基于信任模型的 PBFT 共识机制研究

赵立瑾[1] 王新胜[1] 夏纯中[2]

[1](江苏大学计算机科学与通信工程学院 江苏 镇江 212013)

[2](江苏大学信息化中心 江苏 镇江 212000)

摘 要 针对实用拜占庭容错(Practical Byzantine Fault Tolerance,PBFT)共识机制无法预评估并预先限制恶意节点的问题,提出基于信任模型的拜占庭容错共识机制(Trust Model Practical Byzantine Fault Tolerance,TMPBFT)。该机制通过建立信任模型,针对节点不同行为给予其不同信任评价,判断节点信任状态,提高可信节点投票权,筛选并限制恶意节点。并将信任模型与基于 PBFT 的一致性协议和视图更换协议相结合,加强系统安全性及稳定性。实验结果表明:TMPBFT 能够较好地区分并限制恶意节点,提高系统的稳定性与容错能力。

关键词 区块链 共识机制 信任模型 信任值 节点信任状态 PBFT

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.07.046

PBFT CONSENSUS MECHANISM BASED ON TRUST MODEL

Zhao Lijin[1] Wang Xinsheng[1] Xia Chunzhong[2]

[1](*School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, Jiangsu, China*)

[2](*Information Center, Jiangsu University, Zhenjiang 212000, Jiangsu, China*)

Abstract Aiming at the problem that the practical Byzantine fault tolerance (PBFT) consensus mechanism cannot pre-evaluate and restrict malicious nodes in advance, this paper proposes a trust model practical Byzantine fault tolerance (TMPBFT). The mechanism established the trust model, gave different trust evaluations to different behaviors of nodes, judged the trust status of nodes, improved the voting rights of trusted nodes, filtered and limited malicious nodes. The trust model was combined with the PBFT-based consensus protocol and view replacement protocol to enhance system security and stability. The experimental results show that TMPBFT can better distinguish and limit malicious nodes, and improve the stability and fault tolerance of the system.

Keywords Blockchain Consensus mechanism Trust model Trust value Node trust state PBFT

0 引 言

自诞生以来,区块链以其解决第三方信任问题[1-2]的特性引起关注。其在技术上具有隐私保护性、不可篡改性;在业务上具有可信度高、安全性强等优点[3]。

共识机制是区块链完成数据一致性的核心机制[4]。常见共识机制有 PBFT[5]、XFT[6]、Zyzzyva[7]、PoW[8]、Poxos[9]等。其中实用拜占庭容错(PBFT)共识机制是主流共识机制之一,许多算法基于 PBFT 加以改进[10-11]。

但这些算法都没有解决以下问题:系统中往往存在恶意节点,PBFT 随机选择主节点,恶意节点很可能成为主节点,导致视图切换频繁,系统性能和稳定性下降[12]。且 PBFT 容错能力有限,较多恶意节点会导致共识完成周期过长,系统性能急剧下降[13-14]。

产生以上问题的主要原因是 PBFT 不提前评估节点,仅运行中识别节点行为,判定状况。因此,对节点预评估和提前控制是可行的解决方法。P2P 领域已有

收稿日期:2019-12-20。国家自然科学基金项目(61402205);江苏大学高级技术人才科研基金项目(12JDG104)。**赵立瑾**,硕士生,主研领域:区块链。**王新胜**,副教授。**夏纯中**,高级实验师。

针对节点状况的研究,但不完全适用于区块链领域。

本文针对区块链环境,引入并改进 P2P 信任模型,提出一种基于信任模型的 PBFT 共识机制(TMPBFT)。通过信任模型判断节点行为,评估节点信任值,以信任值区分各种节点状态,提高可信节点投票权,限制恶意节点。并基于 PBFT 共识机制,应用信任模型到一致性协议和视图更换协议中,加强节点管控,提高系统稳定性和容错能力。

1　相关工作

Zhang 等[15]提出一种基于 PBFT 的组层次算法,将节点分组共识以减少共识的消息复杂度。Feng 等[16]提出一种可拓展的动态多代理分层 PBFT 算法,节点分为多个自治系统以减少系统延迟。Hao 等[17]提出一种动态 PBFT,解决了常规 PBFT 中存在的热插拔节点以及处理恶意分类账的问题。Feng 等[18]在改进的 Rollerchain 中利用分片技术和 PBFT 算法提出了一种可分片的区块链协议。Wang 等[19]提出了一种混合区块链方法,该方法混合了区块链架构,根据交易特征选择相应的共识机制。

