FBA-SDN: A Federated Byzantine Approach for Blockchain-based Collaborative Intrusion Detection in Edge SDN

John Hayes, Adel Aneiba, Mohamed Gaber, Md Shantanu Islam, Raouf Abozariba
School of Computing, Engineering and the Built Environment
Birmingham City University, Birmingham, UK
{*john.hayes, adel.aneiba, mohamed.gaber, mdshantanu.islam, raouf.abozariba*}*@bcu.ac.uk*

***Abstract*—This paper presents FBA-SDN, a novel Stellar Consensus Protocol (SCP)-based Federated Byzantine Agreement System (FBAS) approach to trustworthy Collaborative Intrusion Detection (CIDS) in Software-Defined Network (SDN) environments. The proposed approach employs the robustness of Byzantine Fault Tolerance (BFT) consensus mechanisms and the decentralized nature of blockchain ledgers to coordinate the Intrusion Detection System (IDS) operation securely. The federated architecture adopted in FBA-SDN facilitates collaborative analysis of low-confidence alert data, reaching system-wide consensus on potential intrusions. Additionally, the Quorum-based nature of the approach reduces the risk of a single point of failure (SPoF) while simultaneously improving upon the scalability offered by existing blockchain-based approaches. Through simulation, we demonstrate promising results concerning the efficacy of reaching rapid and reliable consensus on both binary and multi-class simulated intrusion data compared with the existing approaches.**

Index Terms—blockchain, SDN, collaborative intrusion detection, scalability, trust management

1. Introduction

Software-Defined Networks (SDN) are a rapidly evolving field with the potential to revolutionise how networks are designed, operated, and secured. One of the primary benefits of SDN is the ability to centralise the control plane and programmatically configure network devices, which in turn improves network visibility, scalability, and flexibility [1]. Furthermore, the virtualisation of network services within edge and fog layers offers multitudinous possibilities for intermediary data analysis, particularly when coupled with traditional resource-constrained IoT networks in edge-cloud scenarios. However, the exponential growth in complexity and size of networks has consequently resulted in an increased attack surface. Therefore, the security of SDN environments has become a significant concern. Intrusion detection and response are critical components of both traditional and SDN network security, but existing approaches tailored towards traditional architectures are often inadequate in dealing with the dynamic and distributed nature of modern network attacks and advanced attack payloads [2].

Collaborative Intrusion Detection Systems (CIDS) emerged recently as one of the potential resolutions, facilitating information exchange and active collaboration between multiple Intrusion Detection Systems (IDS), aiming towards rapidly identifying and responding to network threats [3]. This approach is not unique to SDN; several solutions were deployed to significant effect within traditional network architectures. However, this evolution in IDS architecture has prompted a renewed interest in the composite elements of secure distributed systems and how one can translate these into security-centric solutions, such as CIDS. Scalable data exchange and storage, membership management, and inter-node trust schema propose challenges that often require novel and innovative technological solutions.

Blockchain technology has been proposed as a way to facilitate decentralised, non-repudiable, and tamper-proof consensus-reaching among participant nodes [4]. However, existing blockchain-based CIDS approaches often suffer from scalability and performance limitations [5]. The consensus-reaching process in CIDS can be challenging, particularly across large-scale and distributed networks, such as those deployed in SDN environments [5]. Advances in distributed consensus algorithms, fuelled by continued research interest in the blockchain domain, and swarm learning – a novel distributed approach to ensemble fusion – present a potentially symbiotic approach to resolve this scalability deadlock. Despite exhibiting desirable protocol characteristics, formal analysis and verification [6] [7], and underpinning several of the largest public financial exchange blockchains [8], FBAS-based consensus approaches have seen little consideration within alternative deployment domains, including security-centric applications.

This paper presents FBA-SDN, a novel Stellar Consensus Protocol-based (SCP) [8] Federated Byzantine Agreement System (FBAS) approach to trustworthy Collaborative Intrusion Detection in Software-Defined Network environments. The proposed approach employs the robustness of Byzantine Fault Tolerance (BFT) consensus mechanisms and blockchain ledgers' decentralized nature to coordinate IDS operations securely. The federated architecture adopted in FBA-SDN facilitates collaborative analysis of low confidence and challenge-based alert data, enabling system-wide consensus on potential intrusions. Additionally, the Quorum-based approach reduces the risk of a single point of failure

while improving upon the scalability offered by existing blockchain-based methods and enabling the formation of dynamic and trustworthy expert Quorums.

