Virtual Channels and Rebalancing in State Channel Networks

Arya Stark

Introduction

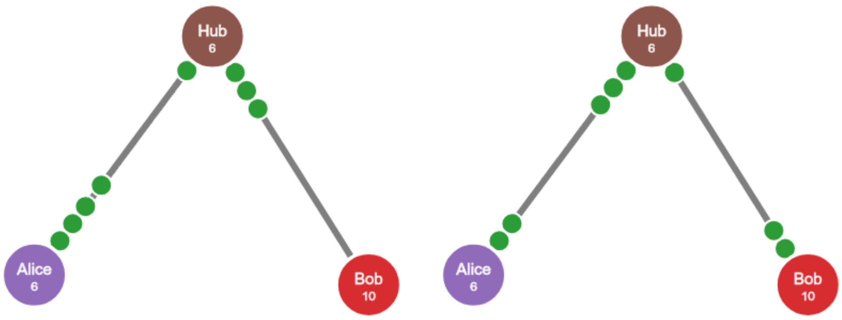
This research note was written because there are some unsolved problems in designing state channel networks that are not well-known or being worked on. These problems reveal themselves when we consider how to build channel networks capable of expressing agents' preferences over network topology (payment capacity) and capital costs, and minimizing blockchain fees.

Relevant Prerequisite Literature

This section goes through relevant existing literature. Note that even if you know all this, many definitions will be used in later sections.

Ball-and-bead model of PCN

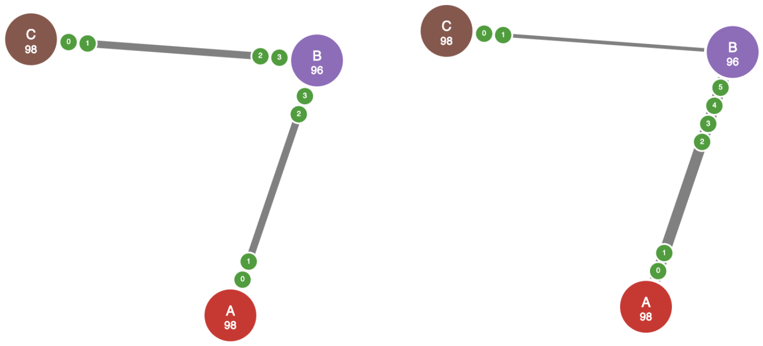
In this model, edges represent two-party channel between the endpoints, and a party's balance within the channel is represented by beads near to them.



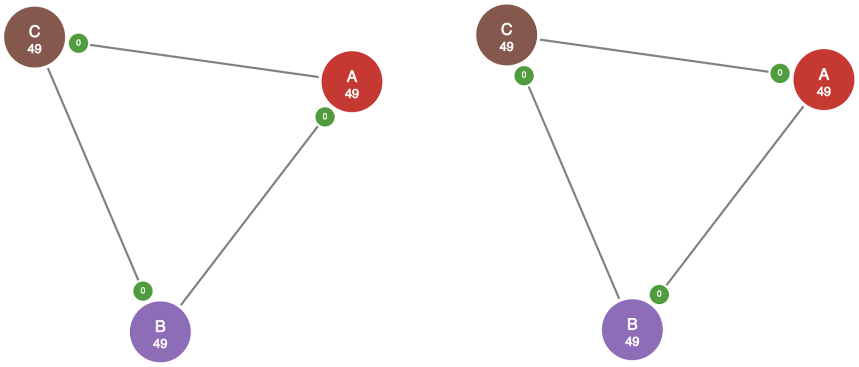
This shows Alice sending Bob 2 beads via Hub. The remaining capacity along this path is $\min(2, 1) = 1$.

Rebalancing

We say that two channel networks are value-equivalent if the set of agents is the same and each agent owns the same. A channel network is rebalanced when it transitions into a value-equivalent state. This transition can include on-chain transactions or not. In the following on-chain rebalance, Bob increases the capacity of the Bob → Alice link.



The following is an example of a rebalance that does not require on-chain transactions.



