Bitcoin Scalability

-- Cryptocurrencies for the Masses

LING Zong, Ph. D.
Senior Software Engineer / Scientist
IBM Almaden Research Center
San Jose, California, U.S.A.

Lecture Outline

- Scalability Problem for Bitcoin
- Proposed Scalability Solutions
  - Blocksize Capacity Increase
  - Segregated Witness
  - Sidechains
  - Lightning Network

Scalability Problem for Bitcoin

Bitcoin Scalability Problem

比特币可扩展性问题是指对比特币网络能够处理的交易数量的限制的讨论。这与比特币区块链中的记录(称为块)在大小和频率上都是有限的有关

The bitcoin scalability problem refers to the discussion concerning the limits on the amount of transactions the bitcoin network can process. It is related to the fact that records (known as blocks) in the bitcoin blockchain are limited in **size and frequency**.

比特币的区块包含了比特币网络上的交易

- Bitcoin's blocks contain the transactions on the bitcoin network.
- The on chain transaction processing capacity of the bitcoin network is limited by the **average block creation time of 10 minutes** and the block size limit.

比特币网络的链式事务处理能力受平均区块创建时间为10分钟和区块大小限制

这些共同限制了网络的吞吐量。事务处理能力的最大值估计在每秒3.3到7个事务之间。
针对这一问题，有各种已提出和已启动的解决方案

These jointly constrain the network's throughput. The transaction processing capacity maximum is estimated **between 3.3 and 7 transactions per second**.

There are various proposed and activated solutions to address this issue.

Scalability

在此上下文中，指系统处理增加的事务量的能力

Scalability: In this context, the ability of a system to deal with increased transaction volume.

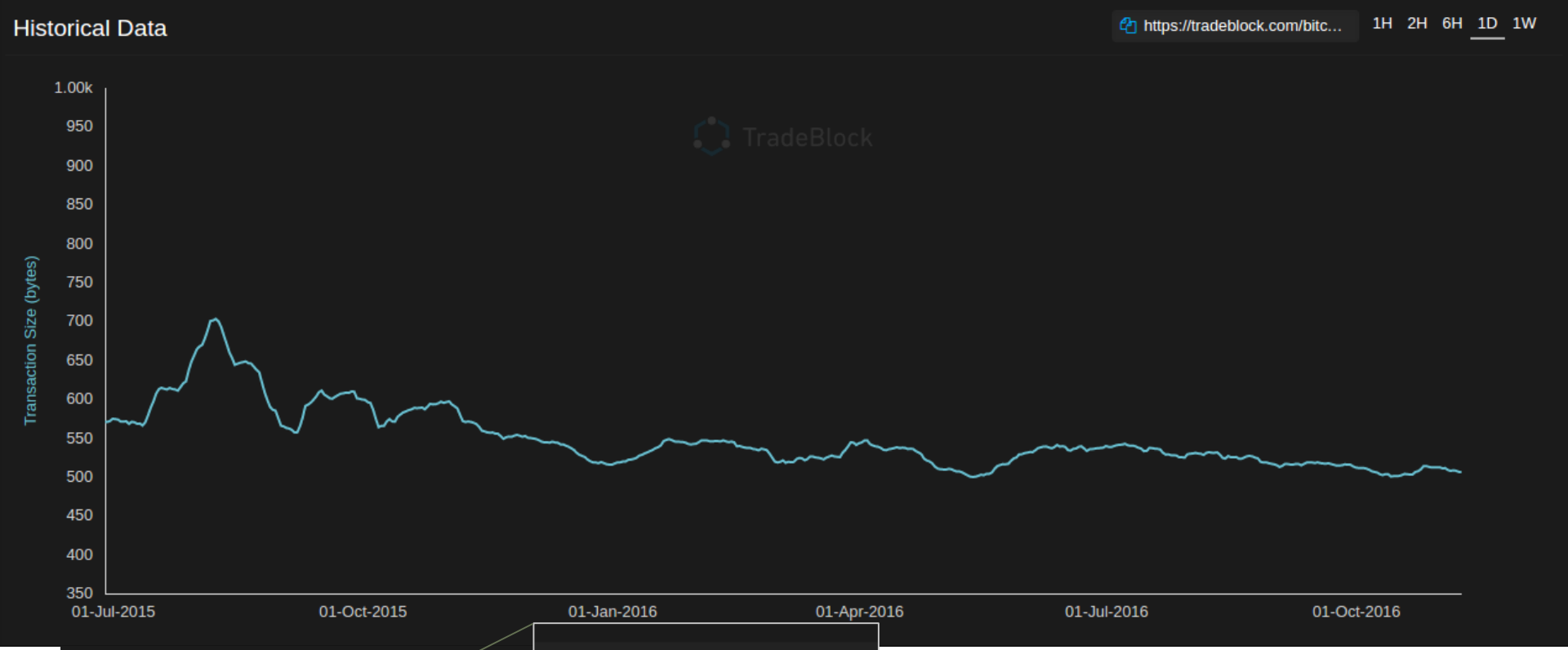
We typically use the unit **tps**, or ***transactions per second***, to describe the scalability of a financial system such as Bitcoin.

我们通常使用单位tps或每秒交易来描述金融系统(如比特币)的可伸缩性。
在考虑可伸缩性时，除了考虑速度，还应该考虑大小。
区块链比特币目前已经有80gb。
我们能让节点有效地将其存储到未来吗?

When thinking about scalability, should also think about size along with velocity.

Bitcoin Blockchain is currently already 80 Gb.

Can we make it so that nodes can store it efficiently into the future?



Statistics

	Transaction Size
Observations	500
Mean	546.38
Median	530.94
Mode	391.77
Std. Dev.	69.58

500

546.38

530.94

https://tradeblock.com/bitcoin/historical/1d-f-tsize_per_avg-00271

2020/10/10

Bitcoin’s Current Scalability

From previous slide:

- Average of 546 bytes per transaction.
- Current blocksize is 1 MiB.
- Expected time to next block is 10 min.

从前一个的幻灯片：
●每个事务的平均546字节。
●当前块大小为1mib。
●到下一个区块的预期时间是10分钟。
因此我们可以计算tps中持续的最大交易量：

Therefore we can compute the sustained maximum transaction volume in tps:

$$\frac{1\ MiB}{1\ block} \times \frac{1\ txn}{546\ bytes} \times \frac{1\ block}{10\ min} \approx 3.2\ tps$$

Scalability Comparison

How does Bitcoin compare with other traditional payment systems?

	Average	High Load / Maximum
Bitcoin	2.2 tps	3.2 tps
PayPal*	150 tps	450 tps
VISA**	2,000 tps	56,000 tps

* http://www.fool.com/investing/general/2016/02/04/5-things-paypal-holdings-inc-wants-you-to-know.aspx
** https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf

Bitcoin's Current Scalability

From previous slide:

- Average of 546 bytes per transaction.
- Current blocksize is 1 MiB.
- Expected time to next block is 10 min.

Therefore we can compute the sustained maximum transaction volume in tps:

$$\frac{1\ MiB}{1\ block} \times \frac{1\ txn}{546\ bytes} \times \frac{1\ block}{10\ min} \approx 3.2\ tps$$

What are the variables we can play with?

- **Size of blocks**
- **Size of transactions**
- **Block creation rate**
- **Whether transactions get added**

Naive Solution

只是增加方块的速度。简单的解决方案。只要减少现时
Just increase the speed of blocks. Easiest solution. Just decrease the nonce.

Cons:

- Less propagation time
- More Block Orphaning
  - More people find solutions at the same time
- Mess up halving
- More Frequent Forks
  - Therefore potential for double spends
- Less likely to be accepted than increasing blocksize
  - 比增加块大小更不容易被接受
  - Seen as a fundamental part of the protocol, rather than just a protective soft limit
    被视为协议的基本组成部分，而不仅仅是一个保护性的软限制

确认的时间更少，但不是真的，因为你现在需要更多的确认
Less time for confirmations but like not really, cause you now need more confirmations

Naive Solution

恰恰提出了这一点，却被完全忽视了

Here is Vitalik Buterin, creator of Ethereum, proposing exactly that and being completely ignored.

Soft-forking the block time to 2 min: my primarily silly and academic (but seemingly effective) entry to the "increase the blockchain's capacity in an arbitrarily roundabout way as long as it's a softfork" competition (self.btc)

submitted 9 months ago * (last edited 9 months ago) by vbuterin

So given that large portions of the bitcoin community seem to be strongly attached to this notion that hard forks are an unforgivable evil, to the point that schemes containing hundreds of lines of code are deemed to be a preferred alternative, I thought that I'd offer an alternative strategy to increasing the bitcoin blockchain's throughput with nothing more than a soft fork - one which is somewhat involved and counterintuitive, but for which the code changes are actually quite a bit smaller than some of the alternatives; particularly, "upper layers" of the protocol stack should need no changes at all.

