Attack on Uniform Random Tip Selection Algorithm in Proposed Distributed Ledger based Vehicle Network

[redacted]

Abstract: *Successful future communications platforms for land air and drone vehicles require trust, timeliness and security. One such platform, which involves the use of a Distributed Ledger, is called IOTA. IOTA is focused on connecting Internet of Things devices. IOTA incorporates a Distributed Ledger Technology (DLT) which stores a record of each exchange between participating nodes. This research explores the robustness of IOTA while under cyber-attack, in order to measure its security and integrity. Attacks on DLT commonly involve overwhelming the network with malicious nodes that seek to verify bogus transactions or data 'messages'. In this way a malicious node might alter the level of confidence of new messages arriving into the ledger. This attack uses fake messages called 'Double Spend' in an attempt to fork the blockchain and create an alternate ledger which replaces genuine messages and transactions with tampered copies. "Double Spend" attacks were first developed to attack Distributed Ledger based crypto-currencies. This research also confirms that Double Spend attacks are more successful when the number of messages and the number of corresponding nodes is low. The work contrasts Uniform Random Tip Selection and Restricted Uniform Random Tip Selection Algorithms. Finally mitigations against Double Spend attacks are detailed.*

Keywords—Distributed Ledger, blockchain technology, Uniform Random Tip Selection (URTS), Internet of Things (IOT), IOTA Tangle, Double Spend, Watts Strogatz Node Clustering, Poisson Distribution, Directed Acyclic Graph, Markov Chain Monte Carlo.

Introduction

According to GSMA Intelligence, the global market for Device to Device (Internet of Things) communications is forecast to be €30 billion by 2035 [1]. Distributed Ledger Technology (DLT) has the potential to offer a highly robust communications platform for vehicle communications on land and in the air [2]. A car, plane or drone's direction and velocity are communicated as data messages, which must be verified by the IOTA Distributed Ledger at speeds close to real time [3]. When these data messages first appear in the DLT based network they are unapproved "tips". If the tip is sent from a node that is already approved by other nodes in the network, then it is more likely that the tip will be "confirmed" as genuine. The IOTA Tangle is a Distributed Ledger Technology (DLT) developed for Internet of Things devices to ensure rapid tip selection and message confirmation through a Uniform Random Tip Selection algorithm (URTS). [4] Unlike crypto-currencies, which use a linear blockchain and have slower confirmation times, IOTA Tangle uses a three dimensional cone shaped Directed Acyclic Graph (DAG). The cone shape results from a message verification process optimised to prevent new valid messages being ignored and isolated to later become "orphans".

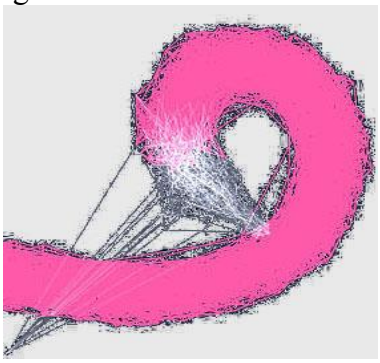


Image 1:Cone shaped DAG in IOTA Tangle (Hornet)

Moreover, this three dimensional shape permits the network to process increased numbers of messages in a faster period compared to a linear blockchain such as Bitcoin, which can take up to ten minutes to confirm a single message or transaction.

Directed Acyclic Graph

The Directed Acyclic Graph is a three dimensional non-repeating graph where nodes represent vertices and links between nodes represent edges. The DAG allows asynchronous parallel attachment of transactions that are faster to confirm than linear blockchain [5]. In computational terms, a DAG is a data structure represented by a matrix which can be coded as a nested list or array. These data structures have no in built security. Unlike linear blockchain, as used in crypto-currencies, IOTA Tangle transactions do not involve financial balance sheets. This minimises the delay in confirmation times for DAG messages. Moreover, DAG performs faster because it does not require an elaborate "proof of work", a time consuming calculation designed to limit participation in crypto-currencies. In a vehicle network, a DAG node may represent a land, sea or drone vehicle. In this configuration, the IOTA tangle provides a network for communicating messages between nodes and offers security against tampering [4]. In a DAG, any new message is allowed to attach itself to the tangle provided it has authenticated or "approved" at least two earlier messages. This agreement or "consensus", builds trust among the participating nodes and replaces the elaborate 'Proof of Work' operation employed in rival blockchain Distributed Ledgers. Nodes verify genuine messages through an incrementing trust system that uses cumulative weights assigned to each node. The first node in a DAG is called the "genesis" and will contain the most cumulative weight. As such, its vote to accept or reject a new message will carry more importance [5]. Unverified new messages that freshly arrive into the DAG are unknown to network nodes. They have no trust associated and no cumulative weight [5]. Newly selected messages are grouped into strong candidates or weak candidates. At this stage the messages become "tips". (Strong tips have stronger links to known nodes). These groups of tips are known as Tip Pools. [6] Then a Tip Selection Algorithm selects them and the tips enter the approval or denial process [7]. Other rival Distributed Ledgers include Obyte (Byteball) and Hashgraph [8] which uses timestamps to determine the correct sequence of transactions [5]. However Hashgraph is a privately owned technology and is legally protected from unsanctioned users.

