计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2024/18(01)-0252-13
doi: 10.3778/j.issn.1673-9418.2211070

DAGGraph: 适合移动自组网的区块链

张文韬,黄建华+,顾　彬,宁宇豪,宫在为
华东理工大学 信息科学与工程学院,上海 200237
+ 通信作者 E-mail: jhhuang@ecust.edu.cn

摘　要:针对区块链与移动自组网结合所面临的挑战,采用有向无环图(DAG)结构适配移动性引发的网络分裂,提出一种基于DAG的系统模型DAGGraph。首先,对分簇算法进行优化,提出了簇内节点密度数量限制算法,从而有效解决了簇内节点数量不受控增加带来的吞吐量减小和能耗增加问题;其次,针对移动自组网节点快速移动引起的网络拆分与合并问题,提出了基于簇首间数据同步的区块恢复算法,通过DAG结构保留所有节点产生的合法区块,在网络合并时由原簇首交换产生的区块分支,实现了对区块分支的同步和恢复;最后,提出了一种简化的区块上链算法,在内部节点可信的前提下简化了区块的上链流程,减小了移动环境下区块传播产生的误差,缩短了区块的确认时间,提高了系统吞吐量。安全性分析表明,系统可以抵御针对区块链的常见攻击,并可以抵御针对移动自组网的拒绝服务攻击。仿真实验结果表明,DAGGraph的时延和吞吐量性能在大部分情况下优于已有的物联网区块链解决方案。

关键词:区块链;移动自组网;分簇算法;共识机制
文献标志码:A　　**中图分类号:**TP393

DAGGraph: Blockchain Suitable for Mobile Ad Hoc Networks

ZHANG Wentao, HUANG Jianhua+, GU Bin, NING Yuhao, GONG Zaiwei
School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

Abstract: Aiming at the challenges faced by the combination of blockchain and mobile ad hoc networks, a system model DAGGraph based on DAG (directed acyclic graph) is proposed, which adopts the DAG structure to adapt to the network split caused by mobility. Firstly, the clustering algorithm is optimized, and an algorithm for limiting the density of nodes in the cluster is proposed, which effectively solves the problem of throughput reduction and energy consumption increase caused by the uncontrolled increase of the number of nodes in the cluster. Secondly, for network splitting and merging caused by the rapid movement of the nodes, a block recovery algorithm based on data synchronization between cluster heads is proposed. The legal blocks generated by all nodes are preserved through DAG. When the network is merged, the original cluster heads exchange their generated block branches, realizing the recovery of the block branches. Finally, a simplified block appending algorithm is proposed, which simplifies the block appending process on the premise that the internal nodes are trusted, reduces the error caused by block propagation in the mobile environment, shortens the block confirmation time, and improves the system throughput. Security analysis shows that DAGGraph can resist common attacks against blockchain, and resist denial-of-service attacks against mobile ad hoc networks. Simulation results show that the latency and throughput of DAGGraph are better than existing IoT blockchain solutions in most cases.

Key words: blockchain; mobile ad hoc networks; clustering algorithm; consensus mechanism

基金项目:国家自然科学基金(62076094)。
This work was supported by the National Natural Science Foundation of China (62076094).
收稿日期:2022-11-17　　**修回日期:**2023-01-12

移动自组网(mobile ad hoc networks, MANET)是一种具有自组织能力、可快速部署的特殊类型物联网(Internet of things, IoT),它无需固定基础设施的支持,在军事战场、紧急救援和环境勘探等特殊环境中应用前景广阔。在执行任务的过程中,MANET的节点之间需要通过无线信道传输数据和通过操作指令进行协同,这些数据和操作指令存在多种安全风险且容易受到攻击。因此,维护MANET数据的机密性、完整性和真实性显得非常重要。

区块链是一种在计算机网络节点之间共享信息的分布式数据库,它以区块的形式组织搜集的信息,通过密码学将区块之间链接在一起,以确保数据的不变性。作为一项新兴的技术,区块链为分布式系统提供了去中心化和防篡改的能力,确保了数据的真实性和完整性。一些学者尝试利用区块链来解决物联网的安全问题。Islam等人[1]提出使用基于区块链的智能合约,实现了移动节点间机器学习模型的安全共享;Abishu[2]与Hassija[3]等学者提出将交易数据上链,实现移动节点间以及移动节点与固定节点间安全的能源交易。

然而,由于MANET的移动性会引发网络拓扑的动态变化,将区块链与MANET相结合面临不少挑战。Laube等人[4]首次探讨了移动性引发的网络拓扑变化问题,提出了应对MANET分裂和合并问题的解决方案,从理论上证明了在MANET中应用区块链技术的可行性。但该方案采用被动的方式检测网络拓扑的变化,存在效率较低下、网络延迟等问题,并且未就检测网络分裂提出有效的解决方案。BlockGraph[5-6]将基于有向无环图(directed acyclic graph, DAG)的区块链模型应用于MANET,重新设计了区块的数据结构和Raft共识算法,解决了传统区块链在MANET分裂和合并下产生的问题,保证了数据的正确性与完整性。但是该模型的共识时延易受到节点数量的影响,在大规模的集群下共识效率不高。虽然以上工作取得一些进展,但是将区块链应用于MANET仍然存在以下问题需要解决。首先,不同类型的任务往往要求移动节点或分散或聚集地开展工作,需设计高效的分簇算法,使得节点快速分簇的同时有效控制簇内节点的数量;其次,受到环境和任务变化等因素的影响,网络的结构将进行分裂和合并等动态调整,而网络的分裂将引发区块链的分叉,在网络合并后需合理地处理这些分叉;最后,节点的通信信号易受到山体、树木和建筑物等大型物体的干扰,在此环境下,节点间传递区块所需的控制和验证信息极易丢失。

针对以上问题,本文解决问题的思路是采用DAG结构来适配移动性引发的网络拓扑的变化,以解决区块链分叉问题;通过簇首控制簇节点密度,以限制入簇节点数量,确保簇的性能;简化出块和验证流程,减小数据丢失对区块上链所带来的影响。提出一种基于DAG结构的系统模型DAGGraph,以有效地控制每个簇的规模,提高分簇的速度,实现网络合并后区块链分支的安全恢复,通过简化上链流程提高共识速度,增加系统的吞吐量和鲁棒性。本文的贡献如下:

(1)提出簇节点密度(数量)限制算法,有效地解决了一个簇的节点数量不受控增加所引发的性能下降问题。

(2)针对网络分裂和合并引发的网络拓扑的变化,提出基于簇首间数据同步的区块恢复算法,在网络合并后由原簇首交换并同步产生的区块分支,实现对区块分支的恢复,以保留所有节点产生的合法区块。

(3)提出一种简化的区块上链算法,简化了区块的上链流程,节点仅需对收到的区块进行本地验证即可完成上链操作,增强了系统的健壮性。

1 相关工作

分簇算法可以解决MANET中节点资源开销大、可扩展性不高等问题。分簇算法将平面网络结构中邻近的节点组成一个簇,一个簇包含一个簇首和若干个簇成员节点,簇首与簇成员协同执行任务。运行分簇算法的节点周期性地发送控制信息,这些控制信息用来进行簇首选举、节点入簇和节点移动等操作。最小ID分簇算法[7]是较早提出的分簇算法,该算法要求每个节点拥有唯一的ID,在网络初始化阶段,节点周期性地向其他节点广播自己的ID,节点通过比较收到的ID,选择ID最小的节点作为网络的簇首。最小ID分簇算法的一个显著特点是算法的收敛性高,实现简单。陈志军等人[8]对最小ID分簇算法进行了改进,将剩余电量和节点相对移动速度作为簇首选举的参考因素,使节点能耗更均衡,网络拓扑更稳定。受到外部环境因素的影响,在实际应用中,节点的移动方向、速度等特性在一定程度上具有某种规律。通过设计合理的数学模型,可以预测节点在某一时间段内的移动特性,降低分簇算法的开销。

