第3章 区块链网络与跨链技术

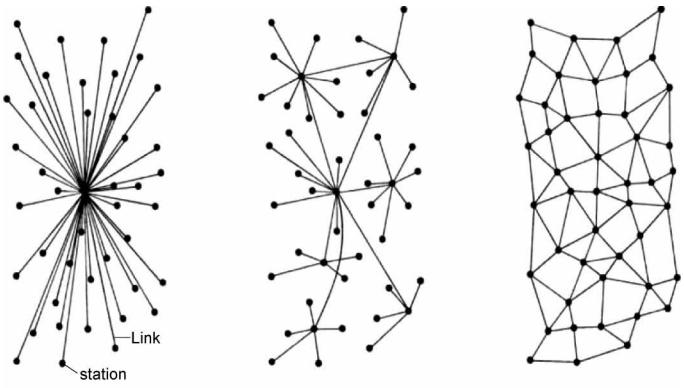
区块链的网络层封装了区块链系统的组网方式、数据传播协议和数据验证机制等要素。本章将重点讨论区块链底层的网络结构与通信协议，以及实现多条区块链之间互操作的跨链技术。

3.1 区块链网络

区块链系统通过其内置的激励机制组织起大量分布式的计算节点来共同完成特定的计算任务，并共同维护分布式数据账本的安全性、一致性和不可篡改性。从这种意义上来说，区块链符合分布式系统的狭义定义，即“网络连接的计算机系统中，每个节点独立地承担计算或者存储任务，节点间通过网络通信协同工作”。从广义角度来看，分布式系统的判定取决于观察者的视角。正如莱斯利·兰伯特(Leslie Lamport)所言，“对于坐在键盘前的用户来说，他的 IBM 计算机并非分布式系统，然而对于在计算机电路板上的跳蚤或者设计该电路板的工程师来说，这台计算机又是典型的分布式系统”。区块链亦是如此，从微观计算节点的角度来看是分布式系统，而从宏观视角来看，大量区块链节点通过共识算法对外提供统一的服务，因而亦可视为非分布式系统。本书旨在探究区块链系统的内在微观运行机理，因而认为区块链是一类典型的分布式系统。

区块链的网络结构继承了计算机通信网络的一般拓扑结构，可以分为如图 3-1 所示的中心化网络、多中心化网络和去中心化网络三类。一般来说，以比特币和以太坊为代表的非授权区块链(Permissionless Blockchain)大多采用图 3-1(c)所示的去中心化网络，其网络节点一般具有海量、分布式、自治、开放可自由进出等特性，因而大多采用对等网络(Peer-to-Peer Network，P2P 网络)来组织散布全球的参与数据验证和记账的节点。P2P 网络中的每个节点都地位对等且以扁平式拓扑结构相互连通和交互，不存在任何中心化的特殊节点和层级结构，每个节点均会承担网络路由、验证区块数据、传播区块数据、发现新节点等功能；随着区块链技术的发展，近年来有些区块链系统尝试采用 Mesh 网络(即网状网)来组织区块链的计算节点。与非授权区块链相比，授权区块链(Permissioned Blockchain)系统大多采用图 3-1(a)和图 3-1(b)所示的中心化星型网络或者多中心网络结构，例如联盟链大多采

用多中心网络，而私有链则可能采用完全中心化的星型网络。



(a) 中心化　　(b) 多中心化　　(c) 去中心化

图 3-1　区块链系统的网络拓扑结构示意图

许多区块链书籍和文章都将图 3-1(b)和图 3-1(c)分别称为去中心化和分布式网络。事实上，这种说法(以及图 3-1 本身)源自于 1964 年美国兰德公司的一份有关分布式通信的报告“On Distributed Communications”。该报告认为大多数网络拓扑结构理论上都可以分解为“中心化星型网络”和“分布式网格或者网状网”，即图 3-1(a)和图 3-1(c)。这两类网络拓扑又可以通过组合构成所谓的“去中心化网络”(Decentralized Network)，例如图 3-1(b)所示的网络结构就是一系列星型结构和分布式结构组合而成的层级结构，这类网络的特点是并不总是依赖单一节点，然而破坏少量中心节点也可以摧毁整个网络的通信。显然，兰德报告的“去中心化网络”实际上是完全中心化网络和完全去中心化网络的中间状态，任何由完全中心化到完全去中心化网络的过渡状态实际上都是这两类网络形态的层次化组合。由此可见，兰德报告的“分布式网络”实际上就是中文语境中常见的去中心化网络。因此，本书认为将图 3-1 所示的三类网络拓扑分别称为中心化、多中心化和去中心化更为合适。

实际上，是否分布式与是否去中心化是两个不同的维度，分布式系统同样可以有中心化和去中心化两种设计和实现方式。此外，完全去中心化网络实际上是一种理想形态，在实际网络系统中较少存在，更多的是不同程度上的多中心化。例如，比特币网络的理论模型是完全去中心化网络，但实际比特币网络中的全节点(中心节点)数量远小于所有节点数量，造成事实上的多中心化。

3.1.1　P2P 网络

P2P 网络的全称为 Peer to Peer Network，即“点对点”或者“端对端”网络，学术界常称之为“对等网络”，其常见的定义是“一种分布式应用体系结构，用于在对等节点之间划分任务或负载。在网络中，对等节点拥有同等特权，就此形成一个点对点网络。对等节点将其部分资源(如处理能力、存储能力或网络带宽)直接提供给其他网络参与者，而不需要中央服务器进行集中协调”。P2P 网络中的参与者既是资源(内容和服务)提供者(Server)，又是资源获取者(Client)，如图 3-2(a)所示，网络节点以扁平拓扑结构相连，彼此交互运作、协同处理，网络整体可靠、开放、去中心化。

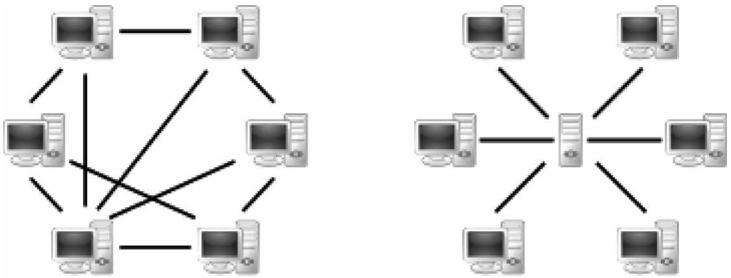


图 3-2　P2P 网络结构与 C/S 网络结构

P2P 模式一般是相对于中心化的客户端/服务器(Client/Server,简称 C/S)模式而言,C/S 模式的显著特点是一台服务器连接多个客户端,如图 3-2(b)所示。C/S 模式在实际系统中更为常见(Browser/Server 模式可以看作 C/S 模式的一种特例)。例如,微信朋友圈发布的图片和文字,首先会上传至微信的中央服务器,由服务器推送至相应好友。这种网络结构还可以类比为铁路网,中央服务器就是铁路网上的大型交通枢纽,管理着资源的调配与分发。随着网络规模的扩大,中央服务器模式的弊端日益凸显,这种方式最大的隐患在于中央服务器本身的单点故障问题。例如海量客户端请求会使中央服务器失效,近年来社会热点事件屡屡导致微博系统瞬间瘫痪;再如中央服务器病毒入侵,会迅速扩散至终端用户,服务器级别越高,社会危害越大。

与 C/S 模式相比,P2P 模式是去中心化的,网络中每个节点的地位是对等的。节点可以自由地加入和退出,网络扩展性强;新节点的加入,可能为系统带来新的资源,整体的多样性得以扩充,服务能力也同步增加,有利于网络的负载均衡;单点故障不会影响整体系统,因而网络的健壮性和可用性高;所有节点都具备中继转发功能,大幅提高通信匿名性,个人隐私得到保护。基于上述优势,P2P 网络在分布式存储、计算能力共享、流媒体等领域具有广阔的应用前景。

根据网络体系结构,一般可以把 P2P 网络分为三类[6]。

1. 混合式对等网络

它是 C/S 和 P2P 两种模式的混合,反映了早期网络从 C/S 到 P2P 的过渡。混合式对等网络最具影响力的代表是 Napster,Napster 是一个为音乐迷们提供交流 MP3 文件的平台,最典型的特点就是存在维护共享文件索引与提供查询的服务端(C/S 模式),但具体内容存储在用户硬盘中,内容的传送只在用户节点间进行(文件交换是 P2P 的)。由于服务端的存在,从这个角度来说,早期的 P2P 网络不是完全去中心化的。

2. 无结构对等网络

这种网络的特点是无固定网络结构图,无中心节点;每个节点既是客户端又是服务端,节点地址没有统一标准,内容存放位置与网络拓扑无关,对等节点间通过客户端软件搜索网络中存在的对等节点,并直接交换信息。典型的无结构 P2P 网络协议如 Gnutella,它是纯粹意义上的 P2P 网络[7]。

3. 结构化对等网络

它以准确、严格的结构来组织网络，一般采用哈希函数将节点地址规范为标准的标识，内容的存储位置与节点标识之间存在映射关系，可以实现有效的节点地址管理，精确定位节点信息。因为所有节点按照某种结构进行有序组织，所以整个网络呈环状或者树状。其最具代表性的经典模型和应用体系如 Chord、Pastry 等。

Napster 是 P2P 技术在文件分享领域的最先尝试，得益于 P2P 模式的天然优点，Napster 迅猛发展。它的创新点在于通过构建存储音乐文件索引与存放位置的信息的 Napster 服务器，实现文件查询与文件传输的分离。用户需要某个音乐文件时，首先与 Napster 服务器相连，检索信息，根据返回的存放节点择优再进行下载。这种模式查询与下载并行，大幅提高了系统整体带宽利用效率，当然，尽管 Napster 服务器只是一个索引和搜索的轻服务，但中心化终归是隐患，不可避免地带来了系统瓶颈、单点故障等问题。

BitTorrent 是 Napster 架构的衍生强化版，分布式的思想在其网络中得到更深层次的渗透。BitTorrent 网络中共享同一文件的用户形成一个独立的子网，从而将服务端分散化了，不会因为单点故障影响整体网络。文件的持有者将文件发送给其中一名用户，再由这名用户转发给其他用户，用户之间相互转发自己所拥有的文件部分，直到每个用户的下载都全部完成。这种方法特点是下载的人越多，下载速度越快。原因在于，每个下载者将已下载的数据提供给其他下载者下载，并通过一定的策略保证上传速度越快，下载速度也越快。然而，不管是 Napster 还是 BitTorrent 都存在一个共同的问题：人们大多只愿意免费“获取”，而不愿耗费资源去“共享”。这就催生了后来引入强制共享机制的电驴——eDonkey。

区块链激励机制则给出了更好的答案。区块链网络层大多是选择 P2P 模式作为其组网模型，其理念就是去中心化和去中介化，不要依赖任何第三方来完成自身系统的运转，而 P2P 网络天然的全网对等的属性与区块链不谋而合，再加上 P2P 已经是发展成熟、经过考验的技术，二者的结合几乎是必然的。

3.1.2 节点类型

比特币网络由多种类型的节点组成，其功能集合一般包括网络路由（Network Route，简写为 N）、完整区块链（Full Blockchain，简写为 B）、矿工（Miner，简写为 M）、钱包（Wallet，简写为 W）。每个区块链节点都参与全网路由，同时也可能包含其他功能。根据节点提供的功能不同，主要分为如下几种，如图 3-3 所示[8]。拥有全部功能集的称为核心客户端（Bitcoin Core）；不参与挖矿，仅提供完整区块链数据与参与全网路由的节点称为完整区块链节点；拥有完整区块链数据，并参与挖矿与路由的节点称为独立矿工；仅提供钱包功能与参与全网路由的节点称为轻量（SPV）钱包。除了这些主要节点类型外，还有一些节点运行其他协议，如挖矿协议，因而网络中还有矿池协议服务器、矿池挖矿节点、Stratum 钱包节点等。

拥有完整的、最新区块链数据的节点也称为“全节点”，这样的节点能够独立自主地校验所有交易；只保留区块头数据，通过“简易支付验证”方式完成交易验证的节点为“SPV 节点”/“轻量级节点”，它们没有区块链的完整拷贝。随着比特币生态的发展，比特币 P2P 协议、Stratum 协议、矿池协议以及其他连接比特币系统组件相关协议综合构成了现在的比特

图 3-3　比特币网络的节点类型

币网络，我们称之为“扩展比特币网络”(Extended Bitcoin Network)，扩展比特币网络包含了多种类型的节点(如核心客户端、完整区块链节点等)、网关服务器、边缘路由器、钱包客户端以及它们互相连接所需要的 P2P 协议、矿池挖矿协议、Stratum 协议等各类协议(矿池协议参见 3.1.5 节)，如图 3-4 所示[8]。

新区块链节点启动后，如何发现网络中其他节点并获知其地址是区块链组网的重要环节，一般通过如下五种方式实现①。

1. 地址数据库

网络节点的地址信息会存储在地址数据库 peers.dat 中。节点启动时，由 address manager 载入。节点第一次启动时，无法使用这种方式。

2. 通过命令行指定

用户可以通过命令方式将指定节点的地址传递给新节点，命令行传递参数格式形如 -addnode <ip> 或者 -connect <ip>。

3. DNS 种子

当 peers.dat 数据库为空，且用户没有使用命令行指定节点的情况下，新节点可以启用 DNS 种子，默认 DNS 种子有 seed.bitcoin.sipa.be，dnsseed.bluematt.me，dnsseed.bitcoin.dashjr.org，seed.bitcoinstats.com，seed.bitcoin.jonasschnelli.ch，seed.btc.petertodd.org。

4. 硬编码地址

如果 DNS 种子方式失败，还有最后的手段，即硬编码地址。需要注意的是，需要避免 DNS 种子和硬编码种子节点的过载。因此，通过他们获得其他节点地址后，应该断开与这些种子节点的连接。