I. Tip Selection

A) *Uniform Random Tip Selection.*

A tip is selected from the group of available messages with a uniform probability, where all outcomes are equally likely. The URTS algorithm has less time to first approval than an Unbiased Random Walk, which depends on the DAG topology and is more chaotic because it has an equal probability of direction for the next hop. [9]. An example of an Unbiased Random Walk with a bias of zero is the Markov Chain Monte Carlo selection algorithm. URTS is more simple than Unbiased Random Walk and has the advantage of requiring less computational cycles. It does not leave any unapproved isolated transactions (orphans). URTS performance is more efficient when the tangle grows to a large size [9]. A weighted URTS enables faster tip selection to increase message throughput [10]. One common problem is a "lazy tip", which is a tip that selects older messages that may already have been validated. This can mean that new messages will be validated less frequently, which can lead to more isolated branches of the network. Non lazy tips will seek to approve messages closer in time to themselves. To discourage "lazy tips", developers introduced a bias called Cumulative Weight into the node walk [11]. This weight is calculated by adding the number of existing messages that directly and indirectly approve a message. By this means, tips are more likely to attach to trusted messages that have trusted links. URTS will preferentially select tips which are 'non-lazy' to maximize confirmation rate and encourage attachments to newer messages. This maintains the health of the tangle and causes it to reform its three dimensional cone shape. [11]

This rate of approving messages for URTS can be calculated using the following formula. Let L(t) determine the average time until a tip is approved. In URTS each message approves two tips, so the rate of approval is 2λ. (This is the rate at which all tips are approved.)

L(t)= number of tips through time.
$L(t)^{-1}$ *= Probability of tip approval.*
λ *= Poisson rate for new message arrival.*
e^{λ} *=time between transactions.*
L(t)= number of tips through time.
h = unit of time between a message selection and its confirmation.

$$R=\frac{2\lambda}{L(t)} = \textit{rate of approvers.}$$

Insert: Uniform Random Tip Selection Rate of Approvers [9]

B) Restricted Uniform Random Tip Selection.

Like URTS, RURTS is a Uniform Probability Distribution wherein all outcomes are equally likely. For this reason it is good at modelling randomness in a finite set of variables within a specific range. RURTS differs from URTS through the addition of a timeout function to eliminate messages that have aged beyond a certain threshold (5 seconds) [11]. Messages are accepted if they pass the timeout stamp check, otherwise they are discarded. RURTS also has an approval switch controlled by the issuer node [7]. This is a mechanism to measure trust in the parents of a message. The ancestry of a message is calculated from its three dimensional "past cone" see (Image 1). Thus a candidate message is eligible if it passes timestamp test, has a valid Ledger State and eligible parents. If a tip's parents cannot be verified then the message is labelled a 'weak tip'. As mentioned above, there are two TIP Selection Pools, weak and strong. Strong tips are selected first. Tips are removed from the Tip Selection pool when they are approved by other messages. When a new message is received, its parents are removed from all Tip Selection pools. (This avoids congestion in the pools). Tips can move to a strong pool if their parents become verified [7].

C) IOTA Tangle and White Flag Ordering

Recall that Iota Tangle is a type of Distributed Ledger which has a higher throughput than blockchains more suited to crypto-currencies. Moreover, security within IOTA Tangle has a defence against 'brute force attacks' powered by future advanced Quantum computers through the use of Winternitz Signatures [5]. (A' brute force' attack is an attempt to calculate and test every possible decryption possibility in an attempt to discover the right solution.) A Winternitz private key is generated by: 1) selecting 32 random 256 bit numbers. 2) Hashing each number 256 times (this is the public key). 3) Hash the entire message using SHA 256 to generate 32 8-bit values. 4) Hash the 8 bit values 256-N times where N is the number of the 8 bit value, this is the Winternitz signature. Winternitz signatures are highly efficient and take little time to complete. The current version of IOTA Tangle uses the Restricted Uniform Random Tip Selection Algorithm [7]. Conflicts of opinion regarding the veracity of messages are resolved using a decision mechanism called White Flag Ordering. [12] In this mechanism, any suspicious or malformed messages are ignored. White Flag Ordering can defend against spamming attacks on the Tangle because it ignores conflicting messages. White Flag ordering is also known as the Hetfield Solution [12].

