现在，让我们从比特币作为一个应用平台的角度来看，进一步加强理解。 现在很多人使用“blockchain”(区块链)这个词来表示任何共享了比特币设计原则的应用平台。 该术语经常被滥用，并被应用于许多不能提供比特币区块链主要功能的事情。

在本章中，我们将介绍比特币区块链作为应用平台所提供的功能。 我们将考虑应用程序构建原语，即构成任何区块链应用程序的构建块。 我们将研究使用这些原语的几个重要应用程序，例如彩色币，付款（状态）渠道和路由支付渠道（闪电网络）。

12.1介绍

比特币系统被设计为一个分布式的货币及支付系统。 然而，它的大部分功能源于可用于更广泛应用中的较低级别的结构。 比特币不是由帐户，用户，余额和付款等组件构建的。 相反的，我们在[交易]章节中看到，它使用的是具有低级加密功能的交易脚本语言。 就像账户，余额和付款的更高层次的概念可以从基本原语中衍生出来一样，许多其他复杂的应用也是如此。 因此，比特币区块链可以成为一个向诸如智能合同等应用程序提供信任服务的应用平台，远远超出了作为数字货币和支付的原始目的。

12.2构建块（原语）

当运行正常且长期运行时，比特币系统提供了一定的保证，可以作为构建块来创建应用程序。 这些包括：

杜绝双重支出 比特币分布式共识算法的最根本保证是确保UTXO不会花费两次。

不可改变性 一旦交易被记录在区块中，并且随后的区块中添加了足够的工作，交易数据就变得不可篡改。 不可改变性是由能源进行承保的，因为重写区块链需要消耗能源才能产生工作证明。 所需的能源以及由此带来的不可变性的程度随着在包含交易的区块之后被提交的工作量而增加。

中立 去中心化的比特币网络传播有效的交易，而不管这些交易的来源或内容如何。 这意味着任何人都可以支付足够的费用来创建有效的交易，并相信他们可以随时传输该交易并使其包含在区块链中。

安全时间戳 共识规则拒绝任何时间戳距离现在太远（过去和将来）的块。 这可以确保块上的时间戳可以被信任。 块上的时间戳意味着对所有其包含的交易的输入之前从未被花过的保证。

授权 被去中心化网络中验证过的数字签名可提供授权保证。没有脚本中指定的私钥的持有人的授权，包含数字签名要求的脚本就不能被执行。

审计能力 所有交易都是公开的，可以被审计。 所有的交易和交易所属的区块都可以以一个不间断的区块链链接起来并最终链接到创始区块。

会计 在任何交易中（coinbase交易除外），输入的金额等于输出的金额加上交易费用。 在交易中不可能创建或销毁比特币价值数值。 输出不能超过输入。

永不过期 有效的交易永远不会过期。 如果今天有效，它将在不久的将来仍然有效，只要输入仍然没有被花费，共识规则没有改变。

公正性 使用SIGHASH_ALL签名的比特币交易或由另一个部分由SIGHASH类型签署的交易，不能在签名还有效的情况下被修改，从而导致交易本身无效。

交易原子性 比特币交易是原子的（译者注：原子性是指交易要么全部执行，要么完全不执行，不存在中间状态）。它们要么是有效的并且经过确认的（挖矿），要么不是。 不存在挖矿出交易的一部分，交易也不存在中间状态。在任何时间点，交易要么被挖出，要么没有被挖，不存在中间状态。

离散（不可分割）价值单位 交易输出是离散和不可分割的价值单位。 他们要么整体被花费要么整体没有花费。 他们不能被分割或者部分被花费。

法定人数（注：让任何预定事物有效的最低参与人数） 脚本中的多重签名限制规定了多重签名方案中的预定义的法定权限。 M-of-N要求由共识规则执行。

时间锁/老化 包含相对或绝对时间锁的任何脚本语句只能在其时间超过指定时间后执行。

复制 区块链的去中心化存储确保了在交易在被开采之后，经过充分的确认，它被复制到整个网络上，并且变得可以耐受得起电力损失，数据丢失等的影响。

伪造保护 每笔交易只能花费现有的经过验证的输出。不可能创建或伪造价值。

一致性 在没有矿工分区的情况下，根据记录的深度，记录在区块链中的块可能会被从新组织或者被不认可的可能性将呈指数级下降。一旦被记录在深层，改变所需的计算和能量将大到不可行的程度。

记录外部状态 每个交易可以通过OP_RETURN提交一个值，表示外部状态机中的状态转换。

可预测发行量 总计不到2100万比特币将会以可以预测的速度发行。

上述构建块区的列表并不完整，还会有新功能都被介绍添加到比特币中。

12.3源于构建区块的应用

由比特币提供的构建区块是可信平台的组成部分，可用于构成各种应用程序。 以下是今天在用的应用程序及其使用的构建区块的一些示例：

Proof-of-Existence（Digital Notary）数字公证 不可篡改性+时间戳+永久性。数字指纹可以通过一个交易提交给区块链，以证明文件在此存档的时间内是存在的（Timestamp安全时间戳）。数字指纹不能在事后修改（Immutability不可改变性），证据将被永久存储（Durability耐久性）。

Kickstarter（Lighthouse） 一致性+原子性+可信。如果您发起众筹活动的一个输入和输出（Integrity公正性），别人可以参与众筹，但在目标（output value输出值）完成之前（Consistency一致性），这笔钱不能被花费出去（Atomicity交易原子性 ）。

Payment Channels 控制法定人数+时间锁+杜绝双重支付+永不过期+耐审查+授权。一个带有时间锁（Timelock时间锁）的法定人数为2-2的（Quorum法定人数）多重签名被作为付款渠道的“结算”交易时，可以被持有（Nonexpiration永不过期）或者可以在任何时间由任何一方授权（Authorization授权）的情况下（Censorship Resistance耐审查）进行花费。然后双方可以在更短的时间锁（Timelock）创建双重支出（No Double-Spend）结算的确认交易。

> （译者注：本段原文如下：Quorum of Control + Timelock + No Double Spend + Nonexpiration + Censorship Resistance + Authorization. A multisig 2-of-2 (Quorum) with a timelock (Timelock) used as the "settlement" transaction of a payment channel can be held (Nonexpiration) and spent whenever (Censorship Resistance) by either party (Authorization). The two parties can then create commitment transactions that double-spend (No Double-Spend) the settlement on a shorter timelock (Timelock).）

12.4染色币（Colored Coins）

我们将讨论的第一个区块链应用是染色币。

染色币是指利用比特币交易来记录除比特币之外的外部资产的创建，所有权和转让的这类技术。 所谓“外部资产”我们是指这些资产不直接存储在比特币区块上，而不是指比特币本身，因为比特币是本身就是这个区块链的固有资产。

染色币用于跟踪第三方持有的数字资产和实物资产，并通过染色币所有权证书来进行交易。 数字资产染色币可以代表无形资产，如股票证书，许可证，虚拟财产（游戏装备）或大多数任何形式的许可知识产权（商标，版权等）。有形资产染色币可以代表商品（黄金，白银，石油），土地所有权，汽车，船只，飞机等所有权。

该术语源于“着色”或标记某名义金额的比特币的想法，例如，1聪，用来表示比特币价值本身以外的东西。打个比方，我们给一美元的钞票标上一行信息说：“这是ACME的股票证书”，或者“这张钞票可以兑换1盎司的银”，然后使用这个1美元钞票与作为其他资产权益证明来进行交易。染色币的第一次实施，名为“基于增强填充订单的着色”或“EPOBC”，将外部资产标记于1聪输出上。这样，因为每个资产作为1聪的属性（颜色）被添加了，它就成了一个真正的“染色币”。

染色币的最新实施使用OP_RETURN脚本操作码将交易中的元数据与将元数据与特定资产相关联的外部数据存储结合在一起。

今天染色币的两个最突出的实现是 Open Assets和Colu的染色币。这两个系统使用不同的方法来染色，并不兼容。在一个系统中创建的染色币在其他系统中无法看到或被使用。

