The rules needed to coordinate decentralized multi-hop payments

Johan Nygren, @Bipedaljoe

The building block is a timeout that defaults to either finish or cancel the payment. It can enforce an action if it causes someone to end up with a net negative balance. When it defaults to finish the payment, it can penalize the person who has a pending outgoing balance but no pending incoming balance (initially the buyer). When it defaults to cancel the payment, it can penalize the person who has a pending incoming balance but no pending outgoing balance (initially the seller).

This building block can be used in either configuration to successfully coordinate a multi-hop payment. When it defaults to finish the payment at the timeout, the person who has a net negative balance is incentivized to cancel the payment unless they are certain everyone has agreed to it. When it defaults to cancel the payment at the timeout, each person who has their outgoing payment claimed from the next hop is incentivized to claim their incoming payment from their previous hop.

When either of those configurations is used alone, the penalty for not succeeding to make or pass on a decision before the timeout is the full payment. I.e., if the buyer does not cancel in time (or if an intermediary does not propagate the decision to cancel in time), they end up paying out the full payment to whoever is the last person in the chain. And if an intermediary does not finalize in time (when the timeout is set to default to cancel), they end up paying for the full payment to the seller.

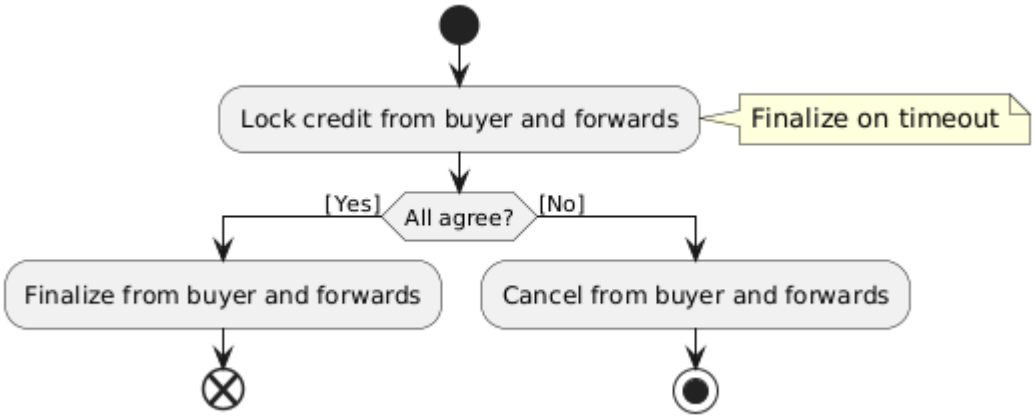
The coordination relies on that people put away money and promise it to the payment. The timeout places a limit on how long the money can be locked. The longer the timeout gets, the bigger the problem with Denial of Service attacks, i.e., payment attempts that are done only to lock the money of other people or sabotaged in such a way that the payment gets stuck.

The penalty can also be done in “chunks”, i.e., after the timeout only a “chunk” of the payment is cancelled or finalized, and then after the next timeout another “chunk” is processed, etc. Such a gradual penalty tends to increase the time until the payment has fully timed out, and therefore worsen the problem with Denial of Service attacks.

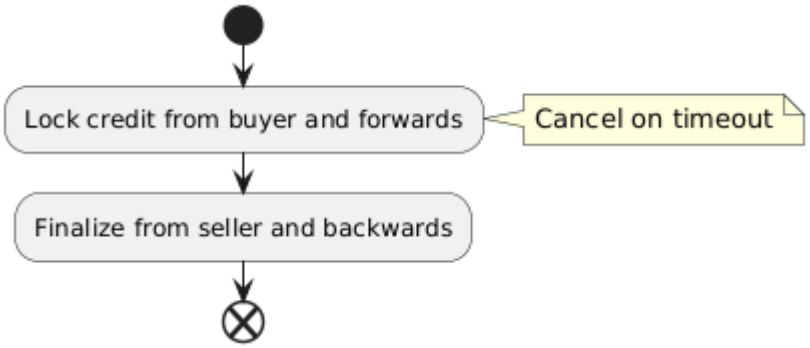
To resolve the issue of Denial and Service in full, the attacker has to always be penalized. When the timeout is used in the second configuration (defaults to cancel and finishes from the seller), the initial agreement to do the payment (that starts the timeout, done from the buyer towards the seller) is not enforced by any penalty. When the timeout is used in the first configuration (defaults to finalize and starts from the buyer), the penalty is active from the start, but once the buyer decides that the payment succeeded (everyone agreed to start it), the penalty does not enforce that each intermediary forwards this decision (the penalty can only enforce that the decision to cancel is forwarded). So the solution is to combine both of these payment solutions, to first start the payment from the buyer with the penalty started right away, and then finish the payment from the seller. As DoS attacks are then prevented, there is no obstacle to using the “chunked penalty” (that tends to increase the time until the payment has timed out in full).

The two-step payment (start from the buyer and finish from the seller, with the timeout initially defaulting to finalizing and then defaulting to cancelling) deters Denial of Service attacks in all scenarios except when the attacker controls both ends of the penalty (the person being penalized and the person receiving the penalty). To deter DoS attacks in that scenario, fees have to be added on top of the payment, paid out in proportion to how long the payment was stuck in DoS attack.

When the timeout is used in the first configuration, finalize on timeout to finish from the buyer and forwards, there is nothing to enforce propagation during the finalization of the payment, thus this configuration by itself is open to Denial of Service attacks.



When the timeout is used in the second configuration, cancel on timeout to finish from the seller and backwards, there is nothing to enforce propagation during the initial payment request sent from the buyer and towards the seller, thus this configuration by itself is open to Denial of Service attacks.



To deter Denial of Service attacks, both configurations have to be combined, with the first configuration leading into the second at the branch where everyone agrees to start the payment.