International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:15, No:4, 2021

Blockchain's Feasibility in Military Data Networks

Brenden M. Shutt, Lubjana Beshaj, Paul L. Goethals, Ambrose Kam

***Abstract*—**Communication security is of particular interest to military data networks. A relatively novel approach to network security is blockchain, a cryptographically secured distribution ledger with a decentralized consensus mechanism for data transaction processing. Recent advances in blockchain technology have proposed new techniques for both data validation and trust management, as well as different frameworks for managing dataflow. The purpose of this work is to test the feasibility of different blockchain architectures as applied to military command and control networks. Various architectures are tested through discrete-event simulation and the feasibility is determined based upon a blockchain design's ability to maintain long-term stable performance at industry standards of throughput, network latency, and security. This work proposes a consortium blockchain architecture with a computationally inexpensive consensus mechanism, one that leverages a Proof-of-Identity (PoI) concept and a reputation management mechanism.

***Keywords*—**Blockchain, command & control network, discrete-event simulation, reputation management.

I. Introduction

IN the era of Multi-Domain Operations (MDO), militaries will be challenged to operate against near-peer threats in a heavily integrated, yet distributed, battlefield environment [1]. This evolution of warfare incentivizes adversaries to dis-integrate, disrupt, and exploit the operational underpinning of a fighting force: its communications and data networks [2]. To counter anticipated initiatives by one's adversaries, communication security should be of the highest priority, specifically with respect to command and control networks. Distributed ledger technology (DLT) is a potential means of accomplishing this objective based on its enhanced ability to validate data by cross-referencing dataflows across an array of independent nodes [3]. The issue in this approach, however, is that DLT requires data to attain some degree of consensus between nodes before it can be appended to the ledger, referred to as the blockchain [4]. This consensus incurs high computational costs on the network and may lead to the network being unable to accommodate its long-term expected dataflow. In our research, we focus on answering the question: Is the application of DLT within military command and control data networks feasible?

B.M. Shutt is a student researcher within the Department of Mathematical Sciences, United States Military Academy, West Point, NY 10996 USA (e-mail: brenden.shutt@westpoint.edu).

L. Beshaj is a Cyber Mathematics Fellow with the Army Cyber Institute, United States Military Academy, West Point, NY 10996 USA.

P.L. Goethals is an Academy Professor in the Department of Mathematical Sciences, United States Military Academy, West Point, NY 10996 USA.

A. Kam is a Chief Engineer and Fellow with Lockheed Martin Corporation, Bethesda, MD 20817 USA.

II. Foundation

A blockchain refers to a cryptographically secured distributed ledger with a decentralized consensus mechanism [5]. Public blockchains enable decentralized data processing and verification services through an anonymous community of data processing nodes, referred to in our research as full nodes. The most popular blockchain to date supports the cryptocurrency Bitcoin which launched in 2009 by Satoshi Nakamoto [5]; however, in 2013, public attention started to shift to how Bitcoin's decentralization and security concept could be used in a wider array of applications such as legal contracts, online marketplaces, and information management [6]. The public blockchain incentivizes full nodes by rewarding them cryptocurrency in exchange for solving computationally expensive hash puzzles whose solutions provide the cryptographical links between blocks of data in the blockchain. Private blockchains, such as Hyperledger [4], provide a similar service but only enlist a small group of trusted full nodes in a permissioned network environment who do not require incentives nor need to compute expensive hash puzzles. Soon, blockchain technology has the potential to provide immense value as a data validation system and a trust management system as data dependency continues to grow within the modern economy [6].

Blockchain technology is still a novel concept and its current capacity to manage modern dataflows is low. The blockchains of both Bitcoin and Ethereum must regulate their data processing capacity to prevent overload within their technical architectures and consensus mechanisms; Bitcoin specifies a limit on data size while Ethereum specifies a limit on the computational load per block [7]. As a result, the Bitcoin blockchain supports an average long-run processing capacity between 3 and 7 transactions per second [8] and the Ethereum blockchain supports an average long-run processing capacity between 10 and 15 transactions per second [9]. Compared to transaction processing companies, such as Visa, Inc. who claims a transaction capacity of 24,000 transactions per second [10], mainstream blockchains greatly lack throughput capacity. To facilitate security for the next generation of blockchain applications, throughput capacity must be raised without compromising the data integrity mechanisms signature to the blockchain concept and blockchain architectures must become more flexible to fit a larger array of use cases [11].

A. Literature Review

Due to the novelty of the blockchain concept, research fundamental to the application of blockchain in modern data systems is still underway. As a result, the majority of research highlighted in this literature review was published within the last five years and focuses on building the next generation of

more efficient and more scalable blockchains. This literature review will focus on three main concentrations: (1) trust and privacy, (2) performance as it relates to throughput and latency, and (3) network architecture designs for modern data systems.

Trust & Privacy: Trust and privacy are the cornerstone value-metrics of a blockchain. Nakamoto's original blockchain design aimed at directly solving the privacy concerns of online transactions, and he conceptualized the first public blockchain to preserve online privacy in a decentralized framework which establishes trust without an intermediary [12]. The tradeoff with blockchain is that more comprehensive trust mechanisms (such as Proof-of-Work consensus algorithms) induce greater costs and inefficiencies for processing and retrieving data within a blockchain. Many researchers suggest a reputation management framework that does not incur large computational loads [13], [14]. This trust mechanism rewards full nodes and users who engage in "honest" activities with greater permissions and efficiency while flagging those who engage in "unfair" or "dishonest" activities [14]. Reputation events can be cataloged, stored, and referenced using smart contracts when determining the level of implied trust in a transaction.

