·热点与综述·　　文章编号：1000-3428(2023)03-0058-09　　文献标志码：A　　中图分类号：TP399

面向工业互联网的区块链分层分片研究

白首圳，陈美娟

（南京邮电大学 通信与信息工程学院，南京 210003）

摘　要：区块链的不可篡改、去中心化等特点能够有效解决工业互联网中日益突出的安全和隐私问题，然而当前主流区块链平台的吞吐量远不能满足工业互联网海量数据快速上链的需求，并且传统区块链采用的高冗余存储机制也无法适用于工业互联网场景。建立分层分片区块链架构，将区块数据分层存储在多个分布式云服务器和边缘服务器中，以应对工业互联网不断增长的数据量。依据边缘服务器之间的拓扑结构，设计一种基于复杂网络社团划分算法的改进区块链网络分片算法，在提升区块链网络吞吐量的同时缩短分片时间。将区块广播过程形式化为生成树论证了区块广播时间对吞吐量的影响，在此基础上提出一种基于生成树的片内主节点选取算法，进一步提升区块链网络的吞吐量。实验结果表明，与经典复杂网络社团划分算法相比，改进的区块链网络分片算法在对大规模网络进行分片时能够在不牺牲分片质量的前提下缩短约36%的分片时间，同时减少了各分片内区块的广播时间。

关键词：工业互联网；区块链；分片；社团划分；生成树

开放科学（资源服务）标志码（OSID）：

中文引用格式：白首圳，陈美娟. 面向工业互联网的区块链分层分片研究[J]. 计算机工程，2023，49(3)：58-66，79.

英文引用格式：BAI S Z，CHEN M J. Research on layering and sharding of blockchain for industrial Internet[J]. Computer Engineering，2023，49(3)：58-66，79.

Research on Layering and Sharding of Blockchain for Industrial Internet

BAI Shouzhen，CHEN Meijuan

（College of Communication and Information Engineering，
Nanjing University of Posts and Telecommunications，Nanjing 210003，China）

【Abstract】The blockchain technology can effectively solve the increasingly prominent security and privacy issues in the industrial Internet because its non-tampered and decentralized characteristics.However，the throughput of the current mainstream blockchain platform is far from satisfying the requirement for rapid on-chaining of massive data in industrial Internet，and the high redundancy storage mechanism adopted by the traditional blockchain can not be applied to industrial scenarios.A layering and sharding blockchain architecture is established.The block data is layered and stored in multiple distributed cloud servers and edge servers to cope with the increasing amount of data in the industrial Internet. Based on the topology between the edge servers，an improved blockchain network sharding algorithm based on complex network community division algorithm is designed to improve the throughput of the blockchain network and shorten the sharding time.The block broadcast process is formalized as a spanning tree，and the influence of block broadcast time on throughput is demonstrated. Based on this，a leader-node selection algorithm in the shard based on spanning tree is proposed to further improve the throughput of the blockchain network.The experimental results show that compared with the classical complex network community division algorithm，the improved blockchain network sharding algorithm can shorten the sharding time by approximately 36% without sacrificing the sharding quality and simultaneously reduce the broadcast time of blocks in each shard.

【Key words】industrial Internet；blockchain；sharding；community division；spanning tree

DOI：10. 19678/j. issn. 1000-3428. 0064338

基金项目：国家自然科学基金“基于区块链和机器学习的移动边缘云网络可信协作机制和资源优化方法”（61871237）；江苏省科技计划重点项目“新型边缘网络协同控制设备关键技术研发”（BE2020084-3）；江苏省重点研发计划“基于区块链的卫星与无线通信融合网络可信数据交换与共享关键技术研究与验证”（BE2019017）。

作者简介：白首圳（1997—），男，硕士研究生，主研方向为区块链技术；陈美娟，副教授、博士。

收稿日期：2022-03-30　　**修回日期：**2022-05-31　　**E-mail：**chenmj@njupt.edu.cn

0　概述

工业互联网作为重要的5G应用场景，将工业生产效率和生产力达到前所未有的水平[1]。目前，工业互联网已广泛应用于智能电网、电子商务、能源控制、高效物流等不同商业和工业领域[2]。然而，工业互联网系统采用的集中式架构会引发安全和隐私方面的问题，并且可能存在单点故障，无法提供稳定的服务[3]。随着连接到工业互联网中的设备数量不断增多，这些问题日益突出。

区块链是一个分布式的共享账本，一个不可逆转的公共数据库，它使不相关的参与者能够根据特定交易或事件的发生达成共识，而无需集中授权[4]。区块链的不可篡改、去中心化、可溯源等特点能够有效解决工业互联网中的安全和隐私问题[5]，但两者的结合仍面临诸多挑战，总结为以下两方面：一方面，工业互联网设备采集和生成的大量数据需要被安全、高效、实时存储，然而当前主流区块链平台的吞吐量（Transactions Per Second，TPS）远不能满足工业互联网中海量数据快速上链存储的需求[6]；另一方面，区块链在不同场景下的应用都因高冗余存储机制（每个节点存储一份完整的账本副本）增强了数据的公开性、透明性，确保了数据不被篡改，但却给区块链带来了巨大的存储压力，传统区块链采用的高冗余存储机制无法适用于工业互联网场景[7]。

分片技术是提升区块链吞吐量最直接有效的手段[8]。将分片技术应用于区块链就是将原始的区块链网络拆分为若干小规模的区块链网络，每个网络都由原始网络中的一部分节点构成，称为分片。在整个网络中的交易会被分配到不同分片进行并行处理，因此能够近似线性地提升区块链的吞吐量[9]。文献[10]提出一种基于公有链的分片方案ELASTICO。在ELASTICO的每个共识周期中，参与者都需要计算出工作量证明（Proof of Work，PoW）答案，该答案用于配置分片。各分片采用实用拜占庭容错（Practical Byzantine Fault Tolerance，PBFT）共识算法验证交易，共识结果将被提交到最终分片，最终分片负责对其他分片的共识结果产生最终决策，决策结果将被返回以更新其他分片。但是，ELASTICO需要在每轮共识后重新配置分片，且任一分片均须存储网络中其他所有分片的区块数据，这会造成计算和存储资源的浪费。为解决ELASTICO存在的问题，文献[11]提出一种分片协议OmniLedger，使用分布式随机数生成方案和可验证随机函数来配置分片，减少了分片过程的计算开销，但OmniLedger在处理跨分片交易时需要向全网广播，并且容错率与ELASTICO相同，仅为1/4。文献[12]提出一种将容错率提升至1/3的分片方案RapidChain。同时，为解决OmniLedger在处理跨分片交易时需要向全网广播的问题，RapidChain设计了分片间路由协议以快速验证跨分片交易，减少了通信开销。但是，RapidChain是基于网络同步的假设设计的，在异步网络中的性能还未经过验证。文献[13]提出一种横向扩展的分片协议Monoxide，通过设计一种特定的异步共识区域，使吞吐量能够随着共识区域数量的增多而线性增加，并且不会影响系统的去中心性。此外，Monoxide还设计了一种用于放大算力的PoW方案，使每个区域的有效算力与整个网络保持相同，从而保证每个分片的安全性。