5. 通过其他节点获得

节点间通过 getaddr 消息和 addr 消息交换 IP 地址信息，具体交互过程详见第 3.1.3 节。

① https://en.bitcoin.it/wiki/Bitcoin_Core_0.11_(ch_4):_P2P_Network

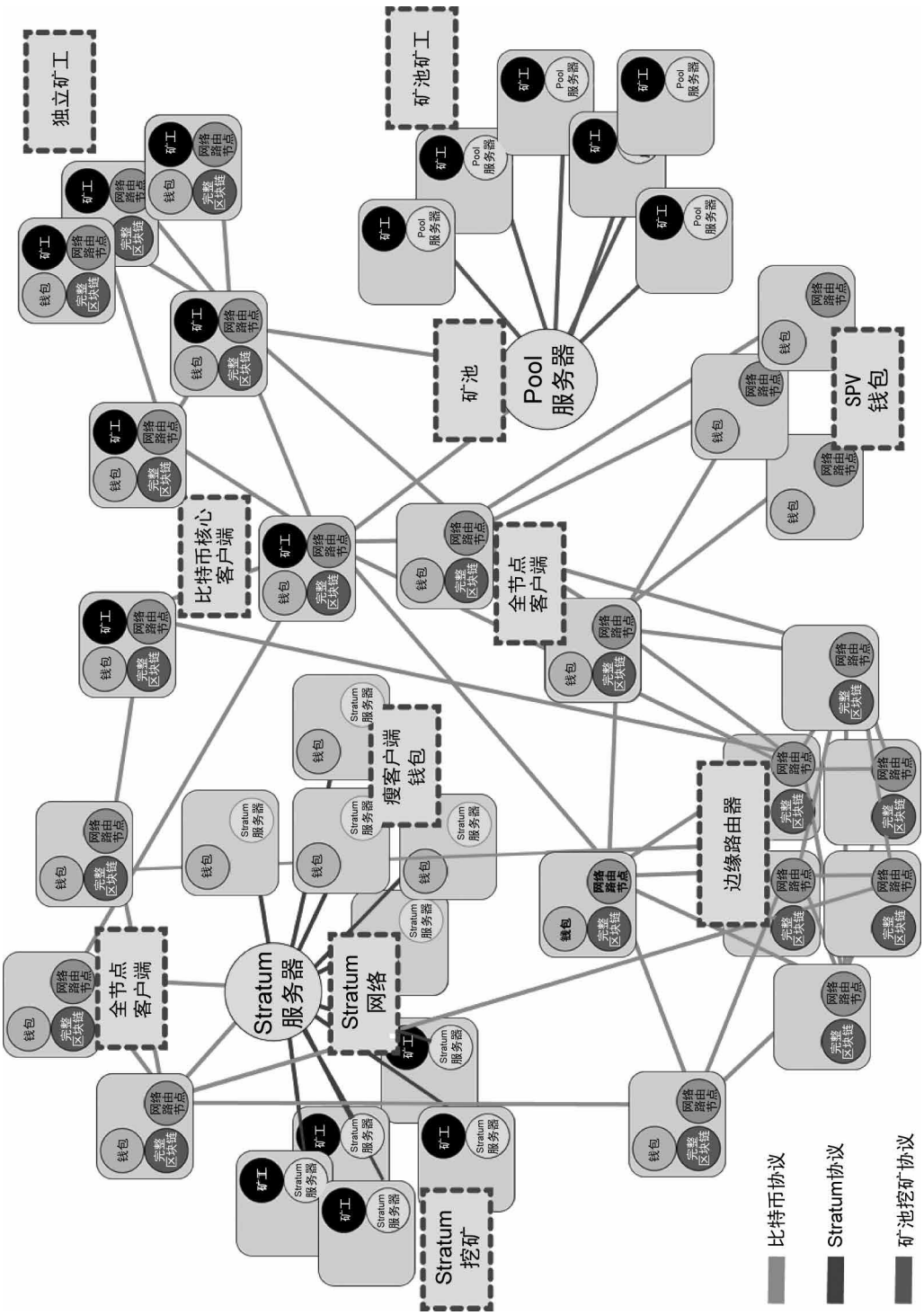


图 3-4　扩展比特币网络示意图

3.1.3 数据传播协议

一般新节点由初始种子启动，再与相邻节点通信获得更多连接。下面将详细介绍节点之间是如何通信的。节点通常采用 TCP 协议，使用 8333 端口与其他对等节点交互。一个通用的区块链网络一般包括如下核心场景。

- 节点入网建立初始连接
- 节点地址传播发现
- 矿工、全节点同步区块数据
- 客户端创建一笔交易
- 矿工、全节点接受交易
- 矿工、全节点挖出新区块，并广播到网络中
- 矿工、全节点接收广播的区块

一般，version 消息和 verack 消息用于建立连接；addr 和 getaddr 消息用于地址传播；getblocks、inv 和 getdata 消息用于同步区块链数据，tx 消息用于发送交易。

1. 建立初始连接

建立连接始于“握手”通信，这一过程如图 3-5 所示[8]。

比特币节点之间的握手过程类似 TCP 三次握手，节点 A 向节点 B 发送包含基本认证内容的 version 消息，节点 B 收到后，检查是否与自己兼容，兼容则确定连接，返回 verack 消息，同时向节点 A 发送自己的 version 内容，如果节点 A 也兼容，则返回 verack，至此连接成功建立。

2. 地址广播及发现

一旦建立连接，新节点将向其相邻节点发送包含自身 IP 地址的 addr 消息。相邻节点则将此 addr 消息再度转发给各自相邻节点，进而保证新节点被更多节点获知。此外，新接入节点还向其相邻节点发送 getaddr 消息，获取邻居节点可以连接的节点列表。整个过程如图 3-6 所示[8]。

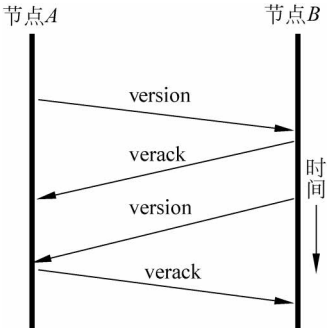


图 3-5　建立初始连接示意图

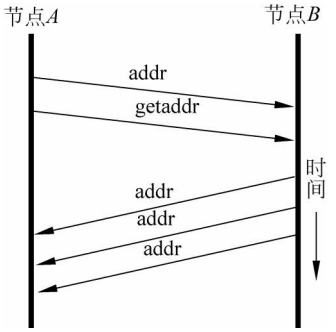


图 3-6　地址广播与发现示意图

3. 同步区块数据

新入网节点只知道内置的创世区块，因此需要同步最新区块。同步过程始于发送 version 消息，该消息含有节点当前区块高度(BestHeight 标识)。具体而言，连接建立后，双方会互相发送同步消息 getblocks，其包含各自本地区块链的顶端区块哈希值。通过比较，区块数较多的一方向区块较少的一方发送 inv 消息。需要注意的是，inv 消息只是一个清单，并不包括实际的数据。落后方收到 inv 消息后，开始发送 getdata 消息请求数据，具体如图 3-7 所示[8]。

需要补充一点的是，我们在 2.3.3 节讨论了简化支付验证技术——SPV，SPV 节点同步的不是区块数据，而是区块头，使用 getheaders 消息，如图 3-8 所示[8]。

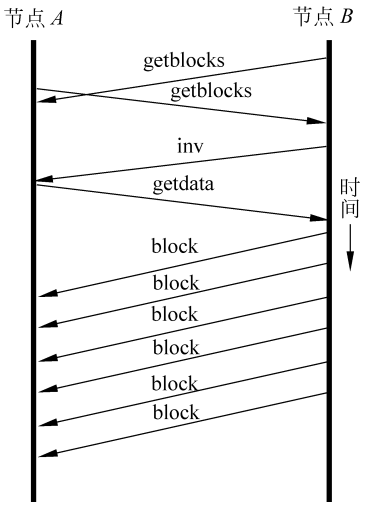


图 3-7　同步区块数据示意图

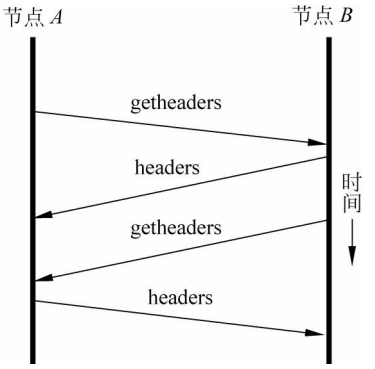


图 3-8　SPV 节点同步区块头示意图

4. 交易传播

交易数据的广播是更为常见的场景：假定有节点要发送交易，那么首先会发送一条包含该交易的 inv 信息给其邻居节点。邻居节点则通过发送 getdata 消息请求 inv 中所有交易的完整信息。如果发送方接收到 getdata 响应信息，则使用 tx 发送交易。接收到交易的节点也将以同样的方式转发交易(假定它是个有效的交易)。比特币系统的交易数据传播协议包括如下步骤：

(1) 比特币交易节点将新生成的交易数据向全网所有节点进行广播；

(2) 每个节点都将收集到的交易数据存储到一个区块中；

(3) 每个节点基于自身算力在区块中找到一个具有足够难度的工作量证明；

(4) 当节点找到区块的工作量证明后，就向全网所有节点广播此区块(block 消息)；

(5) 仅当包含在区块中的所有交易都是有效的且之前未存在过的，其他节点才认同该区块的有效性；

(6) 其他节点接受该数据区块，并在该区块的末尾制造新的区块以延长该链条，而将被接受区块的随机哈希值视为先于新区块的随机哈希值。

5. 检测节点存活

ping 消息有助于确认接收方是否仍处于连接状态。通过发送 ping 消息，可以检测节点是否存活。接收方通过回复 pong 消息，告诉发送节点自己仍然存在。默认情况下，任何超过 20 分钟未响应 ping 消息的节点会被认为已经从网络中断开。

6. Gossip 传播协议

Gossip 协议最初是由艾伦·德默斯(Alan Demers)、丹·格林(Dan Greene)等于 1987 年在论文“Epidemic Algorithms for Replicated Database Maintenance”(用于复制数据库维护的流行病学算法)中提出的。Gossip 协议是一种计算机-计算机通信方式，受启发于现实社会的流言蜚语或者病毒传播模式。Gossip 协议也被称为反熵(Anti-entropy)。熵是物理学中一个概念，代表无序、错乱，而反熵则是在杂乱中寻求一致，这形象地体现了 Gossip 协议的特点：在一个有界网络中，每个节点随机地与邻居节点通信，每个节点都遵循这样的操作，经过一番交错杂乱的通信后，最终所有节点状态达成一致。这种最终一致性是指保证在最终某个时刻所有节点一致对某个时间点前的所有历史达成一致。虽然冗余通信，但具有天然的分布式容错优点，因而被广泛用于分布式系统的底层通信协议，如 Facebook 开发的 Cassandra，就是通过 Gossip 协议维护集群状态。

一般而言，Gossip 协议可以分为 Push-based 和 Pull-based 两种。前者工作流程如下。

(1) 网络中某个节点 v 随机选择其他 n 个节点作为传输对象。

(2) 节点 v 向其选中的 n 个节点传递消息。

(3) 接收到信息的节点重复进行相同的操作。

Pull-based Gossip 协议则相反。

(1) 网络中某个节点 v 随机选择其他 n 个节点询问有没有新消息。

(2) 收到询问的节点回复节点 v 其最近收到的消息。

为了提高性能，也有结合 Push-Pull 的混合协议。Gossip 协议一般是基于 UDP 协议实现。

在区块链领域，Gossip 协议也有广泛应用。面向企业联盟链的超级账本 Fabric 就采用 Gossip 作为其 P2P 网络的传播协议[9]，由 Gossip 协议负责维护新节点的发现、循环检查节点、剔除离线节点、更新节点列表。随着区块链技术的发展，目前涌现出若干新兴的分布式账本数据结构，将单一链式结构的技术范畴拓展为基于图结构的分布式账本，例如哈希图(HashGraph)即采用了 Gossip 协议(详见第 10 章)。

3.1.4 数据验证机制

当新区块在区块链网络传播时，每个接收到区块的节点都将对区块进行独立验证，验证通过的区块才会进行转发，以此尽早杜绝无效或者恶意数据在网间传播，预防小部分节点串通作恶导致无效区块被网络接受，尽最大可能保证网络中传播区块的正确性。以比特币网络为例，节点接收到邻近节点发来的数据后，其首要工作就是验证该数据的有效性。矿工节点会收集和验证 P2P 网络中广播的尚未确认的交易数据，并对照预定义的标准清单，从数据结构、语法规范性、输入输出和数字签名等各方面校验交易数据的有效性，并将有效交易

数据整合到当前区块中。

具体而言，数据验证清单主要包括[①]。

(1) 验证区块大小在有效范畴。

(2) 确认区块数据结构(语法)的有效性(参见 2.1 节区块结构)。

(3) 验证区块至少含有一条交易。

(4) 验证第一个交易是 coinbase 交易(Previous Transaction hash 为 0 且 Previous Txout-index 为－1)，有且仅有一个。

(5) 验证区块头部有效性(参见 2.1.1 节区块头)。

① 确认区块版本号是本节点可兼容的。

② 区块引用的前一区块是有效的。

③ 区块包含的所有交易构建的默克尔树是正确的。

④ 时间戳合理(参见 2.3.1 节时间戳)。

⑤ 区块难度与本节点计算的相符。

⑥ 区块哈希值满足难度要求(参考 2.1.1 节区块头中的区块标识符)。

(6) 验证区块内的交易(参见 2.1.2 节区块体)有效性，具体检查列表如下。

① 检查交易语法正确性。

② 确保输入与输出列表都不能为空。

③ lock_time 小于或等于 INT_MAX，或者 nLockTime 和 nSequence 的值满足 MedianTimePast(当前区块之前的 11 个区块时间的中位数)。

