UWS Academic Portal

Blockchain-enabled nextgen service architecture for mobile internet offload

Singh, Raman; Pervez, Zeeshan; Tewari, Hitesh

Published in:
Future Internet

DOI:
10.3390/fi15050173

Published: 05/05/2023

Document Version
Publisher's PDF, also known as Version of record

Link to publication on the UWS Academic Portal

Citation for published version (APA):
Singh, R., Pervez, Z., & Tewari, H. (2023). Blockchain-enabled nextgen service architecture for mobile internet offload. *Future Internet*, *15*(5), [173]. https://doi.org/10.3390/fi15050173

General rights
Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact pure@uws.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 31 May 2023

future internet

MDPI

Article

Blockchain-Enabled NextGen Service Architecture for Mobile Internet Offload

Raman Singh [1,†], Zeeshan Pervez [1,†] and Hitesh Tewari [2,*,†]

[1] School of Computing, Engineering and Physical Sciences, University of the West of Scotland, Hamilton G72 0LH, UK
[2] School of Computer Science and Statistics, Trinity College Dublin, D02 PN40 Dublin, Ireland
[*] Correspondence: htewari@tcd.ie
[†] These authors contributed equally to this work.

Abstract: The amalgamation of heterogeneous generations of mobile cellular networks around the globe has resulted in diverse data speed experiences for end users. At present, there are no defined mechanisms in place for subscribers of a mobile network operator (MNO) to use the services of third-party WiFi providers. MNOs also have no standardized procedures to securely interact with each other, and allow their subscribers to use third-party services on a pay-as-you-go basis. This paper proposes a blockchain-enabled offloading framework that allows a subscriber of a mobile operator to temporarily use another MNO or WiFi provider's higher-speed network. A smart contract is employed to allow diverse entities, such as MNOs, brokers and WiFi providers, to automatically execute mutual agreements, to enable the utilization of third-party infrastructure in a secure and controlled manner. The proposed framework is tested using Ethereum's testnet on the Goerli network using Alchemy and Hardhat. The analysis of the results obtained shows that the proposed technique helps mobile operators to offer improved user experience in the form of average speed and latency. The experiments show that the average time taken to deliver a 500 MB file is reduced from 10.23 s to 0.91 s for the global average scenario, from 6.09 s to 0.50 s for 5G, from 13.50 s to 0.50 s for 4G-LTE, from 41.11 s to 0.49 s for 4G, and from 339.11 s to 0.49 s for the 3G scenario. The results also show that, with WiFi offloading, users from all cellular generations can enjoy a similar quality of services, because delivery time ranges from 0.49 s to 0.91 s for offloaded experiments whereas for the non-offloaded scenario it ranges from 6.09 s to 339.11 s.

Keywords: blockchain; 5G; WiFi; offload; smart contracts

check for updates

Citation: Singh, R.; Pervez, Z.; Tewari, H. Blockchain-Enabled NextGen Service Architecture for Mobile Internet Offload. *Future Internet* **2023**, *15*, 173. https://doi.org/10.3390/fi15050173

Academic Editor: Sachin Sharma

Received: 2 April 2023
Revised: 28 April 2023
Accepted: 3 May 2023
Published: 5 May 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The global rollout of 5G networks is gathering momentum as more and more countries start deploying this state-of-the-art broadband cellular network technology. At the same time, many countries still have operational legacy mobile networks such as 4G, 4G-LTE, or, surprisingly, 3G. Even in places where 5G networks are available, coverage is not always universal, and many pockets exist that still run older-generation networks. A report published by the GSM Association (GSMA) [1] suggests that, at the end of 2019, 4G coverage was about 50% of the total mobile Internet availability by geographical area. On the other hand, due to advantages such as lower cost and fibre infrastructure, WiFi still provides higher bandwidth speeds to its users. It is popular amongst small and large organizations, and also in retail and residential settings.

Today, the world is using mobile cellular technologies such as 3G, 4G, 4G-LTE and 5G with varying data transfer speeds. We believe that an improved user experience can be gained by offloading such cellular network users to local, higher-speed networks. For example, if WiFi providers could allow mobile cellular subscribers to use their fixed broadband infrastructure, and in return obtain a monetary reward for their services from a mobile network operator (MNO), then the traffic load on cellular networks could be reduced while simultaneously increasing the data speed offered to users.

Our proposed framework allows independent private WiFi operators to be paid for their services by offloading users onto their networks, and this in turn means less capital expenditure investment by the MNOs in their own networks. A second reason for offloading is the guarantee of services to subscribers by an MNO in locations where they do not have a license to operate or have poor signal coverage issues. Subscribers can be offloaded to partner WiFi providers and will be able to enjoy enhanced data speeds. The third rationale for offloading is to ensure better service while roaming. For example, a subscriber who does not have roaming enabled on their device, but wishes to use the Internet for short periods, can be offloaded to one of these high-speed networks. The International Telecommunications Society (ITS) has also discussed the business model for public WiFi and its importance for generating revenue. They also emphasize the importance of mobile offloading, along with other aspects such as physical setting, network ownership, service provisioning, revenue model and offloading process management [2].

To allow for subscriber roaming between operators and the settlement of usage charges, MNOs at present have memoranda of understanding (MoUs) drawn up between them on a bilateral basis. However, MoUs are complex agreements which take time to negotiate, and therefore it would not be practical to negotiate MoUs with large numbers of small and medium-sized WiFi providers on a per-MNO basis.

To enable our proposed offloading process, MNOs could register with a *broker* who can collectively negotiate MoUs with many WiFi providers on their behalf. Additionally, blockchain technology can be used to allow participating entities to trust each other by executing smart contracts. The blockchain, along with an appropriate consensus mechanism, allows for the implementation of smart contracts in real terms, and also digitally facilitates, verifies and enforces the contract between two or more parties [3]. Smart contracts can act to bridge gaps between stakeholders and can provide subscribers with a new level of user experience.