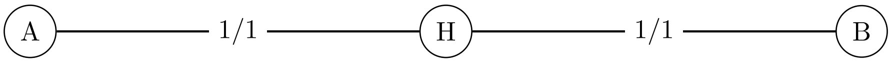
Are off-chain rebalances sufficient? In general, no.

Time-based lockup

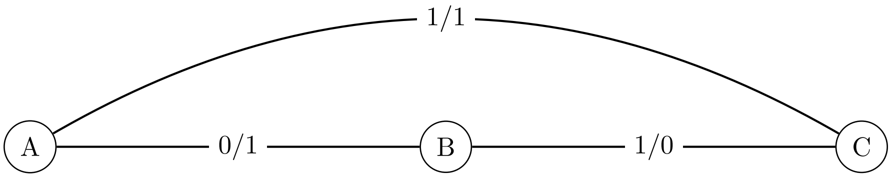
There are two ways that intermediaries can structure agreements-to-route. They can agree to route a single payment. When that payment completes successfully, the balances change and they are free of any further commitments. The second way is to agree to lock up funds for a certain period of time, creating a "virtual channel". The balances in the virtual channel can be updated instantly and many times without the intermediary's participation or knowledge. At the end of the lockup time, the final balances in the virtual channel are agreed to and we update the direct channel balances as though a single set of routed balance updates (routed payments) was made.

Ejection

There is an upper bound on the fee that the intermediary can charge, since any virtual channel can be transformed into a direct channel (i.e. it can be "ejected") by some on-chain transactions, without disrupting the internal state of the virtual channel. Here is an example. Suppose we have this system of direct channels. (Note: we switch to labelling edges by their endpoint balances rather than using beads, due to limitations of latex).



Here, a virtual channel between A and B can be formed, with balance one bead for each of them, locking up two beads from H. The channel could be ejected by an on-chain transaction that transfers one bead out of the A-H channel and H-B channel and atomically updating the internal states of them.



For this reason, thinking of lockups as "time-based" is not strictly accurate; the costs an intermediary would incur in a channel network is both the cost of re-configuring the capacity graph as well as time-based opportunity cost of having their capital in excess of payments they would want to make anyway in the network (e.g. instead of staking), however both can be avoided by on-chain transactions; the instances where on-chain transactions are avoided and off-chain fees paid instead are therefore some complex subset of transactions in general.

Here is a crude fee scheme that captures some of it this: an intermediary charges some large fee F_1 to intermediate a virtual channel, and assumes the responsibility of ejecting the channel; if the virtual channel participants have no further use for it, they negotiate a payment of F_2 from the intermediary to close the virtual channel offline. If the on-chain fee is T, we should have $0 < F_1 - F_2 < T < F_1$. On/off-chain rebalancing is handeled by an independent protocol.

State Channels

By allowing beads to be locked into complex applications (e.g. chess, prediction markets) instead of just payments, the need for virtual channels as well as for ejection is even more apparent. Beads can be freed up and re-used within a virtual channel. The time bound for an application might be long or unknown (e.g., we lock beads up in a bet of whether any US Libertarian Party ever wins a House seat in the next 10 years),

Multipath

TBW

N-Party

TBW

Subchannels

TBW

Plasma

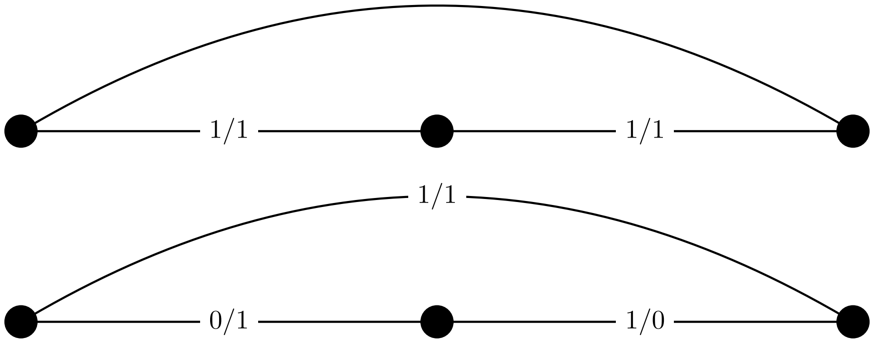
Plasma can be used to fund channels. If all channels are funded by plasma, then rebalancing is a plasma transaction in the normal case, i.e., the entire network can be rebalanced arbitrarily in a single ethereum transaction. I claim that a super-optimized channel network that does not support funding via plasma will be much more expensive than a very suboptimal one that supports funding via plasma.

