Block delivery time in Bitcoin distribution network

Jelena Mišić[1], Vojislav B. Mišić[1], Xiaolin Chang[2], Saeideh G. Motlagh[1], and M. Zulfiker Ali[1]
[1]Ryerson University, Toronto, ON, Canada
[2]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation
Beijing Jiaotong University, Beijing, China

***Abstract*—In this work we provide comprehensive analytical model for Bitcoin distribution network. We apply Jackson network model on the whole Bitcoin network where individual nodes operate as priority M/G/1 queuing systems. Data arrival process to the nodes is modeled as a non-homogeneous Poisson process in which the data arrival rates to the nodes are derived from the analytical model of gossip data delivery protocol. This model considers random probability distribution of node connectivity. Performance results include network distribution time for blocks, node response time for blocks, and populations of data distribution algorithms as functions on network size. Usefulness of this model is demonstrated by efficiently computing the forking probability for the Bitcoin blockchain.**

I. Introduction

Current applications of blockchain technology are mostly focused on financial area, namely on the implementation of replicated ledgers holding financial data such as Bitcoin [1], [2], [14], [15]. The distributed ledger is implemented over a group of connected processing nodes. Ledger data is stored at each node as single linked list of data blocks that contain linked data primitives known as transactions. Transactions arrive at individual nodes which have to verify transaction inputs versus outputs, i.e., whether the value of all claimed inputs is equal to or larger than the sum of new outputs [18], before distributing those transactions to the network for further verification by other nodes.

Transactions verified by a node are linked into one data block as soon as that node has found a nonce (i.e., a random number) which is inserted in the block header with the goal of obtaining the block hash with the required number of leading zeros; this approach is often referred to as proof of work, or PoW. Similar to transactions, the block is distributed to the network for further verification; the node that has created the block is known as a block miner. Upon receiving a new block nodes which were in process of verifying the same set of transaction will abandon their efforts, verify the newly arrived block and insert it into their ledgers, and continue transaction verification from the pool of unverified transactions. To regulate the rate of block mining, the Bitcoin protocol regulates the difficulty of computing the nonce in the block header so that new block can be mined once every 10 minutes on the average [15].

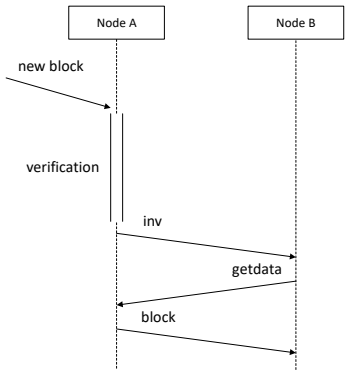
Long delays in forwarding the blocks or transaction may open many security vulnerabilities [6], [5], [19], [17], [16], [11], [3] which affect the consistency of the ledger. Perhaps the most important one is forking which occurs when two (or more) nodes mine and transmit the block in the time window when neither of the blocks has completed distribution through the network. In this case some nodes will mount one or the other block as the head of the blockchain, leaving the distributed ledger in an inconsistent state. Forking can be used for security attack if the attacker can generate blocks at a sufficiently high rate and can, thus, build its own blockchain extension to take over blockchain head. For such situations, fast propagation of transactions and blocks is of paramount importance.

In this work we focus on a detailed analytical model of Bitcoin's blockchain P2P network, data distribution (gossip) algorithm, and queuing issues at individual nodes. Purpose of this model is to find accurate probability distributions of block and transaction delivery times in the Bitcoin network. Such model will enable us to analyze probabilities of success of various situations like forking, double spending attack and many others. Analytical model of data distribution algorithm for the Bitcoin network is crucial for finding data distribution delays and for analysis of security attacks. Variability of the number of TCP connections among the nodes affect data distribution algorithm and introduce variability of data rates coming to nodes. This introduces non-homogenous Poisson process of data arrivals at network nodes. This issue affects further queuing analysis of node which needs to be done in M/G/1 manner due to unknown properties of service time [20]. Another problem is that block and transaction traffic need to be separated in different queues and handled in priority order. Solving all these issues allows analysis of the Bitcoin network using the Jackson network approach [13]. In this paper, due to space limitations, we present the analysis for block traffic only, with elements of transaction traffic analysis used where necessary.

