基于信誉机制的车联网共识算法

李俊吉，张佳琦*，高改梅，杨莉
（太原科技大学计算机科学与技术学院，山西 太原 030024）

摘　要：针对车联网（IoV）中传统共识算法存在的通信开销大、主节点选取随意的问题，提出一种基于信誉机制的 IoV 共识算法 RHotStuff。将 IoV 中的车辆和路旁单元（RSU）作为节点组成共识网络，同时引入投票积极度、历史影响程度、信誉惩罚因子等指标来实现信誉机制，用于评估节点的信誉分数，衡量其可信程度。根据信誉分数将节点划分为主节点、从节点和候选节点。在共识开始前，仅选取信誉分数较高的部分节点作为主节点，和从节点参与共识，以降低通信开销并提高共识性能，其中主节点由信誉分数最高的节点担任，以降低主节点的可预测性。在共识完成后，信誉分数将重新计算，并据此选择下一轮参与共识的节点。此外，主节点会在 Reply 阶段将共识结果发送给其他所有节点，以同步信誉分数和区块。实验结果表明，RHotStuff 具有 $O(N)$ 的通信复杂度，并且其共识成功率相较于 C-HotStuff 提升了约 30%。当节点数量为 93 时，RHotStuff 的共识吞吐量相较于 R-PBFT 提高了 11.68%，同时其共识时延降低了 11.74%。综合来看，RHotStuff 优化了主节点选取方式，具有较低的通信开销和共识时延，同时获得了较高的共识成功率和共识吞吐量，对提升 IoV 通信效率、推动智能交通的发展具有重要意义。

关键词：信誉机制；车联网；共识算法；区块链；路旁单元
源代码链接：https://gitee.com/a75953849/0068957
中图分类号：TP393　　**文献标志码：**A　　**DOI：**10.19678/j.issn.1000-3428.0068957

Consensus Algorithm for Internet of Vehicles Based on Reputation Mechanism

LI Junji, ZHANG Jiaqi*, GAO Gaimei, YANG Li
(College of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, Shanxi, China)

【Abstract】This paper introduces RHotStuff, a reputation-based consensus algorithm for Internet of Vehicles (IoV), to address the issues of high communication overhead and arbitrary selection of master nodes in traditional consensus algorithms for IoV. This algorithm treats vehicles and Road Side Units (RSU) in the IoV as nodes, forming a consensus network. Indicators such as voting activity, historical influence, and reputation punishment factors are introduced to implement the reputation mechanism. This mechanism evaluates the reputation scores of nodes and measures their credibility. Based on their reputation scores, the nodes are divided into master, slave, and candidates. Before the consensus begins, only a subset of nodes with higher reputation scores are selected as master and slave nodes to participate in the consensus. This reduces communication overhead and improves consensus performance. The master node is selected from the node with the highest reputation score, reducing the predictability of the master node. After the consensus is reached, the reputation scores are recalculated, and the next round of nodes participating in the consensus is selected accordingly. Additionally, the master node sends the consensus result to all other nodes during the Reply phase to synchronize the reputation scores and blocks. Experimental results demonstrate that RHotStuff has an $O(N)$ communication complexity, and its consensus success rate is approximately 30% higher than that of C-HotStuff. This helps improve the consensus performance. When there are 93 nodes, the consensus throughput of RHotStuff is 11.68% higher than that of R-PBFT, whereas its consensus delay is reduced by 11.74%. Overall, RHotStuff optimizes the selection method of the master node and has low communication overhead and consensus delay while also obtaining a high consensus success rate and throughput, which is of great significance for improving the communication efficiency of the IoV and the development of intelligent transportation.

【Key words】reputation mechanism; Internet of Vehicles(IoV); consensus algorithm; blockchain; Road Side Units (RSU)

收稿日期：2023-12-06　　**修回日期：**2024-02-28
基金项目：国家自然科学基金（62272336）；山西省教学改革创新项目（J20220723）；山西省研究生科研创新项目（2023KY661）；太原科技大学研究生教育教学改革课题（JG202310）；太原科技大学纪检监察研究项目（JWYB202408）。
通信作者 E-mail：* zhangjiaqi068@163.com

0 引言

随着物联网、5G、人工智能、区块链技术与交通行业的深度融合，智能交通已成为交通技术的新业态、新模式。车联网(IoV)作为智能交通与物联网的融合产物，实现了行人、车辆、道路、路旁单元(RSU)和云端设备等异构设备之间的实时通信，共享交通环境感知数据[1]。IoV 本质上是基于车内网和车外网的集成网络[2]，它能够提供驾驶辅助、交通管理、车载娱乐等服务。

随着 IoV 技术和产业的持续发展，确保这个综合网络的安全性变得愈发关键[3-4]，特别是 IoV 的数据安全问题，可能会引发用户的隐私泄露风险。在过去，IoV 的所有数据信息都被存储在中心服务器中，而该服务器也是其他 IoV 节点的通信枢纽。然而，这种存储方式对 IoV 的信息安全构成了重大威胁。由于中心服务器的处理能力有限，随着网络规模的逐渐扩大，中心服务器的负担也在不断增加，这极易导致网络延迟，甚至造成网络崩溃。此外，由于中心服务器存在单点失效的风险，对其进行攻击可能导致整个 IoV 网络瘫痪。

IoV 对象之间的交通信息通信是通过无线通信技术传输的，攻击者可以攻破无线通信范围内的 IoV 对象，从而篡改、添加或删除这些对象中的交通信息，因此，一些对保持安全驾驶至关重要的交通信息可能会被更改，从而对整体安全性构成威胁。

作为一种全新的去中心化分布式架构，区块链融合了密码学、P2P 网络、共识算法[5-10]等技术，能够确保数据的安全性和不可篡改性。目前，区块链技术已经广泛应用于车联网[11-14]、医疗保健[15-16]、边缘计算[17-18]、电子投票[19]等领域。由于其独特的分布式特性，区块链在 IoV 中的应用有助于减轻中心服务器的负担，并降低单点攻击的风险。

区块链技术具备许多独特优势，能够有效应对当前 IoV 面临的挑战。其中，共识算法是区块链实现完整性和一致性的关键。1999 年，文献[20]提出实用拜占庭容错共识(PBFT)算法，使得拜占庭容错算法在实际系统中的应用变得可行。利用 PBFT 共识算法，可以解决 IoV 节点在共识过程中容易受到恶意节点攻击的问题。然而，PBFT 在 IoV 中的应用存在通信开销大、主节点选取随意的问题。