The main contributions of this paper are as follows:

- We propose FBA-SDN, a novel Federated Byzantine Agreement System approach to trustworthy Collaborative Intrusion Detection in Software-Defined Network edge environments, using blockchain technology, adding accountability and transparency.
- We integrate a dynamic quorum-based topology to mitigate single-point-of-failure vulnerabilities, improving upon the scalability offered by existing blockchain-based approaches and facilitating expert ensemble analysis.
- We employ a distance-based trust algorithm to adjust local trust values for participant nodes, rapidly identifying malicious nodes and modifying quorum and slice structure.
- We further develop a proof-of-concept implementation of our proposed FBA-SDN architecture and evaluate the performance of our proposed approach through simulation experiments, demonstrating its efficacy in reaching rapid and reliable consensus on simulated intrusion data.

The remainder of the paper is structured as follows. Section 2 reviews the related work on blockchain-based CIDS in SDN and IoT environments. Section 3 describes our quorum consensus blockchain-based approach for collaborative intrusion detection in SDN networks. Section 4 presents the implementation and evaluation of our approach using a simulated CIDS network. We discuss the results in Section 5. Finally, The conclusion is presented in Section 6, followed by a brief discussion on future directions.

2. Related Work

The adoption of distributed blockchain technology within IDS, and more broadly IoT and SDN security, has achieved significant research interest in recent years. Fuelled by the continued evolution of the IoT paradigm and improvements in blockchain security and scalability brought about by breakthroughs in distributed consensus, this area of research has produced many promising avenues for consideration, ranging from trust frameworks to smart grids to and energy trading [9], [10].

Blockchain technology exhibits several characteristics desirable within CIDS components, including those outlined in Fig 1. Firstly, it is decentralised and distributed, increasing resiliency during attacks and mitigating the SPoF vulnerabilities present in centralised and hierarchical approaches. Secondly, it is immutable, ensuring records cannot be altered without the remainder of the network detecting malign behaviour. Finally, it is transparent and verifiable, which allows for the secure and transparent storage and verification of security events between nodes. Therefore, several researchers used blockchain technology for IDS in SDN and IoT. Alexopoulos et al. [4] produced seminal

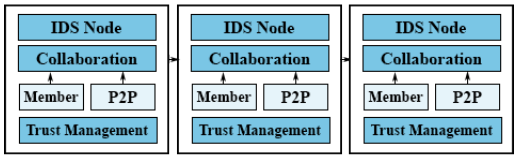

Figure 1. Traditional CIDN component architecture

work in this area, theorising the inclusion of a rudimentary Nakamoto consensus-based approach to signature storage for a distributed Intrusion Detection System. While no implementation was provided, the outlined applicability of blockchain characteristics to secure distributed systems was a noteworthy contribution and spurred ongoing research in this domain.

Putra et al. [11] proposed an approach utilising a hybrid blockchain-IPFS approach for signature communication and storage. Proof of Authority (PoA) is employed as the primary consensus method, with PBFT implemented at layer-2 via a multi-party smart contract-based approach, providing the foundation for a global trust evaluation scheme. However, network-wide contract execution is a complex and computationally heavy process.

ChainGuard, proposed by Steichen et al. [12], aims to secure SDN networks through the use of blockchain technology combined with an OpenFlow firewall to detect malicious activity within the network. While effective, the approach is more IPS focused and protects the blockchain nodes themselves from malicious outsiders.

Several generalised consensus protocol-based voting algorithms have been developed with applicability to this domain. Most notably, Ma et al. proposed a PBFT variant dubbed DWBFT [13], which accounts for dynamic transaction weighting between participants in BFT networks. However, the centralised validation/leader and $\mathbf{O}(n^2)$ complexity PBFT imposes significantly impact applicability to CIDS scenarios. Nonetheless, the consideration of protocol-level behaviour modification rather than execution layer or smart-contract implementations demonstrates a significant step forward in contribution.