以上相关研究在各方面对共识机制有所改进,但均未能解决 PBFT 不能预识别非正常节点的问题。

P2P 信任模型研究评价网络中节点的行为[20]。近年来,信任模型通过综合多因素[21]的方式获取对节点的准确评估。

故本文提出一种结合区块链特点,综合时间、参与频度、具体表现等相关因素的信任模型,并应用在 PBFT 中,以解决 PBFT 无法预识别非正常节点的问题。

2　基于信任模型的 PBFT 共识机制

2.1　信任模型

信任模型通过节点行为评估信任值,借助信任值区分节点信任状态,赋予不同信任状态节点不同权限。

2.1.1　总体框架

节点行为分类及内容情况如表 1 所示。

表 1　节点行为分类和具体内容情况表

节点行为	行为具体内容	信任值
正常行为	正常传递信息	增加
故障行为	1. 信息超时 2. 信息丢失 3. 不同意大多数	减少
恶意行为	1. 篡改发送信息 2. 自称接收到恶意信息	归零

可以看出,节点不同行为会导致信任值变化。信任值是用于反映节点状况的数值,值域为[0,1],由 C_{ij} 表示,代表 $Node_{ij}$(组织 i 中的节点 j)的信任值,组织是承担数据信用责任的区块链系统参与方。C_{ij}大小决定节点信任状态。初始信任值为 C_{init},默认为 0.5。

节点信任状态定义如表 2 所示。

表 2　节点信任状态定义表

节点信任状态	定义
可信节点	可信度高,$C_{\text{good}} < C_{ij} \leqslant 1$,默认 $C_{\text{good}} = 0.8$
普通节点	可信度中,$C_{\text{normal}} \leqslant C_{ij} < C_{\text{good}}$,默认 $C_{\text{normal}} = 0.5$
故障节点	可信度低,$C_{\text{bad}} \leqslant C_{ij} < C_{\text{normal}}$,默认 $C_{\text{bad}} = 0.2$
恶意节点	不可信,存在恶意行为或 $0 \leqslant C_{ij} < C_{\text{bad}}$

表 2 中,C_{good}、C_{normal}、C_{bad}分别代表了可信、普通、故障节点的信任值下限。其中可信节点会积极验证和传递信息;普通节点有部分故障行为,偶然延迟信息传递;故障节点故障行为较多,明显延缓信息传播;恶意节点极有可能篡改信息或消极传播信息。

如表 3 所示,恶意节点为备份节点,于 10 次共识后信任值提升为 C_{bad},作为故障节点参与共识。而在无可信节点时,普通节点作为主节点。

表 3　节点信任状态及权限表

权限	可信节点	普通节点	故障节点	恶意节点
主节点	√	(√)		
副本节点	√	√	√	
备份节点				√

投票权是节点确认信息的影响力。普通、故障节点投票权分别为 1 和 0.5。恶意节点无投票权。可信节点投票权最高,由 V_{good}表示,计算如下:

$$V_{\text{good}} = 1 + \frac{0.5 \times N_{\text{bad}}}{N_{\text{good}}} \tag{1}$$

式中:N_{good}和 N_{bad}分别表示可信和故障节点数量。

2.1.2　信任值计算因素

信任值计算因素体现节点情况,用于计算信任值。

定义 1　节点活跃度。节点活跃度指受评节点在一定时间内参与共识的频度,节点活跃度可通过函数 $\rho(n)$表示,计算如下:

$$\rho(n) = \mathrm{e}^{\left(-\frac{a}{n}\right)} \tag{2}$$

式中:n 为节点参与共识次数;参数 $a(a \in \mathbf{Z}, a \geqslant 1)$可调节增长速度。$\rho(n)$值随 n 值增大而增大。