The rest of the paper is organized as follows. We begin by introducing the topic and discussing the associated research problem. This section highlights the motivation, objectives, contribution, and goal of the research. Section 2 discusses the work carried out by researchers in the related field. Section 3 explains in detail the five phases that comprise our proposed offloading protocol. To test the proposed framework, the offloading of a subscriber from 3G/4G/4G-LTE/5G networks to a fixed broadband WiFi network is simulated, and the results are analyzed. In Section 4 we present the results of our simulation which was carried out using the ns-3 simulator and Ethereum blockchain. Section 5 discusses the deployment of smart contracts on Ethereum's *Goerli testnet* and shows the result obtained. We conclude with some final remarks in Section 6. To summarize, the goals, objectives, main contribution, and motivation of the research are given below:

- The goal of the proposed research is to provide a blockchain-based offloading scheme and to improve the overall user experience of mobile customers. The users of network providers with slow speeds can temporarily access high-speed networks without formally leaving one network and moving to another network.
- The objectives of the research are to provide the architecture of the proposed scheme along with associated discussion such as its implementation and various implications. The proposed scheme is also simulated, and the results are discussed, to present the overall feasibility of the proposed offloading scheme.
- The main contribution of the proposed scheme is to provide a seamless and efficient blockchain-based offloading mechanism. An offloading mechanism is proposed using blockchain technology, which provides efficient offloading and improves user experience.
- There are various offloading mechanisms available, which are discussed in the related work section, but there is no clarity on offloading allowing low-speed users to temporarily use available WiFi in the area where a user is roaming. The motivation of

the research is to test the proposed offloading scheme, to check the feasibility and technicality of allowing users to temporarily use WiFi and improve user experience.

2. Related Work

In 2011, Radisys Corporation published a white paper regarding the offloading of Internet data for mobile operators [4]. The report suggested that, to increase data growth and to provide low-cost services, data can be offloaded to a lower-cost network. In the case of WiFi offload, the report classified it into three categories: hard, optimized and integrated WiFi offload. Hard offload is driven by the user, and the device is configured to work with a private WiFi network. Optimized offload uses SIM-based authentication and should securely connect with public WiFi providers. Integrated offloading is part of the mobile network and supports integration in core networking. All of these offloading mechanisms require credible authentication, execution and pricing mechanisms. Researchers in [5] discussed two approaches to Internet Protocol (IP) flow mobility being developed, the Internet Engineering Task Force (IETF) and the 3rd Generation Partnership Project (3GPP). The researchers suggested that network-based flow mobility is encouraging technology. It can help telecommunication operators to extend their network capability and can help in providing low-cost services. Researchers in [6] discussed different data offloading mechanisms such as Local IP Access (LIPA), Selected IP Traffic Offload (SIPTO), and IP Flow Mobility. The researchers highlighted various issues to be addressed, such as network management, traffic handling, network deployment for data offloading, energy efficiency, etc. It is also suggested to work on providing desirable Quality of Service (QoS) for offloaded traffic and selecting an optimal offloading point.

The coupling between WiFi and a cellular network during offloading in Radio Access Networks (RAN) can be classified into three different classes: loose, tight and very tight coupling [7]. After studying these three couplings, the researcher found that continuity of session during a vertical handover is a major cause of concern for network architectures. The researchers in one study, [8], analysed smartphone users' behavioural patterns during the usage of an offloading service. A total of 298 different users offloaded data from their smartphones to WiFi networks, and the correlation between the preferences of users and mobile data volume was studied. The authors presented a coefficient value-based mechanism to categorize the different types of users into various offloading behaviours. The authors emphasized the use of the offloading technique for cellular network users to save money and access better service. They proposed a WiFi-aware mobile data offloading technique through WiFi networks with deadline constraints. The proposed technique can finish file transmission before the deadline and can find optimal ways to reduce the computation required to decide the offloading schedule [9].

The authors in [10] proposed a Satisfaction-based Dynamic Bandwidth Reallocation (SDBR) scheme to improve network efficiency and user satisfaction. The analysis of their results shows the effectiveness of the methodology in increasing the overall revenue and average user experience. An NP-hard Target Set Selection (TSS) problem and heuristic optimization are used to propose a data offloading scheme. This scheme works on polynomial time complexity and can reduce traffic by 20% compared with the heuristic-only scheme [11]. Efficient offloading for users who are commuting is another problem which needs attention. Researchers in [12] proposed unsupervised learning-based techniques for improved offloading performance for commuting users. In [13], researchers proposed a dynamic game-theoretic model for crowd-sourced WiFi offloading, which can help mobile operators to solve their pricing problems. This technique is developed to support mobile operators to expand crowd-sourced WiFi networks and solve users' ever-increasing demand. Exponential learning-based minority game theory is also used in one study [14] to develop a distributed data offloading scheme. The effectiveness of the algorithm is tested based on parameters such as pricing, cellular offered throughput and temperature coefficient. The researchers in [15] proposed a blockchain-based roaming and offloading mechanism for local 5G operators. This proposed solution uses blockchain for agreements between MNOs for the smooth execution of handover and offloads among the subscribers of various MNOs.

A WiFi offloading platform is proposed by researchers in [16]. In this platform, researchers proposed offloading based on the WiFi provider's previous ratings, Internet speed and signal strength. This offloading is carried out at the mobile network's software-defined network (MN-SDN) and initiated automatically without the intervention of the user. Unintentional offloading may result in frequent offloading without the consent of MNOs and increase MNO costs manyfold. The difference between this platform and the proposed framework is that the offloading is carried out after verifying the user's authenticity, requiring the intention of all parties, such as the user, WiFi provider and MNO. A scheme based on blockchain is proposed by the authors of [17] for offloading among various base stations. This scheme utilizes a directed acyclic graph (DAG) and uses game theory for bargaining cost and time. SDBlockEdge is a token-based resource management scheme, proposed in [18], which can be used in collaborative edge computing for the offloading of multiple tasks and is based on blockchain technology. Researchers in [19] also developed a scheme for mobile edge computing to optimize task offloading.