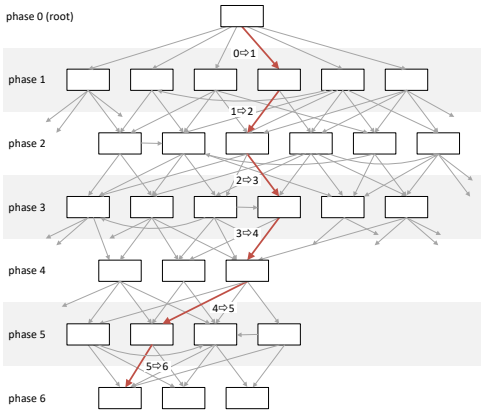
The paper is organized as follows: Section II develops the model of data distribution protocol. Model of incoming data rates is given in Section III. In Section IV we present the queuing model of distribution network, and in Section V we present the model of the total data distribution time in the network with derivation of forking probability. Performance evaluation is presented in Section VI, while Section VII concludes the paper.

II. Block and transaction distribution and propagation protocol

We consider Bitcoin network with N nodes. Node connectivity is presented as a $N \times N$ symmetric random matrix in which each node has a random number of TCP connections



(a) Block/transaction forwarding protocol between neighbors.



(b) Block forwarding protocol in the network.

Fig. 1. Block and transaction distribution in Bitcoin.

towards its neighbors. This probability distribution can be expressed with the Probability Generating Function (PGF) $Cn(z) = \sum_{i=c_{min}}^{c_{max}} pc_i z^i$ with mean value $\overline{Cn} = Cn'(1)$. Mass probabilities pc_i reflect connectivity patterns and dynamics of the network. Diameter of the network $D_{N,Cn(z)}$ depends on the network size and connectivity among the nodes, and the probability that nodes A and B are connected is

$$pb = \sum_{i=c_{min}}^{c_{max}} pc_i \frac{i}{N} = \frac{\overline{Cn}}{N} \tag{1}$$

Block and transaction delivery protocol resembles to some extent the randomized gossip protocol from [12]. It follows a two way handshake over each TCP connection as shown in Fig. 1(a). Node which is the source of block or transaction will send *inv* (inventory) message to its direct neighbors which respond with *getdata* requests if they don't have the block/transaction being announced; nodes which already have the block/transaction will not send get request. In this manner, block and transaction propagate through the network in generations as shown in Fig. 1(b).

We model block and transaction (hereafter referred collectively as data) forwarding through the network using branching processes [8]. Propagation protocol has the same behavior for the block that was mined by the particular node or transaction that has arrived to that node as the ingress of the overlay network. In the first step, the source node will act as the root of the spanning tree of the graph and transmit the data over all its TCP connections. In order to trace phases of the distribution protocol we will assign index $i = 0\,..\,D_{N,Cn(z)} - 1$ to each phase and therefore we also need the initial connectivity PGF as $Cn_1(z) = Cn(z)$.

First hop neighbors will receive the data and they comprise the first generation of nodes that has received the data; their number can be described with the PGF of

$$H_1 = Cn_1(z) = \sum_{i=c_{min,1}}^{c_{max,1}} pc_i z^i \tag{2}$$

When nodes from first generation start distributing data to their neighbors we need to address and model following problems (these problems exist in all later phases):

1) In current generation i, some nodes are interconnected so they will not transmit data to each other since they already got it from nodes in generation $i-1$.
2) Some nodes in target generation i will have connections towards the same node in generation $i+1$. This decreases the number of TCP connections through which a node in generation $i+1$ will further distribute the data to nodes in generation $i+2$. This will decrease node population in generation $i+1$ as well.

Therefore the total number of TCP connections for a node in generation i consists of the connections coming from generation $i-1$, described with PGF $In_i(z)$; connections towards other nodes in the same generation i, described with PGF $Loc_i(z)$; and connections available to transmit data to nodes in generation $i+1$, described with PGF $O_i(z)$. For each node in the first generation, receiving data from the root these PGFs are

$$In_1(z) = z \tag{3}$$

$$Loc_1(z) = \sum_{i=c_{min,1}}^{c_{max,1}} pc_i \sum_{k=0}^{i-1} \binom{i-1}{k} pb^k z^k (1-pb)^{i-1-k}$$

$$O_1(z) = \sum_{i=c_{min,1}}^{c_{max,1}} pc_i \sum_{k=0}^{i-1} \binom{i-1}{k} pb^k (1-pb)^{i-1-k} z^{i-1-k}$$

PGF for the number of connections coming from a first generation node and available to send data towards second generation nodes is

$$\Theta_1(z) = \sum_{i=c_{min,1}}^{c_{max,1}} pc_i \sum_{k=0}^{i-1} \binom{i-1}{k} pb^k (1-pb)^{(i-1-k)} \cdot \frac{\sum_{l=1}^{i-1-k} \binom{i-1-k}{l} pb^{(i-1-k-l)} (1-pb)^l \cdot z^l}{\sum_{l=1}^{i-1-k} \binom{i-1-k}{l} pb^{(i-1-k-l)} (1-pb)^l} \tag{4}$$