定义 2　节点共识完成率。节点共识完成率指受评节点参与共识的完成频度。由共识完成率 γ 表示,

值越大表现越好,计算如下:

$$\gamma = \sqrt{\frac{s}{n}} \tag{3}$$

式中:s 为正常完成共识次数;n 为参与共识次数。

定义 3 历史影响度。历史影响度是指节点历史信任值对当前信任值影响程度。历史影响度 $\omega(\Delta t)$ 计算如下:

$$\omega(\Delta t) = \left(\frac{1}{2}\right)^{\frac{\Delta t}{\lambda}} \tag{4}$$

式中:Δt 为当前时间与上次共识时间的时间间隔;参数 $\lambda(\lambda \in \mathbf{Z}, \lambda > 0)$ 可调整影响变化程度。$\omega(\Delta t)$ 值随 Δt 增大而减小,其值越小历史信任值影响越小。

定义 4 事务影响因子。事务影响因子标识事务重要程度。设交易重要性参数为 m,事务影响因子计算函数 $F(m)$ 计算如下:

$$F(m) = \begin{cases} \dfrac{m}{M_0} & m < M_0 \\ 1 & m \geqslant M_0 \end{cases} \tag{5}$$

式中:M_0 为交易重要性参数的阈值;$F(m)$ 值随 m 值增大而增大,$F(m)$ 越大表示事务重要程度越高。

定义 5 行为评价值。行为评价值指依据受评节点参与共识的表现,给予节点的相应评价值。行为评价值 E 计算如下:

$$E = \frac{1}{2^{(c \times t)}} \times \frac{g}{N} \tag{6}$$

式中:t 为节点完成共识用时;$c(c \in \mathbf{Z}, c \geqslant 1)$ 为评价值调节因子;g 为同意信息的节点数量;N 为节点总数量。式(6)中,节点在大多数节点认同信息时,完成共识用时越短,评价越高。其他情况都将导致评价值较低。

2.1.3 信任值计算

信任值 C_{ij} 是受评节点 $Node_{ij}$ 在共识中的综合评价。信任值计算如下:

$$C_{ij} = \rho(n)\gamma \frac{\sum_{l=1}^{n} E_l \omega(\Delta t_l) F_l}{\sum_{l=1}^{n} \omega(\Delta t_l) F_l} + (1 - \rho(n)) C_{\text{init}} \tag{7}$$

式中:C_{init} 为信任值初值。式(7)结合定义 2 - 定义 6 中因素,综合往次共识情况计算,以精确反映节点情况。

式(7)伪代码如算法 1 所示。

算法 1 信任值计算

输入:$n, s, \Delta t, m, t, g$。

输出:C_{ij}。

```
1. P = e^( -a/n)                                  //节点活跃度计算
2. for (i = 0; i < n; i ++) {
3. if (m < M0) F = m/M0
4. else F = 1                                     //事务影响因子计算
5. W = 0.5^(Δt/r)                                 //历史影响度计算
6. E = (0.5^(c * t)) * g/N                        //行为评价值计算
7. if (E > 0.5)  R = ((s + 1)/(n + 1))^0.5
8. else R = ((s + 1)/n)^0.5                       //节点共识完成率计算
9. Cij = E * W * F/W/F + Cij }
10. Cij = P * R * Sij + (1 - P) * C0              //信任值计算
```

2.1.4 信任值更新

每次共识后,需及时更新节点信任值,以客观体现信任值变化,并根据信任值确定节点信任状态。

设共识前节点共识信任值为 T_{ij},本次共识中节点共识信任值计算为 S_{ij},信任值更新计算如下:

$$C_{ij} = \begin{cases} \theta \times S_{ij} + (1 - \theta) \times T_{ij} & \text{节点无恶意行为} \\ 0 & \text{节点有恶意行为} \end{cases} \tag{8}$$

式中:θ 为信任值更新调节因子。式(8)中若节点本次信任值高于往次,将适当调高信任值,反之调低。

节点若离线,C_{ij} 会衰减,即每次共识式(8)中 S_{ij} 值为 C_{init}。直到 $C_{ij} \in [0.9 \times C_{\text{init}}, 1.1 \times C_{\text{init}}]$,或节点上线。