II. Distributed Ledger Security

A) Double Spend Attacks

This type of attack was named by the crypto-currency community as an attack that would enable a malicious actor to spend non-existent funds or funds that had already been expended in a previous transaction [13], by replicating the trade as a fake message. New transactions cannot approve both the original transaction and the fake, so the tangle is forced to split. If the attack is successful, the original genuine transaction and its main tangle, is ignored by new transactions and isolated. Thereafter, new messages are drawn to verify messages in the forked "parasitic" tangle. The genuine transaction and the genuine tangle, is then isolated and ignored. The attack performs best when the transaction rate is low [14]. In this condition, honest messages can more easily be overwhelmed by malicious messages. A message voting algorithm, to measure the trust afforded to a new message, will minimise the chances of a Double Spend attack [13]. In this work, a delay of 10 seconds was introduced before the double spend transactions were executed. This action decreases the likelihood of the tangle being overwhelmed by fake messages. An effective mitigation for a Double Spend Attack is a timestamp as employed by the Hashgraph Distributed Ledger. [8]

B) Eclipse Attacks

This type of attack involves an attacker isolating a node from the rest of the network and providing it with false information, leading to a situation where the node validates fraudulent transactions [15]. Under normal conditions an isolated node will not reconnect in a network. However in an eclipse attack this is not the case because an attacker will redirect a node's incoming and outgoing communications to only interact with nodes already under malicious control.

C) *Sybil Attacks*

The Uniform Random Tip Selection Algorithm assumes that nodes in the network are honest and operate independently. However, if an attacker gains control of a significant portion of the network, the voting system can be persuaded to validate other malicious nodes. In this scenario, an attacker can create many fake nodes and because the URTS algorithm selects tips at random, at some stage a malicious node will be selected. This increases the chances of a malicious node being asked to validate other nodes, many of which will also be malicious [16]. In IOTA, Sybil protection can be provided by a reputation value assigned to nodes. This measure of reputation is called 'mana', which is an extension of the Distributed Ledger. The 'mana' of a node determines how many messages it can issue [17]. Should a network experience a "Double Spend" attack and encounter two conflicting messages, the verdict of nodes or messages with higher "mana" carries more importance to make a decision about which message is fake.

D) *Spamming*

An attacker floods the network with a large number of fake tips. This causes congestion in the network, which makes it difficult for legitimate transactions to be validated. This paralysis will eventually lead to a denial-of-service condition. [18] One solution to a spamming attack is to employ the "mana" reputation index value described above. [7]

E) *Malicious Nodes*

Malicious nodes change their opinion about the validity of messages. A voting system that relies on a majority consensus can be corrupted as the number of malicious nodes increases. In this way a growing proportion of malicious nodes can be used to designate other malicious nodes as honest, thus corrupting the tangle. [19] One solution to defend against malicious nodes is to implement a White Flag Ordering system which will disregard changes of opinion among participating nodes [12]. An attack using malicious nodes to overwhelm an IOTA tangle is described below. The attack uses a simulator called TangleSIM developed by the IOTA foundation [11].

F) *Confirmed Message*

In the context of the Tangle a confirmed message is one that has been referenced by a periodic milestone, which is a list of approved messages. When the milestone is issued, the messages detailed inside are treated as confirmed. Milestones are hashed with Blake2b-256 and then authenticated by the Edwards Curve Digital Signature Algorithm (Ed25519) [20].

III. TANGLESIM SIMULATOR ALGORITHMS

A) *Poisson*

Poisson is a discrete probability distribution, which determines the probability of an event repeated occurrence for a certain number of times within a given interval of time and space. [23] Poisson distribution is used to calculate the staggered arrival rate of new messages into the simulated network. [24] Poisson distribution is also used to predict call centre traffic and footfall into a public area. Events are calculated at a constant mean rate independent of the time since the previous event. These arrival events must be separated by an interval. Poisson distribution can be replaced by Negative Binomial Distribution depending on the type of event dispersion [13].

B) *Watts Strogatz*

Watts Strogatz is a random probability distribution model that produces graphical representations of small real-world processes. These processes exhibit properties including shorter path lengths and high clustering [21]. One example is the traversal of a honey bee. Consider a shrub with > 50 flower blossoms as nodes , the bee divides the nodes into clusters of seven to eight blossoms each. Within each cluster, it moves between adjacent nodes. Then it moves to another cluster (circle below) and repeats the motion. Watts Strogatz can be used to model different types of propagation including disease transmission and "viral" content on social networks [22]. Watts Strogatz was based on the Erdos Renyi model which has no clustering. Watts Strogatz is not a regular node traversal but neither is it random. It is described as "small world".