④ 交易的字节大小大于等于 100。

⑤ 交易中签名数量小于签名操作数量上限(MAX_BLOCK_SIGOPS)。

⑥ 解锁脚本(scriptSig)只能够将数字压入栈中，并且锁定脚本(scriptPubkey)必须要符合 isStandard 的格式(拒绝非标准交易)。

⑦ 对于 coinbase 交易，验证签名长度为 2～100 字节。

⑧ 每一个输出值，以及总量，必须在规定值的范围内(不超过全网总币量，大于 0)。

⑨ 对于每一个输入，如果引用的输出存在内存池中任何的交易，该交易将被拒绝。

⑩ 验证孤立交易：对于每一个输入，在主分支和内存池中寻找引用的输出交易，如果输出交易缺少任何一个输入，该交易将被认为是孤立交易。如果与其匹配的交易还没有出现在内存池中，那么将被加入到孤立交易池中。

⑪ 如果交易费用太低(低于 minRelayTxFee 设定值)以至于无法进入一个空的区块，则交易将被拒绝。

⑫ 每一个输入的解锁脚本必须依据相应输出的锁定脚本来验证。

⑬ 如果不是 coinbase 交易，则确认交易输入有效，对于每一个输入：

- 验证引用的交易存于主链；
- 验证引用的输出存于交易；
- 如果引用的是 coinbase 交易，确认至少获得 COINBASE_MATURITY(100)个确认；

① https://en.bitcoin.it/wiki/Protocol_rules

- 确认引用的输出没有被花费；
- 验证交易签名有效；
- 验证引用的输出金额有效；
- 确认输出金额小于等于输入金额(差额即为手续费)。

⑭ 如果是 coinbase 交易，确认金额小于等于交易手续费与新区块奖励之和。

如果数据有效，则按照接收顺序存储到交易内存池以暂存尚未记入区块的有效交易数据，同时继续向邻近节点转发。如果数据无效，则立即废弃该数据，从而保证无效数据不会在区块链网络中继续传播。

3.1.5 矿池网络协议

上文我们提及比特币 P2P 主网络上连接着许多矿池服务器以及协议网关，下面将简述矿池挖矿协议。

Getwork 协议可以认为是最早的挖矿协议——实现区块链数据与挖矿逻辑剥离，拥有完整数据的节点构造区块头(参考 2.1.1 节区块头)，即 Version、Prev-block、Bits 和 Merkle-root 这 4 个字段必须由节点客户端提供。挖矿程序主要是递增遍历 Nonce，必要时候可以微调 Timestamp 字段。

对于 Getwork 而言，矿工对区块一无所知，只知道修改 Nonce 这 4 个字节，共计 2^{32} 大小的搜索空间，显然不符合迅猛发展的比特币矿机算力。如果继续使用 Getwork 协议，矿机需要频繁调用 RPC 接口，显然不合时宜。如今比特币和莱特币节点都已经禁用 Getwork 协议，转向更高效的 Getblocktemplate 协议。该协议诞生于 2012 年，其最大的不同点是：Getblocktemplate 协议让矿工自行构造区块。因为由矿工构建 coinbase 交易(参考 2.3.1 节时间戳最后一段基于 coinbase 字段的随机数扩展)，这种方式所带来的搜索空间巨大。

Getblocktemplate 协议虽然扩大了搜索空间，但正常的一次 Getblocktemplate 调用节点都会返回 1.5M 左右的数据，因为要把区块包含的所有交易都交给矿工，数据负载过大。Stratum 协议巧妙解决了这个问题。Stratum 协议是为了扩展支持矿池挖矿而编写的挖矿协议，于 2012 年底出现，是目前最常用的矿机和矿池之间的 TCP 通信协议之一，数据采用 JSON 格式封装，矿机与矿池通信过程一般如图 3-9 所示。

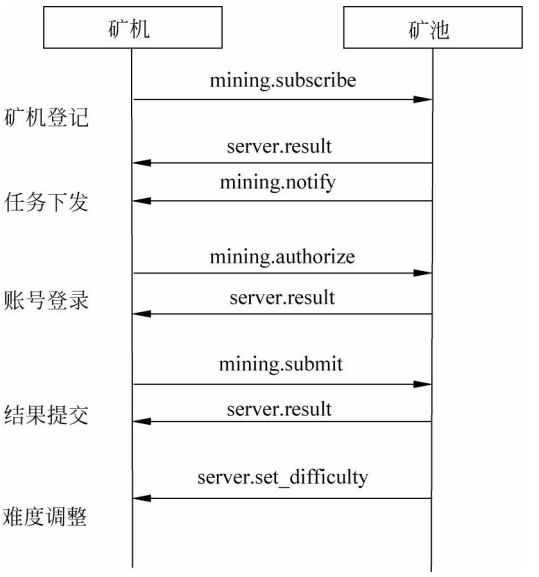


图 3-9 stratum 协议的基本通信接口

Stratum 协议利用 Merkle 树结构特性，从 coinbase 构造 hashMerkleroot，无须全部交易，只要把与 coinbase 涉及的默克尔路径上的 hash 值返回即可。假如区块包含 N 笔交易，这种方式数据规模将压缩至 $\log_2(N)$，大大降低了矿池和矿工交互的数据量。Stratum 协议不但保证给矿工增加足够的搜索空间，而且仅需很少的数据

交互，这就是该协议最优雅的地方。

3.1.6　区块链分叉

分叉(Fork)，是软件开发中常见的一个概念，一般代表“复制并修改”。例如开源项目常见新群体 fork 了一个项目，项目就产生分叉，拆分为两个项目，新群体将沿着这个 fork 向另外的方向独立发展这个项目。

我们以比特币为例讨论区块链分叉，可以从两个层面理解：一种是自然分叉，另一种则是人为的分叉。

1. 自然分叉，即机器共识过程产生的临时分叉

比特币网络是一个去中心化的 P2P 网络。固有的节点地域分布、网络传输延迟，造成节点接收新区块存在一定的时间差异。当两个不同节点近乎同时挖掘出新区块 A、B 并进行广播的时候，就会造成后继区块链分叉的发生。区块 A、B 都是有效的，有些节点先接收到 A，其他节点则先接收到 B。节点会把先接收到的区块加入自己的主链，把晚收到的区块加到由主链分叉出来的支链上，临时的分叉由此产生。当后续再发现新区块 C，而 C 所指向的父区块是 A 的时候，网络中的节点有两种选择：原本主链最后一个区块是 A 的节点，自然正常将区块 C 加入主链，而主链最后是区块 B 的节点，则先在支链上将区块 C 加至区块 A 后面，并且将支链切换为主链。随着区块 C 在全网的转播，最终比特币全网节点的区块链又趋于一致，始终保持算力最大的为主链。在比特币网络中发生一次临时分叉很常见，连续 6 次产生分叉的情况，从概率上来说近乎为 0，这也是为什么区块确认是一般是 6 块之后。

2. 人为分叉，即人的共识失败产生的分叉

区块链的每一次升级都将导致分布式网络中的节点根据其是否接受升级而运行不同规则，于是人为的分叉就会产生。以比特币为例，比特币系统的改进都是通过 BIP 方式进行，现在比特币社区已经累计有数百条 BIP，这是“人的共识”达成的过程，具体又可以分为“软分叉”和“硬分叉”。

1) 软分叉

“软分叉”是向前兼容的分叉。新规则下产生的区块可被未升级的旧节点所接受，旧节点只是无法识别、解析新规则。新、旧版本互相兼容，软分叉对整个系统的影响较小。新的交易类型一般以软分叉的方式升级，例如 P2SH(Pay-to-Script-Hash)以及隔离见证。

根据由谁主导“激活”新的提议过程，又可分为矿工激活软分叉(Miner-Activated Soft Fork，MASF)与用户激活软分叉(User-Activated Soft Fork，UASF)。这里挖矿的节点被称为“矿工”，其他不挖矿的普通节点被称为“用户”。

软分叉在比特币发展过程中多次发生，是一种逐步升级的过程，以 BIP-34 为例：我们在 2.3.1 节提到基于 coinbase 字段的随机数扩展，coinbase data 大小在 2～100 字节，原本可任意定制，但 BIP-34 要求必须在开头包含块高度且更新块版本信息。其升级过程如下：

(1) 初始矿工将块版本号设置为“2”，表示其准备好升级，但此刻并不要求 coinbase data 包含块高度；

(2) 当最近 1000 个区块中超过 75%的版本号是“2”时,整个系统开始强制要求版本号设置为“2”,且要求 coinbase data 包含块高度,但此时版本号为“1”的区块仍被接受;

(3) 当最近 1000 个区块中超过 95%的版本号是“2”时,版本号为“1”的区块将不被接受,迫使最后一小部分节点进行升级。

软分叉是在现有结构基础上进行改进,不增加新字段,分叉过程不影响系统的稳妥运行,但是存在升级空间受限的缺点,硬分叉则相反。

2) 硬分叉

“硬分叉”是不向前兼容的,旧版本节点不会接受新版本节点创建的合法区块,于是新旧版本节点开始在不同的区块链上运行,由于新旧节点可能长期并存,不像软分叉是临时的,硬分叉是有可能会长期存在的,分叉链的存活在于其算力的大小。如果原区块链称为 A 版本,硬分叉产生的同源分叉链称为 B 版本,则具体可以分为如下几种情况。

(1) A 版本仍然被广泛支持,B 版本因算力不足而消亡,即还是保留原链。

(2) B 版本获得广泛支持,A 版本因算力不足而消亡,即保留新链。

(3) A、B 版本都有相当比例的矿工的支持,同时并存,这种情况是最为符合严格意义上的硬分叉,例如 ETH 与 ETC,两者都有其代币,这种分叉存在一定的门槛。

(4) A 版本仍然被广泛支持,B 版本通过代码调整难度,小部分节点也能够让它存活。这种分叉币几乎没有门槛,人人可以分叉。

(5) B 版本获得支持,A 版本调整代码,小算力也可存活。

硬分叉的过程一般经历如下几个阶段。

(1) 软件分叉:新的客户端发布,新版本改变规则且不被旧客户端兼容,首先客户端出现了分叉。

(2) 网络分叉:接受新版的节点在网络运行,其发现的区块将被旧版节点拒绝,旧版节点断开与这些新版节点的连接,因此网络进一步出现了分叉。

(3) 算力分叉:运行不同客户端版本的矿工的算力将逐渐出现分叉。

(4) 链分叉:升级的矿工基于新规则挖矿,而拒绝升级的矿工仍基于旧规则,导致整个区块链出现了分叉。

每一种区块链的背后都有其对应的社区、开发者、矿工等利益、信仰共同体,链的硬分叉同时也会带来对应社区的分裂,受关注度最高当属 2016 年因“The DAO 事件”出现的以太坊经典,2017 年催生出的比特币现金以及 2018 年比特币现金的进一步分叉。例如,2017 年针对扩容问题的意见分歧导致 8 月 1 日比特币的首次硬分叉,比特币社区的分裂催生了比特币现金;2018 年 11 月 15 日比特币现金社区再现分裂,分叉为 Bitcoin ABC 和 Bitcoin Satoshi's Vision(BSV)。由此可见,社区的分裂才是主导区块链分叉的因素。

值得注意的是,2017 年 8 月 1 日,比特币的硬分叉让大量的比特币持有者凭空地增加了一种新的数字货币,即比特币现金(BCH)。这种硬分叉,没有减少资产,反而让人手里多了一种资产,于是区块链分叉就成了一种资产凭空增加的方式。硬分叉这种创造货币的方式和 ICO(Initial Coin Offering,首次代币发行)非常类似,于是一个新的名词诞生了——IFO(Initial Fork Offering,首次分叉发行)。矿工团队在创造分叉的同时,可以在分叉发生的区块中,利用自己的特权,分配一些货币给自己或其他人(直接写成 CoinBase 交易即可),然后再开放让所有人都可以参与挖矿。

3.2　跨链技术

随着区块链技术的广泛应用，形形色色的“链”大量出现，但大多数的链却都是独立的、封闭的，链与链之间高度异构，难以互通，无法对话，各自的数据与价值都局限于各自的“数据孤岛”之中，这种困局与早期的 Internet 非常相似。Internet 由一种支持异构网络互连的 TCP/IP 协议破局，从互不通信的单机逐步走向大规模网络，任何用户在任何地方，只要通过互联网服务提供商，都可以访问到世界任何一个角落的互联网信息。同样，当前区块链之间的互通性和互操作性已经极大地制约着区块链的发展，区块链网络迫切需要能够使众多区块链协同工作的标准化协议，多链互通相关技术已经被认为是区块链未来发展的制高点，跨链技术就在这样的背景下应运而生。正如 TCP/IP 协议之于互联网，跨链技术必将促使区块链向真正的价值互联网演进。

跨链技术最早可以追溯到 2012 年的 Ripple 公司，其致力于建立一套适用于所有记账系统，能够包容所有记账系统差异性的协议；2014 年，BlockStream 团队首次提出楔入式侧链(Pegged Sidechains)方案以寻求与其他区块链的互操作；2015 年 10 月，Ripple 公司进一步引入了一种跨链价值传输的 Inter-Ledger Protocol(ILP)[①]，是跨链转账的首次尝试，其目标是打造全球统一支付标准，创建统一的网络金融传输的协议；2016 年 9 月，以太坊创始人维塔利克·布特林(Vitalik Buterin)为 R3 区块链联盟写了一份关于跨链互操作的报告“Chain Interoperability”(链互操作性)[②]，对区块链互操作性问题做了深度分析，并提出三种实现跨链的策略：公证人机制(Notary schemes)、侧链/中继(Sidechains/Relays)、哈希锁定(Hash-locking)。

随后，闪电网络(Lightning network)提出基于微支付通道构建跨链方案，BTC-Relay 基于中继跨链方式实现从比特币到以太坊的单向流通，万维链(Wanchain)则利用多方计算和门限密钥共享方案，实现公有链间的跨链交易，而以 Polkadot 和 Cosmos 为代表的跨链技术，更多关注的是跨链基础设施建设。第 3.2.3 节将重点介绍若干典型的跨链平台与项目。

