

区块链网络拓扑优化和转发策略设计

霍如^{1,2}, 程祥凤¹, 孙闯³, 汪硕⁴, 黄韬⁴, F.Richard Yu⁵

(1. 北京工业大学信息学部, 北京 100124; 2. 网络通信与安全紫金山实验室, 江苏 南京 211111;
3. 清华大学自动化系, 北京 100084; 4. 北京邮电大学网络与交换技术国家重点实验室, 北京 100876;
5. 加拿大卡尔顿大学信息技术学院, 渥太华 K1S 5B6)

摘要: 为解决区块链网络的数据传输效率低问题, 提出了区块链传输效率优化方法来优化网络拓扑和转发策略。首先, 设计了可信值函数计算区块链节点的可信值, 综合考虑可信值和传输时间构建树形拓扑。然后, 基于树形拓扑设计转发路径选择策略, 以最小整体并发传输时间为目标, 建立节点关于其邻居节点转发次序的转发表。为了减小节点变化对树形拓扑的影响, 提出了拓扑动态优化策略局部调整树形拓扑。转发路径选择策略使整个数据传输过程具有最小传输时间, 动态优化策略避免重构全网拓扑, 有效缩短数据传输时间。仿真结果表明, 与权重优先算法相比, 所提方法的传输时间减小了约 20%, 显著提高了数据传输效率。

关键词: 区块链网络; 树形拓扑; 转发路径选择; 拓扑动态优化; 传输效率

中图分类号: TP393.0

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2022240

Topology optimization and forwarding strategy design for blockchain network

HUO Ru^{1,2}, CHENG Xiangfeng¹, SUN Chuang³, WANG Shuo⁴, HUANG Tao⁴, F.Richard Yu⁵

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

2. Purple Mountain Laboratories, Nanjing 211111, China

3. Department of Automation, Tsinghua University, Beijing 100084, China

4. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

5. Information Technology Department, Carleton University, Ottawa K1S 5B6, Canada

Abstract: In order to solve the problem of the low data transmission efficiency of the blockchain network, an optimization method of blockchain transmission efficiency was proposed to optimize the network topology and forwarding strategy. First, the trusted value function was designed to calculate the trusted value of blockchain nodes, and a tree topology was constructed by comprehensively considering the trusted value and transmission time. Furthermore, the forwarding path selection strategy was designed based on the tree topology, with the minimum overall concurrent transmission time as the goal, and the forwarding table of the node about the forwarding order of its neighbor nodes was established. To reduce the impact of the node changed on the tree topology, a topology dynamic optimization strategy was proposed to adjust tree topology locally. The forwarding path selection strategy enabled the entire data transmission process to have the minimum transmission time, and the dynamic optimization strategy avoided reconfiguration of the entire network topology and effectively shortened the data transmission time. The simulation results show that, compared with the weight-first algorithm, the transmission time of the proposed method is reduced by about 20%, and the data transmission efficiency is significantly improved.

Keywords: blockchain network, tree topology, forwarding path selection, topology dynamic optimization, transmission efficiency

收稿日期: 2022-08-10; 修回日期: 2022-11-06

基金项目: 2020 年工业互联网创新发展工程基金资助项目(工业互联网标识资源搜索系统); 2019 年工业互联网创新发展工程基金资助项目(创新型工业互联网标识解析系统)

Foundation Items: The MIIT of China 2020 (Identification Resource Search System for Industrial Internet of Things), The MIIT of China 2019 (Innovative Identification and Resolution System for Industrial Internet of Things)

0 引言

虽然自比特币诞生以来,人们对其持不同态度,但作为其底层技术的区块链吸引了大众的关注,越来越多的学者开始研究区块链,区块链技术呈现蓬勃发展的态势。区块链凭借自身的去中心化和难以篡改等特性,已经应用于许多领域^[1-2]。随着区块链技术的广泛使用,区块链的低传输效率导致的高交易时延严重限制了其进一步发展^[3]。现已有有向无环图(DAG, directed acyclic graph)^[4]和共识协议^[5]等方法分别从数据层和共识层进行改进,以提高区块链传输效率。然而,不论针对区块链架构的哪一层进行优化,区块链中的交易和区块都会不可避免地涉及数据传输过程。在区块链网络中,交易和区块的生成和传播是很频繁的,并且在节点之间进行传输是相当耗时的。因此,本文对区块链网络层进行重点研究。

区块链网络层为区块链提供底层的网络支撑,搭建节点之间的数据传输环境。网络层包含 2 种机制,即对等(P2P, peer-to-peer)网络、数据传输和数据验证机制^[6]。区块链网络包括无结构、结构化和混合式 P2P 网络^[2]。其中,无结构 P2P 网络是指具有随机图拓扑结构的 P2P 网络。如何优化区块链网络层从而缩短数据传输时间是扩展区块链性能的一个重要研究方向^[7]。

目前,很多研究通过改进区块链网络拓扑结构和数据传输方法来提高区块链数据传输效率。在区块链网络拓扑结构研究方面,Sung 等^[8]通过建立数学模型将数据传输路径转化为最小生成树问题,减少传输链路的重用来最小化处理 and 传输时延。Li 等^[9]提出一种权重优先算法,该算法逐次选取通信权重小以及传输能力强的节点构建通信树,为区块链交易提供路径,从而提高区块链的通信效率。Lin 等^[10]根据节点的可信度和中心性提出一种可靠的去中心化树形算法,该算法利用 Prim 算法构建通信树,提高了区块链交易的速度和可靠性。在区块链数据传输方法研究方面,Ersoy 等^[11]提出一种传输激励机制,节点向邻居节点传输交易时,可以获得交易费的一部分作为奖励,鼓励更多节点参与交易的传输,降低区块链传输时间。Santiago 等^[12]提出一种基于权重的邻居选择方法,通过不断探测节点的新邻居并更新节点间的权重,增加选择最佳通信节点的概率,减少区块链网络的消息传播

时间。Hao 等^[13]从区块链网络拓扑结构和数据传输方法两方面改进,对节点聚类生成的集群构建信任增强的区块链 P2P 网络拓扑,采用并行生成树算法进行数据传输,在加快区块链传输速率的同时保证传输的可靠性。

上述研究未考虑节点状态变化对已建立的网络拓扑产生的影响^[8-10]。如果区块链节点状态发生变化,只能重新构建拓扑结构,从而影响区块链传输效率。此外,大多数研究只考虑对区块链网络拓扑结构和数据传输方法的单方面改进,很少有结合两方面的改进。基于上述问题,本文针对区块链网络拓扑和转发策略进行优化,通过设计的可信值函数计算节点可信值,综合考虑节点可信值和节点之间的传输时间,提出树形拓扑构建方法,在生成的树形拓扑基础上,设计一种转发路径选择策略和拓扑动态优化策略,从而缩短数据传输时间,提高区块链传输效率。本文主要的研究工作如下。

1) 提出树形拓扑构建方法。首先,利用设计的可信值函数得到每个节点的可信值,根据可信值阈值划分节点类型。然后,对可信节点采用 Sollin 算法构建最小生成树,将不可信节点加入拓扑,完成树形拓扑的构建,简化网络的拓扑结构,并提高数据传输过程中的安全性。

