不能对抗拜占庭恶意行为的问题，从而保证共识机

制的安全性。

3.1 节点的分组策略

所提共识机制需要进行网络分片，根据 RBFT 算法的两级共识机制，PBFT 共识需要满足分组数 $k \geqslant 4$，每个组内的 Raft 共识需要满足节点数 $m \geqslant 3$。此外，监督节点需要同时分配到多个组内以实现监督功能。现有的网络分片包括基于协议分片[24]和基于地理位置分片[25]。

文献[24]提出了一种基于 ELASTICO 协议的网络分片机制。节点根据自身 IP 地址、公钥以及一个随机数进行 Hash 运算，将所得的 Hash 值作为自己所在网络分片在区块链系统中的身份，Hash 值的后若干位代表了节点所在的分片 ID。区块链系统中的交易会并发地分配到各个分片中并行处理。

文献[25]提出了一种区块链系统，该系统使用基于地理位置的分片机制。根据节点所处的地理位置范围，将范围内的节点划分到一个分组内。

基于 ELASTICO 协议的分片机制中，Hash 的运算结果的随机性较大，容易导致分片不均；基于地理位置的分片机制中，人为的干预性较大，但是这 2 种分片机制都不能实现监督节点的分配。本文引入一致性 Hash 算法[26]的设计思想，利用一致性 Hash 算法平衡、分散、单调的特点，提出一种适用于 RBFT 的节点分组算法，如算法 1 所示。

算法 1 RBFT 节点分组算法

1) InitHashRing (*k*) //根据分组数 *k* 初始化 Hash 环，确定每个分组 ID 的位置

2) 为监督节点分组

for $i = 0{:}1{:}r$

计算 Hash(nodeID + ip + randomStr)，根据结果将节点映射到 Hash 环上

clockwiseSearch ( ) //根据监督节点位置顺时针查找分组 ID，将监督节点映射到对应的不同分组中

end for

3) 为普通节点分组

计算 Hash(nodeID + ip + randomStr)，根据结果映射到环上

clockwiseSearch ( )

4) 合理性验证

根据组内节点数 $m \geqslant 3$ 及每个组内至少含一个监督节点的条件，确定分组是否合理，若是，转至步骤 5)；若否，转至步骤 2)

5) 分组完毕

一致性 Hash 计算的结果是一个 uint32 类型的大整数，结果同样具有 Hash 的抗碰撞性，根据结果可将节点映射的 Hash 值分布在 $0 \sim 2^{32}$ 的圆上。一致性 Hash 的平衡分散性能够保证计算的结果尽可能地呈均匀分布，并使每个组内的领导者负载均衡，解决分片不均的问题。此外，其单调性降低了节点的加入和退出对分组的影响。

采用一致性 Hash 算法的分组策略如图 2 所示。设每 *r* 组分配一个监督节点，监督节点的数量满足

$$s \geqslant \begin{cases} \dfrac{k}{r}, & k \bmod r = 0 \\ \dfrac{k}{r} + 1, & k \bmod r \neq 0 \end{cases} \tag{1}$$

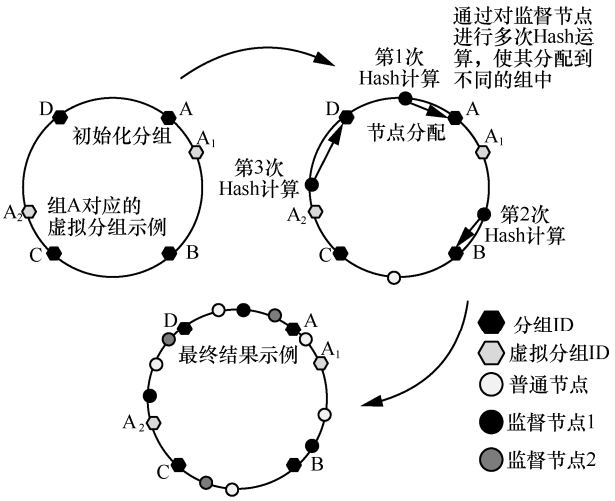
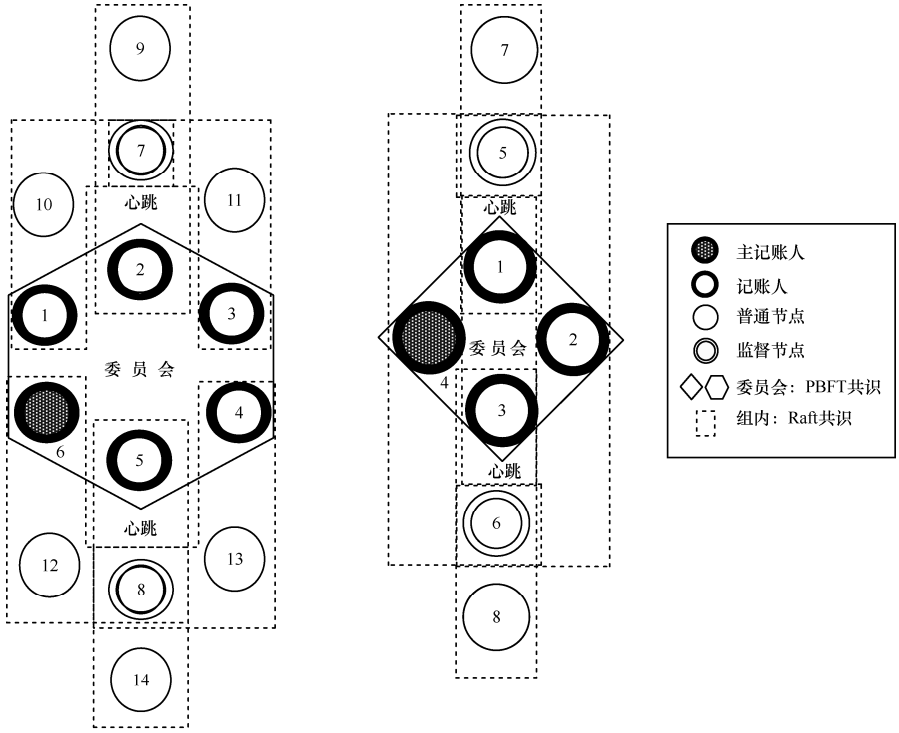