Notes:

- Unlike the "generalized softfork" approach of putting the "real" merkle root in the coinbase of otherwise mandatorily empty blocks, this strategy makes very little change to the semantics of the protocol. No changes to block explorers or wallets required.
- The point of this is largely academic, to show what is possible in a blockchain protocol. That said, if some segwit-as-block-size-increase supporters are interested in segwit because it increases the cap in a way that does not introduce a slippery slope, block time decreases are a viable alternative strategy, as there is a limit to how low block time can go while preserving safety and so the slippery slope has a hard stop and does not extend infinitely.
- My personal *actual* preference would be a simple s/1000000/2000000/g (plus a cap of 100-1000kb/tx to address ddos issues), though I also believe that people on all sides here are far too quick to believe that the other side is evil and not see that there are plenty of reasonable arguments in every camp. I recommend this, this and this as required reading.
- There's some chance that some obscure rule of the bitcoin protocol makes this all invalid, but then I don't know about it and did not see it in the code.

The attack vector is as follows. Instead of trying to increase the size of an individual block directly, we will create a softfork where **under the softfork rules, miners are compelled to insert incorrect timestamps, so as to trick the bitcoin blockchain into retargeting difficulty in such a way that on average, a block comes every two minutes instead of once every ten minutes**, thereby increasing throughput to be equivalent to a 5 MB block size.

First, let us go over the bitcoin block timestamp and difficulty retargeting rules:

- Every block must include a timestamp.
- This timestamp must at the least be greater than the median of the previous eleven blocks (code here and here)
- For a node to accept a block, this timestamp must be at most 2 hours ahead of the node's "network-adjusted time" (code here), which can itself be at most 70 minutes ahead of the node's timestamp (code here); hence, we can never go more than 3.17 hours into the future
- Every 2016 blocks, there is a difficulty retargeting event. At that point, we calculate D = the difference between the latest block time and the block time of the block 2016 blocks before. Then, we "clamp" D to be between 302400 and 4834800 seconds (1209600 seconds = 2 weeks is the value that D "should be" if difficulty is correctly calibrated). We finally adjust difficulty by a factor of 1/D: for example, if D = 604800, difficulty goes up by 2x, if D = 1814400, difficulty goes down by 33%, etc. (code here)

The last rule ensures that difficulty adjustments are "clamped" between a 4x increase and a 4x decrease no matter what.

So, how to we do this? Let's suppose for the sake of simplicity that in all examples the soft fork starts at unix time 1500000000. We could say that instead of putting the real time into blocks, miners should put `1500000000 + (t - 1500000000) * 5`; this would make the blockchain think that blocks are coming 5x as rarely, and so it would decrease difficulty by a factor of 5, so that from the point of view of actual time blocks will start coming in every two minutes instead of ten. However, this approach has one problem: it is not a soft fork. Users running the original bitcoin client will very quickly start rejecting the new blocks because the timestamps are too far into the future.

Naive Solution

- But wait, Ethereum has 17 second block times.
- Less time for confirmations but like not really, cause you now need more confirmations 确认的时间更少，但不是真的，因为你现在需要更多的确认

- Pros and Cons
  - https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/
  - Or, read： **On Slow and Fast Block Times.DOC**

Proposed Scalability Solutions

Blocksize Capacity Increase

Segregated Witness

Sidechains

Lightning Network

Block Capacity Increase

Increase the Block Size

The block size limit has created a **bottleneck** in bitcoin, resulting in increasing transaction fees and delayed processing of transactions that cannot be fit into a block.

区块大小的限制给比特币带来了瓶颈，导致交易费用增加，无法放入一个区块的交易被延迟处理。
关于如何扩大比特币规模，出现了各种各样的提议，并引发了一场激烈的辩论。2017年，“Business Insider”将这场辩论描述为“围绕比特币未来的意识形态之争”。

Various proposals have come forth on how to scale bitcoin, and a contentious debate has resulted. *Business Insider* in 2017 characterized this debate as an "ideological battle over bitcoin's future."

> “
> It can be phased in, like:
> if (blocknumber > 115000)
> maxblocksize = largerlimit
>
> -- Satoshi Nakamoto

Increase the Block Size

如果我们只是增加块大小，我们可以在一个块中容纳更多的事务

Idea: If we just increase the blocksize, we can fit more transactions in a single block.

块越大，传播和验证的时间就越长。其他挖掘器只有在验证了当前块之后才能开始对一个(非空的)后续块进行工作。因此，更大的块会使创作挖掘程序在查找下一个块时获得更大的优势。

Larger blocks take longer to propagate and longer to validate. Other miners can only start working on a (non-empty) succeeding block once they've validated the current. Larger blocks therefore lead to a greater advantage for the authoring miner at finding also the next block.

Pros:

· 这是一个“简单”的实现。只要让矿工们同意就行了(显然我们知道这并不容易)。
· 在闪电网络无论如何可能不得不做
· 降低交易费用(如果你是用户的话)

- It’s an “easy” implementation. Just get miners to agree (Obviously we know this isn’t easy).
- Might have to do it for lightning network anyways
- Lessen transaction fees (if you’re a user)

Cons:

- Hard fork blah blah blah
- Lessen transaction fees (if you’re a miner)
- Size increases very fast
  - “Slippery slope”
  - Blockchain is already 80 Gb!
- Longer Propagation Times
  - Authoring miner has better shot at next block
- One time linear capacity increase
  - Temporary Fix

●难以fork等等
●降低交易费用(如果你是矿工)
●尺寸增长非常快
〇“滑坡”
〇区块链已经是80gb了!
●传播时间更长
〇创作矿工有更好的机会在下一个块
●一次线性容量增加
〇临时修复

Examples

- Bitcoin Classic
  - 2 Mb
- Bitcoin XT
  - 8 Mb
- Bitcoin Unlimited
  - Allows miner to accept a block larger than your maximum accepted blocksize if it reaches a certain number of blocks deep in the chain.
  - Essentially will make blocksize decided by “supply and demand” through a transaction fee market
    - Removes the only point of central control in the Bitcoin economy
    - Free-market economics will force consensus
    - Will adapt in real time to real-world conditions
  - Update on ViaBTC (https://www.viabtc.com/)
    - They were trying to stall SegWitness in support of Bitcoin Unlimited
    - They lost half their hashing power
    - SegWitness passed

〇允许矿工接受一个区块大于你的最大可接受的区块大小，如果它在链中达到一定数量的区块。
〇实质上将通过交易费市场使区块大小由“供求”决定
■消除了比特币经济中唯一的中央控制点
■自由市场经济将迫使各方达成共识
■将实时适应现实环境
〇ViaBTC更新(https://www.viabtc.com/)
■他们试图阻止SegWitness支持比特币无限
■它们失去了一半的哈希能力
■SegWitness通过了

Segregated Witness

隔离验证

Segregated Witness

每个事务的数字签名在每个块中占用大量空间。他们没有必要在那里。让我们删除它们

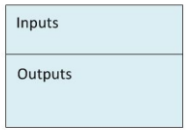
Idea: The **digital signatures** for each transaction take up a lot of space in each block. There's no reason they need to be there. Let's remove them.

How:

Segwit P2W*

For Old Nodes:

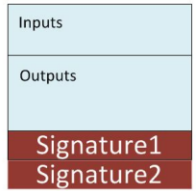
ScriptPubKey: 0 e4873ef43eac347471dd94bc899c51b395a509a5
ScriptSig: Empty
Result: **Valid**



Segwit P2W*

For New Nodes:

ScriptPubKey: 0 e4873ef43eac347471dd94bc899c51b395a509a5
ScriptSig: Empty
WitScript: Signature1
Result: **Valid**



Segregated Witness

隔离见证(缩写为SegWit)是一个已实现的协议升级，旨在提供保护，防止事务延展性和增加块容量。
· SegWit将证人从输入列表中分离出来。见证包含检查事务有效性所需的数据，但不需要确定事务效果。
· 此外，还定义了一个新的重量参数，块最多允许有400万个重量单位(WU)。非证人和前分段证人字节重4 WU，但分段证人数据的每个字节仅重1 WU，允许块大于1 MB，而没有硬叉更改

Segregated Witness (abbreviated as **SegWit**) is an implemented protocol upgrade intended to provide protection from transaction malleability and increase block capacity.

