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

August 6, 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

2 Multi-client payment channels

Partially non-interactive multi-client payment channels are 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 through 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 $\mu \in M$ and $\omega \in \Omega$ 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

$$(I(c), B_\mu(c))$$

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, $\boldsymbol{I(c)}$ 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, $\boldsymbol{B_\mu(c)}$ For each client the current balance amount must be retreivable, $\forall c \in \mathcal{C}, \quad \exists B_\mu(c) \in \mu >= 0$.

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 latest observed on-chain balance of the client, (iii) the total deposit of the client, and (iv) the total owned by the provider. We denote a state $\sigma \in \Sigma$ as

$$(i, B_\sigma(c), D(c), \sum c \rightarrow \mathcal{P})$$

2.4 Channel state variables

We define variables to represent: (i) the lifetime of a single channel in the multi-client channel architecture, (ii) the total amount deposited for a client over the lifetime, and (iii) the total amount own by the provider over the lifetime.

Channel lifetime, $\boldsymbol{L(c \rightarrow \mathcal{P})}$ It exists one lifetime and only one per elements in $\mathcal{C} \times \mathcal{P}$ and begin when the first deposit is made, i.e. one lifetime per client $c \in \mathcal{C}$.

Total deposit, $\boldsymbol{D(c)}$ Total deposit of a client represent the total amount recieved from the begin of the lifetime. Each top up increase the total deposit.

Total sent, $\boldsymbol{\sum c \rightarrow \mathcal{P}}$ Total amount sent to the provider by a client represent the sum of all transaction since the begining of the lifetime. Each transaction increase the total sent.

Minimal available amount, $\boldsymbol{A_m(c)}$ Minimal amount available for a client is computed with the latest channel state σ. Without quering the contract state μ it is impossible to know if a bigger is now available.

Client available amount, $A(c)$ The full amount available into a single channel for a client $c \in \mathcal{C}$ is computed with

$$A(c) = D(c) - \sum c \to \mathcal{P} + (B_\mu(c) - B_\sigma(c)), \quad B_\mu(c) \geq A(c)$$

It is woth noting that $B_\mu(c) - B_\sigma(c)$ is added to the difference of total deposit and total sent in case of on-chain changements, like top up.

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 element in $\mathcal{C} \times \mathcal{P}$.

Minimal message A minimal message $m \in \mathcal{M}$ between a client c and the provider $\mathcal{P}$ is composed of four components: (i) a validity index, (ii) the lastest observed balance, (iii) the total of deposit, and (iv) the total own by the provider. We denote m as the triple

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

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 external events.

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

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

$O(\mathcal{P})$ 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.

$$\begin{aligned} B_\mu(c)\downarrow &= B_\mu(c) - O(\mathcal{P}) \\ &= B_\mu(c) - (B_\mu(c) - (D(c) - \sum c \to \mathcal{P} + (B_\mu(c) - B_\sigma(c)))) \end{aligned}$$

The new client balance is the current contract balance minus the the amount due to the provider.

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

$$\begin{aligned} B_\mu(c) &\overset{?}{=} A(c) \\ &\overset{?}{=} D(c) - \sum c \to \mathcal{P} + (B_\mu(c) - B_\sigma(c)) \end{aligned}$$

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

$$B(c) \overset{?}{>} A(c)$$

In that case the provider amount is

$$O(\mathcal{P}) = B_\mu(c) - A(c)$$

Top up $(i, B_\mu(c)) \xrightarrow{e} (i, B_\mu(c) \uparrow)$. Top up increase the contract balance for a client $c \in \mathcal{C}$. Validity index i must not be incremented during the top up.

Invalid settlement $(i, B_\mu(c)) \to (i, B_\mu(c) \downarrow)$. This transition is invalid because the balance is decreased without incrementing the validity index i. The current set of transactions must be invalidate after the settlement.

Invalid top up $(i, B_\mu(c)) \to (i+1, B_\mu(c) \uparrow)$ Increasing validity index i while increasing the contract balance invalidates the set of transaction i. The current set of transactions must be invalidate only during a settlement.

Invalid validity index $(i, B_\mu(c)) \to (i \downarrow, B_\mu(c) \downarrow\uparrow)$ Decreasing the validity index i is always invalid. Validity index must not be decreased.

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

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

4 Example

	ω	μ	m	σ
	$c \to \mathcal{P}$	$(i, B_\mu(c))$	$(i, B_\sigma(c), D(c), \sum)$	$(i, B_\sigma(c), D(c), \sum)$
init		$(0, 10)$		$\emptyset$
transact	+1		$(1, 10, 10, 1)$	$(0, 10, 10, 1)$
transact	+1		$(1, 10, 10, 2)$	$(0, 10, 10, 2)$
settle		(1,8)		
transact	+2		$(2, 8, 10, 4)$	$(1, 8, 10, 4)$
transact	+2		$(2, 8, 10, 6)$	$(1, 8, 10, 6)$
top up		(1,18)		
transact	+1		$(2, 18, 20, 7)$	$(1, 18, 20, 7)$
settle		(2,13)		

Table 1. State transitions during channel lifetime

	ω	μ	m	σ
	$c \to \mathcal{P}$	$(i, B_\mu(c))$	$(i, B_\sigma(c), D(c), \sum)$	$(i, B_\sigma(c), D(c), \sum)$
init		$(0, 5)$		$\emptyset$
transact	+3		$(1, 5, 5, 3)$	$(0, 5, 5, 3)$
settle		(1,2)		
transact	+1		$(2, 2, 5, 4)$	$(1, 2, 5, 4)$
top up		(1,4)		
settle		(2,3)		

Table 2. Settlement after top up