目前，分片协议多数建立在公有链的基础上，有关联盟链的分片协议研究较少。由于公有链允许任何节点加入且区块数据完全公开，因此对公有链分片时，需要借助大量复杂的计算来增加节点作恶的成本，以提升网络的安全性。然而，联盟链中的节点均是通过证书颁发机构（Certificate Authority，CA）鉴权后加入到网络中的，它们通常只会因为宕机、网络延迟等原因而未能按预期参与共识过程[14]。得益于联盟链网络的封闭性，文献[15]提出一种无需通过复杂计算来确保网络安全性的联盟链分片协议MDIoTSP。该协议能够在保持与ELASTICO相同吞吐量的前提下缩短区块的生成周期，但其分片配置过程仅考虑了节点的地理位置因素，并且缺少分片重配置过程，使得网络在长期运行后可能会因为某些节点发生故障而无法继续正常工作，降低了系统的鲁棒性。

上述研究虽然在一定程度上解决了区块链的性能瓶颈，但是仍存在未考虑区块链容量不足等问题。因此，亟须设计一种新的区块链架构，以应对来自工业互联网的海量数据。为此，本文构建一种分层分片区块链架构LSchain（Layering and Sharding blockchain），关键思想是根据节点之间的拓扑结构将区块链网络划分为多个分片，并为各分片选取能够最小化区块广播时间的主节点。每个分片都对一组不相交的交易运行PBFT共识算法进行验证。在一个共识周期（epoch）内，成功经过验证的交易区块会被主节点打包为一个压缩区块，压缩区块内包含指向这些交易区块的指针。边缘层各分片会周期性地将交易区块卸载到云区块链层进行存储，本地只存储体积更小的压缩区块。

1　LSchain系统模型

针对区块链应用于工业互联网时所面临的问题，本文构建一种适用于工业互联网场景的分层分片区块链架构LSchain，如图1所示，LSchain包括工业互联网平台层、边缘区块链层、云区块链层等3层。

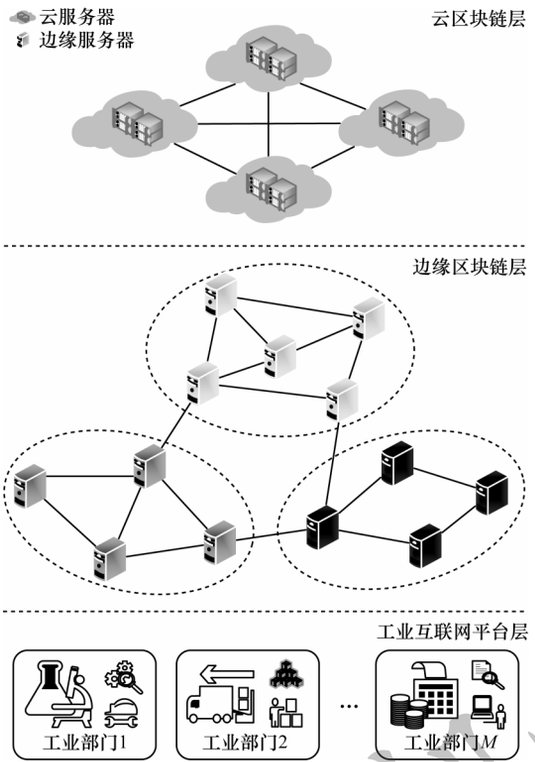

图1 LSchain系统模型

Fig.1 LSchain system model

工业互联网平台层由多个业务相互隔离的工业部门构成,这些工业部门可以是来自垂直行业或水平行业的不同企业和工厂。各工业部门内包含大量的异构设备,这些设备由于算力和存储空间有限,无法充当区块链节点。工业互联网平台层产生的大量数据会被发送到边缘区块链层中进行共识验证。

边缘区块链层由大量边缘服务器构成,是LSchain的核心层。这些边缘服务器拥有更强的算力和更大的存储空间,经由CA鉴权后加入网络。依据边缘服务器之间的拓扑结构,所有边缘服务器会被划分为若干个分片,各分片并行地运行PBFT共识算法验证来自工业互联网平台层的海量数据。不同分片之间的交易数据来自业务相互隔离的不同工业部门,各分片之间的数据互不共享,不存在跨分片交易。由于边缘服务器的存储容量有限,无法满足大规模工业互联网中海量数据长期存储的需求,因此边缘区块链层会周期性地将经过共识验证的交易区块卸载到云区块链层存储,边缘区块链层只存储包含交易区块索引的压缩区块。

云区块链层由多个分布式云(例如,阿里云、腾讯云等云服务租赁平台)构成,它们组成云区块链网络,共同维护从边缘区块链层卸载过来的区块数据。由于云区块链层不是本文研究的重点,因此不展开详细讨论。

2 LSchain协议设计

本节重点介绍LSchain边缘区块链层中涉及的两个重要部分,即区块链网络分片和分片内主节点选取。下面详细介绍每一部分的实现。

2.1 基于社团划分的区块链网络分片算法

边缘层区块链网络的节点由经过CA鉴权的边缘服务器担任,这些节点通常只会因为宕机、网络延迟等原因而未能按预期参与共识过程,但不排除节点被恶意劫持的可能。因此,引入声誉机制,用声誉值r来描述节点的可信程度[16]。根据r的不同,节点的信誉状态可以分为ST0、ST1、ST2、ST3等4类。节点的信誉状态会随着它们在共识过程中的行为发生变化,在共识过程中表现良好的节点会受到声誉值奖励,奖励公式如下:

$$r_{ij}=r_{ij}+r_{\text{reward}} \tag{1}$$

其中:r_{ij}是第j个分片中第i个节点的声誉值;r_{reward}是奖励的声誉值。

在共识过程中做出错误决策的节点会受到声誉值惩罚,惩罚公式如下:

$$r_{ij}=r_{ij}-r_{\text{punishment}} \tag{2}$$

其中:$r_{\text{punishment}}$是惩罚的声誉值。r_{reward}和$r_{\text{punishment}}$的取值可以根据实际应用场景进行调整。