12.4.1使用染色币

染色币被创建，转移，并且通常用特殊的可以理解含有染色币协议元数据的比特币交易的钱包来查看。 必须特别注意避免在常规的比特币钱包中使用染色币相关的密钥，因为常规钱包可能会破坏元数据。 同样地，染色币也不应该被发送到由常规钱包管理的地址，而只能发送到由染色币能够识别的钱包管理的地址。Colu和Open Assets这两个系统都使用特殊的染色币地址来减轻这种风险，并确保染色币不会发送到不能识别的钱包。

染色币对大多数通用的区块链浏览器也是不可见的。 相反，您必须使用染色币浏览器来阐释染色币交易的元数据。

Open Assets兼容的钱包应用程序和区块链浏览器可以在coinprism查找。

Colu染色币兼容的钱包应用程序和区块链探索器可以在Blockchain Explorer中找到。

Copay钱包插件可以在Colored Coins Copay Addon中找到。

12.4.2发行染色币

每个染色币的实现都通过不同的方法创造染色币，但它们都提供类似的功能。创造染色币资产的过程称为*发行*。作为初始交易，发行交易将资产登记在比特币区块链上，并创建用于引用资产的资产ID。一旦发行，资产可以使用转账交易在地址之间传递。

作为染色币发行的资产可以有多种属性。它们可以是可分割的或不可分割的，这意味着转账中的资产量可以是整数（比如5）或具有十进制细分（比如4.321）。资产也可以*固定发行*，意思是一定数量的资产只可以发行一次，或者可以被再次发行，后者意味着原始发行人在初始发行后可以发行新资产单位。

最后，一些染色币启用分红，即允许按照拥有权成比例分配比特币付款给染色币资产的所有者。

12.4.3染色币交易

给染色币交易提供意义的元数据通常使用OP_RETURN操作码存储在其中一个输出中。不同颜色的硬币协议对OP_RETURN数据的内容使用不同的编码。包含OP_RETURN的输出称为*标记输出*。

输出的顺序和标记输出的位置在染色币协议中可能具有特殊含义。例如，在Open Assets中，标记输出之前的任何输出都代表资产发行。标记输出后的任何输出表示资产转账。通过参考各个输出在转账中的顺序标记输出将特定值和颜色分配给其他输出。

在Colored Coins (Colu)中，通过比较，标记输出编码一个定义元数据该如何被理解的操作码。操作码0x01至0x0F表示发行交易。发行操作码通常后面是资产ID或可用于从外部来源（例如bittorrent）取得资产信息的其他标识符。操作码0x10到0x1F表示转账交易。转账交易元数据包含简单的脚本，通过参考输入输出的索引（顺序），将特定数量的资产从输入转账到输出。因此，输入和输出的排序对脚本的解释很重要。

如果元数据太长而不能放入OP_RETURN，则染色币协议可能会使用其他"技巧"在交易中存储元数据。示例包括将元数据放在兑换脚本中，紧接着OP_DROP操作码，以确保脚本忽略元数据。另一种被使用的机制是1-N 多重签名脚本，其中只有第一个公钥是可以花费输出的真实公钥，随后的"密钥"则用被编码的元数据替代。

为了正确解释染色币交易中的元数据，您必须使用兼容的钱包或块资源浏览器。否则，该交易会看起来像一个具有OP_RETURN输出的"正常"比特币交易。

例如，我使用染色币创建并发行了MasterBTC资产。 "MasterBTC"代表了可以获取本书免费拷贝的兑换券。这些兑换券可以使用染色币兼容的钱包进行转让，交易和兑换。

对于这个特定的例子，我使用了https://coinprism.info的钱包和浏览器，它使用了Open Assets染色币协议。

下图12-1在coinprism.info上查看的发行交易 显示使用Coinprism块浏览器的发行交易：[(https://www.coinprism.info/tx/10d7c4e022f35288779be6713471151ede967caaa39eecd35296aa36d9c109ec)

正如你所看到的那样，coinprism显示了发行的20个"精通比特币的免费拷贝"，简称为MasterBTC的资产，发给了一个特殊的彩色币地址：

> akTnsDt5uzpioRST76VFRQM8q8sBFnQiwcx

警告 发送到该地址的任何资金或染色币将永远丢失。 不要发送到这个示例地址！

发行交易的交易ID是"正常"比特币交易ID。下图12-2**不对染色币进行解码的区块链浏览器中看到的发行交易** 显示同一笔交易（和12.1同一笔）在不会对区块链解码的区块浏览器中的样子。 我们将使用blockchain.info：https://blockchain.info/tx/10d7c4e022f35288779be6713471151ede967caaa39eecd35296aa36d9c109ec

正如你所看到的，blockchain.info不认为这是一个染色币交易。 事实上，它以红色字母表示第二个输出"无法解码输出地址"。

如果您在该屏幕上选择"显示脚本和coinbase"，可以看到有关交易的更多详细信息(下图12-3 发行交易的脚本).

再次，blockchain.info并不能理解第二个输出。 它以红色字母表示"奇怪"。 但是，我们可以看到，标记输出中的一些元数据是可读的：

```
OP\_RETURN 4f41010001141b753d68747470733a2f2f6370722e736d2f466f796b777248365559\
(decoded\) "OAu=[https://cpr.sm/FoykwrH6UY](https://cpr.sm/FoykwrH6UY)
```

让我们使用bitcoin-cli检索交易：

```
$ bitcoin-cli decoderawtransaction`bitcoin-cli getrawtransaction
10d7c4e022f35288779be6713471151ede967caaa39eecd35296aa36d9c109ec
```

剥离其余的交易，第二个输出如下所示：

```
{
    "value": 0.00000000,
    "n": 1,
    "scriptPubKey": "OP\_RETURN
4f41010001141b753d68747470733a2f2f6370722e736d2f466f796b777248365559"
}
```

前缀4F41表示字母“OA”，代表“Open Assets”，并帮助我们确定以下元数据是由Open Assets协议定义的紧接着的ASCII编码的字符串是指向资产定义的链接：

```
u=[https://cpr.sm/FoykwrH6UY](https://cpr.sm/FoykwrH6UY)
```

如果我们检索此URL，我们将获得JSON编码的资产定义，如下所示：

```
{
    "asset\_ids": \[
    "AcuRVsoa81hoLHmVTNXrRD8KpTqUXeqwgH"
    \],
    "contract\_url": null,
    "name\_short": "MasterBTC",
    "name": "Free copy of \"Mastering Bitcoin\"",
    "issuer": "Andreas M. Antonopoulos",
    "description": "This token is redeemable for a free copy of the book \"Mastering
Bitcoin\"",
    "description\_mime": "text/x-markdown; charset=UTF-8",
    "type": "Other",
    "divisibility": 0,
    "link\_to\_website": false,
    "icon\_url": null,
    "image\_url": null,
    "version": "1.0"
}
```

12.5合约币（Counterparty）

合约币是在比特币之上建立的协议层。与“染色币”类似的“合约币协议”提供了创建和交易虚拟资产和代币的能力。此外，合约币提供了去中心化的资产交换。合约币还在实施基于Ethereum虚拟机（EVM）的智能合同。

像染色币协议一样，合约币使用OP_RETURN操作码或1-N多重签名的公钥地址将元数据嵌入到比特币交易中，该地址用于代替公共密钥进行元数据编码。使用这些机制，合约币实现了在比特币交易中编码的协议层。额外的协议层可以由能理解合约币的应用程序来解读，如钱包和区块链浏览器，或使用合约币库（library）构建的任何应用程序。

反过来合约币可以用作给其他应用程序和服务的平台。例如，Tokenly是一个建立在合约币之上的平台，允许内容创作者，艺术家和公司发行表达数字所有权的代币，并可用于租赁，访问，交易或购买内容，产品和服务。利用交易合约币的其他应用包括游戏（Spells of Genesis）和网格计算项目（(Folding Coin）。

更多关于合约币的内容参见https://counterparty.io。开源项目可以在https://github.com/CounterpartyXCP中找到。

12.6支付通道和状态通道

支付通道是在比特币区块链之外双方之间交换的比特币交易的无信任机制。这些交易，如果在比特币区块链上结算，则是有效的，然而他们却是在链外被持有的，以期票的形式等待最终批量结算。由于交易尚未结算，因此他们可以在没有通常的结算延迟的情况下进行交换，从而可以满足极高的交易吞吐量，低（亚毫秒）延迟和精细（satoshi级）粒度。