The review of related work concludes that researchers have worked on various techniques for data offloading with the aims of enhancing the performance of bandwidth allocation and providing an optimized pricing mechanism. It was found from the literature review that researchers worked on offloading data from one MNO to another MNO or from one base station to another base station. Blockchain-based offloading from an MNO to a WiFi provider is discussed by one researcher [16], and SDN level reprogramming is proposed to facilitate automatic offloading, but our proposed framework suggests consensus-based agreement and can work within the existing network protocols. The proposed technique supplements the data offloading research by providing efficient offloading management and providing an efficient and secure authentication mechanism.

3. Blockchain-Based Subscriber Offloading Framework

The proposed offloading framework enables subscribers to temporarily offload their data usage from low to high-bandwidth channels without changing their cellular network operator. Figure 1 represents the block diagram of our framework and consists of four primary entities, namely the Mobile Nodes (MNs), a Broker (BK), the MNOs and the WiFi Providers (WPs). The MNs in the system are required to download the NEMO app and configure it with their subscriber information (e.g., IMSI - International Mobile Subscriber Number) and MNO identifier (e.g., HNI - Home Network Identifier) [20]. The BK, as the name suggests, coordinates activities between the entities in the network.

We believe that a GSMA [21]-like entity aptly fits the role of the BK in our system, and all MNOs must register with it. The BK maintains a blockchain node and offers a registration service. The MNO registration process includes setting up a blockchain node for storing smart contracts and transactional data. Organizations that wish to allow their high-speed wireless infrastructure to be used by MNO subscribers must also register with the BK, along with setting up their corresponding blockchain nodes. The BK has oversight during the settlement phase and also acts as a mediator in the case of any disputes.

The third set of entities in the system is the MNOs that allow their subscribers to opt for offloading to a higher-speed network. Reasons for offloading can include low-quality signal coverage, high-speed requirements, roaming to non-serviced areas, or even accessing services provided by particular Data Service providers. This entity includes various functionalities, such as a blockchain interface, authentication module, smart contract module, and billing module. The blockchain interface allows various other blockchain nodes to interact with each other, and to update the ledger periodically, including adding or executing new smart contracts or transactions. The MNO authentication module helps to identify and authenticate a subscriber from the MNO's subscriber database. Because there are many MNOs registered with the BK, and the subscriber should be an active user of that particular

MNO, this module identifies a subscriber from an open smart contract and authenticates its status to verify that the user is a valid subscriber and is authorized to offload.

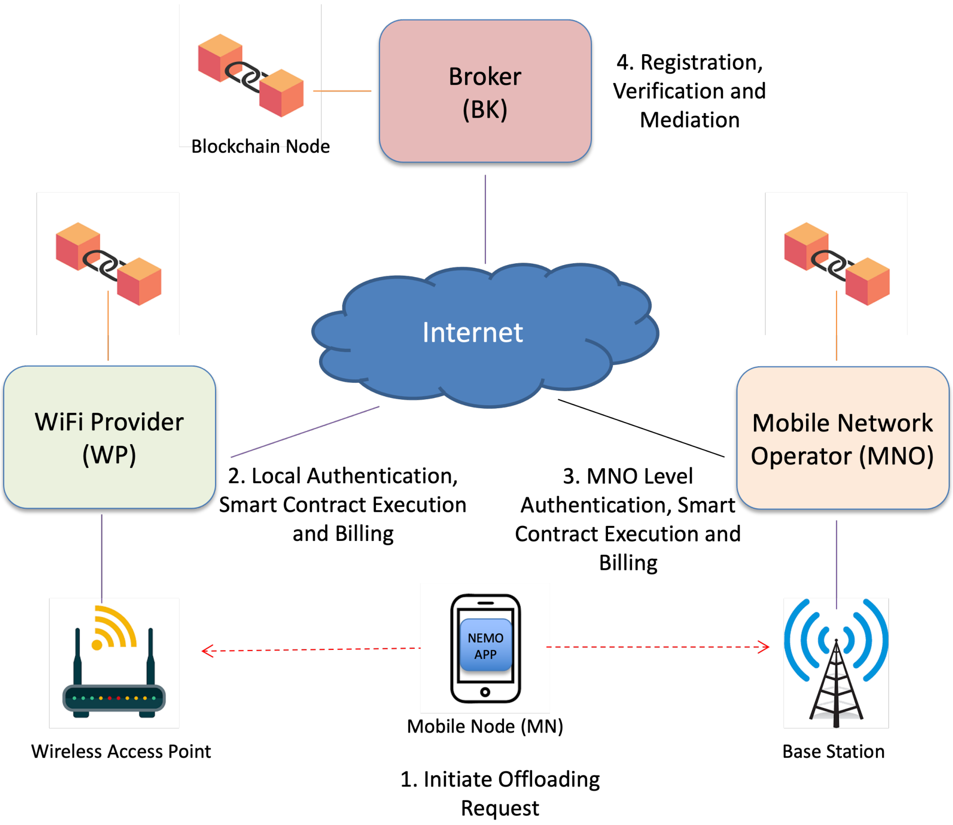

Blockchain Node
Broker (BK)
4. Registration, Verification and Mediation
Internet
WiFi Provider (WP)
Mobile Network Operator (MNO)
2. Local Authentication, Smart Contract Execution and Billing
3. MNO Level Authentication, Smart Contract Execution and Billing
NEMO APP
Wireless Access Point
Mobile Node (MN)
Base Station
1. Initiate Offloading Request

Figure 1. System Architecture.

The smart contract module interacts with open smart contracts to identify users and sets the values of the parameters in the contracts based on the authentication status, such as success or failure. This module can access the smart contract's data based on the authorization allowed and helps in executing it. The billing module can access the transactions stored in the blockchain and create a bill based on the executed smart contract involving a particular MNO. This module then matches the billing amount from the invoice received from the WP and authorizes the payment.