图 2 采用一致性 Hash 算法的分组策略

如图 2 所示，假设分组数为 4，每三组分配一个监督节点，则 $s \geqslant 2$。根据算法 1，按顺时针进行分配，监督节点 1 被同时分配到组 A、组 B 和组 D 中，监督节点 2 被同时分组到组 A、组 C 和组 D 中。

当分组数较少时，容易因为分片不够均匀而造成数据倾斜问题，即部分分片中承载了大多数节点，导致负载失衡。一致性 Hash 的理想结果是如果分组 ID 有 *K* 个，承载的节点的 Hash 值有 *N* 个，那么每个分组应该承载 N / K 个。RBFT 中的分组算法引入了一致性 Hash 算法虚拟节点的概念，对实际分组设置多个虚拟分组来扩大分组数，使分组更均匀。

根据算法 1 得到 RBFT 共识机制的分组结果，分别如图 3 和表 1 所示。



(a) 分组方式1　　(b) 分组方式2

图 3　区块链网络节点实体构成

表 1　　**分组结果**

分组方式	组号	组内节点	注
分组方式 1	①	1,7,10	每组只含一个监督节点
	②	2,7,9	
	③	3,7,11	
	④	4,8,13	
	⑤	5,8,14	
	⑥	6,8,12	
分组方式 2	①	1,5,7	部分组内含不止一个监督节点
	②	2,5,6	
	③	3,6,8,	
	④	4,5,6	

3.2　监督节点的监督策略

监督节点的存在是对 Raft 安全性的一大改进。监督节点的加入使 Raft 具备了抵抗拜占庭恶意节点的能力。

在 Raft 共识机制中，如果领导者作恶，向组内跟随者下发恶意信息，则会破坏共识的一致性。如图 3 所示，假设每三组分配一个监督节点，根据分组数不同，每个组内可能有一个或多个监督节点。监督节点负责监督组内的领导者，因此监督节点不参与领导者选举，只作为跟随者参与组内 Raft 共识。同时，监督节点需要保证匿名性以防领导者进行针对性的欺诈。

RBFT 在 Raft 共识时加入了签名验证环节。领导者下发日志同步消息给跟随者时，需要对消息进行签名。监督节点在收到不同领导者的日志消息后，需要对签名进行验证并比对内容，从而判断领导者是否为拜占庭恶意节点。

监督节点监督领导者的策略如图 4 所示，分为取证、举证和验证 3 个阶段。

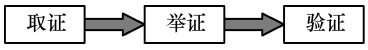


图 4　监督节点监督领导者的策略

1) 取证阶段。进行 Raft 组内共识时，监督节点可以同时监听 3 个分组内的领导者（记账人）下发至跟随者的记账信息，在验证节点签名后对领导者下发的信息进行甄别，当发现一个领导者下发的信息与其他领导者不同时，则可以判定其为恶意节点。根据节点签名锁定节点身份，进行取证。

2) 举证阶段。对恶意节点取证完成后，监督节点打包一个举证消息<Proof, *t*, *m*, *i*>，并将消息发送给成员管理服务。其中，Proof 为消息标识，*t* 为举