- SegWit separates the *witness* from the list of inputs. The witness contains data required to check transaction validity but is not required to determine transaction effects.
- Additionally, a new *weight* parameter is defined, and blocks are allowed to have at most 4 million weight units (WU). Non-witness and pre-segwit witness bytes weigh 4 WU, but each byte of Segwit witness data only weighs 1 WU, allowing blocks that are larger than 1 MB without a hardforking change.

但是现在，区块链没有任何证据表明交易中包含了正确的签名。
●SegWit miner从分离的见证人创建Merkle树，镜像事务树
●该树的默克尔根包含在coinbase交易的输入字段中。
●这会更改coinbase事务的事务ID
●因此，签名影响块标头，并最终影响区块链的组成

But now, the blockchain doesn't have any evidence that correct signatures were included in transactions.

- A SegWit miner creates a Merkle tree out of segregated witnesses that mirrors the transaction tree
- This tree's merkel root is included in the input field of the coinbase transaction.
- This changes the transaction ID of the coinbase transaction
- Therefore signatures influence the block header and, ultimately, the makeup of the blockchain.

在成功激活OP_CLTV和OP_CSV之后，SegWit是使闪电网络安全部署到比特币网络上所需要的最后一个协议更改。由于新的witness字段包含脚本版本控制，还可以对SegWit脚本进行更改或引入新的操作码，这些脚本原本需要额外的复杂性才能在不使用SegWit的情况下运行

After the successful activations of OP_CLTV and OP_CSV, SegWit was the last protocol change needed to make the Lightning Network safe to deploy on the Bitcoin network. Because the new witness field contains Script versioning, it is also possible to make changes to or introduce new opcodes to SegWit scripts that would have originally required additional complexity to function without SegWit.

---- https://en.bitcoin.it/wiki/Segregated_Witness

Segregated Witness

历史和激活：在2016年和2017年，分离证人的激活被矿工利用BIP 9激活机制中的一个缺陷，出于政治原因而被阻止。在技术层面上，比特币的共识规则是由经济上的多数人而不是矿商控制的，因此，通过创建一个用户激活的软叉BIP 148，就有可能解决僵局。在这个系统中，经济上的多数人将绕过封锁的矿商，自己激活隔离的证人。这需要经济上的大多数人进行协调，但最终成功了，在2017年8月1日之后不久就激活了比特币Segwit

History and Activation: During 2016 and 2017 activation of segregated witness was blocked by miners for political reasons by exploiting a flaw in the BIP 9 activation mechanism. On a technical level, the consensus rules of bitcoin are controlled by the economic majority not the miners, so the deadlock was possible to solve by creating a user activated soft fork BIP 148 where the economic majority would bypass the blocking miners and activate segregated witness on their own. This required some coordination amongst the economic majority, but was ultimately successful, activating Segwit on Bitcoin soon after 1st August 2017.

●只有软叉
●修复事务延展性
●允许闪电网络和sidechains工作
●没有滑坡
●效率提升
●小区块链的大小

Pros:

- Only soft fork
- Fixes Transaction Malleability
  - Allows Lightning Network and sidechains to work
- No slippery slope
- Efficiency Gains
- Smaller Size of Blockchain

●一次线性容量增加
●引入新型DOS攻击(go-fish- witt -ddos)
●非常复杂和丑陋(超过500行代码)
●解决问题的其他方法
●钱包必须包含它

Cons:

- One time linear capacity increase
- Introduces new type of DOS attack (go-fish-wit-ddos)
- Very complicated and ugly (Over 500 lines of code)
- Other ways to solve maleability
- Wallets have to incorporate it

Introduces a new type of DOS attack (go-fish-wit-ddos)

攻击者利用网络尚未发现的1000个事务挖掘分段块(攻击者自己创建这些TX)，攻击者有现成的目击数据。当其他的矿机尝试验证这个块时，它们将遍历这些TX中的每一个并说“我没有这个TX ID的目击者数据，我必须调用TCP::GetWitnessData(TX ID) aw是的，这是有效的

An attacker mines a segwit-block with 1000 transactions the network has not yet seen (The attacker creates these TX **her**self) The attacker has the witness data readily available. When other miners try to validate this block they will go through every single one of these TX and say "I don't have the witness data for this TX_ID, I have to call TCP::GetWitnessData(TX_ID) aw yes this is valid"

Schnorr Multisignatures

Schnorr签名是一种通过比特币交易所需的签名聚合来提高比特币网络容量的方法

Schnorr signatures are a method of improving the capacity of the Bitcoin network through the aggregation of signatures required for a Bitcoin transaction.

而不是要求每个成员的签名，组合他们且只有一个签名

Idea: Instead of requiring the signatures of every member, combines them and only has one signature

Pros:

- Can be implemented with either soft or hard fork (cleaner with hard fork)
- Multisig transactions will be significantly smaller
- Faster verification
- Plausible deniability for participants

●软叉或硬叉均可执行(硬叉更cleaner)
●Multisig事务将大大减少
●快速验证
●参与者的合理推诿

Why wasn't it implemented?

比特币刚出现的时候，ECDSA是最受欢迎的，因为Schnorrs还在专利保护之下。现在不是了。在各个方面都好多了。只是需要有人来实现它

When bitcoin first came out, ECDSA was the most popular because Schnorrs was still under patent protection. It's not anymore. Pretty much better in every way. Just needs someone to implement it.

https://en.wikipedia.org/wiki/Schnorr_signature

Bitcoin's Current Scalability

What are the variables we can play with?

- Size of blocks
- Size of transactions
- Block creation rate

$$\frac{1\ MiB}{1\ block} \times \frac{1\ txn}{546\ bytes} \times \frac{1\ block}{10\ min} \approx 3.2\ tps$$

Now what?

We need to change something else. Let's just not use the blockchain!

Sidechains

Sidechains

侧链或挂钩侧链可以使比特币和其他账本资产在多个区块链之间转移。这使得用户可以使用他们已经拥有的资产访问新的和创新的加密货币系统。通过重复使用比特币的货币，这些系统可以更容易地与彼此以及与比特币进行互操作，避免新货币带来的流动性短缺和市场波动。由于侧链是独立的系统，技术和经济创新不会受到阻碍。尽管比特币和挂钩侧链之间可以双向转移，但它们是孤立的:在侧链发生密码中断(或恶意设计)的情况下，损害完全局限于侧链本身

A ***sidechain*** or *pegged sidechain* enables bitcoins and other ledger assets to be transferred between multiple blockchains. This gives users access to new and innovative cryptocurrency systems using the assets they already own. By reusing Bitcoin's currency, these systems can more easily interoperate with each other and with Bitcoin, avoiding the liquidity shortages and market fluctuations associated with new currencies. Since sidechains are separate systems, technical and economic innovation is not hindered. Despite bidirectional transferability between Bitcoin and pegged sidechains, they are isolated: in the case of a cryptographic break (or malicious design) in a sidechain, the damage is entirely confined to the sidechain itself.

https://en.bitcoin.it/wiki/Sidechain

Idea:

有了侧链，人们可以将他们的比特币转移到一个更快、更不安全的区块链上，用来购买他们的早餐咖啡。当链只用于较小的、不重要的事务时，sidechain具有更大的块大小限制这一事实就不是什么问题了

With sidechains, one could move their bitcoin over to a faster, less-secure blockchain for purchasing their morning coffee. The fact that a sidechain has a much larger block size limit would be less of an issue when the chain is only being used for small, unimportant transactions.

Pros:

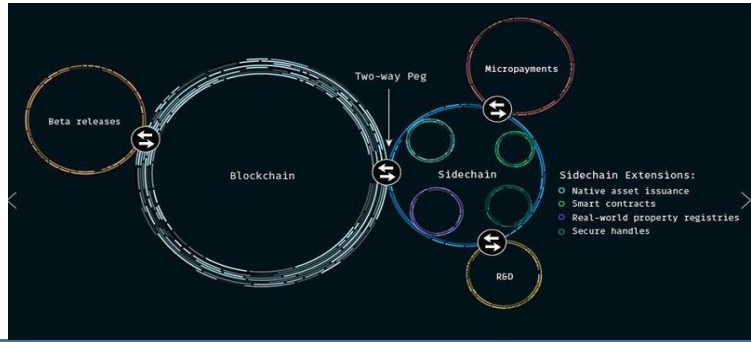
Less things on bitcoin blockchain, but can still be pegged to it.

比特币区块链上的东西少了，但仍然可以盯住它

Cons:

Not “really” a solution. Just moved things to different chains.

