General State Channel

Shu Dong
12/5/2018

Payment Channel

Revocable Sequence Maturity Contract (RSMC)

A special output script within a bitcoin transaction that allows a sender to be able to revoke a payment

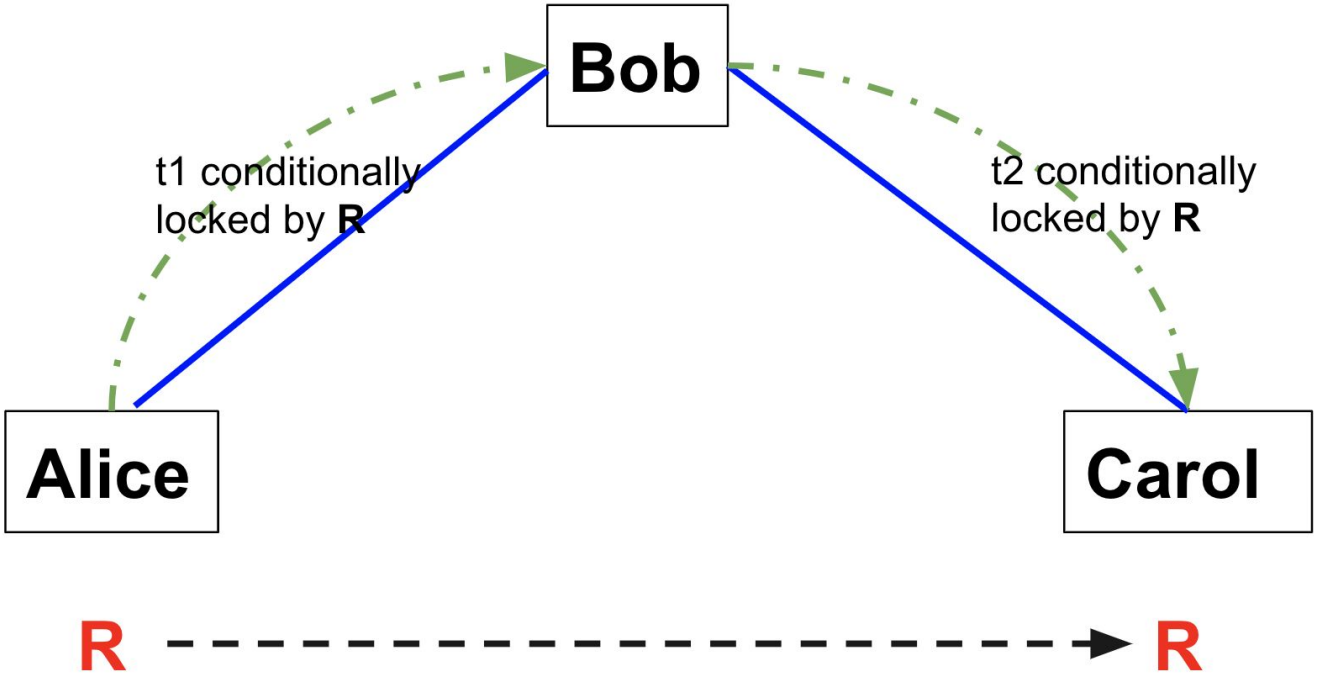
Hash Time-Locked Contracts (HTLC)

Use hashlocks and timelocks to require that the receiver of a payment either acknowledge receiving the payment prior to a deadline by generating cryptographic proof of payment or forfeit the ability to claim the payment, returning it to the payer

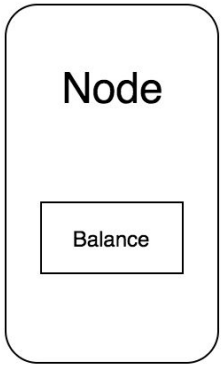
The Netting Channel Smart Contract

Check our previous work to know details of *Lightning Network* and *Raiden Network*

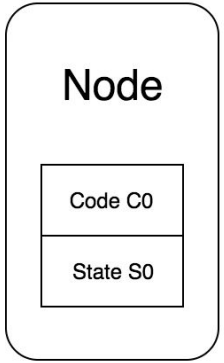
Virtual Payment Channel



State Channel



No Smart Contract



With Smart Contract

Contract Instance
= code + storage

The On-Chain Smart Contract

Step 1: Deploy the Channel Smart Contract Between The Two Parties

- Users should be able to
  - Add/Withdraw the money in the channel
  - Lock/Unlock the money upon requests/state proof
  - Understand the state and able to run the code