Model

TBW

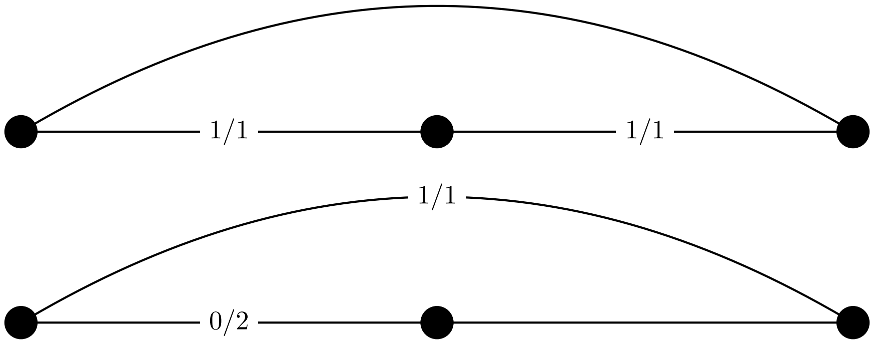
Ejection Test Cases

Symmetric



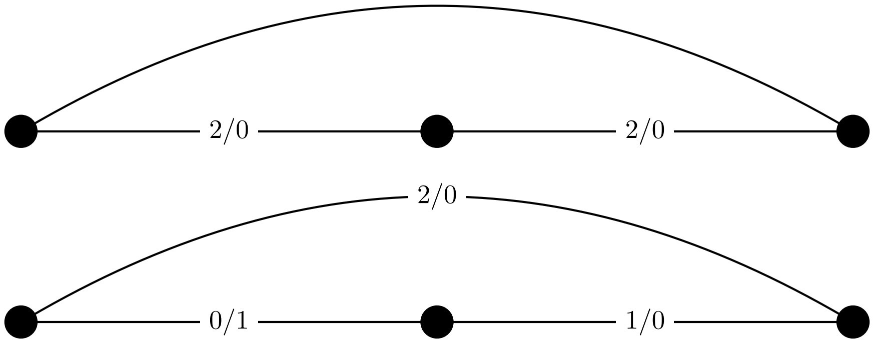
Reviewed previously

Asymetric

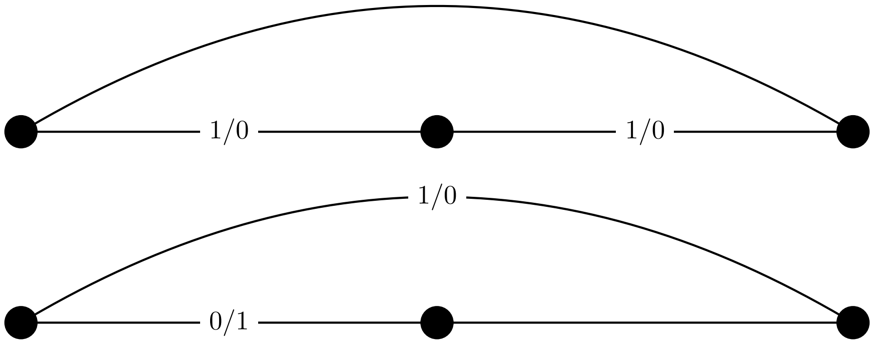


What are the differences between the the assymetric and symmetric ways of ejecting? The symmetric manner might be preferred because it minimal in the number of transactions (1 vs 2). Even if we assume some mechanism to batch two sub-transactions under one (e.g. account abstraction), it is minimal in the number of ERCECOVERS (3 vs 2). On the other hand, the symmetric manner might be preferred for the more "balanced" resulting capacity graph.