The fourth set of entities is the WPs, which open up their infrastructure in a controlled manner to the subscribers of MNOs. To expedite the offloading process, a WP maintains a blockchain node on its premises. This entity also operates on four modules: the blockchain interface, authentication module, smart contract module and billing module. The blockchain interface is responsible for maintaining an up-to-date smart contract and transaction data, along with the full blockchain. The local authentication module ensures the mobile number ownership of the subscriber by validating a one-time password (OTP).

This local authentication of the mobile number also avoids the spamming of mobile users or blockchain data. For example, if the local authentication of a mobile number is not concluded successfully, spammers may create millions of offload requests using random mobile numbers, which in turn would create a corresponding number of open smart contracts, and force denial of service attacks against legitimate users. The smart contract module allows the WP to create a new smart contract and set its variables based on the subscriber authentication mechanism. Once the subscriber is allowed to offload and subsequently terminates their connection, the billing module records the data usage and writes this as a new transaction to the blockchain. Figure 2 details the various phases of the offloading framework.

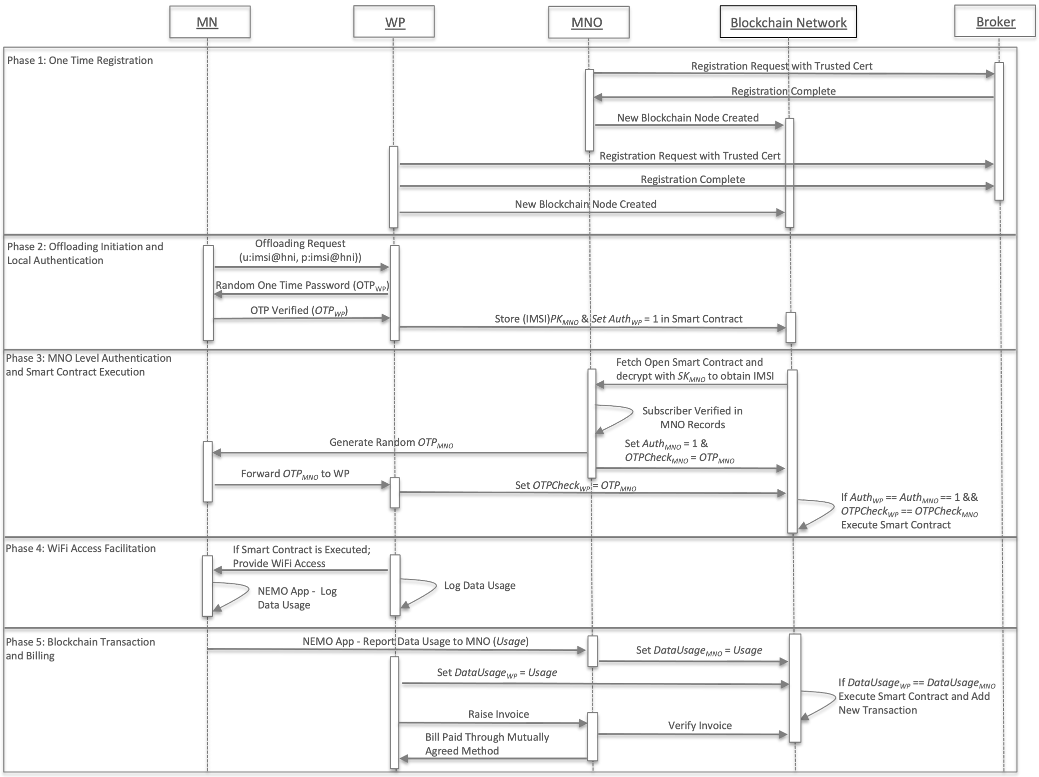

MN
WP
MNO
Blockchain Network
Broker
Phase 1: One Time Registration
Registration Request with Trusted Cert
Registration Complete
New Blockchain Node Created
Registration Request with Trusted Cert
Registration Complete
New Blockchain Node Created
Phase 2: Offloading Initiation and Local Authentication
Offloading Request (u:imsi@hni, p:imsi@hni))
Random One Time Password (OTP_{WP})
OTP Verified (OTP_{WP})
Store (IMSI)PK_{MNO} & Set $Auth_{WP}$ = 1 in Smart Contract
Phase 3: MNO Level Authentication and Smart Contract Execution
Fetch Open Smart Contract and decrypt with SK_{MNO} to obtain IMSI
Subscriber Verified in MNO Records
Generate Random OTP_{MNO}
Set $Auth_{MNO} = 1$ & $OTPCheck_{MNO} = OTP_{MNO}$
Forward OTP_{MNO} to WP
Set $OTPCheck_{WP} = OTP_{MNO}$
If $Auth_{WP} == Auth_{MNO} == 1$ && $OTPCheck_{WP} == OTPCheck_{MNO}$ Execute Smart Contract
Phase 4: WiFi Access Facilitation
If Smart Contract is Executed; Provide WiFi Access
NEMO App - Log Data Usage
Log Data Usage
Phase 5: Blockchain Transaction and Billing
NEMO App - Report Data Usage to MNO (*Usage*)
Set $DataUsage_{MNO} = Usage$
Set $DataUsage_{WP} = Usage$
If $DataUsage_{WP} == DataUsage_{MNO}$ Execute Smart Contract and Add New Transaction
Raise Invoice
Verify Invoice
Bill Paid Through Mutually Agreed Method

Figure 2. Sequence Diagram Illustrating the Various Phases.