不是“真正”的解决方案。只是把东西移到不同的链上



Recall:

Bitcoin Transactions:

当Alice想向Bob支付1btc时，Alice签署一个交易，将其广播到网络，Bob等待一定数量的确认后才认为Alice的支付是有效的。
通过等待在Alice的付款上挖掘区块，Bob可以确保Alice不会重复消费从而欺骗Bob

When Alice wants to pay Bob 1 BTC, Alice signs a transaction, broadcasts it to the network, and Bob waits for some number of confirmations *before* he considers Alice's payment to be valid.

By waiting for blocks to be mined on top of Alice's payment, Bob can ensure that Alice can't double spend and cheat Bob.

Idea:

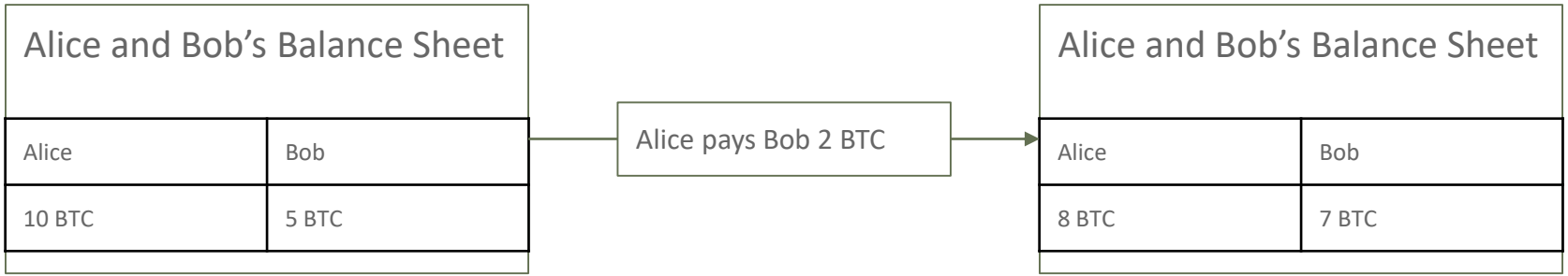
爱丽丝和鲍勃能在不需要经常咨询区块链以防止一方欺骗另一方的情况下彼此支付吗

Can Alice and Bob make payments between themselves without always needing to consult the blockchain to prevent one cheating the other?

Proposal:

如果Alice和Bob维护一个私人资产余额表，更新每笔付款的借据，并且只在一方想结算时才咨询区块链，情况会怎样?

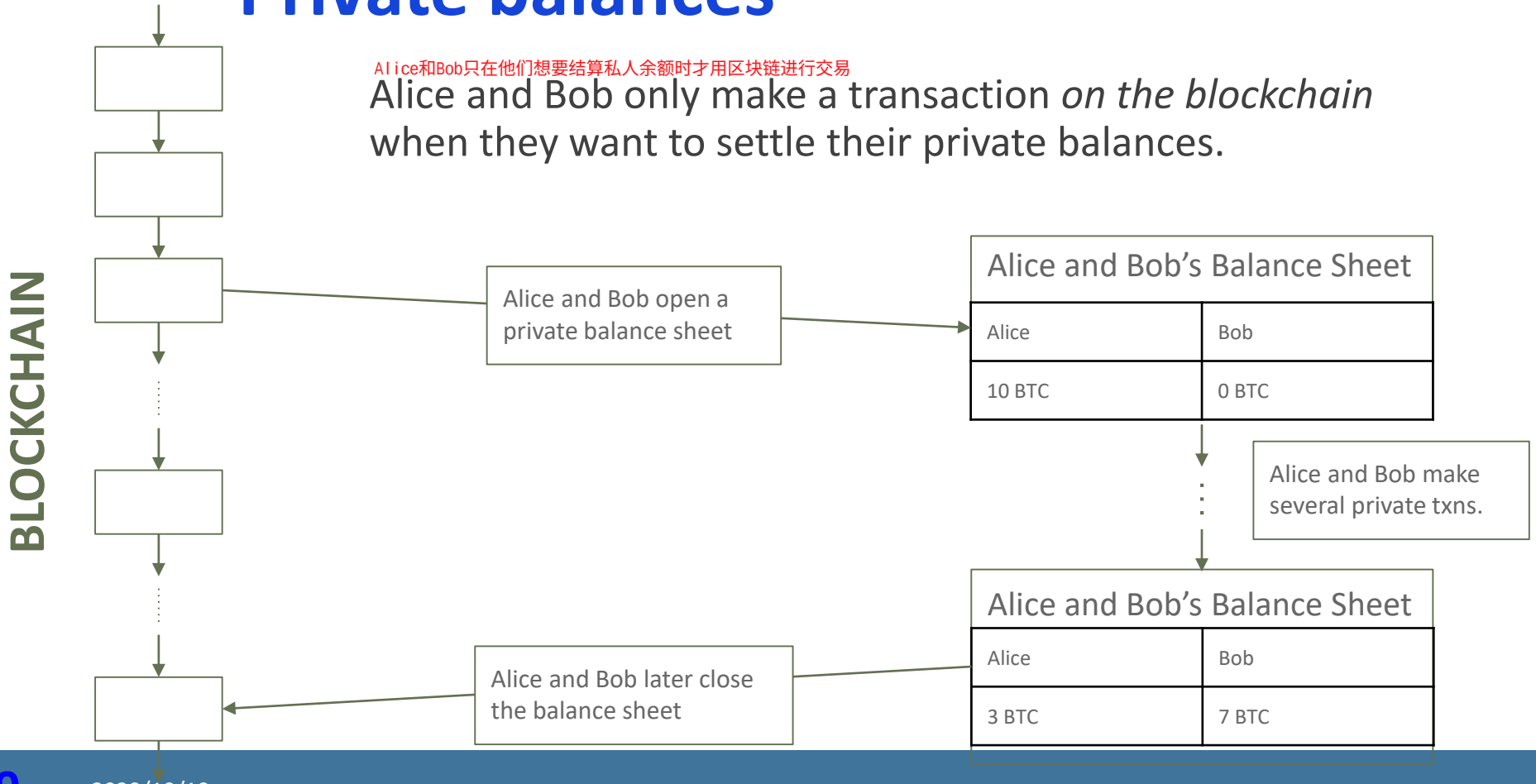
What if Alice and Bob maintain a ***private balance sheet***, updating the IOUs with every payment and only consult the blockchain when one party wants to settle?



Private balances

Alice和Bob只在他们想要结算私人余额时才用区块链进行交易

Alice and Bob only make a transaction *on the blockchain* when they want to settle their private balances.



Payment Channels

微支付渠道或支付渠道是一种技术，旨在允许用户进行多次比特币交易，而无需将所有交易提交给比特币区块链

A **Micropayment Channel** or **Payment Channel** is class of techniques designed to allow users to make multiple Bitcoin transactions without committing all of the transactions to the Bitcoin blockchain.

- In a typical payment channel, only two transactions are added to the block chain but an unlimited or nearly unlimited number of payments can be made between the participants.
- Several channel designs have been proposed or implemented over the years. Many designs are vulnerable to transaction malleability. Specifically, many designs require a way to be able to spend an unsigned transaction, in order to ensure that the channel can be opened atomically. Thus, these designs require a malleability fix that separates the signatures from the part of the transaction that is hashed to form the txid.

· 在一个典型的支付通道中，只有两个交易被添加到区块链中，但是参与者之间可以进行无限或几乎无限的支付。
· 多年来，已经提出或实施了几种通道设计。许多设计易受交易延展性的影响。特别地，许多设计要求能够使用未签名事务的方法，以确保通道可以自动打开。因此，这些设计需要一个延展性修复，将签名与被散列以形成txid的事务部分分离开来。

Idea: Use Bitcoin script to create blockchain-enforceable contracts between Alice and Bob so that neither party can cheat the other, while maintaining the private balance sheet functionality!

使用比特币脚本在Alice和Bob之间创建可执行区块链的合同，这样双方都不能欺骗对方，同时保持私人资产余额表的功能!

In blockchain land, we call these **payment channels**.
How does this *“payment channel”* thing actually work?