Step 2: Users in the channel have to deposit some coins in the smart contract

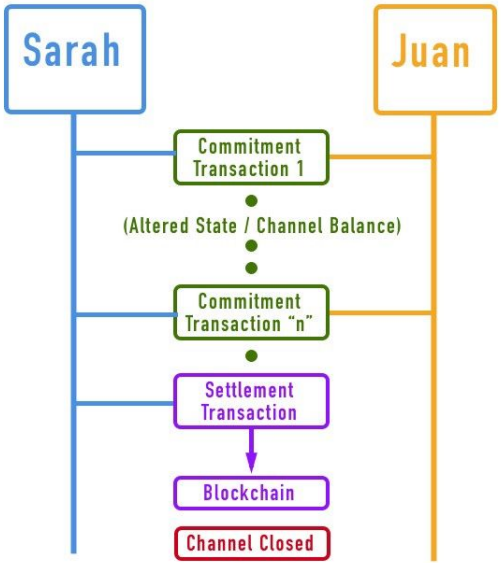
What is the State Proof?

- Latest both agreed state Si
- The coin distribution should be contained in Si

State Channel

State channels are basically two-way pathways opened between two users that want to communicate with each other in the form of transactions.

Not only payments but also smart contracts!



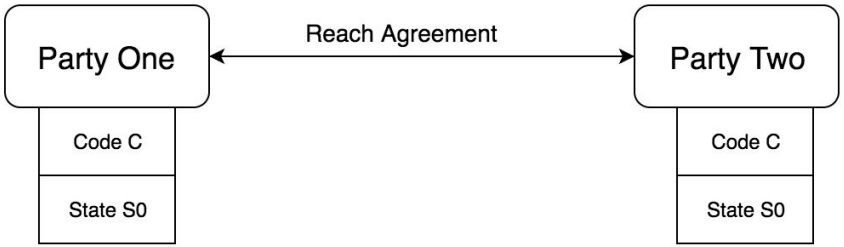
Let's consider the case when

- C1: They are honest
- C2: The off-chain smart contract has been deployed on-chain.
- C3: The functionality of off-chain smart contract doesn't change
- C4: Payment channel is deployed between end users directly
- C5: There are only two end users in the channel

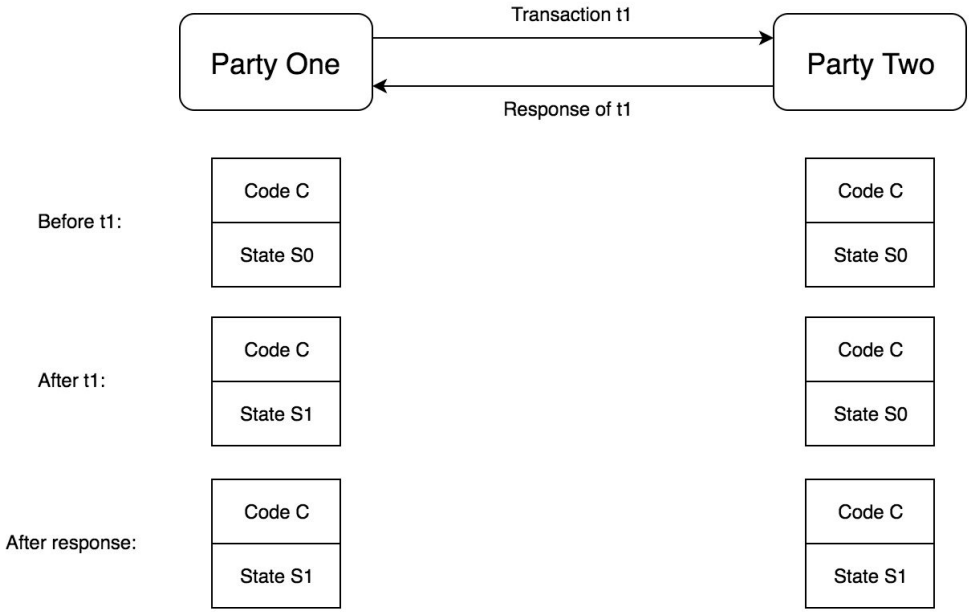
We will relax the restrictions one by one

Off-Chain: Set Up

Both parties have to agree on a **initial code C0** and a **initial state S0.** The coin distribution is contained in the state.



