Performance Evaluation of a Blockchain-based Content Distribution over Wireless Mesh Networks

Yoshihiro Ohba
Kioxia Corporation
Minato-ku, Tokyo, Japan
Yoshihiro.Ohba@kioxia.com

Jing Yi Koh
School of Computing Science
University of Glasgow, UK
2427234K@student.gla.ac.uk

Nigel Ng
School of Computing Science
University of Glasgow, UK
2427257N@student.gla.ac.uk

Sye Loong Keoh
School of Computing Science
University of Glasgow, UK
SyeLoong.Keoh@glasgow.ac.uk

***Abstract*—This paper studies the performance of a proximity content distribution scheme over IEEE 802.11s mesh networks with reasonable user density among combinations of three network configurations and two transport mechanisms. For content access control, Hyperledger Sawtooth Blockchain with PoET (Proof of Elapsed Time) consensus algorithm is used as a decentralised storage of non-repudiated and rapid transactions for granting content access and distributing the content decryption key. An extensive performance evaluation of the content distribution and content access control protocols using ns-3 simulator was conducted. The results show that the integration of Blockchain and UDP multicast content distribution in a hybrid mesh network topology is highly feasible.**

I. Introduction

The current Internet infrastructure is almost exclusively using wired networking as the backbone for foundational routing. It is built over stable networks, where most of the nodes are stationary terminals interconnected over wired networks. In the near-future 5G and wireless environments, this may no longer be the case as improvement in networking technologies can potentially enable wireless connectivity to be used in the backend. This provides great flexibility in establishing network connectivity at anytime and anywhere without much prior planning and implementation. For example, proximity content sharing is considered as one of the key use cases where high-quality and high-volume contents are generated by mobile users in an event field, such as a sports complex or theme park, and available for a short duration (e.g., instagram story). These video streams are then distributed to be shared among the mobile users in the proximity of each other.

Rapid establishment of wireless network connectivity is required to enable proximity-based content sharing and currently there is no systematic approach to achieve this. Firstly, there is a need to investigate the feasibility of using IEEE 802.11s wireless mesh networking [1] and its variants as the backbone for low latency, high availability and reliable content sharing. Secondly, as the content generated are being shared with other users in the vicinity, a decentralised content storage and access architecture must be integrated, without needing to rely on cloud storage and infrastructure. Thirdly, content security must be guaranteed in that it must be encrypted by default and access is granted by the content owner. Bluetooth mesh and other low-power mesh technologies are not considered for proximity content sharing which requires higher throughput.

In this paper, we present an extensive performance evaluation of proximity-based content distribution and content access control over 802.11s mesh networks using ns-3 [2] network simulator, providing a recommendation of the most suitable network configuration and peer-to-peer (P2P) content distribution mechanisms. We considered three network configurations: Basic Service Set (BSS) and Mesh BSS (MBSS) as defined in IEEE 802.11s, as well as *hybrid mesh* which is an interconnection of multiple BSS networks via an MBSS network. Unlike the commonly used hybrid mesh network in which multiple BSS networks and their connected MBSS network are part of the same IP subnet, each of the BSS networks and the MBSS network in our hybrid mesh network forms a distinct IP subnet for multicast traffic segregation. This specific hybrid mesh network topology has not been extensively studied in the literature. As for P2P content distribution mechanisms, we compared unicast-based distribution over TCP and multicast-based distribution over UDP with selective retransmission. We show that the latter is better for all network configurations with a larger number of recipients. The P2P content distribution is integrated with a Blockchain-based content access control for distributing encrypted content decryption key to content recipients. This effectively eliminates the need for a cloud service for decryption key distribution. We used existing Hyperledger Sawtooth open-source platform with PoET (Proof of Elapsed Time) consensus algorithm in our performance evaluation.

