闪电网络学习研究总结

郭世清 2016.07

一.现实世界的需求

随着全球金融贸易电子商务的蓬勃发展，电子支付系统的交易量，峰值不断创下新高。以2015年数据为例，Visa的支付网络峰值能达到5.6万笔每秒，每天交易量达到数亿笔交易。
一个国际化的得到广大公众接受与认可的，取得巨大成功的电子支付系统，从技术角度必须能满足以下特点要求：

1.大规模容量交易的能力
2.高吞吐的网络处理能力
3.安全的支付环境与交易数据
4.可接受的极低的交易费用
5.0延迟的交易确认即即时支付能力

比特币网络作为在全球化分布式账本，随着它不断被公众接受，更广泛被应用在各个领域。比特币网络正在努力成为一个国际化的清算网络，或者一个全球化的支付平台。比特币网络能够在不降低现有的安全性和保证分布式去中心化的前提下，能不断发展为能涵盖全球交易并且能满足以上所列要求的支付网络平台。这一个构想如果能在未来实现，这将会是人类经济活动和社会组织的重要标志性事件。

二.比特币网络的现有障碍

目前，比特币网络有以下三个众所周知的重要系统参数

1.区块大小1MB

2.区块生成时间间隔10分钟

3.6个区块建议确认周期

因为1和2的原因，按每笔交易大小200B计算，一个区块最多可容纳5000多笔交易，比特币网络的交易处理能力理论上约为9笔每秒，但因为网络中区块生成的泊松分布特点，实际真实处理能力上限在5笔每秒。因此，在网络吞吐量指标上，去中心化的比特币网络与中心化的Visa网络相比差距巨大，足足有4个数量级的差异。

另外，目前比特币一天生成144个区块，一天最多容纳交易数不超过100w笔，明显交易容量不足。在网络规模容量上，同样的，去中心化的比特币网络与中心化的Visa网络相比差距巨大。

同时，为了保障资金安全，一笔交易在比特币打包进去区块之后，建议是在获得链后6个区块的确认之后才安全确认。但是因为区块生成和矿工的随机性，往往一个安全的交易确认需要耗时1小时以上。这与目标相差甚远。

综上所述，现行的比特币网络无法在在坚持去中心化特性的同时，并能达到大规模交易，高吞吐，即时支付确认的目标。

因此，解决可扩展问题是达成目标的关键所在。

三.社区提出的可扩展问题解决方案

一.两个思路

1.区块扩容

短期简单有效，显而易见的一个方案是通过区块扩容。改变比特币核心中对区块大小的限制，让区块更大使得容纳更多的交易，网络处理能力可以随着区块大小的增长而线性增长。这个方案虽然一定程度上可以提升比特币网络的处理能力，但依然有其他的问题。

首先改变比特币核心中 MAX_BLOCK_SIZE C参数会带来硬分叉，硬分叉要求所有全节点统一升级，未能升级的全节点可能有资金安全风险，硬分叉实行过程也可能有其他未知风险。

另外，区块中交易数大幅增加之后，验证工作量将加大，节点对区块交易的验证时间也将快速增加，这可能会被黑客加以利用来发起攻击，并且也有可能导致中心化的风险。

还有，该思路的最大问题在于，只能有限程度的解决问题，与目标有几个数量级的差距，并没有解决大规模交易的网络负载问题。可以设想一下，如果要达到Visa网络 5.6万笔交易每秒的交易吞吐率，那一个区块的大小大概需要6000M，一天产生800多G的区块数据，还需要考虑节点对区块交易的验证。这对节点的网络带宽，存储，计算能力的挑战巨大，目前来看完全不可接受。

2.链下支付链上结算

面对区块大小容量和交易处理能力的貌似不可调和的矛盾。社区另辟蹊径，提出了新的思路方案，即off-chain的思路。把尽可能多的交易放到主链以外，尽可能的延后交易提交到主链的时间，如非必要尽量不提交交易到主链。链下系统(或网络)解决交易对手风险，即时支付，交易吞吐和交易规模问题。
基于此思路，社区出现了解决可扩展性三个具体方向：

a.第三方托管

交易各方将资产和交易都托管给同一个被信任的权威第三方，第三方代交易各方持有资产和更新各方的资产更新。交易各方信任第三方的交易安全和资产安全和交易成本。交易各方在该第三方平台上可以无限次交易而无需与公链发生交互。

第三方托管平台内部是中心化架构，一个典型的实现是SQL数据库管理各方交易数据与资产数据。交易各方在第三方托管平台上交易无需与公链发生交互，仅仅是需要在第三方托管平台与公链做资产转移的时候才需要与公链交互。已被实现的典型例子是比特币的交易所。

第三方托管的方案让网络获得了扩展性，也提升了网络整理处理能力。但是中心化架构与管理明显背离比特币网络所倡导的去中心化思路，另外资产安全，缺乏监管，隐私安全，中心化带来的管理风险等等这些都是必须面对的问题。