证时间戳，m 为举证内容，i 为举证监督节点的节点编号。

3) 验证阶段。成员管理服务根据举证内容，判断被举证的恶意节点的公钥等信息并验证举证的合法性，对是否罢免恶意节点的领导者身份做出判决。

3.3 RBFT 共识流程

主记账人将若干条收听到的客户端请求信息打包成一个区块后进行共识，共识流程如算法 2 和图 5 所示。

算法 2 RBFT 共识流程

1) PBFT-Stage: A client sends a request <Request, Client, M> to the primary//PBFT 阶段，客户端发送数据请求

2) The primary broadcasts <Pre-prepare, V, N, $D(M)$, i> to the backups//主节点广播预准备消息

3) Replica broadcasts <Prepare, V, N, $D(M)$, i>;//副节点广播准备消息

4) Replica counts the number of Prepare messages and denotes it as Count1, if Count1$\geqslant 2f$, broadcasts <Commit, V, N, $D(M)$, i>//副节点判断准备阶段是否完成

5) Replica counts Commit messages and denotes it as Count2, if Count2$\geqslant 2f+1$, enters Raft-Stage//副节点判断提交阶段是否完成，进入 Raft 共识阶段

6) Raft-Stage: The leader broadcasts <AppendLog, N, $D(M)$, i> to the followers//Raft 阶段领导者广播消息

7) Followers reply AppendLog Prepared messages to leader//跟随者节点接收消息并反馈

8) The leader counts the number of AppendLog Prepared messages received and denotes it as Count, if Count$\geqslant \frac{N}{2}$, commit Log//领导者节点根据消息反馈结果，判断是否达成共识并提交日志

9) Reply client//共识完成，回复客户端

算法 2 中，V 表示当前视图的编号，N 表示当前请求的编号，M 表示消息的内容，$D(M)$表示消息的摘要，i 表示节点的编号，f 表示拜占庭节点个数。

4 RBFT 的一致性、安全性与活性

根据 FLP（Fischer-Lynch-Patterson）[27]和 CAP（consistency-availability-partition tolerance）理论[28]，异步模型系统中不存在一个可以使系统取得强一致性的确定性共识机制。一个分布式系统最多只能同时满足一致性、可用性和分区容忍性三项中的两项。这里的一致性指强一致性，是系统成功操作返回后所有节点在同一时刻的数据完全一致。大多数情况下要遵循 BASE（basically available-soft state-eventual consistency）理论[29]弱化一致性，但要采取适当的方式达到最终一致性，即所有副本在经过一定时间后，最终能够达成一致的状态。

RBFT 共识机制基于 PBFT 和 Raft 构建，其一致性、安全性和活性继承了 PBFT 和 Raft 原有的优良特性，并通过监督机制和分组增强了安全性和活性。

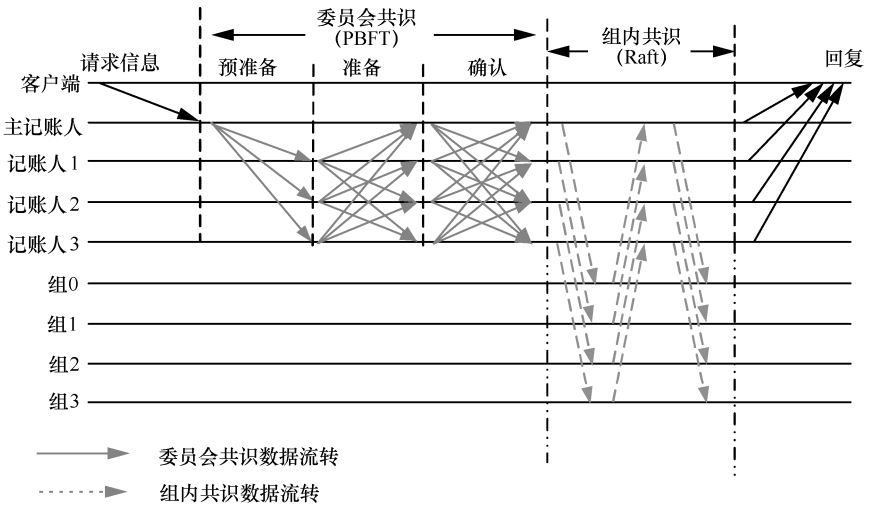


图 5 共识流程

4.1　一致性

RBFT 通过 PBFT 机制保证了共识一致性，通过 Raft 协议的 AppendEntry RPC 机制保障了最终一致性。在 RBFT 两级共识阶段，委员会共识遵从 PBFT 共识协议来保证算法的一致性，在容错范围 $f \leqslant (k-1)/3$ 内可以保证 PBFT 阶段正常共识。组内共识遵从 Raft 共识协议，当故障节点数 $f \leqslant (m-1)/2$ 时，Raft 共识阶段也可以顺利进行。