Some of the connections going from nodes in first to nodes in second generation may end at the same node which will

reduce the size of the second generation; the corresponding PGF which models this overlap is

$$Ot_1(z) = \frac{\sum_{k=1}^{\overline{H_1}} \binom{\overline{H_1}}{k} pb^k (1-pb)^{\overline{H_1}-k} \cdot z^{(1/k)}}{\sum_{k=1}^{\overline{H_1}} \binom{\overline{H_1}}{k} pb^k (1-pb)^{\overline{H_1}-k}} \quad (5)$$

The PGF for the number of TCP connections leaving a node in the first generation and generating one member in the second generation is

$$Cn_2(z) = \Theta_1(Ot_1(z))) \quad (6)$$

which leads to the PGF for the population of the second generation being expressed as

$$H_2(z) = Cn_2(H_1(z)) \quad (7)$$

After this $Cn_2(z)$ and $H_2(z)$ are used to compute node connectivity and population for the third phase and so on. This process is continued for a total number of $D_{N,Cn(z)} - 1$ phases. In each phase mean node population in generation i, $i = 0 \,..\, D_{N,Cn(z)} - 1$ is calculated as $\overline{H_i} = H_i'(1)$ and probability that a given node is reached in i-th generation is

$$Pt_i = \frac{\overline{H_i}}{N} \quad (8)$$

Mean number of nodes reached in the last generation and probability that data will not be forwarded can be then computed as

$$\overline{H_{D_{N,Cn(z)}}} = N - \sum_{i=0}^{D_{N,Cn(z)}-1} \overline{H_i}$$
$$Pnt = 1 - \sum_{i=0}^{D_{N,Cn(z)}-1} Pt_i \quad (9)$$

III. Data rates in the Bitcoin network

Model from section II describes the network distribution of one data item. However there are many mining nodes (if not all) in the network which can inject mined blocks for distribution and verification, and virtually all nodes can inject new transactions. Therefore traffic offered to any network node consists of fresh traffic coming for further distribution and traffic which is in the process of distribution over the network. For the new block traffic we will assume that each node can be a miner, and we will use the standard assumption that the time to complete PoW for a new block is exponentially distributed, i.e., that blocks arrive to network according to homogeneous Poisson process [4], [3], [7], [19].

When a node begins to distribute a newly mined block, other nodes which are in the process of mining the same block will cancel mining process upon hearing the new block. If distribution time of the block is close to zero and nodes have same computational power then Poisson arrival rate of new mined blocks in the network with N miners would be $\lambda_b = \frac{1}{600N}$. However, due to finite block distribution time, it is possible that other block is mined at more than one node in the network and distributed as well, even though previous block has not completed the distribution. This will result in the forking event where different blocks are mounted as blockchain heads in different areas of the network. As a consequence, the mined block Poisson arrival rate per node is

$$\lambda_b = \frac{1}{600N}(1 + P_{fork}) \quad (10)$$

assuming same computational power of nodes. The factor $P_{fork}\lambda_b$ originates from the race situation when two nodes complete mining the block in the time window shorter than transaction distribution time; it will be computed in Section V-A.

We also assume that new transactions arrive to each node with rate of λ_t. In order to model the traffic through the network we need to use PGFs for the node connectivity in each phase. At each node of the network, PGFs for the fresh traffic for blocks and transactions, respectively, entering the network are $\lambda_{b,0}(z) = z^{\lambda_b}$ and $\lambda_{t,0}(z) = z^{\lambda_t}$.

For the initial phase $i = 0$, input and output rates for block traffic are the same, i.e.,

$$\lambda_{b,0}^{in}(z) = \lambda_{b,0}^{out}(z) = z^{\lambda_b}, \qquad \lambda_{t,0}^{in}(z) = \lambda_{t,0}^{out}(z) = z^{\lambda_t}$$

For each further phase i, $i = 1 \,..\, D_{N,Cn(z)} - 1$, since TCP connections are bidirectional, PGFs for the input and output, block and transaction traffic are

$$\lambda_{b,i}^{in}(z) = Cn_i(\lambda_{b,i-1}^{out}(z)) \quad (11)$$
$$\lambda_{t,i}^{in}(z) = Cn_i(\lambda_{t,i-1}^{out}(z))$$
$$\lambda_{b,i}^{out}(z) = \lambda_{b,i}^{in}(z)\left(1 - Pnt - \sum_{j=0}^{i-1} Pt_j\right) + Pnt + \sum_{j=0}^{i-1} Pt_j$$
$$\lambda_{t,i}^{out}(z) = \lambda_{t,i}^{in}(z)\left(1 - Pnt - \sum_{j=0}^{i-1} Pt_j\right) + Pnt + \sum_{j=0}^{i-1} Pt_j$$