本文提出一种基于信誉机制的 IoV 共识算法 RHotStuff。首先，对 HotStuff[21]共识算法进行阐述，并指出其存在的问题；其次，提出 RHotStuff 设计方案，该方案包括为节点设计信誉机制和改进共识节点选取方式；最后，通过实验分析，验证 RHotStuff 的可行性和优越性，并将其与其他共识算法进行对比。本文主要贡献如下：

1)将车辆和 RSU 作为共识节点组成 IoV 共识网络，并引入信誉机制对节点的共识行为进行评分。通过信誉分数的计算和评估，可以有效地衡量节点的可信度和贡献程度，有助于提高共识过程的可靠性和效率。

2)根据信誉分数将共识节点分为主节点、从节点和候选节点。选取信誉分数较高的 1/3 节点作为从节点，其中分数最高的节点担任主节点。通过减少参与共识的节点数量提高共识效率。

3)改进共识过程，主节点将共识结果发送给其他节点，其他节点根据结果计算信誉分数以及同步区块，确保信誉分数和共识结果的一致性。

1 相关工作

许多研究者提出基于 PBFT 的 IoV 方案，旨在实现防篡改的去中心化的安全存储。文献[22]通过引入信誉机制改进了 PBFT，称为 SG-PBFT。在 SG-PBFT 中，每 50 次投票会计算一次信誉分数，信誉分数较低的节点会被移到 IoV 共识节点集合的末尾。该算法选取一半信誉分数较高的节点参与共识，从而提高系统共识的效率。文献[23]使用逻辑回归方法计算信誉值来改进 PBFT，并将其应用于 IoV。该算法选取信誉分数较高的前 1/3 的 IoV 节点(包括车、路、边缘设备等)参与共识，并选取信誉分数最高的节点作为主节点发起区块生成请求。文献[24]提出了一种基于时间序列和 Gossip 协议的高效拜占庭一致性算法 BCA-TG，有效克服了 IoV 在一致性和认证方面的缺陷。BCA-TG 利用时间序列提高节点间通信和达成共识的效率，并利用 Gossip 协议使得 IoV 中任意 2 个网络节点可以在一个通信周期内获得相同的信息。文献[25]提出了一个车联网激励机制 SmartCoin。SmartCoin 由车辆、RSU 和网络监控组成，车辆和 RSU 负责生成、接收和评估 IoV 信息，而网络监控则构建私有链区块网络，挖掘区块并将其附加到区块链中。在私有链网络中，网络监控节点通过轮转法选取下一个区块提议节点，由全网节点进行验证，如果超过 3/4 的节点通过验证，则该区块达成共识，并为相应的信息供应者提供奖励。

然而，当前 IoV 共识算法的通信开销仍然居高不下，导致共识效率低下。为了解决这一问题，研究者提出了一些线性复杂度的共识算法，以降低通信

开销并提高共识效率。文献[26]提出一种可扩展的拜占庭容错共识算法 SBFT,它使用门限签名技术实现收集器,将通信复杂度降低到 $O(N)$,并在拜占庭节点无作恶行为的情况下减少一轮投票的收集过程,提高共识算法的性能。然而,SBFT 的视图切换没有实现线性复杂度。为了解决这个问题,文献[21]提出了一种三阶段投票的拜占庭容错共识算法 HotStuff。在 HotStuff 中,引入门限签名技术实现收集器,同时设计了一个三阶段的投票流程来实现每个视图轮换主节点的方案,将视图切换复杂度降低到 $O(N)$。但是,HotStuff 的主节点选取随意,仅仅使用轮转法来选取主节点,使得主节点很容易被预测。如果主节点作恶或被恶意节点攻击,将导致共识性能下降。因此,为了更好地将 HotStuff 应用于 IoV,需要对其进行改进。

2 HotStuff 共识算法

HotStuff 能够容忍不超过 1/3 的拜占庭节点。在 HotStuff 中,共识节点被分为主节点和从节点,总节点数量为 $N=3f+1$,其中 f 表示 HotStuff 能够容忍的最大拜占庭节点数量。每个共识过程在一个视图内完成,每个视图具有唯一且单调递增的视图编号。在每个视图中,主节点负责主导共识过程,完成共识后将进入下一个视图。HotStuff 共识过程如图 1 所示,它由 New-View 阶段、Prepare 阶段、PreCommit 阶段、Commit 阶段和 Decide 阶段等 5 个阶段组成。其中,Prepare 阶段、PreCommit 阶段和 Commit 阶段是 HotStuff 中的三轮投票阶段。当主节点收到足够的有效投票后会构建决议证书(QC),构建成功后将其广播到所有节点。

1)New-View 阶段。从节点会向其认为的主节点发送 New-View 消息,同时附带自己的 PrepareQC 消息,以启动新的视图进行共识。

2)Prepare 阶段。主节点会等待接收 $N-f$ 个 New-View 消息,并对这些消息中携带的 PrepareQC 进行对比,选择具有最高视图编号的 PrepareQC 作为 HighQC。随后,主节点使用 HighQC 作为证据生成提案,并将提案广播至所有节点。从节点在收到主节点的提案后,会验证该提案是否基于 HighQC 生成并且是否符合安全性规则。如果验证通过,从节点会向主节点发送自己的投票。

3)PreCommit 阶段。主节点会收集 $N-f$ 个投票,并使用门限签名将它们聚合在一起,生成 PrepareQC。随后,主节点会将该 PrepareQC 广播至所有节点。从节点收到该 PrepareQC 后,会对其进行验证。如果验证通过,从节点会向主节点发送自己的投票。

4)Commit 阶段。主节点收集 $N-f$ 个投票,并将它们聚合生成 PreCommitQC,然后将其广播至整个网络。从节点在接收到 PreCommitQC 后,会对其进行验证。如果验证通过,从节点会将该 PreCommitQC 锁定,随后发送自己的投票。

5)Decide 阶段。主节点收集到 $N-f$ 个投票后,会通过门限签名将它们聚合生成 CommitQC,并广播至所有节点。从节点在收到有效的 CommitQC 后,可以在新提议的区块上执行相应的命令。随后,将共识完成的回应发送给客户端。