While the existing works highlight the benefits of blockchain technology within SDN and CIDS, they exhibit shortcomings which can potentially be resolved via the adoption of novel protocol-level improvements. Increase computational overheads associated with BFT-based protocols, in addition to network-wide contract execution place a significant computational burden on devices, limiting applicability within lightweight deployment scenarios. Therefore, we propose an alternative approach to overcome these limitations.

3. Proposed Framework

This section describes our decentralised CIDS architecture, extrapolating the fundamental components and outlining the threat model. To overcome limitations in the existing body of work, we propose an alternative approach utilising

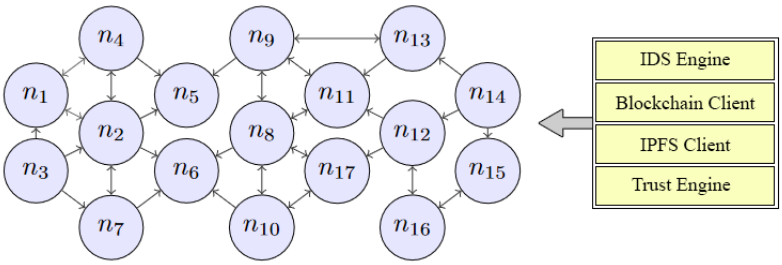

Figure 2. Map of the network showing quorum structure and CIDS services

a FBAS consensus protocol, based on SCP, to maintain the distributed blockchain among IDS nodes. We adopt the quorum structure of SCP, enabling participant nodes to select dynamic validator quorums comprised of trusted neighbour nodes. This achieves three principal functions compared with the existing approaches:

- **Scalability**: We adopt FBAS consensus to improve CIDN scalability, reducing communication overhead between nodes compared with traditional BFT approaches while maintaining availability and finality.
- **Flexibility**: Dynamic configuration of collaboration and priority structure between nodes via quorum slices, enabling real-time optimisation of intrusion evaluation through an ensemble pruning-esque approach.
- **Security**: Blockchain technology ensures the integrity of the committed data and participant nodes, enabling a robust and trustworthy SDN IDS environment.

The framework characteristics are outlined in the following sections.

3.1. FBA-SDN Architecture

We define the system model as $N = \{C, \mathbb{B}, I\}$, where C represents the CIDS network, $\mathbb{B}$ represents the distributed blockchain, and I represents the InterPlanetary File System as in [14]. Each participant node operates both IDS and blockchain validation capacity. The system operates across SDN layers, running on both controllers and dedicated IDS nodes. This deployment architecture is illustrated in Figure 2, with the network architecture outlined in Fig. 3.

For IDS capability, detailed in Section 3.5, we assume each node contains anomaly-based inferencing capacity capable of generating and evaluating alert data. Additionally, to provide the distributed ledger capability detailed in Section 3.3, we assume each node operates a full blockchain node capable of generating and validating transactions under the unified consensus algorithm. To achieve peer-to-peer (P2P) data exchange, we assume the network forms a connected graph with reliable links between nodes. We further deploy a gossip dissemination protocol to enable message exchange between nodes in a multicast.

To complete the system, we deploy an IPFS for the secure, distributed storage of intrusion data. While this data could be stored within the blockchain, the node storage requirement may become untenable over time, particularly in large-scale and architecturally complex scenarios. Therefore, this data is maintained securely outside of the blockchain ledger but remains accessible to all participant nodes.

3.2. Threat Model

Given the system architecture defined above, we can extrapolate a threat model. We assume nodes can exhibit the following behavioural categories in line with the byzantine failure model:

- **Nominal**: Considered to be optimal honest function of an IDS node. In this case, we consider honest function to be genuine and timely evaluation and production of alert data.
- **Byzantine** (Malicious): Intentional anomalous behaviour, actively detrimental to the network function, security and stability.
- **Byzantine** (Crash-fault): Unintentional anomalous behaviour corresponding to system crashes, network dropouts, etc. beyond the control of the participant node.

We define several specific malicious behaviours byzantine nodes can exhibit within CIDS scenarios, exploiting the trust and alert exchange mechanics. Compromised nodes may attempt to increase their own trust value by self-voting, repeatedly submitting identical proposals or decrease the trust of an honest node by intentionally disagreeing with their assessments. Furthermore, malicious nodes may intentionally withhold or delay transaction forwarding across the network, intentionally disrupting the balloting process.