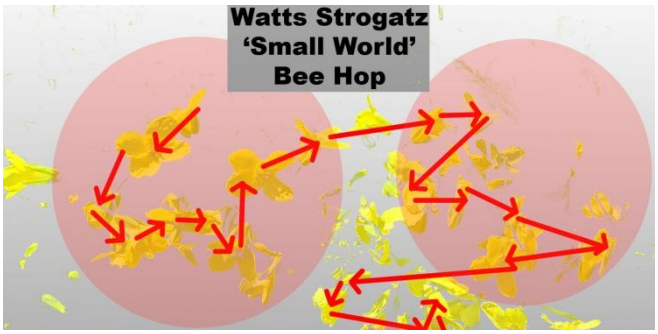


Image 2: Watts Strogatz "Small World" Bee Hop with clustering (circles)

C) *Zipfs Law*

When a list of measured values is sorted in decreasing order, then the value of any entry is inversely proportional to the value of its position (index) in the list. [25]

IV. DESIGN

This work is part of a larger research project to evaluate Distributed Ledger technology in order to develop a trustworthy timely and secure communications platform for future land/air/drone vehicles. The IOTA distributed ledger was selected because it is open source and is being developed for Internet of Things devices. Moreover it offers fast message confirmation rates via simultaneous message processing. An IOTA tangle network simulator was employed to introduce malicious nodes into an IOTA based tangle network. These malicious nodes then reversed the approval value that had been awarded to incoming messages. In this manner, messages that were initially deemed untrustworthy were newly designated as trusted. This network is therefore assumed to be Byzantine because it contains untrustworthy nodes [17]. In this way a fixed number of the malicious nodes attempt to disrupt the operation of the network. In the simulation the number of malicious nodes is held constant as the number of honest nodes is increased up to a limit. Thus the proportion of malicious nodes in the network becomes a decreasing ratio. The simulator has a finite number of nodes which may or may not be confirmed (processed) as the network approaches completion (convergence). Malicious nodes have a tip count allowance which can be manipulated. This can be weighted by a factor of ten to increase the influence of malicious nodes on the simulation. A control was introduced where an IOTA tangle network was created with no malicious nodes. The mana, or reputation score of the malicious nodes, was fixed at 10 per cent of the total mana allocated to all the messages in the network. This total is set at a value of 100,000,000. This ensures that the malicious nodes have a measurable impact on the network. Through this mechanism, malicious nodes will control the number of messages that are issued. In the following graphs this rate of delivery of tips issued from malicious nodes (tipsCount) was multiplied by 1 or 10, in order to augment the attack. All simulations were executed on Core i7 Debian Linux with 16GB RAM. The aim of the experiment is to study the disruption of the network by varying the proportions of malicious nodes compared to honest nodes inside the developing IOTA Tangle network. The objective is to determine the effect of increasing numbers of malicious nodes on the operation of an IOTA tangle distributed ledger based network. In particular, the research seeks to quantify the level of malicious nodes that can be tolerated before the system becomes unstable. This failure can manifest as a significant delay or permanent interruption of the tips processing procedure.

V. RESULTS

Confirmation time is the time taken for a message to be accepted by the tangle. Convergence time is the completion time for the simulation where all nodes and their messages are confirmed.

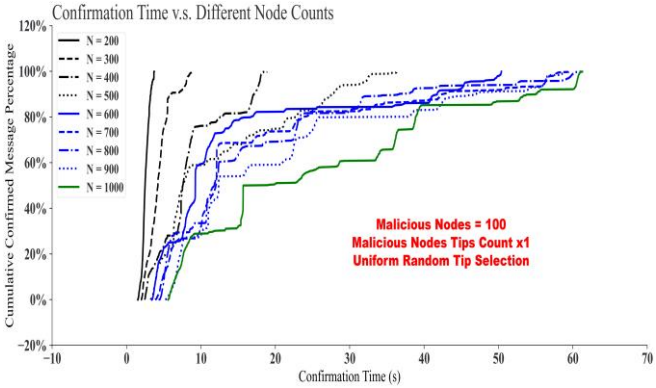
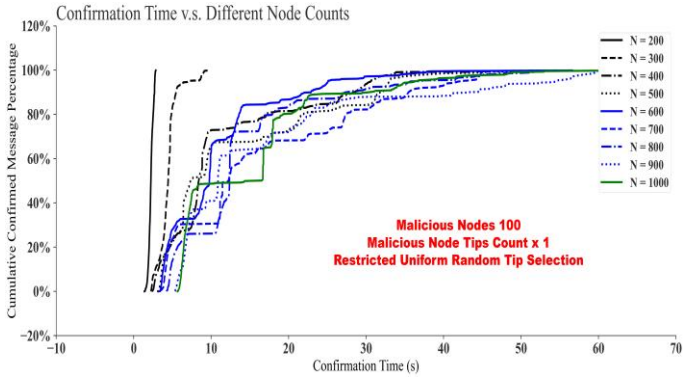


