Bitcoin: SEIR-C propagation models of block and transaction dissemination

Dr. Craig S Wright

Abstract

Bitcoin can be modelled using the mathematics of biological systems. SEIR and SEIR-C models capture the propagation of transactions and blocks within the bitcoin network. With systems reacting as a susceptible or immune carrier, we can model the forces driving cycles of recurrent states. We demonstrate that we can analyse numerically the bitcoin network using a susceptible/exposed/infective/recovered (SEIR) epidemic model with seasonal transmission. Here, the system expresses a periodicity related to the discovery and propagation of blocks that can be extended into a competing epidemic model where blocks are discovered at a similar time or when transactions are double spent. In this paper, we introduce the concept of an SEIR model into the Bitcoin system and demonstrate how this can help in understanding the system.

1. Contents

2. Introduction 3
3. What is Bitcoin? 3
3.1 Overview 3
3.2 Phases of the Bitcoin system 4
(a) Mining 4
(b) Holding Bitcoin 5
(c) Transferring Bitcoin 5
4. Functioning and modelling of nodes 6
4.1 Vertex modelling 8
4.2 Epidemic Models 11
5. Epidemic models of block propagation 13
6. Equilibrium 17
7. Competing Epidemics model of nodes 21
7.1 Reproduction Number 22
8. Conclusion 25
9. References 25

2. Introduction

Even at a low distribution rate from poorly connected nodes with low bandwidth, we can model the maximum time for a transaction to be broadcast across the Bitcoin network. In knowing the conditions that can lead to a low intensity of distribution, these can be modelled so that you can find the optimum strategies to propagate transactions widely without long delays.

The presence of double spends, and transactions that do not conform to the protocol rules, can result in propagation delays. Similarly, the introduction of an increased block size will stress many existing nodes beyond the existing capacity.

The decline of the number of nodes that are incapable of scaling to the new limits, and the impact on the network as these nodes are stressed, can be explained using a series of Markov chain epidemic models. We model the decline in low powered nodes to the function of individuals who are susceptible to infection. In these models, an infective will (in simple terms) either die or recover with a resistance to further infection.

Here we express the modelling of an individual death in epidemic models, to be the failure of a node that leads to its removal or rejection by the network through a series of excess lags and delay. These naïve explanations can be extended, to answer many important quantitative questions related to predicting the behavioural modelling for transaction and block distribution between nodes and users in the Bitcoin network.

An idealised node network, would support all sets of nodes in a manner that allows any pair of nodes to connect randomly and independently at a common rate. It is also expected that any node that cannot handle the transaction load, either as a consequence of low bandwidth or computational power, to exit the network gracefully or be upgraded in a timely manner. This would match an epidemic spread death / recovery process.

In this process, we see the exchange of a transaction between nodes hold and propagate newly received transactions. The transaction is either sent to a node that has not received a conflicting transaction (a double spend), or it is rejected by the node. If the transaction is not rejected, the node will either have the capacity to process it, forward it to further nodes, or fail within an exponential random time.

The set of consequences for the progress of the transaction through the node network, can be modelled in the worst case to have identical consequences. This offers us the extreme case of a system in a decayed state. From this we can look at the improvements that can be made to the network to ensure the controlled distribution of transactions and blocks to all viable nodes on the network. We start with the following basic model to understand the worst-case features of the propagation network.

3. What is Bitcoin?

3.1 Overview

Bitcoin is a "crypto-currency", a form of digital or virtual currency using cryptography to control its creation and operation.

Bitcoin was first launched in 2009, when the Bitcoin algorithm was released online by an anonymous programmer or programmers using the pseudonym "Satoshi Nakamoto" (Nakamoto, 2008).

The creation and transfer of Bitcoin is based on an open-source cryptographic protocol (essentially, a software program that is free to download, with users having access to the source code and ability to modify it), and utilises a peer-to-peer computer network made up of its users' machines (**Bitcoin**

network) to validate transactions by solving complex mathematical equations.[1] A few authors (Grunspan & Pérez-Marco, 2017) have started to note that the initial whitepaper (Nakamoto, 2008) was a gross simplification when all but the most basic of analysis is conducted. Others, (Decker & Wattenhofer, 2013) have conducted a few simple tests that look at times between the nodes without analysing the system. This resulted in a flawed assumption of a high distance network and not the small world system that has resulted in practice. Others (Babaioff et. Al, 2011) have noted that a high distance between nodes would lead to attacks and that a low distance would leave the system immune from many attacks without conducting any further tests.

This validation process is known as "mining". Essentially, the mining process involves the Bitcoin network updating and archiving transactions in a shared public ledger or log on the Bitcoin network (known as the "block chain") to reflect the Bitcoin balances of its users. In addition, new Bitcoin are created with each update to the ledger.[2]

In this way, each Bitcoin represents an allocation of value which is recorded in the public ledger and exist as a chain of digitally signed transactions.

The algorithm that allows Bitcoin to be "mined" also controls the rate at which new Bitcoin are created. The number of new Bitcoin created in each update to the ledger is halved every few years, such that Bitcoin become more difficult to "mine" over time. By the Bitcoin algorithm's design, the maximum number of Bitcoin that can be mined is 21 million. [3] To date, over 17 million Bitcoin are estimated to be in circulation.

However, a Bitcoin can be "divided" into smaller units (currently up to 8 decimal points - a unit known as a "satoshi") and those smaller units can be transferred between users.

3.2 Phases of the Bitcoin system

The Bitcoin system may be broken down into the following "phases":

- Mining;
- Holding Bitcoin; and
- Transferring Bitcoin.

Each of these phases are discussed in turn below.

(a) **Mining**

In its May 2013 report to the United States Senate Committee on Finance, the United States Government Accountability Office described the mining process in the following terms:

> "Bitcoin are created and entered into circulation through a process, called mining, that members of the bitcoin network perform. To perform the work of mining, bitcoin miners

---

[1] United States Government Accountability Office, "Virtual Economies and Currencies: Additional IRS Guidance Could Reduce Tax Compliance Risks" (GAO-13-516), May 2013 (http://www.gao.gov/assets/660/654620.pdf, accessed 12 January 2014)

[2] "The Economist explains - How does Bitcoin work?", *The Economist* (11 April 2013) (http://www.economist.com/blogs/economist-explains/2013/04/economist-explains-how-does-bitcoin-work, accessed 12 January 2014).

[3] Ibid.

> download free bitcoin software that they use to solve complex equations. These equations serve to verify the validity of bitcoin transactions by grouping several transactions into a block and mathematically proving that the transactions occurred and do not represent double spending of a bitcoin. When a miner's computer solves an equation, the bitcoin network accepts the block of transactions as valid and creates 25 new bitcoin and awards them to the successful miner."[4]

As noted, the mining process involves the Bitcoin network updating a shared public ledger or log on the Bitcoin network:

> "The entire network is used to monitor and verify both the creation of new Bitcoin through mining, and the transfer of Bitcoin between users. A log is collectively maintained of all transactions, with every new transaction broadcast across the Bitcoin network. Participating machines communicate to create and agree on updates to the official log. This process, which is computationally intensive, is in fact the process used to mine Bitcoin…" [5]

(b) **Holding Bitcoin**

Each Bitcoin user has a digital wallet known as a "Bitcoin wallet" which incorporates a public/private key pair.