3.3. Blockchain/Consensus Protocol

This section describes the tri-stage SCP-based consensus protocol for exchanging and storing alert data between nodes. These stages provide adequate opportunity for the generation, agreement and settlement of proposals across the CIDS network. The vast majority of communications present $\mathbf{O}(n)$ complexity, though inter-quorum communications can equate to $\mathbf{O}(n^2)$ in worst-case scenarios. We consider unbounded communication time, $t = [0, \infty)$.

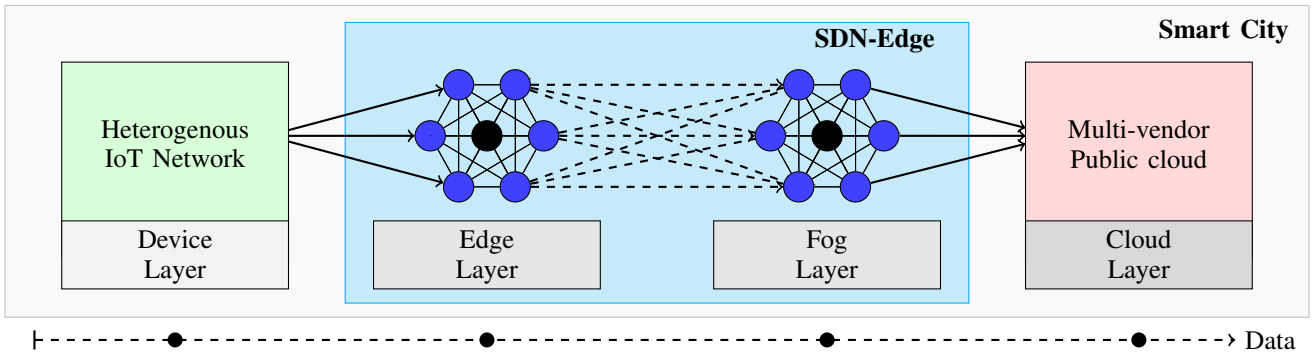

Figure 3. System architecture for an SDN-Edge Smart City deployment, with the system operating in the Edge and Fog layers

1) ***Proposal***: IDS Node proposes a potential anomaly for analysis across the network and proposals follow the structure $prop =< h_{ipfs}, v, \text{signature} >$, where h_{ipfs} represent the IPFS file location hash, v represents the binary classification (anomaly/nominal), and signature represents the signature of the proposing node. Distinct proposals can be generated and submitted asynchronously across the network, representing simultaneous collaborative requests from multiple IDS nodes. We can model the rate of proposal dissemination over time as follows:

$$\frac{dx_i(t)}{dt} = \sum_{j=1}^{N} A_{ij} u_j(t)(p_j(t)v_j(t) - p_i(t)v_i(t)), \tag{1}$$

where N is the number of nodes in the network, A_{ij} is the adjacency matrix representing the connections between nodes, $u_j(t)$ is a control input that determines the rate at which node j sends the proposal, $p_i(t)$ and $p_j(t)$ are binary variables representing if the node i and j have proposed a value respectively. $v_i(t)$ and $v_j(t)$ are binary variables representing the values that has been proposed by node i and j, respectively. A majority value represents the network view of the potential anomaly. This process is visualised in Fig. 4, where a proposal is exchanged between nodes.

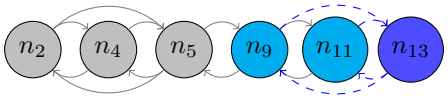

Figure 4. Proposal communication pattern for both intra and inter-quorum.

2) ***Ballot/Slot***: Once a proposal receives a majority of votes, it proceeds to the balloting stage:

$$bal =< h_{ipfs}, v, \text{quorum\_slice}, \text{signature} >,$$

$$\frac{dx_i(t)}{dt} = \sum_{j=1}^{N} A_{ij} u_j(t)(b_j(t)v_j(t) - b_i(t)v_i(t)) \tag{2}$$

where $b_i(t)$ and $b_j(t)$ are binary variables representing if the node i and j have voted on the proposal respectively, and $v_i(t)$ and $v_j(t)$ are binary variables representing the vote cast by node i and j respectively. In the event of inconclusive or stuck ballots, multiple rounds of balloting may be necessary. We can represent the percentage of nodes that have voted 'yes' on the proposal by

$$p_{yes} = \frac{1}{N} \sum_{i=1}^{N} v_i(t) * b_i(t) \tag{3}$$

where $b_i(t)$ is a binary variable indicating whether node i has voted on the proposal, $v_i(t)$ is a binary variable denoting the vote cast by node i.