实际上，*通道* 一词是一个比喻。状态通道是区块链外，由双方之间的交换状态代表的虚拟结构。实际上没有"渠道"，底层数据传输机制并不是渠道。我们使用通道这个术语来表示链外双方之间的关系和共享状态。

为了进一步解释这个概念，想一想TCP流。从高层协议的角度来看，它是一个横跨互联网连接两个应用程序的"socket"。但是，如果您查看网络流量，TCP流只是IP数据包之上的虚拟通道。 TCP流的每个端点通过排序并组装IP数据包以产生字节流的错觉。实际上在背后，所有的数据包都是断开分散的。同理，支付通道只是一系列交易。如果妥善排序和连接，即使您不信任通道的另一方，（经过排序连接后的交易）也可以创建可以信任的可兑换的债务。

在本节中，我们将介绍各种形式的支付通道。首先，我们将检视用于构建计量小额支付服务（如流媒体视频）的单行（单向）支付通道的机制。然后，我们将扩大这一机制，引入双向付费渠道。最后，我们将看看首先在 *闪电网络* 中提出的，如何将路由网络中的双向通道端到端链接从而形成多跳通道。

支付通道是 状态通道的引申概念之一，代表了链外状态的变化，通过区块链上的最终的结算得到保障。支付通道是一种状态通道，其中被改变的状态是虚拟货币余额。

12.6.1状态通道基本概念和术语

通过一个交易在区块链上所锁定的共享状态，在交易两方之间建立了一个状态通道。这被称为资金交易或锚点交易。这笔交易必须传送到网络并开始挖矿被挖矿确认以建立通道。在支付通道的示例中，锁定的状态即为**通道的初始余额（以货币计）。

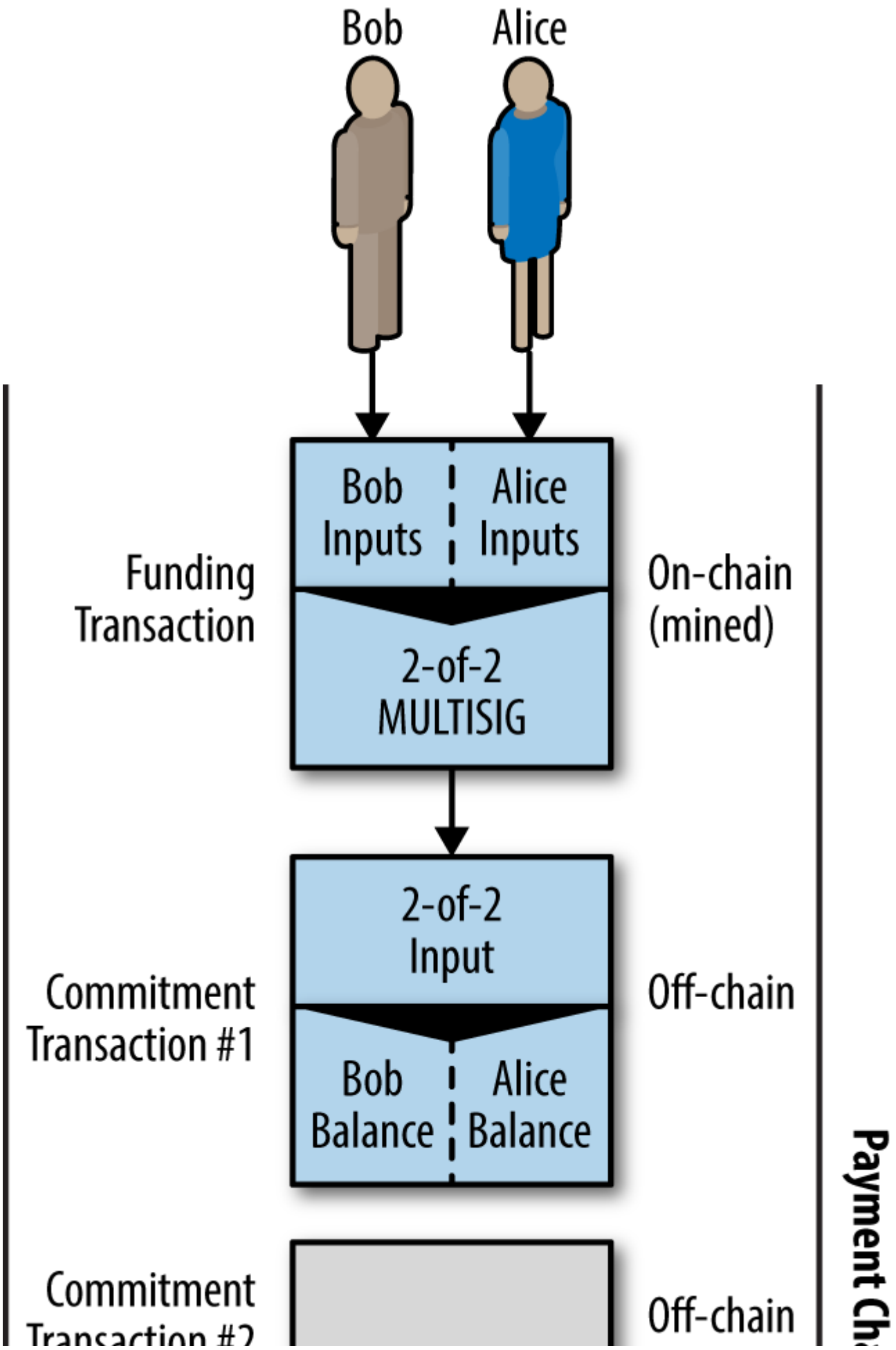
随后双方交换已签名的交易，这被称为"承诺交易"。承诺交易会改变初始状态。这些交易都是有效的，因为任何一方都可以提交结算的请求，不需要等到通道关闭再做结算。任何一方给对方创建、签名和发送交易时就会更新状态。实践中，这意味着每秒可进行数千笔交易。

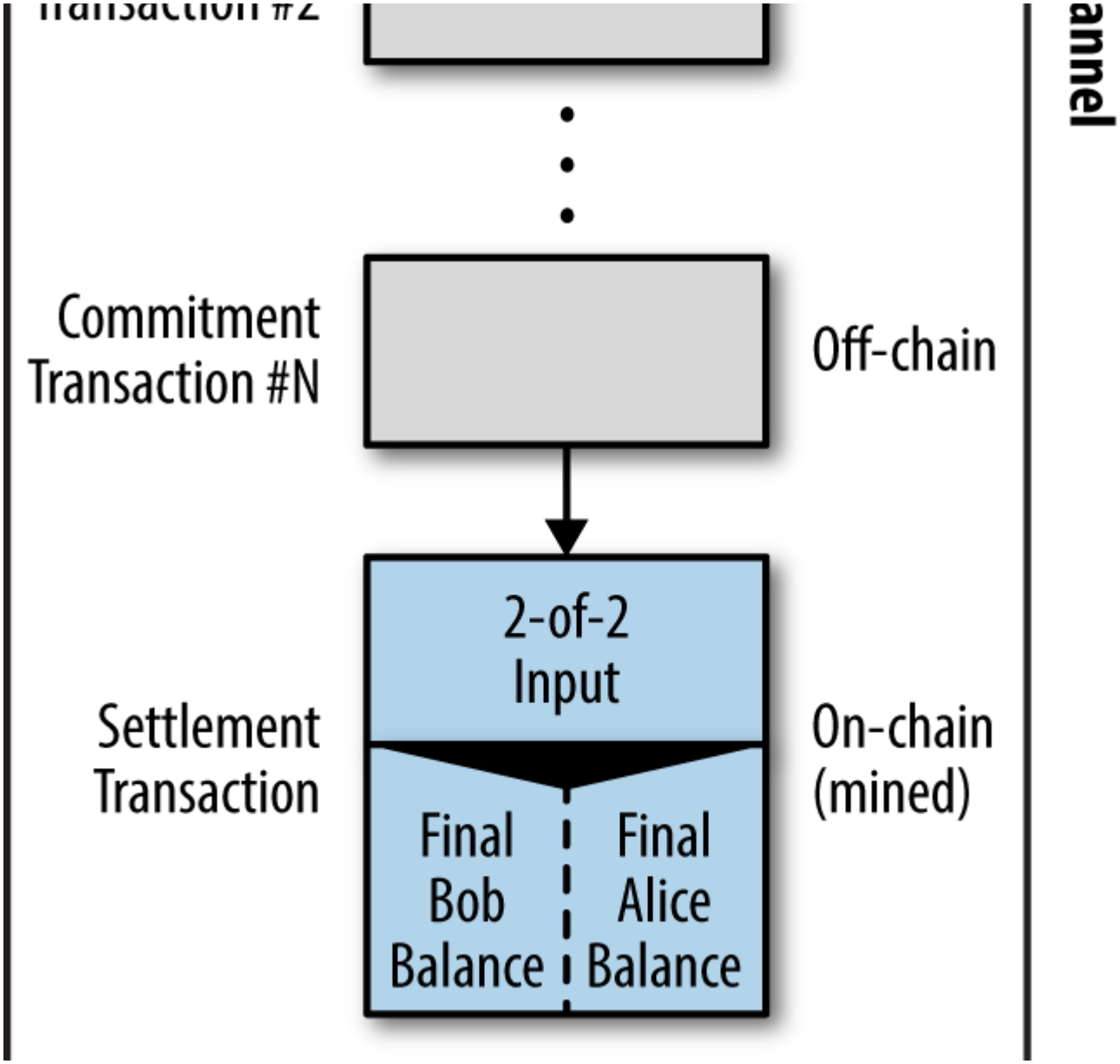
当交换承诺交易时，双方同时废止之前的状态，如此一来最新的承诺交易总是唯一可以赎回的承诺交易。这样可以防止任何一方在通道中某个先前状态比最新状态更有利于己方的时候通过单方面关闭通道来进行欺骗。我们将在本章的其余部分中检视可用于废止先前状态的各种机制。