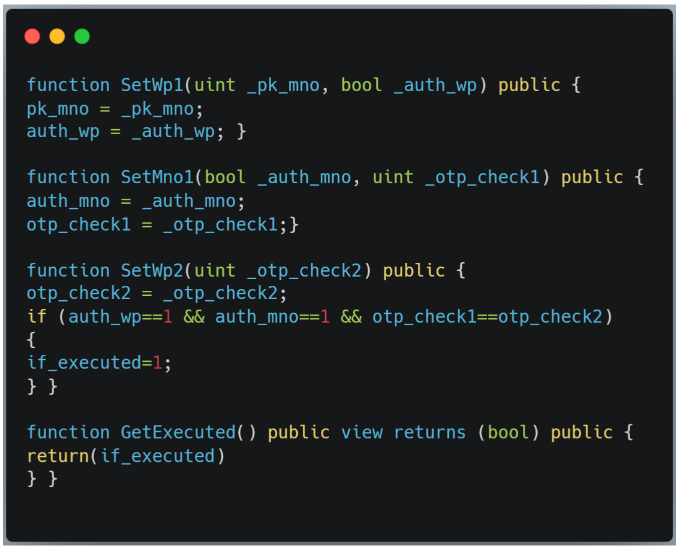
Phase 1: One-time Registration. The registration process for a new entity, such as an MNO or WP, wishing to join the system commences with their trusted third party (TTP) issuing a public-key certificate to the BK. The BK, which maintains its own blockchain node, stores the certificate as a transaction on the blockchain. All the communication amongst the entities in the system is carried out using public-key cryptography, and backed by transparent logs to ensure auditing at a later stage [22]. MNOs and WPs join the blockchain network by creating their corresponding blockchain nodes.

Phase 2: Offloading Initiation and Local Authentication. In phase 2, an offloading request is initiated by the NEMO app on the user's mobile device. The mobile app automatically senses the presence of WPs that are registered with the BK and requests an offload onto their network. The app connects with the wireless access point and accesses the landing page of the WP. The landing page can be common for internal and external users. By supplying the IMSI and HNI of the subscriber, in the form *imsi@hni*, in both the *username* and *password* fields, the app signals that the user is external to the WP and wishes to initiate an offloading procedure.

The WP generates a random one-time password (OTP_{WP}) and sends it to the mobile number entered into the landing page by the app. The mobile app then enters the received OTP_{WP} into the next field of the landing page. Once the WP tests the validity of the mobile number, it generates a new smart contract for this transaction and sets the value of the parameter $Auth_{WP}$ to 1. The WP then encrypts the mobile number of the user with the public key (PK_{MNO}) of the MNO, which it obtains from the blockchain, and assigns this encrypted value to the smart contract. The WP also includes the MNO identifier in the smart contract, so that all other entities know for whom the smart contract is intended.

Phase 3: MNO Level Authentication and Smart Contract Execution. In phase 3, open smart contracts stored on the blockchain are searched by the MNO. Once it finds the intended contract by matching the MNO identifier to itself, it will fetch the encrypted identity of the subscriber and decrypt it using its private key (SK_{MNO}). The decrypted data reveals the mobile number of the subscriber. The MNO tries to verify the identity of the user against its subscriber database, and if successful, sets the value of $Auth_{MNO}$ to 1 in the smart contract. The MNO also generates a random one-time password (OTP_{MNO}), and sets the value of $OTPCheck_{MNO}$ to OTP_{MNO} in the smart contract. The value OTP_{MNO} is also forwarded to the subscriber's MN for further processing.

Once the MN receives OTP_{MNO}, the mobile app enters it into the next field on the landing page of the WP. The WP in turn assigns the OTP_{MNO} value to the $OTPCheck_{WP}$ field of the smart contract. Now, if the values of $Auth_{WP}$ and $Auth_{MNO}$ are both equal to 1, the blockchain deduces that both the WP and MNO have validated the subscriber's identity. If the $OTPCheck_{MNO}$ and $OTPCheck_{WP}$ are the same, it means that the subscriber is authorized to offload, and the smart contract has been executed on the blockchain. The smart contract can also include other information, such as the total time allowed to offload, or any other conditions with the associated offload which need to be honoured by all the parties. The smart contract functionality is implemented in Solidity, and code snippets for various functions are shown in Figure 3.



```
function SetWp1(uint _pk_mno, bool _auth_wp) public {
pk_mno = _pk_mno;
auth_wp = _auth_wp; }

function SetMno1(bool _auth_mno, uint _otp_check1) public {
auth_mno = _auth_mno;
otp_check1 = _otp_check1;}

function SetWp2(uint _otp_check2) public {
otp_check2 = _otp_check2;
if (auth_wp==1 && auth_mno==1 && otp_check1==otp_check2)
{
if_executed=1;
} }

function GetExecuted() public view returns (bool) public {
return(if_executed)
} }
```

Figure 3. Code Snippet of the Smart Contract.

Phase 4: WiFi Access Facilitation and Data Usage Logs. In phase 4, the WP checks the status of the smart contract. If the contract has been executed, the WP allows access to its services to the subscriber provided. Predominantly, the service offered is high-speed Internet access, but it can also be a wide range of other services. When the subscriber terminates the connection, the WP records the data consumed by the subscriber in its logs. We note that the NEMO app also monitors the data usage by the MN, and periodically sends updates to the MNO.

Phase 5: Blockchain Transactions and Billing. Phase 5 deals with the transactions and billing-related procedures. Once the offloaded connection is terminated, the WP will calculate the data usage and execute blockchain transactions along with information such as $DataUsage_{WP}$ and the corresponding smart contract ID. The MNO also enters the data usage of the user, as advised by the NEMO app, into the $DataUsage_{MNO}$ field and executes it as a transaction along with the corresponding smart contract ID. If the $DataUsage_{WP}$ and $DataUsage_{MNO}$ values agree within a predetermined threshold, then this transaction will be used in the next invoice generation. In the case of any dispute, the broker can mediate to resolve it.

The invoice can be generated by the WiFi provider after an agreed period, such as a week or a month. The WiFi provider will access all the transactions made by it from the blockchain data and prepare an invoice based on the mutually agreed price per unit. This invoice will also be verified by the associated MNO and can also be ratified by the BK. Once all parties verify the invoice, the bill will be paid using a mutually agreed out-of-bounds payment mechanism.