The main contributions of this paper are: (1) Design of a robust and scalable hybrid mesh network for proximity-based wireless content distribution with Blockchain-based content access control mechanism and (2) Performance evaluation of network scalability and reliability, Blockchain's PoET consensus and transactions rates, showing the effectiveness of the designed network.

This paper is organised as follows: Section II discusses the background and its related work. Section III presents the proposed network architecture for proximity-based content distribution. Section IV shows the implementation of the proposed content distribution protocol and content access control protocol. Section V provides the performance evaluation of the content distribution protocol, while Section VI provides performance evaluation of the content access control protocol. We conclude the paper with future works in Section VII.

II. Background and Related Work

Related work is categorized into three groups: mesh networking, P2P content distribution, and Blockchain. As for mesh networking, the IEEE 802.11s mesh network [1] is implemented in OSI layer 2, as compared to traditional wireless mesh network in layer 3, the IP layer. Each network node is aware of its radio environment, thus making measurements to several metrics more effective. *Carrano et al.* suggested that a flat mesh with more than 32 nodes exhibits degradation of network performance and efficiency [3]. One of our contributions to 802.11s is to show that the use of UDP multicast over 802.11s can extend the applicability of flat mesh with up to 49 nodes.

As for P2P content distribution, *Lai et al.* [4] evaluated a multicast-based P2P content distribution scheme over 802.11 which works over a dynamically configured set of 802.11 BSS's (Basic Service Sets) in which content is multicast by the Access Point (AP) of a BSS to its non-AP stations (STA) each of which in turn changes its role from STA to AP (known as "role change") to form a new BSS and start redistribution of the content under the new BSS. The UDP multicast mode of our content distribution protocol is based on the scheme in [4], but we adopted the multihop frame forwarding feature of 802.11s and hence it does not require role change between AP and STA to distribute the content across multiple wireless links, eliminating the complexity and possible use case limitations incurred by role change.

In [5], *Ortega et al.* proposed the use of Content-Centric Networking (CCN) instead of TCP/IP for vehicular networks, incorporating Blockchain to provide source reliability, integrity and validity of the information exchanged. CCN is a peer-to-peer architecture with reduced congestion and latency, as nodes communicate based on the data type, the content, and identity of the nodes instead of network addresses. CCN assumes that 5G slicing is used to enable parallel networks to carry vehicular network traffic. Conversely, the proximity-based content distribution architecture introduced in our paper does not require to use a 5G core network at all.

InterPlanetary File System (IPFS)[6] is a unicast-based P2P file sharing system based on DHT (Distributed Hash Table) where a hash of a content is associated with a set of nodes storing the content. Each IPFS node that wishes to retrieve the content will communicate with one of the set of nodes. IPFS does not support multicast-based file sharing.

As for Blockchain, Blockchain of Things (BCoT) [7] is the convergence between Blockchain and Internet-of-Things (IoT). Blockchain can complement IoT systems to enhance the interoperability, security, reliability and scalability of IoT systems deployed in many application domains such as supply-chain management, healthcare, etc. and thus, paving the way for new business models and novel distributed applications [8].

In Proof-of-Work (PoW), an attack on the Blockchain to alter data requires 51% control of assets, while for PoET, (2/3)+1 control of nodes are required due to Byzantine Fault Tolerance (BFT). BFT is faster and cheaper, it provides actual finality over PoW's practical finality. It allows the nodes to choose who to trust rather than who has the most computation power or money. In our paper, it is shown that PoET consensus algorithm is fairly robust when deployed in a wireless network.

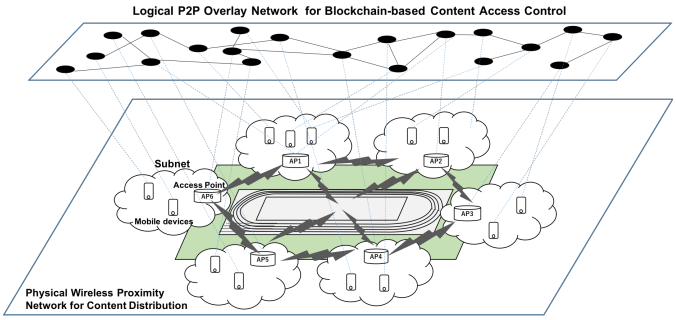

Fig. 1: Architecture of Proximity-based Content Distribution Network.

