Multi-client Partially Non-Interactive and Instantaneous One-way Payment Channel for Ethereum

August 3, 2018 – DRAFT

Thomas Shababi[1], Joël Gugger[1], and Daniel Lebrecht[1]

TrueLevel SA, Neuchâtel, Switzerland
{tom, joel, d}@truelevel.io

Abstract. Ethereum is a distributed computing platform and operating system featuring smart contract functionality. Transactions are faster than other blockchain but not instant and each of them cost some "gas". This gas is used to quantify the amount of fee to pay for computation.

Keywords: Crypto-currencies, Ethereum, Payment channels

1 Introduction

1.1 The value of unidirectional channels

2 Multi-client payment channel

Partially non-interactive multi-client payment channel is composed of clients $c \in \mathcal{C}$ and one provider $\mathcal{P}$. $\mathcal{C}$ is the set of all clients registred in the multi-client payment channel. Clients can send money to the provider through the channel $c \to \mathcal{P}$ but cannot recieve into the channel $\mathcal{P} \not\to c$.

A payment channel is by definition a structure composed of two layer of states. The first layer of states is registred to the blockchain, i.e. "on-chain" or contract states $\mu \in M$, and the second layer of states is kept "off-chain" between the participants, i.e. channel states $\sigma \in \Sigma$.

It is possible to transition from a state σ to another state σ', we denote state transition with $\to$, i.e. $\sigma \to \sigma'$. Modifiers $\omega \in \Omega$ are used to create transition on channel states $\sigma \in \Sigma$. A transition in Σ depends on modifiers and contract states. We denote a transition from $\sigma, \sigma' \in \Sigma$ that depend on $\mu \in M$ as

$$\sigma \xrightarrow{\mu+\omega} \sigma'$$

Messages allow transitions between two states $\mu, \mu' \in M$, also denoted $\mu \to \mu'$. A message m can be "applied" to a state μ, we denote this operation with $m(\mu)$, i.e. $\mu \xrightarrow{m} \mu'$. A message is created based on a channel state

$$\sigma \implies m$$

2.1 Contract states

Each client $c \in \mathcal{C}$ in the contract is defined by an on-chain state $\mu \in M$. States $\mu \in M$ are a tuple $(I(c), B(c))$.

2.2 Contract state variables

We define variables to acces the the contract state $\mu \in M$ for a client $c \in \mathcal{C}$.

Current on-chain index, $I(c)$ For each client the current index i must be retreivable, $\forall c \in \mathcal{C}, \quad \exists I(c) \in \mu >= 0$. The index for a client must start from 0.

Current on-chain balance, $B(c)$ For each client the current balance amount must be retreivable, $\forall c \in \mathcal{C}, \quad \exists B(c) \in \mu >= 0$.

2.3 Channel states

Each client $c \in \mathcal{C}$ is defined by his channel state $\sigma \in \Sigma$. States $\sigma \in \Sigma$ are composed of: (i) a validity index, (ii) the on-chain balance of the client, (iii) the total deposit of the client, and (iv) the total owned by the provider. We denote σ as

$$(i, B(c), D(c), \sum c \to \mathcal{P})$$

2.4 Channel state variables

We define variables to represent: (i) the lifetime of a single channel in the multi-client channel architecture $L(c \to \mathcal{P})$, (ii) the total amount deposited for a client over the lifetime $D(c)$, and (iii) the total amount own by the provider over the lifetime $\sum c \to \mathcal{P}$.

The amount available into a single channel for a client $c \in \mathcal{C}$ is computed with $A(c \to \mathcal{P}) = D(c) - \sum c \to \mathcal{P}, \quad B(c) >= A(c \to \mathcal{P})$.

2.5 Messages

Messages $m \in \mathcal{M}$ are exchanged between clients $\mathcal{C}$ and provider $\mathcal{P}$. A message is related to one and only one relation in $\mathcal{C} \times \mathcal{P}$.

Minimal message components A message $m \in \mathcal{M}$ between a client c and the provider $\mathcal{P}$ has at least tree components: (i) a validity index, (ii) the total of deposit, and (iii) the total own by the provider. We denote m as the triple

$$(i, D(c), \sum c \to \mathcal{P})$$

3 State transitions

3.1 Contract state, $\overrightarrow{M}$

Contract states $\mu \in M$ can transitioned $\mu \to \mu'$ because of a message $m \in \mathcal{M}$ or an external event e.

Settlement $(i, B(c)) \xrightarrow{m} (i{+}1, B(c)\downarrow) \mid m = (i{+}1, D(c), \sum c \to \mathcal{P})$. It is worth noting that with this rule it is not possible to settle a zero amount because the balance amount $B(c)$ must go down!

$$\begin{aligned} O(\mathcal{P}) &= B(c) - (D(c) - \sum c \to \mathcal{P}) \\ &= B(c) - A(c \to \mathcal{P}) \end{aligned}$$

$O(\mathcal{P})$ is the amount owned by the provider at the settlement time. The provider can settle the amount in the balance minus the remaining funds of the client c.

$$\begin{aligned} B(c)\downarrow \ &= B(c) - O(\mathcal{P}) \\ &= A(c \to \mathcal{P}) \end{aligned}$$

The new client balance is the balance minus the settlement or directly the remaing funds of the client.

Full refund $(i, B(c)) \xrightarrow{m} (i+1, 0) \mid m = (i+1, D(c), \sum c \to \mathcal{P})$. It is a full refund if the remaining funds in the channel for the client c $A(c \to \mathcal{P})$ are equal to the current balance $B(c)$

$$\begin{aligned} B(c) &\stackrel{?}{=} D(c) - \sum c \to \mathcal{P} \\ &\stackrel{?}{=} A(c \to \mathcal{P}) \end{aligned}$$

Partial refund $(i, B(c)) \xrightarrow{m} (i+1, 0) \mid m = (i+1, D(c), \sum c \to \mathcal{P})$. It is a partial refund if the remaing funds in the channel is smaller than the current balance

$$\begin{aligned} B(c) &\stackrel{?}{>} D(c) - \sum c \to \mathcal{P} \\ &\stackrel{?}{>} A(c \to \mathcal{P}) \end{aligned}$$

Top up $(i, B(c)) \xrightarrow{e} (i, B(c)\uparrow)$.

Invalid transition $(i, B(c)) \rightarrow (i, B(c)\downarrow)$. This transition is invalid because the balance is changed without incrementing the validity index i.

Invalid transition ??? $(i, B(c)) \rightarrow (i{+}1, B(c)\uparrow)$??? invalidates valide pending indexes i ???

3.2 Channel state, $\overrightarrow{\Sigma}$

Channel state transitions $\sigma \xrightarrow{\mu+\omega} \sigma' \in \Sigma$, i.e. applying a modifier ω on the current state μ to result on σ'.

Initialize channel state $\sigma = (0, B(c), B(c), \texttt{pay})$

4 Protocol

The protocol describe valid and invalid actions for clients and the provider, rules for state transition, and messages.

Client actions A client can send money through the channel, authorize limited payment, and refund his money.

Provider actions The provider can settle a single channel, and "settle and close" a single channel.

4.1 Send money, $c \rightarrow \mathcal{P}$

Two messages are generated: (i) a refund message for the client, and (ii) a settle message for the provider.

Refund message, m_r Refund message m_r must allow the client to claim his remaing funds $A(c \rightarrow \mathcal{P})$ and transfer them after some locktime if the provider does not dispute the refund.

Settle message, m_s

5 Acknowledgement

Loan Ventura, Thomas Roulin and Nicolas Huguenin are acknowledged for their helpful contribution and comments during the completion of this work.