Fig 1. 100 malicious nodes, tipsCount_x1, URTS

Fig 2. 100 malicious nodes, tipsCount_x1, RURTS



Figs 1 and 2 represent a fixed amount of 100 malicious nodes with their message issuing speed (tipsCount) multiplied by 1. Note that with a total network node count of 1000 nodes, RURTS performs better than URTS (green line). This is because RURTS confirms 40 per cent of the total number of nodes before a delay is registered. URTS confirms 20 per cent of the total number of nodes before a delay is registered. This delay is visible as a shelf in graph. In this example malicious nodes represent only ten per cent of the total number of nodes. In contrast, with 200 total network nodes, malicious nodes make up 100, or

50 per cent of the total, but no disruption is visible (black line).

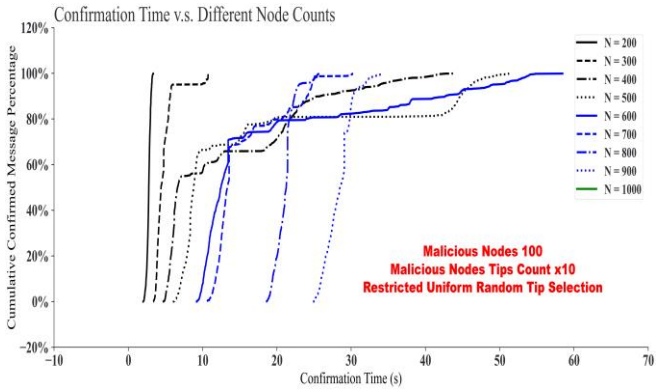


Fig 3. 100 malicious nodes, tipsCount_x10, RURTS. 1000 nodes network has failed (green line).

In Figs 3 and 4 the influence of the same number of malicious nodes is increased by multiplying their tipsCount by ten. With 1000 nodes (900 honest) the RURTS simulation failed to complete. However the corresponding URTS graph demonstrates that messages from 1000 nodes took approximately 60 seconds to complete.

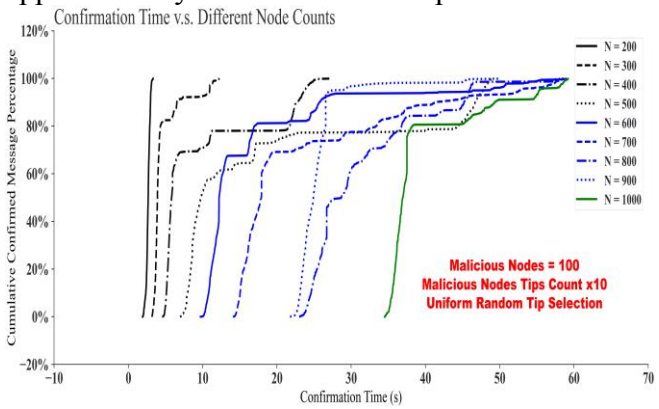


Fig 4. 100 malicious nodes, tipsCount_x10, URTS. 1000 nodes (green line), approx. 60 seconds to complete. Note the delay in times at approx. 60 per cent completion.

Next in Fig 5, 502 malicious nodes were introduced into the network. This comprises a slight majority of malicious nodes when the total is 1000 nodes. In this case disruption is minimal (blue line below). However, 502 forms a majority of malicious nodes when the total number of nodes is 600 (black line).

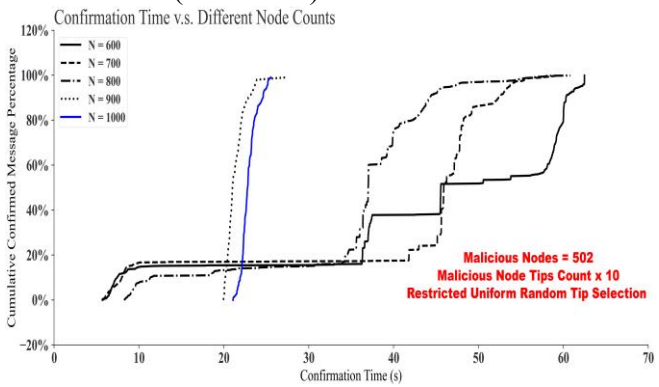


Fig 5. 502 malicious nodes, tipsCount_x10, RURTS.