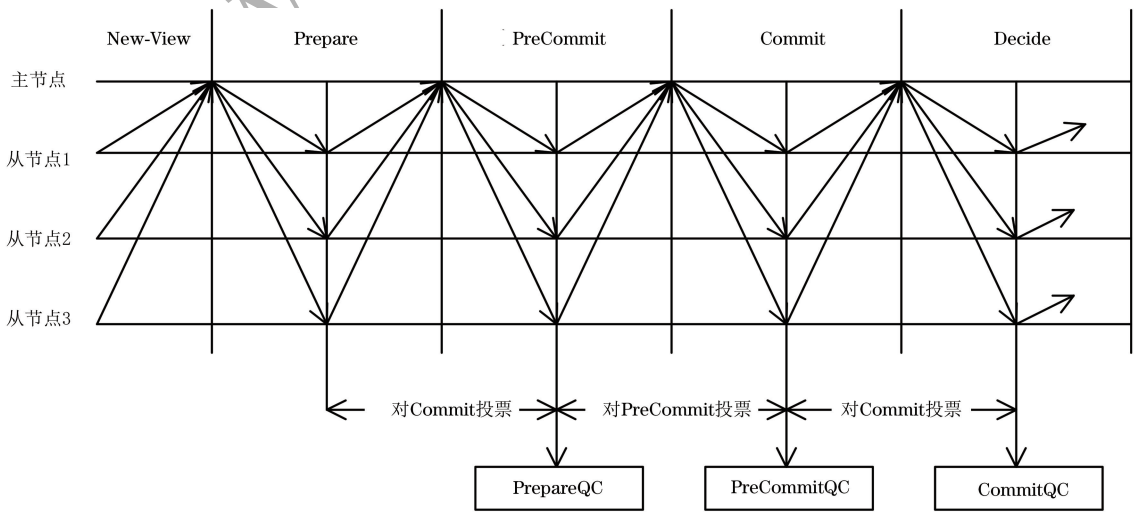

图 1 HotStuff 共识过程

Fig. 1 HotStuff consensus process

在 HotStuff 中,三轮投票过程基本一致,因此可以采用流水线的方式提升共识效率,这种改进方案称为链式 HotStuff(C-HotStuff),如图 2 所示。通过流水线的方式收集投票,可以减少重复工作,从

而提高共识效率。

在 Prepare 阶段，投票是由当前视图对应的主节点 Leader 1 收集的，当收集到足够数量的投票后，Leader 1 会生成一个 PrepareQC。然后，Leader 1 将该 PrepareQC 发送给下一个视图的主节点 Leader 2。在收到 PrepareQC 后，Leader 2 会基于该 QC 开始新的 Prepare 阶段，这个阶段同时也是 Leader 1 的 PreCommit 阶段。类似地，Leader 2 在完成其 Prepare 阶段后，会生成一个新的 PrepareQC，并将其发送给下一个视图的主节点 Leader 3。Leader 3 会开始自己的 Prepare 阶段，这个阶段对于 Leader 1 是 Commit 阶段，对于 Leader 2 是 PreCommit 阶段。以此类推，这种流水线式的投票过程可以减少重复工作，提高共识效率。在 C-HotStuff 中，一轮投票可以同时生成 3 种 QC，从而减少了算法的通信开销。

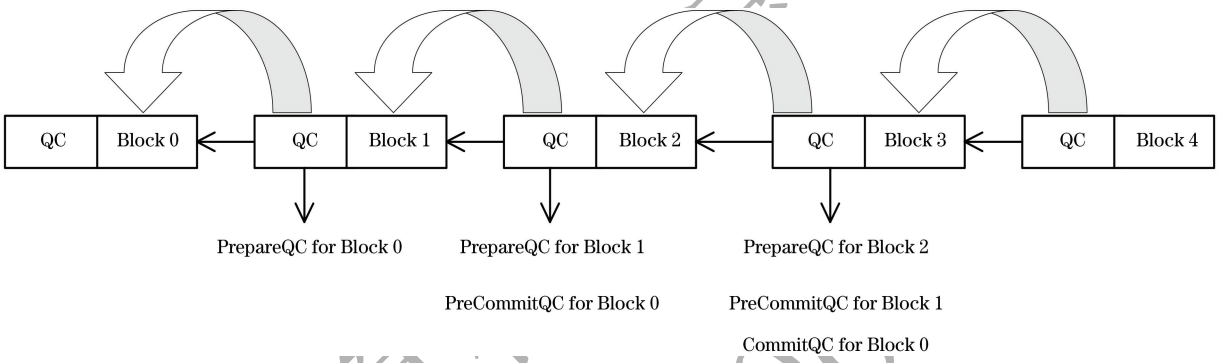

图 2 HotStuff 的流水线方案

Fig. 2 Pipeline scheme of HotStuff

如图 2 所示，Block 1 的 QC 实际上是 Block 0 的 PrepareQC。Block 2 的 QC 是 Block 1 的 PrepareQC，同时也是 Block 0 的 PreCommitQC。同样地，Block 3 的 QC 是 Block 2 的 PrepareQC，同时也是 Block 1 的 PreCommitQC 以及 Block 0 的 CommitQC。

C-HotStuff 使用主节点来驱动共识过程。然而，C-HotStuff 的主节点选取随意，按照主节点的编号顺序依次担任主节点，所有节点都可以预测到下一任主节点，从而有针对性地发动攻击。此外，拜占庭节点担任主节点的概率与诚实节点相同。如果主节点出现故障或作恶行为，共识过程可能会受到干扰，甚至无法成功达成共识。在此情况下，需要进行视图切换更换主节点，从而使得共识算法的性能下降。

为了解决这个问题，本文使用信誉机制对 C-HotStuff 进行改进。通过引入信誉机制，选取具有较高信誉分数的部分节点参与共识，降低通信开销，从而提高共识算法的性能。主节点由信誉分数最高的节点担任，由于信誉分数根据节点的表现来计算，因此任何节点都无法预测。

3 RHotStuff 共识算法

3.1 IoV 共识模型

IoV 共识模型如图 3 所示，IoV 共识节点由车辆和 RSU 共同组成。车辆配备了车载传感器、计算机和通信设备，这些设备用于数据收集、处理和分享。借助车载设备，车辆可以感知实际路况交通的相关事件，并向网络中的其他节点发送消息。RSU 负责协调车辆之间的通信，使不在彼此范围内的车辆能够相互传递消息，并将这些消息广播到 IoV 网络中。IoV 共识节点之间通过车车通信(V2V)、车与基础设施通信(V2I)、基础设施与基础设施通信(I2I)进行交互。在 IoV 的不可信环境中，共识算法至关重要，它能够确保网络中大多数节点对某个状态或决策达成一致，从而极大地提高网络的可靠性和安全性[27]。

由于相距较远的车辆无法直接进行通信，因此需要借助 RSU 来进行信息传递。RSU 将 IoV 中的车辆联系起来，共同构成一个完整的通信网络。在该网络中，采用 RHotStuff 共识算法构建一个区块链网络。RHotStuff 引入信誉机制来评估每个节点的行为。根据节点在共识过程中的表现，计算其信誉分数。随后，选择部分高信誉分数的节点参与共识。每次共识完成后，信誉分数将重新计算，并据此选择下一轮参与共识的节点。