最后，通道可以合作关闭，即向区块链提交最后的结算交易，或者单方面由任何一方提交最后承诺交易到链上。单方面关闭的选项是必要的，以防万一交易中的一方意外断开连接。结算交易代表通道的最终状态，并在链上进行结算。

在通道的整个生命周期中，只有两个交易需要提交给链上进行挖矿：资金交易和结算交易。在这两个状态之间，双方可以交换任何数量的承诺交易，任何其他人永远不会看到，也不会提交到链上。

下图12-4说明了Bob和Alice之间的支付通道，显示了资金，承诺和结算交易。


Bob
Alice
Bob Inputs
Alice Inputs
Funding Transaction
On-chain (mined)
2-of-2 MULTISIG
2-of-2 Input
Commitment Transaction #1
Off-chain
Bob Balance
Alice Balance
Commitment
Off-chain
Payment Ch



12.6.2简单支付通道示例

要说明状态通道，我们必须从一个非常简单的例子开始。 我们展示一个单向通道，意味着价值只向着一个方向流动。 为了便于解释，我们以一个天真的假设开始，假设没有人要试图欺骗他人。 一旦我们解释了基本的通道概念，我们将会接着看看是什么使得支付通道可以无信任化，从而让交易双方哪怕去尝试进行欺骗都无法成功。

对于这个例子，我们假设两个参与者：Emma和Fabian。 Fabian提供由微支付通道支持以秒为单位时长计费的视频流服务。Fabian每秒视频收费0.01毫比（millibits）（0.00001 BTC），相当于每小时36毫比（0.036 BTC）的视频。 Emma是从Fabian那里使用以秒计费的支付通道来购买流媒体视频服务的用户。下图12-5显示Emma使用支付通道从Fabian购买视频流服务。

在这个例子中，Fabian和Emma正在使用专门的处理支付通道和视频流的软件。 Emma在浏览器中运行该软件，Fabian从服务器端运行该软件。该软件包括基本的比特币钱包功能，可以创建和签署比特币交易。“支付通道”的概念和术语对于用户都是完全不可见的。他们看到的是以秒为单位支付了的视频。

为了设置支付通道，Emma和Fabian建立了一个2-2的多重签名地址，双方各持一个密钥。从Emma的角度来看，她的浏览器中的软件提供了一个带有P2SH地址的二维码（以“3”开头），并要求她提交最多1小时视频的“押金”。该地址因而得到了Emma的注资。支付给该多重地址的Emma交易，就是支付通道的资金交易或锚点交易。

就这个例子而言，我们假设Emma支付了36个毫比（0.036 BTC）到通道中。这将允许Emma消费长达1小时的流媒体视频。这笔资金交易设定了可以在这个通道上发送的最大数量（数据量），即设置了通道容量。

资金交易从Emma的钱包中消耗一个或多个输入以集成资金。它创建一个价值为36毫比的输出，支付给Emma和Fabian之间共同控制的多重签名2-2地址。它也可能有一个作为找零钱到Emma的钱包的额外输出。

一旦资金交易得到确认，Emma可以开始观看视频。Emma的软件创建并签署一笔承诺交易，改变通道余额，将0.01毫比归入Fabian的地址，并退回给Emma的35.99毫比。Emma签署的交易消耗了由资金交易创造的36毫比输出，并创建了两个输出：一个用于找钱，另一个用于Fabian的付款。交易只是部分被签署了 - 它需要两个签名（2 - 2），但只有Emma的签名。当Fabian的服务器接收到此交易时，它会添加第二个签名（用于2-2输入），并将其返回给Emma并附带时长1秒的视频。现在双方都有谁都可以兑换的完全签署的承诺交易，这个承诺交易代表着通道中的最新正确余额。双方都不会将此交易广播到网络中。

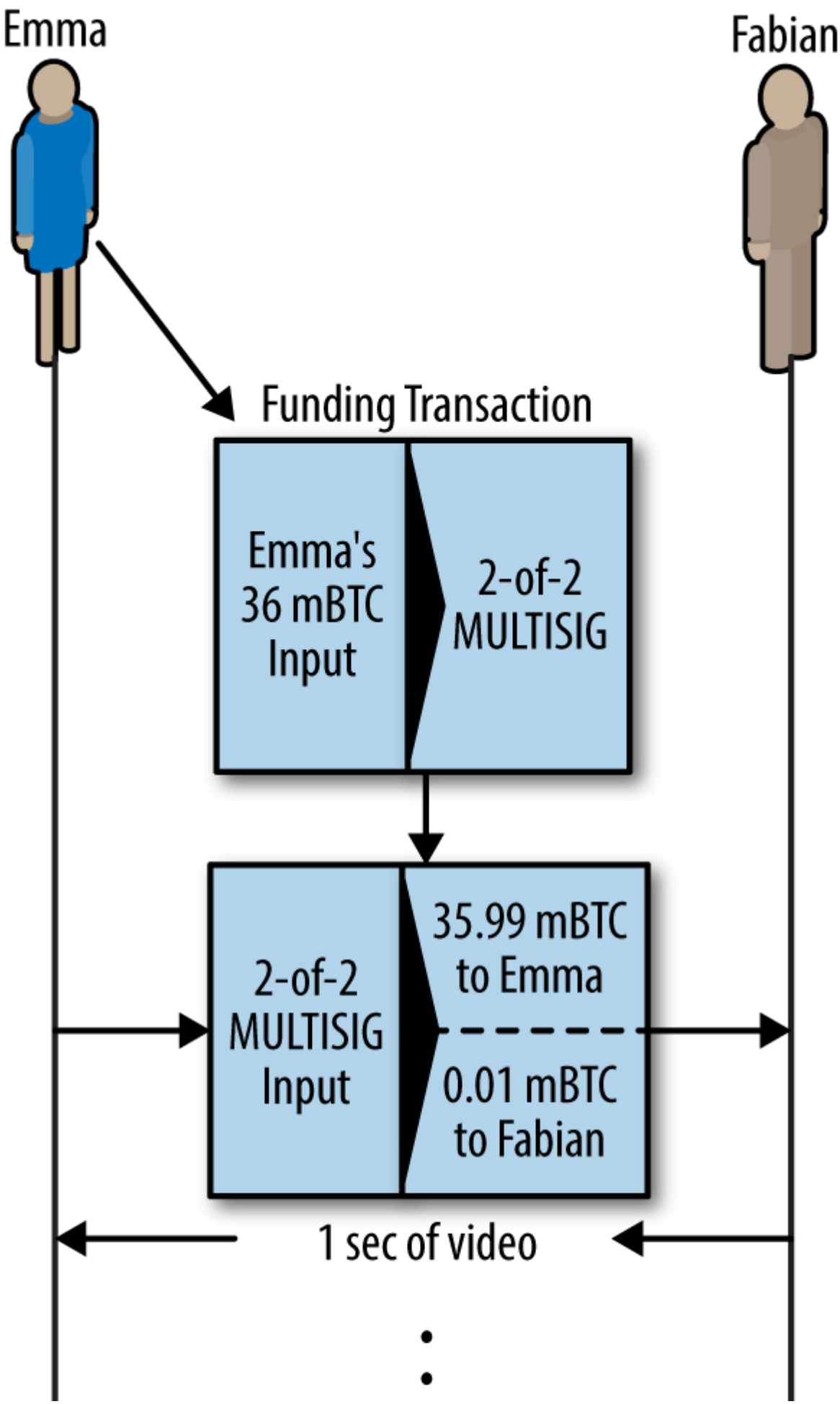
在下一轮，Emma的软件创建并签署另一个承诺交易（承诺2号），该交易从资金交易中消耗相同的2-2输出。二号承诺交易分配一个0.2毫比的一个输出到Fabian的地址，还有一个一个输出为35.98毫比，作为找零返回给Emma的地址。这个新交易支付的是连续两秒的视频内容。 Fabian的软件签署并返回第二个承诺交易，再加上视频的另一秒内容。