Since each node distributes data in all phases, total traffic offered to each node has following PGFs for blocks

$$\lambda_{b,tot}^{out}(z) = \prod_{j=0}^{D_{N,Cn(z)}-1} \lambda_{b,j}^{out}(z) \quad (12)$$
$$\lambda_{t,tot}^{out}(z) = \prod_{j=0}^{D_{N,Cn(z)}-1} \lambda_{t,j}^{out}(z)$$

Therefore, offered traffic and output to each node are random variables due to the combination of distribution of blocks and transactions in different phases and due to the random connectivity of each node towards the rest of the network. and we can get first two central moments as

$$\overline{\lambda_{b,tot}^{out}} = \frac{d}{dz}\lambda_{b,tot}^{out}(z)\Big|_{z=1} \quad (13)$$
$$var(\lambda_{b,tot}^{out}) = \frac{d^2}{dz^2}\lambda_{b,tot}^{out}(z)\Big|_{z=1} + \overline{\lambda_{b_{tot}}^{out}} - \overline{\lambda_{b,tot}^{out}}^2$$

Values for the transaction traffic have analogous form.

Due to the complexity of expressions (12), we model them as Gamma probability distributions with densities:

$$f_{\lambda,b}(y) = \frac{1}{\Gamma(c_{\lambda,b})} b_{\lambda,b}^{c_{\lambda,b}} y^{c_{\lambda,b}-1} e^{-y/b_{\lambda,b}} \tag{14}$$

where values $b_{\lambda,b}$ and $c_{\lambda,b}$ are defined as

$$\begin{aligned} b_{\lambda,b} &= var(\lambda_{b,tot}^{out})/\overline{\lambda_{b,tot}^{out}} \\ c_{\lambda,b} &= \overline{\lambda_{b,tot}^{out}}/b_{\lambda,b} \end{aligned} \tag{15}$$

and values for transaction arrival rates $b_{\lambda,t}$ and $c_{\lambda,t}$ have analogous form.

IV. Queuing model of the distribution network

So far we have seen that each node receives blocks and transactions in different distribution phases over its TCP connections. It will request and receive only the data which it does not currently have. Moreover node needs to prioritize block traffic over transaction traffic in order to minimize the probability of forking. Note that blocks and transactions join different pools for verification. This framework can be modeled using Jackson network approach [13] applied to non-preemptive priority queues as discussed in [9], [10]. Each node receives mined blocks and new transactions, and processes them using two queues organized in priority order of blocks over transactions. Each node also can eliminate a block or a transaction if there is no need for further forwarding. Since data which will not be forwarded does not join forwarding queues, the input rate for one node consists of output rates (of all protocol phases) of all nodes connected to it.

Time to process and forward single block consists of block verification time, time to exchange *inv and* getdata messages which is comprised of two mean round trip times (RTTs), and time to transmit the block. Block verification time $\overline{BV}$ is of the order of ms, while mean RTT takes tens to perhaps two hundred ms. Transmission time of the block over TCP connection is roughly given as $\overline{B}/\overline{R}$ where $\overline{B}$ denotes mean block size in bits and $\overline{R}$ denotes mean throughput of TCP connection (for simplicity, we ignore segment headers). Transmission time has order of magnitude expressed in seconds.

Given different sizes and distributions of verification time, RTT and block transmission time we have assumed that block service time is exponentially distributed with service rate derived from the sum of component means $\mu_b = \frac{1}{\overline{B}/\overline{R}+2\overline{RTT}+\overline{BV}}$, pdf $b_b(x) = \mu_b e^{-mu_b x}$ and Laplace Stieltjes Transform (LST) $B_b^*(s) = \mu_b/(s+\mu_b)$. Note that in our further derivation we will use M/G/1 framework so any other distribution of service time can be used. As the transaction verification and transmission times are much smaller, the main component of the transaction service time are two RTTs. We assume that transaction service time is exponentially distributed with rate $\mu_t = \frac{1}{2\overline{RTT}}$, pdf $b_t(x) = \mu_t e^{-mu_t x}$ and LST $B_t^*(s) = \mu_t/(s+\mu_t)$.