Reputation mechanisms in blockchain prove most applicable in decentralized marketplace platforms where buyers and sellers can exchange goods without requiring an intermediary. Current research addresses data retrieval inefficiencies by suggesting a scalable privacy-preserving query process that allows users to parse a blockchain using full nodes as proxies [15]. This process relies on trusting full nodes to perform queries without violating the data's cryptologic integrity. Some researchers offer a solution to further offset the risk of adversary full nodes by employing an air-gapped wallet framework that keeps private keys stored off untrusted networks [16], though such a design would certainly slow the private key retrieval process or rely on a centralized authority, a blockchain Oracle, to access the air-gapped storage devices.

Throughput & Latency: Performance at scale is the greatest challenge of the modern blockchain concept. Researchers are designing architectures to accommodate the dataflows within large data ecosystems, including the industrial Internet of Things (IoT). Some research suggests a GroupChain concept that greatly reduces consensus latency by using fog computing techniques to localize consensus within the network to a small group of full nodes instead of achieving consensus throughout the entire blockchain's node ecosystem [17]. In our research we refer to this as clustering consensus. This strategy increases efficiency by decreasing the computational workload required for transactions to be appended to the blockchain. Despite increasing efficiency, this strategy also increases both the forking rate within the blockchain and the risk of "dishonest" full nodes controlling the blockchain [17]. In blockchains, forks occur when a blockchain diverges into two or more potential paths and is solved when the majority of full nodes reach consensus on a single path. Forking only hurts the blockchain: forks incur greater computational costs in a network and increases the risk for lost or fraudulent transactions.

FastChain is a similar approach to GroupChain in that it forms consensus clusters by referencing a bandwidth monitor [18]. The intention is to choose full nodes with high bandwidths relative to the node writing the block. Full nodes with higher bandwidth communicate better with other full nodes, and thus can more effectively reach consensus and generate blocks. Despite the higher effective block rate, this concept induces a greater forking rate and creates a biased network environment where high-bandwidth full nodes can hold disproportionate control of the network. Other research suggests utilizing a Directed Acyclic Graph (DAG) structure that contrasts the traditional linear blockchain structure by appending levels of blocks at a time to a blockchain, instead of single blocks [19], [20]. This concept treats the blockchain less like a chain and more like a channel whose width can be adjusted across different levels. The overall theme of research in improving throughput and latency in blockchain networks is that there are tradeoffs: greater throughput usually sacrifices security and a lower latency usually increases the fork rate. What will determine where tradeoffs can be made largely depends on a blockchain's individual use case.

Architectural Design: There are three main blockchain architectures: public, private, and hybrid. Each blockchain type specializes in a different network feature and fits a different use case. Research suggests a hybrid approach to online auctions that uses a private blockchain to facilitate auctions without disclosing bidder information to the user pool and uses a public blockchain to enable and facilitate public access to auctions [21]. Both blockchains operate in sequence through smart contracts built into each blockchain platform and an access control mechanism that allows only authorized parties to view certain data. Additional research builds onto the hybrid architecture by suggesting an attribute-based access control that incorporates variable hash algorithms to give users secure and private access to their data [22]. This concept leverages variable hash algorithms that improve security and offer variable URLs for network users [22].

Other researchers focus their blockchain studies on breaking the blockchain across data layers. In their design, referred to as ChainSplitter, data ingestion from heterogeneous devices occurs at a source layer, nodes reach consensus in an overlay layer, then blocks of data are committed to the cloud layer for storage in the blockchain [23]. This research concentrates on the challenge of processing and validating industrial IoT data within an overlay layer that only holds the most recent blocks of a blockchain; this approach decreases the computational load of overlay nodes, thus allowing them to maintain their utility despite high transaction volumes [23]. Similar concepts to ChainSplitter and Groupchain suggest an alternative concept where a localized blockchain in the overlay layer creates "Bubbles of Trust" within an ecosystem of industrial IoT devices [4].

Blockchain architectures can also be designed to fit property management use cases. Some researchers promote a blockchain concept that leverages consensus mechanisms and

a distributed ledger to facilitate real estate markets that provide greater visibility, greater efficiency, and fewer costs than traditional methods [24]. This application of blockchain represents real assets through easily-transacted digital tokens and stores them in an immutable, highly-visible environment. These applications where blockchain adds value to data-based systems give a reason for the continued research of blockchain technology.

III. Method

The purpose of this research is to test the performance and feasibility of DLT in a simulated brigade-level command and control network. The two performance measures of interest in this research are maximum long-run stable throughput (transactions-per-second) and transaction latency. Our proposed data network is a network that facilitates the transaction of mission-specific data within and between levels of command at the brigade level. Non-military networks equivalent to our proposed network may include low-to-moderate volume corporate communications networks, non-sensor supply-chain networks, and private data repositories. According to MDO, brigades are expected to operate independently and thus will be required to host their own data networks [1], [2]. Possible data categories for this proposed network may be the location data of battlefield assets, mission-status data, or other non-sensor data transactions. We assume the size of a typical military brigade-sized unit to be 3,000 to 5,000 soldiers [25] and we assume a soldier to battlefield asset ratio of two, meaning that we assume 1,500 to 2,000 active devices will be sourcing data into our proposed distributed-ledger network at any time. Based on the nature of the data, we estimate an average of two transactions per device per minute, distributed according to an exponential distribution. This generates an average network dataflow between 0.83 and 1.11 transactions-per-second. We anticipate real-world brigade-level military command and control networks having both fewer devices and less activity than that described with our modeled assumptions; however, we would rather err on the side of overstressing as opposed to under-stressing parameters in our network model. To program and construct our model, the Simpy library in Python was utilized.