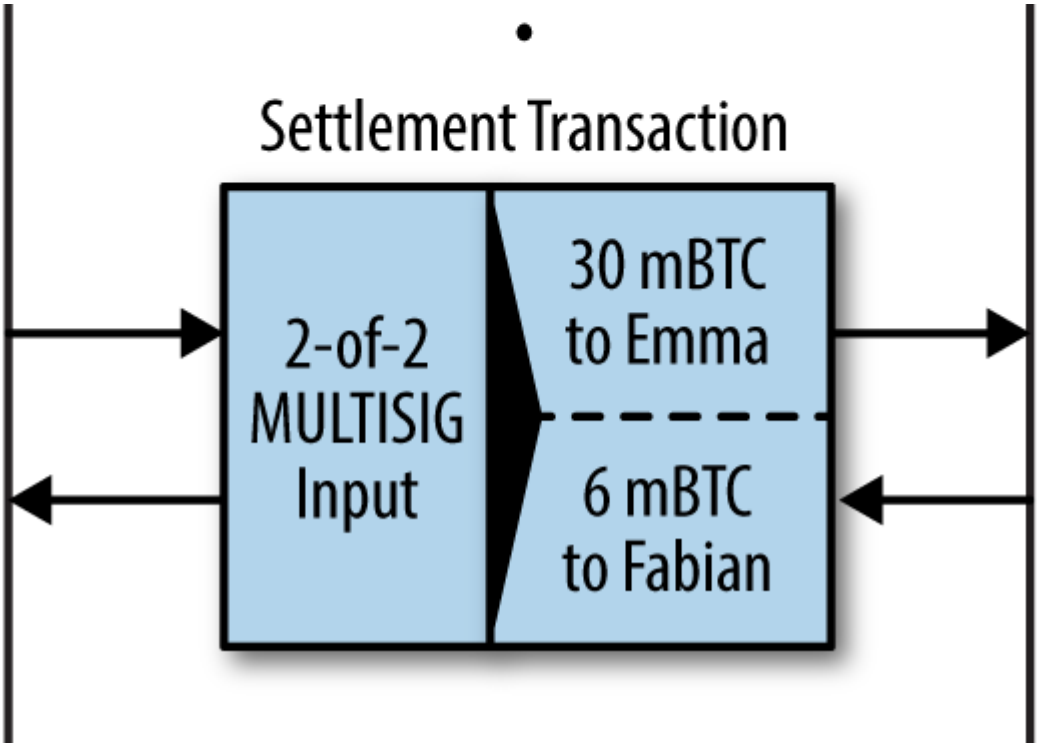
利用上述的方法，Emma的软件继续向Fabian的服务器发送承诺交易，以换取流媒体视频。因为Emma观看了更多秒数的视频，通道中属于Fabian的钱逐渐累积变多。假设Emma观看600秒（10分钟）的视频，创建和签署600笔承诺交易。最后的承诺交易（＃600）将有两个输出，将通道的余额分成两半，分别为6毫比属于Fabian和30毫比属于Emma。

最后，Emma点击“停止”停止流媒体视频。 Fabian或Emma现在可以发送最终状态交易以进行结算。最后一笔交易即为结算交易，向Fabian支付所有Emma消费的视频，并向Emma退还资金交易中剩余的资金。

图12-6显示了Emma和Fabian之间的通道以及更新通道余额的承诺交易。

最后，只有两个交易记录在块上：建立通道的资金交易和在两个参与者之间正确分配最终余额的结算交易。


Emma
Fabian
Funding Transaction
Emma's
36 mBTC
Input
2-of-2
MULTISIG
2-of-2
MULTISIG
Input
35.99 mBTC
to Emma
0.01 mBTC
to Fabian
1 sec of video



12.6.3制造无需信任的通道

我们刚刚描述的通道只有在双方合作，没有任何失败或企图欺骗的情况下工作。我们来看看破坏这个通道的一些场景，并且看看需要什么来修复这类场景：

- 一旦资金交易发生，Emma需要Fabian的签名才能获得给自己的找零。如果Fabian消失，Emma的资金将被锁定在2-2中，并彻底损失。这个通道一旦建立，如果在双方共同签署至少一个承诺交易之前，有任何一方断开，就会导致资金的流失。
- 当通道正在运行时，Emma可以采取Fabian已经签署的任何承诺交易，并将它发回链上。如果她可以发送承诺交易＃1，只支付1秒的视频，为什么要支付600秒的视频？通道失败是因为Emma可以通过广播对她比较有利的先前的承诺来欺骗。

这两个问题都可以用时间锁(timelocks)来解决 - 我们来看看我们如何使用交易级时间锁（nLocktime）。

除非她有保证的找零退款，否则Emma不能冒风险进行2-of-2 签名。为了解决这个问题，Emma同时建立了资金和退款交易。她签署资金交易，但不传送给任何人。Emma只将退款交易传送给Fabian，并获得他的签名。

退款交易作为第一承诺交易，其时间锁确立了通道生命的上限。在这种情况下，Emma可以将nLocktime设置为30天或将来的第4320个区块。所有后续承诺交易必须具有较短的时间锁，以便在退款交易之前能把它们赎回。

现在，Emma已经有一个完全签署的退款交易，她可以自信地发送签署过的资金交易，因为她知道她最终可以在时间到期后最终赎回退款交易，即使Fabian消失也不会有问题。

在通道生命中双方交换的每一项承诺交易都会被时间锁锁进未来时间点。但是，对于每个承诺，延迟时间会稍短一点，所以最新的承诺可以在被它废止的前一承诺之前被赎回（译者注：上文提到如果有个最新承诺，前面的承诺就已经作废）。由于nLocktime，任何一方都只有其时间到期后才能成功传播任何承诺交易。如果一切顺利，他们将合作并通过结算交易合理地关闭通道，这样一来发送中间的承诺交易就不必要了。实质上说，承诺交易只在一方断线而另一方不得不单方面关闭通道时才使用。

例如，如果将来承诺交易＃1被时间锁定到将来的第4320个块，则将来承诺交易＃2被时间锁定到将来的4319个块。（同理可知，）承诺交易＃600则可以在承诺交易＃1变为有效之前600个块的时间被消费。

图12-7显示每个承诺交易设置较短的时间锁，允许在它在之前的承诺变为有效前被花费

每个后续承诺交易必须具有较短的时间锁，以便可以在其前任之前和退款交易之前进行广播。能够尽早广播承诺交易的能力确保了承诺交易能够花费资金输出，并排除任何其他承诺交易通过话费资金输出来赎回。比特币区块链提供的担保，即防止双重支出和执行时间锁定，有效地允许每个承诺交易废止其前任有效性。