节点从故障到恢复后作为 Raft 的跟随者角色存在，从领导者处更新自己的日志列表来保持一致。如图 6 所示，S_1 为一个 Raft 集群的领导者，节点拥有与其他集群一致的日志，S_1 和 S_3 拥有完备的日志列表，S_2 和 S_5 可能处在日志的复制阶段，S_4 可能是宕机节点。当宕机节点从宕机状态恢复后，会从领导者处进行日志同步，从而保证最终所有节点的日志的一致性。

4.2　安全性

RBFT 共识的第一阶段通过 PBFT 的共识特性[7]保证了安全。预准备、准备和提交的三阶段消息必须在同一个视图内完成，且完成的时间必须小于节点需要通过超时触发视图切换的时间，所有的消息均遵从 PBFT 共识协议的摘要、序列号和签名验证等步骤。

RBFT 共识的第二阶段的安全性由监督节点的监督功能来保证。根据 3.2 节监督节点的监督策略可知，监督节点的监督行为使组内共识具备了对抗拜占庭恶意行为的能力，增强了 RBFT 共识第二阶段的安全性。Raft 组内普通节点需要收到领导者提交的消息且收到成员管理服务对消息结果验证为真时，才可提交日志信息上链；若为假（监督节点发现有领导者作恶，领导者身份被罢免），则需等待下一任期同步新领导者的日志，防止恶意信息上链保证安全性。

对于监督节点本身，RBFT 设计了 3 种机制保证安全性。① 增加分组内监督节点的占比。对于网络环境较差或者安全可靠性低的场景，需要适当增加分组内监督节点的占比来提高安全性。当组内部分监督节点受攻击或者宕机时，可以由其他监督节点执行监督职责。② 监督节点重选。监督节点定期向成员管理服务发送监管状态信息<SpyHeartBeat,*t*,*m*,*i*>作为存活证明。如果未收到监管状态信息或者举证信息<Proof,*t*,*m*,*i*>验证不通过，则说明监督节点发生故障，触发监督节点重选。③ 监督节点轮换。为了保证监督节点的匿名性，防止记账节点进行针对性的攻击与共谋，每进行 n 轮共识后对监督节点进行洗牌，再进行重新分组以增强安全性。

4.3　活性

RBFT 算法的活性由 PBFT 阶段的视图切换和 Raft 阶段的领导者选举等提供。在 PBFT 阶段，记账节点通过 $p = v \bmod k$ 来确定主节点，视图编号 v 从 0 开始递增。当副本记账节点的计时器超时或者收到一组 $f+1$ 个有效视图切换消息时，意味着当前视图的主节点已无法完成共识，副本必须进入下一个视图重新进行共识服务。

此外，在 RBFT 算法中，委员会成员是各分组的领导者通过 Raft 的领导者选举产生的。当委员会成员宕机时，其对应分组内的跟随者节点接收不到领导者的心跳检测而触发领导者重选流程，由新领导者替换失效的委员会成员。因此，RBFT 算法增强了委员会 PBFT 共识阶段的节点宕机故障恢复能力，增强了算法的活性。