A. The Blockchain Model

The blockchain model used in our research operates according to this sequence:

1) Transaction requests arrive into the network according to an exponential distribution with an average interarrival time between 0.90 and 1.20 seconds – this is based on the average network dataflow being between 0.83 and 1.11 transactions-per-second. Across trials, the average transaction interarrival time for each trial is determined according to a uniform distribution between 0.90 and 1.20 seconds. The exponential arrival distribution for the transaction interarrival times was chosen to maintain simplicity within the model while catering to the stochastic nature of the network traffic.
2) Transactions are processed by the first available network full node. The queuing logic for transactions within the network is First Come First Served (FCFS), and the number of full nodes is based on the specific blockchain architecture. Full nodes are assumed to have an average download speed of 2 MB/s and an average internal processing speed of 20 MB/s, both distributed using an exponential distribution. In support of feasibility testing, processing speeds believed to be on the lower-end of realistic processing speeds for potential full nodes were chosen. This method compensates for operating environments with connection limitations and the possible implementation of low-capability, often portable, network machines. The transactions are assumed to be 0.2 KB in size. This data size is considered standard for plaintext and plaintext-equivalent data networks and it reflects the data size of Ethereum network transactions [9].
3) Once processed, transactions are relayed to all nodes in the network at the earliest convenience. The established network protocol within our model is that processing new transactions and writing blocks of transactions takes priority over relaying individual transactions. Our protocol scheme is strict and does not offer much flexibility in protocol priority which could be implemented through more advanced network concepts.
4) Blocks are created by a full node within the network. An assigned full node writes all of its known transactions to the block. If a block is unaware of a transaction, then that transaction is not written to its block and must wait for a block to be written by a node that is aware of it. The block creation rate is the rate at which the network generates new blocks and is based on the blockchain architecture and consensus mechanism built into the network. The times at which blocks are expected to be written is referred to as the block schedule.
5) Blocks of transactions, once written by a full node, are then relayed to all active full nodes. Full nodes are then tasked with validating the block itself and the transactions within the proposed block. The extent to which blocks must agree before appending a block to the active blockchain is referred to as the consensus scale. The consensus scale is essentially the proportion of full nodes that much confirm the validity of the transactions in a candidate block before that block can be appended to the active blockchain.
6) Once the transactions within a certain block survive the creation of some number of additional blocks without being revoked or invalidated in the blocks that follow, those transactions are considered to be mature. From a network user's perspective, mature transactions are trusted to be authentic with a high degree of confidence because they have not been canceled or deemed invalid in subsequent blocks.
7) The duration of each trial simulation is 3,000 seconds, or 50 minutes. This will test the network's ability to process between 2,500 and 3,330 transactions per simulation, as well as allow us to determine the stability of network performance over that time. If the network fails to

converge towards stable performance during any trial in the performance test, we increase the length of the time until we can determine the long-term stability of the network. Each performance test consisted of 30 individual trials for our baseline and alternative design evaluations.

B. Baseline Design

Our experiment's baseline model is based on Ethereum 1.0's Proof-of-Stake (PoS) concept operating on a permissioned network. To mimic Ethereum 1.0, our baseline model generates new blocks according to a uniform distribution between 10 and 19 seconds [26]. Based on the scope of our proposed network, we determined that 50 full nodes will operate on the network to facilitate the creation and validation of both transactions and blocks. The consensus scale of the baseline model is 50%, meaning that more than half of the full nodes must validate a block of transactions prior to that block being appended to the blockchain. In our baseline model, we do not incorporate a reputation management mechanism that may alter the size of data transactions and thus increase the computational cost of operating the network. Lastly, the quantity of subsequent blocks required for a block of transactions to be deemed mature in our baseline model is 10 blocks. This compares to the bitcoin network which requires 6 blocks, or roughly 60 minutes, for blocks of transactions to be deemed mature [27]. Ethereum's PoS concept enables it some flexibility in its blocks-to-maturity requirement with research suggesting that 10 to 37 blocks are required to ensure the same degree of confidence as the bitcoin network's standard [27], [28]. We accept a 10-block maturity standard in our baseline model since our baseline's consensus algorithm is comparable to Ethereum 1.0 and operates in a small-scale permissioned network environment, a considerably more secure environment than the Ethereum network [28].

C. Baseline Performance

Our 30-trial performance test on the baseline model indicates that the baseline model's application in a brigade-level military command and control network is feasible. We determine this feasibility based upon the long-run stable performance achieved in each trial. In the baseline model, the average time for a transaction to relay to 100% of the full nodes or be written to a block (whichever occurs first), is 0.061 seconds with an average standard deviation per trial being 0.034 seconds. Within each trial, the majority of full relays follow a uniform distribution roughly between 0.005 and 0.11 seconds, with some outlier instances requiring up to 0.2 seconds to achieve a full relay within the network. The trial-specific distribution of full-relay times is depicted in Fig. 1.

The average time for a transaction to be written to a block in the baseline model is 7.602 seconds, with the average standard deviation within each trial being 4.559 seconds. The time it takes for a block to be written to the blockchain is important because it signifies the first chance that a network user can view data that are appended on the network's blockchain. Within each trial, the distribution of write times is unique with the majority of transactions being written to a block within 12.5 seconds and no write times occurring after 19 seconds. This makes sense given that the block creation rate for the baseline model is a uniform distribution between 10 and 19 seconds. The trial-specific distribution of transaction write times is depicted in Fig. 2.