For non-preemptive priority M/G/1 analysis of block and transaction queues [20] we need probability distributions of the number of block arrivals during block service time (with PGF $A_b(z)$) and number of block arrivals during transaction service time, with PGF $A_{bt}(z)$. Specific challenge in this model is that arrival rates of non-homogeneous Poisson processes for blocks and transactions $\lambda_{b,tot}^{out}$ and $\lambda_{b,tot}^{out}$ are random and framework from [20] had to be modified. Using the individual arrival probabilities for block and transaction service times

$$a_{b,k} = \int_{y=0}^{\infty}\int_{x=0}^{\infty} \frac{(xy)^k}{k!} e^{-xy} f_b(y) b_b(x) dx dy \tag{16}$$

$$a_{bt,k} = \int_{y=0}^{\infty}\int_{x=0}^{\infty} \frac{(xy)^k}{k!} e^{-xy} f_b(y) b_t(x) dx dy \tag{17}$$

we can calculate the corresponding PGFs as

$$\begin{aligned} A_b(z) &= \sum_{k=0}^{\infty} a_{b,k} z^k = \int_{y=0}^{\infty} B_b^*(y(1-z)) f_b(y) dy \\ A_{bt}(z) &= \sum_{k=0}^{\infty} a_{bt,k} z^k = \int_{y=0}^{\infty} B_t^*(y(1-z)) f_b(y) dy \end{aligned} \tag{18}$$

Computational complexity of computing the Taylor series expansion is high but 6-20 series members provide sufficient accuracy of the PGF (i.e., we have been computing mass probabilities until they are larger than 10^{-5}). For block traffic (which has higher priority), the PGF for the number of blocks left in the queue after the block departure and LST for the block response time are, respectively,

$$\begin{aligned} PP_b(z) &= A_b(z) \frac{\overline{\lambda_{t,tot}^{out}} A_{bt}(z) - \overline{\lambda_{tot}^{out}} + \rho\overline{\lambda_{b,tot}^{out}} + z(1-\rho_{tot})\overline{\lambda_{b,tot}^{out}}}{\overline{\lambda_{b,tot}^{out}}(z - A_b(z))} \\ T_b^*(s) &= B_b^*(s) \frac{(1-\rho_{tot})s + \overline{\lambda_{t,tot}^{out}}(1 - B_p^*(s))}{s - \overline{\lambda_{b,tot}^{out}} + \overline{\lambda_{b,tot}^{out}} B_b^*(s)} \end{aligned} \tag{19}$$

where $\overline{\lambda_{tot}^{out}} = \overline{\lambda_{b,tot}^{out}} + \overline{\lambda_{t,tot}^{out}}$, $\rho_{b,tot} = \overline{\lambda_{b,tot}^{out}}/\mu_b$, $\rho_{t,tot} = \overline{\lambda_{t,tot}^{out}}/\mu_t$ and $\rho_{tot} = \rho_{b,tot} + \rho_{t,tot}$. Moments of node response time for blocks can be found as $\overline{T_b}^{(k)} = (-1)^k \frac{d^n}{ds^n} T_b^*(s)\big|_{s=0}$ where index k denotes the order of the moment.

V. Block distribution time

Using the probabilities of phases in data distribution protocol calculated in Section II, we can calculate the LST of duration of total block/transaction distribution as

$$\Xi_b^*(s) = \frac{\sum_{i=0}^{D_{N,Cn(z)}-1} Pt_i (T_b^*(s))^{(i+1)}}{1 - Pnt} \tag{20}$$

and from those LSTs all necessary moments of distribution time can be found. Due to complexity of expressions (20), it is difficult to get probability density functions using inverse transformations. Instead we have estimated data distribution time using Gamma probability distribution with probability density as function of

$$f_{n,b}(x) = \frac{1}{\Gamma(c_{n,b})} b_{n,b}^{c_{n,b}} x^{(c_{n,b}-1)} e^{-x/b_{n,b}} \tag{21}$$

where $b_{n,b} = var(\Xi_b)/\overline{\Xi_b}$ and $c_{n,b} = \overline{\Xi_b}/b_{n,b}$.