宋人杰等人[9]指出移动节点具有分组移动的特性,通过计算得出节点的移动特性并进行分簇,同时将节点性能作为簇首选举的参考因素以提高系统性能。陈宇等人[10]基于卫星节点运行轨迹的可预测性,将簇的初始化阶段交由可信的地面终端完成,简化了分簇算法的复杂度,提高了网络的稳定性。吴振华等人[11]根据车辆移动位置的可预测性,按路段将城市道路划分成簇,通过车辆节点的实时位置预测移动趋势,降低了分簇算法的路由发现及分簇的开销。MANET的节点基于无线方式通信,存在消息泄露风险,网络拓扑易受到攻击。崔朝阳等人[12]根据MANET的特点,提出安全分簇算法。该算法通过证书交换,确保可信节点加入网络,通过协商会话密钥,完成对信息的加密,提高了分簇算法的安全性。

基于DAG的区块链可以实现区块并发写入,且能较好地解决由网络分裂引起的区块链分叉问题,受到研究人员的广泛关注。使用DAG结构的区块链项目主要有Nano(https://nano.org/en/whitepaper)、Byteball(https://byteball.org/Byteball.pdf)、IOTA(http://tanglereport.com/wp-content/uploads/2018/01/IOTA_Whitepaper.pdf)、DagCoin(https://dagcoin.org/whitepaper.pdf)等。DAG账本的每一个子单元可以引用验证多个父单元,一个父单元也可以被多个子单元验证,这种验证关系在提高交易确认速度的基础上确保了DAG账本的安全性和可靠性。为了解决由多分支合并引起的交易冲突问题,GHOST[13]提出主链选择协议,对于两个冲突的交易,将位于主链上的交易视为有效。然而GHOST协议将丢弃主链以外的交易,造成对算力的浪费。Conflux[14]和Inclusive[15]基于GHOST主链选择协议,将节点产生的所有交易都视为DAG账本的一部分,并根据主链对交易进行排序,解决了交易冲突问题。安全性是移动自组网所面临的又一个问题,针对节点的恶意攻击会给MANET带来不可预测的风险和损失。其攻击类型主要包括物理攻击、恶意窃听、洪泛攻击[16]、拒绝服务攻击、双花攻击和女巫攻击[17]等。区块链具有分布式、防篡改等特点,但与移动自组网相结合的区块链也容易受到来自外部和内部的攻击。一些学者对于如何提高区块链系统自身的安全性进行了研究。李忠诚等人[18]根据区块链中诚实节点与恶意节点之间的博弈,提出了一种激励和押金机制。规定每个参与验证的节点都需要缴纳押金,参与共识的诚实节点将按缴纳的押金比例获得奖励,同时恶意节点会被扣除押金,以此激励节点理性共识,限制恶意行为。季钰翔等人[19]引入信任度评估机制,通过对邻居节点行为的监督,以有效检测恶意节点,同时引入工作量证明和押金机制来限制女巫攻击,保证区块链网络的安全性。英特尔软件保护扩展(Intel software guard extensions,SGX)[20]利用可信硬件提供安全容器,以确保安全容器中加载的代码和数据不能被外界篡改,提高了区块链的安全性。

在移动自组网环境中,节点的频繁移动会引起网络拓扑变化,当节点间的距离超出通信范围时,网络会分裂;节点连接恢复后,网络将合并。分簇算法增加了MANET组网的灵活性,提高了网络的性能,然而节点间灵活的组网能力也为区块链在MANET上的应用提出了不小的挑战。传统的单链式区块链根据最长链原则,将丢弃由网络分裂产生的分支,从而造成正常的数据丢失。区块图[5-6]将DAG结构应用于MANET,为区块链应对网络分裂和合并带来的问题提供了解决方案。当网络分裂成两个子网后DAG分叉,各子网在自己的分叉上独立地产生区块并上链;当网络合并后,启动区块恢复程序,恢复由其他子网产生的区块。区块图基于每个子网中的领导者节点完成日志复制和区块恢复等过程,然而区块图并未就网络的分裂和合并等拓扑变化给出具体的检测方案。Laube等人[4]提出了一种被动检测网络拓扑变化的方案,指出网络的出块速度应根据节点的数量动态变化,并给出了优化的方向。然而基于被动的方式可能造成网络延迟。

2 DAGGraph

2.1 系统模型

节点的移动性会引发网络拓扑的动态变化,如何确保使用区块链运行的任何应用在移动自组织网络中保持一致是本文需要解决的关键问题。移动自组网的特点是移动节点间连接的稳定性通常不及固定节点,节点的无线通信范围有限且易受环境因素的影响,节点因执行任务移动到不同位置或外部干扰会造成连接中断,引发网络分裂,形成几个称为簇的独立分区,分区内的节点可以相互通信,但分区之间彼此不能直接通信。当任务发生变化后,分裂的簇又会逐渐靠近,合并成一个网络。

区块链通常假设网络是稳定的并且具有良好的可用性。这些假设不适用于可能出现网络分区的移动自组网,因为网络的分裂和合并都会引发网络拓

扑变化,给区块链网络维持数据一致性带来挑战。在部署区块链的移动自组网中,当网络出现分区时,不同的簇之间不能直接通信,全网节点无法达成整体共识。要确保网络分区后区块链能继续提供服务,不同的簇会独立共识以延展区块链,从而引发区块链分叉。当网络再次合并时,要确保数据保持可用和一致,传统区块链会将多余的分支进行修剪,仅保留最长链分支。删除分支会删除许多有用的数据,引起数据丢失,这是不接受的。

针对以上问题,本文采用DAG结构来适配移动性引发的网络拓扑的变化,以保留网络分裂阶段各簇独立产生的区块链分支,并通过簇首处理区块的恢复和同步问题,提出一种基于DAG结构的系统模型DAGGraph,如图1所示。

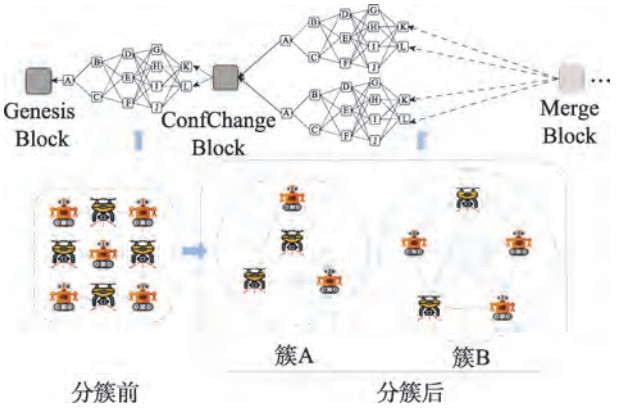


图1 DAGGraph系统模型

Fig.1 DAGGraph system model

图1反映了网络运行过程中网络拓扑的动态变化,网络分裂后形成多个簇(分区),由于通信距离有限,各簇间无法直接通信。每一个簇由一个簇首和多个成员节点组成,同一簇内的节点具有相同的出块权,均可在属于本簇的DAG分支上产生区块。如图1所示,节点移动后网络中产生A与B两个簇,簇A的节点和簇B的节点在各自的DAG分支上产生区块,两个分支可在网络合并后由旧簇首恢复给所有成员节点,同时由新簇首负责生成合并块以收敛此前产生的DAG分支。