Deploying permissioned blockchain with Community Mesh Networks was proposed in [9]. They deployed Hyperledger Fabric (HLF) on *Guifi.net*, and demonstrated the use of Blockchain to track the contributions of nodes, links and consumption of communication network's resources. There is a study related to Blockchain-based access control [10]. However, there is no existing literature that studies the performance of a Blockchain-based access control protocol used for proximity-based content distribution.

Blockchain storage performance using Non-Volatile Memory express Over Fabrics (NVMe-oF) is evaluated in [11] using KUMOSCALE^{TM} [12], where suitability of NVMe-oF SSD storage for Blockchain is shown. Possible applicability of NVMe-oF for a large scale Blockchain network simulation is indicated in Section VII.

III. Proximity-based Content Distribution Architecture

Figure 1 shows the proposed network architecture of proximity-based content distribution. The proximity network is typically formed where a large number of people gather to join or participate in an event. The network consists of nodes serving as IEEE 802.11 Access Points (APs) that are interconnected with each other using IEEE 802.11s wireless mesh protocol. Each AP forms a distinct Basic Service Set (BSS) connecting 802.11 wireless devices or non-AP stations (STAs). Additionally, the AP is a member of the same 802.11s Mesh BSS (MBSS). Contents are generated by a node in the proximity network and then distributed to all other nodes using a P2P content distribution protocol as described in Section III-A. On top of the proximity network, a logical P2P overlay network is formed to facilitate content access control (c.f. Section III-B). In this paper, a content represents a single file containing content data encrypted by the content owner, and we assume that the content owner's wireless device is the content source.

A. P2P Content Distribution Protocol

The P2P Content Distribution (CD) protocol is adapted from [4] with an extended functionality to fully utilise the multi-hop frame forwarding feature of 802.11s. Our CD protocol has two phases, namely *Content Advertisement* and *Content Transfer*.

In the *Content Advertisement* phase, the content source periodically multicasts a short UDP advertisement (ADV) message containing the metadata of the content over its connected wireless network. The metadata consists of *content ID*, *content size* and *content hash value*. Each receiver of the ADV message stores the pair of the metadata and the source IP address of the ADV message. This is followed by the *Content Transfer* phase in which the content file is transferred to the receivers of the ADV message, or content receivers. Two transmission modes are defined for *Content Transfer* phase.

1) TCP Mode: The content file is transmitted over TCP to the content receivers upon initiation of a TCP connection to the content source. With this, reliable message delivery is guaranteed by TCP. However, when a TCP connection for a content receiver is lost, a new TCP connection is re-initiated by the content receiver and the content source will re-send the entire content file over the new TCP connection.

2) UDP Multicast: The content source splits the content into multiple fixed-size chunks and multicasts the chunks to all content receivers on the same IP subnet with an inter-chunk transmission interval d (secs). Each chunk also carries a *content ID* and a *chunk number* used for re-assembling the content and *retransmission* of lost chunks. The chunk retransmission is based on negative acknowledgement (NACK). It contains a list of outstanding chunk numbers, allowing the content source or any other receiver to retransmit (by multicast) the missing chunks in the network only if it has not seen the chunk retransmitted by any other node.