A. Forking Probability

Probability of forking for two blocks equals the probability that block interarrival time is smaller than block distribution time. To model this we need to isolate the node which has mined a block, say, A and the rest of the network where each node can be a miner of another block B with equal probability. If pdf of block interarrival time for the rest of the network containing N_1 nodes is given with $g_b(y)$ then forking probability becomes

$$P_{fork} = \int_{x=0}^{\infty} \left(\int_{y=0}^{x} g_b(y)dy \right) f_{n,b}(x)dx \qquad (22)$$

If the block interarrival time from $N-1$ nodes in the network is exponentially distributed with parameter $(N-1)\lambda_b$, expression (22) becomes

$$P_{fork} = \int_{x=0}^{\infty} (1 - e^{-(N-1)\lambda_b x)}) \cdot \frac{1}{\Gamma(c_{n,b})} b_{n,b}^{c_{n,b}} x^{c_{n,b}-1} e^{-x/b_{n,b}} dx \qquad (23)$$

Unfortunately, P_{fork} from the last expression participates in block arrival rate defined in (10). This results in a system of equations which need to be solved iteratively starting from low tentative values, e.g., $P_{fork}^0 = 0.001$, until the difference between values computed in successive iterations becomes smaller than the predefined threshold.

Using the probabilities of reaching the i-th generation and populations of nodes reached in particular distribution phases it is possible to calculate sizes of network partitions with different heads of the chain.

VI. Performance Evaluation

Our evaluation was conducted for network sizes in the range $N = 2000 \,..\, 4000$ in steps of 250. For each network size, a random symmetric $N \cdot N$ matrix was generated. Number of connections per row followed random probability distribution $Cn(z) = \sum_{i=c_{min}}^{c_{max}} pc_i z^i$ with mean value of $\overline{Cn} = Cn'(1) = 8$ while minimum and maximum numbers were $c_{min} = 5$ and $c_{max} = 13$ respectively. For eight values of network size, network had the diameter of 6. Larger values of diameter (e.g., 7 and 8) were possible due to randomness of connectivity, but for reasons of calibration we have chosen outcomes with diameter equal to 6 since a larger diameter for some outlier cases would have produced larger delays and affect parameters like forking probability.

A. Performance of block distribution algorithm

Probability distributions of node connectivity are shown in Fig. 2. They are close to each other since all network graphs were generated using the same algorithm. Fig. 3(a) shows mean number of hops needed to distribute block/transaction when network size increases from 2000 to 4000 under same diameter and similar connectivity. We observe that mean number of hops drops due to higher number of links available to carry data. However, when the number of nodes increases over 4000 with same connectivity distribution, diameter will inevitably increase and mean number of hops will experience an increase as well. When network size increases (under same diameter and connectivity distribution), more nodes are reached in the last phase. This is further confirmed in Figs. 3(b) and 3(c) which show mean number of nodes reached in each phase and probability that node is reached in phase $i = 0 \,..\, 6$ respectively. Each figure has 8 curves which correspond to one value of network size in the range $N = 2000 \,..\, 4000$. In Fig. 3(b) lowest curve shows case when $N = 2000$ and highest curve shows the case when $N = 4000$ (curves between them are monotonically ordered). All curves show exponential increase of mean populations in each generation until fourth phase. After that gradient towards fifth generation becomes negative for smaller network sizes $N = 2000 \,..\, 2500$ or slows down for $N = 2750 \,..\, 4000$. Consequently, probabilities of belonging to i-th generation (Fig. 3(c)) are obtained by dividing mean generation size with network size. In this case probability of belonging to the fourth generation is highest when $N = 2000$ and lowest when $N = 4000$. Probability of being in fifth and sixth generation declines for $N = 2000$, while for $N = 2250 \,..\, 2750$ the decline starts after fifth generation. For network sizes $N = 3000 \,..\, 4000$ there is no decline but the gradient of increase slows down.