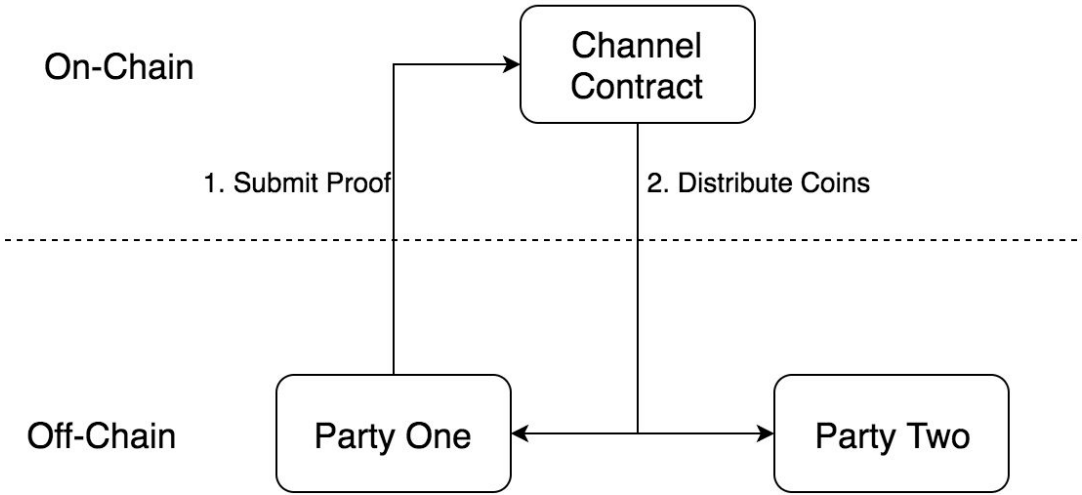
Off-chain: Smart Contract Execution



What does the transaction look like?

- Caller: P
- Function call: f(m)
- Future State: S_f
- Version Number: v

Settlement


On-Chain
Channel Contract
1. Submit Proof
2. Distribute Coins
Off-Chain
Party One
Party Two

Relax Condition C1

~~**C1: They are honest**~~

Assumptions:

- Reliance on Blockchain Liveness
- Reliance on Participant Availability.
- Absence of External Incentives.
- Absence of Software Errors

Griefing: when a user does something unexpected in a system, for no personal reward, just to harm someone else

Relax Condition C1

A sends B a transaction but B doesn't respond...

- Case 1: B is available but didn't receive the transaction
- Case 2: B is not available
- Case 3: B just refuses to respond
  - Case 3.1 B is griefing, in this case B holds a newer state
  - Case 3.2 A sends B a invalid state.

A needs to submit the latest state proof on-chain

- If B doesn't respond in challenge period, B will lose the deposit.
- If B respond in challenge period with newer state, it falls back to a on-chain game

Relax Condition C1

A submits a stale state..

- A definitely is cheating
  - B is not available
  - B is available

On-chain contract starts a challenge period:

- If B doesn't respond in challenge period, B will lose the deposit. The stale state will be finalized.
- If B respond in challenge period and provided a newer state, A will be punished and it falls back to a on-chain game

Initiator always pays the fee!
Submitting stale state(if proved) will be punished!

Relax Condition C1

Solutions of participants availability

- Lighting Network: Watch Tower
- PISA: Decentralized Watch Tower
- Celer Network: Guardian State Network

Relax Condition C2

~~**C2: The off-chain smart contract has been deployed on-chain.**~~

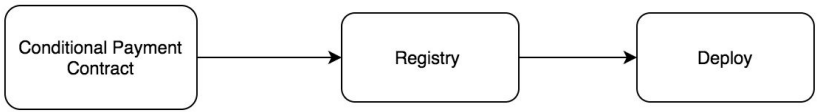
It doesn't have to! We can only deploy the off-chain contract to on-chain once there is a conflict.