MT.GOX 就是一个典型的案例。因此该方案未来可能不会成为主流不会成为社区首选。

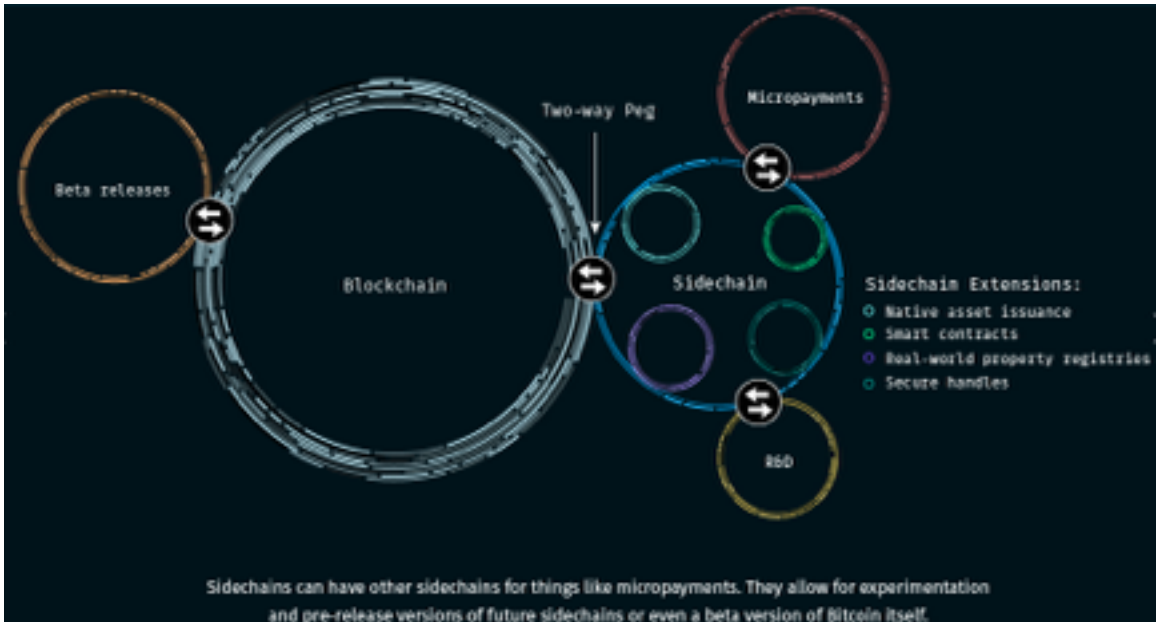
b.侧链

在比特币主链之外，创建一个新的区块链。主链与侧链直接可以双向互操作，资产可以从主链导出到侧链，并且可以从侧链返回主链。白皮书称为“楔入式侧链”。

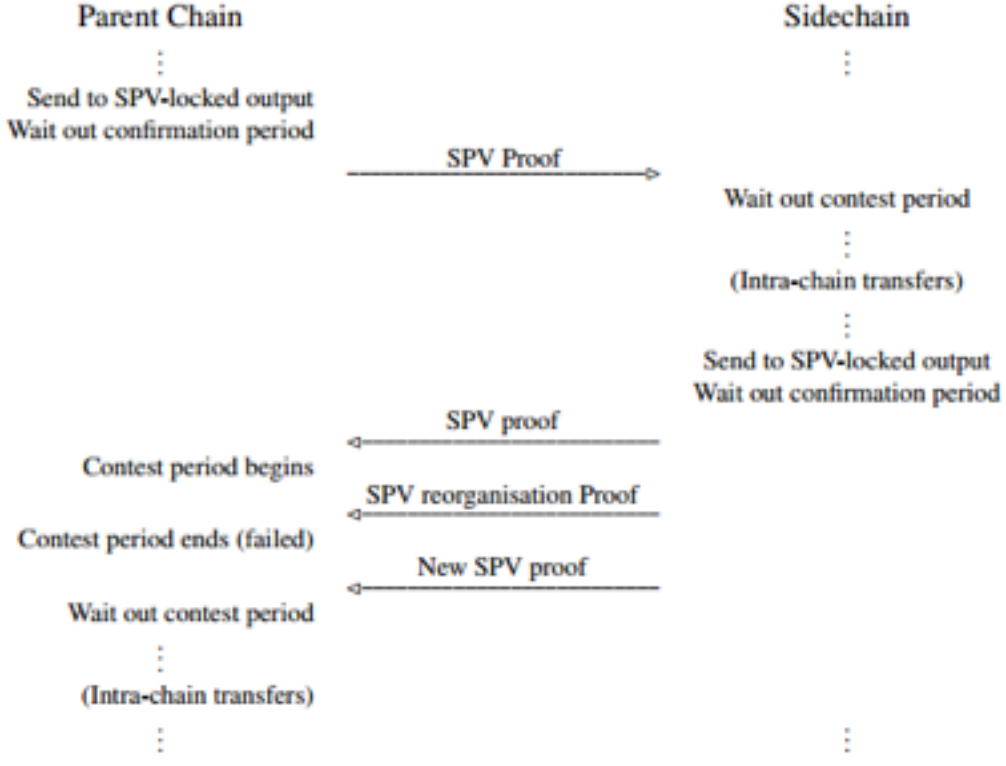
白皮书可查看https://blockstream.com/wp-content/uploads/2014/10/sidechains.pdf

该方案难点在于保证资产在主链与侧链之间的安全转移。该方案基于两点技术基础，增强的比特币脚本系统指令和简单支付验证证明（SPV证明）。

当资产从主链向侧链转移的时候，主链上创建一个锁定资产的父交易，并且通过SPV的密码学证明保证该锁定已正确完成，然后在侧链上创建一笔输入为主链父交易的子交易。该子交易可以在侧链里面无限次交易，该资产可以在侧链里面自由转移而不需与主链进行任何交互。当资产需要从侧链转移到主链时，侧链将资产发送到一个SPV锁定的输出，并产生一个SPV证明该输出已完成，使用该证明来解锁之前主链上被锁定的输出。可以参考以下流程图：



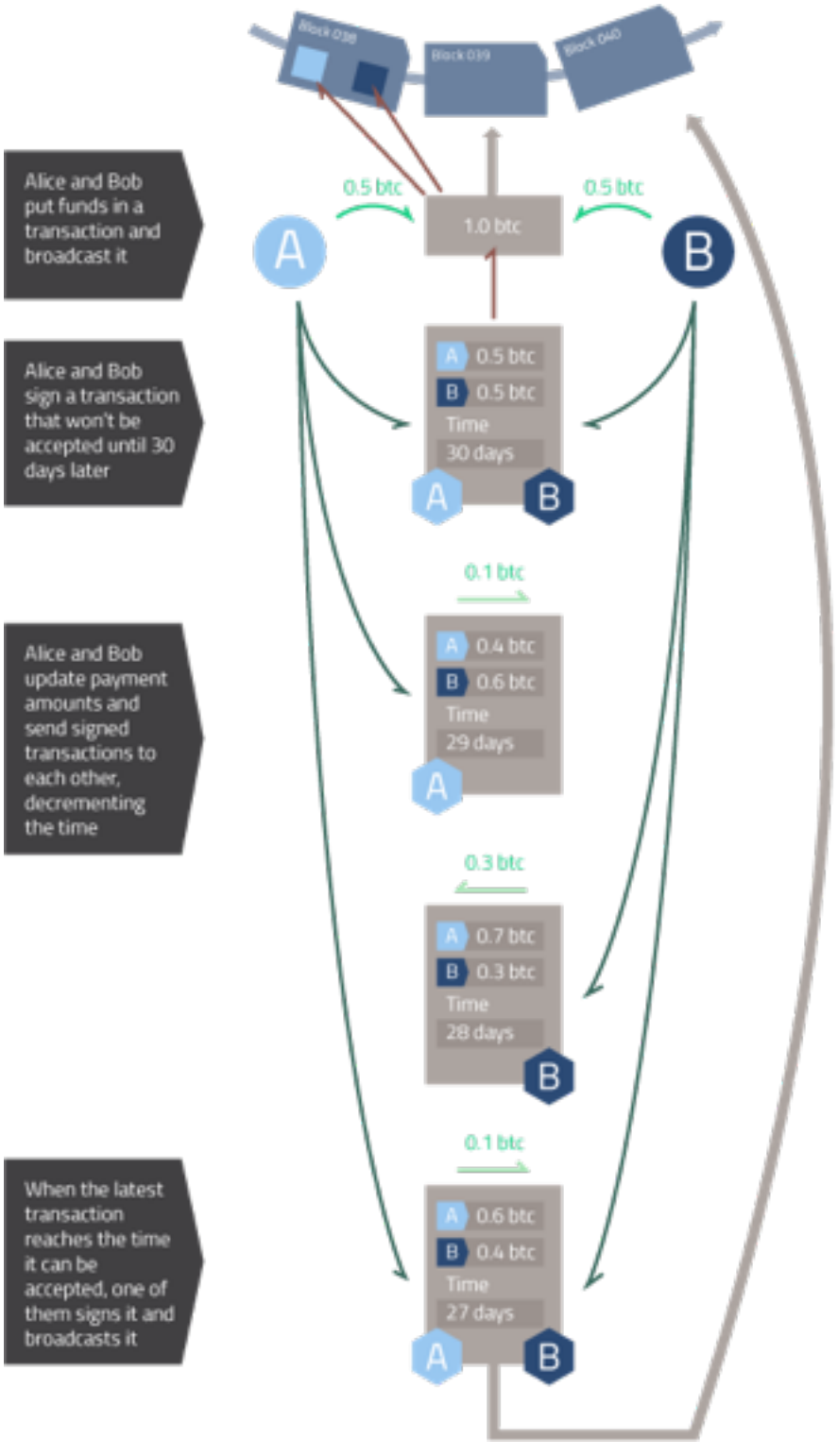
Two-way Peg的一个转移过程



以侧链方案的设想，侧链有更好的网络处理能力和更好的隐私保护，并且解决了链与链之间的双向楔入之后，主链之外可以有多个侧链，主链与侧链，侧链与侧链都可以双向楔入做资产转移。侧链本身可以承载微支付，智能合约，现实世界资产登记，资产转换等等，每个侧链都可以根据需要承载不同的资产和渠道。因此，主链与多个侧链链链相扣，几乎能够涵括了世界所有的金融交易，使得主链获得了几乎无限的扩展性。仅此而言似乎是个覆盖范围广泛，能解决所有困难的金钥匙。
如果侧链内承载微支付能力，侧链本身通过自定制改造不同于主链的区块链以追求更好的交易处理能力，这点是可行的，并且一定可以获得比主链更好的交易处理能力，这个无须怀疑。
但是侧链的基础依然是区块链，区块链在面对区块大小容量和交易处理能力的矛盾，可以做一定的妥协提高，但并不能达到我们的目标，这一点在上面已经论述过了。
因此，侧链是解决可扩展性问题的一把利器，可以让主链获得近乎无限的可扩展性。但在区块链的交易处理能力上似乎并没有看到多个数量级的突破。