系统基于私有链运行,可信证书授权机构(certificate authority,CA)为每一个参与节点颁发证书。节点之间通过互换各自的证书,相互验证,未被授予证书的节点将不被允许加入系统。节点间的消息传输采用会话密钥加密传输,会话密钥采用密钥交换技术协商产生,以确保安全性。

本文将区块链与移动自组网相结合需要解决的问题简化为三个:分簇管理问题、共识问题和区块恢复问题。分簇管理主要基于分簇算法形成簇结构,处理网络初始化和维护簇结构,通过引入加密机制支持每个加入簇的节点与邻居节点协商会话密钥,确保数据安全传输。区块管理模块由共识模块和区块恢复模块组成,其中共识模块允许每个簇独立地产生区块,区块恢复模块负责网络合并后恢复由不同簇产生的区块。

2.2 分簇管理

MANET被广泛应用于抢险救灾、地质勘探和水文观测等领域,这些应用场景往往根据任务形式和物理环境的不同,要求节点分散地开展任务。然而受到通信距离的限制,MANET节点的分布不宜过于松散,节点间的远距离通信造成网络的不稳定连接将频繁引发数据丢包和数据包重传,大大增加通信能耗。利用分簇算法,将松散分布的节点聚集成簇,可在一定程度上缓解此类问题。然而,现有的分簇算法大都基于地理位置对集群进行分簇,若某一位置的节点数量过多,极易造成一个簇内的节点密度过高,节点间的频段干扰以及带宽占用等问题将严重影响节点间的通信效率。因此,本文设计了一种高效的分簇管理机制,包括分簇和簇首选举模块、簇结构维护模块、拓扑变更检测模块、会话密钥协商模块等。该机制以簇的形式组织MANET节点,以发挥MANET节点的团队协作优势,提高工作效率,降低通信能耗。

如图2所示,分簇管理机制中的节点包含五种节点状态:未定状态、游离状态、成员状态、簇首状态、簇首隐藏状态。其中未定状态为每个节点启动后的初始状态;游离状态表示节点未发现邻居节点。簇首选举基于加权分簇算法,该算法综合考虑了节点性能、网络带宽、剩余电量等因素,选举综合素质最高的节点进入簇首状态。簇内的节点状态为成员状态,处于簇首状态和成员状态的节点动态维护簇成

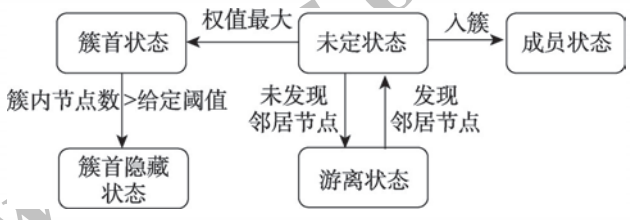


图2 节点状态转换图

Fig.2 Node state transition

员表。此外,为了减小网络通信负担,本文为簇首设计了控制簇内节点数量的机制,当簇内节点数量过多,簇首转为隐藏状态,此时簇首不再允许新节点加入簇。

基于节点灵活移动的特性,分簇管理的任务如下:网络初始化、节点加入、节点退出、节点移动等。为了更加高效地检测网络拓扑变化,本文还定义了两种拓扑状态:拓扑变化状态、拓扑稳定状态。初始阶段节点都处于拓扑变化状态,待节点检测到分簇完成后转为拓扑稳定状态。

2.2.1 网络初始化

网络初始化即簇结构的初始化:将网络中物理距离接近的节点划分成包含一个簇首和多个成员节点的簇,网络初始化流程如图3所示。

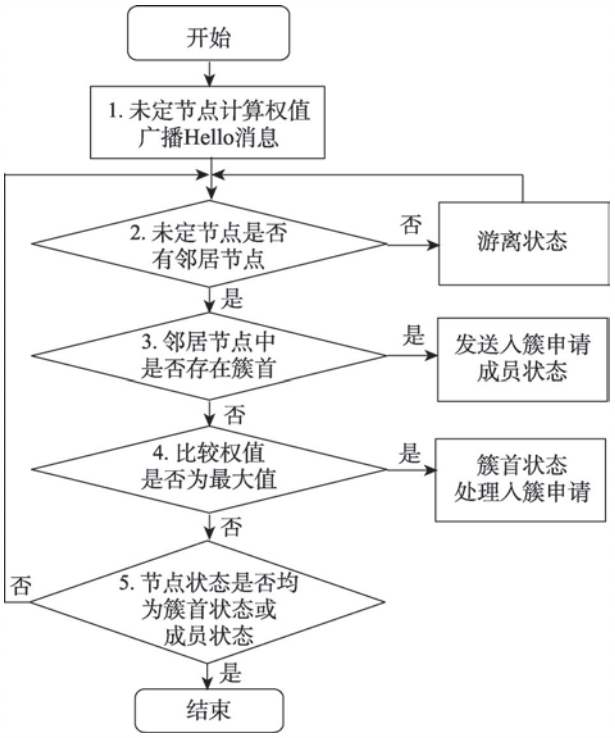


图3 网络初始化流程

Fig.3 Network initialization process

为完成簇结构的初始化,节点需向全网广播Hello消息,Hello消息包含节点ID、节点证书、节点权值、节点状态、时间戳和消息验证码。其中节点权值 W 综合考虑了节点的处理器性能 A、网络带宽 M 和剩余电量 E 等因素,权值计算公式为:

$$W=\alpha\times A+\beta\times M+\gamma\times E \tag{1}$$

$$\alpha+\beta+\lambda=1 \tag{2}$$

当节点收到来自其他节点的Hello消息后,根据消息验证码和节点证书来验证节点的身份和消息的真实性。消息验证通过后,根据节点发出的Hello消息信号强度计算节点间的距离:

$$\|ij\|=\frac{\lambda}{4\pi}\sqrt{\frac{P_T}{P_R}} \tag{3}$$

其中,λ 为波长,P_T 为发送方功率大小,P_R 为接收方功率大小。节点 i 与节点 j 进行 n 次距离的测算,两个节点间平均距离为:

$$\overline{\|ij\|}=\frac{1}{n}\sum_{k=1}^{n}\|ij\|_k \tag{4}$$

若平均距离小于给定的阈值,将该节点信息加入自己的内存,若节点的内存中存在状态为簇首的节点,则发送入簇申请。若节点的内存中存在多个可选的簇首,则计算簇首的优先级:

$$p(i,CH_v)=\frac{w_v}{N\times s} \tag{5}$$

其中,w_v 表示簇首 CH_v 的权值,N 表示簇内节点数量,s 表示节点 i 与簇首 CH_v 的平均距离。节点可向优先级最大的簇首发送入簇申请。

若节点内存中不包含簇首,则选择权值最大的节点作为簇首,其余节点以簇成员的身份加入。簇内节点主要采用广播方式进行通信,若节点数量过多,将会造成网络拥堵和延迟等问题。为控制簇的节点数量,本文提出簇首隐藏状态的概念,当簇首检测到簇内的节点数量超过给定阈值时,将自己的状态转为簇首隐藏状态,不再处理入簇申请,其余节点也不会选择隐藏状态的簇首加入。

2.2.2 簇结构维护

簇结构维护即系统针对簇结构变化的一系列反应,节点的下列三种行为将引起簇结构的变化:节点加入、节点离簇和节点移动。在本文中节点可采用主动或被动的方式监测簇结构的变化。