Raft 的领导者选举和心跳检测机制使副本节点在接收日志超时或者当前任期领导者宕机时开启

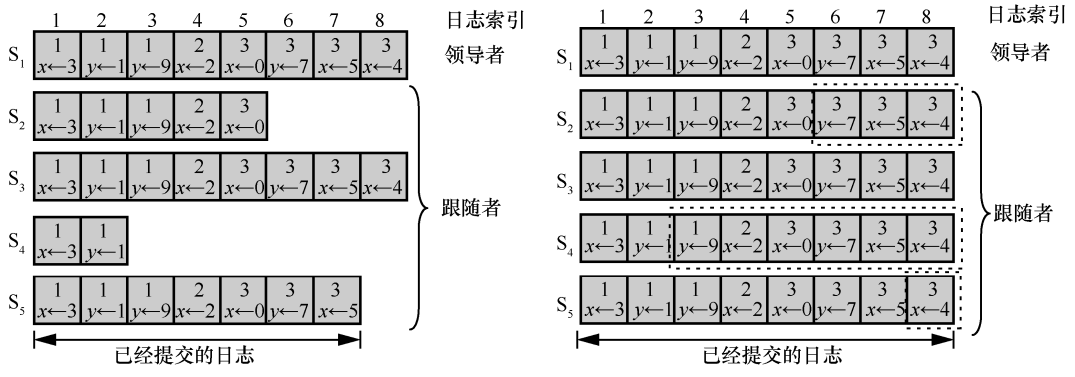


图 6　Raft 共识日志状态的一致性

一个新的任期并重新选举领导者对外提供服务，维持了 RBFT 算法组内共识的活性。

5 仿真与实验分析

本节将从通信开销、共识时延、吞吐量和容错性 4 个方面对比 RBFT 与其他主流算法的表现，以此论证 RBFT 的有效性和可靠性。

5.1 仿真方案与实验设置

测试平台硬件采用了 8 核/48 GB 的 Intel(R)Core(TM) i7-9700 的服务主机，通过配置多台虚拟机加端口映射的方式来模拟多节点环境，映射方式如表 2 所示。测试平台软件的开发环境为 LinuxMint19.3/Golang1.14。

表 2 映射方式

节点编号	映射 IP 地址:端口
0	192.168.1.101:6001
1	192.168.1.101:6002
2	192.168.1.101:6003
⋮	⋮
10	192.168.1.102:6001
11	192.168.1.102:6002
12	192.168.1.102:6003
⋮	⋮

1) 通信开销

PBFT 共识机制需要两两节点进行通信，通信量为 $O(N^2)$（其中 N 为节点数量）。Raft 共识机制的通信量为 $O(N)$，通过网络分片，RBFT 相比于 PBFT 共识机制，通信量由 $O(N^2)$下降到 $O(N/k)+O(k^2)$。

仿真设定一次共识过程中的消息内容 m=256 bit，消息摘要 d=128 bit，回复客户端的信息大小 r=80 bit，Raft 共识过程的心跳包大小 h = 64 bit。

2) 共识时延

算法的共识时延定义为从客户端发起请求到该请求被确认并上链所需的时间。取连续测量 30 次结果的平均值作为算法的共识时延，分别记录相同网络规模下所提算法 RBFT 和对比算法 PBFT、Raft 的数据。

3) 吞吐量（TPS, transaction per second）

系统的吞吐量定义为系统每秒钟处理的事务量。在区块链系统中，吞吐量表现为交易数量 M 与处理对应交易时间 t 的比值，即

$$\mathrm{TPS}=\frac{M}{t} \tag{2}$$

每个节点都可以监听来自客户端的请求，因此需要为每个节点绑定一个客户端程序。客户端每隔 50 ms 发起一次请求，每秒并发度为 $20N$。所有的请求最后由主节点收集打包负责出块，采用 BoltDB 数据库[30]实现对区块数据的持久化存储，来记录每个块包含的交易量 M 及块共识的时间 t。出块间隔设定为 10 s，取系统运行 5 min 后数据库中稳定的 10 组数据的平均值作为该次测试的系统吞吐量。

4) 容错性

令 f_1、f_2 为大于 0 的整数，Raft 组内的节点数 $m \geqslant 2f_1+1$，分组数 $k \geqslant 3f_2+1$。每 r 组分配一个监督节点（$r \geqslant 2$），假设每组的节点数目相同，则总的节点数 N 满足

$$N=km-\left[\frac{k}{r}(r-1)\right]-[l(r-1)] \tag{3}$$

其中，

$$l=\begin{cases}0, k \bmod r=0\\ 1, k \bmod r \neq 0\end{cases} \tag{4}$$

PBFT 共识阶段的最大容错为 $(k-1)/3$，Raft 阶段的最大容错为 $(m-1)/2$。在监督节点的参与下，RBFT 的最大容错为

$$\begin{aligned}F \leqslant & \frac{k-1}{3}m+\left(r-\frac{k-1}{3}\right)\left(\frac{m-1}{2}-1\right)+\\ & \left[\left(k-\frac{k-1}{3}\right)-\left(r-\frac{k-1}{3}\right)\right]\left(\frac{m-1}{2}\right)=\\ & 4f_1f_2+2f_2+f_1-r\end{aligned} \tag{5}$$

仿真中设定每三组分配一个监督节点（即 $r=3$），并假设所有分组中包含的节点数目一致。

5.2 通信开销

图 7 为 RBFT 共识机制与经典 PBFT 共识机制的通信开销比较。从图 7 可以看出，RBFT 共识机制的通信开销远小于区块链网络全网采用 PBFT 共识机制的通信开销，如当网络节点数量为 117 时，经典 PBFT 共识机制的通信开销为 3.51×10^6 bit，RBFT 共识机制的通信开销为 4.06×10^4 bit，通信开销降低了 98.8%，且随着网络节点的增多，RBFT 比 PBFT 节省的通信开销越大。