节点信誉状态的转换过程如图2所示。

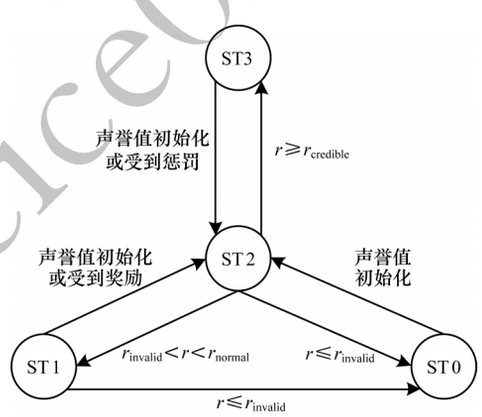

图2 节点信誉状态的转换过程

Fig.2 Conversion process of the node reputation state

在网络初始化时,LSchain会将所有节点的初始r都置为r_{normal}。r_{credible}、r_{normal}和r_{invalid}为3个声誉门限值,它们的关系为$r_{\text{invalid}}<r_{\text{normal}}<r_{\text{credible}}$。状态ST3表示节点在共识过程中表现良好,$r\geqslant r_{\text{credible}}$;状态ST2表示节点在共识过程中表现正常,$r$满足$r_{\text{normal}}\leqslant r<r_{\text{credible}}$;状态ST1表示节点在共识过程中做出过错误决策或对于某些决策未响应,r满足$r_{\text{invalid}}<r<r_{\text{normal}}$;状态ST0表示节点在共识过程中频繁做出错误决策或长时间未响应,$r\leqslant r_{\text{invalid}}$,此时认为该节点已被恶意劫持。为避免低可信度的节点影响到网络的共识过程,根据节点的信誉状态对节点权限进行分类,如表1所示。

表1　节点权限分类

Table 1　Node permission classification

节点信誉状态	是否能竞选主节点	是否能参与共识
ST3	是	是
ST2	否	是
ST1	否	是
ST0	否	否

信誉状态为ST1和ST2的节点都只能作为普通共识节点参与共识过程；信誉状态为ST3的节点才有资格竞选成为主节点；信誉状态为ST0的非法节点将被禁止参与接下来的共识过程。该权限分类可以有效防止恶意节点阻碍共识过程。

基于联盟链的准入机制和上述声誉机制对节点可信状态的描述，恶意节点将被隔离在网络之外。因此，对边缘层区块链网络分片时，无需再通过复杂的计算来确保分片后区块链网络的安全性。根据边缘服务器之间的拓扑结构，可以将边缘层区块链网络表示成如下矩阵形式：

$$\boldsymbol{A}=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \tag{3}$$

其中：$\boldsymbol{A}$为边缘层区块链网络的邻接矩阵；a_{ij}用来刻画节点之间的连通情况，a_{ij}=1表示节点n_i和n_j之间有边相连，a_{ij}=0表示节点n_i和n_j之间无边相连，只有借助其他节点转发才能通信，并且规定$a_{ii}\equiv 0$，$i=1,2,\cdots,n$。

上述工作从复杂网络角度分析了边缘层区块链网络。基于此，LSchain探索将区块链网络的分片问题转化为复杂网络的社团划分问题。复杂网络的社团划分研究主要包含以下3个方面：1）如何定义一个社团；2）如何判定划分结果的优劣；3）如何在一个合理的划分标准下，设计一种高效的划分算法。

经典的社团划分算法GN（Girvan-Newman）通过不断从网络中移除介数最大的边，将整个网络划分为不同的社团[17]，但GN算法的复杂度达到了$O(n^3)$（n为网络中节点的数量），目前仅局限于对中等规模的网络进行社团划分。

基于上述原因，NEWMAN等[18]在GN算法的基础上提出一种快速算法——FN（Fast-Newman）。FN算法继承了贪心算法的思想，从所有节点各自作为一个社团开始，沿着使模块度Q增大最多或减小最少的方向不断合并社团，直至整个网络都合并为一个社团。由于FN算法需要在每轮合并过程中遍历有边相连的社团并计算模块度增量，这一过程的复杂度为$O(m)$（m为网络中边的数量），并且在合并完成后还需要更新社团结构，这一过程的复杂度为$O(n)$，因此每轮合并的复杂度为$O(m+n)$。又因为FN算法最多需要合并$n-1$轮，所以其总复杂度为$O((m+n)n)$。在FN算法中，用模块度Q来衡量社团划分结果的质量，Q的取值范围为[0，1]，值越大说明社团结构越明显，划分结果越好。一般的网络社团划分结果的Q值介于0.3~0.7，Q的计算公式[19]如下：

$$Q=\sum_{i=1}^{k}\left(e_{ij}-a_i^2\right) \tag{4}$$

其中：k为社团数；e_{ij}表示网络中连接两个不同社团s_i和s_j的节点的边相对于所有边的比例；a_i表示与社团s_i中的节点相连的边相对于所有边的比例。

FN算法需要不断合并社团，只有整个网络都合并为一个社团时才收敛。边缘层各分片区块链网络采用的PBFT共识算法对参与共识的节点数量有严格限制，即节点数应不小于4且不大于100[20]。为使FN算法能够应用于边缘层区块链网络的分片过程，LSchain在FN算法的合并过程中改进了合并约束与收敛条件，以控制合并后的分片规模，并减少合并过程的算力消耗与合并轮数。