Hash Time-locked Contract (HTLC)

术语哈希时间锁合同(HTLC)指的是一种特殊功能，用于创建智能合同，可以修改支付渠道。从技术上讲，HTLC特性支持在两个用户之间实现有时间限制的事务。实际上，HTLC交易的接收者必须在指定的时间框架(块数)内提交一份密码证明来确认付款。如果收件人被没收或未能要求付款，款项将退还给原寄件人。HTLC特性应用于双向和路由支付渠道，允许在各种渠道上安全地转移资金，而不需要信任任何中介。

The term **Hashed TimeLock Contract (HTLC)** refers to a special feature that is used to create smart contracts that are able to modify payment channels. Technically, the HTLC feature enables the implementation of time-bound transactions between two users. In practice, the recipient of a HTLC transaction has to acknowledge the payment by submitting a cryptographic proof within a specified timeframe (number of blocks). If the recipient forfeits or fails to claim the payment, the funds will be returned to the original sender. The HTLC feature is applied in both bidirectional and routed payment channels to allow the secure transfers of funds over various channels, without requiring trust on any of the intermediaries.

HTLC与标准加密货币交易有两个关键的区别，它们是：
Hashlock:限制资金支出的函数，直到某个数据片段被公开(作为加密证明)。这样的证明也可以称为hashlock的前图像。前映像只是用于生成hashlock和稍后解锁其资金的信息片段。
Timelock:是一个函数，限制资金的支出，直到未来特定的时间(或块高度)。例如，通过使用CheckLockTimeVerify或CheckSequenceVerify等函数，可以在比特币中实现

There are two key elements which distinguish HTLC from standard cryptocurrency transactions, which are:

- **Hashlock**: a function that restricts the spending of funds until a certain piece of data is publicly disclosed (as a cryptographic proof). Such proof may also be referred to as the pre-image of the hashlock. The pre-image is simply the piece of information that is used to generate the hashlock, and to later unlock its funds.
- **Timelock**: is a function that restricts the spending of funds until a specific time (or block height) in the future. It can be achieved in Bitcoin, for example, using functions like CheckLockTimeVerify or CheckSequenceVerify.

比特币闪电网络是哈希时间锁定合同最受欢迎的使用案例之一。通过在支付渠道中实施HTLC，资金可以通过互联的支付渠道在用户之间进行交易，而不需要任何级别的信任。这个过程称为网络路由。它允许爱丽丝与卡罗尔交换资金，即使他们没有直接通过支付渠道连接。HTLC使Alice能够通过网络的其他参与者(如Bob)将她的资金发送给Carol——hashlock和timelock特性确保了Bob不会拦截资金。除了在Lightning网络上使用外，HTLCs还可以用于其他上下文，如跨链原子交换、金融智能合同和第三方托管等等

The Bitcoin Lightning Network is among the most popular use cases of Hashed Timelocked Contracts. By implementing HTLC into payment channels, funds can be transacted from user to user through interconnected payment channels, without requiring any level of trust. This process is known as network routing. It allows Alice to exchange funds with Carol even if they are not directly connected through a payment channel. HTLC's enable Alice to send her funds to Carol through other participants of the network (e.g., Bob) - and the hashlock and timelock features ensure that Bob cannot intercept the funds. Besides being used on the Lightning Network, HTLCs can also be useful in other contexts, such as cross-chain atomic swaps, financial smart contracts and escrow, and much more.

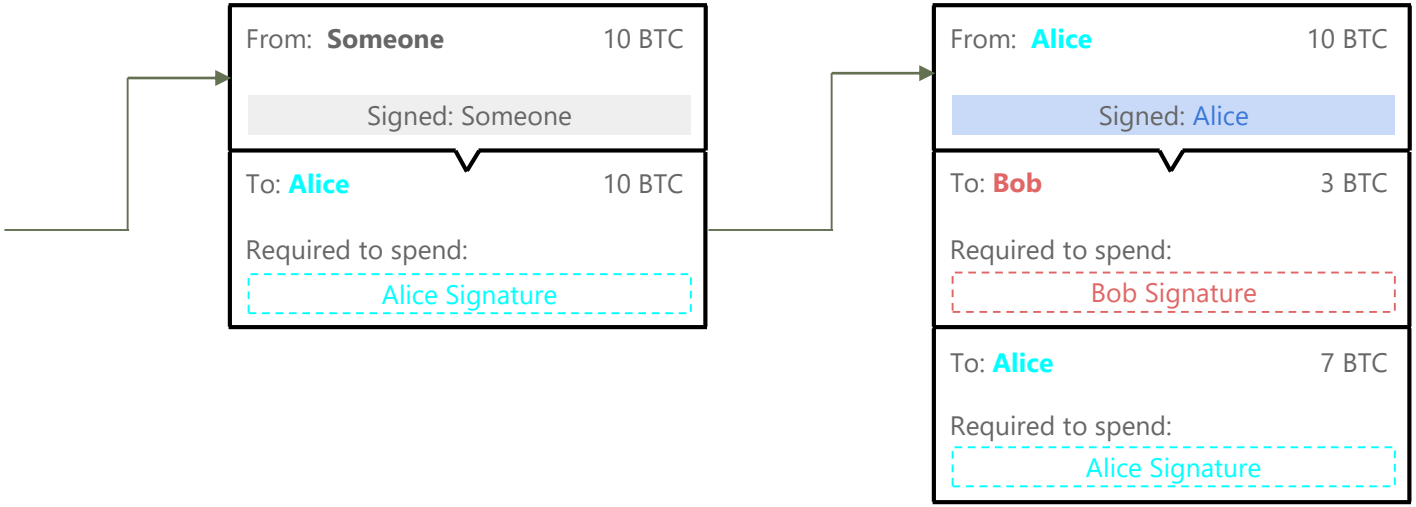
https://captainaltcoin.com/hashed-timelock-contract-htlc/

https://academy.binance.com/glossary/hashed-timelock-contract /

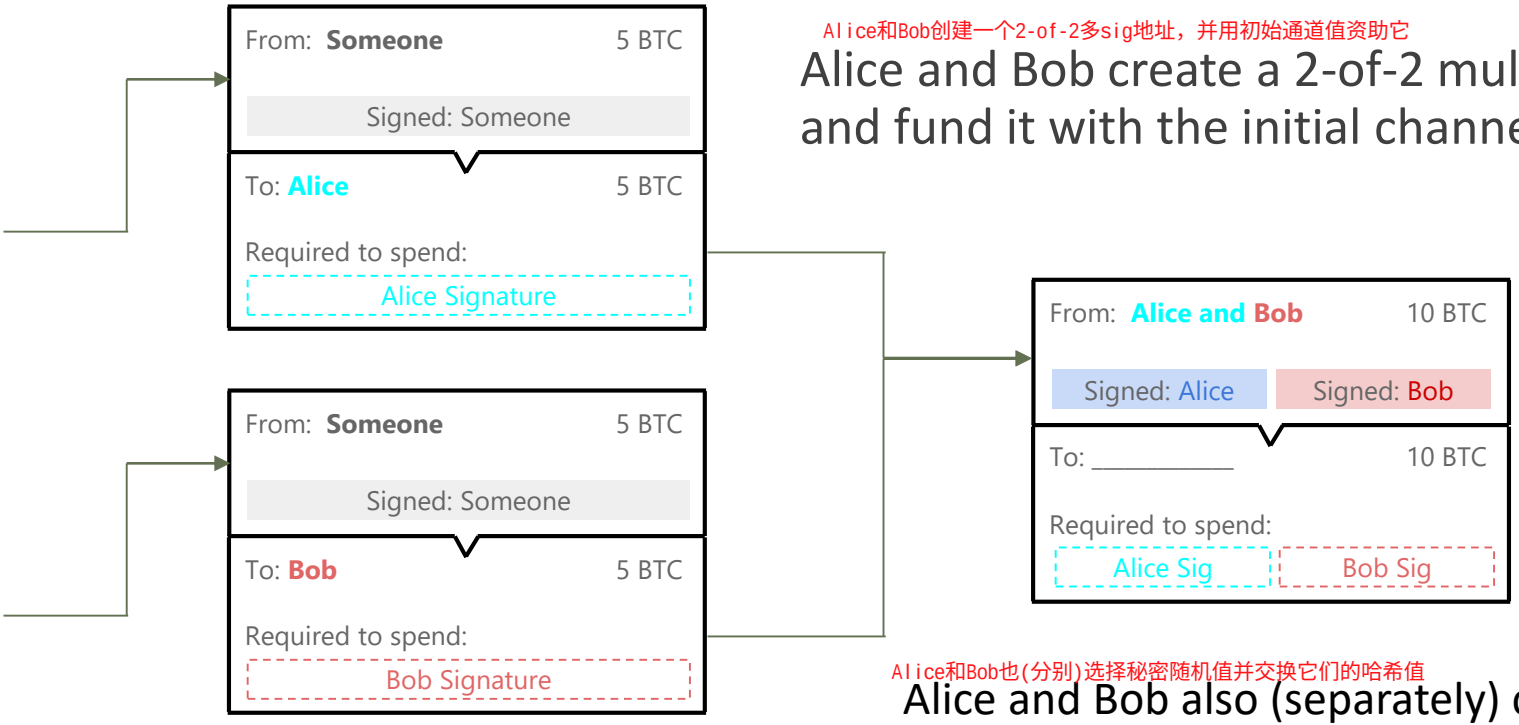
Notations

Briefly, some notations:

+ Alice is spending a 10 BTC txn output
+ Alice sends 3 BTC to Bob and 7 BTC back to herself.