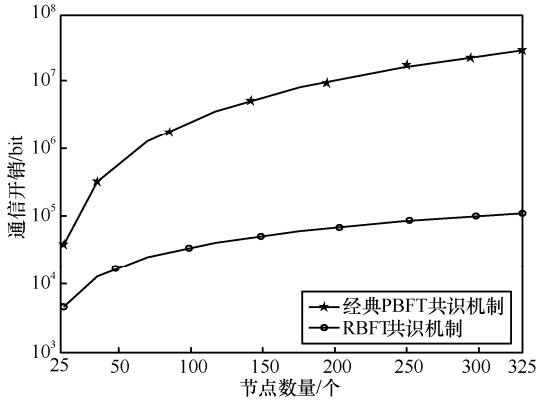


图 7　RBFT 与经典 PBFT 共识机制的通信开销比较

5.3　共识时延

所提算法和对比算法的共识时延实测结果如图 8 所示。从图 8 可以看出，随着节点数量的增加，共识时延逐渐增大。其中，PBFT 算法增长最快，Raft 最慢。对于 RBFT 而言，同样是增加网络节点数，保持组内节点数不变而增加分组数相比保持分组数不变而增加组内节点数，会引起更高的共识时延，如图 8 中 RBFT（组内节点数：10）和 RBFT（分组数：4）所示。这是因为增加分组数实质上是扩大了委员会的规模，增加了 PBFT 阶段的共识耗时，所以增加分组数对共识时延的影响更大。但在同样节点规模下，RBFT 的时延仍远小于 PBFT。从图 8 还可以看出，PBFT 的时延随节点规模的增大急剧增加，而 RBFT 的时延增速则较缓。因此，RBFT 在节点规模扩大时仍能保证高共识效率。

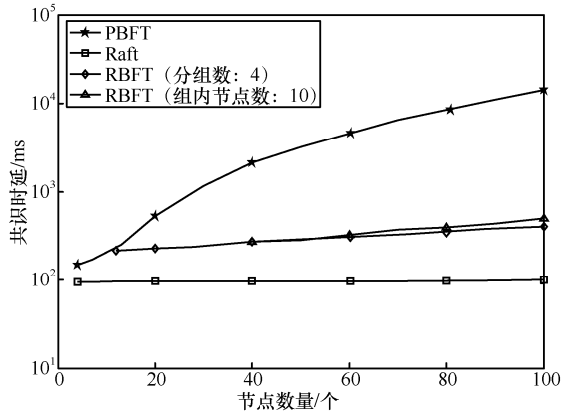


图 8　所提算法和对比算法的共识时延实测结果

5.4　吞吐量

RBFT 与 PBFT 的吞吐量测试结果如图 9 所示。共识效率是影响 TPS 的主要因素，共识效率越高，交易处理能力越强。此外，节点并发度是影响 TPS 的另一因素，节点并发度越高，交易量越大。但区块体积增大，对带宽的要求更高（实验时采用虚拟机模拟环境，并未达到带宽上限，可等效为无穷大）。影响 TPS 的因素还包括各节点对并发数据的处理能力、对数据库的 I/O 读写能力等。

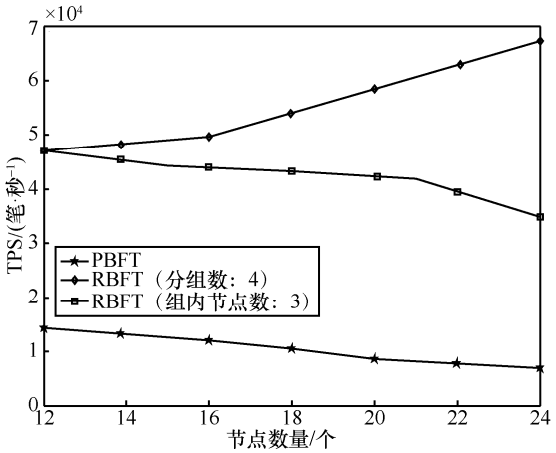


图 9　RBFT 与 PBFT 的吞吐量测试结果

从图 9 可以看出，随着节点数量的增加，节点并发度增加，但由于 PBFT 在节点增加时共识效率降低，交易处理耗时更长，吞吐量降低。对于 RBFT，节点被分为四组，当增加组内节点数时，由 5.3 节可知，算法共识效率影响较小，节点并发度是影响 TPS 的主要因素。随着节点数量的增大，TPS 逐渐增大。当固定组内节点数增加分组数时，共识效率成为影响 TPS 的主要因素。随着分组数的增加，TPS 逐渐下降。但在同等网络规模情况下，RBFT 的 TPS 约为经典 PBFT 的 300%~400%。因此，RBFT 更适用于对 TPS 要求更高的联盟链应用场景。