3) ***Externalise***: Once a majority is confirmed, the value is externalised across the network and permanently stored on the distributed ledger. $ext =< h_{ipfs}, v, \text{signature} >$

$$\frac{dx_i(t)}{dt} = \sum_{j=1}^{N} A_{ij} u_j(t)(e_j(t)v_j(t) - e_i(t)v_i(t)), \tag{4}$$

where N and A_{ij} as defined before, $u_j(t)$ is a control input that determines the rate at which node j sends the externalising message, $e_i(t)$ and $e_j(t)$ are binary variables representing if the node i and j have externally committed to the value respectively, and $v_i(t)$ and $v_j(t)$ are binary variables representing the value that has been externally committed by node i and j respectively. This multi-stage, multi-node evaluation process adopts principles from ensemble and swarm learning algorithms, fusing the opinions of multiple learners into a singular classification. The externalised value is appended to the ledger block for network-wide archival, as

$$\mathbb{B}(t+1) = \mathbb{B}(t) \oplus ext(t) \tag{5}$$

3.3.1. Membership. Accounting for the closed nature of CIDS membership and the permissioned nature of the blockchain approach, we assume each node possesses an Ed25519 (**EdDSA**) generated key pair $\{pk_n, sk_n\}$ for membership verification and proposal signing. Consequently, we assume the existence of *sign*() and *verify*() functions for the purpose of verifiable inter-node communication. For brevity, we do not elaborate on the construction of these functions, as this can be considered standard behaviour within the blockchain domain [8]. Due to the closed nature of the system, we also do not directly consider the impact of Sybil attacks on network performance; however, we discuss a trust algorithm below to disincentivise malign behaviour.

3.3.2. Quorum composition. Given the closed membership detailed above, we define a preset multi-quorum architecture comparable to that displayed in Fig. 2. This structure conforms to the security requirements outlined in Section 3.2, for $\mathbf{Q}_n, \mathbf{Q}_m$ to intersect securely, it is necessary that

$$\forall\ \mathbf{Q}_n, \mathbf{Q}_m \subseteq N, (\mathbf{Q}_n \cap \mathbf{Q}_m \neq \emptyset) \wedge (|\mathbf{Q}_n \cap \mathbf{Q}_m| \geq 2) \quad (6)$$

By ensuring quorums intersect with at least 2 nodes, we mitigate the impact of an intersecting node exhibiting byzantine behaviour and causing disjoint quorums, ensuring system integrity. We can simplify the slices S_i within $\mathbf{Q}_n$ as follows:

$$\mathbf{Q}_n = \bigcup_{i=1}^{m} \mathbf{S}_i \quad (7)$$

where $\mathbf{S}_i$ is a slice and

$$|\mathbf{S}_i \cap \mathbf{Q}_n| \geq 2, \quad (8)$$

ensuring only valid slices are considered, although this simplification does not consider all nuances of quorum structure. We allow nodes to adjust quorum slice composition during deployment based on node trust characteristics, ultimately impacting quorum structure.

3.4. IPFS

To provide decentralised alert data storage for the CIDS we employ InterPlanetary File System (IPFS), avoiding on-chain storage becoming cumbersome over time [14]. These two technologies can operate harmoniously to improve scalability while maintaining the system security requirements. Nodes subscribe to the IPFS storage using the same identifier as their blockchain address, ensuring access is restricted to eligible parties.

Following the generation of a collaboration proposal, the originating node outputs a file containing the alert data $D = (d_1, d_2, \ldots, d_n)$ and corresponding metadata $M = (m_1, m_2, \ldots, m_n)$ to provide additional analysis context, where M_i contains fields for timestamp, filesize and a probability vector representing the output of inferencing. Prior to proposal generation, the hash for a given combination is generated to facilitate system-wide retrieval, where $hash_{ipfs} = hash(D_i, M_i)$. The hash is included in the proposal for subsequent retrieval and analysis.