2-of-2 Multisignatures



Alice和Bob创建一个2-of-2多sig地址，并用初始通道值资助它

Alice and Bob create a 2-of-2 multisig address and fund it with the initial channel values.

Alice和Bob也(分别)选择秘密随机值并交换它们的哈希值

Alice and Bob also (separately) choose ***secret*** random values and exchange their hash values.

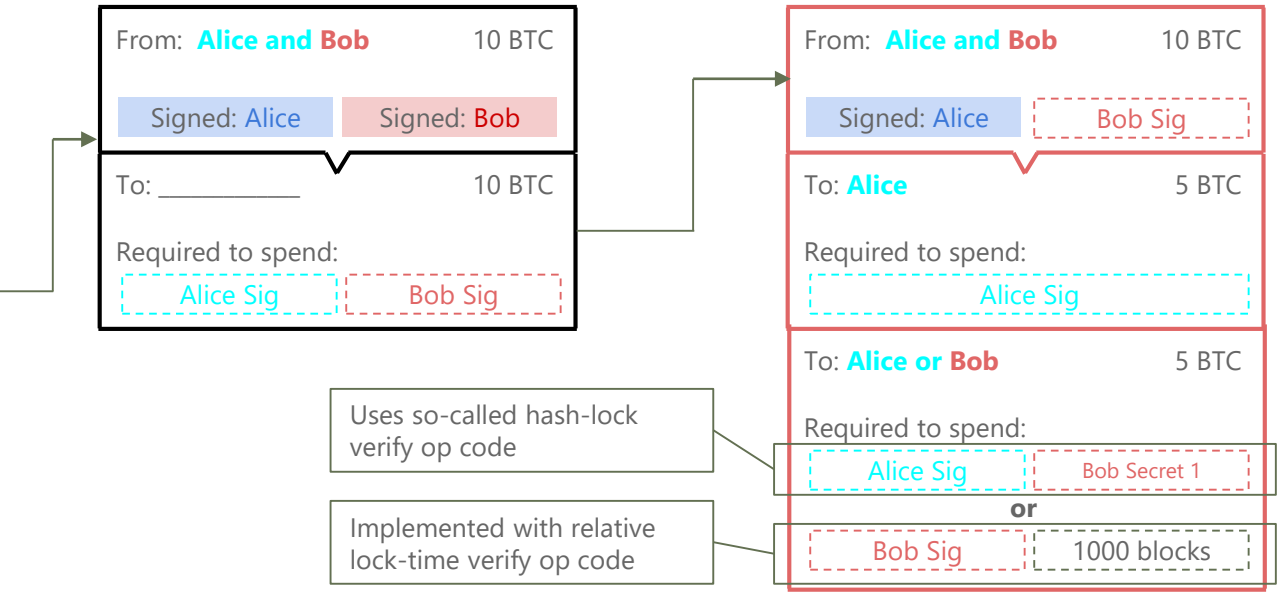
哈希定时锁定的双向支付通道

Hash Time-locked Bi-directional Payment Channels

Alice创建了一个承诺事务。Alice签署了承诺txn并将txn和散列发送给Bob，而没有向网络广播!

Alice creates a commitment transaction. Alice signs the commitment txn and sends the txn and the hash to Bob, ***without broadcasting to the network!***

由于Alice已经签署了multisig输出的一半，Bob可以随时将该事务广播到网络。
但是，如果Bob广播，该事务自动给Alice 5个BTC，并且在等待1000块之后只返回Bob的5个BTC，或者Alice签署并使用Bob的哈希预像(目前只有Bob知道)



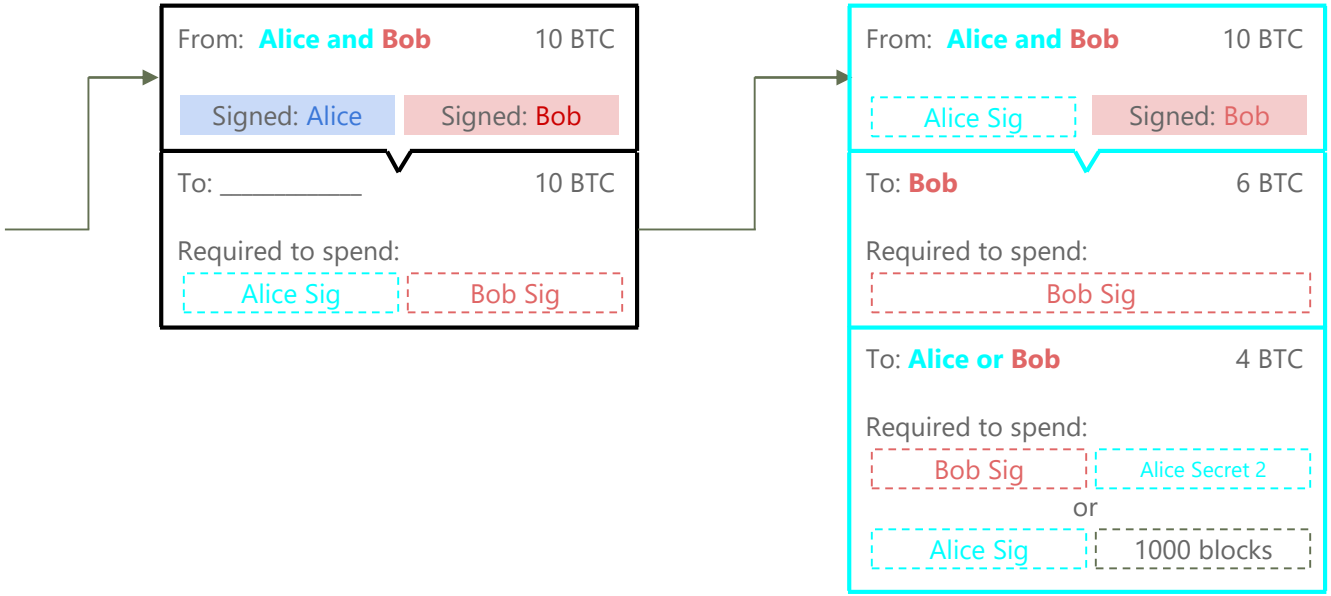
Since Alice has already signed her half of the multisig output, Bob can broadcast this transaction to the network at any time.

However, if Bob broadcasts, this transaction automatically gives Alice 5 BTC, and only returns Bob's 5 BTC *after a waiting period of 1000 blocks*, **or** Alice signs and uses Bob's hash preimage (which only Bob knows currently).

Hash Time-locked Bi-directional Payment Channels

Bob还创建了一个新的承诺txn，它将6个BTC发送回自己，并将4个BTC发送到hash时间锁定契约中。Bob在txn上签名，然后把它发送给Alice

Bob *also* creates a new commitment txn that sends 6 BTC back to himself, and 4 BTC into the hash time-locked contract. Bob signs the txn and sends it over to Alice.