3.2 信誉机制

在 RHotStuff 中，节点总数为 $3N$，实际参与共识的节点数为 $N=3f+1$，其中 f 表示共识节点中最大能够容忍的拜占庭节点数量。所有节点共同维护节点的信誉分数信息。为了更好地描述信誉机制，引入 4 个关键指标，即投票积极度、共识成功率、历史影响程度和信誉惩罚因子。

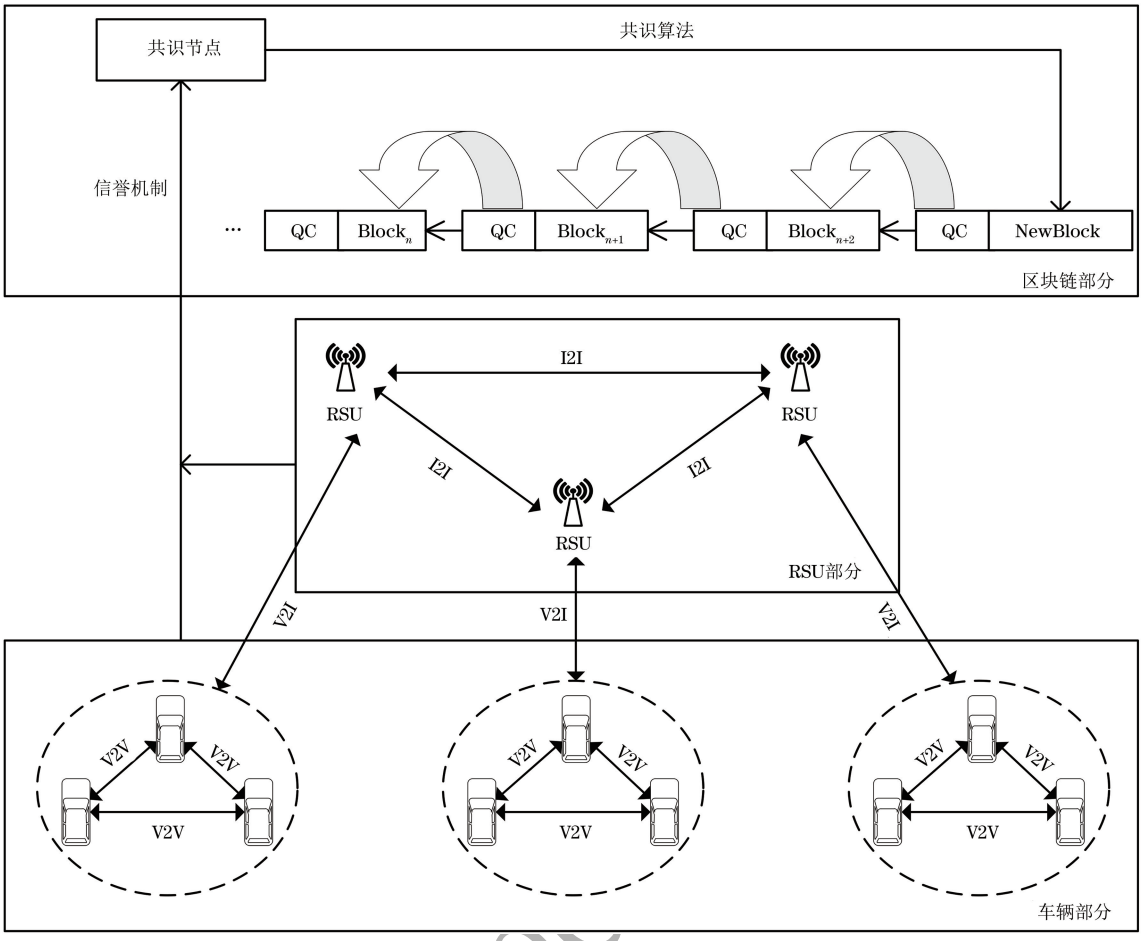

图 3 IoV 共识模型结构

Fig. 3 IoV consensus model structure

1)投票积极度表示共识节点参与投票的活跃程度。节点积极参与投票并正确投票的频率越高,其信誉分数就越高。投票积极度反映了节点对共识网络的贡献度和参与度,其计算公式为:

$$p(i)=\begin{cases}\mathrm{e}^{-\frac{1}{s_i+v_i}}, & s_i+v_i=0\\ 0, & s_i+v_i\neq 0\end{cases} \tag{1}$$

式中:v_i 表示节点投票被打包进 QC 的次数;s_i 表示节点投票被打包进 QC 且该 QC 所在区块被提交的次数。当主节点收到 $2f+1$ 个来自从节点的正确投票后,开始构建 QC,这些参与投票的 $2f+1$ 个从节点的 v_i 计数器加 1,这意味着该节点成功参与了共识过程,其投票被采纳并打包进了 QC。当该 QC 所在的区块被成功提交后,以上 $2f+1$ 个投票节点的 s_i 计数器加 1,这意味着节点的投票不仅被采纳,而且对区块链造成了影响(区块提交)。投票积极度的值在 0～1 之间,v_i 和 s_i 越大,则投票积极度越高。

2)共识成功率表示节点成功参与共识并达成一致的频率。使用 s_i 和 v_i 的比值来表示共识成功率,初始时设置共识成功率为 1,计算公式为:

$$y(i)=\begin{cases}\dfrac{s_i}{v_i}, & v_i\neq 0\\ 1, & v_i=0\end{cases} \tag{2}$$

3)历史影响程度 $\omega(i)$ 表示节点在过去的行为和决策对当前信誉分数的影响程度,计算公式为:

$$\omega(i)=c\sqrt{y(i)} \tag{3}$$

式中:c 为历史影响因子,如果节点 i 上一次的投票成功并被打包进 QC,则 $c=1$,否则 $c=1/2$;$y(i)$ 为共识成功率。

4)信誉惩罚因子表示为 $m(i)$。如果节点存在作恶行为,其信誉分数将会受到惩罚,降低其信誉分数增长速度。$m(i)$ 计算公式为:

$$m(i)=1-\frac{1}{1+\mathrm{e}^{3-x_i}} \tag{4}$$

式中:x_i 指节点 i 存在作恶行为的次数。当节点 i 的作恶行为次数较少时,$m(i)$ 接近 1,反之 $m(i)$ 接近 0。有时诚实的节点可能由于网络延迟等原因被误认为是恶意节点,如果将其误判为恶意节点,可能会影响共识算法的性能。因此,该信誉机制引入信誉惩罚因子,当节点作恶次数小于 3 时,$m(i)$ 接

近 1，反之，迅速下降到接近 0。这样既能对作恶行为进行惩罚，又能减少因对诚实节点的误判而造成的损失。当作恶次数达到 5 次时，将该节点视为恶意节点，禁止其参与共识过程。