Bitcoin are recorded in the shared public ledger on the Bitcoin network as being under the control of a selected individual Bitcoin wallet and key pair. The wallet and key pair are the only way of accessing the Bitcoin. [6]

As such, Bitcoin are "owned" and "held" by the user who has control of the relevant Bitcoin wallet.

(c) **Transferring Bitcoin**

A Bitcoin wallet is also how Bitcoin are transferred between users. Specifically, Bitcoin are sent or received from one or more "Bitcoin addresses" associated with a Bitcoin wallet. A Bitcoin address is an alphanumeric string derived from the public key incorporated in a Bitcoin wallet.

The process of transferring a Bitcoin may be summarised as follows:[7]

- the recipient provides their Bitcoin address to the sender;
- the sender adds the recipient's Bitcoin address and the quantity of Bitcoin to be transferred to a "transaction" message;
- the sender's message is digitally signed by the sender's private key incorporated in the sender's Bitcoin wallet. The sender's public key is also announced to enable verification of the message;

---

[4] Note 1 above, at page 6. It should be noted that the rate of creation of 12.5 new Bitcoin is current rate at the time of this paper, and it is understood that the Bitcoin algorithm is designed to reduce that rate over time.

[5] Note 2 above.

[6] Bollen, Rhys, "The Legal Status of Online Currencies: Are Bitcoin the Future?" , (2013) 24 JBFLP 272 at 275.

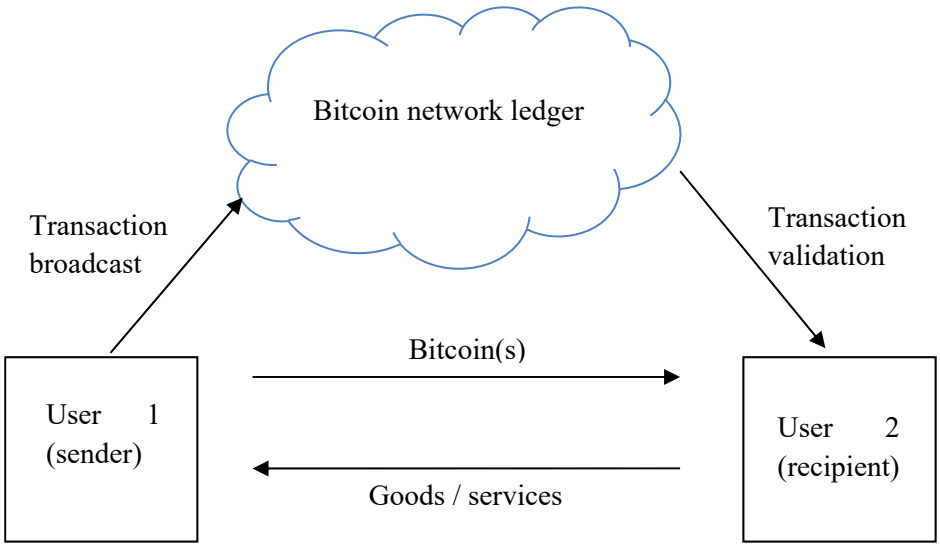
[7] https://en.bitcoin.it/wiki/Introduction accessed 13 January 2014; http://www.forbes.com/sites/investopedia/2013/08/01/how-bitcoin-works/, *Forbes*(1 August 2013), (accessed 13 January 2014).

- the "transaction" message is broadcast over the Bitcoin network and verified, and the Bitcoin transferred from the sender's address to the recipient's address.

Only the first two steps involve action by the sender and recipient. The latter two steps are automatically executed by the Bitcoin software and the Bitcoin network.[8]

Importantly, Bitcoin are transferred directly from the sender to the recipient. Bitcoin "miners" validate and update the transaction in the public ledger, but are not payment intermediaries.

A basic Bitcoin transaction may be shown in diagrammatic form as follows:



4. Functioning and modelling of nodes

A competing block or a transactional double spend can be modelled using 3-dimension Bond Percolation models. In either instance of a double spent transaction or a competing block, it is an issuing node that is in competition with an alternative issuing node. These two issuing nodes are competing for the same population of hosts as the receipt of a conflicting transaction results in the rejection of the later transaction in favour of the first to have been received by a node.

Hence, the receipt of a transaction that conflicts in a manner that would lead to a double spend is not possible following the initial receipt of the first transaction. In the case of competing blocks, the later blocks are placed into an 'orphan pool'. These are maintained in memory in case a longer chain is developed from one of these, but the primary chain selected results from the block that was received first if the chains are the same length.

In this manner, we can state that the receipt of a competing block or transaction will result in a form of 'cross-immunity' between the nodes as once a block or transaction has been accepted, the accepting node will no longer accept another transaction or block that references the same sources.

The dynamics of competition between nodes is, in general, complex. It will depend as to which initiating node starts the process. This is, which of the nodes releases a transaction or block first and how much of a head start this node can exhibit. In a case of the Selfish Miner Scheme, it can be easily

[8] Ibid.

noted that the Selfish Miner node will, by definition, always release a block after the public chain. As the Selfish Miner strategy requires that the node listens for the release of a competing block from the public chain, we can state that the Selfish Miner initiating node must necessarily issue their block after the public chain miners. The issue to be modelled in this case is a consequence of how long the head start from this release is.

In this paper, we will start by studying the case in which the two initiating nodes pass through the node population at time segregated intervals. Here, the first node releases a block or transaction that traverses the node population in an epidemic manner leaving some fraction of the node population immune to the competing transmission of the later initiating node.

A major question to be addressed is if the later transaction or block is able to effectively spread. If a sufficient segment and distribution of the node population is removed from the population of accepting nodes, then the spread of information from the second initiating node becomes infeasible or, at best, probabilistically unlikely. With this information, we can demonstrate that there will exist a threshold value of the bond occupation probability or 'transmissibility' for the first initiating node at which the distribution of the competing transaction or block becomes infeasible.

The 'co-existence threshold' coincides with a continuous phase transition that is similar in form to traditional SIS and SIR epidemic models. With the introduction of a competing initiating node, these two transitions become distinct with the co-existence threshold acting as a new aspect of the network topology model.

The transmission of information from the two competing initiating nodes can only occur in the intermediate zone that exists between the epidemic broadcast and co-existence thresholds of the competing processes. This this paper, we will determine the exact analytic calculation for a model of the Selfish Miner Scheme, the transmission of double spends and the related networks with the associated positions for the respective thresholds.

Bitcoin has been approximated as a scale-free network as all nodes will eventually have a copy of all valid transactions. However, as not all nodes will have received a copy of all transactions within a discrete time period related to the publication of a new block, we can state that Bitcoin is not truly scale-free. In a scale-free network, the 'epidemic threshold' of the first transaction or block from an initiating node is, by definition, zero. The co-existence threshold will not be zero however. A corollary of a scale-free network model of Bitcoin would be that all transactions would be received by all nodes within the censor time even if the transmissibility of each node was lowered significantly. For an approximation of the Bitcoin network however, we obtain the result that for two competing initiating nodes, altering the transmissibility can result in the eradication of one of the transactions or blocks, but never both.