节点收到簇首的Hello消息后,进入节点加入流程。首先计算簇首的优先级,选择向优先级最高的簇首发送入簇申请。簇首收到入簇申请后,验证节点的身份信息,验证通过后,同意节点入簇,广播修改后的簇成员表。成员节点收到广播后,同步修改簇成员表。

根据节点的状态和节点退出簇的方式,可将节点退出分为四类:成员节点正常退出、成员节点非正常退出、簇首节点正常退出、簇首节点非正常退出。节点离簇前需要向簇内各节点广播离簇消息,表示自己即将离簇,成员节点离簇后簇首需要修改并广

播簇成员表,对于簇首的离簇,需要指定或选举出新的簇首。正确广播离簇消息的节点被视为正常离簇。此外,本文采用被动方式检测节点的非正常离簇,正常运行的节点需要在每个周期广播Hello消息,若某节点连续三个周期未广播Hello消息,则其余节点可将其视为非正常离簇。

当节点受到环境变化或任务改变等因素的影响,需要移动较大的距离,其选择退出当前簇加入另一簇的过程就完成了一次节点移动。节点移动是节点退出和节点加入的组合。若节点移动的距离过大,无法接收到任何簇首的Hello消息,则将自己的状态改为游离状态,等待接收簇首的Hello消息。

综上所述,本文主动监测簇内网络拓扑变化的方式为簇首接收到节点的入簇申请,以及簇内节点收到离簇节点的离簇消息;被动监测拓扑变化的方式为连续三个周期未收到节点的Hello消息。将主动与被动的形式相结合,可以更加有效地判断簇结构是否发生变化,增强系统的稳定性。

2.2.3 拓扑状态变更

对于簇首节点,若连续三次广播的簇成员表都相同,则将自己的拓扑状态从拓扑变化状态改为拓扑稳定状态。簇首修改簇成员表后,需要将自己的拓扑状态改为拓扑变化状态。对于簇成员节点,若连续三次收到相同的簇成员表,则将自己的拓扑状态从拓扑变化状态改为拓扑稳定状态。当簇成员收到簇首更新的簇成员表或连续三个Hello消息周期未收到簇首的Hello消息,需要将自己的拓扑状态改为拓扑变化状态。

2.2.4 会话密钥协商

完成分簇的网络初始化操作后,节点处于拓扑稳定状态,进入密钥协商阶段。本文的密钥协商过程基于棣弗-赫尔曼(Diffie-Hellman,DH)密钥交换,如图4所示,簇内的节点两两协商一个共同的会话密钥 K_i。图5展示了节点 i 与节点 j 协商密钥时所发送的消息,密钥协商涉及的相关符号如表1所示。后续阶段如有新节点入簇,新节点需要向簇内的所有节点协商会话密钥,密钥协商结束后,节点运行区块管理模块。

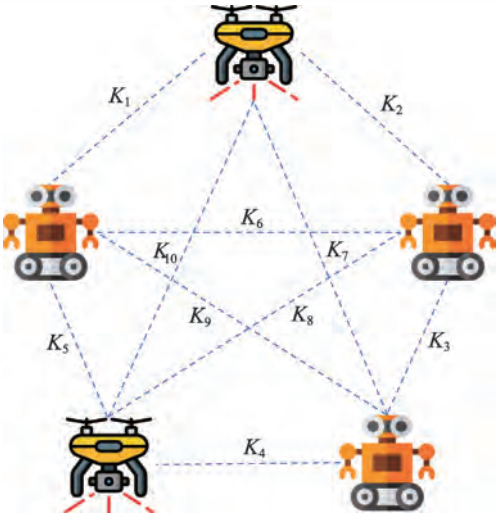


图4 会话密钥协商

Fig.4 Session key negotiation

表1 密钥协商相关符号

Table 1 Relevant notation of key negotiation

符号	说明
$CERT_i$	节点证书
P	大质数
G	大质数的生成元
A, B, R_k	随机数
T_k	时间戳
MAC	消息验证码
K_i	对称密钥
$\{\cdots\}_{K_i}$	使用对称密钥加密消息
$\{\cdots\}_{Priv_i}$	使用私钥加密消息

2.3 区块管理模块

区块管理模块由共识模块和区块恢复模块组成。系统完成密钥协商后,每一个簇开始运行共识模块,独立地进行共识,簇中的每个节点(包括簇首和簇成员)都有相同的概率和地位产生区块,区块由会话密钥加密后在簇内广播。受到任务和环境变化等因素的影响,网络可能经历分裂和合并等拓扑变

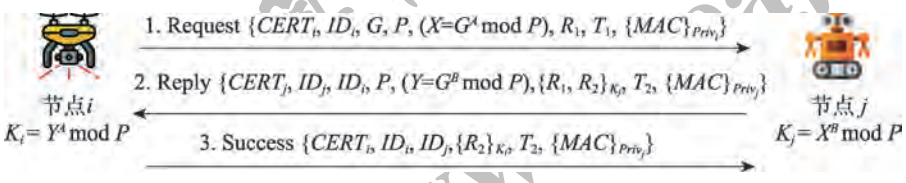


图5 会话密钥协商流程

Fig.5 Session key negotiation process

化，区块恢复模块负责网络拓扑改变后对区块链分支进行恢复。如图6所示，系统基于DAG结构构建区块链，网络分裂成多个簇时，由于网络隔离，区块链会产生分支，每个簇在各自的分支上产生区块，每一条分支也基于DAG结构，提高了系统的吞吐量。当多个簇合并成一个簇后，节点运行区块恢复模块，恢复由其他簇产生的区块。

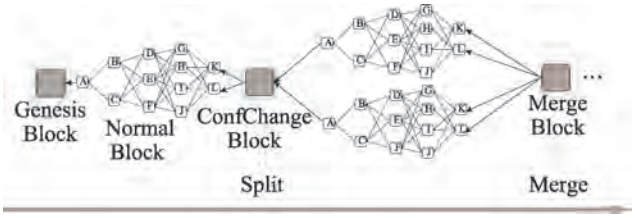


图6 DAGGraph账本

Fig.6 DAGGraph ledger

如图6所示，本系统将区块分为四种类型：创世区块（Genesis Block）、配置更改块（ConfChange Block）、合并块（Merge Block）以及普通块（Normal Block）。创世区块由簇首产生，是系统产生的第一个区块，随后簇中的每一个节点都有相同的概率产生普通块，普通块可包含交易在内的任何类型的事务信息。为简化出块流程，减小网络通信负载，每个普通块只包含一条事务信息，即节点直接将自己的事务打包成区块，交由共识模块共识。网络发生分裂后，将形成多个簇，每个簇的簇首都会在本簇的账本上生成一个相同的配置更改块，用于收敛之前生成的区块。网络发生合并后由新簇首产生合并块，合并块将引用区块链各分支上所有未被引用的区块，用于合并各簇产生的分支。

本系统定义的区块结构如图7所示，主要包含区块类型、区块标记、父哈希列表和随机数Nonce等字段。区块标记包含该区块的创建者信息：若区块类型为普通块，则区块标记为簇ID，同一簇内的节点产生的普通块包含相同的簇ID，簇ID由簇首在拓扑稳定后生成，并广播给簇成员节点。簇ID的计算公式为：

$$簇ID = Hash(\sum 簇内节点ID) \tag{6}$$

若区块类型为簇首产生的创世区块、配置更改块或合并块，则区块标记为簇首ID。为确保系统稳定出块，保证区块链系统的安全性，防止女巫攻击等恶意行为，节点在产生区块前需运行一次工作量证明（proof of work，PoW）算法。随机数Nonce在节点运行PoW算法后产生，父哈希列表包含区块引用的所有父区块的哈希值。