随后，引入节点信誉分数计算公式：

$$R_n^i = R_{n-1}^i + k\left[\frac{m(i)\cdot\omega(i)\cdot p(i)}{e^{R_{n-1}^i}} + \theta\frac{\sum_{j=1}^{3N} R_{n-1}^j}{3N}\right] \tag{5}$$

式中：R_n^i 表示第 n 轮投票节点 i 的信誉分数；$3N$ 表示节点总数；k 表示是否参与本轮投票，参与则为 1，否则为 0。

为了防止主节点信誉分数持续增加并连任主节点，信誉机制引入 θ，当节点 i 为主节点时，$\theta=-0.3$，否则 $\theta=0$。当主节点完成一轮共识后，会扣除其 θ 倍的平均信誉分数，以确保该节点无法再次担任主节点。当主节点被发现存在异常行为时，会扣除其 0.5 倍的平均信誉分数，使其信誉分数快速下降。

3.3 共识节点选取

在 RHotStuff 中，节点总数为 $3N$，包括主节点、从节点和候选节点 3 种。每轮共识开始时，会从 $3N$ 个节点中选取信誉分数较高的 N 个节点参与共识，其中信誉分数最高的节点被设为主节点，其余 $N-1$ 个节点为从节点，剩余未被选中的节点作为候选节点，不参与本轮共识。

为了确保所有节点均有机会参与共识，节点的信誉分数初始值设为 1。如果初始信誉分数设为 0，第一轮未参加共识的节点将一直处于候选节点的角色，只有当共识节点出现多次作恶行为被踢出后，这些候选节点才有机会参与共识。

主节点负责收集 New-View 消息，基于这些消息生成新的提案，并将其发送到所有的从节点。接着，主节点等待从节点的投票。一旦收到 $2f+1$ 个有效的投票，主节点将投票结果打包成 QC，并计算节点的信誉分数，信誉分数计算如算法 1 所示。随后，主节点将 QC 与最终信誉分数一同发送给所有节点。最后，主节点还会向下一任主节点发送 New-View 消息，以便其做好准备承接下一轮共识的主导权。下一任主节点为本轮信誉分数最高的节点。从节点等待主节点的提案，并在收到提案后验证其安全性以及信誉分数计算的正确性。如果验证成功，从节点将向主节点发送自己的投票。同时，从节点会判断该提案的 QC 是否可以作为祖先区块的 PreCommitQC、CommitQC。如果该 QC 可以作为祖先区块的 CommitQC，从节点将提交该提案所对应的祖父区块。

算法 1 信誉分数计算算法

输入 nodes，votes//节点数组和最早接收到的 2f+//1 个有效的投票

输出 scores[] //信誉分数数组

```
1. if(votes. size<2f+1):
2.     return "投票数量不足，无法计算";
3. scores[] ← []; //初始化空的信誉分数数组
4. for i from 0 to 3N:
5.     c ← 1/2;
6.     if(nodes[i]. isQC = true):
7.         c ← 1; //节点 i 上一次的投票成功并被打包
//进 QC
8.     θ ← 0;
9.     if(节点 i 为主节点):
10.        θ ← -0.3;
11.    k ← 0;
12.    if(votes. contains(i)):
13.        k ← 1; //节点 i 本轮被打包进 QC
14.        nodes[i]. isQC ← true;
15.    scores[i] ← 式(5) //使用式(5)计算信誉分数并
//存放在数组中
16. return scores;
```

3.4 RHotStuff 共识过程

RHotStuff 在 C-HotStuff 的基础上进行改进，其共识过程分为 3 个阶段，即 New-View 阶段、Generic 阶段和 Reply 阶段。RHotStuff 的共识过程如图 4 所示。

1）New-View 阶段。从节点根据上一轮的信誉分数选取本轮的主节点，并向其发送 New-View 消息，开启新一轮共识。

2）Generic 阶段。主节点根据收集到的 New-View 消息生成提案，并将提案发送至所有的从节点。从节点收到提案后对其进行验证，如果验证通过，则向主节点发送投票。

3）Reply阶段。主节点收到从节点的投票后，

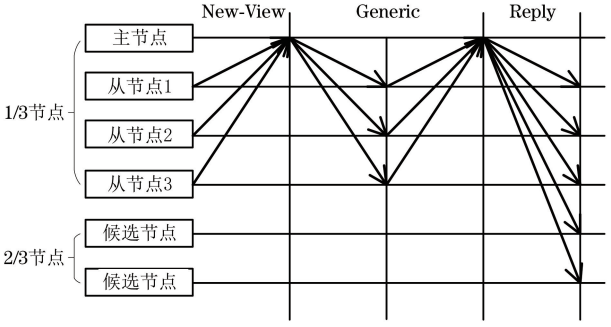

图 4 RHotStuff 共识过程

Fig. 4 RHotStuff consensus process

会验证投票的正确性以及数量是否达到阈值。如果验证通过，主节点会将投票结果打包成 QC，并发送给所有的节点(包括候选节点)。这样做是为了使其他节点能够计算所有节点的信誉分数，并根据该主节点的提案生成区块。

4 实验验证

本文在 64 位 Windows 10 操作系统上进行仿真实验。该系统的 CPU 为 i5-6200U，内存为 8 GB。采用 Java 语言实现 C-HotStuff[21]、R-PBFT[23]以及 RHotStuff，并通过不同的端口进行仿真测试，以模拟车联网环境中的多节点通信。

1)实验 1。

将共识节点数量设置为 48，随机选取 15 个节点作为拜占庭节点。设定共识节点的初始信誉分数为 1。为了模拟拜占庭节点的作恶行为，实验中将拜占庭节点的作恶概率设为 50%。选择首个被踢出系统的节点作为拜占庭节点代表，并随机选择一个诚实节点作为诚实节点代表。实验记录这 2 个节点的信誉分数变化，如图 5 所示。

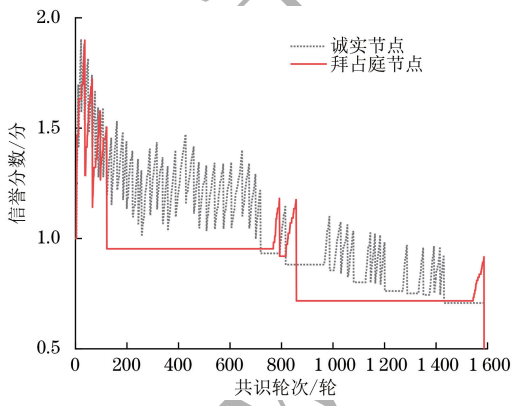

图 5 诚实节点和拜占庭节点的信誉分数变化

Fig. 5 Changes in reputation scores of honest node and Byzantine node

在图 5 中，每个极大值点表示该节点被选为主节点。成功当选主节点后，该节点会扣除 0.3 倍的平均信誉分数，导致该节点的信誉分数迅速下降，从而避免了出现主节点连任的情况。拜占庭节点因作恶被发现后会受到信誉惩罚，并减缓其信誉增长速度。因此，拜占庭节点担任主节点的间隔逐渐加大。