The distribution of information across the network is in effect a model of contacts and exchanges between individual nodes. The Bitcoin network is represented as a graph in which the vertices are the individual nodes and the edges are the transmission of data between the nodes as they exchange a received valid block or transaction.

In this, only some connections between nodes will result in an exchange of information. We assume generalised (SEIR) Susceptible/Exposed/Introduced/Receiving (new) in which each transaction or block speeds across the edges of the network with a probability T that refers to the transmissibility of the system. As the nodes do not differentiate between the blocks or transactions prior to having received

them, we can make the assumption that the transmissibility is the same for both of the initiating nodes when started independently.

This form of dynamics can be modelled as a bond percolation process on the same graph where the bond occupation probability is set to the value assigned to the transmissibility. This connected cluster of vertices in this percolation process will, in this manner, associate to the node groups that would have accepted the transaction if initiated by any node as an initiating node in the cluster.

We see that for small values of T that only small clusters form and that the transaction or block will propagate slowly. We can also demonstrate that when the transmissibility exceeds a critical level, Tc, that an extensive spanning cluster or 'giant component' will form that corresponds to the universal dissemination of the transaction or block to nearly all connected nodes. Where such a 'giant component' forms, the initiating node will be able to reach all active nodes on the network within the range of the nodes that are associated with this cluster. The initiating node will thereby reach an extensive fraction of the node population. The value of T at which a giant component first forms is called the percolation threshold on the contact network.

In analysing the Bitcoin node network, we make the assumption that the class of network modelled has a specified degree distribution but that connections are otherwise random. This is defined as the case within the protocol. The degree of a vertex in a network is defined as the number of edges connected to that vertex. In the Bitcoin node network, most nodes maintain a standard connection pool of connections to 50 nodes, however, some 'super-connected' nodes can maintain connection pools in excess of 500 connections.

Let pk be the fraction of vertices in the network that have degree k. We can consider pk to represent the probability of a randomly selected vertex having degree k. Each vertex located at the conclusion of a randomly selected edge will have degree k where the probability is defined to be proportional to kp? as there will exist k – times as many edges connected to a vertex of degree k than to that of a vertex of degree I. Hence the probability that our edge is connected is a value multiplied by k.

4.1 Vertex modelling

We are most interested in the distribution of the number of edges on the followed vertex. Here the excess degree is one less than the total degree of the vertex and is expressed using the normalised distribution;

$$qk = \frac{(k+1,p_{k}+1)}{\Sigma_k k p_k} = \frac{(u+1)p_{u+1}}{2} \qquad \text{Equation (1).}$$

is the mean degree of vertices in the network.

Here: $2 = \langle k \rangle = \Sigma_k k_{Pk}$

In SIR models, the 'basic reproductive number', R_0 is defined to be the number of bodies that the 'infected' state is transmitted to. In Bitcoin, we define this value to represent the number of nodes that a transmitting node communicates to. In the default client implementation, R_0 = 50. This means that each node will pass a transaction to 50 other nodes on average. A node starts with 8 connections and as it communicates, collects more to build a value based on the power of the machine. Many large nodes hold over 1,000 edges on the Bitcoin network.

When a transaction or block is passed by a node through our network vertex, it can spread onto k other connected and hence neighbouring nodes.

Each transaction and block exchange is completed with probability T resulting in Tk additional nodes having the new block or transaction. When we average across the distribution of qk of k, we obtain the basic reproduction for the network as follows:

$$R_0 = T\sum_{k=0}^{\infty} kqk = \frac{T}{\langle k\rangle}\sum_{k=0}^{\infty} k(k+1)pk + 1 \qquad \text{Equation (2).}$$

$$= T\frac{(k^2)-\langle k\rangle}{\langle k\rangle}$$

Transactions will propagate $iff(T > Tc)$ where:

$$Tc = \frac{\langle k\rangle}{\langle k^2\rangle-(k)} \qquad \text{Equation (3).}$$

The critical value can be impacted and reduced in cases where a non-standard transaction is broadcast across the network and only a limited number of nodes accept the format of the transaction.

In calculating the spread of information across the network, we define the probability generating functions for pk and qk as follows:

$$F_0(x) = \sum_{k=0}^{\infty} P^{kx^k} \qquad \text{Equation (4).}$$

$$F_i(x) = \sum_{k=0}^{\infty} qkx^k = \frac{F_0'(x)}{z}$$

We define F_0' to represent the first derivative of F_0 WRT its argument.

Replacing for Eq. 3 we thus obtain:

$$Tc = \frac{1}{F_1'(1)} \qquad \text{Equation (5).}$$

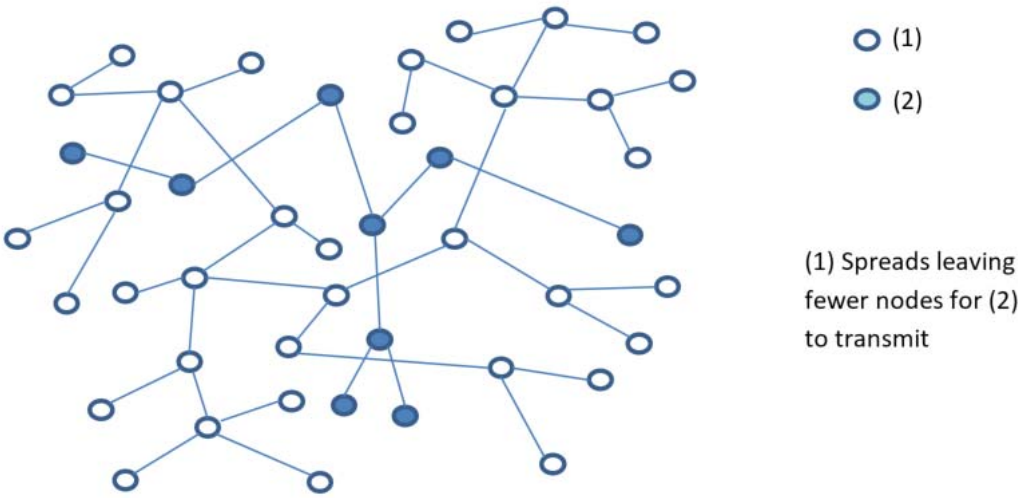


Fig 1 – Geomi / Paket Network model.