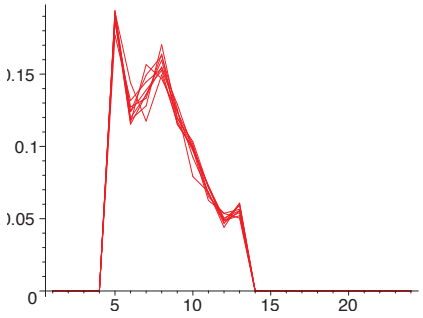


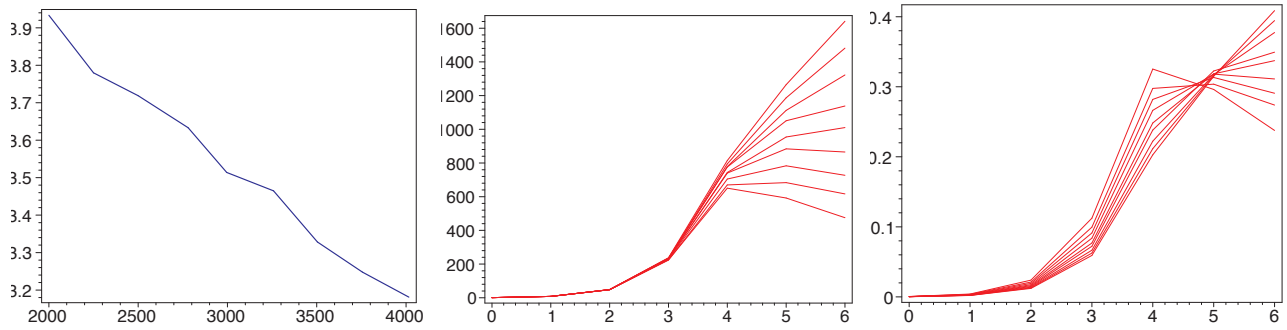
Fig. 2. Distribution of number of connections per node

B. Queuing performance

In our experiment we assumed that mean RTT is 100ms, mean connection throughput is $\overline{R} = 2$Mbps, and mean block size is $\overline{B} = 400$KB, which result in mean block transmission time of 1.6s. This gives $\mu_b = 0.55$ and $\mu_t = \frac{1}{0.2}$.

Results from queuing analysis for block traffic are shown in Figs. 4 and 5. Fig. 4(a) shows mean value while Fig. 4(b) shows coefficients of variation (circles), skewness (diamonds) and kurtosis (boxes) for node response time for blocks. According to the values of higher moments node response time is almost exponential (exponential distribution has coefficients of variation, skewness and kurtosis equal to 1, 2, and 9 respectively). This is reasonable since block traffic is priority traffic, its rate is low and service time is exponentially distributed.

Mean network distribution time for blocks is shown in Fig. 5(a), while Fig. 5(b) shows its coefficients of variation (circles), skewness (diamonds) and kurtosis (boxes). Mean delivery time shows slight decline from 10.6s to 9.8s when



(a) Mean number of hops

(b) mean number of nodes reached in i-th generation

(c) Probabilities that node is reached in i-th generation including diameter

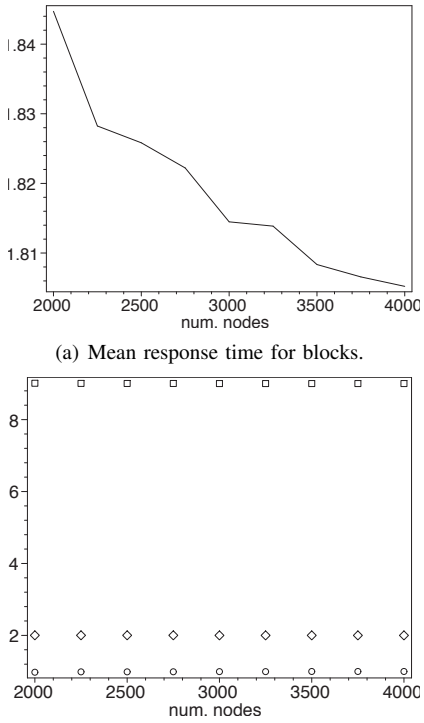
Fig. 3. Topological and data distribution properties of the network.

network size grows from 2000 to 4000 as a consequence of richer connectivity. At the same time higher moments in Fig. 5(b) show behavior of generalized Erlang-k distribution with the number of stages approximately equal to the mean number of hops during distribution. This is expected since expression (20) indicates that network distribution time is a linear combination of partial path times and node response time has distribution close to exponential. This result is different from the observation that block propagation time follows exponential distribution [3], [19].

Forking probability in Fig. 5(c) exhibits decrease when network size is increasing and has the shape close but not identical to the mean block distribution time. This is due to the decrease of the block distribution time and also due to the slight increase of the block B arrival rate with the increase of network size, as shown in (22).

VII. Conclusion

In this work we have developed the analytical model of data delivery protocol over P2P Bitcoin network. We have also developed priority based queuing model of Bitcoin nodes and Jackson network model of the whole network. These models give probability distributions of node populations in data distribution phases, response time of nodes, and network distribution times for blocks and transaction. Our results show strong qualitative and quantitative dependency of network performance on distribution of node connectivity and network size. Intensity of transaction traffic cannot affect performance of block traffic much since it has low priority. We demonstrate use of the integrated model on the calculation of forking probability. This model can be also used for modeling of other attacks in the network which will be the subject of our future work.



(a) Mean response time for blocks.

(b) Coefficients of variation, skewness and kurtosis for block response time.

Fig. 4. Node response time for block and network delivery times for blocks.

References

[1] M. Campbell-Verduyn. *Bitcoin and Beyond: Cryptocurrencies, Blockchains, and Global Governance*. Routledge, 2018.