状态通道使用时间锁来在时间维度中执行智能合约。在这个例子中，我们看到时间维度如何保证最近的承诺交易在任何早先的承诺之前变得有效。因此，最近的承诺交易可以传输，消费输入和使先前的承诺交易无效。绝对时间锁定的智能合同的执行可以防止其中任何一方的欺骗。此实现只需要绝对的交易级时间锁（nLocktime）。接下来，我们将看到如何使用脚本级时间锁定，CHECKLOCKTIMEVERIFY和CHECKSEQUENCEVERIFY来构建更灵活，有用和复杂的状态通道。

第一次出现的单向支付通道在2015年由阿根廷开发商团队演示为视频流应用样板。你仍然可以在streamsium.io看到它。

时间锁并不是使先前的承诺交易无效的唯一方法。在接下来的章节中，我们将看到如何使用撤销密钥来实现相同的结果。时间锁是有效的，但其有两个明显的缺点。在通道首次打开时建立最大时间锁，限制了通道的使用寿命。更糟糕的是，他们迫使通道实现以在允许长期通道，和迫使其中一位参与者在提前关闭的情况下等待很长时间的退款之间取得平衡。例如，如果允许频道保持开放30天，通过将退款时间设置为30天，如果其中一方立即消失，则另一方必须等待30天才能退款。终点设置越远，退款时间越远。

第二个问题是，由于每个后续的承诺交易必须减短时间锁，所以在双方之间可以交换的承诺交易数量有明确的限制。例如，一个30天的通道，设置了位于未来第4320个块的时间锁，在必须被关闭前只能容纳4320个中间承诺交易。将时间锁定承诺交易的间隔设置为1个区块存在危险。通过将承诺交易之间的时间锁设置为1个区块，开发者给通道参与者带来了非常高的负担，参与者必须保持警惕，保持在线并监视，并随时准备传送正确的承诺交易。

现在我们了解如何使用时间锁来使先前的承诺无效，我们可以看到合作关闭通道和通过广播承诺交易单方面关闭通道之间的区别。所有承诺交易都是时间锁定的，因广播承诺交易总是要等待时间到期。但是，如果双方同意最后的余额是多少，并且知道他们都承担最终实现余额的承诺交易，那么他们可以构建一个没有时间锁代表相同余额的结算交易。在合作关闭中，任一方都可以提取最近的承诺交易，并建立一个各方面完全相同的结算交易，唯一差别就是结算交易省略了时间锁。双方都可以签署这笔结算交易，因为知道无法作弊以得到更多的余额。通过合作签署和发送结算交易，可以立即关闭通道并兑换余额。最差的情况下，当事人之一可以是卑鄙小人，拒绝合作，强迫另一方单方面关闭最近的承诺交易。但是如果他们这样做，他们也必须等待他们的资金。

12.6.4不对称可撤销承诺

处理先前承诺状态的更好方法是明确撤销它们。但是，这不容易实现。比特币的一个关键特征是，一旦交易有效，它一直有效，不会过期。取消交易的唯一方法是在交易被挖矿前用另一笔交易双重支出它的输入。这就是为什么我们在上述简单支付通道示例中使用时间锁定，以确保最新的承诺交易可以在旧承诺生效之前被花费。然而，把承诺在时间上排序造成了许多限制，使得支付通道难以使用。

虽说一个交易无法取消，但是它可以被构造成无法再使用的样子。我们这样做我们实现它的方法是通过给予每一方一个 撤销密钥，如果对方试图欺骗，可以用来进行惩罚。撤销先前承诺交易的这种机制首先被作为闪电网络的一部分提出。

为了解释撤销密钥，我们将在由Hitesh和Irene经营的两个交易所之间构建一个更加复杂的支付通道。 Hitesh和Irene分别在印度和美国运营比特币交易所。 Hitesh的印度交易所的客户经常向Irene的美国交易所的客户发送付款，反之亦然。目前，这些交易发生在比特币链上，但这意味着支付手续费用并等待几个块进行确认。在交易所之间设置支付通道将大大降低成本并加快交易流程。

Hitesh和Irene通过合作建立资金交易来启动通道，每人向通道注资5个比特币。初始余额为Hitesh有5比特币且Irene有5比特币。资金交易将通道状态锁定在2-2多重签名中，就像在简单通道的例子中一样。

资金交易可能有一个或多个来自Hitesh的输入（加起来5个比特币或更多），以及Irene的一个或多个输入（加起来5个比特币或更多）。投入必须略微超过通道容量才够支付交易费用。该交易有一个将总共10个比特币锁定到由Hitesh和Irene控制的2-of-2多重地址的输出。如果他们的输入超过他们需要贡献的数值，资金交易也可能有一个或多个输出将找零返回给Hitesh和Irene。这是由双方提供和签署的多个输入形成的单一交易。在发送之前，它必须被合作构建起来并且由各方签署。

现在，代替双方签署单一承诺交易的是， Hitesh和Irene创造了两个不对称 的承诺交易。

Hitesh有一个带有两个输出的承诺交易。第一个输出立即支付Irene欠她的5比特币。第二个输出支付Hitesh欠他自己的5比特币，但条件是只有在1000个区块的时间锁之后。交易输出如下所示：

```
Input: 2-of-2 funding output, signed by Irene

Output 0 <5 bitcoin>:
    <Irene's Public Key> CHECKSIG

Output 1:
    <1000 blocks>
    CHECKSEQUENCEVERIFY
    DROP
    <Hitesh's Public Key> CHECKSIG
```

Irene有带有两个输出的不同的承诺交易。 第一个输出支付Hitesh欠他的5比特币。 第二个输出支付Irene，欠她自己的5比特币，但同样只有经过1000个区块的时间锁。 Irene持有的承诺交易（由Hitesh签署）看起来像这样：

```
Input: 2-of-2 funding output, signed by Hitesh

Output 0<5 bitcoin>:
    <Hitesh's Public Key> CHECKSIG

Output 1:
    <1000 blocks>
    CHECKSEQUENCEVERIFY
    DROP
    <Irene's Public Key> CHECKSIG
```

这样一来，双方各有一笔承诺交易，以花费2 -2的资金输出。 该承诺交易的输入是由对方签署的。 在任何时候，持有承诺交易的一方都可以签字（完成2-2签名）并进行广播。 然而，如果他们广播承诺交易，承诺交易会立即支付对方，而他们自己的必须等待短时间锁到期。通过在其中一个输出强制执行赎回拖延,我们可以做到让各方在选择单方面广播承诺交易时处于轻微的不利地位。 但是单靠时间延迟还不足以鼓励公平的行为。

下图12-8显示两个不对称承诺交易，其中承诺持有人的有延迟支付

现在我们介绍这个方案的最后一个要素：一个撤销密钥，允许被欺诈的一方通过占有通道的所有余额来惩罚骗子。

每个承诺交易都有一个“延迟”的输出。 该输出的兑换脚本允许一方在1000个区块后兑换它，或者另一方如果拥有撤销密钥也可兑换它。 所以当Hitesh为Irene签署承诺交易时，他将把第二个输出定义为在1000块之后可输出支付给自己，或者是任何可以出示撤销密钥的人。 Hitesh构建了这个交易，并创建了一个由他秘密保管的撤销密钥。当他准备转移到新的通道状态并希望撤销这一承诺时， 他才会把撤销密钥透露给Irene 第二个输出脚本如下所示：

```
Output 0<5 bitcoin>:
    <Irene's Public Key> CHECKSIG

Output 1<5 bitcoin>:
IF # Revocation penalty output
    <Revocation Public Key>
ELSE
    <1000 blocks>
    CHECKSEQUENCEVERIFY
    DROP
    <Hitesh's Public Key>
ENDIF
CHECKSIG
```

Irene可以自信地签署这笔交易，因为一旦被发送它将立即支付她被欠的欠款。 Hitesh持有交易，但知道如果他在单方通道关闭时发送，他将不得不等待1000个块才能获得支付。

当通道进入下一个状态时，Hitesh必须在Irene同意签署下一个承诺交易之前撤销此承诺交易。要做到这一点，他所要做的就是将撤销密钥发送给Irene。一旦Irene拥有这一承诺的撤销密钥，她就可以自信地签署下一个承诺。她知道，如果Hitesh试图通过发布先前的承诺交易来作弊，她可以使用撤销密钥来兑换Hitesh的延迟输出。如果Hitesh作弊，Irene会得到BOTH（两方）输出。