3.5. Intrusion Detection System

In addition to independent analysis, the framework facilitates the collaborative analysis of low-confidence alert data, with the intention of reaching a more accurate global opinion. This principle is adopted directly from ensemble fusion, whereby multiple weaker classifiers are combined to achieve improved accuracy. As this paper is focused on the protocol side, we do not elaborate on the ML approaches; however, we assume each node is capable of generating a normalised class probability vector p from input data x, achieved through a SoftMax output layer, where

$$p = \text{softmax}(x) = \frac{x}{\sum_{j=1}^{n} x_j}. \quad (9)$$

Inferencing can occur via two ingestion avenues: Nodes perform inferencing on local traffic and generate corresponding alerts, and during the proposal stage, collaborating nodes ingest remote data from the IPFS to evaluate the nature of the alert and generate their own evaluation.

3.6. Trust framework

A critical component of any distributed CIDN is a decentralised trust framework to disincentivize and punish malign behaviour. Given the behaviours outlined in Section 3.2, we can readily identify potentially malicious participants via challenge-based collaboration, a concept well explored in the literature. To this end, we employ a distance-based evaluation comparison to identify potentially malicious behaviour.

3.6.1. Distance. For a given set of nodes within a quorum slice S_i which generate class probability vectors p, as outlined in Section 3.5. For each vector pair p_i and p_j, where $i \neq j$, the Euclidean distance is calculated:

$$d(p_i, p_j) = \sqrt{\sum_{k=1}^{n} (p_{i,k} - p_{j,k})^2}, \quad (10)$$

where the distance exceeds the acceptable distance threshold $d > \alpha$, nodes receive a fixed local trust score penalty to their trust value T. Conversely, distances falling below a minimum result in a trust bonus applied to the node. A forgetting factor λ is implemented to dynamically modify the influence of events over time, which can be expressed as:

$$T(t+1) = \begin{cases} \lambda(T(t) + \delta_{gain}) & \text{if } d(p_i, p_j) \leq \alpha, \\ \lambda(T(t) - \delta_{loss}) & \text{if } d(p_i, p_j) > \alpha. \end{cases} \quad (11)$$

3.6.2. Quorum removal. Assuming a node's reliability $R(n)$ falls below the trust threshold T, we can remove the node from the quorum provided the intersection of quorum $\mathbf{Q}_n$ with any other quorum $\mathbf{Q}_m$ is not compromised by the removal of n_i, as shown in (12).

In the event of null intersection, the removing node should establish a relationship with a replacement node from the alternative quorum(s) prior to removal to avoid disjoint quorums.

$$\forall\, \mathbf{Q}_n \in \mathbf{Q}, \forall\, n_i \in \mathbf{Q}_n \Rightarrow \left(\bigcap_{\mathbf{Q}_m \in \mathbf{Q}, \mathbf{Q}_m \neq \mathbf{Q}_n} (\mathbf{Q}_n \cap \mathbf{Q}_m) \neq \emptyset \right) \wedge \left(\bigcap_{\mathbf{Q}_m \in \mathbf{Q}, \mathbf{Q}_m \neq \mathbf{Q}_n} (\mathbf{Q}_n \setminus \{n_i\} \cap \mathbf{Q}_m) \neq \emptyset \right) \wedge \left(\left| \{v \in \left(\bigcap_{\mathbf{Q}_m \in \mathbf{Q}, \mathbf{Q}_m \neq \mathbf{Q}_n} (\mathbf{Q}_n \setminus \{n_i\} \cap \mathbf{Q}_m) \right) | R(n) \geq T\} \right| \geq 2 \right) \tag{12}$$

4. Implementation and Analysis

For our preliminary experimentation, we make several apriori assumptions concerning the configuration of the underlying IDS network. We assume each contributing node can output a class probability vector. To represent the class probability vector for this experimentation, we instead opt to pseudo-randomly generate vectors for each node – this consideration provides additional experimentation flexibility to evaluate the distance and trust relationship. This vector can be spiked prior to transmission to simulate nodes deviating from the acceptable range, mimicking the behaviour of a malicious node. Additionally, we assume each node is connected via a reliable link (or series of redundant links) and can therefore reach probabilistic consensus.