图7 区块结构

Fig.7 Block structure

2.3.1 共识模块

共识模块负责区块的生成上链。在DAGGraph中，每个节点均可进行出块，在网络质量良好、信道占用率低的情况下，簇内的节点越多，簇的出块速度越大。然而随着簇内节点数量的增加，节点间因出块进行的通信可能会占用大量的信道资源，对系统的整体性能造成影响。因此，本文除第2.2.1小节所提出的簇首隐藏状态的概念用于限制簇节点密度外，还将通过共识模块，根据簇内节点数量的变化动态调整PoW算法的难度以调整簇的出块速度。出块速度可简化为：

$$Speed \propto \frac{N \times p \times A}{D} \tag{7}$$

其中，N表示簇内节点数量，p表示节点出块的概率，A表示节点性能，D表示共识算法的难度。因此，出块速度与簇内节点的数量、节点出块的概率以及节点的性能成正比，与共识算法的难度成反比。共识模块随节点数量的增加而适当提高共识算法的难度，可在一定程度上限制出块速度，以确保簇内信道占用率处于可接受的水平，保证系统的平稳运行，所选取的PoW难度适中，对系统的整体能耗影响较小。此外，在节点出块前引入PoW算法还可以增加恶意节点作恶成本，提高系统安全性。

共识模块负责产生区块和区块上链。区块的共识不同于Raft算法需要超过半数节点返回确认消息，本系统的节点出块流程如下：首先节点打包自己产生的事务，接着运行一次简单的PoW算法，并广播自己的区块。其他节点收到区块后，需要验证区块是否满足共识模块规定的共识算法难度，如满足则进

入区块上链阶段,如不满足则丢弃。采用工作量证明实现对区块的验证,可减少集群中的通信量,提高共识速度。如图8所示,节点拓扑状态稳定后,由簇首向簇成员节点广播开始共识消息,簇成员收到消息后运行共识模块。共识模块将节点拓扑稳定后的时间划分为多个时间片(Epoch),节点可在每个Epoch产生区块,每个新上链的区块需要引用前一个Epoch产生的所有区块。如果拓扑状态发生改变,如节点退出或入簇,则簇内的节点停止出块,待拓扑状态稳定后重新运行共识模块。退出簇的节点重新组成另一个新的簇后,网络分裂成两个簇,区块链也将分裂为独立的分支,待拓扑稳定后两个簇在各自的分支上继续进行共识。

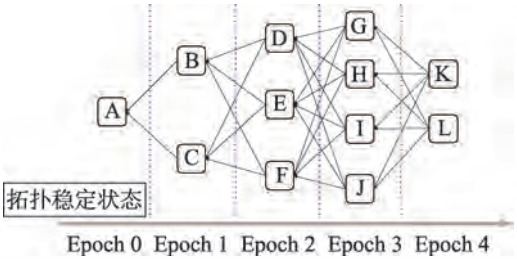


图8 DAGGraph共识机制

Fig.8 DAGGraph consensus mechanism

2.3.2 区块恢复模块

区块恢复模块用于在簇间和簇内,对新加入的节点恢复区块链分支。当网络发生合并时,合并的簇之间需要恢复其他簇产生的分支;节点加入某一簇后,也需要从该簇节点恢复自己所缺失的所有区块。如图9所示,簇A和簇B合并为一个新的簇,以簇A为例,它需要恢复簇B产生的区块。簇A的簇首向簇B的簇首发送请求消息,请求簇B的完整分支,簇A的簇首收到完整分支后在本地进行恢复,然后向自己的簇成员广播来自簇B的分支,实现簇A的所有节点对簇B分支的恢复。簇B也以相同的方式恢复簇A产生的分支。待区块恢复完成后,合并后的簇选举新的簇首,然后由新簇首生成合并块,收敛之前的区块链分支。

如图10所示,簇A的节点1因某种原因离开本簇后加入簇B,节点1需要恢复簇B产生的区块,节点1的区块恢复任务由簇B的簇首负责完成。簇B的簇首检测到节点1的加入后,判定簇内网络拓扑发生变化,立即停止共识,并生成配置更改块,运行区块恢复流程。节点1向簇B的簇首发送请求消息,请求簇B的区块分支,节点1收到簇B分支后进行恢复,并删除由簇A产生的分支。待网络拓扑稳定后,簇B的簇首产生配置更改块,表示节点1的区块恢复成功,簇B的节点可继续进行共识。