5.5　容错性

监督节点的数量不会影响 RBFT 的共识效率与 TPS，这是因为监督节点本身位于多个组内，在参与共识过程中的通信是独立的，监督节点本身的通信强度增加，但是共识算法整体的通信强度不变。在容错性上，一个监督节点出错相当于包含该监督节点的分组都有一个节点出错，RBFT 算法的最大容错由式(5)给出。RBFT 共识机制与经典 PBFT 共识机制、Raft 共识机制的最大容错性能对比如图 10 所示。

由图 10 可以看出，RBFT 具有比 PBFT 和 Raft 更高的容错性。如当网络被分为四组，每三组分配一个监督节点时（相当于图 3(b)分组方式 2），则 N=8。单独采用 PBFT 的最大容错为 2，采用

Raft 的最大容错为 3，而采用 RBFT 的最大容错为 4（相当于三组共识，每组 2 个节点，三组共用一个监督节点的极限情况）。随着网络节点规模的扩大，RBFT 容错性增加。因此，RBFT 具有高安全性。

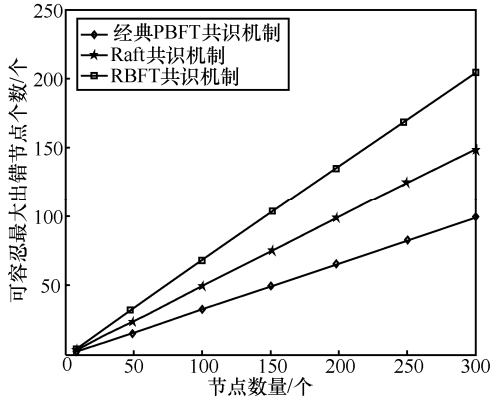


图 10 RBFT 共识机制与经典 PBFT 共识机制、Raft 共识机制的最大容错性能对比

6 结束语

本文提出了一种基于 Raft 集群的拜占庭容错共识机制——RBFT。所提共识机制很好地融合了 Raft 共识效率高和 PBFT 拜占庭容错性好的特点。测试结果表明，该共识机制具有通信开销小、时延低、吞吐量高、容错性强、可扩展性高的优势，适用于大规模节点网络，能有效突破制约当前联盟链落地的性能瓶颈，有助于推动联盟链的发展。未来研究可以结合跨链技术，进一步解决联盟链单链的扩展性低、隐私隔离性差的问题。

参考文献：

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R]. (2008-05-19) [2020-08-19].

[2] 张小军, 曹朝, 胡瑞丰, 等. 华为区块链白皮书[R]. (2018-04-18) [2020-08-19].
ZHANG X J, CAO C, HU R F, et al. Huawei blockchain white paper[R]. (2018-04-18)[2020-08-19].

[3] BREWER E A. Towards robust distributed systems[C]//Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing, New York: ACM Press, 2000: doi. org/10.1145/343477.343502.

[4] WANG W B, DINH T H, XIONG Z H, et al. A survey on consensus mechanisms and mining management in blockchain networks[J]. arXiv Preprint, arXiv:1805.02707, 2018.

[5] BUTERIN V. A next-generation smart con-tract and decentralized application platform[J]. Etherum, 2014(1): 1-36.

[6] LARIMER D. Delegated proof of stake consensus[R]. (2014-04-13) [2020-08-19].

[7] CASTRO M, LISKOV B. Practical Byzantine fault tolerance[C]//Proceedings of the Third Symposium on Operating Systems Design and Implementation. New York: ACM Press, 1999: 173-186.

[8] LAMPORT L. The part-time parliament[J]. ACM Transactions on Computer Systems, 1998, 16(2): 133-169.

[9] ONGARO D, OUSTERHOUT J. In search of an understandable consensus algorithm[C]//Proceedings of USENIX ATC'14: 2014 USENIX Annual Technical Conference. Berkeley: USENIX Association, 2014: 305-320.

[10] KWON J. Tendermint: consensus without mining[R]. (2020-02-26) [2020-08-19].

[11] Thunderchain: trusted blockchain expert[R]. (2017-10-22) [2020-08-19].

[12] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4): 481-494.
YUAN Y, WANG F Y. Blockchain: the state of the art and future trends[J]. Acta Automatica Sinica, 2016, 42(4): 481-494.

[13] 秦明. 区块链技术在供应链物流中的应用探讨[J]. 价格月刊, 2019(12): 64-69.
QIN M. A probe into the application of blockchain technology in the supply chain logistics[J]. Prices Monthly, 2019(12): 64-69.

[14] 王志锌, 柳平增, 宋成宝, 等. 基于区块链的农产品柔性可信溯源系统研究[J]. 计算机工程, 2020, 46(12): 313-320.
WANG Z H, LIU P Z, SONG C B. et al. Research and development of flexible and reliable traceability system for agricultural products based on blockchain[J]. Computer Engineering, 2020, 46(12): 313-320.