At 600 nodes, the network is visibly degraded. At 800 nodes, confirmation times are also severely degraded because only 298 of the nodes are honest.

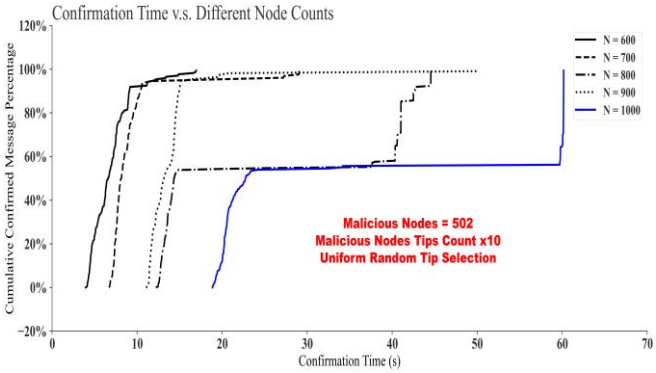


Fig 6. 502 malicious nodes, tipsCount_x10, URTS.

Similar degradation of the network is visible with URTS. In Fig 6 above, at 600 nodes (black line), the network performs better despite only 98 honest nodes. In contrast, note the blue line (1000 nodes) at which point confirmation times are delayed by approximately 40 seconds when 498 (almost half) of the nodes are honest (Fig 6). Next convergence times were plotted against increasing node counts, with a fixed number of malicious nodes.

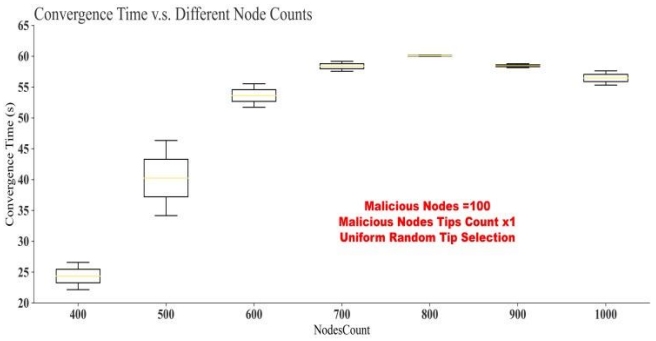


Fig 7. 100 malicious nodes, tipsCount_x1, URTS

Fig 7 shows that a URTS based network with 900 honest nodes and 100 malicious nodes converges in less time than a similar network with 700 honest nodes and 100 malicious nodes, the graph appears parabolic. According to Kusmierz et al, networks forming with both Biased Random Walk and Unbiased Random Walk will grow quadratically with the number of transactions (messages). That is to say the networks will grow with a constant rate of change [9].

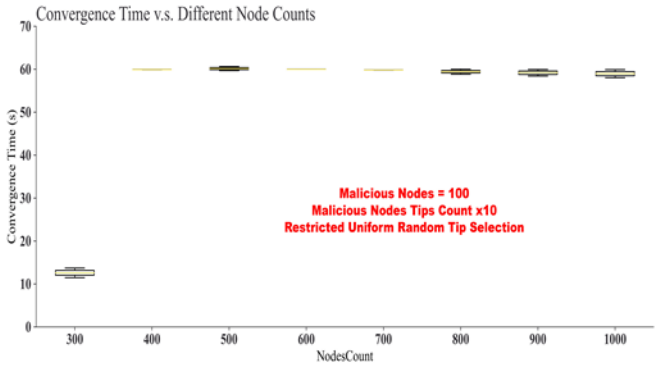


Fig 8. 100 malicious nodes, tipsCount_x10, RURTS

Fig 8 demonstrates convergence times for the RURTS network when the tipsCount allocated to 100 malicious nodes is multiplied by 10. In Fig 9, convergence times for URTS are similar to RURTS i.e. approx. 60 seconds.

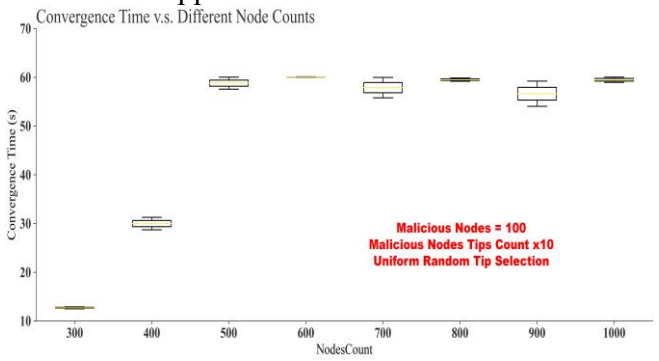


Fig 9. 100 malicious nodes, tipsCount_x10, URTS

In contrast, with 502 malicious nodes and their tipsCount amplified by 10, the RURTS network convergence time is still approximately 60 seconds. (Fig 10.)