c.支付渠道

所谓支付渠道既是交易参与各方无需信任，直接建立联系进行交易，并且在该连结上可以进行多次交易。在该渠道中发生的交易只在该渠道中存在，并不需要与主链发生交互。仅仅在渠道建立和渠道关闭的阶段需要与主链发生交互。渠道建立阶段交易双方通过主链的脚本系统锁定资产，渠道建立后渠道维护参与方的平衡更新，渠道关闭阶段以最新的渠道平衡状态为结算更新到主链上。
在主链以外创建一个网络，该网络并不以区块链为基础，网络中以交易各方为节点，根据交易需要交易各方自动连结成支付渠道，自动形成一个P2P网络。如果该网络中参与方足够多，或者支付渠道足够多，那该网络就可以成为覆盖全网的支付渠道网络。
因为交易只发送在渠道中的交易参与方，网络中其他节点完全无需关注，因此能大量减少在主链上的交易，让网络承载大规模交易成为可能。不但渠道中的交易可以认为是即时支付，并且交易费用也是极低的。
对交易参与方而言，只要渠道没有频繁的建立和关闭，而是长时间稳定存在，并且渠道中交易越多，支付渠道的作用越大，越能体现对主链网络大规模交易处理能力的提升。这点对一个全网规模的支付网络尤为重要。
一个支付渠道的建立过程如下：


Block 038
Block 039
Block 040
Alice and Bob put funds in a transaction and broadcast it
0.5 btc
0.5 btc
A
1.0 btc
B
Alice and Bob sign a transaction that won't be accepted until 30 days later
A 0.5 btc
B 0.5 btc
Time
30 days
A
B
0.1 btc
Alice and Bob update payment amounts and send signed transactions to each other, decrementing the time
A 0.4 btc
B 0.6 btc
Time
29 days
A
0.3 btc
A 0.7 btc
B 0.3 btc
Time
28 days
B
0.1 btc
When the latest transaction reaches the time it can be accepted, one of them signs it and broadcasts it
A 0.6 btc
B 0.4 btc
Time
27 days
A
B

支付渠道是无信任，定期结算，基于时间微支付，去中心化的方案。可以做到即时支付，0交易费用，安全的交易数据和安全的隐私保护。通过众多支付渠道的互相连结而形成的大的可以有无限交易的支付渠道网络，可以很好的解决比特币网络的可扩展问题。

二.链下(off-chain) 的几种可能的支付渠道解决方案

1.闪电网络（比特币）

2015年比特币社区的Joseph Poon和Thaddeus Dryja 提出了一个允许任意双方即时支付的Lightning Network构想。
https://lightning.network/
https://lightning.network/lightning-network-paper.pdf
该构想方案是一个没有第三方信任风险，支持即时支付，大规模交易的去中心化网络系统。
闪电网络主要解决三个问题，一个是即时支付，在比特币网络里面发起一个交易从广播到等待6个区块的确认大约需要1个小时，在闪电网络里面，支付不需要等待区块的确认，它完全是即时和原子的。第二个小额支付，闪电网络的交易费用极低，比现在比特币网络中一笔交易最小输出还要小几百倍，这将使得小额支付在无托管风险的清空下在比特币网络真正可行。第三个是可扩展性，闪电网络的交易发生在Off-chain,交易各方在闪电网络中可以发起无限的交易，极大的扩展了比特币网络的大规模交易容量和网络处理能力。
目前有一个开源的alpha status的项目版本 开发语言Java https://github.com/blockchain/thunder

2.Raiden 网络（以太坊）

受启发于闪电网络Off-chain交易On-chain结算的思路，以太坊平台上也提出了基于以太坊的支付渠道网络雷电网络解决方案。雷电网络设计思路与闪电网络非常类似，技术实现由于以太坊平台本身的支持相对闪电网络在比特币网络上的实现可能会更为容易些。
目前有一个开源的开发中的项目版本，开发语言Python
https://github.com/raiden-network/raiden

3.DMC（比特币）

2015年苏黎世联邦理工学院的 Christian Decker和Roger Wattenhofer教授提出比特币网络Off-Chain的新的微支付方案Duplex Micropayment Channels（DMC）。DMC与闪电网络最大的不同在于，闪电网络需要对私钥进行更变合同状态的更变，而DMC，则使用了减退的 timelocks。
Paper地址http://www.tik.ee.ethz.ch/file/716b955c130e6c703fac336ea17b1670/duplex-micropayment-channels.pdf
DMC项目未开源。

三.其他小额支付方案

1.ChangeTip

https://www.changetip.com/
基于现有社交网络的比特币小额支付平台。类似于第三方托管的模式。
最新消息Airbnb收购了ChangeTip团队。

2.BlockCypher

http://www.blockcypher.com/
BlockCypher是一家提供区块链API和比特币钱包服务的提供商。
他们宣称Confidence Factor（置信因子）技术可以让交易在比特币网络区块链上快速处理，基于此，他们的Microtransaction API 可以让主链上小额支付成为可能。

四.闪电网络

一.闪电网络的几个前置依赖

1. Relative lock-time using consensus-enforced sequence numbers

https://github.com/bitcoin/bips/blob/master/bip-0068.mediawiki
BIP68提出利用交易数据中每个txin的nSequence 字段实现相对时间锁的序列号，以保证一个签名有效的交易在一个特定的相对时间周期内不会被矿工挖到发布到区块链上。
相对时间锁分为两种模式。一种是基于时间的，必须在该输出被矿工挖到之后至少等待N个512秒单位之后该交易才能被矿工放入新的块。另外一种是基于块高度的，必须在该输出被矿工挖到之后至少等待N个块该交易才能被矿工放入新的块。
该序列号可以通过第31位bit设置来声明是否启用该特性，第22位bit为来声明使用何种类型时间锁。第15位到第0位代表时间锁的值。
注意交易结构中的nSequence 与nLockTime的不同在于，nLockTime是要求该交易必须在指定的日期点之后才能被矿工挖到新区块中，而nSequence 是要求在一个特定的时间周期间隔之后才能被矿工挖到新区块中。

这个特性功能是实现闪电网络的Hashed Timelock Contracts (HTLC) 的基础之一。

2.CSV（CHECKSEQUENCEVERIFY）

https://github.com/bitcoin/bips/blob/master/bip-0112.mediawiki
该BIP提出比特币脚本系统的新的opcode CHECKSEQUENCEVERIFY，帮助上面BIP68在比特币区块链上的实现。
矿工从内存池中检出交易来放入新区块的时候，如果交易的兑现脚本中有CHECKSEQUENCEVERIFY，则将CHECKSEQUENCEVERIFY的参数与交易的nSequence 指定的块高度条件或者时间周期进行对比，如果没有达到，则校验失败，不会放入当下新区块。
一个例子如下：

```
IF
   2 <Alice's pubkey> <Bob's pubkey> <Escrow's pubkey> 3 CHECKMULTI-
SIGVERIFY
ELSE
   "30d" CHECKSEQUENCEVERIFY DROP
   <Alice's pubkey> CHECKSIGVERIFY
ENDIF
```