对于具有n个节点的边缘层区块链网络，改进的FN算法包括以下3个执行步骤：

步骤1　将网络初始化为n个分片，即每个节点各自作为1个分片。此时Q=0，e_{ij}和α_i满足式（5）和式（6）：

$$e_{ij}=\begin{cases}\dfrac{1}{2m}, n_i\sim n_j \\ 0, \text{其他}\end{cases} \tag{5}$$

$$a_i=\frac{k_i}{2m} \tag{6}$$

其中：$n_i\sim n_j$表示节点n_i和n_j之间有边相连；k_i为节点n_i的度。

步骤2　遍历有边相连的分片对，判断将分片对合并后是否满足约束：

$$\text{card}\left(s_i\cup s_j\right)\leqslant 100 \tag{7}$$

其中：$s_i\cup s_j$表示将分片s_i和s_j合并；card()函数用于计算有限集合的元素个数。式（7）用于控制合并后的分片规模，确保每个分片包含的节点数不超过100，并减少ΔQ的计算次数。若分片对满足式（7），则计算合并后的ΔQ_{ij}：

$$\Delta Q_{ij}=e_{ij}+e_{ji}-2a_ia_j=2\left(e_{ij}-a_ia_j\right) \tag{8}$$

根据贪心算法的原理，从满足式（7）的分片对中选取能使Q增大最多或减小最少的分片对合并。在每轮合并后，更新对应的e_{ij}，并将$\boldsymbol{E}=(e_{ij})_{n\times n}$矩阵中对应$s_i$、$s_j$分片的行和列相加，然后计算$Q=Q+\max\{\Delta Q\}$。

步骤3　重复执行步骤2以不断合并分片，直至满足以下收敛条件：

$$\forall s_k,s_l\in S,\text{card}\left(s_k\cup s_l\right)\geqslant 100 \tag{9}$$

式（9）的物理含义为：继续合并网络中任意两个分片后均会使合并后的分片规模大于100。借助式（9），本文改进的FN算法可以减少合并轮数，从而

提升算法的时间性能。

改进合并约束与收敛条件的FN算法流程如图3所示。

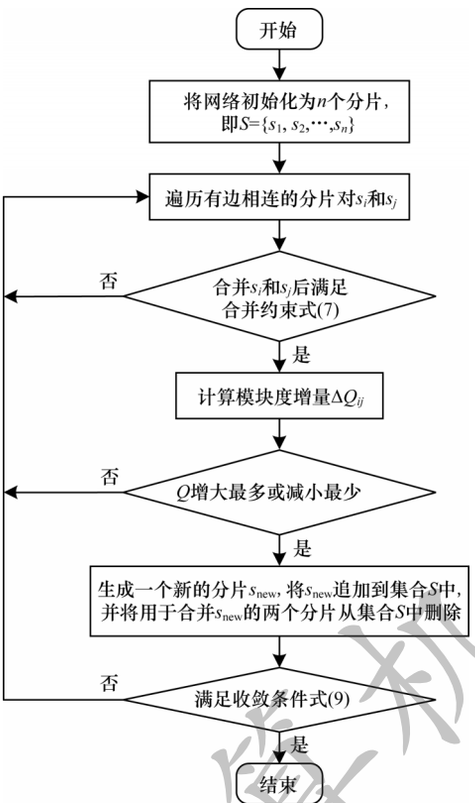

图3 改进合并约束与收敛条件的FN算法流程

Fig.3 FN algorithm procedure for improving merger constraints and convergence conditions

根据边缘区块链层节点数量的不同，算法的运行结果分为2种情况：1）如果$n \leqslant 100$，则改进FN算法与原始FN算法的运行结果相同，均会得到一个分片结构分解的树状图，通过选择在不同位置断开可以得到不同的分片配置方案；2）如果$n>100$，则改进FN算法的运行结果为包含多个分片结构分解树状图的森林，其中的每一颗树代表一个分片，可以根据实际应用场景的需求进一步拆分其中的某些分片，以获得更多的分片数量。改进的FN算法通过减少合并过程中ΔQ的计算次数与合并轮数将算法的复杂度降为$O((m'+n)n')$（$m' \leqslant m$且$n' \leqslant n$，等号仅在$n \leqslant 100$时成立），从而提高了算法的时间性能，减少了边缘层区块链网络分片花费的时间。

2.2 基于生成树的分片内主节点选取算法

在工业互联网中的许多应用场景都对时延提出了很高的要求，而区块链中的共识机制往往非常耗时[21]。无论是公有链广泛采用的PoW、权益证明（Proof of Stake，PoS）共识算法，还是受联盟链和私有链青睐的拜占庭容错（Byzantine Fault Tolerance，BFT）类共识算法，都需要耗费大量的时间用于向全网广播区块，包括边缘层各分片区块链采用的PBFT共识算法。在PBFT中，一个主节点将交易打包成区块，并将其广播至全网供其他节点验证的过程可以转化为一颗生成树。以一个随机生成的小规模网络拓扑为例，如图4所示，在这个拓扑中有9个节点和16条边。假设节点6是网络中的主节点，由主节点打包生成的区块需要广播至整个网络，在图5中的生成树可以表示区块的广播过程。

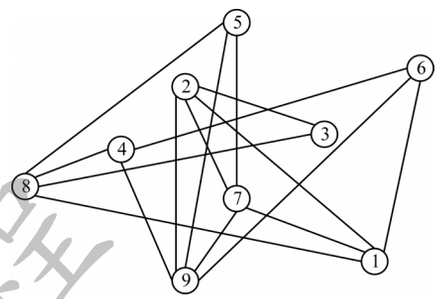

图4 随机生成的网络拓扑

Fig.4 Randomly generated network topology

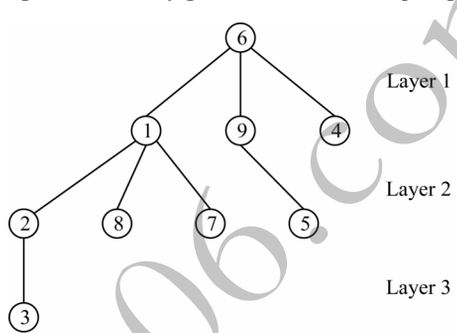

图5 基于生成树的区块广播过程

Fig.5 Block broadcast process based on spanning tree

首先，以节点6为根节点求图4的生成树，可以确保区块以最少的通信次数广播至整个网络。在类似图5的生成树中，假设v_i是节点n_i验证区块所需的时间，则所有节点的验证延迟集合V表示如下：