在 150 轮共识之前，拜占庭节点担任主节点的间隔较小，这是为了避免误判诚实节点为恶意节点。在作恶次数较少时，其增长速度与正常情况相差无几。但由于拜占庭节点作恶、经常性的不投票或者投错误票，导致其投票无法被打包进 QC，进而无法累积信誉分数。因此，拜占庭节点担任主节点的次数少于诚实节点。

在 150～750 轮时，拜占庭节点的信誉分数保持不变。这是由于拜占庭节点多次作恶后，其信誉分数降低到平均值以下，导致其很长时间后才有机会参与共识。诚实节点也存在一段时间信誉分数不变的情况，这是因为每次共识会选取 1/3 的节点参与，所以每个节点都会存在不参与共识的时期。最终，在 1 500 轮之后，拜占庭节点作恶次数达到 5 次，该节点被踢出系统。

该实验表明，当拜占庭节点存在作恶行为时，它被选为主节点的次数明显少于诚实节点。此外，在出现多次作恶行为之后，拜占庭节点的信誉分数恢复速度也变得相对较慢，多次作恶后被踢出系统。

2)实验 2。

该实验的节点设置与实验一相同，当拜占庭节点作恶次数达到 5 次时，该节点被踢出系统。记录第 8 000 轮共识中各个节点的信誉分数以及担任主节点的次数，如图 6 所示。在图 6 中，信誉分数为 0 和担任主节点次数为 0 表示该节点作恶次数超过 5 次，已被踢出共识。可以看出，在第 8 000 轮 15 个拜占庭节点均已被踢出共识。相对地，诚实节点信誉分数和担任主节点的次数都比较接近，这表明每个诚实节点被选为主节点的概率大致相等。此外，主节点的选取依据是信誉分数，并且该分数与节点行为紧密相关，从而降低了主节点的可预测性。

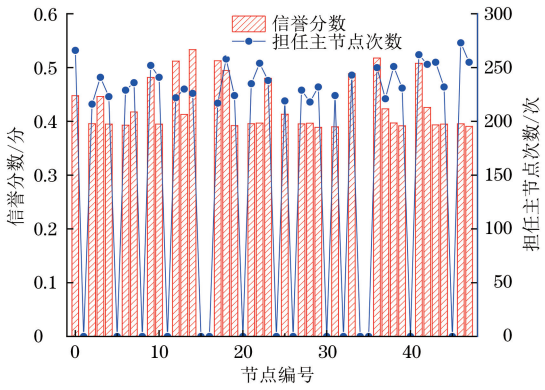

图 6 节点的信誉分数与担任主节点的次数

Fig. 6 The reputation score of the nodes and the number of times as the leader

3)实验 3。

该实验与 C-HotStuff 进行对比。在实验中，节点数量设置为 48，共识节点的初始信誉分数设为 1。当拜占庭节点作恶达到 5 次时，该节点将被踢出系统。由于 C-HotStuff 没有信誉机制，因此将 C-HotStuff 节点数量设置为 16，使这 2 种算法参与共识的节点数量相同。实验进行 1 000～8 000 轮次，记录 2 种算法下区块成功提交的次数以及区块提交

率(区块成功提交的概率)。区块提交数量对比如图 7 所示。

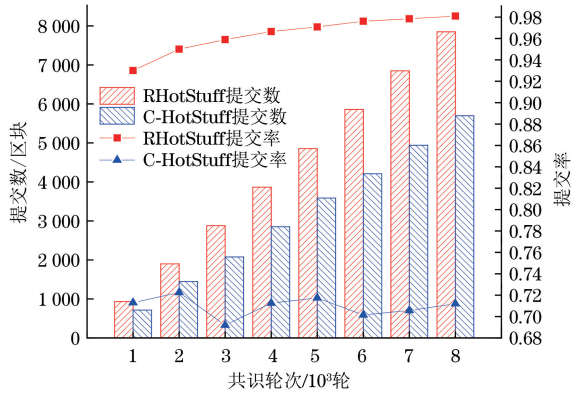

图 7 区块提交数量对比

Fig. 7 Comparison of block submission quantity

RHotStuff 的区块成功提交数量以及区块提交率明显大于 C-HotStuff。这是因为 RHotStuff 引入了信誉机制,如果在共识过程中发现拜占庭节点有作恶行为,就会对其进行惩罚,导致其担任主节点的次数减少。随着共识轮次的增加,拜占庭节点的数量逐渐减少,从而降低出错概率。然而,C-HotStuff 并未引入信誉机制,即使发现作恶行为也不会进行惩罚,因此 C-HotStuff 的区块提交率大约维持在 70%的水平。相比之下,当共识轮次达到 8 000 时,RHotStuff 的区块提交率高达 98%。总体来看,RHotStuff 的共识成功率相较于 C-HotStuff 提升了约 30%。

4)实验 4。

该实验对 RHotStuff、C-HotStuff 和 R-PBFT 的通信开销进行了比较。设节点数量为 N,计算 3 种共识算法在一轮共识后所需的通信次数,得到通信开销对比如表 1 所示。

表 1 通信开销对比

Table 1 Comparison of communication overhead

共识算法	通信开销
R-PBFT	$2N^2/9-N/3+1$
C-HotStuff	$4N-4$
RHotStuff	$2N-4$

分别设置节点数量为 4～58,得到各共识算法通信开销对比图,如图 8 所示。从图 8 中可以看出,当节点数量较少时,3 种算法的通信开销相差不大。随着节点数量的增加,RHotStuff 的通信开销增长相对缓慢,逐渐小于 C-HotStuff 和 R-PBFT。这主要是因为 RHotStuff 在通信开销方面的复杂度为 $O(N)$,与 C-HotStuff 相同,但 RHotStuff 只选择了 1/3 的节点参与共识,因此其通信开销实际上比 C-HotStuff 更低。相比之下,R-PBFT 的通信开销复杂度为 $O(N^2)$,这意味着当节点数量增加时,R-PBFT 的通信开销会迅速增加。该实验表明,RHotStuff 具有较低的通信开销。