2) 设计转发路径选择策略。该策略以最小整体并发传输时间为目标,为每个区块链节点建立关于其邻居节点转发次序的转发表,节点根据转发表依次选择其邻居节点进行数据转发,使整个传输过程具有最小并发传输时间,有效缩短数据传输时间,同时避免对节点重复传输同一数据。

3) 设计拓扑动态优化策略。针对节点加入、节点离开、节点可信值变化和节点间权重变化情况,利用破圈法和 Sollin 算法设计了相应的拓扑动态优化策略,只对节点状态变化影响的部分拓扑结构进行调整。与全网重构拓扑相比,该策略花费的代价较小,并且能够减少对拓扑进行调整所带来的开销。

1 区块链传输效率优化方法

为了缩短区块链交易数据在网络中的传输时间,本文从区块链网络的拓扑结构和节点的转发策略两方面进行改进,提出区块链传输效率优化方法,如图 1 所示。其中,树形拓扑构建模型负责生成区块链网络的树形拓扑,转发路径选择模型基于树形拓扑结构为区块链节点建立转发路径,拓扑动

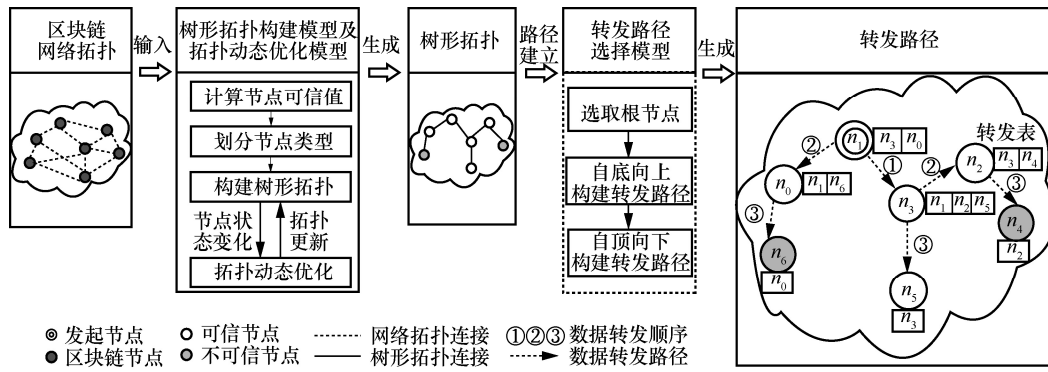


图 1 区块链传输效率优化方法

态优化模型主要对树形拓扑中状态发生变化的节点局部拓扑进行调整。

树形拓扑构建模型主要负责将区块链无结构 P2P 网络构建为树形拓扑结构，通过优化网络拓扑结构来减少冗余传输。考虑到数据传输过程中的安全性问题，设定节点可信值函数并结合可信值阈值将区块链网络节点划分为可信节点和不可信节点，其中，可信值大于阈值的节点为可信节点，否则为不可信节点。针对可信节点和不可信节点采用不同的方式构建网络的树形拓扑。

转发路径选择模型基于树形拓扑结构设计区块链节点的数据转发策略，以加快数据的传输。首先，为每个节点建立关于其邻居节点的转发表，转发表中的节点依据其邻居节点所在子树的最小并发传输时间由大到小顺序依次存放。然后，从发起节点开始转发数据，节点按照转发表中的次序向未接收数据的邻居节点传输数据，经过多轮次的数据转发，从而使全网节点接收到数据。

拓扑动态优化模型根据区块链节点的状态变化情况，包括节点加入、节点离开、节点可信值变化和节点间权值变化，分别设计了拓扑动态优化策略，降低调整网络拓扑的开销。当区块链网络的节点状态发生变化时，拓扑动态优化模型选择相应的拓扑动态优化策略来调整因节点状态变化所影响的局部拓扑结构，然后更新树形拓扑结构。

在区块链网络中，使用本文提出的区块链传输效率优化方法对 P2P 网络建立树形拓扑以及生成节点的转发表，由记账节点作为数据发起节点开始传输交易数据。在数据传输过程中，节点按照转发路径选择策略进行数据转发。节点在接收到交易数据后会验证交易的合法性，若交易合法，则存储到账本。如果节点状态变化导致树形

拓扑发生改变，则利用拓扑动态优化策略局部调整树形拓扑。

1.1 树形拓扑构建方法

本文将区块链网络拓扑抽象为图，节点之间的路径抽象为边，边的权值代表节点之间的数据传输时间。考虑节点的可信值和节点之间的权值，提出树形拓扑构建方法生成树形拓扑结构，简化网络拓扑的同时提高数据传输过程的安全性。

区块链网络中的节点并不一直都是诚实节点，存在一些恶意节点，恶意节点可能产生恶意行为，篡改或者破坏已接收到的数据，再将数据转发给其他节点，从而影响数据的可信性，存在安全性问题^[14]。因此，本文设计了可信值函数计算区块链节点的可信值，利用节点的可信值描述节点自身的可信程度。设置新加入节点的初始可信值为 0，根据节点与其他节点进行交易的成功次数和失败次数定义节点的可信值。节点的可信值为

$$C_i(s, u, n) = \begin{cases} \frac{s}{s+u} + \frac{2 \arctan(s)}{\pi}, u = 0 \\ \frac{s}{s+u} + \frac{2 \arctan(s)}{\pi} \varphi(n), u \neq 0 \\ 0, s = u = 0 \end{cases} \quad (1)$$

其中， i 为节点的标号； s 为节点交易成功的总次数； u 为节点交易失败的总次数； n 为节点交易失败到下一次失败之间累计的连续交易成功次数； $\frac{2 \arctan(s)}{\pi}$ 为奖励函数，表示给予交易成功节点的可信值奖励，交易成功的次数越多，节点获得的奖励值越大，激励节点进行更多成功的交易； $\varphi(n)$ 是惩罚函数，在节点交易失败时，惩罚函数会降低节点的可信值。惩罚函数定义为

$$\varphi(n) = \begin{cases} 1 - \frac{1}{\sqrt{1+n}}, & u_x - u_{x-1} = 0 \\ 0, & u_x - u_{x-1} = 1 \end{cases} \quad (2)$$

其中, u_x 表示节点本次交易结束后总的交易失败次数; u_{x-1} 表示节点上一次交易结束后总的交易失败次数。然后, 根据 $u_x - u_{x-1}$ 的值来判断本次交易是否成功。当节点的本次交易失败时, $\varphi(n)$ 取值为 0, 则不会获得来自奖励函数的可信值奖励, 同时清空连续交易成功的累计次数 n 。当节点的本次交易成功时, 惩罚函数为 $1 - \frac{1}{\sqrt{n+1}}$,

节点获得奖励函数的少量奖励值, 同时累计次数增加 1。

区块链节点利用可信值函数计算自身的可信值, 依据可信值阈值 $C_{\text{threshold}}$ 将可信值大于 $C_{\text{threshold}}$ 的节点划分为可信节点, 小于或等于 $C_{\text{threshold}}$ 的节点划分为不可信节点。划分节点的目的是让可信节点主要承担数据传输任务, 向其他节点转发数据。不可信节点则负责接收数据, 防止出现由不可信节点担任转发节点而引起数据篡改或被破坏的安全问题。基于上述思路, 选取可信节点以及它们之间的边, 从而得到由可信节点组成的可信拓扑。假设 $G = (N, E, C, W)$ 是一个区块链网络的拓扑, N 为节点集, E 为边集, C 为节点可信值集, W 为边的权值集。 n_i 表示区块链节点并且 $n_i \in N$, c_i 表示节点的可信值并且 $c_i \in C$, $w_{i,j}$ 表示 n_i 与 n_j 相连边的权值并且 $w_{i,j} \in W$ 。设 $\{n_0, n_1, n_2, n_3, n_5\}$ 中的节点可信值大于 $C_{\text{threshold}}$, $\{n_4, n_6\}$ 中的节点可信值小于 $C_{\text{threshold}}$ 。