3.SegWit （隔离见证）

https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki
隔离见证主要是解决交易延展性问题。所谓交易延展性问题既是同一个交易(同样的输入和输出)有可能有不一样的交易ID。
交易延展性问题影响到了未确认的交易。对于等待6个区块确认的交易而言，它并没有任何影响，不会出现双花之类的严重问题。但是对比特币区块链的扩展应用像闪电网络就非常重要。因为有了交易ID的唯一性保证，闪电网络可以无交易对手风险的创建一个未经确认的交易。

为了解决交易延展性问题而提出的隔离见证BIP的原理很简单，因为之前的设计上的缺陷，交易数据和签名(见证)数据混在了一起，导致同一个交易有可能有不一样的交易ID，所以现在把签名数据拆分出来单独存放到新的结构就可以了。这个既是所谓的隔离。

下面简单回顾下交易延展性问题的存在。

Txid = double SHA256 ([nVersion][txins][txouts][nLockTime] 序列化)

这个是交易ID的生成方式。所以对同一个交易而言(同样的输入和输出)，如果上面的这些字段里面有任何的改变都有可能出现不同的交易ID。

问题出在txins里面。典型的txin 可能是这样的<...><...><scriptSig><...>。主要关注scriptSig。 scriptSig 是私钥签名+txin的地址公钥。但注意私钥签名是对整个交易数据进行签名，但是这个签名又是交易数据的一部分，签名是做不到自己对自己签名的。

所以现在比特币是这么处理的，第一步先构造整个交易结构数据，scriptSig部分先用对应txin的兑现脚本填充，第二步用私钥对交易数据进行签名，第三步将签名+对应txin的公钥地址替换到交易数据里面去。

ECDSA签名有个重要的特点，对签名数据取反，它依然可以验证通过。所以对收到交易的矿工而言可以很容易对交易数据里面数据进行替换但依然有效。对同一个交易而言，并不会出现资产被盗取的问题，因为交易的输入和输出都是不变的，但可以制造出不同的交易ID。这个就是交易延展性问题所在。

BIP141 提出的是把签名数据移到 新结构 witness中

Wtxid=double SHA256 （[nVersion][marker][flag][txins][txouts][witness][nLockTime]）

隔离见证是实现闪电网络重要先行条件。因为如果交易ID可以变化，那么依赖父交易的退款交易合同和违约承诺合同都将无效。同时唯一的交易ID对交易参与方或第三方实现对特定交易的监控也有帮助。

二.理解RSMC与HTLC

1.RSMC (可撤销的序列号到期合同-无条件的有期限的资金转移)

Recoverable Sequence Maturity Contract定义了双向支付渠道的最基本工作方式。
有期限的意思是指交易在区块链上的发布生效有时间限制。回顾上文提到的CSV，利用nSequence 指定的时间周期，如果一个交易的nSequence 没有到期，这笔交易虽然能够广播，但是不会立即发布到区块链上生效。
因为有期限的特性支持，交易也支持可撤销特性。
闪电网络里面允许创建多个可以消费同一个父交易输出的不同子交易。但这些子交易是只能一个生效到区块链(不然就是双花)，其他的都是无效的，所以称这个交易为可撤销的就是这个意思。之所以能做到这点也是利用了nSequence。这点可以回顾上面的CSV BIP。
无条件只是一个相对HTLC而言的概念，既是解锁交易输出的时候只要满足脚本签名即可，而HTLC的有条件是指除了脚本签名之外还必须有个哈希锁需要满足，这个哈希锁相当于接收方才拥有的一个私有R才能打开。

2.HTLC(哈希时间锁合同-有条件的有期限的资金转移)

Hashed Timelock Contract 比RSMC更进一步，在交易的资产转移上设定了条件，因此比RSMC要多一些步骤。也因为有条件才转移这个特性，才使得在网络中交易双方通过连结起多个中间节点形成多跳支付链路，让资产按照设定方向拉动，最终让接收方安全接受资产成为可能。
所谓哈希锁合同就是有条件的意思，交易的接收方向发起方提供一个私有R生成的Hash(R)，发起方构建交易的时候把匹配Hash(R)作为解锁输出的一个条件(如果没

有匹配这个Hash值条件资金自动退回发起方)，这样交易发出去之后安全的保证了只有知道R的接收方才能收到资产。
时间锁和上面RSMC的有期限是类似一样的意思，只是用了利用了不同的扩展特性，一个是相对时间，一个是绝对时间。请回顾下上面nSequence 与nLockTime的不同。

3.SIGHASH NOINPUT交易

我们都知道，每个区块链的交易都有输入和输出，当前交易的输入是父交易的输出，当前交易的输出是下一个交易的输入。并且当前交易里面必须有父交易输出的签名。那这样的话，我们无法在知道父交易签名之前就创建一个消费父交易输出的子交易。因此，SIGHASH NOINPUT解决的问题就是允许通过没有父交易输出签名的条件下创建该输出的消费子交易。

SIGHASH NOINPUT 操作顺序如下：

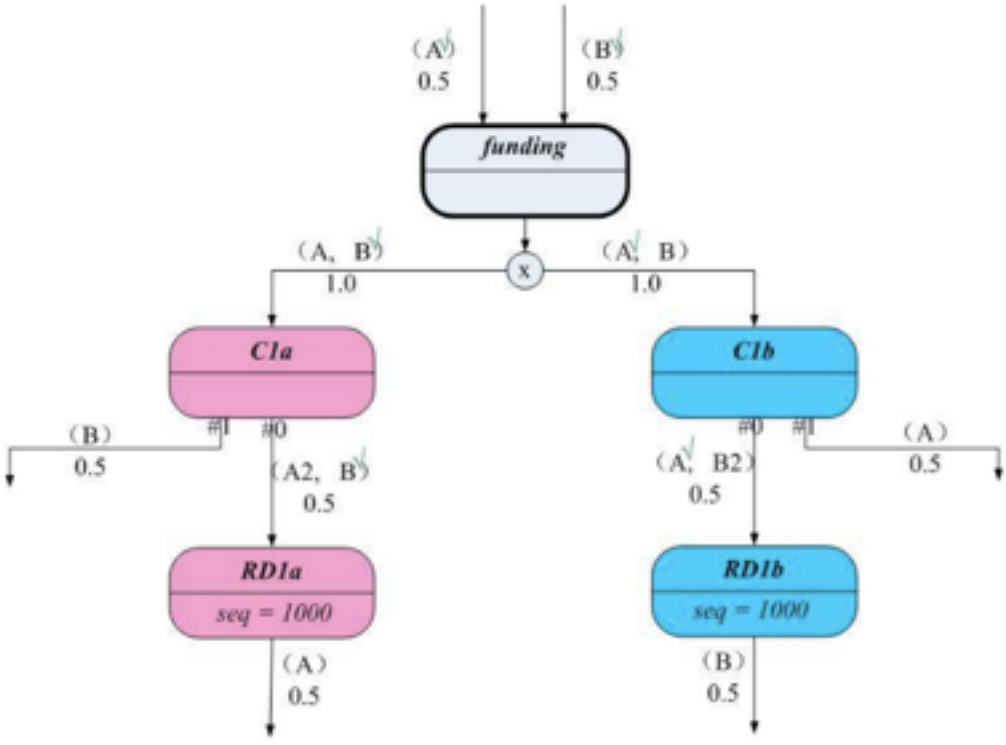
1.创建父交易
2.创建父交易下的花费子交易和子交易下的花费孙交易
3.交易双方交换各自在子交易和孙交易的签名
4.交易双方各自对父交易签名
5.交易双方交换4里面的签名
6.在区块链上面发布父交易签名