信任值更新调节因子 $\theta(0 \leqslant \theta \leqslant 1)$ 由式(9)确定:

$$\theta = \begin{cases} \dfrac{1}{2 \times \dfrac{S_{ij}}{T_{ij}}} & \dfrac{S_{ij}}{T_{ij}} \geqslant \dfrac{1}{2} \\ 1 & \dfrac{S_{ij}}{T_{ij}} < \dfrac{1}{2} \end{cases} \tag{9}$$

其使信任值与上次信任值差距较大时,较小的信任值所占权重较大。可避免节点的反常行为得逞。

式(9)伪代码如算法 2 所示。

算法 2 信任值更新

输入:S_{ij}, T_{ij}。

输出:C_{ij}。

```
1. if (节点本次有恶意行为) {Cij = 0}
2. else {
3.     if (Sij/Tij >= 0.5) B = Tij/2/Sij
4.     else B = 1      }                          //信任值调节因子计算
5. Cij = B * Sij + (1 - B) * Tij                  //信任值更新
6. 广播 Cij
```

2.2 一致性协议

本文结合信任模型设计协议,令协议在达成数据一致性的同时,分辨节点信任状态。具体过程如下。

pre-prepare 阶段:主节点验证客户端信息后,发送

预准备信息给副本节点,进入 prepare 阶段;副本节点收到并验证预准备信息后,进入 prepare 阶段。

prepare 阶段:副本节点发送准备信息给所有节点,收到其他节点发送的 $2\times f(f$ 为节点总数的三分之一)条准备信息后,验证所有准备信息,验证通过后进入 commit 阶段。记录节点行为。

commit 阶段:所有节点发提交信息给其他节点,收到 $2\times f$ 条提交信息后,验证所有提交信息,验证通过后执行客户端请求。记录节点行为,计算并更新节点信任值和节点信任状态。

2.3　视图更换协议

本文提出的视图更换协议中,主节点在可信状态的节点中随机选择。具体过程如下(设当前视图为 v)。

view-change 阶段:副本节点进入视图 $v+1$,随机从信任状态为可信节点的节点中选一个非活动节点作为主节点。副本节点向其他节点发 view-change 信息。

view-change-ack 阶段:节点收到 $2\times f+1$ 条 view-change 信息后,发送 view-change-ack 信息到视图 $v+1$ 的主节点。新主节点收到 view-change 信息和 view-change-ack 信息后进入 new-view 阶段。

new-view 阶段:新主节点选择检查点作为 new-view 请求初始状态,根据本地区块链数据执行一致性协议。

3　实验分析

根据基于信任模型的 PBFT 共识机制,实现了一个实验系统。实验系统为 Ubuntu 16.04.6 LTS 64 位,实验平台为 Hyperledger Fabric v0.6。

3.1　节点信任值奖励及惩罚

图 1 中实验展示的是节点行为评价较高时,普通节点信任值奖励的情况。

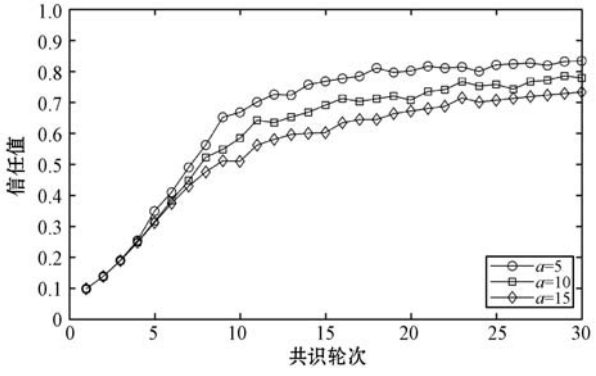

图 1　信任值奖励

图 2 展示了是节点行为评价较低时,普通节点信任值奖励的情况。

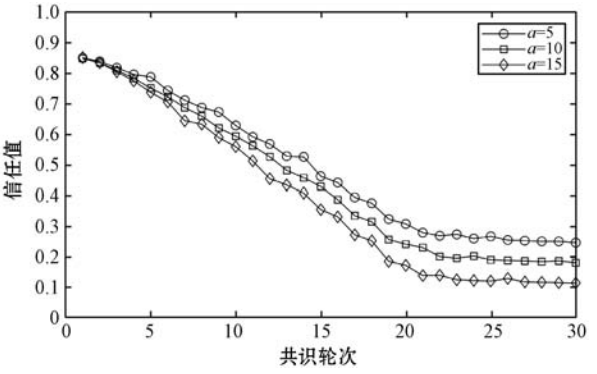

图 2　信任值惩罚

综合图 1、图 2 情况,活跃度调节因子 a 值越小,信任值奖励增速越高,惩罚降速越低。故系统可信度高时可设置较小 a 值。节点连续表现良好时信任值奖励大,节点连续表现差时信任值惩罚大,但都有限度。