$$V=\{v_1,v_2,\cdots,v_I\} \tag{10}$$

其中：I是生成树中的节点数。

然后，用d_{ij}表示任意两个有边相连的节点（如节点n_i和它的邻居节点n_j）之间的传输延迟；用c_m表示节点n_m接收到区块并完成区块验证所花费的时间；用N_m表示从主节点到节点n_m的路径上所有节点的标识集（其元素按接收区块的顺序排列）。基于此，区块从主节点传播至节点n_m的时间c_m可以表示如下：

$$c_m=\sum_{N_m} v_i+\sum_{N_m} d_{ij} \tag{11}$$

其中：$\sum_{N_m} v_i$表示传播路径上所有节点的总验证延迟；$\sum_{N_m} d_{ij}$表示传播路径的总传输延迟。

最后，用C来表示区块传播至整个树所需的时间，这也是网络中最后一个节点接收到区块并完成验证所需的时间。区块的总传播时间可以表示如下：

$$C=\max\{c_1,c_2,\cdots,c_l\} \tag{12}$$

将式(11)代入式(12)后发现，路径上的节点数与路径上的最后一个节点接收区块并完成验证所花费的时间之间存在很强的相关性。每条传播路径对应于树的一个分支，路径上的节点数量决定了分支的长度。由此可以推断出：网络中参与区块验证的最后一个节点位于最长分支的末端，该分支的长度等于生成树的深度。综上所述，减少生成树的深度意味着减少区块广播至全网的时间，因此可以将区块的广播时间问题转化为生成树的深度问题。

基于上述结论，本文提出一种通过寻找分片内的最小深度生成树来选取分片内主节点的算法，以取代PBFT随机选取主节点的方案，详见算法1。

算法1　最小深度生成树算法

输入　分片重配置间隔w，k个分片的集合$S=\{s_1,s_2,\cdots,s_k\}$，边缘层的n个节点在过去w轮epoch中累积的声誉值集合$R=[\{r_{11},r_{12},\cdots,r_{1l_1}\},\{r_{21},r_{22},\cdots,r_{2l_2}\},\cdots,\{r_{k1},r_{k2},\cdots,r_{kl_k}\}]$，声誉门限值$r_{threshold}$

输出　k个分片的主节点

1.for each s_i in S do //遍历k个分片

2.$D_i=P_i=[\ ]$//初始化k个分片，用于存放备选节点和备//选节点的生成树深度的空数组

3.for each n_{ij} in s_i do //遍历第i个分片中的I_i个节点

4.if $r_{ij}\geqslant r_{threshold}$ then //判断第i个分片中的第j个节点能否//竞选主节点

5.$n_{ij}\rightarrow P_i$ //将节点n_{ij}加入分片s_i的备选节点集中

6.Find spanning tree T_{ij} of shard s_i with n_{ij} as root node and the depth of the tree T_{ij} is denoted as d_{ij}

7.$d_{ij}\rightarrow D_i$ //将生成树T_{ij}的深度d_{ij}存入数组D_i

8.else

9.continue

10.end if

11.end for

12.if not only one node in P_i has the minimum value in D_i then

13.Compare their r and make the node with higher r as the leader node of shard s_i

14.else

15.Select the node with the minimal value in D_i and make it as the leader node of shard s_i

16.end if

17.end for

算法1通过寻找分片内区块传播深度最小的生成树，为每个分片选取可以最小化区块广播时间的主节点。因为网络初始化时会将所有节点的声誉值都置为r_{normal}，所以在每次网络初始化后第一次运行最小深度生成树算法时，声誉门限值应设为r_{normal}。经过几轮epoch共识后，为选取更加可靠的节点担任分片内的主节点，声誉门限值应设为$r_{credible}$。

下面以图6中4个节点运行PBFT算法的共识流程为例，分析由最小深度生成树算法选取的主节点对PBFT共识算法吞吐量的影响。

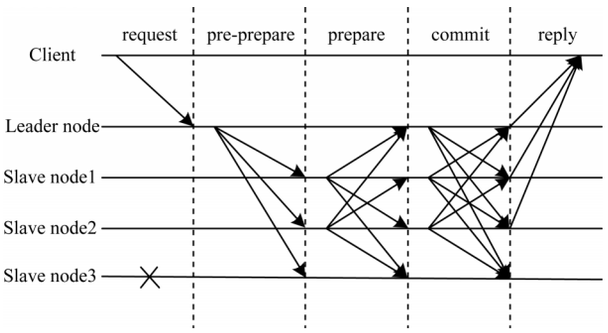

图6　4个节点的PBFT共识流程

Fig.6　PBFT consensus flow of four nodes

共识流程的3个核心阶段分别为pre-prepare阶段、prepare阶段和commit阶段[22]，其中，pre-prepare阶段是最小深度生成树算法关注的由主节点向全网广播区块的阶段。PBFT共识算法的吞吐量可以表示如下：

$$T_{TPS\_PBFT}=N/T \tag{13}$$

其中：N表示时间T内包含的交易数量。

将T分解为通信时间T_c和验证时间T_v，则式(13)可以表示如下：

$$T_{TPS\_PBFT}=N/\left(T_c+T_v\right) \tag{14}$$

将通信时间进一步拆分为算法5个阶段的通信时间$T_{c_request}$、$T_{c\_pre\text{-}prepare}$、$T_{c\_prepare}$、T_{c_commit}和T_{c_reply}后，式(14)可以表示如下：

$$T_{TPS\_PBFT}=N/\left(T_{c\_request}+T_{c\_pre\text{-}prepare}+T_{c\_prepare}+T_{c\_commit}+T_{c\_reply}+T_v\right) \tag{15}$$

重点关注区块广播pre-prepare阶段，将其余4个阶段的通信时间合并为T_{c_others}，式(15)可以进一步表示如下：

$$T_{TPS\_PBFT}=N/\left(T_{c\_pre\text{-}prepare}+T_{c\_others}+T_v\right) \tag{16}$$

假设将pre-prepare阶段的区块广播过程转化为生成树后，树的深度为d，则式(16)可以表示如下：

$$T_{TPS\_PBFT}=N/\left(T_{c\_depth_1}+T_{c\_depth_2}+\cdots+T_{c\_depth_d}+T_{c\_others}+T_v\right) \tag{17}$$

