

I use the term "transaction" in the non-financial sense

While existing models and protocols can provide a significant increase in potential speed, they do not allow for unbounded scalability and concurrency. Most require either a complex data structure or consensus mechanisms. With proof of fraud we can differ from a traditional proof-of-work consensus in that every participant grows and maintains their own chain of transactions. Instead of a global chain containing all transactions performed by users, each block generated contains at most one transaction. Each transaction upon completion is signed by both parties and appended to a new block on their local chain.

What if a user decided not to submit and append the block generated to their chain? A pointer in each block is added that points back to the last block in the chain of the transaction counterparty(s). Each encounter/transaction has at least two incoming and at least two outgoing pointers. This is a mechanism of localized consensus. The validation of blocks is performed prior to a block being committed to the localchain of each user. During the commit process, pointers, sequence numbers, transaction data and scheme and signatures are verified. Once the block is marked valid it is inserted in localchain and shared with other network participants, and eventually committed to the globalchain.

Max Flow: This computes and measures the maximum flow passing between two nodes.

Reputation of a node can be computed as:

$$\arctan(f_{ji} - f_{ij})/(\pi/2)$$

Where every reputation value is normalized with the factor

$\pi/2$  so that it is in  $(-1, 1)$

Eigenvector centrality: calculates a metric for the importance of a node in a network. A real-world application of this is Google's PageRank. Eigenvector's basic idea is that interactions with highly reputed nodes contribute more to the reputation of that node. In Google's PageRank power iteration is defined:

$$r_{t+1} = dAr_t + [(1 - d)/N]\mathbf{1}$$

Taking into account that the contributions of nodes in the computation of reputations are not equal in quality and quantity, we could use a reduced history graph to increase processing.