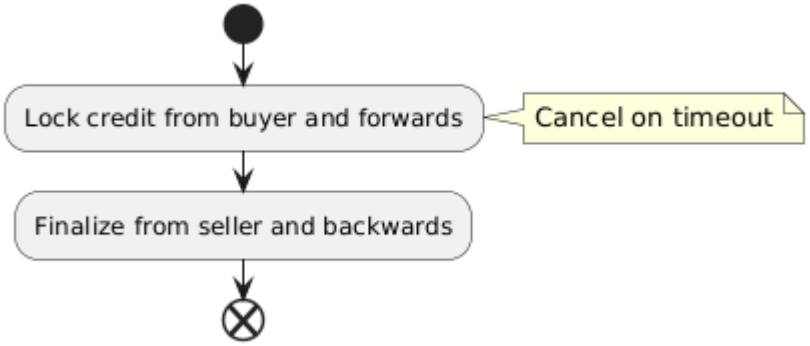
Rules to deter Denial of Service (DoS) attacks in multi-hop payments

Johan Nygren, @Bipedaljoe

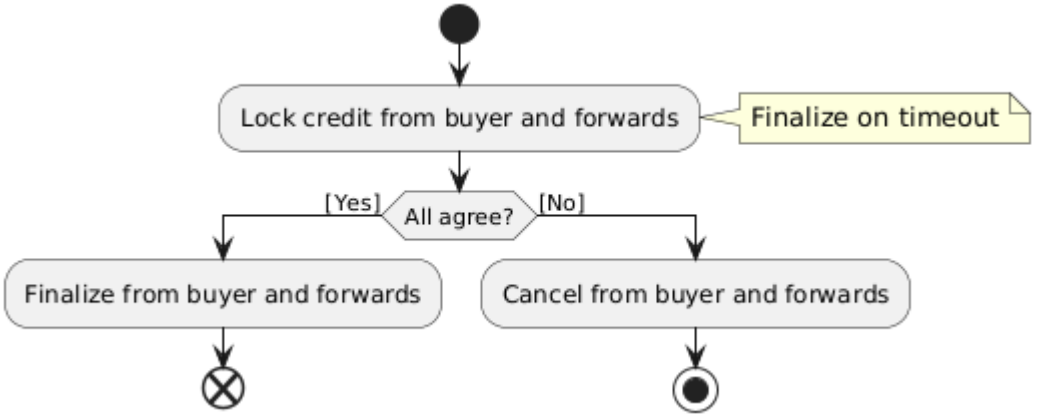
Multi-hop payments have a problem with Denial of Service (DoS) attacks, that an attacker can cause a payment to get stuck as pending. This is a problem since each node in the payment has to reserve money for the payment, and if the payment gets stuck that reserved money is also stuck. The trivial solution to Denial of Service (DoS) attacks is a timeout (that either cancels or finishes the payment).

The problem is that a timeout can solve DoS attacks but it causes another problem in that it risks unfairly punishing a non-attacker. To solve that new problem, the duration of the timeout has to increase which means the timeout in itself no longer solves DoS attacks. The penalty can still solve DoS attacks, but a timeout set to either cancel or finish the payment cannot penalize every single DoS vector. What is needed is to combine two timeouts, one that cancels the payment and one that finishes the payment, to deter every single DoS vector (and thereby allowing continuous timeouts that process just “chunks” of the payment, rather than a single timeout that processes the entire payment).

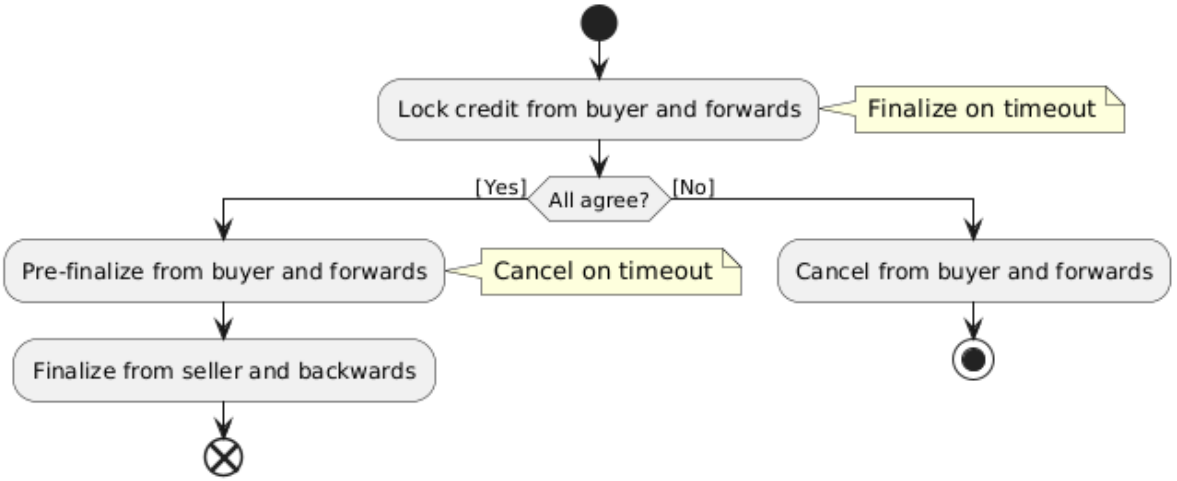
The trivial solution with a timeout that cancels the payment will not penalize anyone while agreeing to do the payment (from the buyer towards the seller), but it risks penalizing someone when finishing the payment (from the seller and backwards). With this solution alone, the penalty cannot be “chunked” as that makes the “start payment” step vulnerable to DoS attacks, and without a “chunked penalty” the “finish payment” step is vulnerable to a non-attacker being punished.



The trivial solution with a timeout that finishes the payment will not penalize while finishing the payment (from the buyer towards the seller) but it will risk penalizing someone when cancelling the payment (from the buyer towards the seller). With this solution alone, the penalty cannot be “chunked” as that makes the “finish payment” step vulnerable to DoS attacks, and without a “chunked penalty” the “start payment” step is vulnerable to a non-attacker being punished.



The solution is to combine both trivial solutions, to start the payment with the timeout that defaults to finish the payment and to finish the payment with the timeout that defaults to cancel the payment. With such a solution, the penalty can deter DoS attacks, and there is no need to rely on the timeout itself (thus, the total length of the combined “chunked penalty” timeouts is no longer a factor as the trivial timeout is not what avoids DoS attacks).



The two solutions combined (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.

So, a solution (timeout) to a problem (DoS) caused another problem (risks penalizing non-attacker) which required another solution (do continuous timeouts with just “chunk” of payment as penalty each time) which ruined the original solution (the combined timeout duration now allows DoS again) while the penalty the timeouts allowed (that could be a solution) did not cover all DoS vectors in the trivial solutions alone (only one of either timeout that cancels or timeout that finalizes). The complete solution requires the combination of both trivial solutions.