从跨链的发展历程中，可以看到其离不开侧链的身影，从早期的 Blockstream 的开源侧链项目元素链，这可以看作是跨链技术实现的雏形之一；到后来基于智能合约的 BTC-Relay，实现从比特币到以太坊单向流通。侧链与跨链的概念交错耦合，两者技术内容方面是具有共性的，简言之，两者技术天然相通，应用的侧重点不同。一般侧链服务于主链，侧重币币的兑换，而如今跨链所涵盖的范畴已经扩大，链与链之间的关系不仅仅是主侧关系，也可以是对等的，跨链通信不限于转账，更多关注打通不同区块链之间的信息、资产、状态，跨链旨在链之间价值、服务与功能的连通，跨链的应用场景比侧链更为丰富。

3.2.1　概念与定义

1. 跨链技术所在层级

在深入讨论跨链技术之前，首先分析一下未来基于去中心化服务的 Web 3 愿景(如

① https://interledger. org/interledger. pdf

② https://www. r3. com/wp-content/uploads/2017/06/chain_interoperability_r3. pdf

图 3-10 所示),以便直观感受跨链在整个技术栈中的位置。

(1) Layer 0 层是数据传输层,主要指 P2P 网络和传播机制,侧重区块链与传统网络的结合。

(2) Layer 1 层是 On-Chain 层,侧重底层账本公链自身,包括数据存储协议(如 IPFS)、去信任协议(如比特币)等。

(3) Layer 2 层是 Off-Chain 层,链下的意思是脱离公链,侧重扩展性延伸和链上链下打通。

(4) Layer 3 层主要包括开发 API 和语言,如 Web3.js、Solidity 等。

(5) Layer 4 层主要包括协议可扩展的用户接口,如 Metamask、Parity 等。

通过上述的分析,可以看出跨链技术处于 Layer 2,不影响公链本身情况下扩展现有区块链,是重要的基础协议。

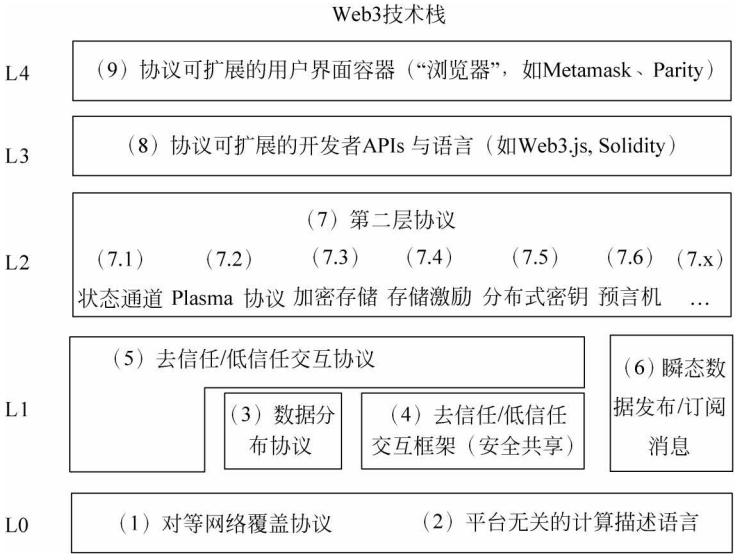


图 3-10 Web 3 技术栈概念图

2. 互操作性

互联网技术使得全世界的计算机设备能够互相通信与相互操作。正是这种"互操作性",使得计算机集群成为一个整体。1988 年,美国 DARPA(Defense Advanced Research Projects Agency,国防高级研究计划局)在"The design philosophy of the DARPA Internet protocols"中指出互联网三大基本目标为:

(1) 可生存性(Survivability):尽管网络或网关受损,互联网通信仍必须能够继续进行;

(2) 服务类型的多样性(Varieties of Service Types):互联网必须支持多种类型的通信服务;

(3) 网络的多样性(Varieties of Networks):互联网必须可以承载各种各样的网络。

如果区块链系统要成为未来全球经济的重要基础设施,成为全球商业与价值分布式网

络，那么其体系结构也必须满足互联网的基本目标，区块链系统间互操作性将是一个核心需求。

麻省理工学院连接科学实验室（MIT Connection Science）首席技术官托马斯·哈德约诺（Thomas Hardjono）在论文“Towards a Design Philosophy for Interoperable Blockchain Systems”中提出一个可互操作的区块链架构应具备的特征[①]：可互操作的区块链体系结构是可区分的区块链系统的组合，每个区块链系统代表一个分布式数据账本，其中交易执行可能跨越多个区块链系统，并且其中记录在一个区块链中的数据可以通过语义兼容的方式被另一种可能来自外部的交易访问和验证。

一般来说，区块链的互操作性可以满足以下场景需求。

（1）便携式资产（Portable Assets）：即资产转移场景，数字加密货币或资产可以在不同链之间来回转移。

（2）银货两讫（Payment-versus-delivery）：即一手交钱一手交货，强调链间资产的同时交换，即原子互换（Atomic Swap）概念。同步交收（Payment-versus-payment）也是类似意思。

（3）跨链预言机（Cross-chain Oracles）：链 A 上的智能合约的触发和执行依赖另一条链 B 上的预言机的证据，即具备他链信息或事件的读取与验证能力。

（4）资产留置（Asset Encumbrance）：链 A 上的资产被锁定，解锁条件取决于链 B 的行为。如金融衍生品的抵押品，破产追回，法院命令和涉及保证金的各种场景。

（5）通用跨链合约（General Cross-chain Contracts）：根据链 A 上的资产证明在链 B 上分发股息。

3. 跨链的定义

跨链技术目前尚未形成行业公认的定义，结合跨链设计目标，跨链是指实现区块链账本之间资产的互操作，即在可以引入第三方但不改变原生链的前提下实现区块链之间资产的互换、转移。因此，目前对跨链的研究主要集中在资产互换与资产转移两个场景。

（1）资产互换：通常指发生在两条链之间不同用户间的资产互换。与在中心化交易所中的币币兑换类似，只是这个过程是在链上进行的。例如用户 A 和 B 在比特币和以太坊两条链上都有相应的账户，其中用户 A 在比特币区块链上有 10 个比特币，用户 B 在以太坊区块链上有 300 个以太币。假设当时 1 个比特币等值于 30 个以太币，他们不想在中心化交易所进行，希望彼此能够直接在链上、点对点、不通过第三方进行交换，则资产互换场景使得用户 A 换得 300 个以太币，用户 B 获得 10 个比特币，两条链上的资产总量不发生变化，只是相应资产的所有权发生转换。这个过程中突出的问题是任何一方违背交易，如何保障另一方的利益？

（2）资产转移：通常指发生在两条链之间单用户的资产迁移，可以分为单向或者双向，例如用户 A 想将比特币区块链上的 10 个比特币兑换成以太币，即将资产转移到以太坊区块链上。那么，与资产互换不同之处在于没有用户 B 去承接这 10 个比特币，这些币原则上是要被销毁或冻结，两条链上的资产总量不再保持不变，各自是需要相应地增加或减少的。

① https://arxiv.org/pdf/1805.05934.pdf

3.2.2 难点与解决方案

跨链交易提供了一种链间清算机制，清算的本质就是精确记账，因此跨链传递的不仅是信息流，更在于其背后对应的需要精确记录的价值。结合当前区块链自身的特点，要实现跨链交易，首先需要解决两个难题。

一是如何实现对交易的确认。一方面，区块链系统自身缺乏主动获取外界信息的机制，而原链上的交易状态对于另一条链来说就是外部信息但又不可或缺，如何获得正确的原链上的交易状态信息是跨链交易的关键；另一方面，许多区块链（如 PoW 共识算法）对交易的确认是有等待时间的，需要获得足够算力才能保障交易的有效，因为理论上任一笔交易都有可能被撤销。一个交易被确认之后依然有可能作废，如何验证交易将加大跨链交易的难度。

二是如何保证交易的原子性。跨链交易要么全发生，要么都不发生，始终保持两条链上的账本的同步性，确保账本的变动的一致性。如果发生错误，要有相应的回滚机制，保障交易双方的利益，系统回到交易前的状态。回撤机制是跨链交易发生异常的重要保障。

1. 难点一：如何实现对交易的确认

区块链系统是彼此封闭的。两个独立的、不兼容的系统要做到信息的互通，则需要“中间人”的角色来搭线以实现对交易的确认。“中间人”这个机制可以有多种实现方式，可以是中心化方式，也可是去中心化，中间人节点可以是单个的也可以是集群。根据交易信息传输和验证方式不同，具体方案可分为公证人机制、中继机制以及侧链机制[①]。

1）公证人机制

公证人机制是一种直接和自然地实现跨链的思路，其最大的特点是不用关注所跨链的结构，因此也是较通用与成熟的模式。在双方无法互信的场合下，就需要一个中间人进行监督和公证。此时的“中间人”一般是可信第三方，担任“中立方”的角色，除了收集交易状态数据，还负责交易的验证。根据中心化程度，又可分为单签名公证人机制、多签名公证人机制以及分布式签名公证人机制。

（1）单签名公证人机制：单签名公证人机制也称为中心化公证人机制（Centralized Notary Schemes），公证人通常由单一指定的节点或者机构担当，在交易过程充当交易确认者和冲突仲裁者的角色。以中心化的机构保障信用，实现最为简易，交易处理速度快，兼容性好，跟日常生活中支付宝承担的角色很类似。该机制一般被区块链交易所采纳。其缺点也很明显，即过度依赖于公证人的可靠性，高度的中心化会带来安全隐患、性能瓶颈。

（2）多签名公证人机制：为了改善单签名公证人机制的过度中心化问题，多重签名公证人机制（Multi-signature Notary Schemes）引入多位公证人共同签名达成共识后才能确认交易。该机制约定只要达到一定的公证人签名数量或者比例，交易就能被确认，因此，少数公证人作恶或者被攻击不影响系统的运行，安全性提高。公证人选取方式可以多种，如随机选择、可信联盟的可信节点等。实践中一般利用多重签名脚本实现，所以该机制要求跨链交易的双方链本身支持多重签名功能。

① https://www.8btc.com/article/294037

(3) 分布式签名公证人机制：分布式签名技术综合利用分布式密钥生成、门限签名等密码学算法，从最底层密码学算法层面解决跨链去中心化交易确认问题，使得跨链过程中的资产保管人角色由全网节点承担，而不是少数第三方。相较于多重签名，安全性更高，但实现更复杂。

公证人机制技术架构简单，对原链基本没影响，中心化程度越低，安全性越高，实现越复杂，需综合场景需求进行权衡。

2) 中继机制

中继的概念在生活中常见于通信场景，基站与基站之间搭建中继节点，以满足信号的多次转发。在跨链中，中继机制不依赖可信的第三方帮助其进行交易验证，“中间人”仅仅负责交易相关数据的收集与转发，目标链可以在拿到发送链的数据后自行验证。需要注意的是，两条链不能同时验证对方的交易，否则会陷入互为等待对方交易确认的死循环。整个过程中，“中间人”更多体现的是桥接的功能。因此相比于其他跨链技术，中继方案松耦合、更加灵活且易于扩展，具有多种实现形式，如 Cosmos 中的 Hub，Polkadot 中的 Relay Chain 等。

3) 侧链机制

侧链被定义为可以验证来自其他区块链数据的区块链，实现数字资产在多个区块链间的转移。侧链和中继的技术基础存在一定共性。一般说来，链 A 能够读懂链 B，那么表示 A 是 B 的侧链，主链可以不知道侧链的存在，但侧链必须知道主链，中继则必须知道两条链；侧链一般锚定主链，中继不存在这种从属关系，只负责数据传递；侧链一般基于 SPV 证明(参见 2.3.3 节中的简化支付验证)验证数据，需要同步所有的区块头，只能验证交易是否发生，中继一般不需要下载所有区块头。

早期的侧链方案基本都是针对比特币提出的，重构比特币的基础框架以弥补当初的设计缺陷(如吞吐量低、不支持图灵完备的智能合约)是极具风险的，因而利用侧链技术间接地扩展比特币的性能与功能，因此侧链受到主链的技术限制较多，可认为是一种强耦合结构的跨链模式，而中继机制更像是从各主链抽离出来的一个松耦合操作层。

对于交易的最终确定性，即应对交易可能撤销的场景(例如 PoW 共识只有概率确定性，存在分叉可能)的常见方案如下：

(1) 等待足够多的确认，这种方案的不足在于拉长了处理周期；

(2) 区块纠缠，令两个链之间建立依赖关系，当一个链上区块被撤销时，级联撤销关联链上相关区块；

(3) 使用强一致性的共识算法，如 PBFT 等。

以上只是概念范畴的辨析，实际应用中会根据项目愿景、应用场景进行各方权衡，组合使用各种技术，核心在于把握原链交易信息是如何传递、接受链如何对交易进行验证，最终确认交易。

2. 难点二：如何保证交易的原子性

跨链交易包含不同链上的多个子交易，这些子交易集构成一个事务，所有子交易要么都成功，要么都失败。跨链事务管理是实现跨链的关键技术，当前保证交易原子性的解决方案离不开“原子互换”，基于“原子互换”概念的具体实现是“哈希时间锁定合约”与其延伸的“哈希时间锁定协议”。

1）原子互换（Atomic Swaps）

原子跨链交换是一种实现多方跨多个区块链交换资产的分布式协调任务。原子性是计算机领域非常重要的概念，原子操作是不可分割的，在执行完毕之前不会被任何其他任务或事件中断，整个操作要么成功、要么失败，不存在中间状态。在区块链领域，原子互换已经在不断探索与尝试。2017 年 11 月，Lightning Labs 宣布它成功完成比特币和莱特币之间的首次链下原子互换。原子互换以去中心化的方式实现资产交易，在点对点的基础上实现两种加密货币的交换，无须第三方介入，也不存在交易一方在交易中违约的风险。原子互换是保障区块链间跨链交易原子性的基础协议，协议的具体实现有多种方式，下面设定一个交换场景，通过实例来理解原子互换的一般流程。

场景：两人想要以不通过中心化交易所的方式进行比特币与以太币的交换，在兑换比率达成一致后，用户 A 与 B 即可互换。然而，由于区块链上交易不可逆转，如果 A 先发送给 B 比特币，于 A 不利，因为并不确定 B 会发送给他以太币。因此，为了能使这种场景的交易履约进行，需要设计一种机制能够确保 A 和 B 都不会违背交易。原子互换通过“定时智能合约”（Timed Smart Contracts）来解决这种问题[①]，实现思路如图 3-11 所示。