The Ethereum blockchain supports scalability for large-scale offloading requests/ transactions using a combination of sharding and side chains. To test the performance of the Ethereum blockchain, researchers measured four million transactions for 380 h [23]. The experiment concluded that throughput decreases, whereas latency increases linearly, if we increase the block period, which is fixed as per the difficulty level of PoW. In our proposed framework, we are using a proof-of-authority (PoA) strategy, so this bottleneck should not affect the overall throughput and latency of the proposed system.

To decrease the time required to complete the workload, the study suggests that powerful machines with high memory and CPUs should be used as blockchain nodes in PoA mode; for example, the computation time for a workload can be reduced by 25% if the memory is increased from 4 GB to 24 GB. The performance of the blockchain network can be improved by keeping the network size as small as possible. For example, in the study, it is found that the successful search ratio for a blockchain with a small size is 90–100%, whereas the successful search ratio is merely 60–75% for larger networks for a given time slot.

4. Case Studies and Result Analysis

The proposed framework was implemented using the ns-3 network simulator [24]. Various ns-3 nodes were created to simulate the different entities, namely, MN, MNO, WP, BK and a Data Server. To implement the blockchain functionality, the Ethereum [25] blockchain is implemented on the Docker [26] platform. Each node in the ns-3 network is connected to a Docker container using the tap-bridge arrangement of ns-3 [27]. In the simulation environment, the subscriber is initially connected to the MNO node, and when the smart contract is executed, the connection is switched over to the WP. The simulation was carried out for 350 s on an Ubuntu Linux-based computer running a virtual machine with 8GB RAM, an Intel i5 2.50 GHz processor, and 100 GB of allocated memory.

The experimentation was conducted over five separate case studies. The first case study takes the global average of Internet speed and latency for fixed broadband and mobile Internet. The bandwidth given for the fixed broadband is assigned to the WP link, whereas the bandwidth given for the mobile Internet is assigned to the MNO link. In this case study, the MN is offloaded from mobile Internet to fixed broadband as per the speed suggested by the global average. In the subsequent case studies, the MN is offloaded from 3G, 4G, 4G-LTE, and 5G mobile Internet to fixed broadband i.e., the WP. The various data transfer speeds and latencies considered for all case studies are provided in Table 1.

Table 1. Case Studies and Associated Parameters.

Case Studies	Network Generation	Average Speed (Mbps)	Latency (ms)	Reference
Case Study 1: Global Average	Fixed Broadband	92	21	[28]
	Mobile Internet	46	36	[28]
Case Study 2: Comparative Average	Fixed Broadband	241	13	[28]
	5G	71	20	[28]
	4G-LTE	50	50	[29]
	4G	10	100	[29]
	3G	1.5	500	[29]

Figure 4 shows the packet delivery percentages for all case studies. The packet delivery is analyzed for all types of flows, such as when the subscriber's packets are not offloaded, for offloaded packets, packets transmitted through the WiFi link, and other packets of different users which are being transmitted through the MNO network.

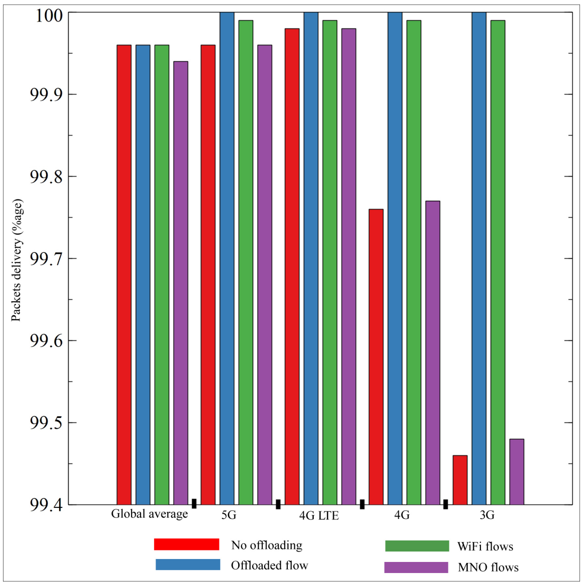


Figure 4. Packet Delivery Analysis for Various Types of Flows.

For the global average case, the packet delivery percentage is the same for all cases except for the MNO flows; however, this difference is insignificant. In this case, 99.96% of packets are delivered for non-offloaded flows, offloaded flows, and WiFi flows, whereas 99.94% of total packets are delivered for the MNO flows. For all other case studies, it is evident from Figure 4 that 100% of offloaded flows are delivered, primarily because of the enhanced data speed of the WiFi link. A slightly smaller number of packets are delivered in the global average case study, compared with all other case studies, because the fixed broadband speed of the global average is lower when compared with other case studies. As the data transfer speed decreases in the 4G and 3G case studies, we can see an increase in packet drop ratio for non-offloaded and MNO flows. If these flows are offloaded to the WiFi network, then the packet delivery ratio rises to a better quality of service requirement.

Figure 5 displays the time taken to deliver a 500 MB file, and a total of 10 such requests are made for each WiFi and MNO network. One request to transfer a 500 MB file is then offloaded to a high-speed network. From this figure, it is evident that the time taken to transfer files is significantly reduced in the case of the offloaded flow. In the global average case study, the non-offloaded flow takes 10.23 s to transfer one file, whereas the time is reduced to 0.91 s if the request is offloaded. Similarly, the graph shows a drastic reduction in delivery time in all other case studies. The longest time taken is 339.11 s by the 3G network to deliver the file; this time is reduced to only 0.49 s if this flow is offloaded to the WiFi network.

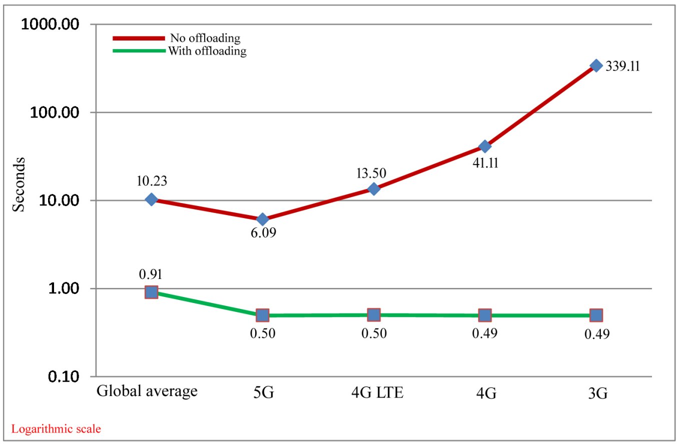


Figure 5. Total Flow Duration Analysis for Offloading and No Offloading.