Problem

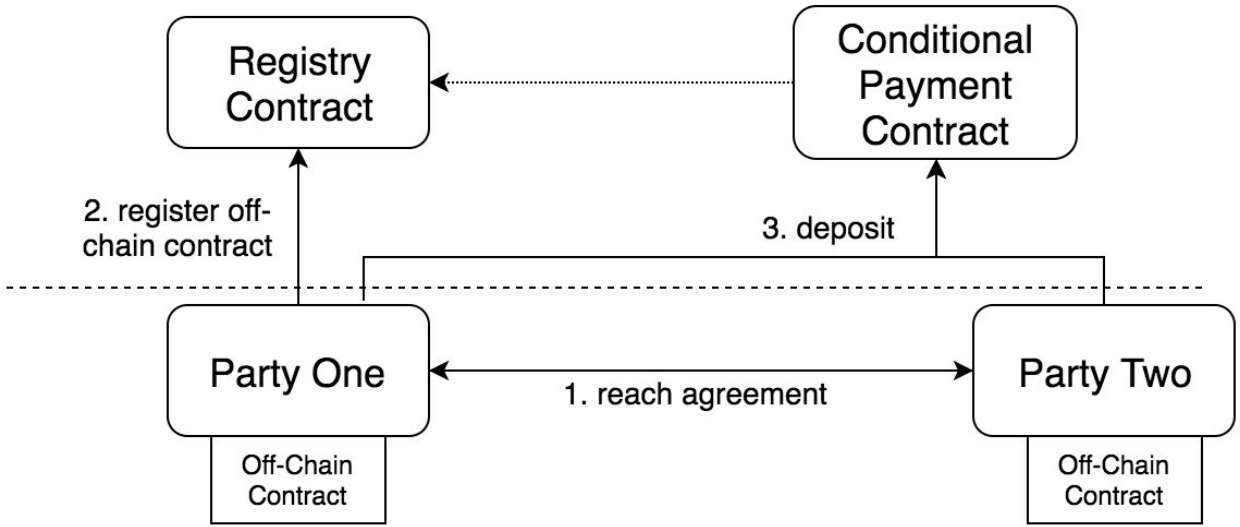
The state-deposit-holder does not know in advance which address the counterfactually instantiated contract will have, but needs to make commitments that involve this address.

Steps

1. Decouple the conditional payment logic with the application logic
2. Create a counterfactual instantiated contract off-chain
3. Register the off-chain contract by assigning it an on-chain address