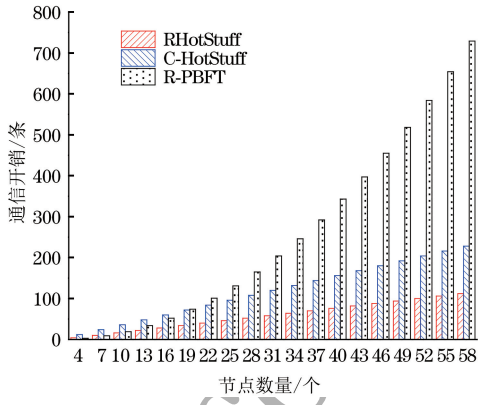

图 8 各共识算法的通信开销对比

Fig. 8 Comparison of communication overhead among consensus algorithms

5)实验 5。

在该实验中,设置不同的节点数量,分别为 48、57、66、75、84、93。客户端并行发送 5 000 条交易,每 100 条交易被打包成一个区块。记录不同节点数量下各共识算法的共识吞吐量和共识时延,并将结果与 R-PBFT 进行比较。在不同节点的情况下,共识吞吐量和共识时延对比如图 9 所示。

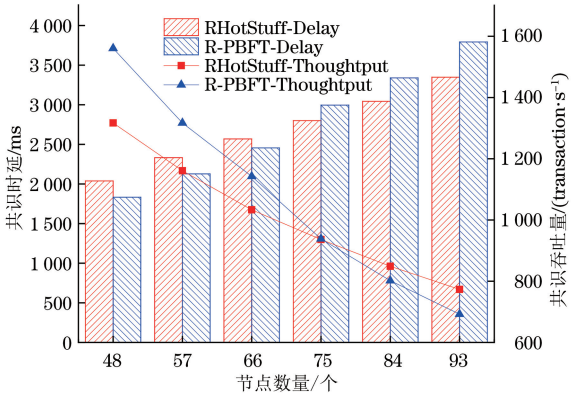

图 9 不同节点数量下的共识吞吐量和共识时延对比

Fig. 9 Comparison of consensus throughput and consensus delay under different numbers of nodes

从图 9 中可以看出,当节点数量少于 75 时,RHotStuff 的共识吞吐量相较于 R-PBFT 偏低,同时其共识时延也略大。这主要是因为 RHotStuff 采用了门限签名来实现共识算法,导致了一定的时间消耗。因此,在共识节点数量较少的情况下,R-PBFT 在共识吞吐量和共识时延方面仍表现出良好的性能。然而,随着共识节点数量的增加,RHotStuff 的性能开始逐渐超越 R-PBFT。这是因为 RHotStuff 具有线性的通信复杂度,随着共识节

点数量的增加，其性能损失较慢。相比之下，R-PBFT 具有较高的通信开销，其性能迅速下降。当共识节点数量达到 93 时，RHotStuff 的共识吞吐量相较于 R-PBFT 提高了 11.68%，同时其共识时延降低了 11.74%。该实验表明，RHotStuff 具有较高的共识吞吐量和较低的共识时延。

6)实验 6。

在该实验中，设置节点数量为 93，客户端分别并行发送 1 000、2 000、3 000、4 000、5 000、6 000、7 000、8 000、9 000、10 000、11 000 条交易，每 100 条交易被打包成一个区块。记录不同交易数量下 RHotStuff 和 R-PBFT 的共识吞吐量和共识时延，如图 10 所示。

从图 10 中可以看出，随着交易数量的增加，RHotStuff 和 R-PBFT 的共识吞吐量和共识时延均呈现上升趋势。当交易数量达到一定的阈值后，2 种算法的共识吞吐量增长速度有所减缓。具体而言，RHotStuff 的共识吞吐量在交易数量为 8 000 后增长变得缓慢，而 R-PBFT 的共识吞吐量在交易数量为 3 000 后几乎趋于稳定，这表明交易数量太大，已经接近或达到系统处理瓶颈。相较于 R-PBFT，RHotStuff 在交易数量为 8 000 后才达到瓶颈，这表明 RHotStuff 的处理开销相对较低。随着共识吞吐量的增加，共识时延也在不断增加，这 2 个性能不能同时达到最优，因此需要选取一个合适的平衡点。本文将交易数量设置为 5 000，此时共识吞吐量相对较高且共识时延也适中。

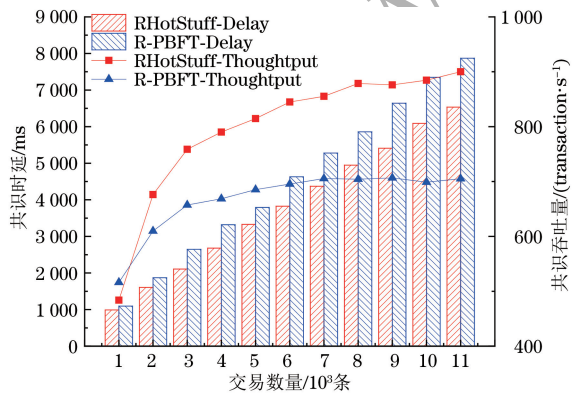

图 10 不同交易数量下的共识吞吐量和共识时延对比

Fig. 10 Comparison of consensus throughput and consensus delay under different transaction amounts

5 结束语

针对传统 IoV 共识算法中存在的通信开销大、主节点选取随意的问题，本文提出了一种 IoV 环境下的共识算法 RHotStuff。该算法在 C-HotStuff 的基础上进行改进，引入信誉机制，并选取 1/3 的节点参与共识，以降低通信开销，其余节点作为候选节点，不参与当前轮次的共识。主节点由信誉分数最高的节点担任，这使得在新一轮共识开始前主节点才能被确定，从而降低了主节点的可预测性。此外，RHotStuff 能够对作恶的拜占庭节点进行有效惩罚，使其信誉分数快速下降，并减缓其信誉分数的增长速度。实验结果表明，RHotStuff 具有较低的通信开销和共识时延以及较高的共识成功率和共识吞吐量。通过 RHotStuff，车辆能够迅速交换信息并达成共识，有效提升通信效率，从而推动城市交通的智能化和高效化发展。但 RHotStuff 仍有不足之处，后续将进一步提高 RHotStuff 的可扩展性。

参考文献

[1] TAN H W, CHUNG I. RSU-aided remote V2V message dissemination employing secure group association for UAV-assisted VANETs[J]. Electronics, 2021, 10(5): 548.