Figure 6 presents the analysis of delay sum and jitter sum for all the case studies. The delay sum is the addition of all delays for each packet for the full duration of the flow, whereas the jitter sum is the addition of all jitter for every packet for any particular flow. For the global average, the delay sum is calculated as 598.50 s. For other case studies, such as 5G, 4G-LTE, 4G, and 3G, it is computed as 390.32, 806.39, 1749.86, and 8466.05, respectively, for the non-offloaded flows. If compared with the delay sum obtained by the offloaded flows, we can see a drastic reduction. The delay sums obtained for the offloaded flows are 45.34, 16.95, 16.75, 16.42, and 16.42 for the global average, 5G, 4G-LTE, 4G and 3G.

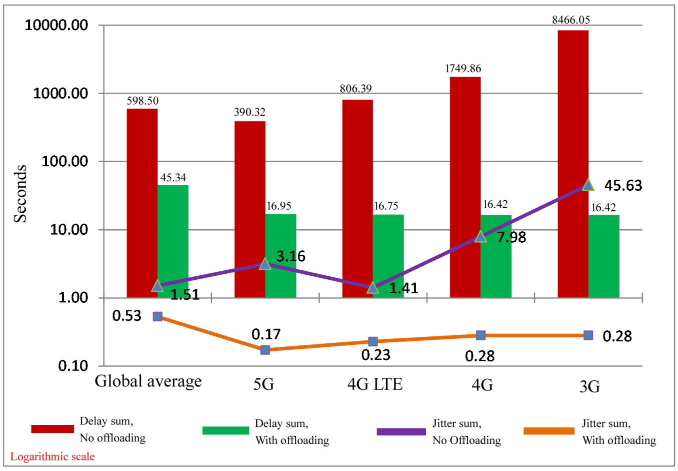


Figure 6. Delay and Jitter Sum Analysis for Offloading and No Offloading.

Jitter sum is also improved in the case of offloading, whereas it is high if no offloading is performed. The jitter sum of 1.51 s is reduced to 0.53 s for the global average case study if offloading is performed. It is reduced to 0.17 s from 3.16 for 5G, 0.23 s from 1.41 s for 4G-LTE, 0.28 s from 7.98 s for 4G, and an impressive 0.28 s from 45.63 s for the 3G case study if offloading is performed. The enhanced packet delivery ratio, along with improved delivery time, reduced delay sum and reduced jitter sum, enhance the overall quality of the experience provided by the MNOs to their subscribers.

5. Deployment and Testing on Blockchain Network

The smart contract of the proposed framework is deployed on Ethereum's *testnet* to understand the behaviour of the smart contract in a real blockchain environment. The working conditions of the *testnet* are similar to *mainnet*, except the factor that *Ether* used in *testnet* have no real monetary value. The smart contract is first tested using *hardhat* [30], which is an Ethereum development environment using various test cases. A *Metamask* wallet [31] is created to store the *Ether* used for transactions on the *testnet* blockchain. Alchemy [32] is a web3 development platform used to deploy the smart contract on *testnet*. This platform provides node management for decentralized applications and helps in creating, testing and monitoring web3 applications. The smart contract is deployed on *Goerli testnet* because it uses proof-of-authority (PoA) as a consensus algorithm, and the proposed framework suggests this consensus algorithm. The deployment process of a smart contract on the Ethereum testnet can be followed using Web3 University's deployment guide [33].

The execution of the smart contract on Ethereum's *testnet* is analysed based on various parameters, such as computer units, median response time, transaction fee, gas price, and Mempool time.

Table 2 shows the results obtained from *testnet* for three use cases: deployment, successful offloading and rejected offloading. The use case of "deployment" corresponds to the first-time deployment of the smart contract on the blockchain. The use case of "Successful Offloading" indicates a case when the user, WiFi provider and MNO complete all procedures correctly and offloading is offered successfully. The use case of "Rejected Offloading" means that something wrong happens, such as a user providing the incorrect number, the user providing the incorrect OTP, etc., and the smart contract is not executed, hence, offloading is rejected.

Table 2. Performance Evaluation Results of the Deployed Blockchain.

Use Cases Parameters	Deployment	Successful Offloading	Rejected Offloading
Total requests	1	538	311
Average Compute Units	1.9	4.95	4.34
Median Response (ms)	19	21	20
Min Transaction Fee (Ether)	$1.067369019 \times 10^{-8}$	$3.96528108050540 \times 10^{-4}$	$4.35436518074810 \times 10^{-4}$
Max Transaction Fee (Ether)	-	$6.04122568830420 \times 10^{-4}$	$6.00562306691220 \times 10^{-4}$
Min Gas Price (Gwei)	43.370762893	14.970103747	16.448946739
Max Gas Price (Gwei)	-	19.635407054	19.526036567
Min MEMPOOL Time	00:00:08	00:00:07	00:00:03
Max MEMPOOL Time	-	00:00:30	00:00:35
Average MEMPOOL Time	-	00:00:17	00:00:20

The experiment was carried out multiple times to obtain the average results; for example, “Successful Offloading” use case requests are made 538 times, whereas for the “Rejected Offloading” use case, 311 requests are made. In the “deployment” use case, 1.9 of the computing unit is used, and the median response is received in 19 ms. For “Successful Offloading”, i.e., successful smart code execution, 4.95 of the average computing unit is used, whereas the response is received in 21 ms.