[2] S. Davidson, P. De Filippi, and J. Potts. Economics of blockchain. In *Proc. of Public Choice Conference*, Ft. Lauderdale, FL, May 2016.

[3] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *Proc. 13th IEEE Int.l Conf. Peer-to-Peer Computing (P2P'13)*, volume 26, 2013.

[4] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. *arXiv preprint arXiv:1311.0243*, 2013.

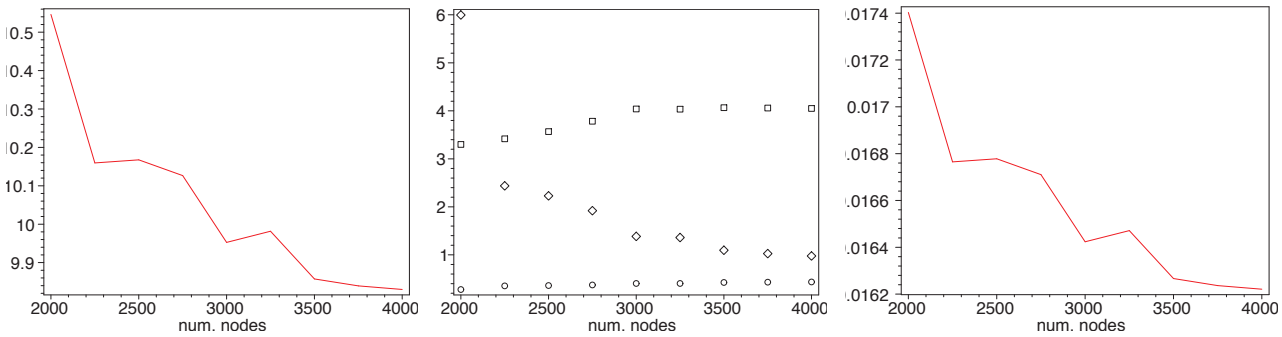
[5] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proc. ACM SIGSAC*, pages 3–16, 2016.

[6] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Čapkun. Tampering with the delivery of blocks and transactions in bitcoin. In *Proc. ACM SIGSAC*, pages 692–705, 2015.

[7] J. Gobel, H. Keeler, A. Krezinski, and P. Taylor. Bitcoin blockchain dynamics. *Performance Evaluation*, 104:23–41, Oct. 2016.

[8] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, Oxford, UK, 2nd edition, 1992.

[9] M. Harchol-Balter and T. Osogami. Multi-server queueing systems with multiple priority classes,. *Performance Evaluation*, pages 331–360, 2005.



(a) Network mean delivery time for blocks.

(b) Coefficients of variation, skewness and kurtosis for block delivery time in the entire network.

(c) Forking probability.

Fig. 5. Block delivery time and forking probability.

[10] E. Kao and K. Narayanan. Computing steady-state probabilities of a non-preemptive priority multiserver queue. *Journal on Computing*, 2(3):211–218, 1990.

[11] G. O. Karame, E. Androulaki, and S. Capkun. Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 906–917. ACM, 2012.

[12] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proc. Ann. Symp. Foundations of Computer Science*, pages 565–574, Redondo Beach, CA, 2000.

[13] L. J. Kleinrock. *Queuing Systems*, volume I: Theory. John Wiley and Sons, New York, 1972.

[14] P. Müller, S. Bergsträßer, A. Rizk, and R. Steinmetz. The bitcoin universe: An architectural overview of the bitcoin blockchain. In *11. DFN-Forum Kommunikationstechnologien*. Gesellschaft für Informatik eV, 2018.

[15] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[16] T. Neudecker, P. Andelfinger, and H. Hartenstein. A simulation model for analysis of attacks on the bitcoin peer-to-peer network. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1327–1332, 2015.

[17] T. Neudecker and H. Hartenstein. Network layer aspects of permissionless blockchains. *IEEE Communications Surveys Tutorials*, 2018. 10.1109/COMST.2018.2852480.

[18] S. Neumayer, M. Varia, and I. Eyal. An analysis of acceptance policies for blockchain transactions, 2018. https://hdl.handle.net/2144/31455.

[19] N. Papadis, S. Borst, A. Walid, M. Grissa, and L. Tassiulas. Stochastic models and wide-area network measurements for blockchain design and analysis. In *INFOCOM 2018*, pages 2546–2554, 2018.

[20] H. Takagi. *Queueing Analysis*, volume 1: Vacation and Priority Systems. North-Holland, Amsterdam, The Netherlands, 1991.