Ripple Inter Server Protocol built on single-hop consensus and a penalty system

Johan Nygren, @BipedalJoe

ABSTRACT: Ripple Inter Server Protocol requires a person-to-person consensus mechanism to solve the Two Generals' Problem as well as a penalty system to resolve “stuck payment attack”. The ideal consensus mechanism is to take turns to authorize every other transaction (that can be submitted by either person) and the ideal penalty system is to increase outgoing balance relative to incoming for the person who can break that attack (in all cases but one this will be the attacker themselves). With a consensus mechanism and a penalty system, Ripple Inter Server Protocol can be organized with person-to-person messages only and with people sharing state only with the people they know personally and have a trustline to.

Introduction

Historically, money has been organized into either central accounting (“banks”) or decentralized routing where one bank could send money to another bank via a number of intermediary banks. With Ripple, the “bank” component is reduced to just two people and all other payment functionality is moved to the “inter-bank” multi-hop payments. Thus for the central accounting part, Ripple needs only consensus between two people. Traditionally, “inter-bank” multi-hop payments have typically used a central authority to coordinate, but such payments can also be done in a decentralized way. For multi-hop payments without a central authority, a penalty system is needed to enforce that the “banks” all follow the rules. Such a penalty system is fully decentralized and enforces the resolution of “stuck payment attack” in all scenarios.

Technical challenges in multi-hop payments

The technical challenges in multi-hop payments are the Two Generals' Problem, and the “stuck payment attack”. The solution to these issues are a consensus mechanism, and a penalty system that enforces the resolution of the “stuck payment attack”.

The Two Generals' Problem is that agreements are impossible if a communication channel is unreliable and no consensus mechanism is used, as it is impossible to be certain if an acknowledgement was delivered. The problem comes from equal authority, and the solution is to agree on a central authority, *a single general*. The ideal solution is to take turns being the authority.

The “stuck payment attack” is that a person could decide to not pass on a decision, thus causing the payment agreement to get stuck. Since the payment agreement requires money to be locked, such attacks can cause a lot of money to become unusable. The solution is to enforce the resolution of the attack by penalizing the person that has the authority to break the attack (in all cases but one this is the attacker themselves).

The person-to-person consensus mechanism

The ideal consensus mechanism is that the two people take turns being the “general”. They coordinate this with the use of a counter, and agree that one person will validate every odd number and the other every even number. The person who is not the "general" at a turn can propose transactions to the "general". Each person stores the instruction they last validated in permanent storage until they receive the next round's validated transaction (thus continuity is guaranteed.) In permanent storage you thus maintain a turn bit (0 or 1) and a turn counter for *counter mod turnbit*, and the last validated instruction (an instruction being a command with arguments). The "state transition function" includes incrementing the turn counter and setting the last validated instruction, as well as the state changes that the instruction performed, and is "atomic", all-or-nothing. Thus, the two people are in perfect agreement over every decision that they make.

The penalty system

A penalty can be exerted onto a person if their incoming balance is reduced relative to their outgoing balance. Practically this is done by reducing the pending payment amount on their incoming balance but not on their outgoing balance. The payment coordination is done in steps or “states” where in one “state” people will reduce the pending payment amount and in another “state” people will not. A person in-between “states” (i.e., a decision has gotten stuck at them, the decision is what changes the “state”) will thus be penalized. Besides this mechanism, separate fees (paid by the buyer) are also used to cover the scenario where the attacker is both the buyer and another person.

The steps of a multi-hop payment

The payment coordination consists of three steps. The first and second step are used to enter the payment and the third step is used to leave the payment. In the first step, the payment either continues into the second step or cancels. The second step always leads into the third step (or, the equivalent of the third step happening from that the penalty system has used up and thus paid out the entire payment).

The payment starts from the buyer who sends a “commit” decision towards the seller. If each intermediary agrees and “commit” reaches the seller, the seller will signal the buyer directly. During this step, the payment will start to be continuously paid out (very slowly, in tiny “chunks”) after a time out (or, at a slow enough rate that this penalty can start directly). This means that the buyer has an incentive to cancel the payment unless the “commit” step succeeds (as they end up sending money to whoever caused the payment to get stuck…)

If “commit” succeeded the buyer will move to step two, the “seal” decision. Here, the buyer is formally revoking their right to cancel the payment, and the payment is now irreversible (although still not finished). When a person has forwarded “seal”, they will stop continuously paying out the payment and instead they will start to continuously cancel the payment (slowly, in tiny “chunks”). Thus, for a person who has received but not yet forwarded “seal”, the incoming balance from the buyer side will decrease relative to their outgoing balance (that is still in the “commit” state).

Once “seal” reaches the seller, the seller will move to the third step and issue “finalize” and send it towards the buyer. Recall that the third step is used to leave the payment. Thus, every person who has forwarded “finalize” will also have finished the payment. “Finalize” pays out the payment minus the amount cancelled from the penalty system in the “seal” state. Thus if “finalize” gets stuck at an intermediary, the amount they will be able to receive to their incoming balance will become relatively less than what they already paid out to their outgoing balance.

Buyer fees deter combination attacks

The penalty system mechanism that reduces incoming balance relative to outgoing works in all scenarios except for when the buyer is attacking together with a person at whom “commit” is stuck. As then, the attacker is just sending money to themselves. This scenario is deterred by adding fees on top of the payment, paid by the buyer and paid out in proportion to how long the payment was stuck. The fees also have a second functionality, they compensate all victims of a “stuck payment attack” (and as long as the buyer is not the attacker, the buyer is also being compensated as the attacker is paying for the payment to the seller).