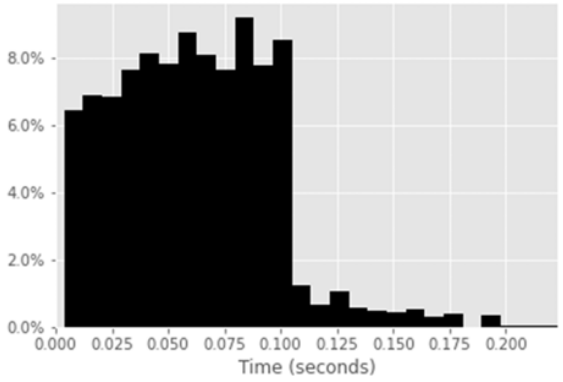

Fig. 1 Distribution of Transaction Full Relay Times in Baseline

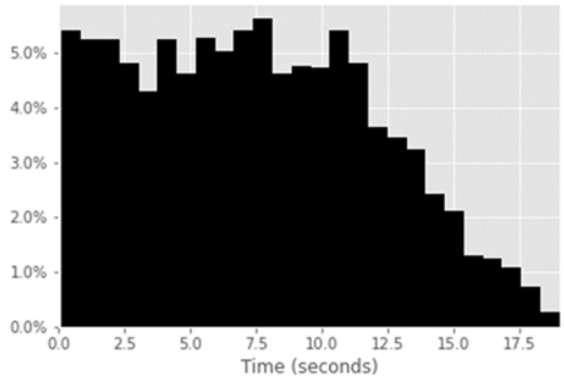

Fig. 2 Distribution of Transaction Block-Write Times in Baseline

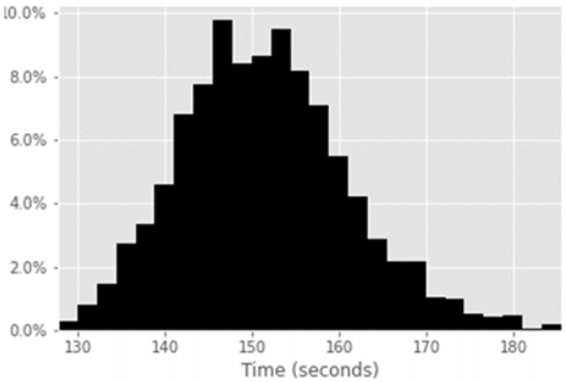

Fig. 3 Distribution of Transaction Maturity Times in Baseline

The average time for a transaction to reach maturity status in our baseline model is 152.403 seconds with an average standard deviation of 9.670 seconds for each trial. The maturity times for each trial are normally distributed. This

distribution makes sense given that the maturity time is the average of the ten sequential block-write times in the network. Given that the distribution of block-write times is uniform between the values of 10 and 19, the Central Limit Theorem supports our finding that the distribution of maturity times – which is an average of 10 sequential block-write times – is normally distributed with respect to the global mean [29], [30]. The trial-specific distribution of transaction maturity times is depicted in Fig. 3.

D.Sensitivity Analysis

Sensitivity analysis allows us to determine the effect that incremental changes of network parameters have on the performance of the network. The purpose of the analysis is to determine which network components have the greatest influence on network performance. In the process of sensitivity analysis, four parameters are analyzed: (1) the number of network nodes, (2) the interarrival rate of transactions entering the network, (3) the consensus scale required by the network, and (4) the average size of transactions on the network. We performed a sensitivity analysis on the baseline model by individually and incrementally changing each parameter of interest and measuring the latency of transactions across different phases of the network.

Network Nodes: A sensitivity analysis was conducted to investigate the latency of the baseline model across different quantities of full nodes, from 10 to 200. Our analysis suggests that there is a positive linear relationship between the number of network nodes and the time required for transactions to attain a 100% relay among the full nodes.

Specifically, the relationship between the number of network nodes (n) and a transaction's full relay time (T_R) in seconds is

$$T_R = 0.0553 + 0.000107n \quad (1)$$

This relationship is logical because a greater number of nodes in a network requires that a transaction be relayed more to attain a 100% relay status. The relationship between nodes and the average full relay times is further depicted in Fig. 4.

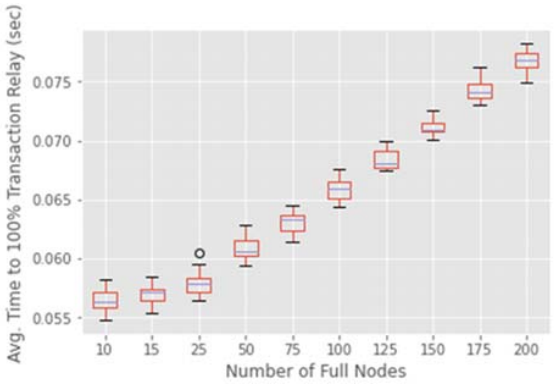

Fig. 4 Number of Nodes Sensitivity on Transaction Full Relay Times

The time it takes transactions to be written to blocks and to attain a maturity within the network is not significantly affected by the quantity of nodes within the network. This is because the network's protocol hierarchy puts a priority on creating new blocks and receiving new data transactions. It is anticipated that with quantities of nodes much greater than 200, transactions will take longer to relay throughout the full node population. Resultingly, an increase in nodes will further decrease the probability that a transaction is relayed to the full node that is selected to create the next block, thus increasing the average time for a transaction to be written to a block. For this research, we do not expect the application of our proposed model to utilize more than 200 active full nodes, especially given that this study involves a permissioned network.

Transaction Arrival Rate: A sensitivity analysis also examined the latency of the baseline model across different frequencies of transaction arrivals, from 0.25 to 9 transactions-per-second (tps). Our analysis suggests that there is a positive quadratic relationship between the average frequency of transaction arrivals and the time required for transactions to attain a 100% relay among the full nodes.

Specifically, the relationship between the arrival frequency of transactions (t) in tps and a transaction's full relay time (T_R) in seconds is

$$T_R = 0.0074t^2 - 0.0288t + 0.0819 \quad (2)$$

This relationship establishes that the average time for a transaction to achieve 100% relay becomes more sensitive to the arrival frequency of transactions as the frequency of transaction arrivals increases. This quadratic relationship is logical for distributed networks because these networks have a comprehensive set of functional dependencies and therefore are more susceptible to having their performance suppressed due to heightened levels of network traffic compared to traditional network architectures. The relationship between transaction arrival rate and the average full relay times is further depicted in Fig. 5.