Sollin 算法是一种结合 Prim 和 Kruskal 算法思想构建的最小生成树算法, 其每次迭代可同时扩展多棵子树, 使其建树的效率较高。因此, 本文采用 Sollin 算法构建最小生成树。本文提出的树形拓扑构建方法对可信节点和不可信节点按不同的方式构建网络的树形拓扑。在可信拓扑基础上, 该算法首先利用 Sollin 算法根据可信节点之间的边和权值, 选择权值最小边构建最小生成树, 生成可信树形拓扑; 然后, 对不可信节点进行处理, 选择与可信邻居节点相连的权值最小边加入树形拓扑, 不可信节点作为树形拓扑的叶子节点; 将所有不可信节点加入树形拓扑之后完成网络树形拓扑的构建。

构建网络的树形拓扑示例如图 2 所示, $\{n_0, n_1, n_2, n_3, n_5\}$ 为可信节点集, $\{n_4, n_6\}$ 为不可信节点集。首先, 利用 Sollin 算法对集合 $\{n_0, n_1, n_2, n_3, n_5\}$ 中的可信节点逐次选取权值最小边构建一棵最小生成树, 得到图 2 中的实线边即最小生成树的边。然后, 不可信节点 n_4 在其可信邻居节点 $\{n_2, n_3, n_5\}$ 中选择权值最小边 $w_{2,4}$ 加入树形拓扑。同理可知, 不可信节点 n_6 选择 $w_{0,6}$ 加入树形拓扑, 从而完成区块链网络的树形拓扑构建。

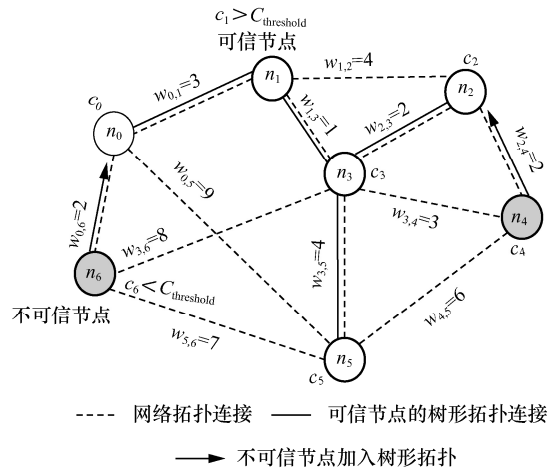


图 2 构建网络的树形拓扑示例

通过树形拓扑构建方法将原本的区块链无结构 P2P 网络拓扑构建为树形拓扑, 极大地简化了网络的拓扑结构。树形拓扑结合了可信值和最小生成树的优势, 并且将不可信节点作为树的叶子节点, 很大程度上提高了数据传输过程中的安全性, 而且由于树形拓扑具有更小的权值之和, 因此有效地缩短了区块链整体的数据传输时间。

1.2 转发路径选择策略

区块链采用 P2P 网络组织节点之间进行数据传输, 每个节点都独立承担验证数据、网络路由、转发数据和发现新节点等任务^[15]。本文采用并发传输机制, 即获得数据的不同节点可以在同一时间段内转发数据, 实现不同节点之间的并发传输。节点通过并发传输机制进行数据传输花费的时间称为并发传输时间。基于并发传输机制, 本文设计了一种转发路径选择策略, 指导节点选择邻居节点进行数据转发。该策略首先为树形拓扑中的每个节点建立关于其邻居节点的转发表, 然后设置节点转发数据的规则。由并发传输过程构建孩子-兄弟通信树示例如图 3 所示。

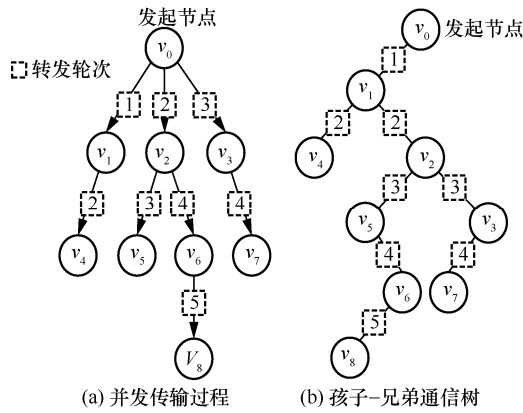


图 3 由并发传输过程构建孩子-兄弟通信树示例

设 $Tree = (V, E, W)$ 是一个树形拓扑，并发传输过程如图 3(a)所示。图 3(a)中虚线框内的数字为转发轮次，表示节点经过第几轮次转发后接收到数据。从 v_0 开始转发数据，下面以节点间的前两轮转发为例介绍并发传输。在第一轮转发中， v_0 将数据传输给 v_1 。在第二轮转发中，由于 v_1 已经从 v_0 接收到数据，因此 v_1 将数据转发给下一个子节点 v_4 ， v_0 也继续向 v_2 传输数据。

为了研究节点如何选择邻居节点转发数据使整个数据传输过程花费的并发传输时间最小，本文引入孩子-兄弟通信树，如图 3(b)所示。图 3(b)中的孩子-兄弟通信树是根据图 3(a)中的并发传输过程建立的，节点 v_0 的深度为 0。以节点 v_2 为例介绍孩子-兄弟通信树的构建过程如下。由于在图 3(a)中 v_0 先将数据转发给 v_1 ，那么在图 3(b)中 v_1 为 v_0 的左孩子。之后 v_0 转发数据给 v_2 ，那么在图 3(b)中 v_2 为 v_1 的右兄弟。孩子-兄弟通信树中节点的深度代表节点在并发传输过程中第几轮次接收到数据，即节点并发传输的转发轮次。在孩子-兄弟通信树中，计算某个支路的通信时间只需计算从发起节点到该支路最后一个节点所经过的边的权值之和。由此借助孩子-兄弟通信树把研究从发起节点传输数据到全网节点花费的最小并发通信时间这个问题转化为最小化孩子-兄弟通信树中的支路最大传输时间问题。其中，支路最大传输时间是指在所有支路传输时间中并发传输时间最大的支路时间。当节点按照它的邻居节点（子节点）所在子树的最小并发传输时间由大到小顺序选择邻居节点转发数据时，能使从发起节点传输数据到全网节点花费的并发传输时间最小，证明过程如附录 1 所示。基于上述思想，本文为每个节点建立各自的转发表，转发表中按照最小并发传输时间由大到小的顺序，依

次存储节点所有邻居节点的名称和邻居节点所在子树的最小并发传输时间。转发表的建立过程如下。首先随机选择一个节点作为根节点，得到以该节点为根节点的树形拓扑；然后，经过自底向上和自顶向下 2 个过程建立每个节点的转发表。

