The paper starts off with the discussion of traditional blockchain protocols and the issues with their design. The author uses several examples to show why the existing protocols either compromise on some very essential features or they are not scalable with the number of users. One such example is of the proof-of-work feature of the Nakamoto consensus. Some of the drawbacks of this PoW-based consensus are low transaction throughput, poor energy efficiency and inability to scale out its transaction processing with the numbers of participants. In addition to the stated problems, in Bitcoin’s case, the user needs to download and verify the entire blockchain to be able to join the network. The paper then describes that the so far proposed solutions either compromise on some existing feature or introduce new problems to solve some existing problem.

Next the authors make an introduction of their way of approaching the problem. They call their protocol RapidChain that, they claim, is better at scaling and is more secure than the prior solutions. RapidChain divides the network into groups which they call *committees* and the size of each committee is much smaller as compared to total nodes in the system. The key highlights of their protocol are: **Sublinear Communication, Higher Resilience, Rapid Committee Consensus, Secure Reconfiguration, Fast Cross-Shard Verification** and **Decentralized Bootstrapping**.

Before getting into further details of the actual protocol, the paper makes certain assumptions. The first one is for the underlying network that it is a peer-to-peer system of n nodes with established identities (public/private keys) and secure channels for communication. The system is assumed to have a constant upper bound for the time required to spread a message to the entire network. They state that for intra-committee consensus, the communication has to be synchronous. The protocol proceeds in intervals which the paper calls *epochs* and it is assumed that nodes may disconnect during and between each epoch due to some local failure. Further the possibility of Byzantine attack is also taken into account and the attacker can only corrupt additional nodes in-between epochs but its influence never exceeds n/3 nodes.

The actual protocol consists of three main procedures: Bootstrap, Consensus and Reconfiguration. As stated earlier, these procedures occur in different epochs. Starting off with bootstrapping, the protocol, in this phase, elects a group of nodes the paper calls *root group*. The responsibility of this group is to establish a reference committee which generates and assigns committees to all nodes. This phase runs once and only at the start of the protocol. Once every node is assigned a committee and all of them know they are done with bootstrapping, the system enters the consensus phase. In this phase, the system receives a set of transactions to be processed such that each user sends its transactions to a fixed subset of the nodes. These handler nodes batch and forward the transaction to the corresponding committees. The committees run an intra-committee consensus protocol, that is discussed later, to approve the transactions and add them to their local ledgers. The reconfiguration component, that is also explained later, allows the system to take in new participants without the need of regenerating all committees.

The intra-committee consensus comprises of two main components: A gossiping protocol to propagate a message to the entire committee and a synchronous consensus to agree upon the block header. The authors describe their own version of gossiping protocol which they call IDA-Gossip. While IDA-gossip is not a reliable protocol, the paper claims that the protocol requires much less communication and is much faster than the existing reliable protocol when used to propagate large messages. For synchronous consensus, the use a modified version of the existing Ren protocol which allows smaller committee size with a higher resilience.

One problem with such a distributed-ledger protocol is that for some transaction tx, its inputs and outputs could belong to different and multiple committees. The paper solves this issue by allowing the owner of the transaction to communicate to any committee that routes tx to the output committee through an inter-committee routing protocol which allows the user to make to quick lookup for the right committee for each transaction.

The protocol uses PoW technique to allow new participants to enter the system. All committees are provided with a newly generated random puzzle by the reference committee at the start of each epoch. Any new node that wishes to join the network from next epoch onwards reaches out to a random committee and is given this computational puzzle. Nodes that solve the puzzle in the current epoch send the solution to the reference committee for the verification. The reference committee finally assigns a committee to each of the new nodes.

**Conclusion:**

The paper proposes a new technique of storing the blockchain and processing new transaction in a distributed fashion. The authors also provide a detailed analysis and evaluation to show how their technique is better than the existing protocols. Although, the overall distributed design enables their protocol to outperform almost all existing protocols, the paper fails to address some very critical issues.

While the paper starts off with stating the problems and issues with existing blockchain protocols and claiming that their technique takes care of all those issues, instead of solving, it actually seems to some of those problems from one part of the protocol to another. One such example is the notion of honest nodes while stating the network setting and environment. This assumption is problematic because the authors do not explain how honesty certification is assigned and it seems to go against the basic goal of achieving complete decentralization.

Next, the paper makes the assumption that the adversary can never be able to control more than 1/3 of network. The assumption is used at several places in the paper but they never state what happens if this assumption breaks and how the system tries to mitigate such situation.

While paper is well written, in an effort to answer some very important existing questions, it leaves the reader with some new unanswered questions.