2.4 节点注册

CA具有授权DAGGraph节点的权限。在实际应

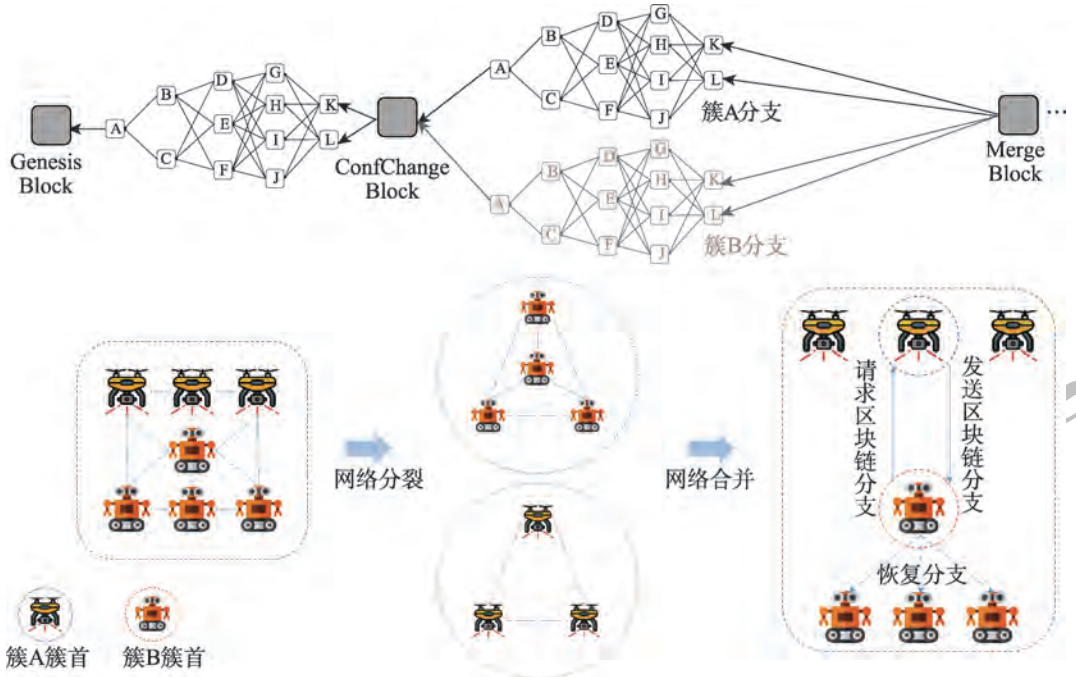


图9 簇间区块恢复

Fig.9 Inter-cluster block recovery

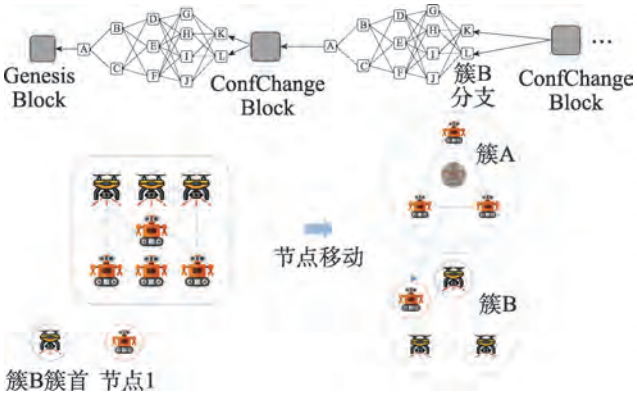


图10 簇内区块恢复

Fig.10 Intra-cluster block recovery

用中,一批运行DAGGraph系统的节点可以先后向CA提出注册申请。CA完成对节点的验证后,向节点颁发证书。考虑到运行DAGGraph系统的节点通常不具备较高的算力,因此需要一个安全高效的对称加密方案完成证书的传递。每个节点开始注册前,先通过椭圆曲线数字签名算法(elliptic curve digital signature algorithm,ECDSA)生成自己的公钥 *Pub* 和私钥 *Priv* 。如图11所示,CA通过节点的公钥向节点传输对称密钥,并最终使用对称密钥构建的安全信道向节点传递证书,完成节点注册,节点注册所涉及的相关符号如表2所示。

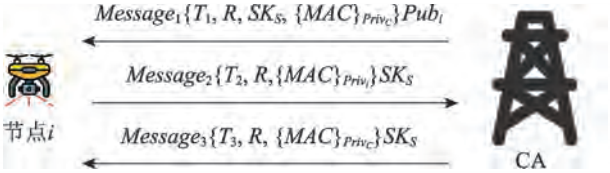


图11 节点注册

Fig.11 Node registration

表2 节点注册相关符号

Table 2 Relevant notation of node registration

符号	说明
T_k	时间戳
R_k	随机数
SK_S	对称密钥
$Priv_i$	节点或机构的私钥
Pub_i	节点或机构的公钥
MAC	消息验证码
$\{\cdots\}_{Priv_i}$	使用节点或机构的私钥加密消息
$\{\cdots\}_{Pub_i}$	使用节点或机构的公钥加密消息
$\{\cdots\}_{SK_S}$	使用对称密钥加密
$CERT$	证书

2.5 网络合并检测

本文将网络合并定义为当前系统只存在一个簇,所有节点都已加入该簇,采用对比簇成员信息的方式检测是否发生网络合并。本系统基于私有链,当一批节点完成向CA的注册后,表示节点成功加入DAGGraph系统。所有节点完成注册后,CA将它们写入节点信息表。节点信息表包含所有加入系统的节点的身份信息。其他簇的节点入簇后,原有簇的节点比较簇成员表和节点信息表,如果两表内包含的簇成员节点信息相同,则表示发生了网络合并,节点需要运行区块恢复模块,完成对分支的恢复。

3 安全性与通信复杂度分析

3.1 安全性分析

本节讨论系统的分簇管理模块和共识模块易受到的安全攻击,并分析系统应如何防范。

拒绝服务(denial of service,DoS)攻击:在分簇阶段,恶意节点将大量无效的Hello消息广播给邻居节点,使节点无法处理正常的分簇请求。本系统采用CA颁发证书的形式,确保每个节点都可被验证身份,对于未被验证的节点,系统将直接忽略其发送的消息。

双花攻击:在传统链式结构的区块链中,双花攻击指恶意节点在区块链上发起一笔交易,当交易被打包成区块后,在该区块之前重新发布一个包含冲突交易的区块,造成区块链的分叉。若分叉链能够取代主链,则恶意节点的双花攻击成功。在基于DAG结构的区块链上,双花攻击指的是恶意节点在区块链的分支上发布两个包含冲突交易的区块,DAG结构虽然能够容忍分叉的产生,但冲突的交易也将影响区块链系统的正常运行。本系统根据时间片对区块进行排序,同一个时间片内的区块按哈希值进行排序,通过这种方式确定DAG账本内区块的总顺序,对于冲突的交易,将排序靠前的交易视为有效交易。

女巫攻击:女巫攻击指恶意节点冒充多个身份,争夺网络控制权,影响共识结果以及干扰节点正常工作等。进行女巫攻击的节点需要将自己的计算资源分配给多个身份,每个身份分配到的资源有限。本系统规定节点出块前需要运行一次PoW算法,有效地限制了女巫攻击。此外,CA负责为每个节点颁发证书,同一节点无法向CA申请多个证书,不被CA

验证的节点将无法运行本系统,从根源上杜绝了女巫攻击的发生。

3.2 共识阶段的通信复杂度分析

BlockGraph采用Raft算法,假设节点总数为 n ,则区块由leader发送后至少需要 $n/2$ 条确认消息才能上链,BlockGraph在共识阶段的通信复杂度为 $O(n)$。DAGGraph簇中的簇首和簇成员均可产生区块,且节点采用PoW算法验证区块的正确性,通过验证的区块即可上链,而节点无需向簇首或其他任意节点返回确认消息。因此DAGGraph的共识时延仅由区块的打包和传播时间以及节点运行PoW算法验证区块的时间构成,而区块的传播时间可忽略不计。区块的打包和验证仅需节点在本地计算和验证PoW算法,因此共识时延与节点个数无关,DAGGraph在共识阶段的通信复杂度为 $O(1)$,其通信效率要明显高于BlockGraph。

4 相关工作实验结果与分析

4.1 实验环境

实验采用Docker虚拟化技术模拟节点和节点运行所需的网络环境,Docker运行的软硬件环境如表3所示。本实验所采用的代码库均由论文作者编写,代码运行环境由python:3.5.4-jessie提供支持,实验所需的Docker镜像基于Python3编写生成。实验开始前先在Docker中创建一个网络以供节点通信使用,并使用Docker镜像在网络中批量生成模拟节点行为的Docker容器,容器可模拟BlockGraph和DAGGraph的共识行为,容器间使用用户数据报协议(user datagram protocol,UDP)进行通信。为便于仿真实验,本文假设一个区块内仅包含一笔由节点产生的交易,并将区块大小设置为8 KB。实验主要模拟BlockGraph和DAGGraph的出块和上链过程,通过对比二者在不同节点数量以及不同簇数量下的共识时延与吞吐量,来评估方案的性能。

表3 实验软硬件环境

Table 3 Software and hardware environments of experiment

软硬件	配置
Docker Engine	版本:20.10.17
CPU	i5-5257U CPU@2.70 GHz
Docker内存	6 GB
宿主机操作系统	MacOS 12.4

4.2 共识时延测试

共识时延指的是簇内完成一次共识所需的时间,共识时延的大小将在一定程度上影响区块链系统的性能。本文对比的区块链系统为BlockGraph。BlockGraph的共识机制基于Raft算法,核心是在每个分区内选举出领导者节点,由领导者节点向追随者节点复制区块,实现账本的一致性。Raft算法要求领导者节点收集超过半数的确认消息,才能确保区块复制成功,因此等待确认消息的时间将影响共识时延的大小。本文提出的DAGGraph将每一个簇内的节点分为一个簇首节点和多个簇成员节点,且每个节点均可以产生区块。为加强系统的安全性,节点产生区块前将运行一次简单的PoW算法,节点收到区块后仅需要验证必要的身份信息和区块的正确性,简化了区块的确认过程。图12为一个分区内BlockGraph和DAGGraph的共识时延随节点个数的变化关系。从实验结果可以看出,DAGGraph的共识时延明显优于BlockGraph。DAGGraph的共识时延随节点数量的变化缓慢波动,基本维持稳定的值。而BlockGraph的共识时延随节点数量的增加呈近似线性增长,原因是BlockGraph共识的通信复杂度为 $O(n)$,随着分区内节点数量的增加,领导者节点所需等待的确认消息越多,共识时延越长。如第3.2节所述,DAGGraph在共识阶段的通信复杂度取决于解决PoW难题和验证PoW所花费的时间,因此DAGGraph在共识阶段的通信复杂度为 $O(1)$,共识所需的时间仅取决于PoW算法的难度,与节点个数无关。