其中：$T_{c_depth_i}$表示生成树中第i层的通信时间。

假设任意两个边缘服务器之间的通信时延趋向于一个相近值$T_{c_average}$，则式(17)可以表示如下：

$$T_{TPS\_PBFT}=N/\left(d\times T_{c\_average}+T_{c\_others}+T_v\right) \tag{18}$$

由上述分析可知，将算法1中求出的生成树深度最小的节点作为分片内的主节点，可以缩短PBFT共识过程花费的时间，进而提升边缘层各分片区块链网络的吞吐量。

3　实验设置与结果分析

本节首先通过仿真实验对所提的改进FN算法和最小深度生成树算法进行性能分析，然后对LSchain的安全性进行理论分析。

3.1 性能分析

采用MATLAB进行仿真测试，实验环境如下：处理器Intel® Core™ i5-10210U，主频1.60 GHz，内存16 GB，操作系统Microsoft Windows 10 64位。

3.1.1 改进FN算法的性能分析

2个性能指标为分片时间和分片结果Q值，对比算法为文献[18]所提的原始FN算法，实验结果均为多次运行结果取平均值。首先，对比2种算法对同一网络分片时所花费的时间，以验证改进的FN算法在缩短分片时间方面的性能。然后，对比2种算法对同一网络分片后的模块度Q值，以衡量改进FN算法的分片质量。实验所用数据集为来自斯坦福大学的大型网络数据集网站和Mark Newman的个人数据集网站的5个关系网络，5个网络的规模信息如表2所示。

表2 数据集基本信息

Table 2 Dataset basic information

编号	网络名称	节点数/个	边数/条
1	Dolphin social network	62	159
2	American college football	115	613
3	Neural network	297	2 359
4	Email-Eu-core network	1 005	25 571
5	Political blogs	1 490	16 718

从图7可以看出，改进的FN算法在对小型网络分片时花费的时间与原始FN算法几乎相同，但随着网络规模的增大，改进的FN算法对网络分片时花费的时间明显少于原始FN算法，并且时间差越来越大，对于5号这一大型网络，改进的FN算法能够在不牺牲分片质量的前提下将分片时间缩短约36%。这是因为：当网络中的节点数不超过100时，改进的FN算法等同于原始FN算法；当节点数超过100时，改进的FN算法通过改进合并约束与收敛条件，减少了合并过程中ΔQ的计算次数与合并轮数，从而提高了算法的时间性能，缩短了分片时间。

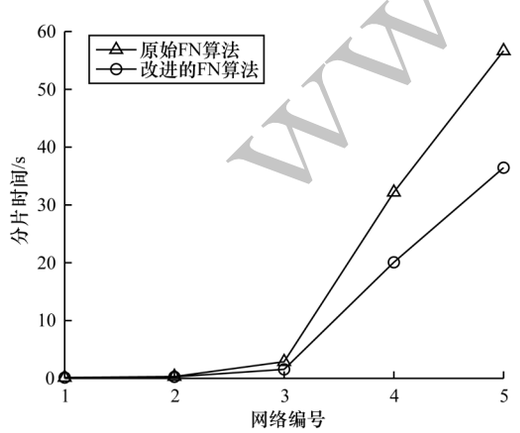

图7 分片时间对比

Fig.7 Comparison of the sharding times

从图8可以看出：当网络中的节点数不超过100时，2种算法的分片结果Q值相同，再次印证了对于节点数不超过100的小规模网络，改进的FN算法等同于原始FN算法；当网络中的节点数超过100时，改进FN算法的分片结果Q值围绕原始FN算法的分片结果Q值在不超过0.1的范围内波动。这是因为改进的FN算法在继承原始FN算法最大化Q值思想的同时，也不可避免地继承了原始FN算法只能选取局部最优解的特性（根据2.1节的描述，算法在合并过程中的某几轮可能会使Q值减小）。对网络编号为2~5的网络进行分片后，改进的FN算法与原始FN算法相比减少了合并轮数，而减少的这几轮合并过程既有可能使Q值继续增大，也有可能使Q值减小，导致了改进的FN算法的分片结果Q值与原始FN算法相比时高时低，但2种算法的Q值差距很小，且2种算法的Q值均在正常范围内。

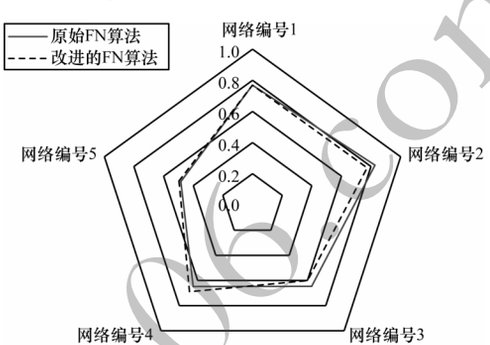

图8 分片结果Q值对比

Fig.8 Comparison of the Q value of sharding results

从图7、图8的实验结果可以看出，本文改进的FN算法在不牺牲分片质量的前提下提升了分片效率，从而减少了区块链网络分片以及分片重配置的时间。

3.1.2 最小深度生成树算法的性能分析

为了验证最小深度生成树算法在减少区块广播时间方面的性能，选取主节点的生成树深度作为性能指标。对比方案为文献[22]所提共识算法采用的主节点选取策略，实验结果均为多次运行结果取平均值。首先对比由最小深度生成树算法选取的主节点和由PBFT算法选取的主节点的生成树深度，为保证实验的公平性，规定PBFT算法只能从最小深度生成树算法的备选节点集中选取主节点；然后对比分片数量对最小深度生成树算法性能的影响；最后测试网络的连通性对最小深度生成树算法性能的影响。

实验数据由随机网络生成模型Salama[23]生成。Salama模型中有2个重要的网络特征参数α和β，其中，α代表网络中短边相对于长边的比例，β代表网络中边的密度。α和β共同决定了网络的连通性，α/β越大，网络的连通性越好。实验通过控制Salama模型的网络特征参数α和β，生成规模不同但连通性相同的网络以模拟各分片区块链网络。下面如无特殊说明，均由相同比例的α/β生成实验网络，以确保