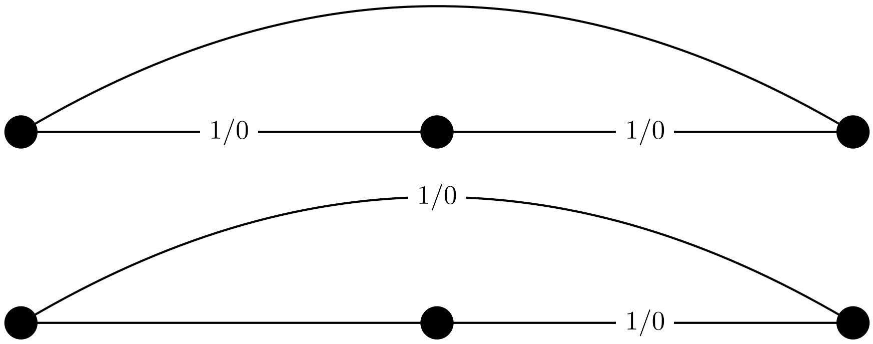
Symmetric Unidirectional



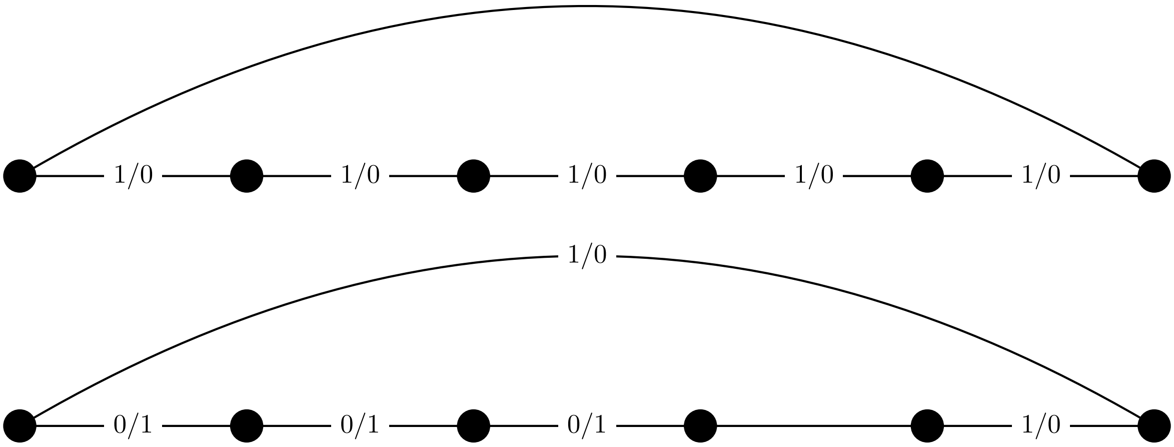
Asymmetric Unidirectional 1



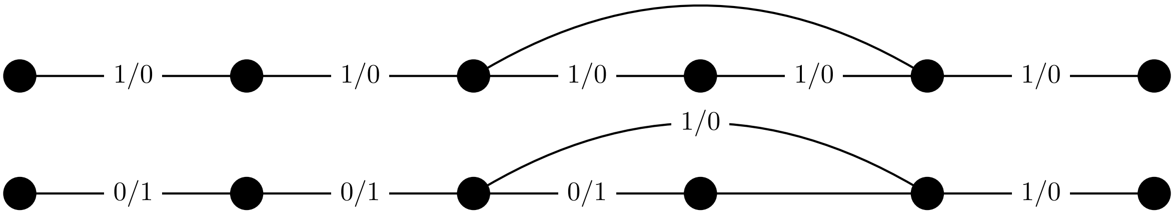
Asymmetric Unidirectional 2



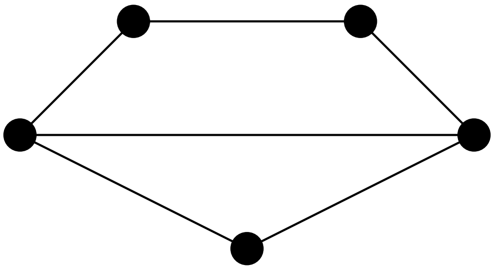
Long-Chain Large-Radius



Long-Chain Short-Radius



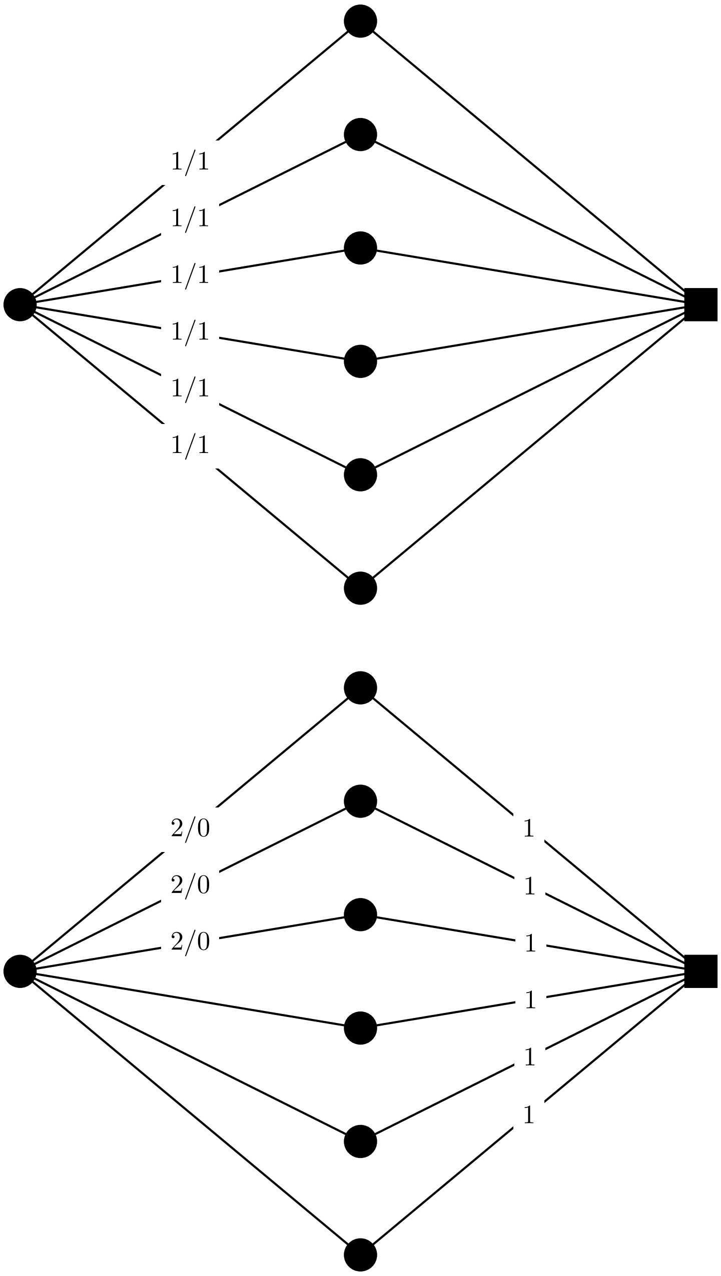
Multipath



Balances and ejection to be filled in.

Thanos Star

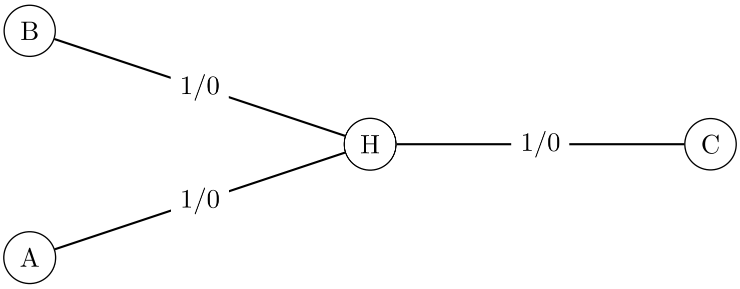
New notation: a square node is not a participant, but it means that its neighbours are all connected in a direct channel.



A 6-party virtual channel gets ejected into a 6-party direct channel. This manner of ejecting is minimal in number of transactions.

Other test cases

Under-Capacity Three Party Channel



If A,B,C form a virtual channel through H, they end up in a three-party virtual channel where A can send C one bead, and B can send C one bead, but the two cannot happen at the same time. The challenge is that it is inappropirate to represent this as a three-party channel where A and B's balances are both one.

Self Lock-in

TBW