图 3 中实验展示的是节点行为评价较高时,故障节点与普通节点信任值奖励相比较的情况。

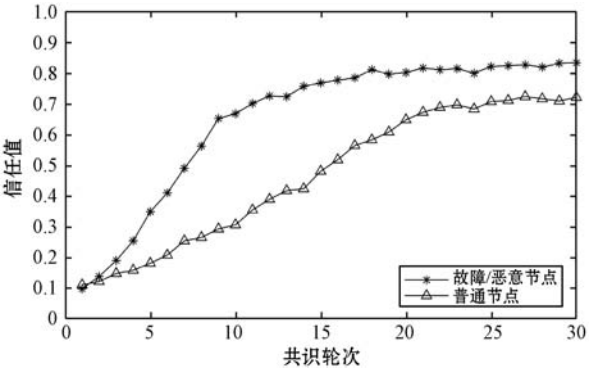

图 3　故障节点与普通节点信任值奖励比较

可以看出,故障节点信任值增长速率明显较普通节点低。因为节点共识完成率记录了节点表现,历史故障或恶意行为会对当前信任值奖励有负面影响。

3.2　节点信任值排名

图 4 实验展示的是某几个不同信任状态的节点在共识中的节点信任值排名。

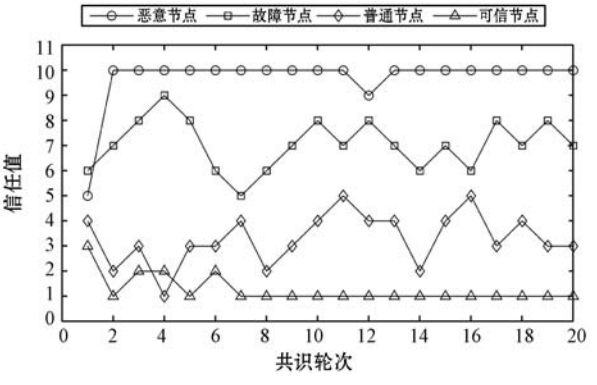

图 4　节点的信任值排名变化情况

图 4 中,整体排名高到低分别是可信、普通、故障、恶意节点。因为按照可信、普通和故障节点的顺序,故障行为占总行为比例逐渐上升。结果表明信任模型可

有效识别节点信任状态。

3.3 恶意节点占比

图 5 展示了恶意节点占比的变化情况。

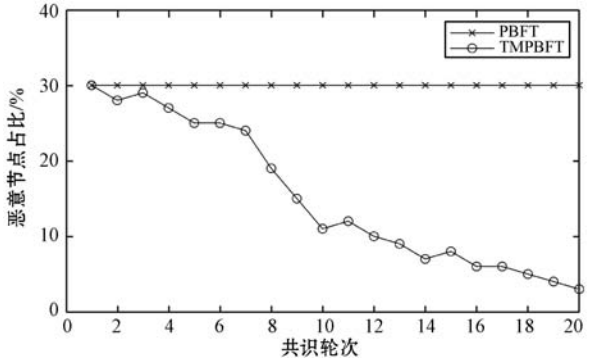

图 5 恶意节点占比变化

可以看出,PBFT 恶意节点占比不变,TMPBFT 逐渐降低。因为 TMPBFT 使用信任模型识别限制恶意节点,减少了系统中的恶意节点,提高了容错能力。

3.4 系统运行吞吐率以及延迟

图 6、图 7 展示了 TMPBFT 吞吐率和延迟与 PBFT 比较情况,系统中存在恶意节点。

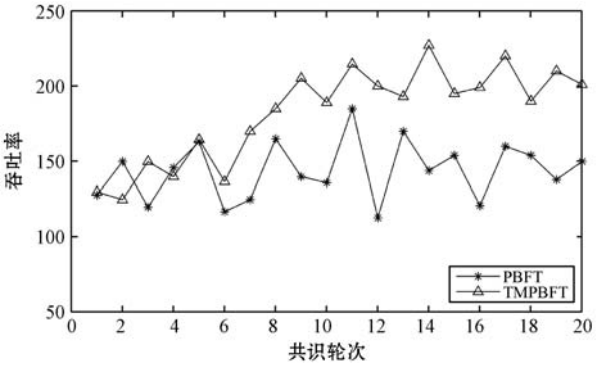

图 6 TMPBFT 与 PBFT 的吞吐率比较情况

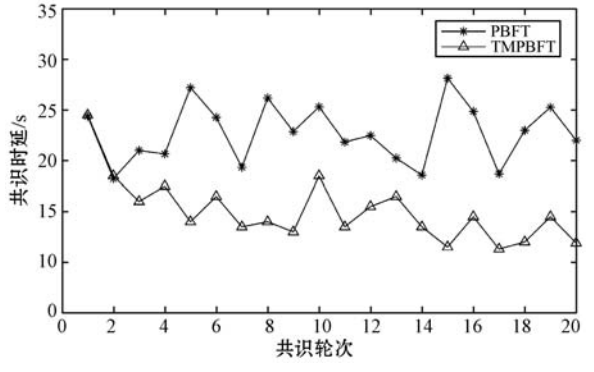

图 7 TMPBFT 与 PBFT 延迟比较情况

图 6、图 7 中,TMPBFT 在运行时整体吞吐率较 PBFT 高,延迟较 PBFT 低。因为 TMPBFT 在逐步限制恶意节点的同时,会降低故障节点投票权,提高可信节点投票权,以此在达成超过三分之二节点投票权的前提下降低了达成共识所需节点数,提高了系统整体性能。

4 结 语

针对 PBFT 无法预判断和控制节点的问题,本文提出一种基于信任模型的 PBFT 共识机制(TMPBFT)。构建信任模型,判断节点行为,通过信任值体现节点信任状态,区分节点,提高可信节点投票权,限制恶意节点。并基于 PBFT 共识机制,结合信任模型改进设计一致性协议和视图更换协议,加强了节点的管理,促进了性能提升,提高了系统的稳定性及容错能力。

参 考 文 献

[1] Hellani H, Samhat A E, Chamoun M, et al. On blockchain technology: Overview of bitcoin and future insights[C]// 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET),2018.