1) 自底向上

自底向上建立节点转发表的过程是以最小整体并发传输时间为目标，从树形拓扑的叶子节点逐步向根节点方向计算节点的子节点所在子树的最小并发通信时间，子节点向父节点发送计算结果，父节点存储结果并依次计算以及向上发送结果，直到根节点为止。由于叶子节点没有子节点，因此叶子节点所在子树的最小并发传输时间为 0，将 0 发送给它的父节点，父节点按照最小并发传输时间由大到小的顺序，存储该叶子节点名称和它所在子树的最小并发传输时间 0。针对节点是非叶子节点的情况，在接收到所有子节点发送的它们各自所在子树的最小并发传输时间后，根据它的转发表和权值计算该非叶子节点所在子树的最小并发传输时间。将计算结果发送给它的父节点，父节点按照由大到小顺序进行存储。在同一转发表中，如果存在多个子节点的最小并发传输时间相同，那么子节点所在子树连通度大的子节点优先存储；如果连通度相同，则优先存储与节点相连权值小的子节点。当根节点转发表建立完成时，则自底向上建立转发表的过程结束。转发表建立示例如图 4 所示。

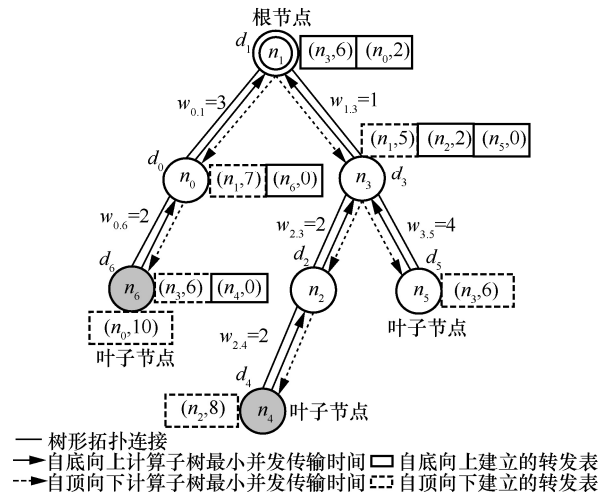


图 4 转发表建立示例

随机选择树形拓扑中的 n_1 作为根节点，以自底向上建立 n_1 的转发表为例介绍该过程。由于 n_4 、 n_5 和 n_6 是叶子节点，因此其子树的最小并发传输时间

为 0, 分别将节点名称和 0 发送给它们的父节点 n_3 、 n_2 和 n_0 。 n_2 收到子节点 n_4 的信息按由大到小存储到转发表, 然后 n_2 计算其所在子树的最小并发传输时间为 $w_{2,4} = 2$, 并将 n_2 和 2 发送给 n_3 。 n_3 按照最小并发传输时间由大到小的顺序存储到转发表。在 n_3 收到子节点 n_2 和 n_5 的信息之后, n_3 根据转发表的转发顺序构建以 n_3 为根节点, 以 n_2 和 n_5 为子节点的通信树, 类似于图 3(b), 计算支路最大传输时间, 得到 n_3 所在子树的最小并发传输时间 6, 发送给 n_1 节点并排序。同理可得 n_0 所在子树的最小并发传输时间, n_1 节点的转发表建立完成。

2) 自顶向下

当自底向上建立转发表的过程结束后, 除根节点外的其他节点的转发表中只缺少一项由父节点作为子节点时其所在子树的最小并发传输时间。因此, 采用自顶向下过程补全其他节点的转发表。自顶向下建立转发表的过程是以最小整体并发传输时间为目标, 根据每个节点转发表中已存储的节点信息, 从树形拓扑的根节点逐步向叶子节点方向计算将父节点看作子节点时其所在子树的最小并发传输时间, 完成每个节点转发表的建立。

以图 4 中的 n_3 为例介绍自顶向下转发表的建立过程。 n_3 的父节点是 n_1 , 将父节点 n_1 看作 n_3 的子节点时, n_1 所在子树包含节点 n_1 、 n_0 和 n_6 , 计算该子树的最小并发传输时间为 $w_{0,1} + W_0 = 5$, 其中 W_0 是 n_1 转发表中存储的 n_0 所在子树的最小并发传输时间, 将 n_1 和 5 发送给 n_3 节点并排序, 如图 4 中 n_3 的虚线方框所示, 此时 n_3 得到完整的转发表。同理可得其他节点的完整转发表。

节点转发数据的规则是将发起交易数据的节点指定为发起节点, 并作为树形拓扑的根节点, 从根节点开始进行数据转发, 直到树中所有节点接收到数据为止。如果节点是发起节点, 那么节点完全按照转发表依次选择邻居节点进行数据的转发即可。如果节点是转发节点, 不再向给它发送数据的节点转发数据, 按照转发表依次选取其余节点转发数据。本文提出的转发路径选择策略具有最小整体并发传输时间, 有效缩短了数据传输时间, 同时避免对节点重复传输同一数据, 极大提高了区块链的传输效率。

1.3 拓扑动态优化策略

区块链是由节点自主管理和维护的一种 P2P 网

络。在 P2P 网络中, 每个节点都可以随时加入或退出网络^[16]。区块链节点的加入和离开网络等行为会引起网络拓扑发生变化, 本文利用提出的拓扑动态优化策略局部调整拓扑, 在一定程度上减小因拓扑动态变化而引起的额外传输时间, 从而减少区块链出现分叉的情况, 确保区块链账本的唯一性和一致性。拓扑动态优化策略作用于区块链网络层, 对区块链网络层的改进不会影响区块链共识层使用共识算法来保障区块链账本的唯一性与一致性。针对树形拓扑存在节点加入行为、离开行为、可信值变化以及节点间权值发生变化的情况, 本文设计了相应的拓扑动态优化策略, 只对状态变化节点的局部范围进行拓扑调整, 避免重构整个网络的拓扑。下面介绍该策略使用的 2 个算法以及提出的 4 个子拓扑动态优化策略。拓扑动态优化示例如图 5 所示。

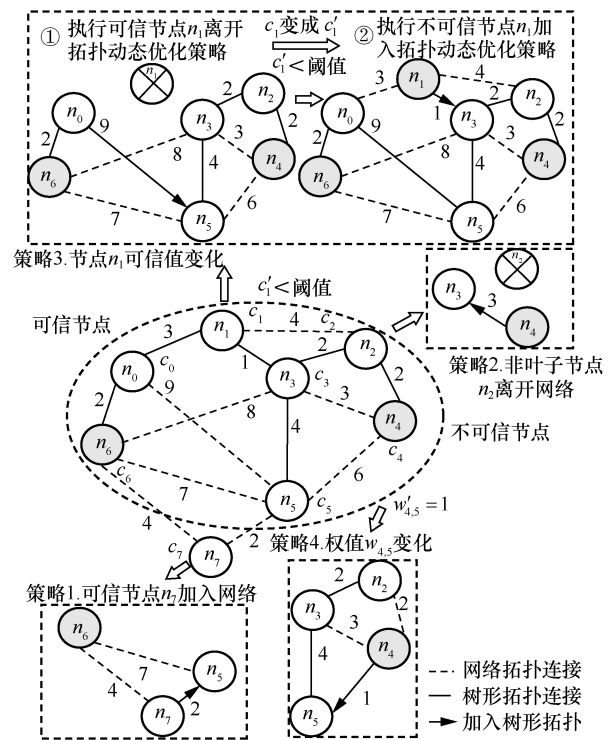


图 5 拓扑动态优化示例

破圈法。对于存在回路的树形拓扑, 每次删除回路中权值最大的边, 同时确保树形拓扑是连通的, 直到树形拓扑不存在回路。

Sollin 算法。对于目前存在的多棵子树, 依次选取每棵子树与其他子树相连的权值最小边, 然后合并子树, 直到构成一棵最小生成树。

策略 1 节点加入拓扑动态优化策略

1) 可信节点加入拓扑动态优化策略是选取该节

点与可信邻居节点相连的所有边加入树形拓扑, 此时树形拓扑可能存在回路, 若所有边加入树形拓扑存在回路, 则对树形拓扑采用破圈法, 得到可信节点加入后的树形拓扑。图 5 的可信节点 n_7 加入网络时, 选取 n_7 与可信邻居节点 n_5 相连的边加入树形拓扑, 由于树形拓扑没有回路, 因此直接得到 n_7 加入后的树形拓扑。