不同规模的网络连通性相同。

从图9可以看出，在不同分片规模（包含的节点数不同）下，最小深度生成树算法选取的主节点的生成树深度均要小于PBFT算法选取的主节点的生成树深度。这是因为PBFT共识算法的主节点是在每轮共识开始之前随机选取的。最小深度生成树算法通过遍历备选节点的生成树，选取生成树深度最小的节点来担任主节点。因此，在减少区块广播时间方面，最小深度生成树算法的主节点选取策略优于PBFT算法的主节点选取策略。

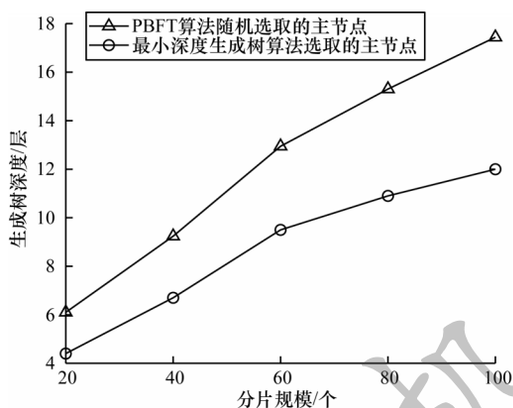

图9　主节点的生成树深度对比

Fig.9　Comparison of the spanning tree depth of leader node

从图10可以看出，当同一个网络被分为不同分片时，各分片由最小深度生成树算法选取的主节点的生成树深度与分片数成反比。这是因为实验为控制变量，假设每个分片的规模相同（包含的节点数相同）。分片数越多，每个分片的规模越小，由最小深度生成树算法选取的主节点的生成树深度就越浅。基于此，LSchain可以根据工业互联网平台层的业务需求动态地划分边缘层区块链网络：当网络负载大时，将网络分为更多的片，以优先满足吞吐量需求；当网络负载小时，将网络分为更少的片，以更好地保障网络的安全。

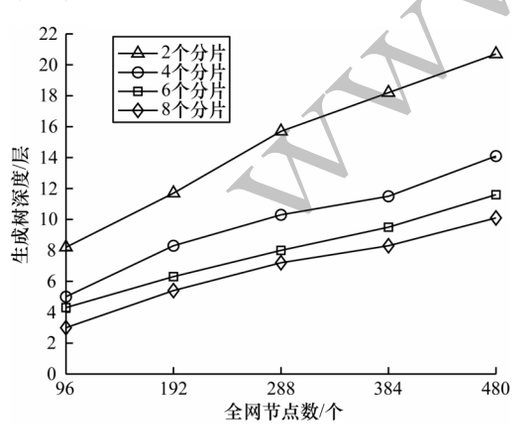

图10　同等网络规模下不同分片数的生成树深度对比

Fig.10　Depth comparison of the spanning tree with different shards number in the same network scale

从图11可以看出，α/β越大（即网络的连通性越好）的分片，由最小深度生成树算法选取的主节点的生成树深度就越浅。这是因为连通图（任意两个节点都有边相连的网络）的生成树深度具有理论上下限。节点数量为n的全局耦合网络仅需1跳即可到达网络中的任意位置，这种网络的生成树深度恒为1，连通性最好；任一节点数为n，边数为$n-1$的网络，其生成树的最小可能深度为$(n-1)/2$，最大可能深度为$n-1$，这种网络的连通性最差。所以，连通性越好的网络，其生成树的最小可能深度越接近1。

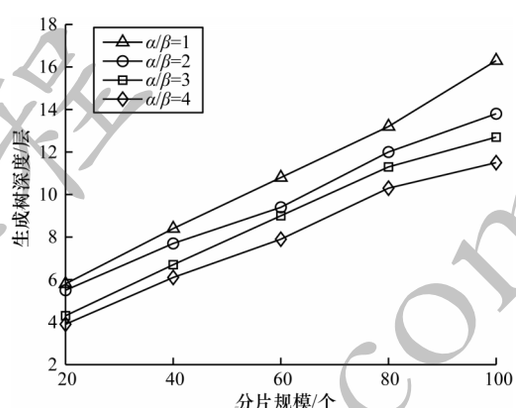

图11　同等网络规模下不同连通度分片的生成树深度对比

Fig.11　Comparison of the spanning tree depth of the shards with different connectivity in the same network scale

3.2　安全性分析

在边缘层区块链网络中，承担区块链节点角色的各边缘服务器可能会遭到恶意攻击，从而影响系统的正常运行。本节主要从LSchain协议能够抵御单分片接管攻击和主节点自私攻击两方面进行说明。

3.2.1　单分片接管攻击抵御

为确保边缘层区块链网络的安全，所有分片均需满足PBFT算法的容错限制，即恶意节点数f不超过节点总数的1/3。任何分片突破这个限制，都会导致分片被恶意节点劫持，从而影响整个边缘层区块链网络的正常运行，这被称为单分片接管攻击[24]。为了能及时侦测到被恶意劫持的分片，LSchain规定各分片内的节点应定时向CA发送心跳消息，心跳消息中包含各节点用自己的私钥对所处分片安全状态的决策签名。当诚实节点在共识过程中检测到分片内的恶意节点数超过PBFT容错限制时，会向CA发送否认分片状态安全的决策签名。在各分片的心跳消息集中需要包含至少$f+1$（确保至少有1个诚实节点可以向CA反馈真实的网络状态）条为安全状态背书的签名，该分片才能继续运行。若某分片的心跳消息集中少于$f+1$的节点认为该分片安全，则停用该分片，并计算该分片中各节点在本轮epoch中累积的r。根据节点的r判断节点的信誉状态，状态为ST0

的节点将被禁止参与接下来的共识过程，直至其经过安全漏洞修复并向CA申请才能重新加入网络。因此，LSchain协议在一定程度上抵御了单分片接管攻击。

3.2.2 主节点自私攻击抵御

PBFT共识算法需要在每轮共识开始前随机选取主节点，而LSchain通过最小深度生成树算法为各分片选取的主节点可以最小化区块广播时间，并且具有较高的声誉值。因此，各分片内的主节点在没有不良行为的情况下无需频繁更换，但当分片内的其余节点检测到主节点作恶或长时间未响应时均会触发视图切换（view-change）过程[25]。view-change会重新运行最小深度生成树算法以选取新的主节点，同时作恶主节点的声誉值将会被直接清零。该方式与普通节点作恶的处置方式相同，直至经过安全漏洞修复并向CA申请才能重新加入网络。因此，本文方案在一定程度上抵御了主节点自私攻击。