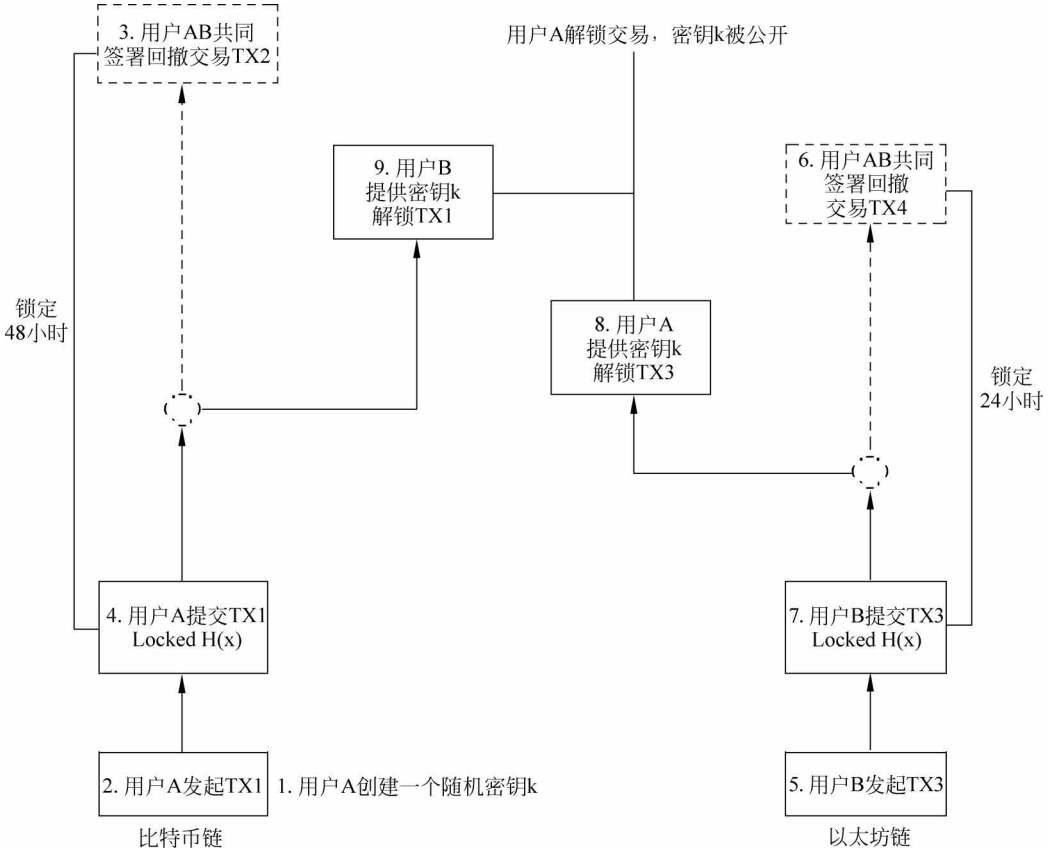


图 3-11　原子互换交易示例

① https://en.bitcoinwiki.org/wiki/Atomic_Swap

(1) 用户 A 创建一个随机密钥 k,该密钥只有用户 A 知道。

(2) 用户 A 在比特币链上创建交易 TX1:“Pay w BTC to <B's public key> if (k for Hash(k) known and signed by B)”,即用户 A 发起了向用户 B 转 w 个比特币的交易,解锁条件是提供密钥 k 与用户 B 的签名。

(3) 在 TX1 广播之前,用户 A 先在比特币链上广播一个回撤交易 TX2:“Pay w BTC from TX1 to <A's public key>,locked 48 hours in the future,signed by A”,即如果 48 小时内未有人解锁 TX1,那么将 w 比特币返还给用户 A。TX2 需要双方共同签名,才能生效。用户 B 同意交易,便签署 TX2,并返回给用户 A。

(4) 用户 A 在比特币链上提交 TX1,向全网广播。

(5) 用户 B 在以太坊币链上创建交易 TX3:“Pay v ETH to <A-public-key> if (k for Hash(k) known and signed by A)”,解锁条件是提供密钥 k 与用户 A 的签名。

(6) 同样,TX3 广播之前,用户 B 先在以太坊链上广播一个需要双方共同签名的回撤交易 TX4:“Pay v ETH from TX3 to <B's public key>,locked 24 hours in the future,signed by B”,即如果 24 小时内未有人解锁 TX3,那么将 v 以太币返还给用户 B。用户 A 看到用户 B 发起的 TX4,附上自己的签名,返回给用户 B。

(7) 用户 B 在以太币链上提交 TX3,向全网广播。

(8) 用户 A 为了获得 v 个以太币,便在以太币链上提供密钥 k,并附上自己的签名以解锁 TX3,交易成功后,用户 A 获得 v 个以太币,用户 B 也知晓密钥 k。

(9) 用户 B 利用密钥 k 与自己的签名在比特币链上解锁 TX1,最终获得用户 A 的 w 个比特币。

整个过程的关键在于用户 A 和用户 B 商定一个“定时智能合约”并先后锁定待转账的资产,“定时智能合约”约定如下:

(1) 条件 a:如果有人能在 T 小时内向智能合约输入随机密钥 k',并且能够验证 Hash(k')==m,那么用户 B 锁定的以太币将发送给用户 A,超时则将以太币返还用户 B;

(2) 条件 b:如果有人能在 $2T$ 小时内将原始密码 k 发送给智能合约,则用户 A 的比特币将自动转给用户 B,否则返还给用户 A。

条件 a 是约束用户 A 的,密钥只有 A 唯一知道,只要他提供密钥 k,合约验证肯定通过,只要不超时便可获得用户 B 被锁定的以太币。同时,密钥 k 被公开。用户 B 便可拿着公开的密钥 k,在 T 到 $2T$ 小时内发给合约,依照条件 b 获得用户 A 锁定在合约中的比特币。

依照流程,交易可以成功完成,我们再分析一下其他情况。

(1) 如果用户 A 始终不提供密钥 k,那么超时后锁定资产返回原所有者。

(2) 如果用户 A 在 T 至 $2T$ 时段提供密钥 k,那么用户 B 不仅会获得用户 A 的资产,原本锁定的资产也会返回。

(3) 如果用户 B 未在 T 至 $2T$ 时段提供密钥 k,那么用户 B 会丢失自己的以太币,而且也拿不到用户 A 的比特币。

情况(2)和情况(3)都包含着强约束,即“时间限制”和“强制执行交易”的机制迫使用户 A、B 理性选择,交易原子性可以获得保证。

2）哈希时间锁定合约

哈希时间锁定合约可以看作是原子互换的一种具体实现。哈希时间锁定合约（Hashed Timelock Contract，简称 HTLC）包含哈希锁定（Hashlock）以及时间锁定（Timelock）两个部分，这两个锁定机制保障了交易的原子性。HTLC 的典型代表就是比特币的闪电网络，其通过微支付通道（一种离链 off-chain 策略）来提升比特币交易处理能力。利用哈希锁定将发起方的交易代币予以锁定，再结合时间锁定，让接收方在某个约定时刻前生成支付的密码学证明，如果与先前约定的哈希值一致，则可完成交易。

（1）哈希锁定：我们在原子互换实例中提及的交易条件中出现了 Hash 函数，这种函数是单向，用户 A 可以用 Hash 函数对密钥 k 作计算得到摘要 m，但无法通过 Hash 函数与 m 反向计算得到 k，因此，Hash 函数与 m 都可以告诉用户 B。只要用户 A 公开密钥 k'，则用户 B 就可以利用 Hash 函数与 m 验证其提供的 k'是否就是真实的密钥 k。在比特币系统中，哈希计算操作通常用 OP_SHA256 或 OP_HASH160 来实现。

（2）时间锁定：我们在原子互换实例中还提及了回撤交易，只要交易未在指定时间范围内生效，则自动返回锁定的资产。时间锁定在比特币系统中有两种实现方式。

① OP_CHECKLOCKTIMEVERIFY：该操作码通常简称为 CLTV，是在 BIP-0065 中提出的，将特定事务冻结到将来的某个特定点，即允许在特定时间内冻结比特币交易。限定的时间是绝对时间，有两种表达方式，一种是时间戳，另一种是块高度。一般与 nLockTime 字段结合使用，当该操作码被调用时，会检查 nLockTime 字段，只有当 nLockTime 的时间大于或者等于 CLTV 参数指定的时间时，交易才会被完整执行。

② OP_CHECKSEQUENCEVERIFY：该操作码通常简称为 CSV，是在 BIP-0112 中提出的，相对于 CLTV 锁定的是绝对时间，CSV 锁定的是相对时间，例如：一年之后币可用。该码与 nSequence 字段配合使用，系统会检查 nSequence 字段，若其表示的相对时间大于或者等于 CSV 参数的时间，则交易开始执行。

3）哈希时间锁定协议

哈希时间锁定协议（Hashed Timelock Agreement，HTLA）可以看作是 HTLC 概念的泛化，可以用来在不支持 HTLC 的账本间执行 HTLC，跨的不仅是链，中心化或者去中心化账本都支持，在 Interledger 中应用了该理念。Interledger（ILP）是由 Ripple 发起的一个跨账本协同的协议，专注于实现连接各个账本不同资产的统一支付标准。支持的账本不仅包括区块链，还有银行、金融相关的各类传统中心化账本。在付款方和收款方中间，起到核心作用的是一系列的连接器（Connector），每个连接器犹如路由器节点一般试图将 ILP 数据包转发到更接近终点的地方，最终收敛到终点。付款方和连接器之间，连接器之间，连接器和收款方之间都是通过哈希时间锁定的概念来完成有条件的转账，并且可以扩展到支持多跳支付。每个参与者只需要信任直接对接的上下家即可。整个过程详情可以参见 3.2.3 节。

综上所述，第 3.2.2 节所有内容实际上对应着维塔利克·布特林（Vitalik Buterin）提出的三种跨链技术：公证人机制、中继/侧链、哈希锁定机制。下面就上述三种主要的跨链策略从互操作支持类型、信任模型、支持跨链资产互换、支持跨链资产转移等 7 个维度进行比较，如表 3-1 所示。

表 3-1　三种主要跨链策略对比

	公证人机制（Notaries）	中继（Relays）	哈希锁定机制（Hash-locking）
互操作支持类型	全部	全部（需要两条链上都有中继，否则仅支持单向的）	只支持交叉依赖
信任模型	大多数公证人诚实	链不宕机或者遭受“51%攻击”	链不宕机或者遭受“51%攻击”
支持跨链资产互换	是	是	是
支持跨链资产转移	是（但是要求共同的、长期公证人可信）	是	否
支持跨链预言机	是	是	不直接支持
支持跨链资产质押	是（但是要求长期公证人可信）	是	多数情况下支持，但是有难度
实现难度	中	高	低

3.2.3　典型跨链案例

跨链技术最早可以追溯至 2012 年，Ripple 公司致力于建立一套包容所有记账系统差异性的协议；随后，2014 年 BlockStream 团队首提侧链方案以寻求与其他区块链的互操作；再者有了 Ripple 对公证人机制、BTC-Relay 对中继机制的综合实践，再到现在 Cosmos、Polkadot 和万维链等关注跨链基础设施建设的不懈努力。下面将重点介绍若干典型的跨链平台与项目。

1. Interledger Protocol

Ripple 的 Interledger Protocol（ILP）是公证人机制的典型代表，ILP 让 Ripple 账本既可以连接其他区块链系统，也可以连接银行、自动交换中心（Automatic Clearing House，ACH）等传统金融机构的账本，其定义是“一个在不同分类账之间转移价值的协议”。该协议规定了 Sender（发起付款的一方）、Connector（转发数据包的，介于发送者和接收者之间的媒介）以及 Receiver（支付的最终接收者）三种角色（如图 3-12 所示），以及数据交换的顺序和内容。

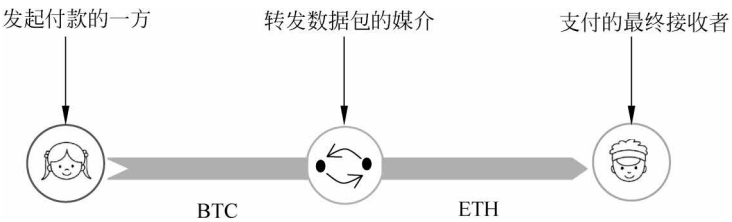


图 3-12　ILP 协议中 3 种角色示意图

该协议的核心之一就是 ILP packet，是上述三种角色通信最基础的数据结构。ILPv4 规定了三种 packet 类型：Prepare、Fulfill 与 Reject，大致对应请求、响应和错误消息。

Connector 转发从 Sender 到 Receiver 的 Prepare 数据包，Connector 回复从 Receiver 到 Sender 的 Fulfill 或者 Reject 数据包。

再者，“在不同分类账之间转移价值”这是将跨链的范畴扩大，不再局限于两个区块链账本间的资产互操作，而是任意账本之间（包含了中心化的银行账本等形式），旨在构建全球统一支付标准以解决银行间转账与汇款的高昂手续费用问题。

Interledger Protocol 包括以下核心内容。

1）协议簇

ILP 不是单个“Interledger Protocol”大型协议，而是由负责各种精细功能的协议簇组成，包含：

（1）Interledger Payment Request Protocol（IPR），支付申请；

（2）Pre-Shared Key Transport Protocol（PSK），负责预共享密钥传输；

（3）Simple Payment Setup Protocol（SPSP），负责简单付款设置；

（4）Interledger Quoting Protocol（ILQP），报价协议，也就是手续费；

（5）Connector-To-Connector Protocol（CCP），负责 connector 间的连接。

2）地址和路由

每一个账号在 ILP 网络中都有对应的唯一地址，地址是由“.”字符分隔的段组成的分层结构字符串，这是在 ILP 上识别账号的机制，形式与互联网的地址比较类似，例如：

```
g.crypto.xrp.ra5Kc69XKen6AHvsdFTKHfpG8VcrvUm9E8
```