Hash Time-locked Bi-directional Payment Channels

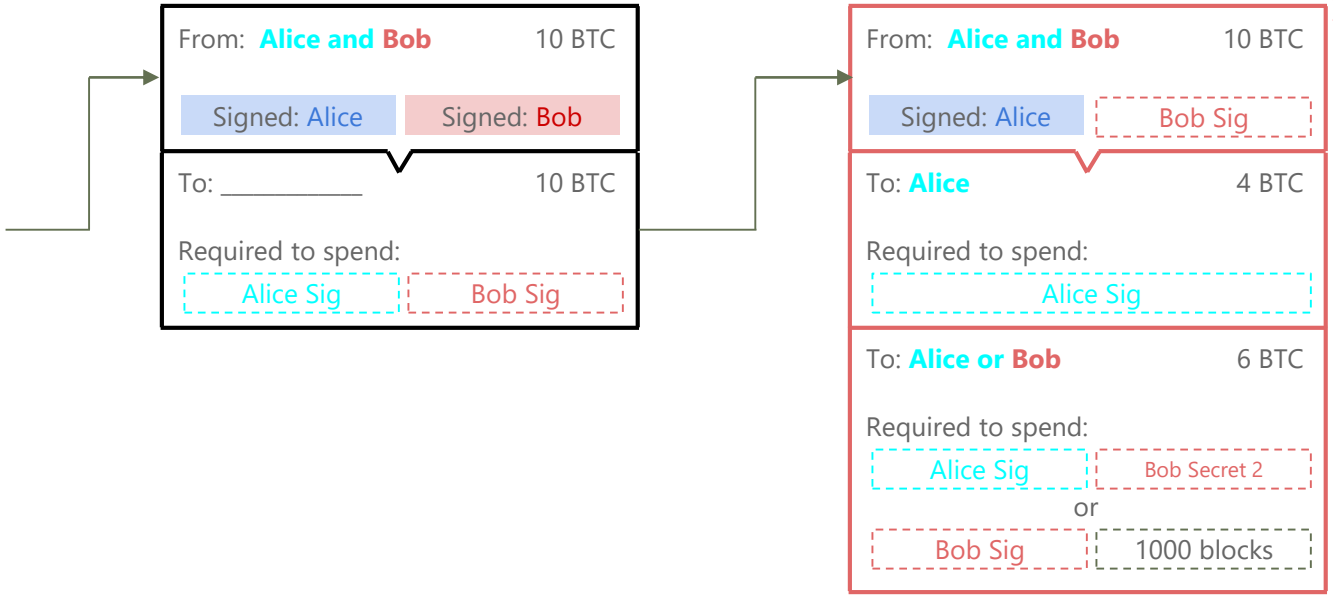
Alice now wants to pay Bob 1 BTC through the payment channel.

Alice现在想通过支付通道向Bob支付1比特币。
Alice和Bob都选择新的秘密值并交换它们的散列

Both Alice and Bob choose new secret values and exchange their hashes.

Alice creates a new commitment txn that sends only 4 BTC back to herself, and 6 BTC into the hash time-locked contract. Alice then signs the txn and sends it over to Bob.

Alice创建了一个新的承诺txn，它只向自己发送4个BTC，并将6个BTC发送到hash时间锁定契约中。然后Alice在txn上签名并发送给Bob



Key Step

Alice和Bob现在交换前两个秘密值以锁定更新的通道余额

Alice and Bob now exchange the first two secret values to **lock in** the updated channel balances!

What if Alice wants to get her 1 BTC back and cheat Bob?

Alice将旧的承诺txn广播到网络
Bob自动接收到他原来的5个BTC。
Alice现在必须等待1000块才能取回她的5个BTC。
然而，由于爱丽丝和鲍勃刚刚交换了他们最初的秘密，鲍勃现在知道了爱丽丝的秘密1!鲍勃可以看到爱丽丝试图欺骗他，并拿了全部10个比特币

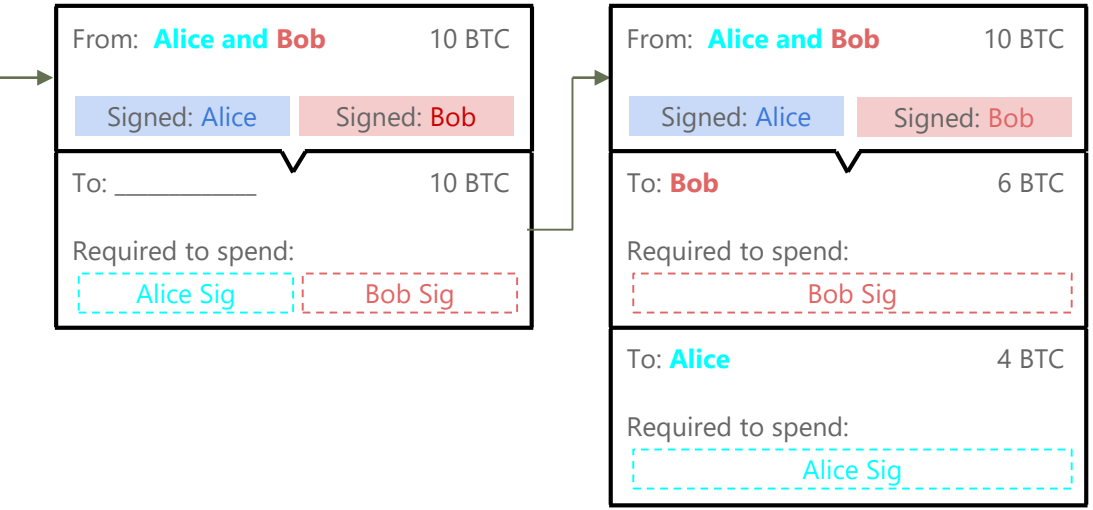
Alice broadcasts the old commitment txn to the network…

Bob automatically receives his original 5 BTC.

Alice now has to wait 1000 blocks in order to get her 5 BTC back.

However, since Alice and Bob just exchanged their original secrets, Bob now knows Alice Secret 1! Bob can see Alice trying to cheat him and take all 10 BTC!

Settlement



当Alice和Bob希望将其余额结算给区块链时，他们只需配合并从原始存款地址开始消费，或者等待上一个承诺的锁足够长的时间过期

When Alice and Bob want to settle their balances to the blockchain, they simply cooperate and spend from the original deposit address, or wait long enough for the lock on the previous commitment to expire.

如果Alice和Bob总是合作，他们永远不需要接触区块链，除非是在创建支付通道和结算余额时。

Observation: If Alice and Bob always cooperate, they never have to touch the blockchain, except when creating the payment channel and settling the balance.

爱丽丝和鲍勃永远都不需要信任对方，因为如果一个人试图欺骗，另一个人总是可以推翻并拿走存款中的所有钱

Observation: Alice and Bob never have to trust each other, since if one tries to cheat, the other can always override and take all the money in the deposit.

Issues

Alice和Bob需要把资金锁在HTLC(哈希时间锁合同/渠道)中，然后才能互相汇款

Issue1: Alice and Bob need to have capital locked up in this HTLC (Hashed Time-Lock Contract/Channel) before they can send money between each other.

有了这个支付渠道，Alice和Bob只能在他们之间轻松地、可伸缩地汇款

Issue2: With this payment channel, Alice and Bob can only *easily* and scalably send money between themselves.

What if Alice wants to send money to Charlie without touching the Blockchain, but she doesn't have or want a payment channel set up between herself and Charlie?

如果爱丽丝想要寄钱给查理而不碰区块链，但她没有或想要在自己和查理之间建立一个支付渠道
我不想和一些亚马逊商户建立支付渠道，如果我只和他们交易一两次。
想法：创建一个支付渠道网络。只要爱丽丝和查理有关系，她就可以给他寄钱

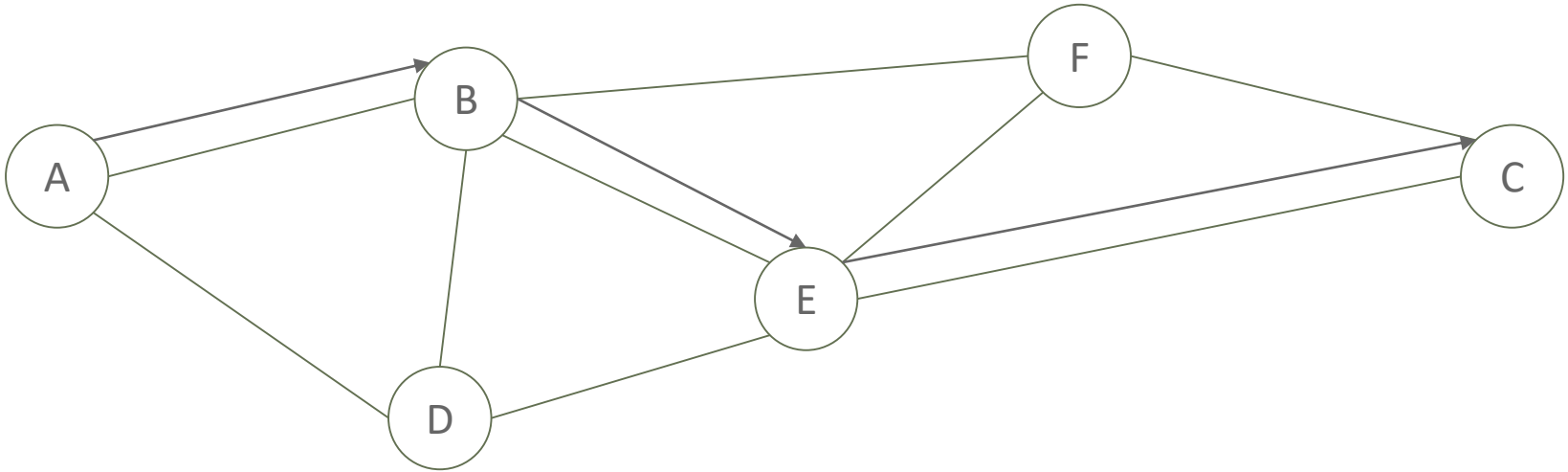
- e.g. I don't want to set up a payment channel with some Amazon merchant if I'm only going to transact once or twice with them.