In order to deal with large-sized wireless networks with a large number of content receivers, we define a "**relay**" function by which a content receiver that has received the full content, in turn starts acting in the same way as the content source. Essentially, the relay function can improve the reliability of content delivery by splitting the end-to-end path between the content source and the content receiver into multiple path segments each of which consists of smaller number of hops. The relay function is defined for both transmission modes. We define a configuration parameter c for the relay function such that relaying is disabled if $c = 0$ and enabled if $c > 0$. In TCP mode with $c > 0$, the content source and relaying nodes accept at most c TCP connections at the same time and receivers choose the relay node with the minimum ICMP Ping response time. On the other hand, for UDP multicast mode with $c > 0$, a content receiver with the full content becomes a relay node if there are fewer than c nodes advertising ADV messages for the content.

As the network architecture must be scalable, it is important that a source node is able to distribute its content across IP subnets. We have also defined "**forward**" function in which a gateway node or a relaying node that is connected to multiple IP subnets, can re-distribute the content received over one IP subnet to other IP subnet(s).

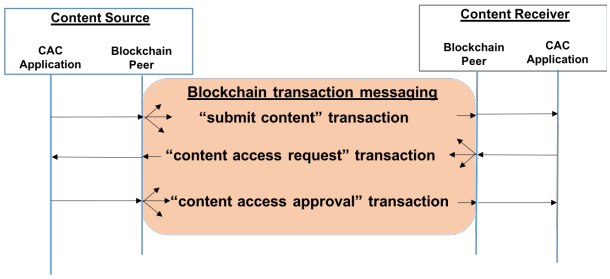

Fig. 2: Content Access Control (CAC) Protocol

B. Content Access Control Protocol using Blockchain

The content being distributed is encrypted by the content source. We use a fully decentralised Content Access Control (CAC) protocol for realising content access authorization without dedicated infrastructure to store the content and its decryption keys, so that the authorisation can be done dynamically on the fly. The CAC is based on a sequence of Blockchain transactions used for securely distributing content decryption keys from a content source to receivers as well as to record all content access transaction requests in a distributed ledger. As the validated blocks are replicated to all the Blockchain peers in the network, receivers are able to retrieve the content key to decrypt the content without relying on any cloud storage infrastructure. Figure 2 illustrates the CAC protocol, which is defined as follows:

- First, the content source generates a *submit content* transaction for the content.
- Each content receiver wishing to view the content generates a *content access request* transaction containing its public key, when receiving the *submit content* transaction.
- In response to the *content access request* transaction, the content source generates a *content access approval* transaction containing the content decryption key encrypted with the public key of the content receiver.
- The content receiver decrypts the *content access approval* transaction using its private key, and then uses the key to decrypt the content data.

The CAC protocol can be executed independently of the CD protocol, i.e., a CAC sequence for a given content to obtain the decryption key can happen before, after or during the content distribution process. For stability of Blockchain operations over the *hybrid mesh* network, we assume that Blockchain peers on stationary devices (such as 802.11 Access Points) are always operating. We also assume that the proximity-based content distribution network is transient by nature (e.g., re-initialized a day), the current chain is to be moved to an off-chain back-up storage before the re-initialization of the proximity network.

IV. Implementation

The CD and CAC protocols were implemented as Python scripts running inside Docker containers where each Docker container represented a distinct node in the proximity content distribution network and had a distinct TCP/IP stack with a distinct routable IP address assigned. Each Docker container also ran Hyperledger Sawtooth [13] programs.

As for the layer 2, ns-3 [2] was used to simulate IEEE 802.11 and 802.11s MAC protocols. ns-3 TapBridge module which enables bridging of IP packets between Linux host and an ns-3 net device via a Linux tap interface was used. In order to realise inter-networking between ns-3 and docker containers, we used *ConfigureLocal mode* of TapBridge and a Docker network driver called hostnic[1], which allows a tap interface created by ns-3 to be used as a network interface of the Docker container.

The ns-3 simulation and all Docker containers ran on a single server machine that has Intel Xeon CPU (E5-2699 v4 @ 2.20GHz, 44 virtual CPU cores x 2 sockets), 256GB RAM and 8TB NVMe SSD. We used serial simulation mode of ns-3 to allocate sufficient computing resources to the processes running on the Docker containers over the single machine.