撤销协议是双边的，这意味着在每一轮中，随着通道状态的进一步发展，双方交换新的承诺，交换用于之前承诺的撤销密钥，并签署彼此的承诺交易。当他们接受新的状态时，他们通过给予对方必要的撤销密钥来惩罚任何作弊行为，使先前的状态不可能再被使用。

我们来看一个它的工作例子。 Irene的客户之一希望向Hitesh的客户发送2比特币。要通过通道传输2比特币，Hitesh和Irene必须更新通道状态以反映新的余额。他们将承诺一个新的状态（状态号2），通道的10个比特币分裂，7个比特币属于Hitesh和3个比特币属于Irene。为了更新通道的状态，他们将各自创建反映新通道余额的新承诺交易。

如上述内容所说，这些承诺交易是不对称的，所以每一方所持的承诺交易都迫使他们等待兑换。至关重要的是，在签署新的承诺交易之前，他们必须首先交换撤销密钥以使先前的承诺无效。在这种情况下，Hitesh的利益与通道的真实状态是一致的，因此他没有理由广播先前的状态。然而，对于Irene来说，状态号1中留给她的余额比状态2中的更高。当Irene给予Hitesh她以前的承诺交易（状态号1）的撤销密钥时，她实际上废除了自己可以回滚通道状态到前一状态而从中获益的能力。因为有了撤销密钥，Hitesh可以毫不拖延地兑换先前承诺交易的两个输出。也就是说一旦Irene广播先前的状态，Hitesh可以行使其占有所有输出的权利。

重要的是，撤销不会自动发生。虽然Hitesh有能力惩罚Irene的作弊行为，但他必须勤勉地观察区块链中作弊的迹象。如果他看到先前的承诺交易广播，他有1000个区块时间采取行动，并使用撤销密钥来阻止Irene的欺骗行为并占有所有余额也就是全部10比特币来惩罚她。

带有相对时间锁（CSV）的不对称可撤销承诺是实现支付通道的更好方法，也是区块链技术非常重要的创新。通过这种结构，通道可以无限期地保持开放，并且可以拥有数十亿的中间承诺交易。在闪电网络的原型实现中，承诺状态由48位索引识别，允许在任何单个通道中有超过281兆（2.8×10^14）个状态转换！

12.6.5哈希时间锁合约（Hash Time Lock Contracts，HTLC）

支付通道可以通过特殊类型的智能合同进一步扩展，以允许参与者将资金用于可赎回的具有到期时间的秘密（secret）。 此功能称为哈希时间锁定合约或HTLC，并用于双向和路由的支付通道。

首先我们来解释HTLC的"哈希"部分。 要创建一个HTLC，预期的收款人将首先创建一个秘密（secret）R。他们然后计算这个R的哈希H：

```
H = Hash\(R\)
```

这步产生可以包含在输出的锁定脚本中的哈希H。 知道秘密的任何人可以用它来兑换输出。 秘密R也被称为哈希函数的*前图像*。 前图像就是用作哈希函数输入的数据。

HTLC的第二部分是"时间锁"组件。 如果秘密没有被透露，HTLC的付款人可以在一段时间后得到"退款"。 这是通过使用绝对时间锁CHECKLOCKTIMEVERIFY来实现的。实现HTLC的脚本可能如下所示：

```
IF
   # Payment if you have the secret R
   HASH160 <H> EQUALVERIFY
ELSE
  # Refund after timeout.
  <locktime>
  CHECKLOCKTIMEVERIFY DROP
  <Payee Pubic Key> CHECKSIG
ENDIF
```

任何知道可以让哈希等于H的对应秘密R的人，可以通过行使IF语句的第一个子句来兑换该输出。

如果秘密没有被透露，HTLC中写明了，在一定数量的块之后，收款人可以使用IF语句中的第二个子句申请退款。

这是HTLC的基本实现。 任何 拥有秘密R的人都可以兑换这种类型的HTLC。通过对脚本进行微调，HTLC可以采用许多不同的形式。 例如，在第一个子句中添加一个CHECKSIG运算符和一个公钥来限制将哈希值兑换成一个指定的收件人，这个人必须知道秘密R.

12.7可路由的支付通道（闪电网络）

闪电网络是一种端到端连接的双向支付通道的可路由网络。这样的网络可以允许任何参与者穿过一个通道路由到另一个通道进行支付，而不需要信任任何中间人。闪电网络由Joseph Poon和Thadeus Dryja于2015年2月首次描述，其基础是许多其他人提出和阐述的支付通道概念。

"闪电网络"是指路由支付通道网络的具体设计，现已由至少五个不同的开源团队实施。这些的独立实施是由"闪电技术基础"（BOLT）论文中描述的一组互通性标准进行协作。

闪电网络的原型实施已经由几个团队发布。现在，这些实现只能在testnet上运行，因为它们使用segwit，还没有在比特币区块主链（mainnet）上激活。

闪电网络是实现可路由支付通道的一种可能方式。还有其他几种旨在实现类似目标的设计，如Teechan和Tumblebit。

12.7.1闪电网络示例

让我们看看它是如何工作的。

在这个例子中，我们有五个参与者：Alice, Bob, Carol, Diana, and Eric。这五名参与者已经彼此之间开设了支付通道。Alice和Bob有支付通道。Bob连接Carol，Carol连接到Diana，Diana连接Eric。为了简单起见，我们假设每个通道每个参与者都注资2个比特币资金，每个通道的总容量为4个比特币。

下图12-9显示一系列通过双向支付的通道连接在一起形成闪电网络以支持一笔从Alice到Eric的付款 展示了闪电网络中五名参与者，通过双向支付通道连接，可从Alice付款到Eric（路由支付通道（闪电网络））。

](h

Alice想要支付给Eric1个比特币。 不过，Alice并未通过支付通道连接到Eric。 创建支付通道需要资金交易，而这笔交易必须首先提交给比特币区块链。 Alice不想打开一个新的支付通道并支出更多的手续费。 有没有办法间接支付Eric？

下图12-10 显示了通过在连接各方参与者的支付通道上通过一系列HTLC承诺将付款从Alice路由到Eric的逐步过程。

Alice正在运行闪电网络（LN）节点，该节点正在跟踪其向Bob的付费通道，并且能够发现支付通道之间的路由。 Alice的LN节点还具有通过互联网连接到Eric的LN节点的能力。 Eric的LN节点使用随机数生成器创建一个秘密R。 Eric的节点没有向任何人泄漏这个秘密。相反，Eric的节点计算秘密R对应的哈希H，并将此哈希发送到Alice的节点（请参阅图12-10步骤1）。

现在Alice的LN节点构建了Alice的LN节点和Eric的LN节点之间的路由。所使用的路由算法将在后面进行更详细的解释，但现在我们假设Alice节点可以找到一个高效的路由。

然后，Alice的节点构造一个HTLC，支付到哈希H，具有10个区块时间的退款超时（当前块+10），数量为1.003比特币（参见图12-10的步骤2）。额外的0.003将用于补偿参与此支付路由的中间节点。Alice将此HTLC提供给Bob，从和Bob之间的通道余额中扣除1.003比特币，并将其提交给HTLC。 该HTLC具有以下含义：“如果Bob知道秘密，Alice将其通道余额的1.003支付给Bob，或者如果超过10个区块时间后，则退还入Alice的余额”。 Alice和Bob之间的通道余额现在由承诺交易表示，其中有三个输出：Bob的2比特币余额，Alice的0.997比特币余额，Alice的HTLC中承诺的1.003比特币。承诺在HTLC中的金额从Alice的余额中被减去。

Bob现在有一个承诺，如果他能够在接下来的10个区块生产时间内获得秘密R，他可以获取Alice锁定的1.003。手上有了这一承诺，Bob的节点在和Carol的支付通道上构建了一个HTLC。Bob的HTLC提交1.002比特币到哈希H共9个区块时间，这个HTLC中如果Carol有秘密R她可以兑换（参见图12-10步骤3）。Bob知道，如果Carol要获取他的HTLC，她必须出示秘密R。如果Bob在9个区块的时间内有R，他可以用它来获取Alice的HTLC给自己。通过承诺自己的通道余额9个区块的时间，他也赚了0.001比特币。如果Carol无法获取他的HTLC，并且他也无法获取Alice的HTLC，那么一切都将恢复到以前的通道余额，没有人会亏损。 Bob和Carol之间的通道余额现在是：2比特币给Carol，0.998给Bob，1.002由Bob承诺给HTLC。

