My design about defence of blockchain

Briefly,messages from high credit nodes never queue,high credit nodes reconnect usually,update the nodes of trust network randomly and connect randomly,the trust nodes address saved in block,low possibility your trust nodes are all connected by adversity.

At first,I thought we should build an evaluation system to select nodes as high credit nodes.But then I thought it provided the adversary a way to attack the defence.So now I decide to select high credit nodes randomly(which is uncomforable, beacuse we usually select nodes that are well behaved.But it is more safe,I do not know which nodes will be selected ,so does the adversary in a way.).So it is inaccurate to call the nodes high credit nodes,but let me just use.And all design I distinct miners and clients in order to let the high credit network smaller,although the main work of high credit nodes is to send message immediately so clients' nodes are ok.//Maybe we can allot the number of trust nodes by how powerful the node group has.

Now the first question is coming:Why I want to build a high credit network?

We know most attacks of blockchain is based that the victim does not know the up to date information about transaction or block,the information is dalayed or doscarded by the nodes of adversary.So let us build a high credit network to transmit the information.I assume high credit nodes have to 8 TCP connections to connect to each other to build the network,and the rest TCP connections to connect normal nodes and clients.The most important thing is that every node connects to high credit nodes randomly and I am sure normal nodes want to connect high credit nodes as many as possible (I assume normal nodes can connect 8 high credit nodes at most).According to the papper tampering with the Delivery of Blocks and Transactions , every client has 125 TCP connections, but I do not understand he said only 8 TCP connections are outgoing TCP connections.

Even though the high credit network only has 90 percent of all nodes are belong to adversary (which is despairing),it is 43 percent that a node connects to 8 nodes adversary.

Some adversary may use delay attack by using inv message to fill the buffer of information and never send the entire information , so I design that if high credit nodes send inv message ,it will be dealt immediately and never wait.Since it is only 8 connections, I think it is acceptable.

Now I introduce how trust network update.Of course the high credit network should be dynamic(at fist I think it's safer to be static if we have many third trusted parties to provide enough trust nodes , but it seems to loss the advantage of distributed system).I design that every day 10 percent of high credit nodes will be exchange by new nodes.When a node is adding to the high credit network,it will be recorded on the block ,then it cut the 8 TCP connections used to connect trust nodes.The new node uses his private key of the pubilc key pair to announce that it wants 8 connections.The trust nodes then use his public key to encrypt and send the IP address to this new node,and new node selects 8 to connect randomly.According to the papper tampering with the Delivery of Blocks and Transactions , when 2 nodes have a TCP connection , they will exchange bitcoin version message and it contains the hight of blockchain , which means the length of blockchain (belong to selfish miner) will expose.10 days later, this node will be replaced by new node.You can count time from the record of blocks.When the node leave the trust net ,it has allready connected 8 trust nodes (it is possible that some nodes leave the trust net,too.Need to find a new one),I am not sure whether this node need to connect 8 new trust nodes.

The problem is I am not sure how to select new nodes to join the trust net.I think the trust nodes must add information in block,but if I need to add all address into block, it seems too heavy.I think it is ok when the node is going to leave the trust net,'it has to select a node from his connection to normal node.But it seems easy to be attacked.

About how to random, I think a node can use hash.

About scalability,assume the number of trust nodes is x percent of all (including clients), 125-8x¿8(1-x) x¿0.064 ,I think it has acceptable scalability.

About safty. 1:It's possible that a node belongs to adversary selected as a trust node, if adversary have ability to select next node rather than random, it can select his another node (I am not sure if this is possible,and his node should connect to his own nodes or it will lose the hight of blockchain from selfish mining).It seems one day adversity will occupy the trust net.I do not have good solution so I hope they can not select.But it will behave strange if you read the record of blockchain about the trust net.Some nodes that never be the trust nodes can hint the all users. 2:adversary can use useless IP address to connect trust nodes,like said in the papper Eclipse Attacks on Bitcoin's Peer-to-Peer Network,and the papper provide solution.

Assuming the system run according to schedule.If adversity is lucky enough, they will occupy much of trust net, it is nearly impossible to control the entire trust net(from Expect,it still has trust node),and most important, adversity have a lot of nodes and spend a lot of time waiting to occupy most trust net.If adversary succeed and control the blockchain ,which leeds to lose clients and miners ,finally the coins will lose value,and what adversary put into the bitcoin is waste.They would earn more since they have so much computing power.So what's wrong with you ?hahaha.

I need data from experiment to define the defence is effective and the throughput is ok.I do not know the detail how nodes of blockchain connects to each other.