Let μ be the mean probability that a vertex has not received a new transaction or block that is not within its pool from a neighbouring vertex node in the prior time-period. This value equals the probability that a transmission of information did not occur between two adjacent vertices or nodes and is represented as $(1 - T)$ plus the probability that the nodes express sufficient connectivity, bandwidth and time in a receptive state but that node did not receive a new transaction or block from an initiating node. The probability that a vertex node did not receive a new transmission from an initiating node is equal to the probability that it also did not receive the transmission from any of its (k) other connected and neighbouring nodes. This value is expressed by μ^k with k being distributed using qk. From this we can see that the mean probability of a neighbouring node successfully receiving a transmission sourced from an initiating node is:

$$\sum_{k=0}^{\infty} q_{k\mu} k = F_1(\mu) \qquad \text{Equation (6).}$$

Hence, μ must satisfy:

$$\mu = 1 - T + TF_2(\mu) \qquad \text{Equation (7).}$$

From this we can derive the probability of a randomly chosen vertex or node having not successfully received and processed a transmission from the initiating node to be defined as:

$$\sum_{k=0}^{\infty} pk\mu^k = F_0(\mu) \qquad \text{Equation (8).}$$

where the proportion S of the vertices that have received, processed and accepted the transmission is:

$$S = 1 - F_0(\mu) \qquad \text{Equation (9).}$$

We can calculate the rate and extent of the propagation of a transmission to the node network at a given time instant by solving Eq. (7) for μ and then substituting the answer obtained into Eq. (9).

S may also represent the probability at a point in time of the transmission having reached the majority of nodes when the transmission begins from a single randomly interconnected node with probability $1 - S$.

Where S is considered to be an order parameter for our model, then the transition can be modelled as a continuous phase transition in the mean-field universality class for percolation.

In cases, such as a Selfish Miner scheme, where the initial transmission from the earliest releasing initiating node has a transaction or block that would be accepted by the majority of nodes on the network, we can consider that the initial propagation in the first time period of transmission prior to any transmission from the second initiating node will result in a fraction S of the vertices (nodes) being in a state where these nodes will reject the subsequent transmission from the later transaction or block.

In order to represent this mathematically, we need to delete the nodes from the network that have received the initial transmission and to no longer express these as vertices in our network model (see Fig. 1). In this case, the second transmission will only be successfully transmitted to a majority of nodes if the residual graph as a giant component and the transmissibility is great enough for this to subsequently spread.

In the Bitcoin network, we will not have a condition where $T = 0$ unless the initial transmission is considered invalid. If this does occur, no transmission results leaving the entire network to reach a consensus on the later transmission.

In the case where $T = 1$, the first transmission will have been received and be processed by the entire giant component of the graph. Once the giant component has received the transmission, the second transmission cannot spread. We can state this as for all large networks (and with over 5,000 nodes, the Bitcoin network is considered a large network) random networks can express only one giant component.

As such, the values for a given propagation time period where $0 < T < 1$ are of interest. Where T lies between these values, it would be expected to find a transition state. This state is what we seek to investigate. To do this, it is necessary to begin by deriving the degree distribution of the residual graph. As we know that this graph is uncorrelated, we can easily determine if it expresses a giant component.

4.2 Epidemic Models

We denote the number of nodes that are in a state where they receive a new transaction by Si, and the number of nodes that have received a block or transaction and are propagating it by It. In the idealised model $Xi = (Si, It)$ performs a Markov chain process on $\left(\mathbb{Z}^+\right)^2$ with the following transition rates:

$$g(s,i)(s-i,i+1) = \lambda si, g(s,i)(s-i,i-1) = \mu i \qquad \text{Equation (10).}$$

For some $\lambda, \mu \in (D, \infty)$.

As $Si + Ti$ does not increase, we exhibit a finite state space. In this space, the states $(S,0)$ for $S \in \mathbb{Z}^+$ are all absorbing and all other states remain transient. In this idealised model, we can treat all of the communicating classes as singletons. Even without an external transaction, the transaction propagation must complete with either:

- Nodes supporting a transaction
- Nodes that have rejected a transaction
- Nodes able to communicate via the network to all other viable nodes.

Nodes that are inadequately powered are excluded. The absorption probabilities provide the distribution of the remaining nodes in a survival model. They can also be used in modelling the number of nodes who do not receive and transmit a transaction in a particular block. This is a simple calculation whenever So and Io is small.

This can be extended to where we have a large node population of size N. Here, we can consider the proportions $St^n = \frac{St}{N}$ and $it^n = \frac{Ii}{n}$ and extend this for $\lambda = \frac{\nu}{N}$, where we define ν as being independent of N. Taking a sequence of models as $N \to \infty$ we choose $So^n \to So$ and $io^n \to io$. From this, it is not too difficult to demonstrate that as $N \to \infty$ the process $\left(Si^n, it^n\right)$ converges to the solution (St, it) of the following differential equations:

$$\left(\frac{d}{dt}\right)St = -\nu St\ it$$
$$\left(\frac{d}{dt}\right)it = \nu St\ it - \mu it$$

Equation (11).

Starting from (So, io), convergence here represents:

$$\mathrm{E}\left[\left|\left(St^n, It^n\right) - \left(St, It\right)\right|\right] \to 0\ \forall\ t \geq 0$$

Equation (12).

We can further consider the instance at which:

$$\delta o = N - 1, Io = 1, \lambda = \frac{1}{n} \text{ and } \mu = 0$$

Equation (13).

In this case, we have a transaction that has been sent to a single node. This node attempts to transmit this transaction to any other node it connects to. We can model the connections between nodes as randomly distributed continuous time intervals with a Poisson process rate of $\lambda = 1$ for a simplified example.

At the point where i nodes have a copy of the transaction, $N - i$ nodes have not received a copy. So the rate at which the transaction or new blocks is propagated around the entire network is defined by:

$$gt = \frac{i(N-i)}{N}$$

Equation (14).

The expected time to elapse before all nodes know the transaction is:

$$\sum_{i=1}^{N-1} gi^{-1} = \sum_{i=1}^{N-1} \frac{N}{i(N-1)} = \sum_{i=1}^{N-1}\left(\frac{1}{i} + \frac{1}{N-1}\right) = 2\sum_{i=1}^{N-1}\frac{1}{i} \sim 2\log n \text{ as } N \to \infty$$

Equation (15).

The result is not a limit above, but rather an asymptotic equivalence. The fact that the expected time increases with N is related to the fact that we don't scale Io with N. From this we see that when a small number of poorly connected nodes hold a transaction, and when nearly all nodes have a copy of a block or transaction, the node's propagation to the remaining nodes in a Gossip network occurs very slowly.

We can now extend this model to account for censoring and the fact that when a block is discovered, the block will reset and restart the process.

Each block received by a node is validated and checked for veracity. When two blocks are mined at nearly the same time, the first block to be received by a node is included and is verified. The next block will be placed into the "orphan pool" and is validated after the initially received block. This orphan block is only added to the main chain for the node, if a further block is added making this a longer chain.

Each node verifies a block before it propagates it to the connected peer nodes. In this way only valid blocks are propagated, and any invalid blocks are quickly isolated. The Bitcoin Core client lists all of the validation requirements in the following functions:

- CheckBlock
- CheckBlockHeader

In validating a block, the node will ensure that:

- The data structure of the block is formatted correctly.
- The block header hash is less than or equal to the set target difficulty for the two-week period.
- The timestamp on the block is less than 120 minutes into the future. This allows for some time drift on mining nodes.
- The block size is less than or equal to the value set in the MAX_BLOCK_SIZE parameter.
- The initial transaction is a coinbase generation transaction. This is transaction zero for the block, and pays the winning miner.
- No other coinbase transaction is present in the block.
- Each transaction in the block is valid. Each transaction, as well as its hash, must be individually checked and validated.