A. Network Topologies

Using Docker containers and ns-3, three network topologies were configured: namely *BSS*, *MBSS* and *Hybrid Mesh* to simulate and evaluate their performance for content distribution.

1) BSS: A typical 802.11 WiFi network that allows for multiple STAs to be connected to an AP. The AP is stationary, serving as the network backbone to ease the bootstrapping process and to handle the core routing in the network.

2) Mesh BSS (MBSS): It uses the 802.11s layer 2 protocol, allowing multiple Mesh Points (MP) to form a single mesh network and each contributing to routing using the Hybrid Wireless Mesh Protocol (HWMP). MBSS can be bootstrapped easily from scratch, not requiring any static APs to be deployed. All nodes subscribe to the same *mesh ID*. All routing between nodes in MBSS is handled at layer 2 by 802.11s, transparent to the IP and layers above. This includes determining the position of mesh points relative to each other and the most optimal paths between them. If paths are stable and well established, high network throughput is possible.

3) Hybrid Mesh: Figure 3 shows the most scalable network topology using MBSS as the core backbone and each node with an MP function also serves as an AP, forming a BSS at the edge. The main goal of hybrid mesh is to equip the last mile communication with the ability to serve hundreds or thousands of nodes using WiFi infrastructure protocols, which have been proven to be efficient and scalable to large number of nodes.

B. Integrated CAC with Hybrid Mesh Network

We have integrated the CAC protocol with the CD protocol on hybrid mesh such that each STA and AP also serves as a Blockchain peer. As shown in Figure 3, each Blockchain peer (a.k.a *validator* V) is responsible for handling transactions, processing and verifying the blocks using PoET consensus algorithm. Each node having a validator component also has the following components: a client, a REST API, a PoET consensus engine and transaction processors. The client is an application program that provides an interface for submission of content and request of content decryption key in the form of transactions. Batches of transactions created by the client are sent to the REST API, which is a Hyperledger Sawtooth component for submitting the batches to the Blockchain P2P network through the validator. Transaction processors process transactions received by the validator.

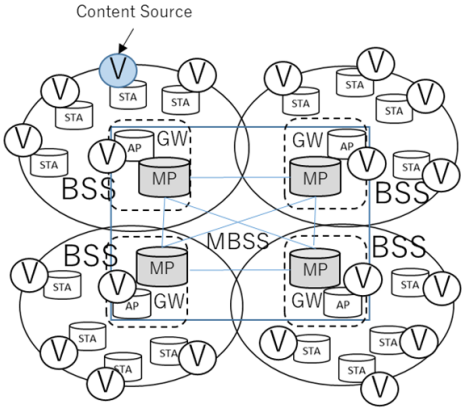

Fig. 3: Hybrid Mesh with CAC Protocol using Blockchain

We used three types of transaction processors, namely, *settings-tp*, *poet-validator-registry-tp* and *cac-tp* where the first two types are provided by Hyperledger Sawtooth. The *settings-tp* is responsible for configuring the validators. The *poet-validator-registry-tp* is used for the PoET sign-up phase. *cac-tp* is our CAC protocol implementation using Smart Contract.

For measuring the Blockchain statistics, *InfluxDB* was used for storing metrics or the time series of statistical data generated by Hyperledger Sawtooth. The performance metrics are visualised using *Grafana*.

V. Performance Evaluation: CD Protocol

We evaluated the performance of the proposed CD protocol in the following three network configurations.