这种地址称为目标地址，即接受付款的完整地址，这个地址与分类账中的账户是一一对应的，不可以以“.”符号结尾。此外，还有一种地址称为“地址前缀”，是一种模糊匹配的模式，代表一组账户、一类账户或者一个分区账户等。这类地址必须以“.”符号结尾，如 g.us.。这种地址是不可以接受付款的。

Connector 维护整个网络的路由表，当连接器接受查询时，会按照最长前缀匹配项原则基于若干连接器的路由表递归查找。Connector 基于路由表与数据包中目的地 ILP 地址进行数据包的转发。

3）报价

主要是估算向连接器支付的费用，即转账手续费，Interledger Quoting 协议详细规定了相关估算方法。

4）哈希时间锁定协议（HTLA）

哈希时间锁定协议上文已经提及，ILP 是依据资金托管设置的条件和时间限制来保障交易原子性的。该协议采用密码算法用 Connector 为这两个记账系统创建资金托管，当所有参与方对交易达成共识时，便可相互交易。

下面以 Alice 与 Bob 之间的支付场景为例，Alice 是 Sender，在区块链上有一个实现 HTLCs 的账户，Bob 是 Receiver，在银行有一个不实现 HTLCs 的账户，Chloe 是 Connector。Alice 有比特币账户，Bob 有美元账户，Chloe 拥有比特币账户和美元账户。Alice 只有比特币，想给 Bob 转账，但 Bob 只接受美元。

Interledger 转账流程如下所示（见图 3-13）。

（1）Alice 和 Bob 商议一个共享密钥（也许基于 Pre-Shared Key Transport Protocol）。

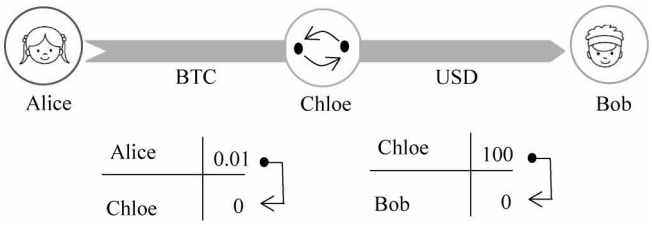


图 3-13　Interledger 转账示例

(2) Chloe 也许是一个流动性提供商，Alice 向 Chloe 咨询比特币与美元之间的汇率，假设是 0.01：100，同时 Chloe 收取一定手续费，最终 Alice 获知需要向 Chloe 支付 0.010 000 01 个比特币。

(3) Alice 构建 ILP 数据包，目标地址为 Chloe，数量是 0.010 000 01 个比特币，并附上基于共享密钥生成的托管条件以及超时时间，并在比特币账本系统发起“托管”操作。

(4) Chloe 监测到涉及自己的“托管”操作，解析获知自己需向 Bob 转 100 美元，因此，将 ILP 数据包中目标地址改为 Bob。

(5) Chloe 基于与 Bob 共享的 trustline 发起一个“托管”操作，设置了步骤(3)中的“托管”条件以及一个超时时间(要求小于步骤(3)中的超时时间)。

(6) Bob 监测到涉及自己的“托管”操作，在设定超时时间之前提供“共享密钥”，以通过“托管”操作携带的“条件”。

(7) Bob 确定后在 trustline 上发起一个“托管”确认操作，附上共享密钥，trustline 上的“托管”交易完成，Bob 获得 100 美元。

(8) Chloe 监测到涉及自己的“托管”确认操作，解析后获得共享密钥。

(9) Chloe 在比特币账本系统发起一个“托管”确认操作，附上共享密钥，比特币账本系统上的“托管”交易完成，Chloe 获得 0.010 000 01 个比特币。

对于没有直接支付通道的两个账本系统，还可以通过多跳间接跨账本交易，如图 3-14 所示。

图 3-14　多跳间接跨账本交易示意图

传统的中心化交易所其实也属于公证人机制，在交易双方都信任交易所的前提下，由交易所背书进行交易。这种方式适用性广，交易速度快，因此，仍然是目前最广为接受的方式，未来在跨链交易中，中心化交易所还是不可或缺的。

不过，完全中心化潜在的安全问题不可忽视，因此，去中心化交易所的概念不断被提出、践行，如 0x Project、Loopring 等项目。其核心在于由智能合约承担公证人角色，自动撮合交易。

2. BTC-Relay

侧链机制最初是为了扩展比特币功能而提出的，一般情况下，比特币为主链，作为侧链是要求能够读懂主链状态的。侧链机制可以看作比特币与其他区块链跨链交易的有益尝

试，ConsenSys 团队推出的 BTC-Relay[①] 是区块链生态系统中公认的第一条侧链，它是一种让比特币能够在以太坊区块链系统流通的跨链方案。BTC Relay 的本质其实是以太坊的一个智能合约，扮演了 Oracle 这样的角色。官网给出的架构如图 3-15 所示，图中涉及 Relayer、比特币交易与用户自定义的智能合约三种实体，它们之间的交互如下所述：

1）Relayers 持续往以太坊的 BTCRelay 提交比特币区块头信息；

2）以太坊上只需花费很少的手续费就可以通过 BTCRelay 验证比特币交易的有效性；

3）交易验证通过与否的信息会中继至用户自定义的智能合约。

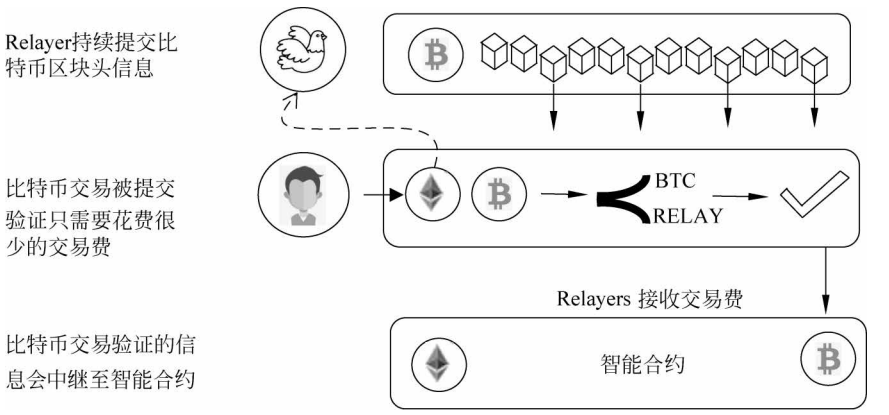


图 3-15　BTCRelay 示意图

BTC-Relay 跨链的实现是基于智能合约与 SPV 证明。这种部署在以太坊上的特殊的智能合约，能够让用户在以太坊上不通过任何第三方媒介就能够安全验证比特币交易，可以支持用户在以太坊 dApp 中使用 BTC 支付。这种特殊的智能合约与 SPV 证明之间离不开一个称为 Relayer 的角色，由其调用该智能合约，从而将最新的有效的比特币区块头信息不断地存入智能合约，并且设置了激励机制，每次调用验证比特币的交易，发起者都需要向提供交易的区块头的 Relayer 支付一笔手续费。为了防止 Relayer 要求过高的手续费，其他 Relayer 可以提供更低手续费来替换前者。因为任何人都可以向这个智能合约提交比特币区块头，这就存在合法性验证问题。该合约采用比特币一样的主链验证方法。当有新区块头提交，会先找到该区块头的父亲，进而计算整条链的工作量，只有累积工作量大于当前链的工作量，该新区块头才被接收。

下面以一个具体例子来阐述基于 BTC-Relay 跨链的流程，如图 3-16 所示。

场景是 Alice 有 BTC，想从 Bob 那儿换点 ETH，因此两人在以太坊区块链上构建一个“BTCSwap”的智能合约，Bob 将他的以太币发送给“BTCSwap”合约，并将之锁定。

（1）Alice 在比特币区块链上将 BTC 发送给 Bob，同时，她希望“BTCSwap”这个合约能够知晓这笔交易，并将 Bob 冻结的 ETH 转给她；

（2）Alice 基于比特币上的交易信息与“BTCSwap”合约地址来调用 BTCRelay. relayTx()；

（3）BTCRelay 首先调用 verifyTx 方法验证比特币交易有效性；

（4）调用 BTCRelay 验证比特币，需要向提供相关比特币区块头的 Relayer 支付一些手续费；

① https://github.com/ethereum/btcrelay

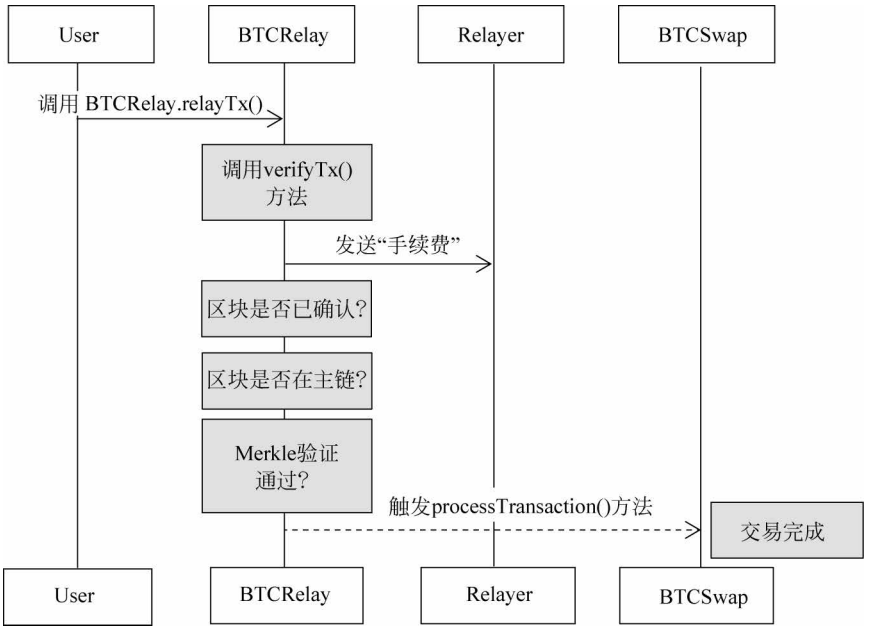


图 3-16　BTCRelay 跨链流程

(5) 一旦通过“区块是否已确认,区块是否在主链”等规则验证,将触发 BTCSwap 合约里面的 processTransaction()方法;

(6) BTCSwap 合约被触发后首先确认这个 BTCRelay 的合法性,通过后将解冻之前 Bob 的 ETH,整个交易完成。

受限于比特币的脚本语言,BTC-Relay 是单向解决方案,且由于 Relayer 提交区块头需要耗费一定的手续费(gas),假如调用 Relay 交易的奖励无法覆盖该手续费,从成本角度来讲较难持续。尽管如此,BTC-Relay 作为跨链通信的先驱者,对跨区块链通信做出了一次积极有意义的尝试,也启发了后来的一批项目,如双向锚定的元素链、非比特币的侧链 Lisk 等。

3. Cosmos

Cosmos 是由 Interchain 基金会(ICF)提供支持,Tendermint 团队发起的一个公有链项目,旨在构建“区块链的互联网”。

Tendermint 团队采用的是自下而上的设计方式,致力于构建底层的跨链基础设施,设立协议/标准,以便其他区块链能够便捷接入,实现天然具备跨链功能的平台。Tendermint 团队对区块链的业务逻辑进行了抽象,设计的框架如图 3-17(a)所示。

共识引擎与网络层封装为 Tendermint Core(这部分由 Tendermint 团队维护),Tendermint Core 包含了区块链运转所必备的基本功能,使得区块链开发人员从底层脱离,侧重自身应用逻辑的编写(这部分是由各个区块链开发团队完成)。其中共识机制采用的是拜占庭 PoS 算法,P2P 网络采用的是 Gossip 协议,Core 与应用层之间通过 ABCI 接口交互。这个共识算法是拜占庭容错的,只要网络中小于 1/3 的验证人是拜占庭节点,区块链就不会分叉,具备即时最终性,不像 PoW 是基于概率的,而且 TPS 是千级别的。基于这种架构模

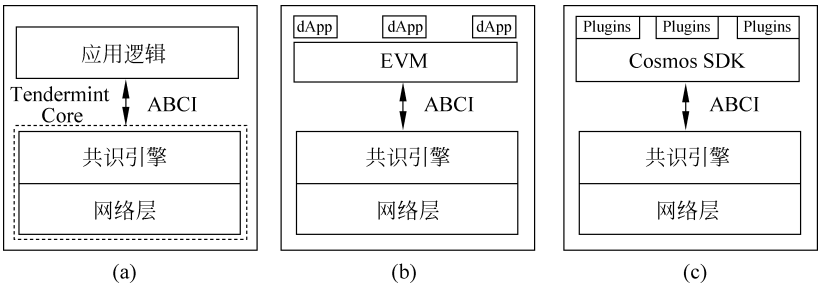


图 3-17　Cosmos 框架结构与实例化的产品

式，应用开发者可以迅速便捷地搭建自己的区块链。

如果我们基于上述框架实现比特币，那么，Tendermint Core 将实现：

(1) 节点间共享区块和交易；

(2) 建立权威的/不可更改的交易顺序。

Application 将负责：

(1) 维护 UTXO 数据库；

(2) 验证交易的数字签名；

(3) 阻止无效交易(如试图花费从不存在的交易输出)；

(4) 允许客户端查询 UTXO 数据库。

其次，Tendermint 团队基于该框架实现了以太坊的逻辑，该产品称为 Ethermint(Ether 与 Tendermint 的合成词)，此产品更换了原先以太坊底层的 PoW 共识机制，替换为高效 PoS 机制，而以太坊原先的智能合约等其他逻辑都可以迁移至应用层，Ethermint 的逻辑结构如图 3-17(b)所示。

此外，为了方便开发者实现基于 PoS 的区块链，Tendermint 团队还推出了 Cosmos SDK——一个可以快速开发区块链的工具套件，区块链相关常见功能都封装在 SDK 中，开发者只需专注应用逻辑的开发，如需定制化开发，SDK 提供 Plugin 机制进行扩展。Cosmos SDK 产品架构如图 3-17(c)所示。