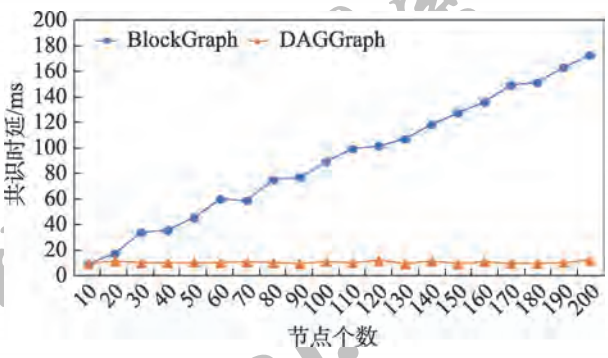


图12 不同节点个数下的共识时延对比

Fig.12 Comparison of consensus delay under different number of nodes

在实际应用中,整个网络内的节点数量往往是固定的,因此随着网络分区数量的增加,每个分区内的节点数量会相应地减少。在本实验中,将系统中

的节点总数固定为200个，分区的初始数量为20个，假设节点均匀地分布在各分区中，此时每个分区内的节点数量为10个；之后逐渐减少分区的数量，最终分区的数量减少为1个，分区内的节点数量为200个。图13显示了节点总数不变的前提下，分区数量与共识时延的关系。由实验结果可以得出，DAGGraph的共识时延在分区数量不同的情况下，仍能保持稳定。这是因为DAGGraph的共识时延仅取决于节点本地运行PoW算法的时间，与分区内节点个数无关，分区数量变化所引发的分区内节点数量的变化不会对DAGGraph的共识时延造成影响。而BlockGraph的共识时延随分区数量的减少而增加，受分区数量变化的影响较大。这是由于BlockGraph采用的Raft算法的性能随节点数的增加而下降。在节点数量不变的前提下，分区数量越少，分区内的节点数量越多，BlockGraph共识所需等待的确认消息越多，共识时延越大。

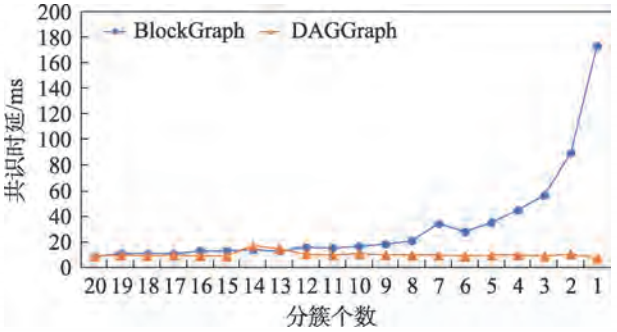


图13 不同分区数量下的共识时延对比

Fig.13 Comparison of consensus delay under different number of partitions

4.3 吞吐量测试

吞吐量指的是单位时间内区块链系统处理的交易数量，与分区数量、节点数量和区块内包含的交易数量成正比，与平均共识时延成反比。为便于仿真模拟，本实验假设每个区块包含一个交易，测试BlockGraph和DAGGraph的吞吐量与节点数量和分区数量的关系。

图14为吞吐量与节点数量的关系，可以看出DAGGraph的吞吐量整体高于BlockGraph，且随着节点数量的增加整体呈波动上升的趋势，具有一定的可扩展性。这是因为在DAGGraph中，簇首与簇成员是对等节点，均有出块权，网络中节点数量越多，则表示出块节点的数量越多，吞吐量越高。而BlockGraph基于Raft算法，在一个簇内只有一个领导者节点可以产生区块，因此吞吐量基本不随节点数量变化，可扩展性较差。

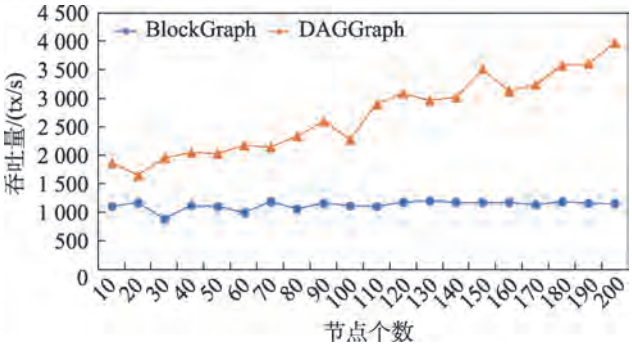


图14 不同节点个数下的吞吐量对比

Fig.14 Comparison of throughput under different number of nodes

在分区数量与吞吐量的实验中，同样保持节点总数为200个不变，且节点均匀分布在各分区中，则分区内的节点数量将随着分区数量的增加而减少，在此前提下测试吞吐量与分区数量的关系。如图15所示，随着分区数量的增加，两种方案的吞吐量均有一定程度的增长。DAGGraph比BlockGraph表现出更好的吞吐量性能，吞吐量的增长速度也更快。这是因为BlockGraph基于Raft算法，采用链式结构维护区块链账本，并由领导者节点复制区块；而DAGGraph的每个节点均可产生区块，且以DAG的形式维护账本，在分区数量更多的情况下，DAGGraph的吞吐量性能优势会更明显。

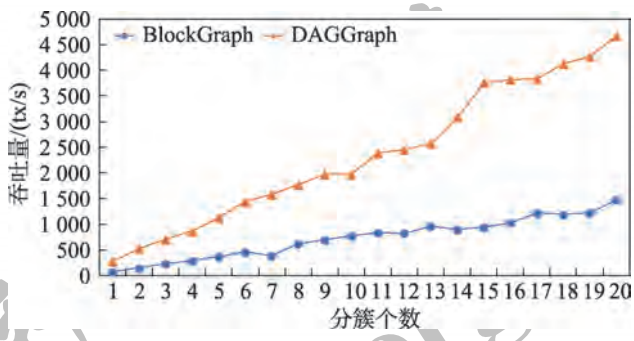


图15 不同分区数量下的吞吐量对比

Fig.15 Comparison of throughput under different number of partitions

5 结束语

移动自组网允许节点自发地形成网络拓扑，具有组网速度快、拓扑变化灵活等特点。区块链以安全和不可篡改等特性著称，将区块链与移动自组网

相结合是一种新颖的尝试。移动自组网节点的移动性会引发节点随时离开或加入,这种网络拓扑的动态变化给区块链的运行带来巨大挑战。本文提出区块链与移动自组网的结合需要解决的三个问题:分簇问题、共识问题和区块恢复问题。针对这三个问题提出了DAGGraph,一种基于DAG结构的系统模型。通过引入一种高效的分簇算法,使得节点通过主动发现方式完成网络拓扑变化(分裂)后的高效成簇并快速开始共识。在共识阶段,DAGGraph规定一个区块只包含一笔交易,简化了出块流程,节点只需验证区块的正确性和身份信息即可完成区块上链。在网络合并阶段,通过簇首节点进行分叉交换和区块同步,实现了对缺失区块的快速恢复。网络中传输的所有区块信息均进行加密,确保了区块传输的安全性。此外,DAGGraph以颁发证书的形式对节点进行验证,可以有效抵御DoS攻击、双花攻击、女巫攻击等恶意攻击。实验结果表明,DAGGraph在共识时延和吞吐量方面的性能较BlockGraph均有明显的优势。

参考文献:

[1] ISLAM S, BADSHA S, SENGUPTA S. A light-weight blockchain architecture for V2V knowledge sharing at vehicular edges[C]//Proceedings of the 2020 IEEE International Smart Cities Conference. Piscataway: IEEE, 2020: 1-8.