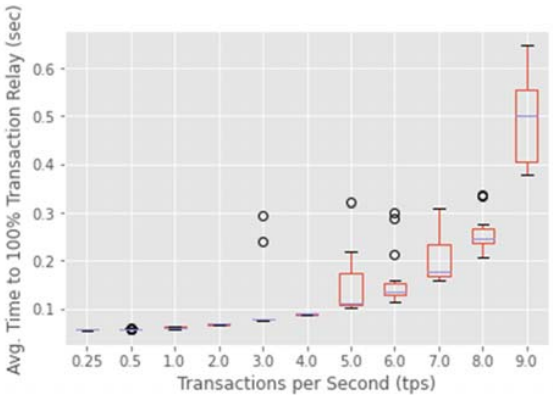

Fig. 5 Transaction Rate Sensitivity on Transaction Full Relay Times

Our analysis also suggests that there is a positive linear relationship between the average frequency of transaction arrivals and the time required to write a transaction to a block.

Specifically, the relationship between the arrival frequency of transactions (t) in tps and a transaction's time to be written to a block (T_W) in seconds is

$$T_W = 7.4907 + 0.0473t \quad (3)$$

The effect of the transaction arrival frequency on the time it takes a transaction to be written to a block is likely caused by the network's diminished ability to relay transactions between full nodes under high dataflows. This causes each transaction to be less visible to the community of full nodes and decreases the probability that a transaction is logged by the full node that is selected to write the next block to the blockchain. This increased time to write a transaction is echoed in the time it takes transactions to achieve maturity, meaning that higher transaction arrival frequencies also increase the time it takes a transaction to achieve maturity in the network. Figs. 5 and 6 communicate that both the variance in the time to achieve a full transaction relay and to write a transaction to a block, respectively, increases with an increased transaction arrival frequency. When testing the baseline model, it proved capable of maintaining stable long-term performance with an average transaction arrival rate of 10 tps, or lower. This is consistent with Ethereum 1.0's long-term transaction capacity [9].

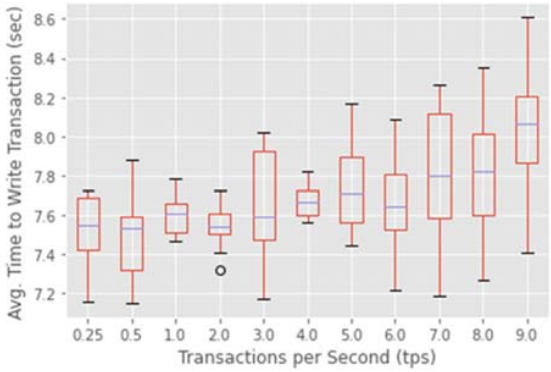

Fig. 6 Transaction Rate Sensitivity on Block-Write Times

Consensus Scale: Sensitivity analysis was administered to examine the latency of the baseline model across different consensus scales, from 10% to 75%. Our analysis suggests that there is no statistically significant relationship between the consensus scale and the latency of the network at any phase. This is likely because the quantity of nodes in the baseline model – 50 nodes – is so low that a more comprehensive consensus scale only requires a marginally greater number of nodes to valid the candidate block before it is appended to the network's blockchain. For the baseline model, the difference between a 50% consensus scale and a 75% consensus scale is effectively 13 full nodes. However, in a larger network of 2,000 active full nodes, the same difference in consensus scale would be 500 full nodes. This analysis suggests that consensus scale is not sensitive for smaller networks – such as in our baseline model – but will likely grow in sensitivity within a network as the number of full nodes increases.

Transaction Size: Finally, sensitivity analysis was performed on the latency of the baseline model across different average transaction sizes, from 0.2 to 1.0 KB. Our analysis suggests that there is a positive linear relationship between the average size of transactions and the time required for transactions to attain a 100% relay among the full nodes. Specifically, the relationship between the size of transactions (s) in KB and a transaction's full relay time (T_R) in seconds is

$$T_R = 0.0560 + 0.0252s \quad (4)$$

This relationship is logical because larger data transactions will increase the computational cost of processing and relaying transactions throughout the network, thus increasing the average time required for a transaction to achieve 100% relay within the network. The relationship between average transaction size and the average full relay times is further depicted in Fig. 7.

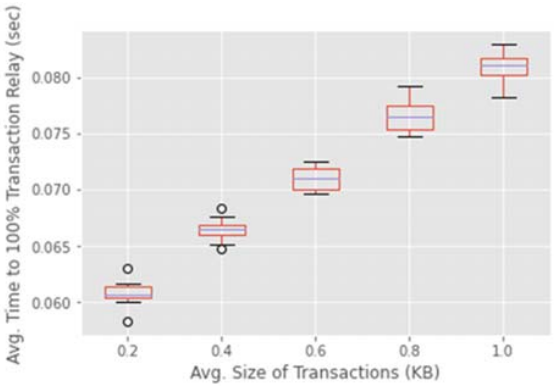

Fig. 7 Transaction Size Sensitivity on Transaction Full Relay Times

Similar to the sensitivity analysis on the number of nodes in a network, the size of transactions has no significant effect on the average time it takes transactions to be written to blocks or to achieve maturity within the network. We expect that large data transactions – such as video, audio, or image data – will significantly decrease the distributed network's ability to process, relay, and write data. Though, we do not anticipate that our proposed command and control network will host average data transaction sizes larger than 1 KB and therefore we did not incorporate them into our sensitivity analysis.