回顾下上面提到的SegWit隔离见证，已经解决了交易延展性问题。

三.双向无条件支付渠道(RSMC)

下面的图例来自 http://www.8btc.com/ln-rn-corda 图例符号含义请参考链接
红色是A持有的交易，蓝色是B持有的交易，funding是发布在区块链上的交易
用圆框的交易是2-OF-2交易

1.创建渠道



这个是渠道初建立时候的交易关系，是交易双方A，B在链下互相交互签名建立起来的。请回顾上面的SIGHASH NOINPUT。创建过程如下

1.创建funding父交易(只创建，不签名)，输出要求A与B的签名

2.A创建C1a，RD1a 交易，B创建C1b，RD1b交易

3.A给B的 C1b，RD1b签名，B给A的C1a，RD1a签名

4.A和B各自对funding父交易签名

5.A与B交换 funding父交易的签名

6.发布funding父交易到区块链

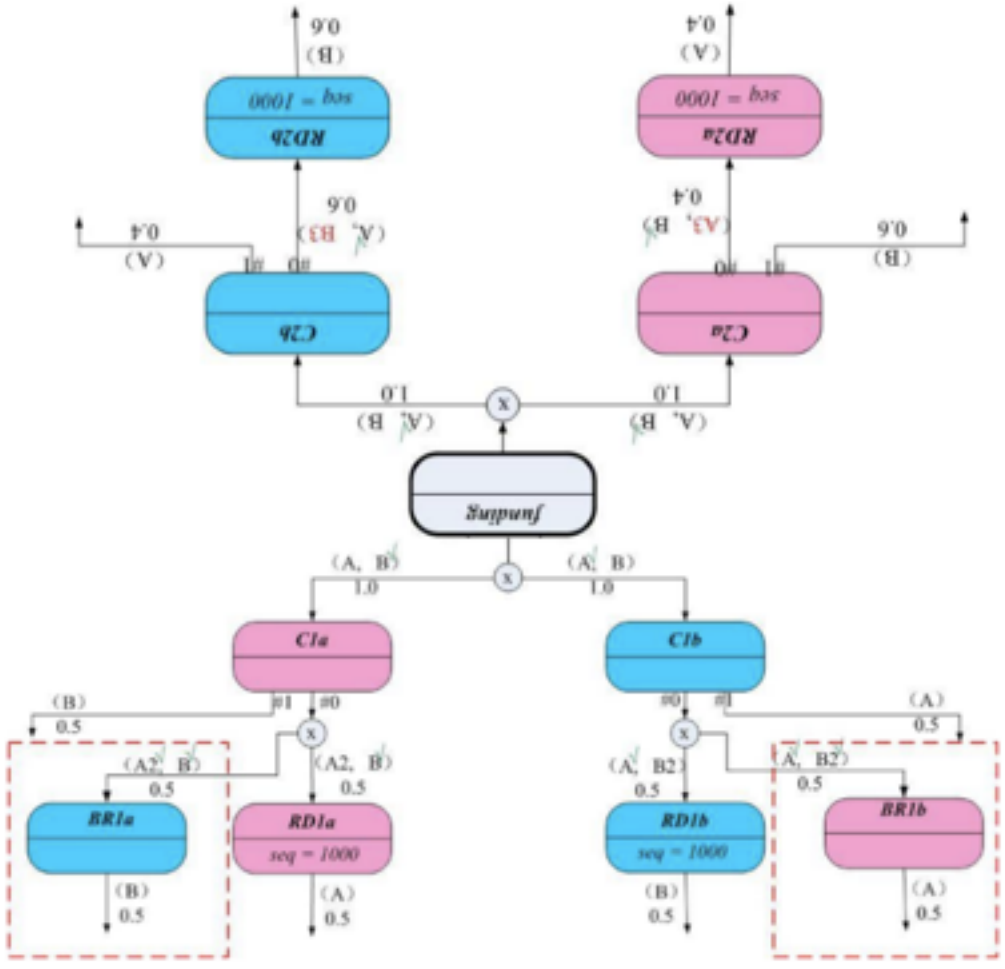
funding父交易是A，B各投入0.5作为共同基金。后续在渠道中的平衡均以此为基础。

渠道建立之后，正如我们上面所提，C1a和C1b都是消费funding父交易的同一个输出，所以它们中只有一个会被发布生效在区块链上。

但不管是C1a还是C1b发布到区块链上，A和B都不会有意见，因为A和B都对对方持有的交易做了认可，给了自己的签名。
A可以随时选择公布C1a，因为在上面的创建过程中，A已经拿到了C1a中B的签名，也可以在C1a之后公布RD1a，因为A也有RD1a中B的签名。B也同理。但也可以选择不公布C1a或C1b，而是在渠道内再次交易更新渠道新的平衡。

对手交易风险处理：
创建渠道过程中，必须在第五步完成之后，再执行第六步，因为funding父交易是2-OF-2交易，必须集齐A，B的签名才能获得解锁交易的输出。如果先发布funding父交易到区块链，但没有拿到对方的签名，并且对方不配合，那将永远无法解锁交易的输出资金将被锁定。

2.更新平衡



现在渠道中需要更新一个新的平衡，B是0.6，A是0.4

步骤如下：

1.类似建立渠道过程的步骤，A构造C2a和RD2a，B构造C2b和RD2b。但B先不给C2a和RD2a签名，A也不给C2b和RD2b签名。

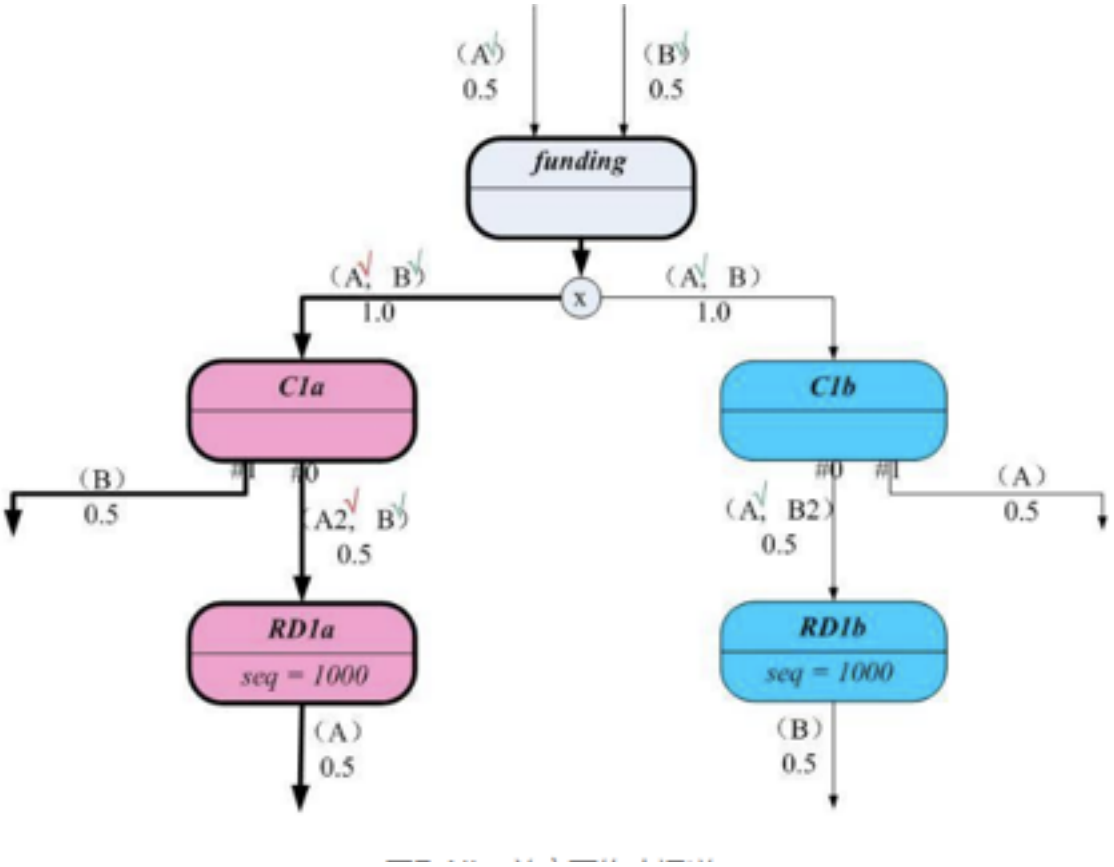
2.B对C1a的本来属于A的资金的第#0输出，构建一个seq=0的BR1a交易，B签名并让A也签名。如果A不对BR1a签名，B就不会对2中的C2a和RD2a签名。A也类似，A对C1b的本来属于B的资金的第#0输出，构建一个BR1b交易，A签名并让B也签名。如果B不对BR1b签名，A就不会对2中的C2b和RD2b签名。这步是相当于双方对作废旧平衡C1a和C1b达成共识，承诺若公布旧平衡就自愿接受惩罚。