基于上述统一标准化的框架或者工具套件，各自专注实现各自的业务逻辑，各个区块链之间以 IBC 交互，就会形成 Cosmos 生态系统，如图 3-18 所示。

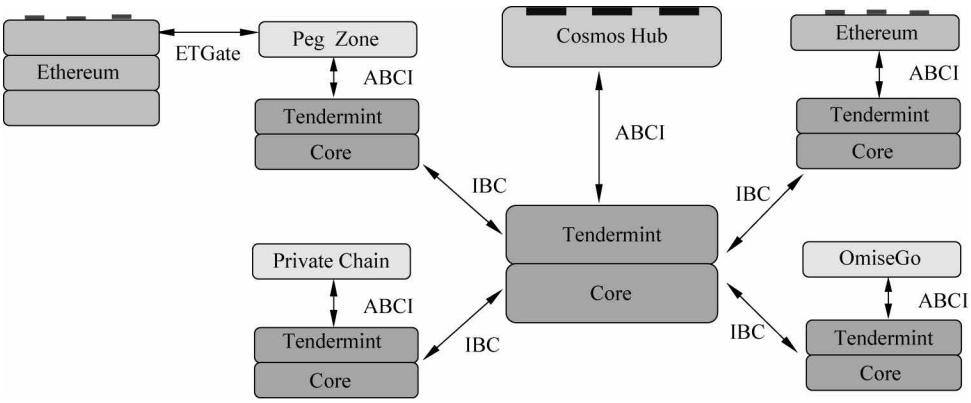


图 3-18　Cosmos 生态系统

在 Cosmos 中称一系列独立的区块链为“Zone”，这些 Zone 可以是 Ethermint，基于框架实现的私有链、联盟链，或者基于 Peg Zone 桥接的现有链（如基于 PoW 共识的），这些区块链依附于一个成为“Hub”的中心区块链，Zone 与 Hub 之间通过 IBC（Inter-Blockchain Communication）通信协议交互。

综上，可以认为 Cosmos 的架构支持以下 3 大基本功能：

（1）实现了区块链的网络以及共识机制（Tendermint Core）；

（2）支持各个应用（区块链）之间的 Token 的互换（IBC）；

（3）兼容现有区块链，支持以 Peg Zone 方式接入。

下面将重点阐述跨链实现的关键——IBC 协议[①]，Hub 与 Zone 之间的交互就遵循这个协议，并且 Token 的跨链转移也是通过它来实现的。假设现在有 3 个区块链：“Zone A”“Zone B”以及“Hub”，“Zone A”生成了一个数据包想通过“Hub”发送给“Zone B”。为了让数据包可以从一个区块链转移到另一个区块链，那么就需要在接收方区块链上发布一个证明，以明确发送链已经发起了一个到宣称目的地的数据包。接收方为了验证这个证明，就必须和发送方区块头保持一致。这种机制就与侧链采用的机制类似，它需要两个相互作用的链，通过双向的存在证明数据流（交易），来“知晓”另一方的情况。

因此，IBC 协议定义了两种交易类型的数据包：一种是 IBCBlockCommitTx，它允许区块链向任何观察员证明其最新区块哈希值，实际上是把发起链的当前最新区块头发送给目标链；另一个数据包类型是 IBCPacketTx，它不仅传递了跨链转 Token 的交易信息，还可以证明某个数据包确实是由发送者发布，通过默克尔证明机制（Merkle-proof）验证的。

其实，数据包从 Zone A 传到 Hub，再由 Hub 传到 Zone B，这里面涉及一个很重要的模式——中继，两个不同区块链之间的交互由一个第三方中继实现，它负责从发起链生成 Merkle Proof 并组装数据包，然后将其转发到目标链上，大致流程如图 3-19 所示[②]。

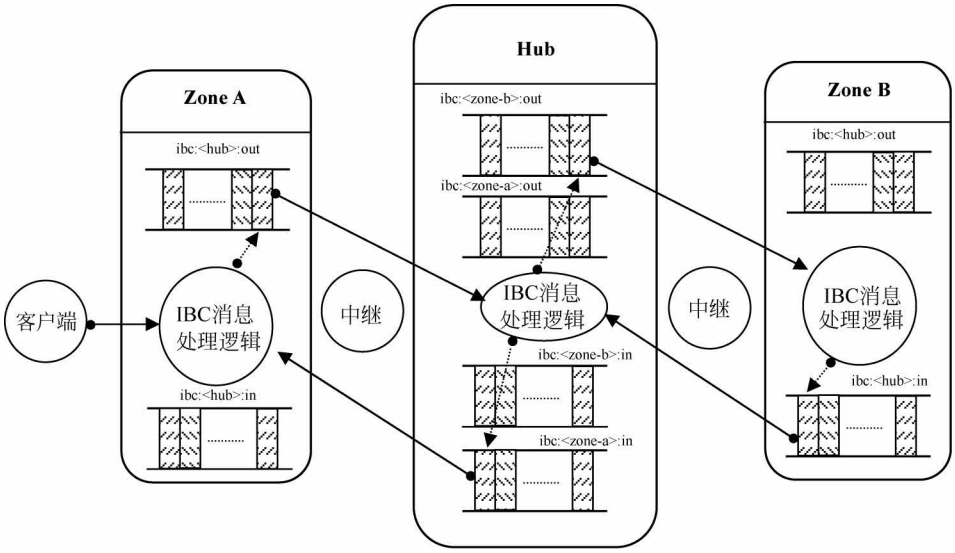


图 3-19　Cosmos 上跨链交易示意图

① https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md#inter-blockchain-communication-ibc

② https://xw.qq.com/cmsid/20180124A0QLPV

假设客户端想发起一个从 Zone A 到 Zone B 的一些 Token 的转移。消息先广播到 Zone A 处，Zone A 对消息进行处理，然后通过中继生成一个 Merkle Proof 并传送至 Hub，在 Hub 中进行验证后再通过中继传递至 Zone B，反之亦然。

Cosmos 的 IBC 协议不仅实现了数字 Token 跨链转移，其 IBC 协议中的 Payload 预留扩展机制，理论上可以传输其他类型的数据结构。Irisnet 将其扩展为一个跨链服务基础协议，让大家可以跨链调用不同的服务。

4. Polkadot

Polkadot 是 Web3 基金会支持，由以太坊前任 CTO 加文·伍德(Gavin Wood)主导的团队(开发以太坊钱包 Parity 的团队)研发的跨链项目。Polkadot 字面意思是 Polka dot(波尔卡点)，是一种均匀分布的圆点图案，寓意多链的并行存在。此外，Polka 还有另外一层寓意，它还指波尔卡舞，在 Tendermint 白皮书共识章节的配图中就有跳波尔卡舞的，可以看出 Polkadot 一定程度受到 Tendermint 的影响，两者都致力于实现万链互联的宏愿。具体而言，Polkadot 是一种可扩展的异构多链技术，它提供了一套通用的跨链协议，只要兼容此协议的区块链系统都可以实现跨链互联，Polkadot 所构想的是一种新型的区块链形态，它不只是一种区块链，而是一种新的范式——“链网”，由单独的中继链去统一管理共识安全和数据交互，用百花齐放的并行链技术去满足各种应用需求，进一步分离共识和状态转换，旨在解决当前区块链技术的伸缩性(Scalability)和隔离性(Isolatability)问题，以提供众多异构区块链系统之间去信任、去中心化的通用的互访问性、互操作性为目标。Polkadot 宣称是未来 Web 3 时代的基础设施之一(参见图 3-10 Web 3 技术栈概念图)，是一个未来“区块链互联网”的基础协议。

1) 总体框架

Polkadot 的架构如图 3-20 所示，由 Parachain(并行链)、Relay chain(中继链)以及 Bridge(桥接器)组成。

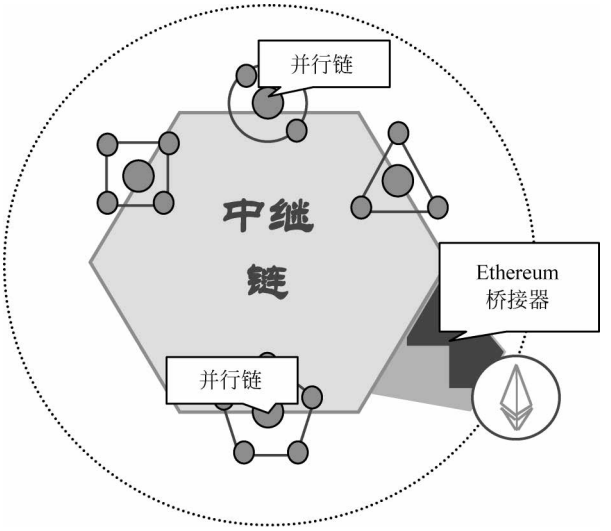


图 3-20　Polkadot 架构示意图

Relay chain 是核心，是一个功能类似 Cosmos Hub 的枢纽连接器，主要负责协调共识、并行链间跨链交易转发。Polkadot 受 Tendermint 和 HoneyBadgerBFT 启发[①]，采用了一种异步拜占庭容错算法以达成区块的共识。此外，为了激励验证人参与系统，还额外使用 PoS 共识机制选取验证人。Relay chain 本身不包含任何应用，这些均在 Parachain 上实现。

Parachain 是 Parallelizable Chain 的缩写，意思是可并行化的链，是指多条收集和处理交易并且传送至 Relay chain 的成员区块链，它们的安全性是附着于 Relay chain 的，每个参与的 Parachain 专注于特定领域的应用，这与 Cosmos Zone 较为类似，它们都是符合 Polkadot 规定标准/协议的新型区块链系统；Bridges 是兼容现有区块链（例如以太坊这类具有独立共识系统的链）的机制，当以太坊需要与 Polkadot 交互，则需要先针对以太坊开发相应的桥接器，这样以太坊的数据就可以转为 Polkadot 所需的交互格式。它的作用同 Cosmos Peg Zone 类似，Cosmos 和 Polkadot 都希望在启动时能够快速连接到以太坊主网。

因此 Polkadot 平台要求必须具备的特征如下所示。

（1）最小化（Minimal）：Relay chain 功能越少越好。

（2）简单性（Simple）：基础协议尽可能简单，尽量将功能推至 Parachain。

（3）通用性（General）：不该对加入的 Parachain 设置限制或先决条件，尽可能用抽象的模型来优化 Polkadot。

（4）健壮性（Robust）：Polkadot 需要提供一个稳定的基础层。

综上，Polkadot 只有保持最小的功能性，尽量把复杂的应用逻辑推至并行链，保持通用的跨链协议，核心在于保障基础层的安全与健壮性，才能让兼容此协议的外围链通过 Polkadot 互通互联，专注于自身的业务与创新。

为了支撑中继链，网络中设置了四种角色：收集人（Collator）、钓鱼人（Fisherman）、提名人（Nominator）、验证人（Validator）。

提名人是持有 Token（DOT）的权益群体，他们将手中的 Token 投票给候选人，其中得票最高的上百个候选人将成为验证人，验证人拥有最高权力——全系统出块权。同时，验证人需要抵押足够多的押金（Token），如果不履行职责，押金可能扣减或者全部丧失。当然，出块也有奖励，与之关联的提名人也会根据他们入金比例同奖同罚。收集人负责采集并行链的信息并把信息打包提交给验证人，他们是帮助验证人创造有效区块的群体，一般他们会运行某个并行链的全节点，他们为了更多的手续费，竞争性地收集信息。与验证人、收集人不同，钓鱼人并不直接参与区块打包相关的过程，他只负责监督系统内的非法行为，检查出来获得奖励。非法行为包括在并行链上批准一个无效的区块等。成为钓鱼人也是需要支付一定押金的，押金不多，主要用于预防浪费验证人计算时间和资源的女巫攻击。四种角色的关系如图 3-21 所示。

具体而言，经选举后中继链上会有 100 多个验证人（一般 144），中继链上包含了所有并行链的协议，能够识别并行链的交易格式，如比特币 UTXO 格式，还能解析并行链交易相关的证明数据。对于每一个区块，验证人都先随机分组，各个小组负责不同的并行链。每条并行链都有相关的收集人，他们把交易收集到区块里，并附上一个非交互的零知识证明（Noninteractive Zero-knowledge Proof），用来证明本子块的父块是有效的，一起交给负责该并行链的验证人

① https://polkadot.network/PolkaDotPaper.pdf

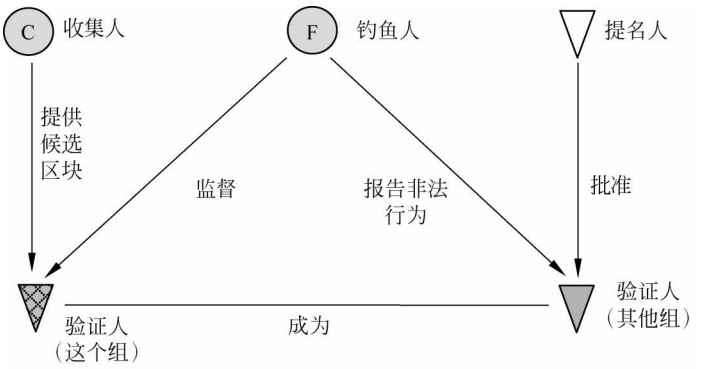


图 3-21　Polkadot 中 4 种角色交互示意图

小组，验证人小组随机选定候选块进行验证，共识出该高度的并行链区块。待所有并行链区块都确认，全体验证人再把消息路由至相应目的并行链，之后再共识中继链区块。下一周期，验证人再度打散，并行链将一致性让渡给中继链，自身只负责交易有效性，如此循环。

2）跨链流程

Polkadot 上跨链交易示意图如图 3-22 所示。有了 Relay chain 对全系统的安全性保障，Polkadot 上的链间通信就相对简单了：每个 Parachain 会维护一个出口（egress）以及入口（ingress）交易队列，队列利用 Merkle 树来保证数据真实。当一个 Parachain（A）向另一个 Parachain（B）发起跨链交易时，该交易会被推送到 A 的出口队列，然后由 Relay chain 把 A 的出口队列中的交易传送到 B 的入口队列，然后再由 B 自行处理其入口队列中的交易。