Idea: Create a network of payment channels. As long as Alice is connected to Charlie, she can send him money.

Proposal

爱丽丝通过这个假设的支付渠道网络把钱发送给查理

Alice sends money to Charlie through this hypothetical payment channel network



Can we do this securely?

事实证明，在HTLC建设的基础上添加一些小功能，我们就可以放心地通过HTLC网络汇款！

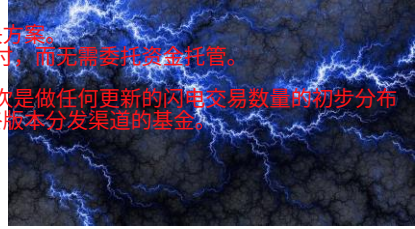
Turns out, with some **small additions** on top of our HTLC construction, we can *trustlessly* send money across a network of HTLCs!

⇒ **Lightning Network**

Lightning Network

Lightning Network

闪电网络是在区块链(最常见的比特币)之上运行的“第二层”支付协议。
·它支持参与节点之间的快速交易，并被吹捧为比特币可伸缩性问题的解决方案。
·它以p2p系统为特色，通过双向支付渠道网络进行数字加密货币的小额支付，而无需委托资金托管。
·Lightning网络实现简化了原子交换。
闪电网络的正常使用由打开一个付款通道通过提交资金交易有关区块链,其次是做任何更新的闪电交易数量的初步分布区块链不广播频道的基金,可选地,后面可以跟关闭付款通道广播事务的最终版本分发渠道的基金。



The **Lightning Network** is a "second layer" payment protocol that operates on top of a blockchain (most commonly Bitcoin).

- It enables fast transactions between participating nodes and has been touted as a solution to the bitcoin scalability problem.
- It features a peer-to-peer system for making micropayments of digital cryptocurrency through a network of bidirectional payment channels without delegating custody of funds.
- Lightning Network implementation simplifies atomic swaps.

Normal use of the Lightning Network consists of opening a payment channel by committing a funding transaction to the relevant blockchain, followed by making any number of Lightning transactions that update the tentative distribution of the channel's funds without broadcasting to the blockchain, optionally followed by closing the payment channel by broadcasting the final version of the transaction to distribute the channel's funds.

The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments

--By Joseph Poon & Thaddeus Dryja https://lightning.network/lightning-network-paper.pdf

What does the Lightning Network mean for scalability?

1)如果我们假设这个支付渠道网络有足够的资金，人们可以立即进行支付。
2)只使用比特币区块链作为仲裁机构来解决纠纷和关闭支付渠道，这意味着区块链上的交易少得多(昂贵的)。
3)比特币网络可以支持1000秒以上的tps，而不是3个tps，通过将支付委托给每个支付渠道的简单记账，99%的时间保持外链！
4)由于闪电网络交易相对便宜，费用可能也会便宜得多。

1) If we assume that there is enough capital in this payment channel network, people can make payments *instantly*.
2) Only use the Bitcoin Blockchain as an *arbiter* to settle disputes and close out payment channels, meaning far fewer (expensive) transactions on the Blockchain.

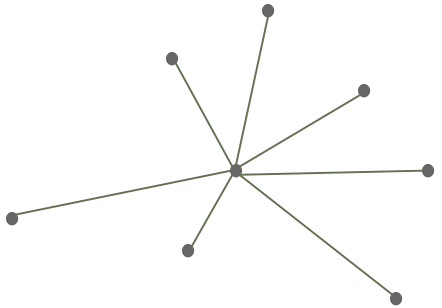
3) Instead of 3 tps, the Bitcoin network can support 1000's+ of tps by delegating the payments to simple bookkeeping in each payment channel, which is kept off-chain 99% of the time!
4) Since Lightning Network transactions are relatively cheap, the fees will likely be much cheaper as well.

Issues with Lightning Network

- Nodes need to keep very large amounts of capital locked up in payment channels.
- Strong centralization force, since only nodes with significant capital can afford to hold payment channels for long.
- Less capital is required with less nodes on the network ⇒ tendency towards hub-and-spoke network topology.

●节点需要将大量资金锁定在支付渠道中。
●集中力强，只有资本雄厚的节点才有能力长期持有支付渠道。
●网络上的节点越少，所需要的资本就越少，因而网络趋向于轮辐式网络拓扑结构

Hub-and-Spoke Topology



2019 bitcoin lightning torch

· 2019年1月19日，匿名推特用户hodlonaut开始了一个类似于游戏的闪电网络推广测试，向一个受信任的接收者发送100,000 satoshis(0.001比特币)，每个接收者添加10,000 satoshis(当时为0.34美元)发送给下一个受信任的接收者。
· “闪电火炬”的支付对象包括Twitter首席执行官杰克·多尔西、lightning Labs首席执行官伊丽莎白·斯塔克和Binance首席执行官“CZ”赵长鹏等知名人士。
· 闪电火炬通过了292次，最终达到了先前硬编码的439万satoshi的上限。闪电火炬的最后一笔款项于2019年4月13日捐赠了429万satoshis(当时约合217.78美元)给委内瑞拉比特币(Bitcoin Venezuela)，这是一家在委内瑞拉推广比特币的非营利组织。

- On January 19, 2019, pseudonymous Twitter user hodlonaut began a game-like promotional test of the Lightning Network by sending 100,000 satoshis (0.001 bitcoin) to a trusted recipient where each recipient added 10,000 satoshis ($0.34 at the time) to send to the next trusted recipient.

- The "lightning torch" payment reached notable personalities including Twitter CEO Jack Dorsey, Lightning Labs CEO Elizabeth Stark, and Binance CEO "CZ" Changpeng Zhao, among others.

- The lightning torch was passed 292 times before reaching the formerly hard-coded limit of 4,390,000 satoshis. The final payment of the lightning torch was sent on April 13, 2019 as a donation of 4,290,000 satoshis ($217.78 at the time) to Bitcoin Venezuela, a non-profit that promotes bitcoin in Venezuela.

Summary

比特币和其他类似的区块链有一个可伸缩性的问题:
a)如果这些技术想要在全球范围内使用，它们需要支持适当的交易量。
b)我们能否在不损害比特币安全、去中心化、无信任支付的初衷的情况下解决这个问题?

- Bitcoin and other similar Blockchains have a scalability issue:
  a) If these technologies want to be used on a global scale, they need to support appropriate transaction volumes.
  b) Can we solve this issue without compromising Bitcoin's original vision of secure, decentralized, trustless payments?
- Proposed Solutions:
  a) Blocksize Capacity Increase
  b) Segregated Witness
  c) Sidechains
  d) Lightning Network

1)区块容量增加
a)大块的小可伸缩性提升。
b)当节点的最小服务器需求增加时，集中化的风险。
2)隔离证人(SegWit)
a)小的可伸缩性提升，因为块不需要存储签名。
3) Sidechains
a)大规模可伸缩性提升的潜力
b)具有更好可扩展性的潜在新侧包(尚待实践)
4)闪电网络
a)大规模可伸缩性提升的巨大潜力。
b)支付流程的根本性重组。
c)资本先决条件导致的集中风险

1) Blocksize Capacity Increase
   a) Small scalability boost with larger blocks.
   b) Centralization risk as minimum server requirements for nodes increases.
2) Segregated Witness (SegWit)
   a) Small scalability boost since blocks don't need to store signatures.
3) Sidechains
   a) Potential for large scalability boost
   b) Potential novel sidechains with better scalability (yet to be seen in practice)
4) Lightning Network
   a) Large potential for orders-of-magnitude scalability boost.
   b) Fundamental restructuring of payment process.
   c) Centralization risk due to capital prereqs.

Lecture Outline

- ✓ Scalability Problem for Bitcoin
- ✓ Proposed Scalability Solutions
  - Blocksize Capacity Increase
  - Segregated Witness
  - Sidechains
  - Lightning Network

完

धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

Thank You

English

شكراً

Arabic

Obrigado

Brazilian Portuguese

Grazie

Italian

多谢

Simplified Chinese

Danke

German

Merci

French

நன்றி

Tamil

ありがとうございました

Japanese

감사합니다

Korean