3.B对BR1b签名后，B同意对C2a和RD2a签名。A对BR1a签名后，A同意对C2b和RD2b签名。

4.渠道新的平衡建立起来了，C2a和C2b就是新的平衡。旧平衡C1a和C1b已经被作废。

对手交易风险处理：
因为旧平衡C1a(A 0.5 B 0.5)相比新平衡C2a(A 0.4 B 0.6)对A更有利。假如A决定公布旧平衡，A对C1a签名，对RD1a也签名发布。但因为RD1a的seq=1000，它需要等待C1a被发布1000个块之后才能生效。但由于B对网络交易有监控，他发现C1a被违约公布之后，B立即发布BR1a，因为在上面它的解锁签名已经就绪了，所以可以马上收到资金。这样，作为公布旧平衡的承诺惩罚，B收到了原本属于A的资金，违背承诺的一方将损失资金。因此交易双方没有动力做出违约的动作。

3.关闭渠道



渠道建立之后，闪电网络鼓励交易双方维持渠道的长时间建立，避免非必要的频繁建立和关闭。

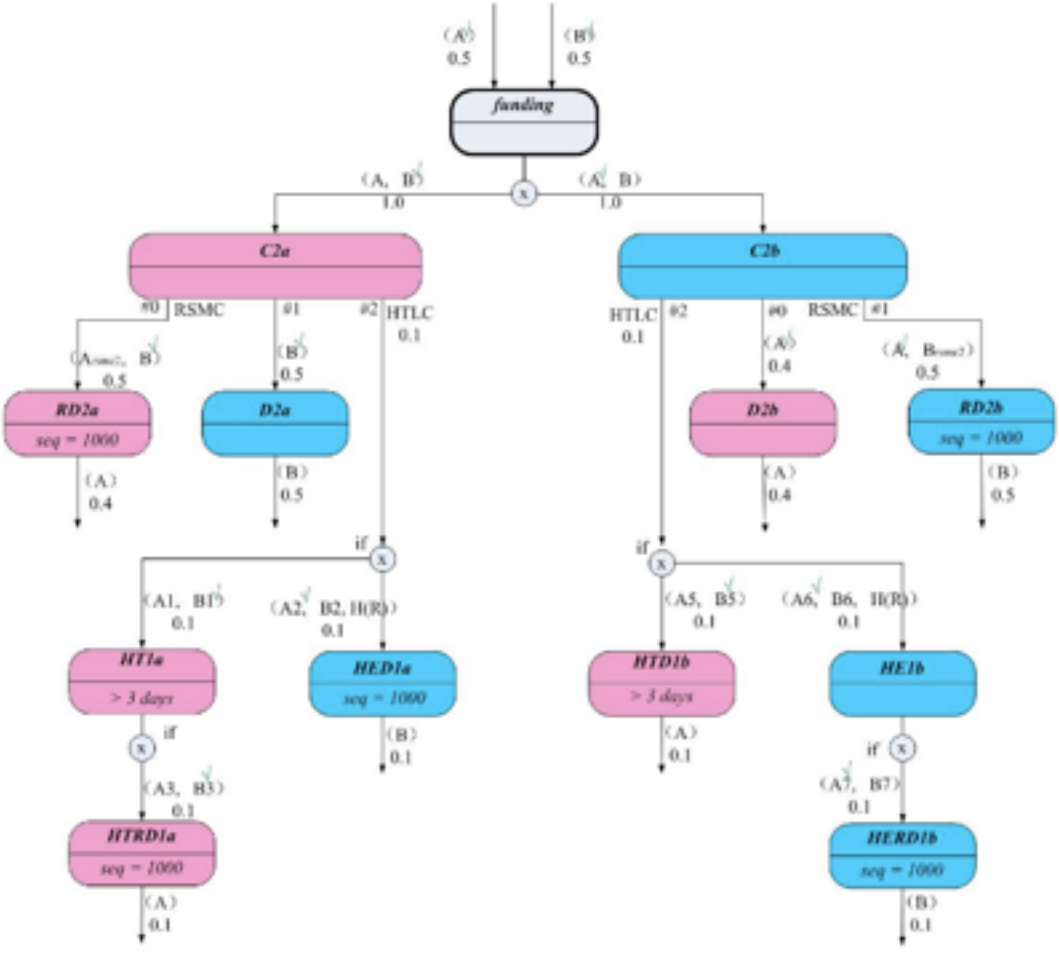
假设目前渠道中只有上图的这些交易关系，这时如果A决定单方面关闭渠道，那A会把C1a发布到区块链上，接着用A2对RD1a签名发布出去，C1a的第#1个输出，将立即支付0.5给B，B可以用私钥立即收到（注意，这个需要B或委托第三方节点对网络交易进行监控才能做到），而第#0个输出RD1a交易，因为seq=1000（回顾CSV BIP），RD1a交易必须在C1a被发布的区块之后1000个区块之后才能被发布生效，即A到1000个区块之后才能收到0.5。因此RD1a中的seq=1000是作为一种对主动关闭渠道方的惩罚措施。

对手交易风险处理：

无论是A发布C1a还是B发布C1b，他们都能收到自己认可的平衡下的资金，不会有任何风险。

四.双向有条件支付渠道(RSMC+HTLC)