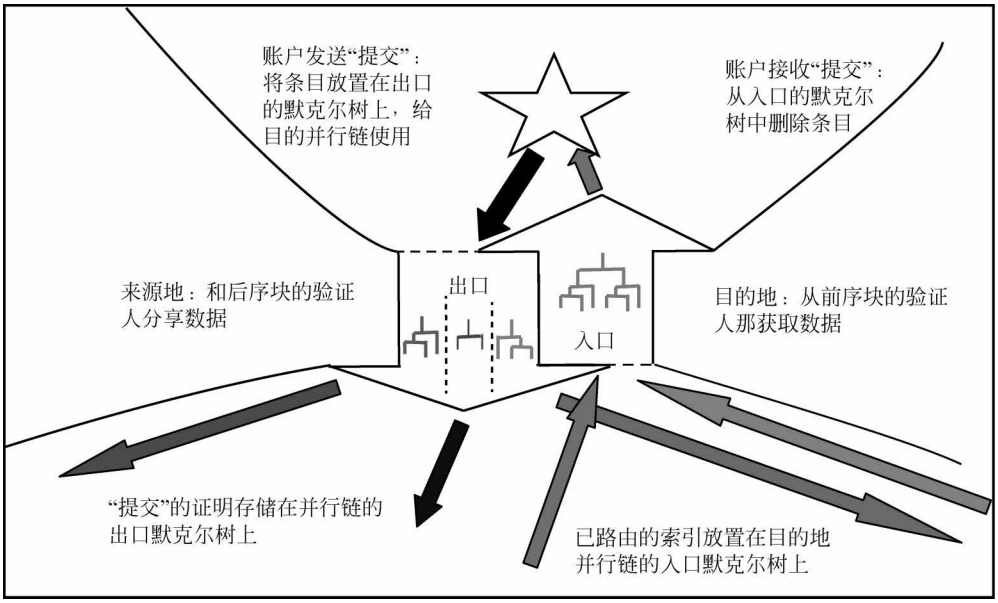


图 3-22　Polkadot 上跨链交易示意图

3）Substrate

类似 Cosmos 中 Cosmos SDK，Substrate 是面向区块链开发者的一个用于构建区块链的工具套件，使得开发者可以更方便地在网络上创建自己的区块链。

Substrate 主要提供如下功能：

（1）共识机制：提供拜占庭容错机制的功能；

（2）P2P 网络：例如 P2P 节点的搜索、同步等；

（3）WebAssembly 虚拟机：可以运行智能合约；

（4）轻客户端：支持低性能设备的直接接入；

（5）多语言版本：官方将提供 Rust、JS 的全协议客户端。

Polkadot 网络建立在 Substrate 之上，就像 Cosmos 网络建立在 Cosmos SDK 之上。开发者不用关心共识，而只要专注于区块链应用本身。

5. Wanchain

Cosmos 和 Polkadot 是设立标准/接口，其他区块链遵循，以主动兼容的方式去实现跨链交互，而 Wanchain（万维链）则是一条条去适配现有的异构链，将其纳入 Wanchain 平台，Wanchain 3.0 已打通了比特币、以太坊之间的跨链交易。

万维链项目由国内企业发起，旨在建立一个分布式的“银行”，一个可随意交易所有数字资产的未来银行。正如传统银行是现代金融的基础设施一样，万维链致力构建一个全新的、分布式的数字资产金融基础设施，以去中心化的方式完成不同区块链网络的连接及价值的交换。万维链的整体模型如图 3-23 所示[①]。

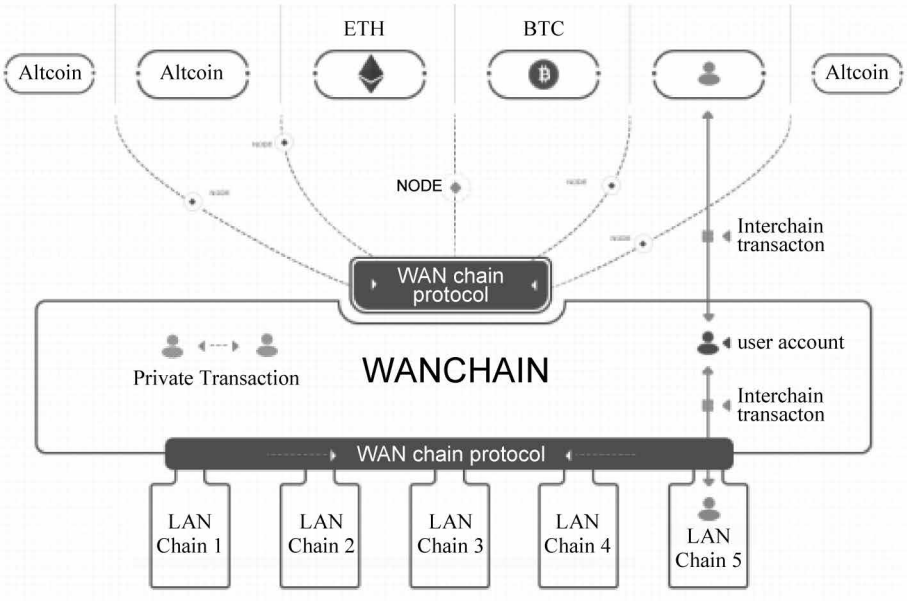


图 3-23　万维链的模型

从图 3-23 我们可以看出万维链主要面向两种区块链：一是现有区块链，如比特币链、以太坊链；二是遵循万维链协议开发的区块链，即图 3-23 中所示的 LAN Chain，这是面向传统金融资产上链的形式，这样不同类型的资产能够链接起来。

① https://wanchain.org/files/Wanchain-Whitepaper-EN-version.pdf

万维链通过安全多方计算(Secure Multi-Party Computation)和门限密钥共享机制(Threshold Key Sharing Scheme)并结合多角色节点设计,来完成跨链锁定账号的分布式秘钥管理,在不改变原有链机制的基础上通过跨链通信协议实现最小代价接入,从而链接不同区块链网络及数字资产,实现多种资产在万维链网络中的自由流通。

万维链采用的是 PoS 共识,万币(WANCoin)是万维链的原生币,总量为 2.1 亿个。万维链设想的跨链场景是这样的:假设 Alice 要将 1 个 BTC 转移至万维链,那么首先她需要在比特币链上发送一个 BTC 到锁定账户;万维链验证完毕后,万维链上 Alice 的账户会生成等值的锚定币 WBTC(即万维链上特有的 BTC),Alice 可以使用锚定币在万维链上流通,例如在去中心化交易所买 Bob 的 10 个 WETH(ETH 的锚定币),WETH 也可以转至 Alice 以太坊上的账户,变成 ETH。一旦资产返回原链,WBTC 或 WETH 会被销毁。不同公链的锚定币可以与万币兑换,支持者越多,意味着万维链接入应用越广。

区块链中打包交易并进行交易验证的节点称为验证节点,在万维链中,验证节点又细分为普通验证节点(Validator)、跨链交易证明节点(Voucher)和锁定账户管理节点(Storeman)。其中 Voucher 负责在跨链交易过程中提供原有链账户与锁定账户之间交易的证明,前提是 Voucher 需要缴纳一定的保证金,保证金交得高,它所提供的证明被采纳的概率也越高,与此同时,若提供虚假证明,它的保证金将会被扣除并剥夺其身份;Voucher 验证原链交易确认后,向 Validator 发送“true”的信号,而后 Validator 通知 Storeman 对锁定账户进行相关操作,并完成万维链账本的记账操作。Storeman 负责控制锁定账户,该类节点掌握锁定账号私钥的份额,在跨链交易过程中,使用掌握的密钥份额生成签名份额并合成完整签名,进而对锁定账户进行相关操作。Storeman 管理私钥的机制使用安全多方计算+门限密钥的技术,Storeman 必须共同参与计算才能生成锁定账号的公私钥,而私钥只是理论存在,从未出现在网络中,以碎片的方式分散在各 Storeman 手中,交易时参与方要再次合力才能共同构造签名,且互不泄露碎片。为了保证可用性,只需要一定比例的 Storeman 参与计算即可构造签名。

1) 跨链通信协议

万维链的跨链通信协议是万维链与其他区块链之间数据传输的规范,实现链间数据流转,由三个功能模块组成。

(1) 注册模块:主要实现两项功能。接入万维链的其他区块链首要完成的工作就是完成在万维链上的注册,万维链基于特定算法为原有链生成唯一标识,维护原有链标识注册表,避免虚假链造成欺骗;再者对请求转入的资产也需要完成万维链上的注册,确保万维链可以对该资产进行唯一识别,保证该资产在万维链上的流通性。

(2) 跨链交易数据传输模块:主要实现三项功能。一是支持原有链向万维链发起跨链交易请求;二是万维链验证节点应对跨链交易请求并返回操作成功与否的回执;三是万维链验证节点向原有链发送合法交易,以实现原有链资产转回。

(3) 交易状态查询模块:主要实现查询原有链状态数据的功能,以确认原有链上用户向万维链锁定账户转入资产的交易是否已被确认、万维链锁定账户向原有链上用户转回资产的交易是否已被确认。

2) 跨链交易示例

万维链实现跨链的关键在于锁定账户,当资产从原链转到万维链时,原链上的资产必须

被安全锁定；反之，当资产返回原链时，资产也能安全解锁。下面以以太坊为例详细说明公有链与万维链之间资产的转入转回，如图 3-24 和图 3-25 所示。

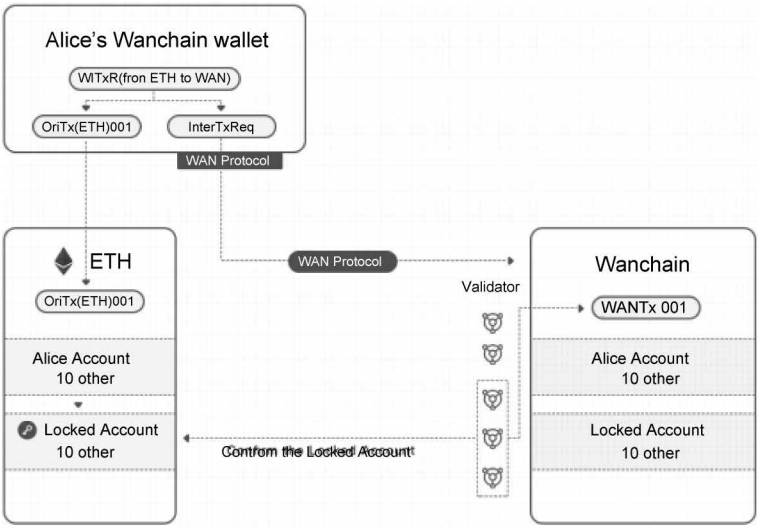


图 3-24　从以太坊至万维链跨链交易示意图

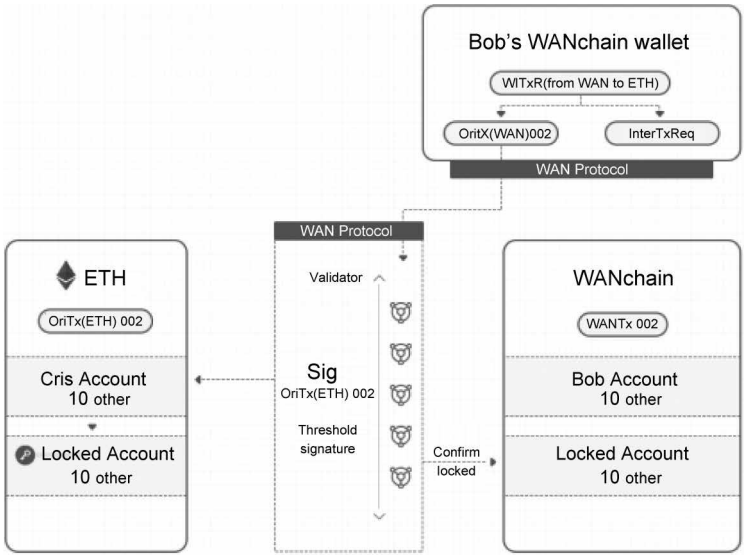


图 3-25　从万维链至以太坊跨链交易示意图

（1）转入过程：假设 Alice 和 Bob 在以太坊和万维链上都拥有账户，Alice 需要向 Bob 转移 10 个 ETH。首先 Alice 利用万维链钱包发起一个跨链交易请求，发送方是以太坊上的账户，接收方是万维链在以太坊上的跨链锁定账户（Locked Account）；万维链验证节点收到跨链交易请求并验证以太坊上该交易已完成记账，然后在万维链上创建一个新的智能合约代币 ETH'，ETH'就是 Alice 需要跨链转移的 ETH 在万维链上的等价物，并将该资产链转移到 Bob 在万维链的账户中。

(2) 转回过程：Bob 将从 Alice 处收到的 10 个 ETH 转给 Cris。Bob 使用他的万维链钱包向 ETH'资产合约发起一笔跨链交易，验证节点一旦收到该请求即将 Bob 的该笔资产对应的价值 10 个 ETH'转为锁定状态；成功锁定后，验证节点利用门限密钥共享机制构造出一笔以太坊交易，交易的转出方就是之前被锁定的 Alice 的锁定账户(Locked Account)，转入方是 Cris 在以太坊上的账户；在验证节点验证以太坊上的交易确认后，将 Bob 账号下锁定的 10 个 ETH'将被清空，意味着等值的资产已经回到原有链。

上述基于万维链的整个跨链过程比较安全(密码学保证)，无须第三方参与，原有链接入门槛较低，但是跨链的关键锁定账户的异常情况(如 Storeman 凑不齐密钥还原所需份额)处理还待完善。总之，万维链为跨链交易提供了新的方向，基于分布式密钥控制在跨链上实现锁定账户的管理，这是万维链不改变原链机制而实现万链互联的一个重要模式。