Next the block is validated to ensure that:

- The transaction index validity is correct.
- The transaction hashes are correct.
- Spend txouts have been spent in the main chain.
- All prevouts are marked as spent.
- Spent txouts are spent by a valid transaction that consumes them.

As UTXO caching saves received transactions, the validation of each transaction should be fast as long as it has been received, cached and saved in the pool. The case where a transaction is not in the UTXO cache slows the process. The node requests a copy of the transaction from the network, validates it and adds this to the hash checks.

5. Epidemic models of block propagation

We have used the notation from Elveback et al (1964) and transcribed this to fit the model of information dissemination in the Bitcoin protocol. Information is transmitted through the Bitcoin node structure via the gossip protocol in the form of a SEIR model.

We define the status that exist when competing information is broadcast over the network in table 1. Information can be in competition for a couple of reasons in the Bitcoin protocol. A double spent transaction can lead to two or more UTXO addresses being simultaneously allocated to multiple transactions.

Table 1: State Acceptance in Block Propagation

State	Description
1	Accepts transaction A or B.

2	Processes transaction A and validates it. Rejects transaction B.
3	Processes transaction B and validates it. Rejects transaction A.
4	No longer broadcasting transaction A or B, but may send the transmission if a request a made.
5	Acceptive to transaction A, and temporarily rejects transaction B.
6	Transmits transaction A, and rejects transaction B.
7	Acceptive to transaction B, and temporarily rejects transaction A.
8	Transmits transaction B, and rejects transaction A.

In a double spend, a client attempts to spend the same ledger entry in two places, and to separate end addresses, at the same time. The nature of the protocol is such that only one of these competing transactions can be allocated and recorded into the blockchain. Once an amount has been removed from the UTXO pool, it cannot be used again.

The other form of competing information exchange is a conflicting block transmission. Here two or more valid blocks have been mined and released to the network at approximately the same time. This can occur when two miners (specialised nodes designed to solve the Bitcoin protocol puzzle) each find a solution to the current block, and release this solution within a time period that is sufficiently close, such that the block has not had sufficient time to be accepted throughout the entire node network.

In each instance, a proportion of nodes on the network accept one of the competing transactions or blocks to the exclusion of the other. Where a transaction is involved, a node will reject any competing transactions outright. In the case of a block, the competing block that arrives at a node will be added to the "orphan block" pool, and held in case the later received block grows to become the main chain. The core implementation is set to accept the first transaction or block that is received, and to take this as the master that it will work from. The timestamp in a block is not used in the decision to keep a block, unless it is more than two hours out of sync from the receiving node. In this case, the block will be rejected. The number of nodes in state $i\left(i=1,\ \ldots,8\right)$ is denoted by ηi.

We will define the variables as follows:

- λ as the rate of which nodes are added to the network.
- μ as the rate of which nodes are removed from the network.

These are in addition to the transmission between the states listed in Table 1.

When a node joins the network, it starts at state 1. The rate at which nodes leave the network is independent of the state it exists at when it leaves the network (i.e. $i=1,\ \ldots,\ 8$). The Bitcoin network has grown large and resilient enough that we can mostly ignore these values and assume a negligible removal rate during any individual block. We have simplified the state table in Figure 1 by not incorporating the transitions for new nodes, or for nodes that are removed permanently from the network.

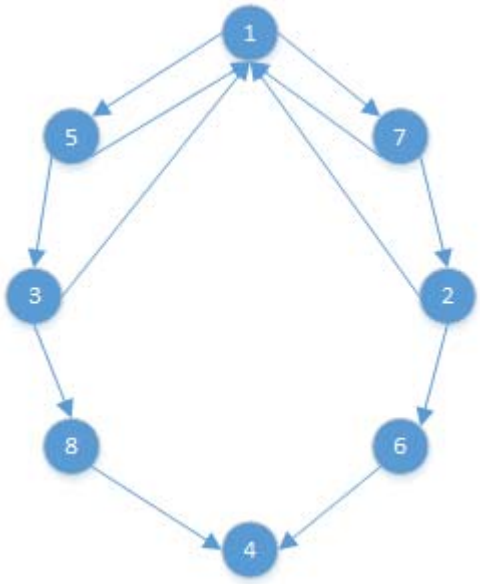


Figure : 2

States 5 and 7 will return to state 1 iff the transmission is interrupted. States 2 and 3 will return to state 1 iff the received transaction does not validate on that node and is rejected.

For the purposes of our model, we can ignore the state transitions 5 to 1, 3 to 1, 7 to 1, and 2 to 1, as we are not modelling invalid transactions or blocks. We are focused on analysing valid, but not competing transmissions.

We can model these states using the system of differential equations:

$$\frac{dn_1}{dt} = \lambda - \left[\beta_A\left(n_5 + n_6\right) + \beta_B\left(n_7 + n_8\right) + \mu\right]n_1$$

$$\frac{dn_2}{dt} = \gamma_B n_7 - \left[\beta_A\left(n_5 + n_6\right) + \mu\right]n_2$$

$$\frac{dn_3}{dt} = \gamma_A n_5 - \left[\beta_B\left(n_7 + n_8\right) + \mu\right]n_3$$

$$\frac{dn_4}{dt} = \gamma_A n_6 + \gamma_B n_8 - \mu n_4$$

$$\frac{dn_5}{dt} = \beta_A\left(n_5 + n_6\right)n_1 - \left(\gamma_A + \mu\right)n_5$$

$$\frac{dn_6}{dt} = \beta_A\left(n_5 + n_6\right)n_2 - \left(\gamma_A + \mu\right)n_6$$

$$\frac{dn_7}{dt} = \beta_B\left(n_7 + n_8\right)n_1 - \left(\gamma_B + \mu\right)n_7$$

Equation (16.1 – 16.8).

$$\frac{dn_8}{dt} = \beta_B \left(n_7 + n_8 \right) n_3 - \left(\gamma_B + \mu \right) n_8$$

We define β_A and β_B to be the respective transmission rates of the two transactions at a defined point in time. These values would be expected to remain the same if both transactions were released to the same number of nodes at precisely the same time. This is not necessarily the case for the following reasons:

- If node A maintains more connections to other nodes in the network than does node B, we see that it will initially transmit its information across the network at a higher rate.
- If node A starts transmitting information earlier than node B, the number of secondary nodes transmitting this information at a subsequent point in time will increase the propagation rate of the node respectively, due to the advantage of sending to more seed nodes.

We define the rate to which a node becomes immune to receiving a competing transmission as γ_A and γ_B This rate refers to the rate at which a node will accept and process a transaction moving the node to a state where any future competing transaction will be rejected.