[2] ABISHU H N, SEID A M, YACOB Y H, et al. Consensus mechanism for blockchain-enabled vehicle-to-vehicle energy trading in the Internet of electric vehicles[J]. IEEE Transactions on Vehicular Technology, 2021, 71(1): 946-960.

[3] HASSIJA V, CHAMOLA V, GARG S, et al. A blockchain-based framework for lightweight data sharing and energy trading in V2G network[J]. IEEE Transactions on Vehicular Technology, 2020, 69(6): 5799-5812.

[4] LAUBE A, MARTIN S, AL A K. A solution to the split & merge problem for blockchain-based applications in ad hoc networks[C]//Proceedings of the 8th International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks, Paris, Nov 26-28, 2019. Piscataway: IEEE, 2019: 1-6.

[5] CORDOVA D, LAUBE A, PUJOLLE G. BlockGraph: a blockchain for mobile ad hoc networks[C]//Proceedings of the 4th Cyber Security in Networking Conference, Lausanne, Oct 21-23, 2020. Piscataway: IEEE, 2020: 1-8.

[6] MORALES D C, VELLOSO P B, LAUBE A, et al. C4M: a partition-robust consensus algorithm for block-graph in mesh network[C]//Proceedings of the 5th Cyber Security in Networking Conference, Abu Dhabi, Oct 12-14, 2021. Piscataway: IEEE, 2021: 82-89.

[7] EPHREMIDES A, WIESELTHIER J E, BAKER D J. A design concept for reliable mobile radio networks with frequency hopping signaling[J]. Proceedings of the IEEE, 1987, 75(1): 56-73.

[8] 陈志军, 史杏荣. 一种适合移动自组网的分簇算法[J]. 计算机工程与应用, 2007, 43(29): 159-161.
CHEN Z J, SHI X R. A clustering algorithm suitable for mobile ad hoc network[J]. Computer Engineering and Application, 2007, 43(29): 159-161.

[9] 宋人杰, 邹振婉, 周欣欣. 基于节点移动特性的移动P2P网络分簇算法[J]. 东北电力大学学报, 2016, 36(1): 87-90.
SONG R J, ZOU Z W, ZHUO X X. Clustering algorithm of mobile P2P network based on node mobility characteristics [J]. Journal of Northeast Electric Power University, 2016, 36(1): 87-90.

[10] 陈宇, 张勇, 陈实. 大规模卫星集群网络自适应加权分簇算法[J]. 北京理工大学学报, 2021, 41(11): 1188-1192.
CHEN Y, ZHANG Y, CHEN S. Adaptive weighted clustering algorithm for large-scale satellite cluster networks[J]. Transactions of Beijing Institute of Technology, 2021, 41 (11): 1188-1192.

[11] 吴振华, 胡鹏. VANET中一种优化路由开销算法研究[J]. 南昌航空大学学报(自然科学版), 2016, 30(2): 28-36.
WU Z H, HU P. Research on an optimization routing cost algorithm in VANET[J]. Journal of Nanchang Hangkong University (Natural Sciences), 2016, 30(2): 28-36.

[12] 崔朝阳, 孙甲琦, 徐松艳, 等. 适用于集群无人机的自组网安全分簇算法[J]. 山东大学学报(理学版), 2018, 53(7): 51-59.
CUI C Y, SUN J Q, XU S Y, et al. A secure clustering algorithm for ad hoc networks suitable for swarming UAVs[J]. Journal of Shandong University (Natural Science), 2018, 53 (7): 51-59.

[13] SOMPOLINSKY Y, ZOHAR A. Secure high-rate transaction processing in bitcoin[C]//LNCS 8975: Proceedings of the 19th International Conference on Financial Cryptography and Data Security, San Juan, Jan 26-30, 2015. Cham: Springer, 2015: 507-527.

[14] LI C, LI P, ZHUO D, et al. Scaling Nakamoto consensus to thousands of transactions per second[J]. arXiv:1805.03870, 2018.

[15] LEWENBERG Y, SOMPOLINSKY Y, ZOHAR A. Inclusive block chain protocols[C]//LNCS 8975: Proceedings of the 19th International Conference on Financial Cryptography and Data Security, San Juan, Jan 26-30, 2015. Cham: Springer, 2015: 528-547.

[16] 肖德琴, 张焕国, 胡月明, 等. 无线传感器网络攻击与防范[J]. 传感器与微系统, 2006, 25(8): 22-24.
XIAO D Q, ZHANG H G, HU Y M, et al. Attacks and defenses in wireless sensor network[J]. Transducer and Microsystem Technologies, 2006, 25(8): 22-24.

[17] YU H, GIBBONS P B, KAMINSKY M, et al. SybilLimit: a near-optimal social network defense against sybil attacks[J]. IEEE/ACM Transactions on Networking, 2010, 18(3): 885-898.

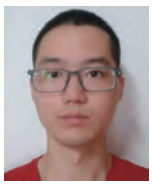
[18] 李忠诚, 黄建华, 唐瑞琮, 等. 一种基于权益代表的可扩展共识协议[J]. 应用科学学报, 2020, 38(1): 51-64.
LI Z C, HUANG J H, TANG R C, et al. A scalable consensus protocol based on stake representation[J]. Journal of Applied Sciences, 2020, 38(1): 51-64.

[19] 季钰翔, 黄建华, 王喆, 等. 基于信任度匹配的改进PBFT共识算法[J]. 计算机科学, 2021, 48(2): 303-310.
JI Y X, HUANG J H, WANG Z, et al. Improved PBFT consensus algorithm based on trust matching[J]. Computer Science, 2021, 48(2): 303-310.

[20] Intel® software guard extensions programming re-ference: Ref 329298-002[Z]. Intel, 2014: 1-6.

张文韬(1999—),男,浙江绍兴人,硕士研究生,CCF学生会员,主要研究方向为区块链。
ZHANG Wentao, born in 1999, M.S. candidate, CCF student member. His research interest is blockchain.

黄建华(1963—),男,湖南怀化人,博士,副教授,CCF区块链专业委员会执行委员,主要研究方向为计算机网络与信息安全等。
HUANG Jianhua, born in 1963, Ph.D., associate professor, executive member of CCF Blockchain Professional Committee. His research interests include computer network and information security, etc.



顾彬(1999—),男,上海人,硕士研究生,CCF学生会员,主要研究方向为区块链。
GU Bin, born in 1999, M.S. candidate, CCF student member. His research interest is blockchain.

宁宇豪(1998—),男,山西运城人,硕士研究生,CCF学生会员,主要研究方向为区块链。
NING Yuhao, born in 1998, M.S. candidate, CCF student member. His research interest is blockchain.

宫在为(2000—),女,黑龙江密山人,硕士研究生,CCF学生会员,主要研究方向为区块链。
GONG Zaiwei, born in 2000, M.S. candidate, CCF student member. Her research interest is blockchain.