1.创建渠道

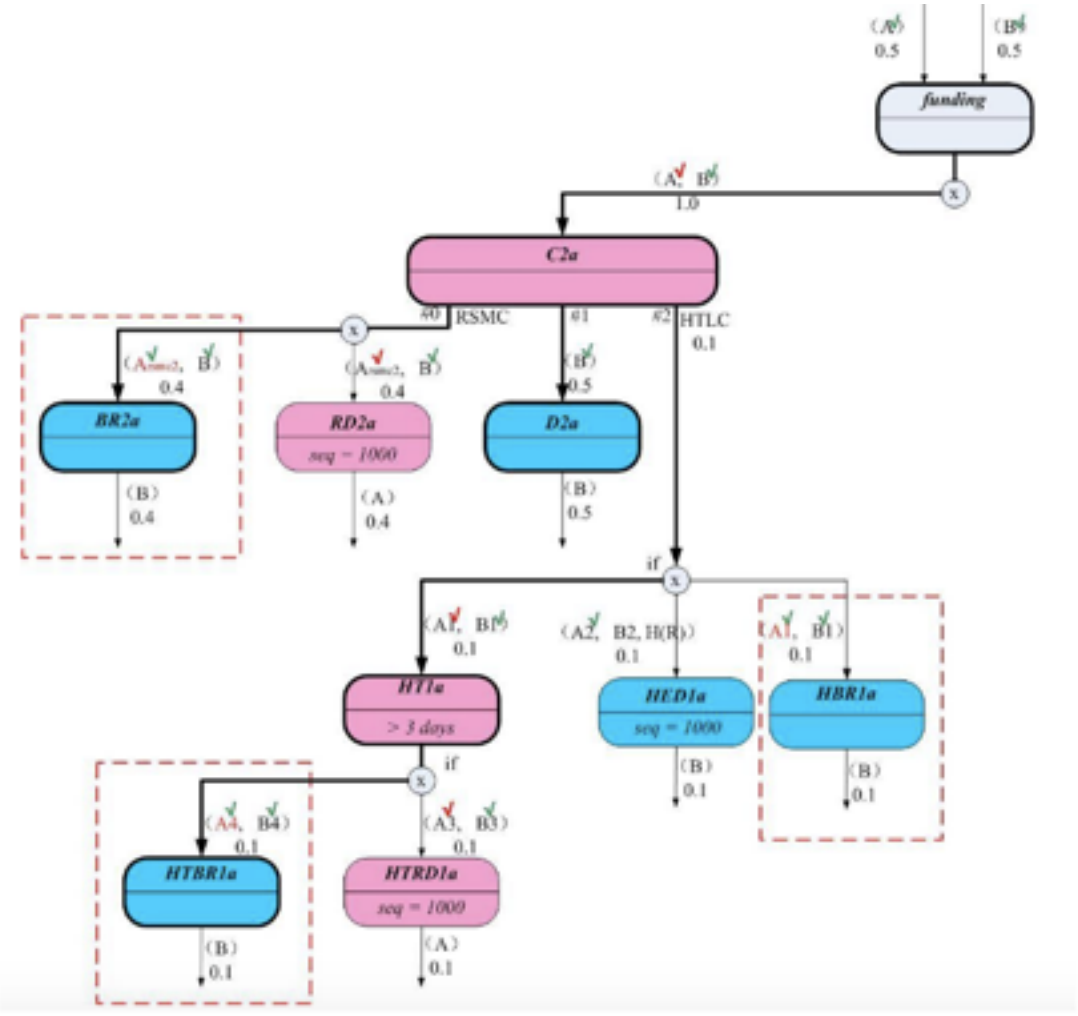


创建渠道的步骤和上面的创建一个无条件双向支付渠道是类似的。只是在创建双方认可的平衡状态的承诺交易的时候增加了一个输出，该输出分支就是所谓的HTLC合同。
譬如A下面的C2a，第#0个输出是 A的0.4资金，第#1个输出是B的0.5，下面的第#2个输出是个HTMC锁住了0.1的资金，下面有三个交易，HT1a，HED1a，HTRD1a。HT1a和HED1a对同一个输出的不同交易是互斥的，注意到他们的不同在于，HED1a的seq=1000,并且解锁条件增加了H（R）,而HT1a的时间锁是>3day。
HED1a的意思是说，如果C2a被A主动公布，但是B能够在3天内提供一个正确的R得到一个正确的H（R），B就能在seq=1000之后得到0.1的资金。否则三天之后，HT1a生效，它的消费交易HTRD1a将在seq=1000之后生效，资金将回到A的手里。
交易双方对所有子交易做好构建签名和互换之后再发布funding 交易。渠道建立之后，无论是哪个交易交易被任何一方发布，因为对方都持有自己认可的交易，都能够做对应的平衡处理，自身利益不会受损。

对手交易风险：
同上面的无条件支付渠道的建立过程。

2.更新平衡

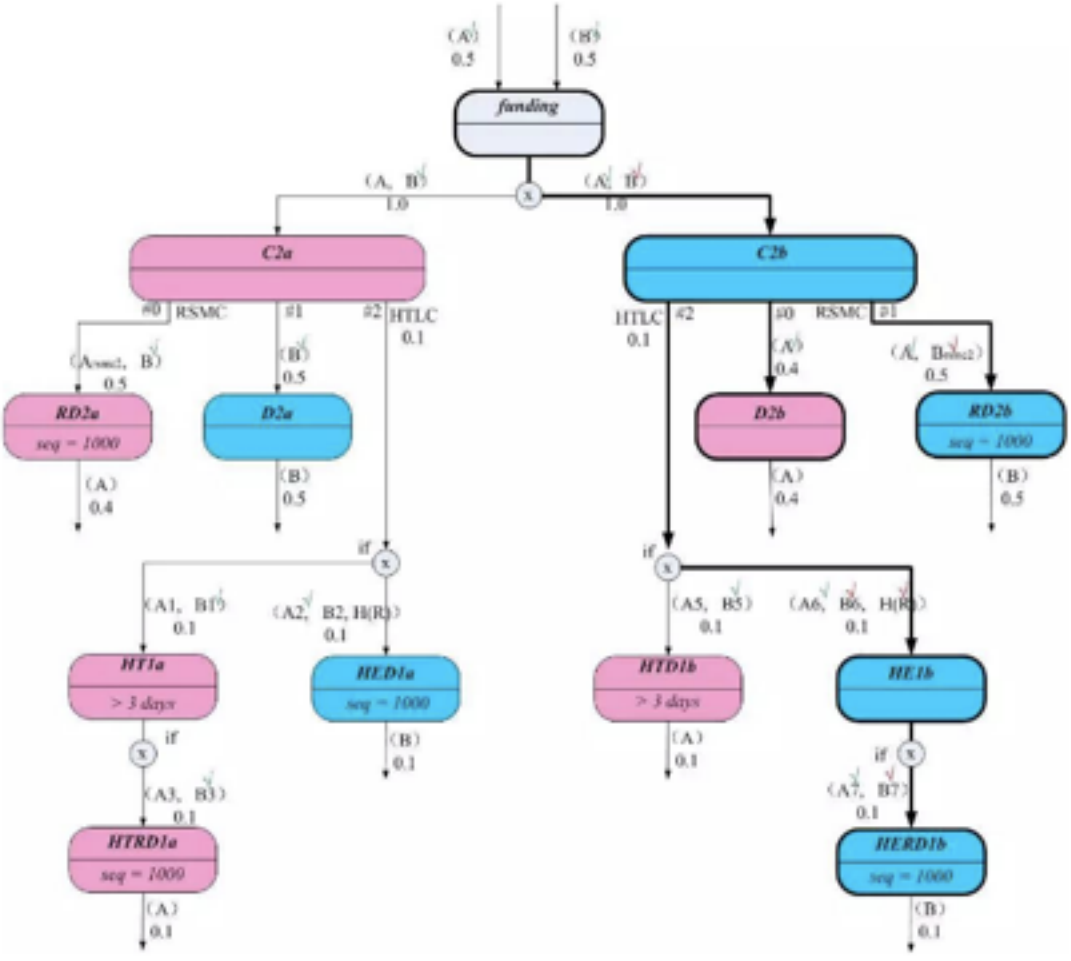
和上面无条件支付渠道的更新承诺交易一样，双方在达成新的平衡C3a和C3b之前，必须先作废旧平衡C2a和C2b，并且同意对违约公布旧平衡达成惩罚约定，作废的方法就是对C2a和C2b补充新的惩罚交易。
新平衡C3a和C3b和上面的渠道建立过程的C2a和C2b是类似的，不再赘述。
重点看下对旧平衡补充的惩罚交易。下图是对C2a补充的惩罚交易。