In the case where a transaction or block is rejected out of hand due either to all responsive nodes having already accepted a competing transmission, or the transaction or block being rejected for protocol violations, we can set the respective transmission rate to zero. For instance, $\beta_B = 0$ if no node remains responsive following the transmission by A. In this case, the system equation 1 is reduced to:

$$\frac{dn_1}{dt} = \lambda - \left[\beta_A \left(n_5 + \mu \right) \right] n_1$$

$$\frac{dn_5}{dt} = \beta_A n_5 - \left(\gamma_A + \mu \right) n_5$$

$$\frac{dn_3}{dt} = \gamma_A n_5 - \mu n_3$$

Equation (17 a - c).

If we sum the equations from equation 1 and let:

$$n = \sum_{t=1}^{8} n_i$$

Equation (18).

We obtain the following linear equation:

$$\frac{dn}{dt} = \lambda - \mu n$$

Equation (19).

As $t \to \infty$ the size of the population will tend towards the limit:

$$n^* = \lambda / \mu$$

Equation (20).

This allows us to use the relative size of the variables $n_i,\ i = 1,\ ...,\ 8$ as:

$$\mu_i = {n_i}/{n^*}$$ Equation (21).

We will further assume that $n(0) = n^*$ as:

$$\sum_{i=1}^{8} \mu_i = 1$$ Equation (22).

By substituting equation 7 into equation 1 we see that:

$$\frac{dn_1}{dt} = \mu - \left[\beta_A n^* (\mu_5 + \mu_6) + \beta_B n^* (\mu_7 + \mu_6) + \mu\right]\mu_1$$

$$\frac{dn_2}{dt} = \gamma_B \mu_7 - \left[\beta_B n^* (\mu_5 + \mu_6) + \mu\right]\mu_2$$

$$\frac{dn_3}{dt} = \gamma_A \mu_5 - \left[\beta_B n^* (\mu_7 + \mu_8) + \mu\right]\mu_3$$

$$\frac{dn_5}{dt} = \beta_A n^* (\mu_5 + \mu_6)\mu_1 - (\gamma_A + \mu)\mu_5$$

$$\frac{dn_6}{dt} = \beta_A n^* (\mu_7 + \mu_\partial)\mu_2 - (\gamma_A + \mu)\mu_6$$

$$\frac{dn_7}{dt} = \beta_B n^* (\mu_7 + \mu_8)\mu_1 - (\gamma_B + \mu)\mu_7$$

$$\frac{dn_8}{dt} = \beta_B n^* (\mu_7 + \mu_8)\mu_3 - (\gamma_B + \mu)\mu_8$$

Equation (23.1 - 23.8).

6. Equilibrium

We can state that the system will resolve to an equilibrium state where only one of the transactions in transmitted, or is valid.

If we let:

$$R_A = {\beta_A n^*}/{(\gamma_A + \mu)}$$

$$R_B = {\beta_B n^*}/{(\gamma_B + \mu)}$$

$$q_A = \frac{\mu}{\gamma_A + \mu}$$

$$q_B = \frac{\mu}{\gamma_B + \mu}$$

Equation (24).

Where R_A and R_B represent the propagation rates at the transmission of information from either the initiating node A or B respectively, and allow this to refer to the mean number of accepted transmissions that result from the initial propagation in the first time period in a population that has not received any transmissions from either initiating node A or B for a node population of total size n^* at a given time.

The values q_A and q_B are defined to represent the proportion of the transfer period ($\frac{1}{(\gamma_A+\mu)}$ and $\frac{1}{(\gamma_B+\mu)}$ respectively) with respect to the censored time expectancy for the nodes. Here μ equals 600 seconds[9] is defined in the protocol if a linear hashing power is maintained.

Where we have $R_A > 1$ and $R_B > 1$, we obtain the values $G_2(G_3)$ as follows:

$$G_2 : (\mu_1, \ldots, \mathrm{u}_8) = \left(\frac{1}{R_A}, 0, \left(1-\frac{1}{R_A}\right)(1-q_A), \left(1-\frac{1}{R_A}\right)q_A, 0, 0, 0\right)$$
$$G_3 : (\mu_1, \ldots, \mu_8) = \left(\frac{1}{R_B}, \left(1-\frac{1}{R_B}\right)(1-q_B), 0, 0, 0, \left(1-\frac{1}{R_B}\right)q_B, 0\right) \quad \text{Equation (25).}$$

If we let:

$$\lambda_A = \beta_A n^* (\mu_5 + \mu_6)$$
$$\lambda_B = \beta_B n^* (\mu_7 + \mu_8) \quad \text{Equation (26).}$$

We can state that $\lambda_A(\lambda_B)$ is set to refer to the state where B follows a transaction released by A. In this instance, the components of G_4 can be determined through a process of successive diminution and substitution of the system outlined in equation 8. The states in this system can be reduced through the following state equation:

$$Y\mu = T_1 + p_{15}\left\{T_5 + p_{53}\left[T_3 + p_{38}\left(T_8 + p_{84}T_4\right)\right]\right\} + p_{17}\left\{T_7 + p_{72}\left[T_2 + p_{26}\left(T_6 + p_{64}T_4\right)\right]\right\} \quad \text{Equation (27).}$$

In this equation p_{ij} forms the probability measuring the rate of a transmission from state i into state j. Here the state transition times are provided by:

[9] On average based on a Poisson model of hash solutions.

$$
\begin{aligned}
T_1 &= \frac{1}{(\lambda_A + \lambda_B + \mu)} \\
T_2 &= (\lambda_A + \mu)^{-1} \\
T_3 &= (\lambda_B + \mu)^{-1} \\
T_4 &= 1 \big/ \mu \\
T_5 &= (\gamma_A + \mu)^{-1} \\
T_6 &= (\gamma_A + \mu)^{-1} \\
T_7 &= (\gamma_B + \mu)^{-1} \\
T_8 &= (\gamma_B + \mu)^{-1} \\
P_{15} &= \lambda_A \big/ (\lambda_A + \lambda_B + \mu) \\
P_{53} &= \lambda_A \big/ (\lambda_A + \mu) \\
P_{38} &= \lambda_B \big/ (\lambda_B + \mu) \\
P_{84} &= \lambda_B \big/ (\lambda_B + \mu) \\
P_{17} &= \lambda_B \big/ (\lambda_A + \lambda_B + \mu) \\
P_{72} &= \lambda_B \big/ (\lambda_B + \mu) \\
P_{26} &= \lambda_A \big/ (\lambda_A + \mu) \\
P_{84} &= \lambda_A \big/ (\lambda_A + \mu)
\end{aligned}
$$

Equation (28).

From this we can calculate the quantities u_i for $(i = 1, \ ..., \ 8)$:

$$
\begin{aligned}
u_1 &= \mu \big/ \lambda_A' + \lambda_B + \mu \\
u_2 &= \mu\lambda_B \, \gamma_B \big/ \left[(\lambda_A + \lambda_B + \mu)(\gamma_B + \mu)(\gamma_A + \mu)\right] \\
u_3 &= \mu\lambda_A \, \gamma_A \big/ \left[(\lambda_A + \lambda_B + \mu)(\gamma_A + \mu)(\gamma_B + \mu)\right] \\
u_4 &= \frac{\lambda_A\lambda_B\gamma_A\gamma_B}{\lambda_A + \lambda_B + \mu(\gamma_A + \mu)(\gamma_B + \mu)}\left(\frac{1}{(\lambda_A + \mu)} + \frac{1}{(\lambda_B + \mu)}\right)
\end{aligned}
$$

Equation (29).