E. Alternatives

The purpose of examining alternative models in our research is to compare the performance of differently configured distributed-ledger network architectures. An analysis of performance will enable us to determine which architecture designs perform better and which architectures perform worse, relative to the baseline and other alternatives. Based upon the research findings, three alternatives in addition to the baseline model are proposed: (1) a minimum security network design that utilizes a Proof-of-Elapsed-Time (PoET) consensus algorithm, (2) a consortium network design that

utilizes a PoI consensus algorithm, and (3) a security-focused network design that, like the baseline, utilizes a PoS consensus algorithm but with a more comprehensive consensus scale, a larger full node population, and the addition of a reputation management system.

TABLE I
ALTERNATIVE BLOCKCHAIN NETWORKS

Alternative	Nodes	Consensus Algorithm	Consensus Scale	Reputation Management
Baseline	50	Ethereum 1.0	50%	No
Minimum Security	25	PoET	33%	No
Consortium	75	PoI	50%	Yes
Comprehensive Security	100	Ethereum 1.0	75%	Yes

The baseline model is meant to portray a standard blockchain architecture with a moderately comprehensive consensus mechanism and no additional mechanisms aimed at increasing either performance or security. In contrast, the alternatives are meant to depict blockchains with different priorities and purposes. For instance, the minimum security network, which can also be referred to as the high performance network, is built with the intent of increasing network performance at the cost of minimal network security. This alternative features a PoET consensus algorithm that creates a new block every 1 second within the network and a consensus mechanism that selects 66.7% of the full nodes at random to form a static cluster. This static cluster is then tasked with validating each block with over 50% consensus. This strategy enables the network to only require 33.3% of full nodes in the network to validate newly created blocks. The idea behind this static cluster approach is to free-up the already small full node population from having to validate blocks so they can instead receive and relay new transactions. The major assumption with the permissioned-private network is that the network is restricted to only pre-authorized full nodes and does not allow unidentified and potentially dishonest nodes from contributing to the network. Due to the reduced number of nodes, the expedited consensus algorithm, and the small consensus scale, the network is susceptible to being compromised if malicious actors gain access to the network.

The consortium network design provides a duality of features, with some features focused on performance and other features focused on security. The consortium blockchain would, in theory, operate across layers of command with a mixture of battalion and brigade-level assets collaborating within the network, as opposed to only brigade-level assets operating as full nodes. This allows a greater number of full nodes to operate on the network. The consensus mechanism for the consortium network is PoI which relies heavily on cryptographic keys to identify honest nodes within the network [31]. A node's key determines a full node's authenticity and thus enables greater network flexibility than a PoS concept because there is a pre-established trust between full nodes. A conservative estimate is that this established trust will allow a block creation rate between 5 and 10 seconds according to a uniform distribution. A more rapid block creation rate may be feasible if this network is not of a consortium design and the network machines are assumed to be more directly connected at the brigade-level, though this may increase the network's forking rate. Despite the increased potential for performance, the caveat to PoI is that network activity needs to be tracked using a reputation management system [31]. The reputation mechanism helps ensure accountability between nodes and creates a system of checks and balances between nodes within the network. To accomplish this task, network activity for each full node is collected in a communal database for universal reference. The downside of a reputation management system is that it will incur a greater computational cost for full nodes to operate on the network because the nodes must process the transactions themselves as well as the meta-data associated with the specific network activity. A conservative estimate is that this reputation mechanism will increase the computational cost of operating on the network fivefold.

Lastly, the security-focused network design concentrates solely on providing comprehensive security to the network. It does so by adding 50 full nodes to the node population from the baseline model, making it considerably more difficult to compromise the node population with a 51% attack. Since the Ethereum 1.0 consensus algorithm requires substantial and continuous effort from full nodes for them to maintain their ability to operate on the network, Ethereum 1.0 is chosen to operate on the security-focused network. Unlike the baseline model, however, the security-focused model requires 75% of active nodes to validate a block before it can be appended to the active blockchain, a 50% increase from the baseline model. The intent behind a larger consensus scale is to reduce the probability of an effective spoof on the distributed-ledger network. Like the consortium design, the security-focused network will host a reputation management system that requires nodes to process network-specific meta-data in addition to transactions which both increase the computational cost required by the network and also provide a means of accounting for node behavior within the network.

IV. ALTERNATIVE BLOCKCHAIN PERFORMANCE

A. Network Latency

When assessing the latency of distributed networks, the most important factor to consider is the time it takes for transactions to be written to blocks upon entering the network. When a transaction is written to its block, it presents the first opportunity for the user to see the transaction and possibly incorporate its underlying data into additional networks or systems. The average time to write transactions to blocks for each alternative network across 30 trials is summarized in Fig. 8. The time it takes for a transaction to attain a 100% relay to all full nodes aids in identifying network potential but it offers little direct value to the network users. Additionally, the time it takes for a network to attain maturity is not a known, nor uniform standard [28]. Without extensive network-specific testing, one cannot definitively conclude how many block

creation cycles are required for a transaction to attain some critical degree of maturity. For the baseline model in this study, the Ethereum 1.0 standard of 10 blocks was followed, but for the alternative networks, the number of blocks required to attain transaction maturity is unknown and therefore we cannot use the maturity time to effectively compare performance between these alternatives.

