Adding Redundancy to Splitting Protocols for a Better Performance

TU Delft

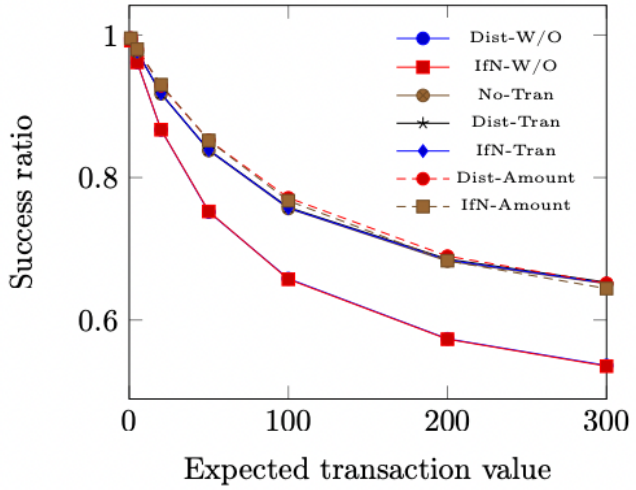
Ivaylo Georgiev
I.Georgiev@student.tudelft.nl
Supervisors: Stefanie Roos and Oguzhan Ersoy

1 Background

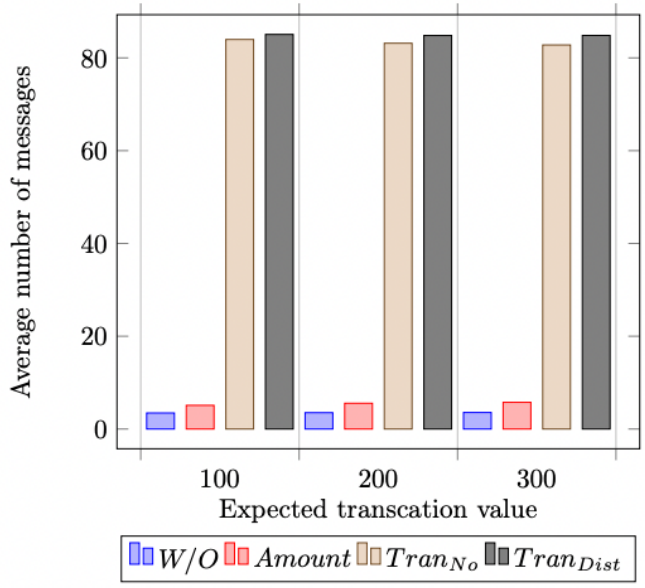
- Payment-channel networks were developed to tackle blockchain's limited scalability
- Splitting payments protocol was created to fix the issue of intermediaries having insufficient funds to forward a payment
- **Aim:** Combine a redundancy protocol (such as Boomerang) with the already existing splitting one and measure the new protocol's effect on the performance

2 Protocol Design

- **Tran:** the payment is divided into ***v*** equal transactions, which are then forwarded through the network one after another; failed transactions are rollbacked and can be resent at most ***u*** times
- **Amount:** the whole payment is forwarded; in case of a partial transaction failure only this part is resent from the beginning; total redundant amount that can be resent is ***u***



Success ratio for different combined algorithm settings



Average number of messages for different algorithms

3 Results

- **Success ratio:** both combined protocols increase the success ratio; in general, **Tran** achieves higher success ratio than **Amount**; success ratio is affected by ***v*** and ***u*** (in both designs)
- **Overhead:** number of exchanged messages for one transfer increases; when using **Tran** this increase is very significant; slows down transfers in static scenario

4 Conclusion

- Success ratio increases when redundancy is added, but overhead also increases, mostly when the **Tran** design is used; slows down payment process in the static case quite significantly
- **Future work:** implement concurrent transactions - more realistic case; this way messages could also be sent concurrently, and thus not slow down the whole process