交易对手风险：

假如A要违约公布旧平衡C2a，B监测到了A违约。因为各个违约惩罚交易的签名已经就绪。B立即发布BR2a，拿到原来属于A的0.4,并且立即发布D2a，也拿到了0.5。HTLC里面的0.1，如果A公布HT1a，因为A的HTRD1a的seq=1000必须等待，B就在HTRD1a生效之前公布HTBR1a拿到0.1资金，如果A不公布HT1a，B就发布HBR1a立即拿到0.1资金。这样B拿到的全部的资金，主动违约的一方A没有任何好处。因此理智的交易方不会发布旧平衡。

3.关闭渠道



上面的渠道建立之后，如果B拥有正确的R，并且决定关闭渠道拿回自己的资金。
B可以发布C2b交易，接着发布RD2b，等待seq=1000之后，B拿到了0.5资金，这时候A监测到B发布了C2b交易，A也立即发布D2b交易，拿回了0.4资金。
如果B接着提供了R产生了正确的H（R）,那HE1b被发布，接着HERD1b被发布，等待seq=1000之后，B拿到了0.1资金。如果B无法提供正确的R，那HTD1b会被发布，A将拿到0.1资金。
A要是主动发布C2a，同理也是一样。

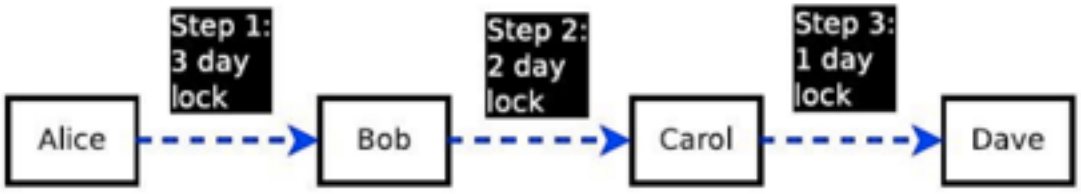
对手交易风险处理：
无论是A发布C2a还是B发布C2b，他们都能收到自己认可的平衡下的资金，至于HTLC锁定的资金，只要接收方能提供正确的R就能收到资金，如果不能资金将退回发起方，双方都不会有任何风险。

五.多方支付渠道链路

有了RSMC和HTLC的支持。当闪电网络里面的参与方足够多的时候，只要双方同意，渠道可以随意发起建立。再通过HTLC的条件特性作为授权条件，把渠道参与节点逐个连结，形成链路支付渠道。当A需要向D付款的时候，虽然A和D没有直接建立支付渠道，但A可以通过中间节点B，C形成链路向D支付。

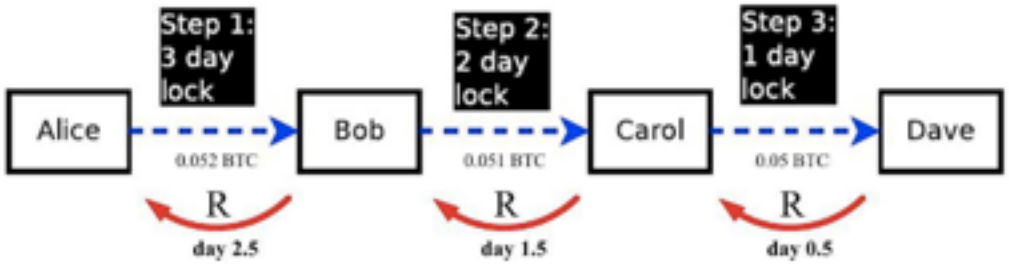
当渠道足够多的时候，就建立起了覆盖全网的完整的支付网络。

1.链路的建立



链路建立的过程利用了HTLC的timelocks的递减，逐跳递减。至于中间节点的选择可能是从第三方提供服务来选择，A看不到C，D可能也不知道B。
建立前，D给A提供一个H(R),R只有D才知道，A依此作为交易的解锁条件。
A与B建立一个HTLC，约定3天内，如果B能提供正确的R就可以拿到钱，否认3天后钱自动退回给A。
B与C也建立一个类似的HTLC，只是timelocks变成2天。
C和D也类似。

2.资金的拉动



D在和C的HTLC约定时间内提供了正确的R，D顺利的从C拿到了钱。C也因此得到了正确的R，C也从B处拿到了钱。B也一样从A处拿到了钱。
这个过程，B和C都收取了一小部分的手续费。这个是在HTLC里面约定好的。
因此一个链路上的资金是由资金接收方D提供的R拉到的，因为R的存在，资金可以按照预定的方向流动到目标方。

3.链路的关闭

1.接收方提供了正确的R，沿途拉动资金，最终成功收到资金，链路中各方也收到了手续费。HTLC从后往前，逐跳关闭，链路关闭。
2.接收方没有提供正确的R，HTLC从后往前，逐跳到期，最终链路关闭。

4.资金的安全保证

因为HTLC的timelocks是逐跳递减，只要资金接收方提供正确的R，链路中各方都能收到资金。如果没有正确的R，HTLC也会逐个到期，资金也会从后逐个退回。

六.其他问题?

1.交易费用

考虑两个点：
1.多方支付渠道链路中，中间节点需要收取手续费用，如果链路很长中间节点很多，有可能手续费用相对本身的资金价值变得昂贵。
2.RSMC和HTLC中对交易对手的交易的公布行为需要进行监控，这可能需要依赖第三方来实现，第三方可以以此来收取费用。

2.密钥存储与交换

3.中间节点的在线问题

多方支付渠道链路中的中间节点的掉线状况与不合作的状况是一样的。都会导致链路断掉等待时间周期到期重新路由。

4.重新路由带来的时间价值损耗

对多方支付渠道链路的交易发起方而言，只要有任意中间节点的不合作就会导致必须等待时间锁的到期才能退回资金，然后再重新选择新链路进行路由。在此期间有资金的时间价值损耗。另外也需要考虑时间延期对支付行为的影响。因此HTMC的timelocks选择需要与资金价值和支付行为影响进行权衡考虑。

5.市场波动，渠道中资金价值波动

支付渠道的建立最初是通过交易各方投入资金建立基金建立的。在渠道关闭之前，资金一直在渠道内不断变更平衡。
在此期间，区块链上的资金汇兑可能会发生变化，汇兑的波动会带来渠道中沉淀资金价值的波动。类比现实世界里面的银行，沉淀资金有利息收益。但在无中心的闪电网络里面，可能要交易方自己承担汇兑损失。

6.对第三方的依赖

为了资金安全，交易方在对监控交易对手对过期承诺交易的发布上，可能需要依赖第三方节点对网络交易的监控。因为SegWit的实现，对特定交易的监控是完全可以做到的。
另外多方支付渠道链路的建立也需要第三方提供中间节点的选择或者自动匹配。或者还有其他场景需要依赖第三方。
所以这些对第三方节点的依赖是否会形成中心化或多中心化的趋势，这将有可能对去中心化的区块链形成挑战。

五.开源的闪电网络实现项目

https://www.blockchain.com/thunder/
代码build步骤
安装java8
执行./build.sh 即可