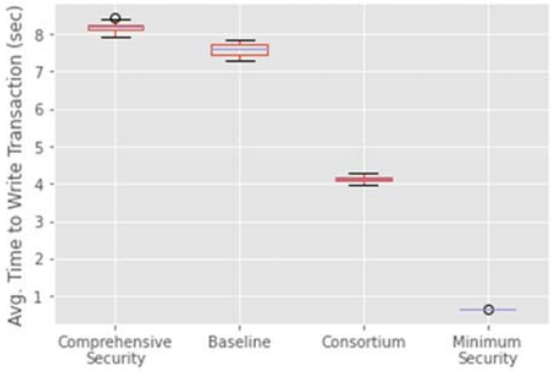

Fig. 8 Comparison of Average Alternative Block-Write Times

The results of our experiment identified that the minimum security network hosts the most optimal transaction latency of any alternative network, as expected. Across 30 trials, the average time for a transaction to be written to a block is 0.654 seconds, with an average standard deviation of 0.295 seconds. The lack of security mechanisms in this alternative is significant compared to the baseline and the other alternative networks. The PoET consensus mechanism empowers low transaction latency but does not vet potentially malicious actors [32], and the static cluster consensus scale potentially compromises honest network actors to invalidate transactions prior to them being written to blocks [17]. This alternative would best be utilized if the user wanted a distributed network that performed closer to a traditional network, with a slight increase in network security from the network's distributed nature. A possible improvement to this network would be reinstating the 50% consensus scale as our sensitivity analysis concluded it had no statistically significant effect on network latency, especially within small networks.

Based upon experimental results, the second most optimal network in terms of latency was the consortium network which exhibited an average transaction write time of 4.145 seconds with an average standard deviation of 2.397 seconds. Compared to the minimum security network, the consortium network offers a larger variance relative to the average in write times for its transactions. This larger variance indicates a greater relative uncertainty in the consortium network when compared to the minimal security network. However, the consortium network offers a more secure consensus mechanism and a 75-node architecture, in addition to the use of a reputation management system. For users who need a more than minimally secure network but also need a transaction latency better than our baseline network, the consortium network is likely the best candidate as it offers a fairly comprehensive set of security measures while also offering a latency half that of the baseline network design.

The worst performing alternative network, in terms of latency, was the comprehensive security network. This network offered the same consensus mechanism as the baseline but doubled the number of full nodes, increased the consensus scale by 50% from the baseline, and included a reputation management system. The comprehensive security network demonstrated an average transaction write time of 8.188 seconds with an average standard deviation of 4.590 seconds. Most notably, this network alternative performs worse than the baseline model at the expense of additional security measures. This network would be a favored alternative if the user greatly favored security over latency in the network.

B. Network Throughput

Quite peculiarly, each alternative network design had the same maximum average transaction throughput: 10 tps. Extensive research into the networks was performed to find that this is likely due to the strict and inflexible protocol hierarchy used by each network. As previously mentioned, each network node has priorities of work: the first priority is writing and validating new blocks of transactions, the second priority is receiving transactions from data sources, and the last priority is relaying transactions. This work concludes that the failure to attain an increased maximum throughput from the baseline was due to transactions not being effectively relayed in the network. A further analysis found that once a proportion of transactions in the network went unwritten in a block creation cycle (the specific value was not consistent between trials and network designs), then the network never recovered and from that point only increased the proportion of transactions that go unwritten to blocks during each block creation cycle. This continues until the majority of processed transactions are only known by the specific nodes that processed them and thus can only be written to blocks by the node through which it was processed.

Additionally, each alternative had logical tradeoffs in the amount of throughput it could handle. The minimal security network had only 25 network nodes so when the transaction arrival rate increases, the utility of each node rapidly stretches until each node is being overutilized. The consortium and comprehensive security network had more nodes than the baseline design and thus it was more difficult for each transaction to be fully relayed within the network, especially since each relay required additional computation due to the reputation management protocols inherent in both networks.

V. Future Work

To improve our findings, future research should focus on quantifying the value of security for different network architectures. This research effort defined network performance only in terms of throughput and latency but, debatably, the most appealing feature of distributed networks is the security it provides in otherwise untrustworthy environments. Future work could leverage our findings on the

relative performance of contrasting network designs to suggest with greater confidence what the optimal distributed network is for this use case. Depending when future research occurs, Ethereum 2.0 could be used in place of Ethereum 1.0 for modeling purposes. Such research could focus on quantifying the improved capability of the Ethereum network across versions and could identify the potentially greater applicability of the Ethereum concept in military data networks. Future work should also focus on replicating our study by using actual networks operating with real data transactions within environments expected of the use case. This strategy would provide an opportunity to validate our findings and determine with greater confidence the feasibility of different network architectures across dynamic dataflows. This method would also provide researchers the opportunity to introduce simulated network threats and dishonest nodes to determine which networks are better at repelling threats and which networks are unable to routinely deter expected attack vectors.

VI. Conclusion

In the era of MDO, militaries will encounter a myriad of new threats on the cyber front. As forces build their capacity to defend themselves from these cyber threats, DLT may be a potential solution for protecting low to moderate volume data networks. Our research determines that distributed networks are capable of maintaining stable long-run performance when processing dataflows expected of a brigade-level military command and control network. The main tradeoff is between security and performance where an increase in network security measures results in a decrease of network latency, and vice-versa. The exact tradeoff point between network performance and security is ultimately up to the network user; this research work proposes different feasible network designs that offer different combinations of performance and security. This research study evaluates the consortium design to be the best performing network among the alternative designs. Its construct includes a cryptographically-focused PoI consensus mechanism, a standard 50% consensus scale, 75 nodes distributed across layers of command within a brigade, and a reputation management system that maintains a record of network activity as a means of determining the authenticity of data and the trustworthiness of network nodes. Future work on this topic may focus on quantifying the value of security within each network design and testing these designs within actual networks as a means of further determining the feasibility of DLT within brigade-level military command and control networks.

References

[1] A. Feickert, "Defense Primer: Army Multi-Domain Operations (MDO)", in *LexisNexis® Congressional Research Digital Collection*, 2020, pp. 1-3.

[2] Maj. K. Nettis, "Multi-Domain Operations: Bridging the Gaps for Dominance, " *Wild Blue Yonder*, 1 March 2020.