Carol现在有一个承诺，如果她在接下来的9个区块时间内获得R，她可以获取Bob的锁定1.002比特币。现在她可以在她与Diana的通道上构建HTLC承诺。她提交了一个1.001比特币的HTLC到哈希H，共计8个区块时间，如果Diana有秘密R ，她就可以兑换（参见图12-10步骤4）。从Carol的角度来看，如果能够实现，她就可以获得的0.001比特币，否则也没有失去任何东西。她提交给Diana的HTLC，只有在R被泄漏的情况下才可行，到那时候她可以从Bob那里索取HTLC。Carol和Diana之间的通道余额现在是：2给Diana，0.999给Carol，1.001由Carol承诺给HTLC。

最后，Diana可以提供给Eric一个HTLC，承诺1比特币，7个区块时间，到哈希H（参见图12-10的步骤5）。Diana与Eric之间的通道余额现在是：2给Eric，1给Diana，1由Diana承诺给HTLC。

然而，在这条路上，Eric拥有秘密R，他可以获取Diana提供的HTLC。他将R发送给Diana，并获取1个比特币，添加到他的通道余额中（参见图12-10的步骤6）。通道平衡现在是：1给Diana，3给Eric。

现在，Diana有秘密R，因此，她现在可以获取来自Carol的HTLC。Diana将R发送给Carol，并将1.001比特币添加到其通道余额中（参见图12-10的步骤7。现在Carol与Diana之间的通道余额是：0.999给Carol，3.001给Diana。Diana已经“赚了”参与这个付款路线0.001比特币。

通过路由回传，秘密R允许每个参与者获取未完成的HTLC。Carol从Bob那里获取1.002个比特币，将他们通道余额设为：0.998给Bob，3.002给Carol（参见闪电网络步骤8）。最后，Bob获取来自Alice的HTLC（参见闪电网络步骤9）。他们的通道余额更新为：0.997给Alice，3.003给Bob。

在没有向Eric打开通道的情况下，Alice已经支付了Eric 1比特币。付款路线中的中间方不必要互相信任。在他们的通道内做一个短时间的资金承诺，他们可以赚取一小笔费用，唯一的风险是，如果通道关闭或路由付款失败，退款有段短短的延迟时间。

12.7.2闪电网络传输和路由

LN节点之间的所有通信都是点对点加密的。 另外，节点有一个长期公钥，它们用作标识符并且彼此认证对方。

每当节点希望向另一个节点发送支付时，它必须首先通过连接具有足够容量的支付通道来构建通过网络的路径。节点宣传路由信息，包括他们已经打开了什么通道，每个通道拥有多少容量，以及他们收取多少路由支付费用。路由信息可以以各种方式共享，并且随着闪电网络技术的进步，不同的路由协议可能会出现。一些闪电网络实施使用IRC协议作为节点宣布路由信息的一种方便的机制。路由发现的另一种实现方式是使用P2P模型，其中节点将通道宣传传播给他们的对等体，在“洪水泛滥”模型中，这类似于比特币传播交易的方法。未来的计划包括一个名为Flare的建议，它是一种具有本地节点“邻居”和较长距离的信标节点的混合路由模型。

在我们前面的例子中，Alice的节点使用这些路由发现机制之一来查找将她的节点连接到Eric的节点的一个或多个路径。一旦Alice的节点构建了路径，她将通过网络初始化该路径，传播一系列加密和嵌套的指令来连接每个相邻的支付通道。

重要的是，这个路径只有Alice的节点才知道。付款路线上的所有其他参与者只能看到相邻的节点。从Carol的角度来看，这看起来像是从Bob到Diana的付款。 Carol不知道Bob实际上是中继转发Alice的汇款。她也不知道Diana将会向Eric中继转发付款。

这是闪电网络的一个重要特征，因为它确保了付款的隐私，并且使得很难应用监视，审查或黑名单。但是，Alice如何建立这种付款途径，而不向中间节点透露任何内容？

闪电网络实现了一种基于称为Sphinx的方案的洋葱路由协议。该路由协议确保支付发送者可以通过闪电网络构建和通信路径，使得：

- 中间节点可以验证和解密其部分路由信息，并找到下一跳。
- 除了上一跳和下一跳，他们不能了解作为路径一部分的任何其他节点。
- 他们无法识别支付路径的长度，或者他们自己在该路径中的位置。
- 路径的每个部分被加密，使得网络级攻击者不能将来自路径的不同部分的数据包彼此关联。
- 不同于Tor（互联网上的洋葱路由匿名协议），没有可以被监视的“退出节点”。付款不需要传输到比特币区块链，节点只是更新通道余额。

使用这种洋葱路由协议，Alice将路径的每个元素包裹在一层加密中，从尾端开始倒过来运算。她用Eric的公钥加密了Eric的消息。该消息包裹在加密到Diana的消息中，将Eric标识为下一个收件人。给Diana的消息包裹在加密到Carol的公钥的消息中，并将Diana识别为下一个收件人。对Carol的消息被Bob的密钥加密。这样一来，Alice已经构建了这个加密的多层“洋葱”的消息。她发送给Bob，他只能解密和解开外层。在里面，Bob发现一封给Carol的消息，他可以转发给Carol，但不能自己破译。按照路径，消息被转发，解密，转发等，一路到Eric那里。每个参与者只知道各自这一跳的前一个和下一个节点。

路径的每个元素包含承载于HTLC的必须扩展到下一跳的信息，HTLC中的要发送的数量，要包括的费用以及CLTV锁定到期时间（以块为单位）。随着路由信息的传播，节点将HTLC承诺转发到下一跳。

在这一点上，您可能会想知道节点怎么知道路径的长度及其在该路径中的位置。毕竟，他们收到一个消息，并将其转发到下一跳。难道它不会将路径缩短，或者允许他们推断出路径大小和位置？为了防止这种情况，路径总是固定在20跳，并用随机数据填充。每个节点都会看到下一跳和一个要转发的固定长度的加密消息。只有最终的收件人看得到没有下一跳。对于其他人来说，似乎总是有20多跳要走。

12.7.3闪电网络优势

闪电网络是第二层路由技术。它可以应用于支持一些基本功能的任何区块链，如多重签名交易，时间锁定和基本的智能合约。

如果闪电网络搭建在在比特币网络之上，则比特币网络可以大大提高容量，隐私性，粒度和速度，而不会牺牲无中介机构的无信任操作原则：

隐私 闪电网络付款比比特币区块链的付款更私密，因为它们不是公开的。虽然路由中的参与者可以看到在其通道上传播的付款，但他们并不知道发件人或收件人。

流动性 闪电网络使得在比特币上应用监视和黑名单变得更加困难，从而增加了货币的流动性。

速度 使用闪电网络的比特币交易将以毫秒为单位，而不是分钟，因为HTLC在不用提交交易到区块上的情况下被结算。

粒度 闪电网络可以使支付至少与比特币“灰尘”限制一样小，甚至更小。一些建议允许子聪级增量（subsatoshi increments）。

容量 闪电网络将比特币系统的容量提高了几个数量级。每秒可以通过闪电网络路由的付费数量没有具体上限，因为它仅取决于每个节点的容量和速度。

无信任操作闪电网络在不需要互相信任就可以作为对等体使用的节点之间使用比特币交易。因此，闪电网络保留了比特币系统的原理，同时显著扩大了其操作参数。

当然，如前所述，闪电网络协议不是实现路由支付通道的唯一方法。其他被提出的系统包括Tumblebit和Teechan。然而，在这个时候，闪电网络已经部署在testnet上了。几个不同的团队已经开发了正在竞争的LN实现，并且正在努力实现一个通用的互操作性标准（称为BOLT）。闪电网络很可能是第一个部署在生产实际中的路由支付通道网络。

12.8结论

我们仅研究了几个可以使用比特币区块链作为信任平台构建的新兴应用程序。这些应用程序将比特币的范围扩大到超出付款和超越金融工具的范围，以涵盖许多信任至关重要的其他应用程序。通过去中性化的信任基础，比特币区块链是一个会释放将在各种行业中产生许多革命性应用的平台。