[15] 王赞强, 方新国. 基于区块链技术的智能制造的 P2P 协同设计[J]. 机械设计与研究, 2020, 36(2): 91-94.
WANG Z Q, FANG X G. P2P collaborative design of intelligent manufacturing based on blockchain technology[J]. Machine Design & Research, 2020, 36(2): 91-94.

[16] SCHWARTZ D, YOUNGS N, BRITTO A. The ripple protocol consensus algorithm[J]. Ripple Labs Inc White Paper, 2014, 5(8): 1-8.

[17] GILAD Y, HEMO R, MICALI S, et al. Algorand: scaling Byzantine agreements for cryptocurrencies[C]//Proceedings of the 26th Symposium on Operating Systems Principles. New York: ACM Press, 2017: 51-68.

[18] SUKHWANI H, WANG N, TRIVEDI K S, et al. Performance modeling of hyperledger fabric (permissioned blockchain network)[C]//2018 IEEE 17th International Symposium on Network Computing and Applications. Piscataway: IEEE Press, 2018: 1-8.

[19] 陈子豪, 李强. 基于 K-medoids 的改进 PBFT 共识机制[J]. 计算机科学, 2019, 46(12): 101-107.
CHEN Z H, LI Q. Improved PBFT consensus mechanism based on K-medoids[J]. Computer Science, 2019, 46(12): 101-107.

[20] 刘肖飞. 基于动态授权的拜占庭容错共识算法的区块链性能改进研究[D]. 杭州: 浙江大学, 2017.
LIU X F. Research on performance improvement of blockchain based on dynamic fault authorization of Byzantine tolerance consensus algorithm[D]. Hangzhou: Zhejiang University, 2017.

[21] GAO S, YU T, ZHU J, et al. T-PBFT: an eigen trust-based practical Byzantine fault tolerance consensus algorithm[J]. China Communica-

tions, 2019, 16(12): 111-123.

[22] 赖英旭, 薄尊旭, 刘静. 基于改进 PBFT 算法防御区块链中 sybil 攻击的研究[J]. 通信学报, 2020, 41(9): 104-117.
LAI Y X, BO Z X, LIU J. Research on sybil attack in defense blockchain based on improved PBFT algorithm[J]. Journal on Communications, 2020, 41(9): 104-117.

[23] WANG R, ZHANG L, XU Q, et al. K-Bucket based Raft-like consensus algorithm for Permissioned blockchain[C]//2019 IEEE 25th International Conference on Parallel and Distributed Systems. Piscataway: IEEE Press, 2019: 996-999.

[24] LUU L, NARAYANAN V, ZHENG C, et al. A secure sharding protocol for open blockchains[C]//Proceedings of the 2016 ACM SIGSAC Conference. New York: ACM Press, 2016:17-30.

[25] HYUNKYUNG Y, JONGCHOUL Y, SUNME K. The blockchain for domain based static sharding[C]//2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering. Piscataway: IEEE Press, 2018: 1689-1692.

[26] KARGER D, LEHMAN E, LEIGHTON T, et al. Consistent Hashing and random trees: distributed caching protocols for relieving hot spots on the world wide Web[C]//In Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing. New York: ACM Press, 1997: 654-663.

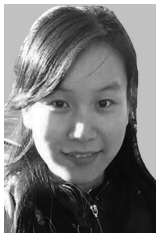
[27] FISCHER M J, LYNCH N A, PATERSON M S. Impossibility of distributed consensus with one faulty process[J]. Journal of the ACM, 1985, 32(2): 374-382.

[28] FOX A, BREWER E A. Harvest, yield, and scalable tolerant systems[C]//Proceedings of the Seventh Workshop on Hot Topics in Operating Systems. Piscataway: IEEE Press, 1999: 174-178.

[29] PRITCHETT D. BASE: an acid alternative[J]. Queue, 2008, 6(3): 48-55.

[30] BEN J. BoltDB: an embedded key/value database for Go[R]. (2014-11-28) [2020-08-19].

[作者简介]



黄冬艳（1984− ），女，广西南宁人，博士，桂林电子科技大学副教授、硕士生导师，主要研究方向为区块链技术、共识机制等。

李浪（1996− ），男，湖北黄冈人，桂林电子科技大学硕士生，主要研究方向为区块链技术、物联网等。

陈斌（1994− ），男，河南驻马店人，桂林电子科技大学硕士生，主要研究方向为区块链共识算法等。

王波（1977− ），男，陕西西安人，博士，桂林电子科技大学讲师、硕士生导师，主要研究方向为区块链、移动边缘计算、认知无线电等。