[3] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi and J. Wang, "Untangling Blockchain: A Data Processing View of Blockchain Systems," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366-1385, 1 July 2018.

[4] M.T. Hammi, B. Hammi, P. Bellot and A. Serhrouchni, "Trust-free systems: A literature review on blockchain technology and trust in the sharing economy," in *Electronic Commerce Research and Applications*, vol. 29, pp.50-63, 2019.

[5] M. Swan, *Blockchain: Blueprint for a New Economy* (Book style), 1st ed., Sebastopol, CA: O'Reilly, 2015, pp. 9-21.

[6] W. Mougayar, *The Business Blockchain: promise, practice, and application of the next Internet technology* (Book style), Hoboken, NJ: John Wiley & Sons, Inc., 2016, pp. 132-144.

[7] H. Paik, X. Xu, H. M. N. D. Bandara, S. U. Lee and S. K. Lo, "Analysis of Data Management in Blockchain-Based Systems: From Architecture to Governance," in *IEEE Access*, vol. 7, 2019, pp. 186091-186107.

[8] Blockchain, "Blockchain Charts" (Data repository), *Blockchain*, 2020 https://www.blockchain.com/charts/.

[9] Etherscan, "The Ethereum Blockchain Explore" (Data repository), *Ethereum*, 2020. https://etherscan.io/.

[10] Visa Inc., "Visa Acceptance for Retailers" (Company website), *Visa*, https://usa.visa.com/run-your-business/small-business-tools/retail.html.

[11] V. Morabito, *Business Innovation through Blockchain* (Book style), Cham, Switzerland: Springer Nature, 2017, pp. 34-70.

[12] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World* (Book style), New York, NY: Penguin Random House LLC, 2016, pp. 39-50.

[13] S. Malik, V. Dedeoglu, S. S. Kanhere and R. Jurdak, "TrustChain: Trust Management in Blockchain and IoT Supported Supply Chains," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 184-193, 2019.

[14] T. Salman, R. Jain and L. Gupta, "A Reputation Management Framework for Knowledge-Based and Probabilistic Blockchains," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 520-527, 2019.

[15] S. Linoy, H. Mahdikhani, S. Ray, R. Lu, N. Stakhanova and A. Ghorbani, "Scalable Privacy-Preserving Query Processing over Ethereum Blockchain," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, 2019, pp. 398-404.

[16] A. Davenport and S. Shetty, "Air Gapped Wallet Schemes and Private Key Leakage in Permissioned Blockchain Platforms," *2019 IEEE International Conference on Blockchain (Blockchain),* Atlanta, GA, pp. 541-545, 2019.

[17] K. Lei, M. Du, J. Huang and T. Jin, "Groupchain: Towards a Scalable Public Blockchain in Fog Computing of IoT Services Computing," in *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 252-262, 1 March-April 2020.

[18] K. Wang and H. S. Kim, "FastChain: Scaling Blockchain System with Informed Neighbor Selection," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 376-383, 2019.

[19] S. Yang, Z. Chen, L. Cui, M. Xu, Z. Ming and K. Xu, "CoDAG: An Efficient and Compacted DAG-Based Blockchain Protocol," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 314-318, 2019.

[20] K. Tsoulias, G. Palaiokrassas, G. Fragkos, A. Litke and T. A. Varvarigou, "A Graph Model Based Blockchain Implementation for Increasing Performance and Security in Decentralized Ledger Systems," in *IEEE Access*, vol. 8, pp. 130952-130965, 2020.

[21] H. Desai, M. Kantarcioglu and L. Kagal, "A Hybrid Blockchain Architecture for Privacy-Enabled and Accountable Auctions," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 34-43, 2019.

[22] H. Guo, W. Li, M. Nejad and C. Shen, "Access Control for Electronic Health Records with Hybrid Blockchain-Edge Architecture," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 44-51, 2019.

[23] G. Wang, Z. Shi, M. Nixon and S. Han, "ChainSplitter: Towards Blockchain-Based Industrial IoT Architecture for Supporting Hierarchical Storage," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 166-175, 2019.

[24] S. Latifi, Y. Zhang and L. -C. Cheng, "Blockchain-Based Real Estate Market: One Method for Applying Blockchain Technology in Commercial Real Estate Market," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, pp. 528-535, 2019.

[25] U.S. Department of Defense, "Military Units: Army," *U.S. Department of Defense*, 2021. https://www.defense.gov/Experience/Military-Units/Army/.

[26] YCharts, "Ethereum Average Block Time" (Data repository), *Ycharts*, 2021. https://ycharts.com/indicators/ethereum_average_block_time/.

[27] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, Srdjan Capkun, "On the Security and Performance of Proof of Work Blockchains," *Association for Computing Machinery*, New York, NY, 2016.
[28] V. Buterin, "On Slow and Fast Block Times," *Ethereum Foundation*, Bern, Switzerland, 14 September 2015.
[29] S. G. Kwak, J. H. Kim, "Central Limit Theorem: The Cornerstone of Modern Statistics," in *Korean Journal of Anesthesiology*, vol. 7, no. 2, pp. 144-156, 2017.
[30] S. Savage, *The Flaw of Averages* (Book style), Hoboken, NJ: John Wiley & Sons, 2015, pp. 67-77.
[31] X. Yang, W. Li, "A Zero-Knowledge-Proof-based Digital Identity Management Scheme in Blockchain," in *Computers & Security*, vol. 99, 2020.
[32] L. Chen, L. Xu, N. Shah, Y. Lu, "On Security Analysis of Proof-of-Elapsed-Time (PoET)," in *Stabilization, Safety, and Security of Distributed Systems*, pp. 282-297, 7 October 2017.