2) 不可信节点加入拓扑动态优化策略是从不可信节点与可信邻居节点相连的所有边中选取权值最小边, 将该边加入树形拓扑, 得到不可信节点加入后的树形拓扑。

策略 2 节点离开拓扑动态优化策略

1) 叶子节点离开拓扑动态优化策略是删除该节点以及与其他节点的相连边。

2) 针对树形拓扑中非叶子节点离开行为, 采取的节点离开拓扑动态优化策略如下所述。在删除离开的非叶子节点以及与其他节点的相连边之后, 树形拓扑分裂为多棵不连通的子树, 这些子树的构成分为 2 种情况, 一种是由多个节点构成, 另一种是仅由一个节点构成。非叶子节点离开拓扑动态优化策略如下。首先, 采用 Sollin 算法对由第一种情况构成的多棵不连通的子树进行树形拓扑的构建, 对新选取的边进行判断。如果新选取的边包含不可信节点, 则该边不能作为树形拓扑的边, 继续对不连通的子树运行 Sollin 算法, 直到选取的边不含不可信节点。然后, 对第二种情况的子树进行处理, 选取这个子树节点与其可信邻居节点直接相连的所有边中的权值最小边, 将该边加入第一种情况更新后的树形拓扑, 更新树形拓扑。

图 5 中, 非叶子节点 n_2 离开拓扑优化策略是从网络中删除 n_2 以及与其他节点的相连边, 采用 Sollin 算法对由多个节点构成的多棵不连通子树构建树形拓扑。因为只有一棵由多个节点 $\{n_0, n_1, n_3, n_5, n_6\}$ 构成的连通子树, 所以不需要利用 Sollin 算法进行构建。将由一个节点 n_4 构成的子树加入由节点 $\{n_0, n_1, n_3, n_5, n_6\}$ 构成的树形拓扑, 选取 n_3 和 n_4 相连的边作为树形拓扑的边。

策略 3 节点可信值变化拓扑动态优化策略

节点的可信值变化可能影响节点的属性, 例如, 由可信节点变为不可信节点, 不可信节点变为可信节点, 从而引起树形拓扑结构的改变。

1) 当节点由可信节点变为不可信节点时, 判断该节点是否为叶子节点。如果该节点是叶子节点, 则

其拓扑结构不发生变化, 节点属性变为不可信节点。如果该节点不是叶子节点, 那么先对该可信节点执行策略 2, 然后对可信值变化后的不可信节点执行策略 1, 更新树形拓扑, 如图 5 中的 n_1 可信值变化过程所示。

2) 当节点由不可信节点变为可信节点时, 对该节点的可信邻居节点和不可信邻居节点采取不同的调整策略。首先, 将可信邻居节点与该节点连接的所有边加入树形拓扑, 运行破圈法。然后, 对该节点的所有不可信邻居节点先后执行策略 2 和策略 1, 得到节点可信值变化后的树形拓扑。

策略 4 节点间权值变化拓扑动态优化策略

1) 如果节点之间权值变化的边是树形拓扑的边并且权值变小, 或该边不是树形拓扑的边并且权值变大, 则拓扑结构不变, 只更新该边的权值。

2) 针对权值变化边连接的 2 个节点是可信节点情况, 如果该边是树形拓扑的边并且权值变大, 更新该边的权值, 在树形拓扑中删除该边, 对分裂后的多棵子树采用 Sollin 算法选择新连接的边。若新选取的边连接的节点都是可信节点, 那么选择该边连接分裂的子树, 更新树形拓扑。若新选取的边连接的节点存在不可信节点, 则选择原先删除的边来连接分裂的子树, 此时树形拓扑结构不变。

如果该边不是树形拓扑的边并且权值变小, 更新区块链网络拓扑中该边的权值, 将该边加入树形拓扑, 对构成回路的节点运行破圈法, 更新树形拓扑。

3) 针对权值变化边连接的 2 个节点存在不可信节点情况, 如果该边是树形拓扑的边并且权值变大, 更新该边的权值, 在树形拓扑中删除该边, 对分裂后的多棵子树采用 Sollin 算法选择新连接的边。若新选取的边连接的节点只有一个不可信节点, 那么选择该边连接分裂的子树, 更新树形拓扑。若新选取的边连接的节点都是不可信节点, 则选择之前删除的边来连接分裂的子树, 此时树形拓扑结构不变。

如果该边不是树形拓扑的边并且权值变小, 更新区块链网络拓扑中该边的权值, 删除该边所连接的不可信节点在树形拓扑中的边, 再对该节点执行策略 1, 更新树形拓扑。图 5 中 n_4 和 n_5 之间边的权值 $w_{4,5}$ 变小, 更新区块链网络拓扑中该边的权值为 $w'_{4,5}$, 删除该边连接的不可信节点 n_4 在树形拓扑的边, 即 n_2 和 n_4 之间的边, 再对不可信节点 n_4 执行策略 1, 选择 n_4 和 n_5 之间的边作为树形拓扑的边, 更新树形拓扑。

2 算法设计与实现

区块链传输效率优化方法通过树形拓扑构建方法生成树形拓扑, 利用转发路径选择策略建立转发路径, 采用拓扑动态优化策略调整局部拓扑, 从而缩短区块链数据传输时间, 提高数据传输效率。

2.1 树形拓扑构建算法

可信拓扑是树形拓扑构建的前提, 树形拓扑构建算法利用 Sollin 算法对可信拓扑构建最小生成树, 之后将不可信节点作为最小生成树的叶子节点加入树形拓扑。具体的算法流程如算法 1 所示。

算法 1 树形拓扑构建算法

输入 可信拓扑 $G_T(V_T, E_T, W_T)$, 不可信节点集 U

- 1) 定义森林 F , 森林由多棵不连通的树构成, 初始化 F 为 G_T 的每个节点
- 2) while 森林 F 中树的个数 > 1 do
- 3) 根据可信拓扑 G_T , 为森林 F 中每棵树选择权值最小边
- 4) if 权值最小边连接的是 2 棵不同树 then
- 5) 合并 2 棵树, 将权值最小边加入森林 F , 加入后不构成回路
- 6) end if
- 7) end while
- 8) for node in U do
- 9) 选择 node 相连的可信节点的权值最小边, 将 node 和该边加入 F
- 10) end for