Substituting for $u_{1,}$ u_2 and u_3 into the equations for λ_A and λ_B we get:

$$R_A\left(u_1+u_2\right)=1$$
$$R_B\left(u_1+u_3\right)=1$$

Equation (30).

From which we obtain:

$$\left(\lambda_A+\lambda_B+\mu\right)\left(\lambda_A+\mu\right)=\mu R_A\left(\lambda_A+\lambda_B P_B+\mu\right)$$
$$\left(\lambda_A+\lambda_B+\mu\right)\left(\lambda_B+\mu\right)=\mu R_B\left(\lambda_A P_A+\lambda_B+\mu\right)$$

Equation (31).

Here:

$$p_A=1-q_A+p_B=1-q_B$$

Equation (32).

These equations are defined in Figure 2 where we can differentiate to obtain the related maxima and minima. We can also determine the intersection of these sets for a solution of the positive values of λ_A and λ_B.

$$B_1=\left\{R_B:R_A\succ\frac{R_A}{\left(1+\left(R_{A-1}\right)pa\right)}\right\}$$
$$B_2=\left\{R_A:R_B\succ\frac{R_B}{\left(1+\left(R_{B-1}\right)pb\right)}\right\}$$

Equation (33).

Reducing this we obtain:

$$\lambda_A^{(t+1)}=\mu\left[\frac{R_A\left(\lambda_A^{(t)}+\lambda_B^{(t)}pb+\mu\right)}{\left(\lambda_A^{(t)}+\lambda_B^{(t)}+\mu\right)}-1\right]$$
$$\lambda_B^{(t+1)}=\mu\left[\frac{R_B\left(\lambda_A^{(t)}pa+\lambda_B^{(t)}+\mu\right)}{\left(\lambda_A^{(t)}+\lambda_B^{(t)}+\mu\right)}-1\right]$$

Equation (34).

Using these equations, we can find the solutions of the equilibrium points $G_k, k=1,\ ...,\ 4$ as long as the Eigen values of the matrix with the elements:

$$\left\{\frac{\partial f_t\left(G_k\right)}{\partial u_i}\right\}$$

Equation (35).

Have positive components. In this ft represents the R.H.S. of the equation for Ui. We can now use this to demonstrate that G_2 has a local stability at:

$$R_B<\frac{R_A}{\left[1+\left(R_A-1\right)P_A\right]}$$
$$R_A>1$$

Equation (36).

Using these values, the heterogeneous spatial distribution comes from the differential of the transmission rates from R_A and R_B, where the population is composed of a group of mutually isolated sub-populations. The temporal sequence of the two transmissions allows us to create a model of the dynamic behaviour of these transmissions, and the manner on how small timing on interaction based oscillations impact the contact rates.

Using the mean difference from the release times for the transmissions of form A(B), which we denote as $L_A\left(L_B\right)$, we can deduce:

$$L_A = {}^{1}\!\!\diagup\!\!{}_{\lambda_A} + \left[\frac{\lambda_B}{\left(\lambda_A + \lambda_B\right)}\right] / \gamma_B$$

$$L_B = {}^{1}\!\!\diagup\!\!{}_{\lambda_B} + \left[\frac{\lambda_A}{\left(\lambda_A + \lambda_B\right)}\right] / \gamma_A$$

Equation (37).

We can determine the domain of co-existence for these competing transmissions, and from this use the relationships to evaluate the set of conditions that can probabilistically result in cases where competing transmissions could lead to states, where a later transmission leads to a scenario where the primary transmission is orphaned unexpectedly.

From this we also see that the attacker needs to create a scenario where the attacker invests in a one-time attack.

In the case of a selfish miner attack, the success of the second competing block being accepted by a significant (even if minor) percentage of nodes, comes as a relationship between the transmissibility of the initially released transaction and the subsequent competing block, following an acknowledgement and receipt from the initial block transmission at the selfish miner node.

As the selfish or dishonest miner cannot advertise the attempted economic attack, they must control release to associated nodes and not distribute this within a pool.

7. Competing Epidemics model of nodes

We can model both transaction and block transmission in the Bitcoin network using an SEIR epidemic model. In this, we have the following states:

S – The fraction of nodes that have not received a block or transaction but are in a state to receive one.

E – This is the fraction of nodes that have received a new transaction or block and are validating it to ensure its veracity.

I – This the fraction of nodes that are able to forward a validated transaction or block.

R – This is the faction of nodes that have transmitted a transaction or block to all connected nodes and are no longer forwarding. This includes nodes that have received a second or later block in a contagious chain.

The fraction of nodes is normalised such that:

$$S + E + I + R = 1$$

We can also state that the process progresses via the following states:

$$S \rightarrow \mathcal{E} \rightarrow I \rightarrow R$$

We can define the node network to have the following modification rates:

u - Average rate of existing nodes that are leaving the network

β - The contact rate averaged across the network

βi - The contact rate for node i

B - The average rate for new nodes to join the network

Further, we define the following variables for the network propagation:

$1/\varepsilon$ - Average latency period. This is the time an average node takes to verify a transaction or block and to include it in its list of valid forwarding processes.

$1/y$ - The average period where a node will forward a transaction or block

From this we can also define the following network properties:

N : Total population of nodes at a given time

R_0 : The basic reproduction number for the propagation of a block or transaction

7.1 Reproduction Number

A threshold quantity exists which can be used to determine whether the transmission of a transaction will propagate throughout the entire network or if it will 'die out' by a process where it is not transmitted to a sufficient volume of nodes in a sufficient amount of time.

We define this as the basic reproduction number R_0, which we define as the number of secondary transmissions that result from a single transmitting node comprised entirely of nodes in a susceptible state ($S(\theta) = N$) over the course of the propagation from this single node of the transaction or block.

The transmitting node makes β connections per unit time which results in the acceptance of the block or transaction at the connected node at a mean transmission period of $1/y$. Hence, we can state that the reproduction number is:

$$R_0 = \beta/y \qquad \text{Equation (38).}$$

This value quantifies the propagation of a new transaction or block through the node network.

In the case where the basic reproduction number falls below one, ($R_0 < 1$) which is a state where the transmitting node makes too few connections to pass the transaction or block on in the specified life span of the transaction or block, the transmission 'dies out'.

Here, the life span of a block is set by the time until a subsequent second block is added to the chain confirming the first and creating a longer chain.

The life span of a transaction is undefined within the protocol. It is set through the node's memory pool. Most nodes will remove an unconfirmed transaction from their unconfirmed transaction pool if it has not been successfully included in a block within 3 to 4 days. At this point, the transaction can be resent.