[2] 邵奇峰,金澈清,张召,等. 区块链技术:架构及进展[J]. 计算机学报,2018,41(5):969 - 988.

[3] 邵奇峰,张召,朱燕超,等. 企业级区块链技术综述[J]. 软件学报,2019,30(9):2569 - 2592.

[4] Zheng Z, Xie S, Dai H, et al. An overview of blockchain technology: Architecture, consensus, and future trends [C]//2017 IEEE International Congress on Big Data,2017.

[5] Castro M, Liskov B. Practical byzantine fault tolerance and proactive recovery[J]. ACM Transactions on Computer Systems,2002,20(4):398 - 461.

[6] Liu S, Viotti P, Cachin C, et al. XFT: Practical fault tolerance beyond crashes[C]//12th USENIX Symposium on Operating Systems Design and Implementation, 2016: 485 - 500.

[7] Kotla R, Alvisi L, Dahlin M, et al. Zyzzyva: Speculative byzantine fault tolerance[C]//ACM SIGOPS Operating Systems Review,2007,41(6):45 - 58.

[8] Dwork C, Naor M. Pricing via processing or combatting junk mail [C]//Annual International Cryptology Conference, 1992:139 - 147.

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

[10] Yaga D, Mell P, Roby N, et al. Blockchain technology overview[EB]. arXiv:1906.11078,2019.

[11] Crosby M, Pattanayak P, Verma S, et al. Blockchain technology: Beyond bitcoin[J]. Applied Innovation, 2016, 2(6 - 10):71.

[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.

[13] 钱卫宁,邵奇峰,朱燕超,等. 区块链与可信数据管理:问题与方法[J]. 软件学报,2018,29(1):150 - 159.