2.2 转发路径选择算法

根据转发路径选择策略, 设计转发路径选择算法, 算法初始时随机指定树形拓扑的某一节点作为根节点, 然后按照自底向上和自顶向下 2 个过程建立关于节点的邻居节点转发表, 最后根据转发表和转发数据规则进行数据的转发。转发路径选择算法如算法 2 所示。

算法 2 转发路径选择算法

输入 树形拓扑 $T(V, E, W)$

- 1) 随机选择 T 的一个节点作为根节点 root, T 转换为以 root 为根节点的树
- 2) 定义 T 的每个节点的转发表 S 、节点的父节点集 P 和子节点集 RC , 初始化 P 为叶子节点的父节点, RC 为根节点的子节点以及叶子节点的最小并发传输时间为 0

- 3) for parent in P do
- 4) if parent \neq root 且 parent 接收所有子节点的最小并发传输时间 then
- 5) 计算以 parent 为根的子树最小并发传输时间, 更新 S , parent 的父节点加入 P
- 6) end if
- 7) if parent=root 且 parent 接收所有子节点的最小并发传输时间 then
- 8) for child in RC do
- 9) 计算将 child 父节点作为 child 子节点时最小并发传输时间, 更新 S , child 子节点加入 RC
- 10) end for
- 11) end if
- 12) end for

2.3 拓扑动态优化算法

本文针对节点的加入行为、离开行为、可信值变化和节点间权值变化情况设计了拓扑动态优化策略, 根据该策略设计了拓扑动态优化算法, 算法主要利用破圈法和 Sollin 算法实现对状态变化节点拓扑的局部调整。拓扑动态优化算法如算法 3 所示。

算法 3 拓扑动态优化算法

输入 树形拓扑 $T(V, E, C, W)$, 网络拓扑 $G(V, E, C, W)$

- 1) if 节点加入网络 then
- 2) if 节点是不可信节点 then
- 3) 节点选择相连可信节点的权值最小边加入拓扑 T 和 G
- 4) else 针对节点及与网络中相连的节点和边采用破圈法, 加入拓扑 T 和 G
- 5) end if
- 6) end if
- 7) if 节点离开网络 then
- 8) 在 T 和 G 中删除离开节点, 对生成的森林运行 Sollin 算法
- 9) end if
- 10) if 节点的可信值发生变化 then
- 11) if 节点由可信节点变为不可信节点 then
- 12) 针对节点运行节点离开算法, 利用节点加入算法将该节点加入拓扑 T 和 G
- 13) end if
- 14) if 节点由不可信节点变为可信节点 then
- 15) 将节点与可信节点相连的所有边

```

    加入  $G$  并运行破圈法, 更新拓扑  $T$  和  $G$ 
16) end if
17) end if
18) if 节点间的权值发生变化 then
19)   if 权值变化边的 2 个节点是可信节点 then
20)     if 权值变化边是  $T$  中的边 and 权值变
        大 then
21)       更新  $G$  中该边的权值, 删除  $T$  中变
        化的边, 对  $T$  运行 Sollin 算法, 将
        选择的边加入  $T$ 
22)     if 选择的边连接的 2 个节点都为可信
        节点 then
23)       添加该边, 更新拓扑  $T$  和  $G$ 
24)     else 更新拓扑  $T$  和  $G$  中权值变化边
        的权值
25)     end if
26)   end if
27)   if 权值变化边不是  $T$  中的边 and 权值
        变小 then
28)     该边加入  $T$ , 对  $T$  运行破圈法, 更
        新拓扑  $T$  和  $G$ 
29)   end if
30)   else if 权值变化边是  $T$  中的边 and 权值
        变大 then
31)     更新  $G$  中该边的权值, 删除  $T$  中变
        化的边, 对  $T$  运行 Sollin 算法, 将
        选择的边加入  $T$ 
32)   end if
33)   if 权值变化边不是  $T$  中的边 and 权值
        变小 then
34)     根据该边连接的不可信节点, 选取
        它与可信节点相连的权值最小边
        加入, 更新拓扑  $T$  和  $G$ 
35)   end if
36) end if
37) end if

```

3 仿真分析

为了验证所提方法和策略的有效性, 本文使用 Python 进行仿真实验, 使用 Python 的 networkx 库生成网络拓扑。实验设置构建网络拓扑的节点个数为从 100 到 1 000 的等步长取值, 节点之间的权值设置为 [1,50] 随机取值, 权值的单位为 s。可信值阈

值用于划分节点类型, 当节点传输成功率小于 0.6 时, 该节点划分为不可信节点^[10]。将其值代入式(1)计算得出可信值阈值为 1.2。为了避免节点同时与多个邻居节点进行数据传输时, 一些节点受自身性能的限制无法同时传输给多个节点, 从而导致数据丢失问题, 设定节点在同一时间段内只能与一个节点进行数据传输。本文选取平均并发传输时间和转发次数作为实验指标, 将本文提出的转发路径选择算法与权重优先算法^[9]、随机路径算法^[10]进行对比。另外考虑到网络中节点状态变化情况, 本文选取拓扑调整时间比和节点的参与率作为实验指标, 将本文提出的拓扑动态优化算法与全网重构拓扑算法进行对比。本文的仿真参数设置如表 1 所示。

表 1 仿真参数设置

参数	数值
网络拓扑	小世界网络拓扑
重连概率	0.8
邻居节点数/个	5
通信节点数/个	1
节点可信值	[0, 2]
权值	[1, 50]
可信值阈值	1.2

权重优先算法是通过优先选取与当前节点相连的权重最小节点, 按照左孩子右兄弟原则构造通信树。随机路径算法是指节点随机选择除向它发送数据的节点之外的其他邻居节点进行数据转发。本文提出的转发路径选择算法与权重优先算法、随机路径算法在不同网络节点数量下的平均并发传输时间如图 6 所示。其中, 平均并发传输时间是指将每个节点作为数据发起节点计算数据传输到全网节点的并发传输时间之和的平均值。由于随机路径算法随机选择的邻居节点很大概率不是最优节点, 因此具有最高平均并发传输时间。权重优先算法在逐次选择节点加入通信树时, 只选择与当前节点具有最小权重的节点加入, 但是整体通信树的并发传输时间不一定是最小的。本文算法综合考虑了转发节点的子节点和子节点所在子树的最小并发传输时间两方面, 并且以全局最优为目标选择节点进行数据的转发。因此, 转发路径选择算法具有最小的平均并发传输时间, 较大程度地减少了数据的并发传输时间, 提高了数据传输效率。

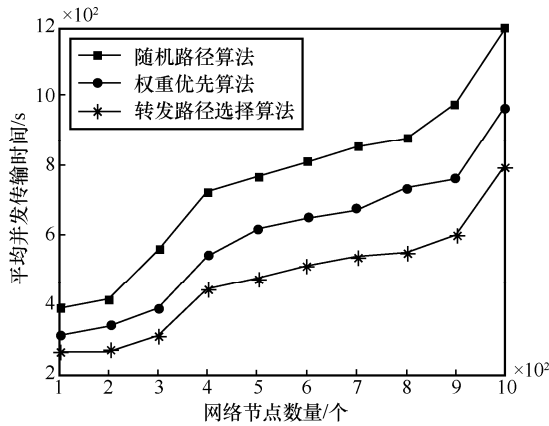


图 6 不同网络节点数量下的平均并发传输时间

本文提出的转发路径选择算法与权重优先算法、随机路径算法在不同网络节点数量下的转发次数如图 7 所示。其中，转发次数是指从发起节点开始转发数据直到网络中最后一个节点接收到数据的总轮次。转发次数等同于图 3(b)孩子-兄弟通信树的路径长度，那么转发次数越少，路径长度越短。由于转发路径选择算法相比于随机路径选择算法和权重优先算法，具有较短的传输路径。因此，从图 7 可以看出，本文提出的转发路径选择算法的转发次数一直是最少的，而权重优先算法和随机路径算法的转发次数较多。随着网络节点数量的增加，参与数据转发的节点数量逐渐增多，所以 3 种算法的转发次数呈逐渐上升趋势。但是本文提出的转发路径选择算法仍明显优于其他 2 种算法。因此，使用转发路径选择算法完成整个区块链网络的数据传输只需较少的转发次数。

