Game theory to enforce multi-hop payments (deter “reserve credit attacks”)

Johan Nygren, @bipedaljoe

A major problem in multi-hop payments is the “reserve credit attack”, that a person can initiate a payment and then cause it to never finish, thus locking credit forever. This attack can be prevented by financially punishing the attacker. The mechanism for such penalty differs depending on who the attacker is, and a complete solution requires a three step payment agreement, “commit”, “seal”, “finalize”, with fees paid by the buyer added on top of the payment and agreed on beforehand.

The game theory can be explained by starting with the last step and then showing how the steps leading up to it are required to organize that last step. During “finalize”, each intermediary from the direction of the seller and towards the buyer will complete the payment. At the same time, the amount that can be finalized starts to gradually decrease after the timeout has been reached. Thus, an intermediary who does a “stop-propagation attack” (the mechanism to cause the “reserve credit attack”) will end up with less incoming payment than their outgoing payment they already finalized. They thus end up having to pay a penalty. This step thus requires that there is a “continuous cancellation” on the buyer side of the intermediary who propagates “finalize”. This “continuous cancellation” is set up using the two previous steps, “seal” and before it “commit”.

The “seal” step is a bridge between the first step “commit” and the “finalize” step. Primarily, “seal” signals that the buyer has revoked their right to cancel (the right of the buyer to cancel is indispensable to organize the first step, “commit”). It is very important that the buyer revokes this right and that each intermediary knows that they did. On top of that, any user who has propagated “seal” will start to continuously cancel the payment (thus adding the enforcement penalty onto “finalize”). This “continuous cancellation” also serves as an enforcement to propagate “seal” itself, but only because the previous step, “commit”, had set up a “continuous finalization”. Thus an intermediary who does a “stop-propagation attack” on “seal”, will end up having their outgoing pending payment gradually finalized while their incoming pending payment is gradually cancelled (at equal rates seems best). They thus end up having to pay a penalty. This penalty that enforces “seal” thus required the “continuous finalization” to have been organized originally during the “commit” step.

The “commit” steps main role is to ensure everyone is on board with the payment and goes “all-in” on it. To ensure this, we rely on that the buyer will be punished if everyone has not agreed yet and the payment times out (and thus forced to use “buyer cancels” to cancel the payment). For this punishment, we need to add a fee on top of the payment, that has been agreed on by all people in the payment. This fee will be a percentage of the payment, and each person in the payment will demand a certain rate and the buyer will prefer the path with the lowest total fees. The fees are paid out at the same rate as the “continuous cancellation” and “continuous finalization” (i.e., a single rate that says “how much of the payment has been either cancelled or finalized as well as paid out as fees yet”). The fees are paid out in every step of the payment (both “commit”, “seal” and “cancel”) and they compensate the people who are victims of a “reserve credit attack” (and if no attack or network failure occurs and the payment finishes before timing out, no fees is paid out). The fees are paid out in proportion to how much of the payment had time to be “processed”, if the payment was stuck for a half of the time it would have taken to cancel/finalize the entire payment, half the fees are paid out.

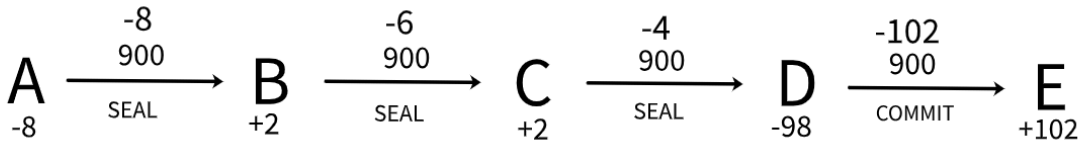
So, we have single ticking rate for cancelling the payment, finalizing the payment, and paying out the fees. The fees deter the buyer from doing the “reserve credit attack”. The cancellation and finalization deters everyone else (for the seller, they are always guaranteed to get the full amount as long as “commit” succeeds and the seller themselves do not do “stop-propagation” at “finalize”, so the buyer will know that the payment was successful or that if it was not it was the seller themselves attacking and then the seller cannot claim that they were not paid).

Agreeing on the “buyer fees” and “cleanup rate”

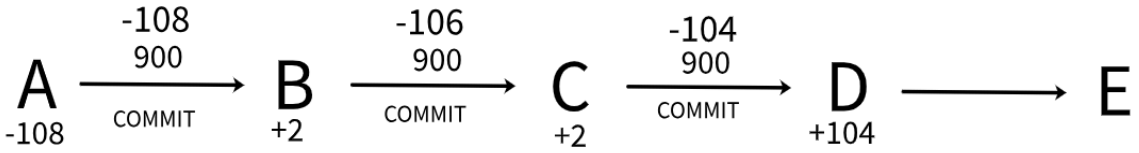
Simplest is that the buyer/seller says what fee the buyer will provide as well as what “cleanup rate” (how fast the continuous cancellation, continuous finalization and fee payout happens, i.e., how fast a stuck payment is cleaned up and the pending payment object removed), and select paths that accept these values. People then have a lower bound for what fee they accept and a lower and upper bound for what “cleanup rate” they accept. Thus a selection system is formed between buyer and intermediaries. Since penalty system exists to deter attackers, people would generally work to set good “norms” for those two parameters, collectively (although in a decentralized way).