4 结束语

本文针对区块链应用于工业互联网时所面临的吞吐量不足和存储压力较大的问题：首先构建分层存储架构，将工业互联网产生的海量数据在边缘区块链层和云区块链层中进行分层维护，解决了区块链存储容量不足的问题；然后基于经典社团划分算法设计并改进区块链网络分片算法，在提升区块链吞吐量的同时缩短了分片时间；最后论证了区块广播时间对吞吐量的影响，提出一种能够最小化各分片内区块广播时间的主节点选取算法，进一步提升了边缘层区块链的吞吐量。实验结果表明，与经典社团划分算法以及随机选取主节点的策略相比，所提方案能够同时减少分片时间和各分片内区块的广播时间。下一步将完善边缘层交易区块的卸载机制，并针对工业互联网场景设计合理有效的区块存储卸载算法，以降低边缘层区块链节点的存储压力。

参考文献

[1] LI G C, ZHAO Q L, ZHANG D B, et al. GT-Chain: a fair blockchain for intelligent industrial IoT applications[J]. IEEE Transactions on Network Science and Engineering, 2022, 9(5): 3244-3257.

[2] SEN S, SONG L. An IIoT-based networked industrial control system architecture to secure industrial applications [C]//Proceedings of 2021 IEEE Industrial Electronics and Applications Conference (IEACon). Washington D. C., USA: IEEE Press, 2021: 280-285.

[3] LENG J W, YE S D, ZHOU M, et al. Blockchain-secured smart manufacturing in industry 4.0: a survey[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2021, 51(1): 237-252.

[4] WU Y, DAI H N, WANG H. Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0[J]. IEEE Internet of Things Journal, 2020, 8(4): 2300-2317.

[5] ZENG P J, WANG X L, DONG L Z, et al. A blockchain scheme based on DAG structure security solution for IIoT [C]//Proceedings of the 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). Washington D. C., USA: IEEE Press, 2022: 935-943.

[6] JAVAID U, SIKDAR B. A checkpoint enabled scalable blockchain architecture for industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2021, 17(11): 7679-7687.

[7] QI X D, ZHANG Z, JIN C Q, et al. A reliable storage partition for permissioned blockchain[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(1): 14-27.

[8] HAFID A, HAFID A S, SAMIH M. Scaling blockchains: a comprehensive survey[J]. IEEE Access, 2020, 8: 125244-125262.

[9] CAI X J, GENG S J, ZHANG J B, et al. A sharding scheme-based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial Internet of Things [J]. IEEE Transactions on Industrial Informatics, 2021, 17(11): 7650-7658.

[10] LUU L, NARAYANAN V, ZHENG C D, et al. A secure sharding protocol for open blockchains[C]//Proceedings of 2016 ACM SIGSAC Conference on Computer and Communications Security. New York, USA: ACM Press, 2016: 17-30.

[11] KOKORIS-KOGIAS E, JOVANOVIC P, GASSER L, et al. OmniLedger: a secure, scale-out, decentralized ledger via sharding[C]//Proceedings of IEEE Symposium on Security and Privacy. Washington D. C., USA: IEEE Press, 2018: 583-598.

[12] ZAMANI M, MOVAHEDI M, RAYKOVA M. RapidChain: scaling blockchain via full sharding[C]// Proceedings of 2018 ACM SIGSAC Conference on Computer and Communications Security. New York, USA: ACM Press, 2018: 931-948.

[13] WANG J, WANG H. Monoxide: scale out blockchains with asynchronous consensus zones [C]//Proceedings of USENIX Symposium on Networked Systems Design and Implementation. Boston, USA: USENIX, 2019: 95-112.

[14] YEASMIN S, BAIG A. Permissioned blockchain-based security for IIoT [C]//Proceedings of 2020 IEEE International IoT, Electronics and Mechatronics Conference. Washington D. C., USA: IEEE Press, 2020: 1-7.

[15] TONG W, DONG X, SHEN Y, et al. A hierarchical sharding protocol for multi-domain IoT blockchains[C]// Proceedings of 2019 IEEE International Conference on Communications. Washington D. C., USA: IEEE Press, 2019: 1-6.

[16] HUANG C, WANG Z, CHEN H, et al. RepChain: a reputation-based secure, fast, and high incentive blockchain system via sharding[J]. IEEE Internet of Things Journal, 2020, 8(6): 4291-4304.

[17] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2002, 99(12): 7821-7826.

（下转第79页）

（上接第66页）

[18] NEWMAN M E J, GIRVAN M. Fast algorithm for detecting community structure in networks[J]. Physical Review E: Statistical Nonlinear & Soft Matter Physics, 2004, 69(6): 066133.

[19] SHEN M E, PENG M F, LI S J, et al. Optimal island partition of ADN based on complex network community structure [C]//Proceedings of the 9th International Conference on Power Science and Engineering(ICPSE). Washington D. C. , USA: IEEE Press, 2020: 17-21.

[20] SUKHWANI H, MARTÍNEZ J M, CHANG X L, et al. Performance modeling of PBFT consensus process for permissioned blockchain network (Hyperledger Fabric) [C]//Proceedings of the 36th Symposium on Reliable Distributed Systems. Washington D. C. , USA: IEEE Press, 2017: 253-255.

[21] YU Y, LIU S M, YEOH P L, et al. LayerChain: a hierarchical edge-cloud blockchain for large-scale low-delay industrial Internet of Things applications[J]. IEEE Transactions on Industrial Informatics, 2021, 17(7): 5077-5086.

[22] CASTRO M, LISKOV B. Practical Byzantine fault tolerance [C]//Proceedings of the 3rd Symposium on Operating Systems Design and Implementation. New York, USA: ACM Press, 1999: 173-186.

[23] SALAMA H F. Multicast routing for real-time communication of high-speed networks[D]. Raleigh, USA: North Carolina State University, 1996.

[24] LI W Y, FENG C L, ZHANG L, et al. A scalable multi-layer PBFT consensus for blockchain [J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(5): 1146-1160.

[25] WANG G, SHI Z J, NIXON M, et al. SMChain: a scalable blockchain protocol for secure metering systems in distributed industrial plants[C]//Proceedings of International Conference on Internet of Things Design and Implementation. New York, USA: ACM Press, 2019: 249-254.

编辑　陆燕菲