- *BSS* configuration — A single BSS has n_{sta}(=10,20) nodes (including AP) randomly placed within a circle of 160 meter diameter with the AP located at the centre.
- *MBSS* configuration — A single MBSS has n_{mp} MPs placed on a grid of $n_{mp} = n_{rows} \times n_{cols}$ crosspoints with $w = 150$ (m) distance between neighboring crosspoints where $n_{rows} = n_{cols}$ = 2,...,7 (i.e., n_{mp} = 4,...,49).
- *Hybrid Mesh* configuration — A single MBSS configured in the same way as *MBSS* configuration except $n_{rows} = 2$ and $n_{cols} = 2, 3$ (i.e., n_{mp} = 4,6) and n_{mp} BSSes configured in the same way as *BSS* configuration, with each MP paired with a distinct AP, forming a gateway

[1]https://github.com/yunify/docker-plugin-hostnic

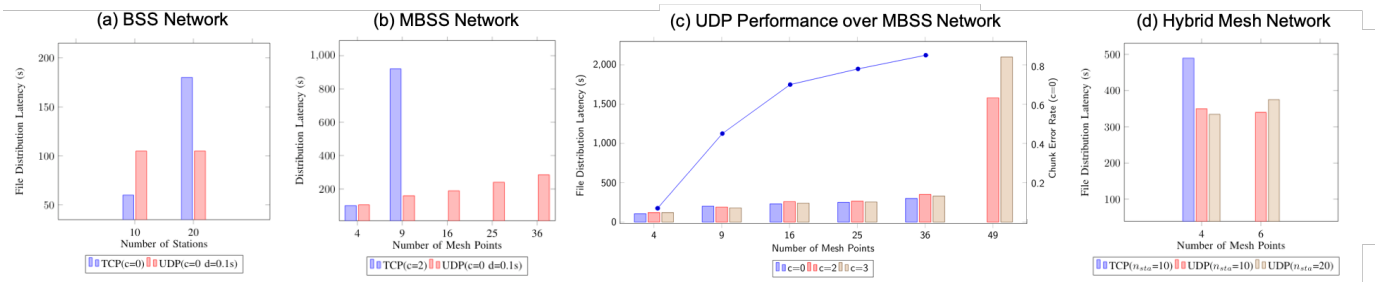

Fig. 4: File Distribution Latency over Various Network Topologies (a) BSS (b) MBSS (c) UDP over MBSS (d) Hybrid Mesh.

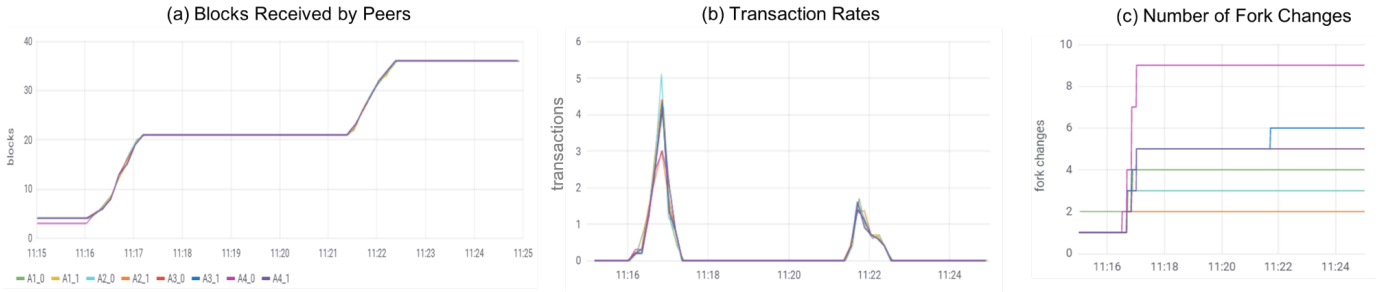

Fig. 5: (a) Number of Blocks Received over Time (b) Transaction Rates (c) Fork Changes over Time.

node. The total node count n is given as $n = n_{rows} \times n_{cols} \times n_{sta}$ (counting a gateway node as one node).

The content source which is an STA located in a given BSS distributes a single content file of size 1MB. In UDP multicast mode, the simulation used a chunk size of 1KB and set the inter-chunk transmission interval $d = 0.1$s. For both TCP and UDP multicast modes, several concurrency parameter c values were simulated and examined.