Implementation and experimentation were conducted using Python version 3.11.0 on a Windows 10 Desktop PC running an Intel Core i9 9900ks @ 4.00GHz. Performance benchmarking also took place on a 24-node Raspberry Pi 4B research cluster.

4.1. Consensus

We evaluated the time taken to reach consensus on a given proposal value across a network, considering various node configurations and proposal frequencies. The proposal represents a binary classification of alert data, with nodes individually evaluating and classifying each item. The aim was to assess the efficacy of a proposed approach for consensus-building. A network with a quorum structure similar to Fig.2 was simulated to achieve this. The simulation considers all three consensus phases. The time required to reach consensus with each process was measured and compared.

Figure 5 compares the latency in each of the protocol phases as the network nodes increase, whereas Fig 6 shows the latency as we increase the number of transactions. Both results indicate that the proposed method is efficient in reaching consensus, especially as the network complexity increases as we increase nodes or the number of transactions. The latency increase observed becomes more apparent with an increase in the number of nodes, highlighting the viability of the approach.

In Fig. 7, FBA-SDN is compared with two alternative consensus protocols from [13] to evaluate its performance in small scale networks. The results highlight the validity of our SCP-based approach and demonstrate lower latency across all network sizes.

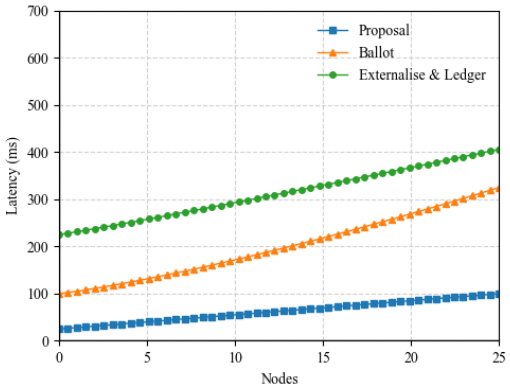

Figure 5. Latency as the number of IDS nodes increases

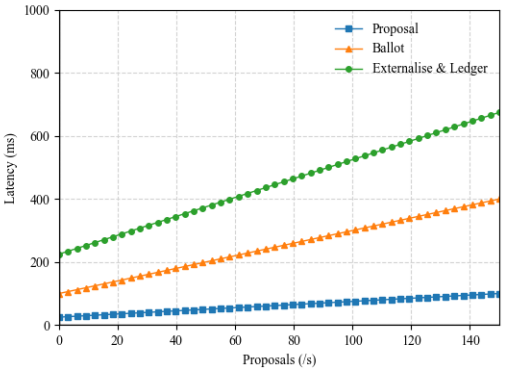

Figure 6. Latency as the number of proposals increases

4.2. Trust

Figure 8 shows the trust exhaustion time for a malicious node by considering the frequency and significance of the malign behaviour. To achieve this, we evaluated the behaviour of nodes over a period of 100 epochs to understand the impact of various levels of honesty. We analysed the evolution of trust for three types of nodes: an honest node, an eventually malicious node, and a fully malicious node. The objective was to understand the effect of malicious behaviour on the trust score and how it impacts the stability and reliability of a network.

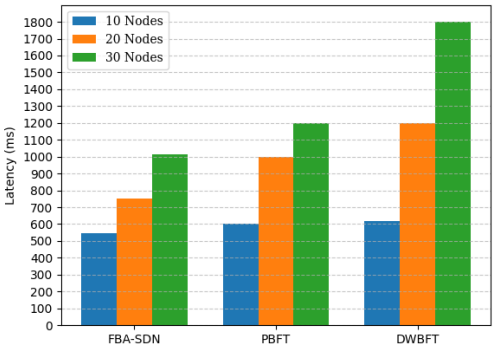

Figure 7. Latency comparison to alternative consensus protocols from [13]