[2] 况博裕，李雨泽，顾芳铭，等. 车联网安全研究综述：威胁、对策与未来展望[J]. 计算机研究与发展，2023，60(10)：2304-2321.
KUANG B Y, LI Y Z, GU F M, et al. Review of Internet of Vehicle security research: threats, countermeasures, and future prospects[J]. Journal of Computer Research and Development, 2023, 60(10): 2304-2321. (in Chinese)

[3] 李兴华，钟成，陈颖，等. 车联网安全综述[J]. 信息安全学报，2019，4(3)：17-33.
LI X H, ZHONG C, CHEN Y, et al. Survey of Internet of Vehicles security[J]. Journal of Cyber Security, 2019, 4(3): 17-33. (in Chinese)

[4] 刘媛妮，李奕，陈山枝. 基于区块链的车联网安全综述[J]. 中国科学：信息科学，2023，53(5)：841-877.
LIU Y N, LI Y, CHEN S Z. A survey of Internet of vehicles/vehicle to everything security based on blockchain[J]. Scientia Sinica (Informationis), 2023, 53(5): 841-877. (in Chinese)

[5] 苏瑞国，阳建，秦继伟，等. 基于物联网区块链的轻量级共识算法研究[J]. 计算机工程，2023，49(2)：175-180.
SU R G, YANG J, QIN J W, et al. Research on lightweight consensus algorithm based on IoT blockchain[J]. Computer Engineering, 2023, 49(2): 175-180. (in Chinese)

[6] 刘泽坤，王峰，贾海蓉. 结合动态信用机制的 PBFT 算法优化方案[J]. 计算机工程，2023，49(2)：191-198.
LIU Z K, WANG F, JIA H R. Optimization scheme of PBFT algorithm combining dynamic credit mechanism[J]. Computer Engineering, 2023, 49(2): 191-198. (in Chinese)

[7] 吴昕怡，沈航，白光伟，等. 面向车联网切片的区块链辅助资源交易方法[J]. 小型微型计算机系统，2024，45(1)：160-167.
WU X Y, SHEN H, BAI G W, et al. Blockchain-assisted resource transaction for 5G sliced vehicular networks[J]. Journal of Chinese Computer Systems, 2024, 45(1): 160-167. (in Chinese)

[8] CONG Y Z, DU H B, LIU B B, et al. Distributed constrained finite-time consensus algorithm for second-order multi-agent systems[J]. Information Sciences, 2023, 626: 773-786.

[9] MENG X W, LIU Q S. A consensus algorithm based on multi-agent system with state noise and gradient disturbance for distributed convex optimization[J]. Neurocomputing, 2023, 519: 148-157.

[10] WANG X, DUAN S S, CLAVIN J, et al. BFT in

blockchains: from protocols to use cases [J]. ACM Computing Surveys, 2022, 54(10): 1-37.
[11] 刘雪娇, 曹天聪, 夏莹杰. 区块链架构下高效的车联网跨域数据安全共享研究[J]. 通信学报, 2023, 44(3): 186-197.
LIU X J, CAO T C, XIA Y J. Research on efficient and secure cross-domain data sharing of IoV under blockchain architecture[J]. Journal on Communications, 2023, 44(3): 186-197. (in Chinese)
[12] YAN K, ZENG P, WANG K, et al. Reputation consensus-based scheme for information sharing in Internet of Vehicles[J]. IEEE Transactions on Vehicular Technology, 2023, 72(10): 13631-13636.
[13] TU S S, YU H Y, BADSHAH A, et al. Secure Internet of Vehicles (IoV) with decentralized consensus blockchain mechanism[J]. IEEE Transactions on Vehicular Technology, 2023, 72(9): 11227-11236.
[14] WANG Y S, YUAN L M, JIAO W H, et al. A fast and secured vehicle-to-vehicle energy trading based on blockchain consensus in the Internet of electric vehicles [J]. IEEE Transactions on Vehicular Technology, 2023, 72(6): 7827-7843.
[15] HEGDE P, MADDIKUNTA P K R. Secure PBFT consensus-based lightweight blockchain for healthcare application[J]. Applied Sciences, 2023, 13(6): 3757.
[16] 温亚兰, 陈美娟. 融合联邦学习与区块链的医疗数据共享方案[J]. 计算机工程, 2022, 48(5): 145-153, 161.
WEN Y L, CHEN M J. Medical data sharing scheme combined with federal learning and blockchain[J]. Computer Engineering, 2022, 48(5): 145-153, 161. (in Chinese)
[17] HUANG Y D, ZHANG J R, DUAN J, et al. Resource allocation and consensus on edge blockchain in pervasive edge computing environments[C]//Proceedings of the 39th IEEE International Conference on Distributed Computing Systems. Washington D. C., USA: IEEE Press, 2019: 1476-1486.
[18] ZHANG Y F, GAN Y Z, LI C L, et al. Primary node selection based on node reputation evaluation for PBFT in UAV-assisted MEC environment [J]. Wireless Networks, 2023, 29(8): 3515-3539.
[19] BAHRI J, BOROJENI H R S. Electronic voting through DE-PBFT consensus and DAG data structure[C]//Proceedings of the 9th International Conference on Computer and Knowledge Engineering. Washington D. C., USA: IEEE Press, 2019: 391-396.
[20] CASTRO M. Practical Byzantine fault tolerance[EB/OL]. [2023-09-05]. https://www.scs.stanford.edu/nyu/03sp/sched/bfs.pdf.
[21] YIN M F, MALKHI D, REITER M K, et al. HotStuff[C]// Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing. New York, USA: ACM Press, 2019: 347-356.
[22] XU G Q, BAI H P, XING J, et al. SG-PBFT: a secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of Vehicles[J]. Journal of Parallel and Distributed Computing, 2022, 164: 1-11.
[23] KUMAR A, VISHWAKARMA L, DAS D. R-PBFT: a secure and intelligent consensus algorithm for Internet of Vehicles[J]. Vehicular Communications, 2023, 41: 100609.
[24] HU W, HU Y W, YAO W H, et al. A blockchain-based Byzantine consensus algorithm for information authentication of the Internet of Vehicles[J]. IEEE Access, 2019, 7: 139703-139711.
[25] VISHWAKARMA L, DAS D. SmartCoin: a novel incentive mechanism for vehicles in intelligent transportation system based on consortium blockchain [J]. Vehicular Communications, 2022, 33: 100429.
[26] GOLAN GUETA G, ABRAHAM I, GROSSMAN S, et al. SBFT: a scalable and decentralized trust infrastructure[EB/OL]. [2023-09-05]. https://arxiv.org/abs/1804.01626.
[27] 杨磊, 龙伟. 基于区块链的车联网信任机制[J]. 计算机应用研究, 2023, 40(7): 1957-1963.
YANG L, LONG W. Blockchain-based trust mechanism in VANET[J]. Application Research of Computers, 2023, 40(7): 1957-1963. (in Chinese)

编辑 吴云芳