Illustrations

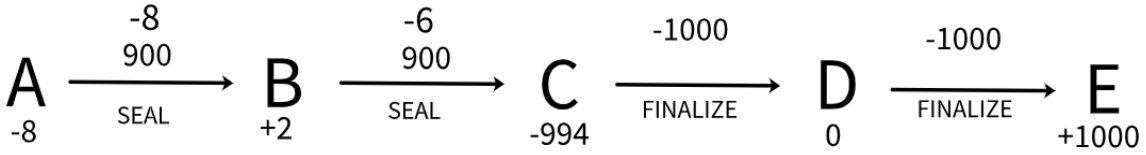
A payment of 1000 XYZ gets stuck at D during the “seal” step (D is doing a “stop-propagation attack” or there was a network failure). The “penalty rate” is 1 XYZ per hour (thus a little more than a month for 1000 XYZ), and 100 hours have passed (roughly 4 days), during which the payment has been stuck at D ever since it timed out (and the period until timeout was maybe 5 minutes then after the payment had been started 4 days prior to the moment illustrated below). The “buyer fees” are 2%, i.e., for the full payment being “cleaned up” each person would get 20 XYZ and now that only 10% has been “cleaned up” each person has received 2 XYZ. The buyer thus has a negative balance of 8, while the seller has received 102. The buyer has thus been compensated with 92 XYZ (as their payment to E always succeeds as long as the buyer issued “seal” and the seller themselves is not the attacker). Each intermediary as well as the seller has received 2 XYZ from the buyer fees, and thus been compensated for the stuck payment. Note, 900 is the pending payment amount, thus 1000 XYZ minus 100 XYZ.



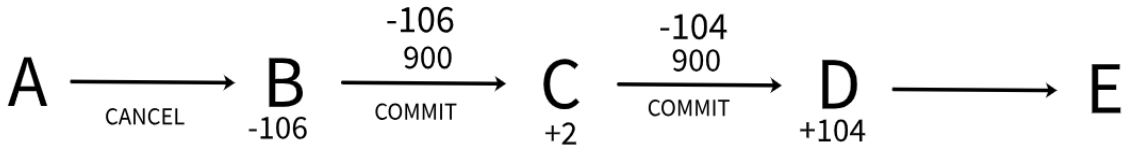
Another payment also at 1000 XYZ and also at “penalty rate” 1 XYZ per hour as well as buyer fees at 2% that has been stuck for 100 hours (roughly 4 days) because of a problem at D (either a network failure or an attack). Here it is illustrated that the buyer A has an incentive to cancel the payment - and should have done so 4 days ago the moment it timed out 5 minutes or so after it was started! It would go without saying that the buyer cannot issue “seal” here as they never got the signal from the seller E that “commit” succeeded (since “commit” never succeeded, it got stuck at D…)



And then a payment example also of 1000 XYZ with 1 hour per 1 XYZ penalty and having been stuck for 100 hours (with a 2% buyer fee). This time, it is stuck at C during “finalize” (either because of a network failure or because C is an *attacker*). If C issued “finalize” at the moment shown in the illustration below, they would be left with a penalty of 94 XYZ. The reason the illustration shows -994 is because they already finalized 1000 XYZ, but 94 XYZ is the actual penalty they would receive if the penalty period ended in the moment shown below (i.e., if C issued “finalize”). Note that the buyer has here been rewarded with 992 XYZ as their payment still succeed (the payment always succeeds as long as the buyer does “seal” and the seller is not the cause of the stuck payment).



And a final example, where the payment got stuck at “commit” (at D) but then the “cancel” signal also got stuck (at B), both either because of a network failure or an *attack*. Here the buyer is not penalized at all. Instead, the penalty starts to affect B, thus it enforces the propagation of “buyer cancels” perfectly.



Implementation

This game theory is very simple to implement. As it centers around a “ticker” that is just the machine clock, any recordkeeping about how much was cancelled or finalized or paid out as fee only has to be done when there is a state transition such as from “commit” to “seal” or if the full payment was “cleaned up” (the ticker reached the full payment amount) and then via a separate “cleanup_payment” command handler. The game theory most likely works for something like Interledger and it also works for the person-to-person money system Ripple (i.e., it should work regardless of what a “node” is defined as).