It is important to note that even if a transaction is not received by all nodes within a single block period, the inclusion of the transaction into a block at any point means that it will be requested by all nodes receiving that block. This comes about as the nodes validate all of the hashes that are received in a given block.

Where a node has received a block, it will validate this block for possible inclusion by checking the hashes of all transactions stored within the block. In order for a node to successfully complete this process, it will need to obtain a copy of all transactions that have been included in the block and to validate the accuracy of these transactions by hashing the transaction and checking the hashes of all transactions reported.

In the case where $R_0 > 1$, the propagation of transactions or blocks across the node network forms an epidemic response over the node population. In those cases where $R_0 = 1$, the propagation of transactions or blocks becomes endemic. This refers to a state where a transaction remains, throughout the node, set at a constant rate.

Where we have instances of different transmission rates between nodes, the basic reproduction number can be calculated as the sum of the reproduction number for each node or node form on the network.

A simple model can occur where there are two levels of node propagation, here:

$$R_0 = R_{0\text{FAST}} + R_{0\text{SLOW}} \qquad \text{Equation (39).}$$

If we need to more accurately model this state, we can sum the propagation rates of each node (i) in our network population N, here:

$$R_0 = \sum_{i=1}^{N} R_0 i \qquad \text{Equation (40).}$$

In this model, $R_0 i$ is the individual propagation rate that is associated with node (i).

This model is similar to an SIR model of epidemics. The propagation of transactions and blocks within the bitcoin network differs from a simple SIR model as an SIR model requires each node to transmit a received transaction or block immediately. As a node is required to validate the transaction or block prior to transmitting it, we have a period that we can model as a latent or exposed phase. In this state, the node has accepted receipt of a transaction but is not forwarding it to other nodes as it has not yet validated it.

As such, we can segregate the node population into four states: S.E.I.R. as we defined above.

The number of nodes in each state or the densities of these nodes are denoted respectively by:

$$S_{(t)}\ E_{(t)}\ I_{(t)} \text{ and } R_{(t)}$$

This can be expressed as:

$$N = S_{(t)} + E_{(t)} + I_{(t)} + R_{(t)} \qquad \text{Equation (41).}$$

Where:

$$\frac{ds}{dT} = \text{B - } \beta\text{SI} - u\text{S}$$

$$\frac{dE}{dT} = \beta\text{SI} - (\varepsilon + u)\text{ E}$$

$$\frac{dI}{dT} = \varepsilon\text{E} - (y + u)\text{ I} \qquad \text{Equation (42).}$$

$$\frac{dR}{dT} = y\text{ I - } u\text{R}$$

Several assumptions exist within this model and in the formulation of the equations. First, we have assumed that each node has an equal probability to all other nodes, of transmitting a valid transaction with a rate of β, which is considered the contact rate of the network. Hence a node with a validated transaction will contact other nodes in its connection pool at a rate of βN other nodes being contacted per unit time. The fraction of contacts from a node transmitting a valid transaction with a node in a receiving state (here represented as $S_{(t)}$) is represented by the relationship $^{S}/_{N}$.

The number of new nodes that have accepted the transaction for verification in unit time per transmitting node is βN ($^{S}/_{N}$) providing the rate of transmission of a propagating valid transaction as:

$$\beta N\ (^{S}/_{N})\ I = \beta\text{SI} \qquad \text{Equation (43).}$$

To drive the 2nd and 3rd level equations, we consider the population of nodes that are leaving the state S to be equal to the number of nodes entering state E (Braver & Castillo – Chavez, 2001).

Once the node has successfully validated the transaction, it moves into state I.

We use a condition known as the 'Law of Mass Action' to simplify the calculations and assume that the rate of contact between the classes of nodes at any given unit time in the node population is proportioned to the size of the groups at that moment (Daley & Gani, 2005).

Next it is assumed that the rate of transaction propagation ($S_{(t)}\ E_{(t)}$) is much faster than the time scale of nodes entering or leaving the network population.

These factors can be ignored in the simplified model when these assumptions are maintained.

This is a stochastic model in that the propagation is deemed to form a random pattern.

The model is deterministic in that a set path from S → E → I → R is followed in all cases. Each state of a transaction being propagated and validated can be isolated and modelled independently.

These transition rates from one class to another are mathematically expressed as derivatives. As such, each of the models can be formulated using differential equations. In building the models presented, it must be assumed that the size of the node population in a compartment or class is differentiable W.R.T. time and that the transmission process is deterministic between the nodes. Hence, we can say that the changes to the ratio of nodes in one class out of the node population to another can be calculated using only the history of the process in the development of the model (refer (1) above).

We can further extend this approach using discrete analysis on a lattice (ie. Using a 2-dimensional matrix) where the updates and class/state changes are enacted through a process of asynchronous single-site updates. This can be conducted using a process known as 'Kinetic Monte Carlo' or through synchronous updating in a process referred to as 'Cellular Automata'.

This lattice or matrix approach enables the calculation of inhomogeneities and for the introduction of statistical clustering.

Where the variable β depends on time, we can express it as a 3-dimensional non-autonomous system. This can be converted into an autonomous system in four dimensions (Aron & Schwartz, 1984).

8. Conclusion

This research is the initial stage of a series of efforts to present the mathematics of peer to peer cryptocurrencies and in particular, Bitcoin. In these, we shall demonstrate that an SEIR model of mining nodes and a SEIR-C model with wallets that save the Blockchain and do not mine can be mathematically constructed. The planned deliverable will incorporate a software package that enables the testing of conditions and changes to the network.

Even at a low distribution rate from poorly connected nodes with low bandwidth, we can model the maximum time for a transaction to be broadcast across the Bitcoin network. In knowing the conditions that can lead to a low intensity of distribution, these can be modelled so that you can find the optimum strategies to propagate transactions widely without long delays.

The presence of double spends, and transactions that do not conform to the protocol rules, can result in propagation delays. Similarly, the introduction of an increased block size will stress many existing nodes beyond the existing capacity.

The decline of the number of nodes that are incapable of scaling to the new limits, and the impact on the network as these nodes are stressed, can be explained using a series of Markov chain epidemic models. We will define and model the decline in low powered nodes to the function of individuals who are susceptible to transaction acceptance (infection). In these models, an infective will (in simple terms) either die or recover with a resistance to further infection. This use of SIER-C epidemic models for transaction and Block propagation will allow for the more effective implementation of systems and nodes within the cryptocurrency.

9. References

(1) Aron, J.L. & Schwartz, I.B (1984) 'Seasonality and Period Doubling Bifurcations in an Epidemic Model', Theoretical Biology, 110: 665-679
(2) Braver & Castillo – Chavez, 2001, 'Mathematical Models in Population Biology and Epidemiology', NY Springer.
(3) Daley, D.J. & Gani, J (2005) 'Epidemic Modelling: An Introduction' NY, Cambridge University Press.
(4) Elveback, L., Fox, J.P., Varma, A. (1964) "An extension of the Reed-Frost epidemic model for the study of competition between viral agents in the presence of interference" American Journal of Hygiene V.80 P 356-364.