Figure 4(a) illustrates the performance of content distribution protocol using TCP and UDP multicast for BSS topology. It is observed that TCP's file distribution latency increases with n_{sta} in BSS topology, while it is relatively consistent for UDP multicast in that the file distribution latency does not change with the increase of n_{sta}.

Figure 4(b) shows the performance of file distribution latency using TCP and UDP multicast for MBSS topology. Similarly, UDP multicast outperformed TCP (with c=2) with a much smaller file distribution latency, except for $n_{mp} = 4$, where both TCP and UDP multicast have comparable performance. It is observed that the latency for TCP mode is extremely high when the number of mesh points increased. This has resulted in the inability of all mesh points to complete file distribution within a reasonable time for $n_{mp} > 9$.

Figure 4(c) shows the performance of UDP multicast using MBSS topology with different number of relays c. Although the file distribution latency and chunk error rate increase with n_{mp}, UDP multicast can tolerable up to $n_{mp} = 49$ when the relay function (i.e., $c > 0$) was used. It appears that MBSS exibits some limitations in terms of scalability and may not be suitable to support a large number of nodes on its own.

Figure 4(d) shows the performance of TCP and UDP multicast for content distribution in *Hybrid Mesh* topology. It is clear that UDP multicast outperformed the TCP mode. Unlike UDP, file distribution using TCP could not be completed with the backend MBSS configuration of $n_{mp} = 6$. Based on the results of MBSS and BSS, it can be inferred that the *Hybrid Mesh* topology can potentially be scaled up to 36 to 49 MPs serving as the backend network, while each MP forms a BSS with other nodes (e.g., up to 980 nodes can be accommodated when $n_{mp} = 49$ and $n_{sta} = 20$). For more than 1000 nodes, use of an extended hybrid mesh network with interconnecting multiple MBSS'es can be considered.

VI. Performance Evaluation: CAC Protocol

The CAC protocol was evaluated over *Hybrid Mesh* topology with $n_{col} = n_{rows} = 2$, $n_{sta} = 5$, where each BSS had two Blockchain peers out of five nodes (i.e., 8 Blockchain peers in the entire network). The content source (which was an STA located in a given BSS) distributes a content file of size 1MB using UDP multicast mode with $d = 0.1$s, $c = 0$ and the chunk size of 1KB.

The simulations were conducted with the following steps: In *Step 1*, the CAC protocol was executed to obtain the decryption key for the first content. At this time, it was assumed that the first content had already been distributed and thus there was no content distribution traffic in the network. In this way, we could measure the Blockchain performance of the CAC protocol without any other interference. In *Step 2*, the CAC protocol was executed again to obtain the decryption key of second content. Concurrently, the CD protocol was executed to distribute the third content in the *Hybrid Mesh* network. Note that both steps generated the same number of CAC transactions.

As shown in Figure 5(a), all blocks were successfully received by all Blockchain peers during the simulation run. This indicates that the *Hybrid Mesh* is robust and stable, despite both CD and CAC protocols were executed concurrently.

Figure 5(b) shows the Blockchain Transaction Execution Rate (TER) at each peer. All nodes were run as both the content receiver and the Blockchain peer, the TER was higher in *Step 1* of the simulation as the nodes' resources were dedicated fully to validate Blockchain transactions. Conversely, the TER was significantly affected when the CD protocol was executed concurrently in *Step 2* though the reliability of both the CAC and the CD protocols were not affected.

Figure 5(c) shows the number of fork changes (i.e., the number of changing from one branch to another in the blockchain when adding a new block) at each Blockchain peer. In the simulation, the CAC protocol in *Step 1* started at *11:16*. This was followed by the CD protocol and the CAC protocol in *Step 2* which started at *11:18* and *11:21* respectively. It can be seen that the number of fork changes during the execution of *Step 2* protocols (i.e., CD and CAC) was small, indicating that the PoET consensus algorithm is robust under highly loaded conditions. The reason for seeing more frequent fork changes during *Step 1* than *Step 2* is because the Z-test used by PoET for ensuring stability of consensus decisions requires all Blockchain peers to receive a certain number of blocks to be effective.