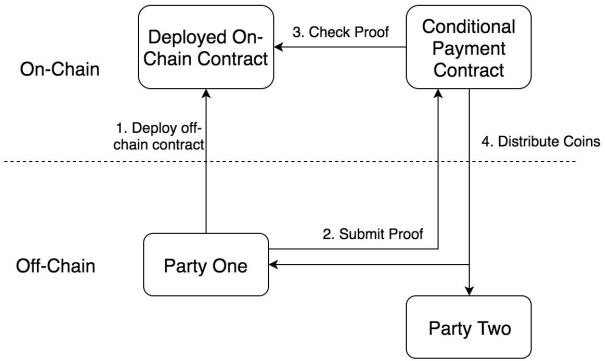


Relax Condition C2: Revisit Set Up

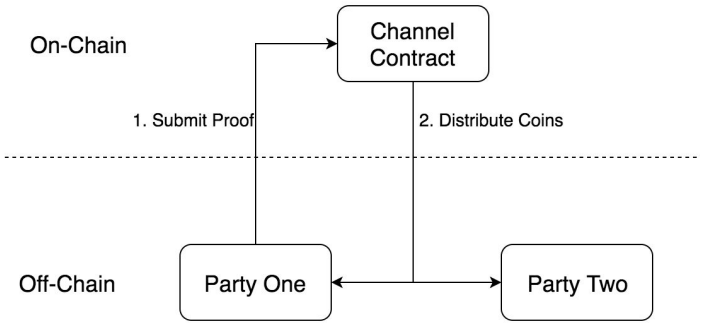


Relax Condition C2: Revisit Settlement

~~**C2: The off-chain smart contract has been deployed on-chain.**~~



New Settlement



Previous Settlement

Relax Condition C3

~~**C3: The functionality of off-chain smart contract doesn't change**~~

This is very easy once we relaxed C2

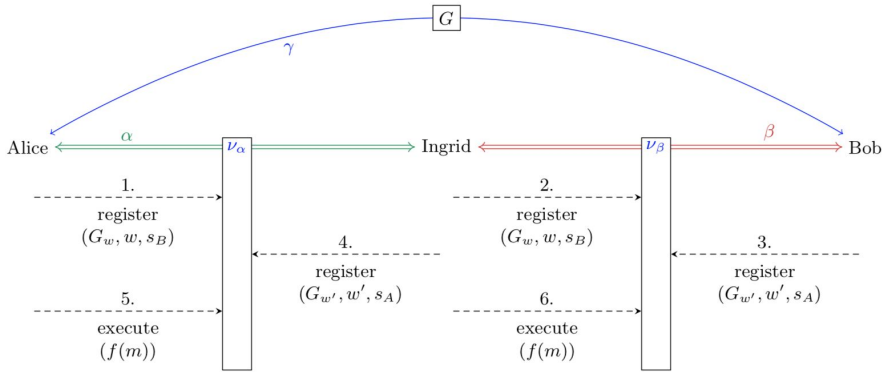
Steps

1. Create a new counterfactual instantiated contract off-chain
2. Register the new off-chain contract by assigning it an on-chain address
3. Put deposit in the state deposit holder(i.e. the conditional payment contract)

We don't need to deploy any new contract on-chain if there is no conflict!

Relax Condition C4

~~**C4: Payment channel is deployed between end users directly**~~



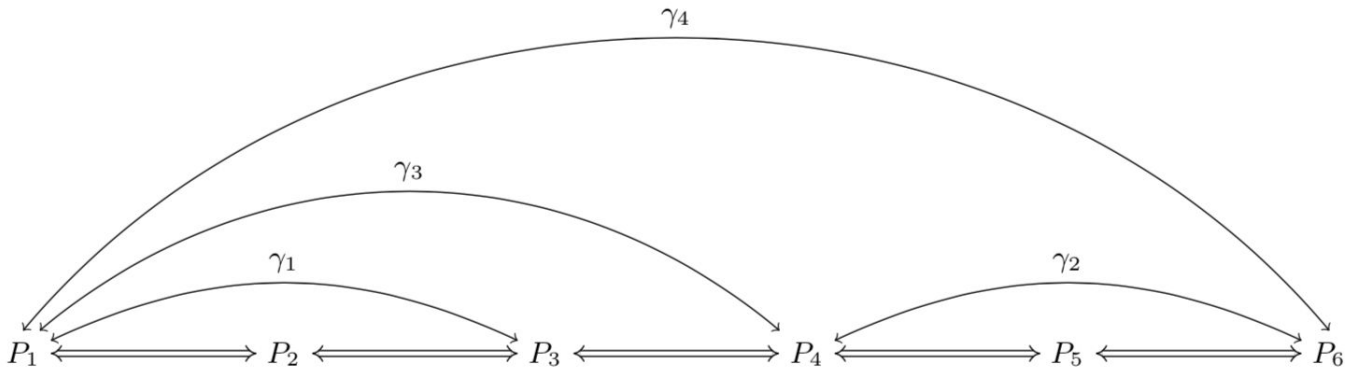
Ledger Channel vs **Virtual Channel**

1. Once virtual channel is created, the end users don't need to communicate everything through the third party but just balance transfers. A.k.a. The ledger channels only serve as a payment channel

2. The underlying ledger channels can support different coins. The third party node can do the exchange.

Multi-hop Virtual Channel Creation

We follow a modular recursive approach where virtual state channels are built recursively on top of ledger or other already constructed virtual state channels.



Multi-hop Virtual Channel Creation

How Counterfactual Addressing helps virtual channel set up?

Without Counterfactual Addressing, a dispute in a multihop virtual channel must be (trustlessly) resolved through the intermediary**.** Thus each intermediary node need to install contract in worst case. i.e. If there is an Alice-Bob virtual through Ingrid with a lot of functionality instantiated, Alice can grief Ingrid by forcing her to play out all the moves in the meta channel

With Counterfactual Addressing, Alice can only do so to Bob, and Alice can only grief Ingrid for the value of the fees needed to instantiate up to the payment object.

Relax Condition C5

~~**C5: There are only two end users in the channel**~~

Total consent

State channels are further differentiated from Plasma in that they proceed by total consent. Any change of state within a state channel requires explicit cryptographic consent from all parties designated as “interested” in that part of the state. Because of this continual need for obtaining and providing direct consent, state channels depend heavily on the records and responsiveness of state channel participants (or their delegates)

Total consent is a high cost consensus protocol

- **Total consent:** Reach agreement on all members
- **PBFT:** Reach agreement on ⅔ members
- **PoW:** Reach agreement on 50% members (assuming the each node has the same computation power)

Thank you!

Shu Dong
12/05/2018