[14] Underwood S. Blockchain beyond bitcoin[J]. Communications of the ACM,2016,59(11):15 – 17.

[15] Zhang L, Li Q. Research on consensus efficiency based on practical byzantine fault tolerance[C]//2018 10th International Conference on Modelling, Identification and Control (ICMIC),2018.

[16] Feng L, Zhang H, Chen Y, et al. Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain[J]. Applied Sciences, 2018, 8(10): 1919.

[17] Hao X, Yu L, Liu Z, et al. Dynamic practical Byzantine fault tolerance[C]//2018 IEEE Conference on Communications and Network Security (CNS),2018:1 – 8.

[18] Feng X, Ma J, Miao Y, et al. Pruneable sharding-based blockchain protocol[J]. Peer-to-Peer Networking and Applications,2019,12(4):934 – 950.

[19] Wang Y, Zhao H, Li T, et al. Hybrid-chain: An innovative and efficient mixed blockchain architecture[C]//2018 3rd International Conference on Electrical, Automation and Mechanical Engineering (EAME 2018),2018.

[20] 刘建生,游真旭,乐光学,等. 网络信任研究进展[J]. 计算机科学,2018,45(11):13 – 28.

[21] 刘义春,梁英宏. 基于上下文因素的 P2P 动态信任模型[J]. 通信学报,2016,37(8):34 – 45.

(上接第 240 页)

[8] Wang X, Su Y. Segmentation-Based stereo matching using improved self-adapting dissimilarity measure based on second order smoothness priors[J]. Journal of Computational and Theoretical Nanoscience,2015,12(9):2701 – 2709.

[9] Sun J, Zheng N N, Shum H Y. Stereo matching using belief propagation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2003,25(7):787 – 800.

[10] Zhu S, Gao R, Li Z. Stereo matching algorithm with guided filter and modified dynamic programming[J]. Multimedia Tools and Applications,2017,76(1):199 – 216.

[11] Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05),2005:807 – 814.

[12] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436 – 444.

[13] Mayer N, Ilg E, Hausser P, et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR),2016.

[14] Pang J, Sun W, Ren J S J, et al. Cascade residual learning: A two-stage convolutional neural network for stereo matching[C]//IEEE International Conference on Computer Vision Workshops,2017:887 – 895.

[15] Kendall A, Martirosyan H, Dasgupta S, et al. End-to-end learning of geometry and context for deep stereo regression[C]//IEEE International Conference on Computer Vision, 2017:66 – 75.

[16] Chang J R, Chen Y S. Pyramid stereo matching network[C]//IEEE Conference on Computer Vision and Pattern Recognition,2018:5410 – 5418.

[17] 曾军英,冯武林,秦传波,等. 实时自适应的立体匹配网络算法[J]. 信号处理,2019,35(5):843 – 849.

[18] 王玉锋,王宏伟,于光,等. 基于三维卷积神经网络的立体匹配算法[J]. 光学学报,2019,39(11):227 – 234.

[19] 肖进胜,田红,邹文涛,等. 基于深度卷积神经网络的双目立体视觉匹配算法[J]. 光学学报,2018,38(8):179 – 185.

[20] Fu J, Liu J, Tian H, et al. Dual attention network for scene segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition,2019:3146 – 3154.

[21] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//European Conference on Computer Vision(ECCV),2018:3 – 19.

[22] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[EB]. arXiv: 1508.04025,2015.

[23] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//IEEE Conference on Computer Vision and Pattern Recognition,2018:7132 – 7141.

[24] Zhang H, Goodfellow I, Metaxas D, et al. Self-attention generative adversarial networks[EB]. arXiv:1805.08318, 2018.

[25] Žbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches[J]. The Journal of Machine Learning Research, 2016, 17(1): 2287 – 2318.

[26] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012:3354 – 3361.

[27] Menze M, Geiger A. Object scene flow for autonomous vehicles[C]//IEEE Conference on Computer Vision and Pattern Recognition,2015:3061 – 3070.

[28] Kingma D P, Ba J. Adam: A method for stochastic optimization[EB]. arXiv:1412.6980,2014.

[29] Liang Z, Feng Y, Guo Y, et al. Learning deep correspondence through prior and posterior feature constancy[EB]. arXiv:1712.01039,2017.