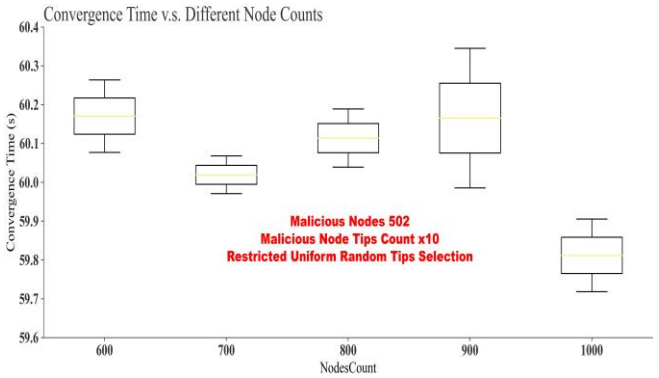


Fig 10. 502 malicious nodes, tipsCount_x10, RURTS

This compares unfavourably with URTS, which shows faster convergence times for networks of 600 nodes to 900 nodes. Note the URTS 600 node network convergence time of 20 seconds, this despite 502 malicious nodes (Fig 11).

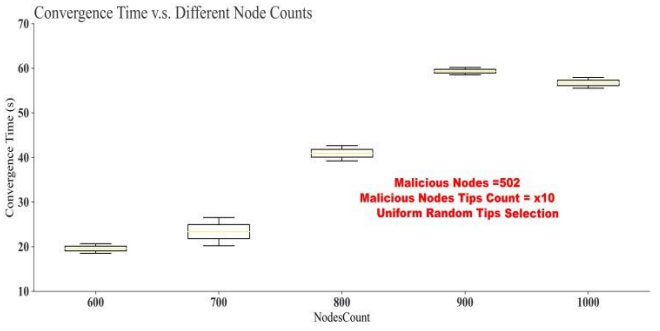


Fig 11. 502 malicious nodes, tipsCount_x10, URTS

The control for the simulation features a RURTS network and a URTS network with no malicious nodes.

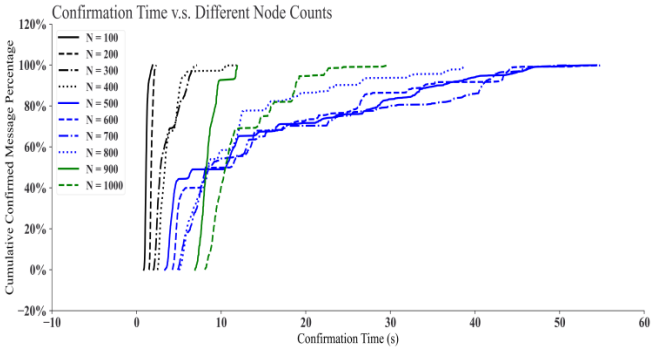


Fig 12. RURTS simulation with 0 malicious nodes.

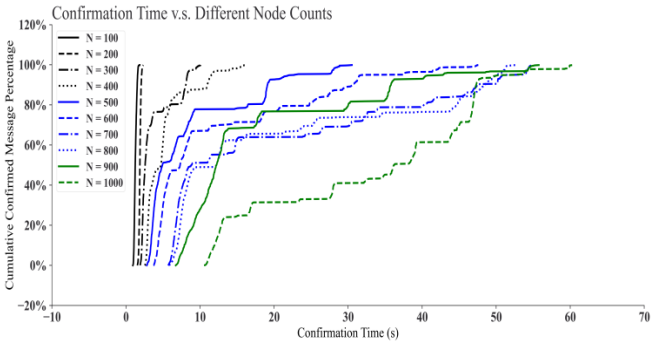


Fig 13: URTS simulation with 0 malicious nodes. Note time delay in networks with greater than 400 nodes. All networks converge by 60 seconds.

VI. PARASITIC TANGLES

An attacker that successfully dupes a network into accepting a malicious or fake node can begin to approve other fake nodes using the primary fake. As a result a "parasitic chain" of fake nodes may emerge from the genuine tangle and assume the role of approving new messages and gradually accumulate more cumulative weight with every approval [13]. (Recall that cumulative weight represents trusted pathways through a network.) As a result the genuine tangle is isolated and stagnates. Both Unbiased Random Walk (URW) and URTS are both susceptible to parasitic chains. Fig 14. shows the appearance of a sub-tangle and

7

the cumulative weight it acquires at the expense of the main tangle.