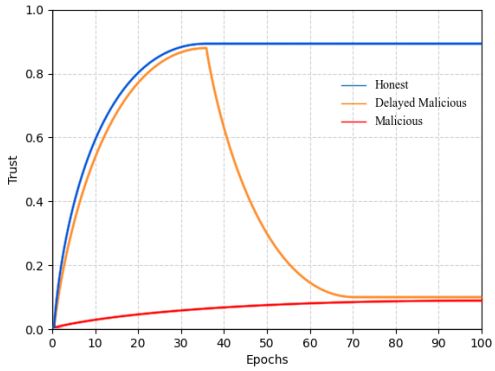

Figure 8. Trust score value over time

This experiment demonstrates the trust score of the honest node steadily increases towards 1, whereas the selectively malicious node's trust value initially rises, then rapidly decreases to the baseline value with the fully malicious node once it exhibits byzantine behaviour. These preliminary results demonstrate the viability of FBAS-based approaches for the management and operation of security-centric applications such as CIDS.

5. Conclusion

In this paper, we presented a preliminary look at a quorum consensus blockchain-based approach for collaborative intrusion detection in software-defined networks. Our approach uses a decentralised and distributed blockchain platform to store and verify security events and a quorum consensus algorithm to ensure the efficient and reliable operation of the IDS. Overall, our quorum consensus blockchain-based approach for collaborative intrusion detection represents a promising and innovative solution for improving the security posture of SDN networks. We believe that our approach has the potential to significantly enhance the security of SDN networks and contribute to the development of more secure and resilient networking systems.

In future work, we intend to explore the viability of deploying lightweight quantized anomaly detection models to participant nodes in combination with our FBA-based consensus process, implementing a scalable IoT CIDS testbed environment. Furthermore, the transparent and dynamic nature of quorum structure and alert evaluation present potential research avenues for CIDS optimisation by adopting ensemble pruning principles.

References

[1] S. Tomovic, M. Pejanovic-Djurisic, and I. Radusinovic, "SDN based mobile networks: Concepts and benefits," *Wireless Personal Communications*, vol. 78, pp. 1629–1644, 2014.

[2] K. Scarfone, P. Mell *et al.*, "Guide to intrusion detection and prevention systems (IDPS)," *NIST special publication*, vol. 800, no. 2007, p. 94, 2007.

[3] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1–33, 2015.

[4] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," in *Critical Information Infrastructures Security: 12th International Conference, CRITIS 2017, Lucca, Italy, October 8-13, 2017, Revised Selected Papers 12*. Springer, 2018, pp. 107–118.

[5] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Towards scalable and trustworthy decentralized collaborative intrusion detection system for iot," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2020, pp. 256–257.

[6] J. Yoo, Y. Jung, D. Shin, M. Bae, and E. Jee, "Formal modeling and verification of a federated byzantine agreement algorithm for blockchain platforms," in *2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. IEEE, 2019, pp. 11–21.

[7] M. Florian, S. Henningsen, C. Ndolo, and B. Scheuermann, "The sum of its parts: Analysis of federated byzantine agreement systems," *Distributed Computing*, vol. 35, no. 5, pp. 399–417, 2022.

[8] D. Mazieres, "The stellar consensus protocol: A federated model for internet-level consensus," *Stellar Development Foundation*, vol. 32, pp. 1–45, 2015.

[9] A. Jindal, G. S. Aujla, N. Kumar, and M. Villari, "GUARDIAN: blockchain-based secure demand response management in smart grid system," *IEEE transactions on services computing*, vol. 13, no. 4, pp. 613–624, 2019.

[10] A. Jindal, G. S. Aujla, and N. Kumar, "Survivor: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment," *Computer Networks*, vol. 153, pp. 36–48, 2019.

[11] G. D. Putra, V. Dedeoglu, A. Pathak, S. S. Kanhere, and R. Jurdak, "Decentralised trustworthy collaborative intrusion detection system for IoT," in *2021 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2021, pp. 306–313.

[12] M. Steichen, S. Hommes, and R. State, "ChainGuard—A firewall for blockchain applications using SDN with openflow," in *2017 Principles, Systems and Applications of IP Telecommunications (IPTComm)*. IEEE, 2017, pp. 1–8.

[13] C. Ma, Y. Zhang, B. Fang, and H. Zhang, "DWBFT: a weighted byzantine fault tolerant protocol with decentralized trust," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 5384–5390.

[14] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.