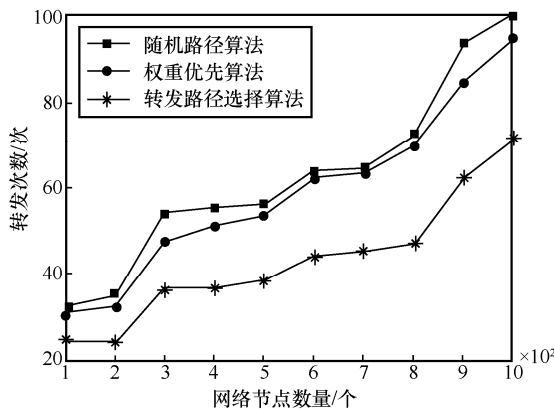


图 7 不同网络节点数量下的转发次数

当节点加入区块链网络时，节点与其邻居节点建立拓扑连接，之后利用拓扑动态优化算法加入树形拓扑，网络规模扩大。当节点离开网络时，可能

引起树形拓扑不连通，利用拓扑动态优化算法将其调整为连通的树形拓扑，网络规模减小。当节点的可信值以及节点之间的权值发生变化时，可能导致一些节点之间的树形拓扑结构发生变化，利用拓扑动态优化算法局部调整树形拓扑，网络规模不变。不同网络节点数量下的拓扑调整时间比如图 8 所示，拓扑调整时间比是指采用全网重构拓扑算法与拓扑动态优化算法调整拓扑花费时间的比值。全网重构拓扑算法是指当已生成的树形拓扑中的节点状态变化时，利用树形拓扑构建算法为全网节点建立新的树形拓扑。由图 8 可知，采用节点间权值变化拓扑动态优化算法调整变化节点的拓扑结构的效果最好。这是因为节点间权值变化拓扑动态优化算法是对权值发生变化的边进行调整，存在权值发生变化时不需要调整节点之间的拓扑结构，只更新边的权值的情况。此外，节点加入、节点离开和节点可信值变化拓扑动态优化算法的拓扑调整时间约为全网重构拓扑算法的 $\frac{1}{2}$ 。因此，针对状态发生变化的节点，采用拓扑动态优化算法实现拓扑调整花费的时间较少，从而提高了数据的传输效率。

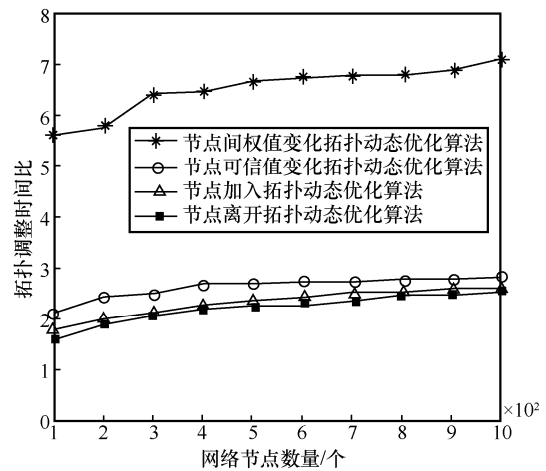


图 8 不同网络节点数量下的拓扑调整时间比

本文提出的拓扑动态优化算法与全网重构拓扑算法在不同网络节点数量下的节点参与率如图 9 所示，节点参与率是指参与调整拓扑的节点个数占网络总节点数的百分比。参与拓扑结构调整的节点个数越少，说明节点状态变化影响的范围越小，则调整拓扑结构花费的时间越少。由图 9 可知，采用拓扑动态优化算法的节点参与率低于 20%，显然优于全网重构拓扑算法，因为拓扑动态优化算法只调整变化节点影响的小范围局部拓扑结构，不对其他

节点的拓扑结构进行改动。节点离开和节点可信值变化拓扑动态优化算法的节点参与率略高于其他 2 种拓扑动态优化算法，这是因为节点离开拓扑动态优化算法需要删除离开节点，然后为不连通的多棵子树进行多轮选边，而节点可信值变化拓扑动态优化算法利用节点离开拓扑动态优化算法对节点和拓扑进行处理。因此，本文提出的拓扑动态优化算法可以减少调整网络拓扑所带来的额外开销，加快数据之间的传输。

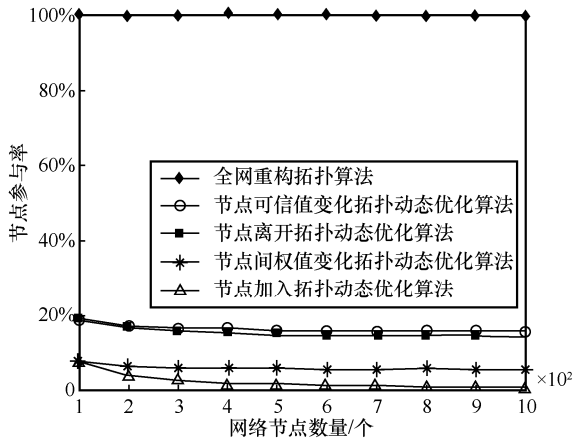


图 9 不同网络节点数量下的节点参与率

4 结束语

本文提出的区块链传输效率优化方法由树形拓扑构建方法、转发路径选择策略和拓扑动态优化策略构成。通过构建的树形拓扑可以简化网络的拓扑结构。转发路径选择策略根据转发表指导节点进行数据转发，从而缩短数据传输时间。拓扑动态调整策略针对状态变化节点局部受影响的拓扑进行调整，减少重构全网拓扑所带来的额外开销。下一步工作考虑对区块链网络节点进行分簇，为节点设计一种基于分簇的数据传输策略，以加快区块链数据的传输，更好地适应于大规模的区块链网络。

附录 1 最小并发传输时间证明

证明 当节点按照其子节点所在子树的最小并发传输时间由大到小的顺序转发数据时，整体具有最小并发传输时间，即证按照此转发顺序构建的通信树的支路最大传输时间是最小的。

假设根节点 n_0 的 n 个子节点所在子树的最小并发传输时间为 $W_1 \geq W_2 \geq \dots \geq W_n$ ， w_b 是 n_0 与子节点之间的传输时间 ($1 \leq b \leq n$)。以根节点按照其子节点所在子树的最小并发