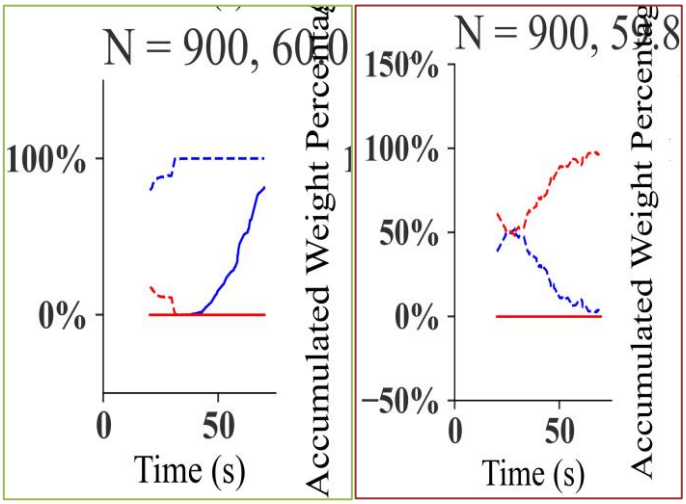


*Fig 14. **(left)** 100 malicious nodes - red, 800 honest nodes - blue, tipsCountx0.5 URTS. **(right)** 500 malicious nodes (red), 400 honest nodes (blue) tipsCount x10.URTS*

Fig 14 left, displays the blue (honest nodes) in the ascendancy and the red sub-tangle supressed. However with the malicious nodes increased both in quantity and number of messages (right) the cumulative weight accumulated by the red sub-tangle increases to a maximum while the blue honest (main) tangle declines. Dashed lines are messages that are registered or "liked" but not confirmed. Solid lines are confirmed messages.

VI. CONCLUSIONS

A critical component of any communications platform governing messages between moving vehicles is rapid delivery time. This work demonstrates the critical importance of securing the IOTA tangle network against the influence of malicious nodes which may result in delays. RURTS has a built in delay factor which is a disadvantage compared to URTS. With malicious nodes set to 100, RURTS performs marginally better than URTS. A network failure was prompted when the malicious node tipsCount was scaled by a factor of ten. With 1000 nodes, 100 of which were malicious, RURTS failed to complete. RURTS performs better when the ratio of malicious to honest nodes equalises. As this ratio increases URTS performs better. Analysis confirms that URTS becomes more efficient at larger tangle sizes with increased number of nodes [9]. In the case of URTS, the rate of tips that malicious nodes produce has less impact on the tangle's performance than the number of malicious nodes contained inside it. This 'malicious to honest' node ratio has more impact on the network as the number of malicious nodes dominates the number of total nodes. Moreover the influence of malicious nodes was less noticeable in smaller networks.

Malicious node attacks in the simulations were deliberately severe in order to try and replicate real world worst cases. The simulations are stochastic and a degree of randomness does affect the outcome. Graphs did vary to some degree with repetition. For this reason, simulations were repeated to confirm the findings. Fortunately, the randomness inherent to the simulations, did not substantially affect the outcome. A degree of randomness is expected in the tip selection algorithms. This improves coverage and ensures that lighter branches of the tangle are approved as well as dense areas. This encourages the tangle to include messages that might otherwise be ignored [6]

Both RURTS and URTS are high performance tip selection mechanisms. The disruption of RURTS based simulations was more pronounced than URTS with comparative numbers of malicious nodes (Figs 5 and 6). The experiments show a handicap in convergence times for RURTS possibly because of its extra security measure. The most pronounced differences appear in the simulations with less nodes. It should be noted however that some disruption was noted in the control simulations despite the fact that there were no malicious nodes. This may be a hardware processing issue in the CoreI7. In this case RURTS performed better. Both URTS and RURTS showed similar convergence times with 1000 nodes. This may be a factor of the Tips Per Second (TPS) value which was set to 100 for all simulations. It was decided to vary the tipsCount value and set TPS to static in an attempt to obtain better comparable data.Both URTS and RURTS converge rapidly with comparable confirmation times in healthy conditions. However with more malicious nodes, there exists a considerable risk of a parasitic chain attack especially with low message activity or in the early stages of infant tangle formation (Fig 14). Fig 5 shows that for RURTS, a ratio of at least two to one genuine to malicious messages is recommended to mitigate the attack and preserve the original Distributed Ledger.