VII. Conclusions

This paper has presented an extensive study on the design of a new network architecture that is robust and scalable for proximity-based content distribution network with decentralised storage and access control mechanism. We have made three contributions: First, *hybrid mesh* is the best suited topology for establishing a dynamic wireless ad-hoc network. We concluded that a pure MBSS network would not be scalable as the underlying HWMP routing is unable to settle the routing paths when there are more than 49 MPs in the network. Second, a fully decentralised storage and access control protocol based on Blockchain can work in an isolated wireless environment with 802.11s mesh. This means that the content distribution network can be protected using Blockchain without relying on any dedicated infrastructure (within the network or in the cloud) to manage the content decryption keys. Third, we also concluded that UDP multicast (without congestion control) for content distribution has resulted in congestion in the hybrid mesh network. Consequently, this has affected the performance of Blockchain, causing latency for distributing transactions and blocks and reduction of TER.

One enhancement that can be made is to adopt a separate communication network for Blockchain processing for transactions and blocks, e.g., the content distribution protocol is deployed on the hybrid mesh network, while the Blockchain network communicates through the 4G or 5G cellular network.

Finally, we have identified that use of distributed ns-3 simulation using multiple physical computing machines is necessary for simulating a large scale Blockchain network consisting of more than 10 real Blockchain peers. Distributed ns-3 simulation with the use of NVMe-oF such as KUMOSCALE, which provides a high-throughput networked SSD storage with allocating per-machine SSD volumes, is left for future work. We also plan to develop efficient Blockchain-based distributed storage systems in the future.

References

[1] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "IEEE 802.11s: The WLAN Mesh Standard," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, 2010.

[2] Nsnam, "ns-3 network simulator," [Accessed: 2020-06-10]. [Online]. Available: https://www.nsnam.org/

[3] R. C. Carrano, L. C. Magalhães, D. C. Saade, and C. V. Albuquerque, "IEEE 802.11s multihop MAC: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 1, pp. 52–67, 2011.

[4] Y. J. Lai, Y. Ng, T. Sakoda, Y. Bando, A. Miyamoto, M. Ishiyama, K. I. Maeda, and Y. Doi, "Real and simulator testbeds for content dissemination in high-density large-scale WANET," in *14th IEEE Annual Consumer Communications and Networking Conference (CCNC), Las Vegas, USA*, July 2017.

[5] V. Ortega, F. Bouchmal, and J. F. Monserrat, "Trusted 5G Vehicular Networks: Blockchains and Content-Centric Networking," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 121–127, 2018.

[6] IPFS, "InterPlanetary File System," [Accessed: 2020-10-06]. [Online]. Available: https://ipfs.io/

[7] H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[8] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[9] M. Selimi, A. Kabbinale, A. Ali, L. Navarro, and A. Sathiaseelan, "Towards blockchain-enabled wireless mesh networks," in *1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock) , Munich, Germany*, June 2018.

[10] S. Rouhani and R. Deters, "Blockchain based access control systems: State of the art and challenges," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI), Thessaloniki Greece*, October 2019.

[11] Z. E. Lee, R. L. H. Chua, S. L. Keoh, and Y. Ohba, "Performance Evaluation of Big Data Processing at the Edge for IoT-Blockchain Applications," in *IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA*, December 2019.

[12] "Kumoscale," [Accessed: 2020-06-10]. [Online]. Available: https://kumoscale.kioxia.com/

[13] "Hyperledger Sawtooth," [Accessed: 2021-02-08]. [Online]. Available: https://sawtooth.hyperledger.org/