传输时间由大到小顺序转发数据为基础，假设并发传输时间最大支路中的子节点为 n_x ，支路最大传输时间为

$$T = T_x = W_x + w_1 + \dots + w_x \quad (3)$$

任意调换 2 个子节点 n_i 和 n_j 的转发顺序后的支路最大传输时间为 T' ，其中， $i < j$ 且 $W_i \geq W_j$ 。

在调换 n_i 和 n_j 转发顺序后， n_i 和 n_j 所在支路的并发传输时间为

$$T'_i = W_i + w_1 + \dots + w_i + \dots + w_j \quad (4)$$

$$T'_j = W_j + w_1 + \dots + w_{j-1} + w_j \quad (5)$$

已知 $W_i \geq W_{i+1} \geq \dots \geq W_j$ ， $\{n_i, n_{i+1}, \dots, n_j\}$ 中节点所在支路的支路最大传输时间为 $\max\{T'_i, T'_{i+1}, \dots, T'_j\} = T'_i$ 。已知 n_x 为调换前并发传输时间最大支路中的节点，下面根据 x 的取值情况进行讨论。

1) $1 \leq x < i$ 或 $j < x \leq n$ 。除 $\{n_i, n_{i+1}, \dots, n_j\}$ 之外其他所有子节点所在支路中的支路最大传输时间为 $\max\{T_1, \dots, T_{i-1}, T_{j+1}, \dots, T_n\} = T$ 。若 $T'_i \leq T$ ，则调换 n_i 和 n_j 转发顺序后的支路最大传输时间 $T' = T$ 。若 $T'_i > T$ ，则调换 n_i 和 n_j 转发顺序后的支路最大传输时间 $T' = T'_i > T$ 。故 $T' \geq T$ 。

2) $i \leq x \leq j$ 。由于调换顺序后，除了 $\{n_i, n_{i+1}, \dots, n_j\}$ 之外其他所有子节点所在支路的传输时间与调换顺序之前的时间相同，并且 T_x 是调换顺序之前的支路最大传输时间，因此这些节点调换顺序后的支路最大传输时间为 $\max\{T'_1, \dots, T'_{i-1}, T'_{j+1}, \dots, T'_n\} < T_x$ 。由式(3)和式(4)可知， $T = T_x$ ， $T'_i > T_x$ ，则调换 n_i 和 n_j 转发顺序后的支路最大传输时间 $T' = T'_i > T$ 。故 $T' > T$ 。

综上所述，任意调换 2 个子节点转发顺序使子节点不按照由大到小顺序转发数据的支路最大传输时间 $T' \geq T$ 。因此，当节点按照其子节点所在子树的最小并发传输时间由大到小的顺序转发数据时，整体具有最小并发传输时间。

证毕。

参考文献：

- [1] 郭才, 李续然, 陈炎华, 等. 区块链技术在物联网中的应用概述[J]. 物联网学报, 2021, 5(1): 72-89.
GUO C, LI X R, CHEN Y H, et al. Blockchain technology for Internet of things: an overview[J]. Chinese Journal on Internet of Things, 2021, 5(1): 72-89.
- [2] 曾诗钦, 霍如, 黄韬, 等. 区块链技术研究综述: 原理、进展与应用[J]. 通信学报, 2020, 41(1): 134-151.
ZENG S Q, HUO R, HUANG T, et al. Survey of blockchain: principle, progress and application[J]. Journal on Communications, 2020, 41(1): 134-151.

- [3] ZHOU Q H, HUANG H W, ZHENG Z B, et al. Solutions to scalability of blockchain: a survey[J]. IEEE Access, 2020, 8: 16440-16455.
- [4] CUI L Z, YANG S, CHEN Z T, et al. An efficient and compacted DAG-based blockchain protocol for industrial Internet of things[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4134-4145.
- [5] XIAO Y, ZHANG N, LOU W J, et al. A survey of distributed consensus protocols for blockchain networks[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 1432-1465.
- [6] BHUTTA M N M, KHWAJA A A, NADEEM A, et al. A survey on blockchain technology: evolution, architecture and security[J]. IEEE Access, 2021, 9: 61048-61073.
- [7] ROHRER E, TSCHORSCH F. Blockchain layer zero: characterizing the bitcoin network through measurements, models, and simulations[C]//Proceedings of 2021 IEEE 46th Conference on Local Computer Networks. Piscataway: IEEE Press, 2021: 9-16.
- [8] SUNG F Y, HAN H C, FU W Y, et al. Adaptive broadcast routing assignment algorithm for blockchain synchronization services[C]//Proceedings of 2018 10th International Conference on Ubiquitous and Future Networks (ICUFN). Piscataway: IEEE Press, 2018: 487-492.
- [9] LI J, CHEN L. A concurrent weighted communication scheme for blockchain transaction[C]//Proceedings of 2020 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs). Piscataway: IEEE Press, 2020: 177-182.
- [10] LIN C H, CHEN W Y, LIN P S. A decentralized tree-based algorithm for reliable blockchain communications[C]//Proceedings of 2020 International Symposium on Networks, Computers and Communications (ISNCC). Piscataway: IEEE Press, 2020: 1-6.
- [11] ERSOY O, REN Z J, ERKIN Z, et al. Transaction propagation on permissionless blockchains incentive and routing mechanisms[C]//Proceedings of 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). Piscataway: IEEE Press, 2018: 20-30.
- [12] SANTIAGO C, LEE C. Accelerating message propagation in blockchain networks[C]//Proceedings of 2020 International Conference on Information and Communication Technology Convergence (ICTC). Piscataway: IEEE Press, 2020: 157-160.
- [13] HAO W F, ZENG J J, DAI X H, et al. Towards a trust-enhanced blockchain P2P topology for enabling fast and reliable broadcast[J]. IEEE Transactions on Network and Service Management, 2020, 17(2): 904-917.
- [14] LEI K, ZHANG Q C, XU L M, et al. Reputation-based Byzantine fault-tolerance for consortium blockchain[C]//Proceedings of IEEE 24th International Conference on Parallel and Distributed Systems. Piscataway: IEEE Press, 2018: 604-611.
- [15] YANG J M, BI H, LIANG Z H, et al. A survey on blockchain: architecture, applications, challenges, and future trends[C]//Proceedings of International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics. Piscataway: IEEE Press, 2020: 749-754.
- [16] WU M L, WANG K, CAI X Q, et al. A comprehensive survey of blockchain: from theory to IoT applications and beyond[J]. IEEE Internet of Things Journal, 2019, 6(5): 8114-8154.

[作者简介]



霍如 (1988-), 女, 黑龙江哈尔滨人, 博士, 北京工业大学讲师, 主要研究方向为未来网络、工业互联网、边缘计算、网络资源管理、区块链等。

程祥凤 (1997-), 女, 山东潍坊人, 北京工业大学硕士生, 主要研究方向为区块链、工业互联网、边缘计算等。

孙闯 (1989-), 男, 黑龙江哈尔滨人, 博士, 清华大学助理研究员, 主要研究方向为先进传感技术与能源互联网等。

汪硕 (1991-), 男, 河南灵宝人, 博士, 北京邮电大学讲师, 主要研究方向为数据中心网络、软件定义网络、网络流量调度等。

黄韬 (1980-), 男, 重庆人, 博士, 北京邮电大学教授, 主要研究方向为未来网络体系架构、软件定义网络、网络虚拟化等。

F. Richard Yu (1974-), 男, 加拿大卡尔顿大学教授、加拿大工程院院士, 主要研究方向为互联网自主智能、自动驾驶、网络空间安全等。