For the “Rejected Offloading” use case, an average computing unit of 4.34 is used, and the response is received in 20 ms. The transaction fee for the “deployment” use case to deploy the smart contract on blockchain is 0.00000001067369019 Ether, whereas a gas price of 43.370762893 is used. For the use cases “Successful Offloading” and “Rejected Offloading”, the minimum transaction fees are 0.000396528108050536 and 0.000435436518074808 Ether, respectively, among all the requests made during the experiments. For the use cases “Successful Offloading” and “Rejected Offloading”, the maximum transaction fees during all the experiments are 0.000604122568830418 and 0.000600562306691219 Ether, respectively.

The minimum gas prices during the experiments are 14.970103747 and 16.448946739, whereas the maximum gas prices are 19.635407054 and 19.526036567 for the use cases “Successful Offloading” and “Rejected Offloading”, respectively. The Mempool time in the case of deploying the smart contract is 00:00:08. For other use cases, the minimum Mempool times received are 00:00:07 and 00:00:03, whereas the maximum Mempool times are 00:00:30 and 00:00:35, for “Successful Offloading” and “Rejected Offloading”, respectively. The average Mempool times for both use cases are calculated to be 00:00:17 and 00:00:20 respectively.

The results obtained are promising, but, for smooth quality of service, the proposed framework suggests using private blockchain, which should significantly reduce the median response time. In future, we would like to test this hypothesis by testing the proposed framework on a private blockchain. The transaction fees and gas fees are costly if the system uses Ethereum’s *mainnet*, and hence, using private blockchain will remove the cost issue of the deployed blockchain.

6. Conclusions

In this paper, an offloading framework is presented with its several advantages and infrastructural benefits. It can facilitate MNOs to allow their subscribers to benefit from third-party operator high-speed infrastructure for a particular time period. Our proposed framework deploys smart contracts for authentication, thereby allowing the subscriber to offload, and utilizes blockchain transactions to record and generate invoices. The MNO and private WiFi provider both authenticate the subscriber and their mobile number to rule out any spamming of the system. All the transactions are verified by each participating entity, and new blocks are added using a PoA consensus mechanism to minimize the mining effort.

The proposed framework was simulated using ns-3, and the Ethereum blockchain was integrated into the simulation environment using Docker containers. A total of five case studies, i.e., global average speed, 5G, 4G-LTE, 4G, and 3G to WiFi offloading, were tested. The offloading delivered 100% of packets because of improved bandwidth availability. It is found that even 3G customers can enjoy improved flow duration if offloading services are provided to them. For example, a 500 MB file is delivered in just 0.49 s for the offloaded customer, compared with 339.11 s for the non-offloaded customer. The results show improved delay sums of 45.34, 16.95, 16.75, 16.42, and 16.42 for the global average, 5G, 4G-LTE, 4G and 3G, for the offloaded traffic. Similarly, jitter sums of 0.53, 0.16, 0.23, 0.28, and 0.28 s are obtained for the global average, 5G, 4G LTE, 4G and 3G, respectively, for the offloaded traffic. The final analysis shows that offloading results in improved packet delivery ratios and reduced total flow duration, total delay, and total jitter. These parameters suggest that offloading can help in enhancing end users' quality of service experience. At present, Internet speeds vary geographically as well as between operators. A user has no choice but to switch operators if they need higher speeds or services that cannot be provided by their operator. Our proposed offloading framework can be a great leap forward for subscribers who can enjoy higher bandwidth speeds on an on-demand basis.

There is a need for more research on the scalability part of the proposed offloading scheme, so, in the future, challenges such as the scalability of blockchain transactions, simultaneous subscriber load, etc. can be analysed. The performance of the proposed scheme has not been tested for a varying load of users' offloading requests; hence, a time lag analysis of the high load of subscriber offloading requests can also be carried out. In addition, the same offloading framework can be investigated in relation to the automated switching of network traffic from high-congested channels to low-congested channels. This automated and agent-based framework could support load balancing and the optimization of next-generation network infrastructure. One limitation of the proposed scheme is that it has yet to be evaluated on network-based parameters such as switching between networks, frequency of disconnections during fast movement, etc. In the future, the proposed scheme can be tested and further improvements can be proposed for such parameters.

Author Contributions: Conceptualization, R.S., H.T. and Z.P.; methodology, Z.P.; software, Z.P.; validation, H.T. and Z.P.; formal analysis, R.S., H.T. and Z.P.; investigation, Z.P.; resources, H.T.; data curation, R.S. and H.T.; writing—original draft preparation, R.S.; writing—review and editing, H.T. and Z.P.; visualization, R.S.; supervision, H.T.; project administration, H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
TTP	Trusted Third Party
PUF	Physical Unclonable Function
PoA	Proof-of-Authority
mbps	Megabits per second
ms	Milliseconds
deg_{ij}	Degree of Chebyshev polynomial for an IoT device (jth device of ith TTP)
var_{ij}	Value of variable for a given Chebyshev polynomial
Tdevij	Chebyshev value for an IoT device
Tblockij	Chebyshev value for computed by smart contract for an IoT device
Tdevblockij	Chebyshev value computed by smart contract using Tdevij

References

1. The State of Mobile Internet Connectivity. 2020. Available online: https://www.gsma.com/r/wp-content/uploads/2020/09/GSMA-State-of-Mobile-Internet-Connectivity-Report-2020.pdf (accessed on 28 March 2022).
2. Kaiser, T.W.; Verbrugge, S.; Van der Wee, M.; Colle, D. An Overview of Different Business Models for Public Wi-Fi and Their Implications on Indirect Revenue. 2017. Available online: https://biblio.ugent.be/publication/8531252/file/8531254.pdf (accessed on 2 May 2023).
3. Wang, S.; Ouyang, L.; Yuan, Y.; Ni, X.; Han, X.; Wang, F.Y. Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2266–2277. [CrossRef]
4. Brown, G. *Internet Offload for Mobile Operators*; Heavy Reading, White Paper. 2011. Available online: http://go.radisys.com/rs/radisys/images/paper-dpi-internet-offload.pdf (accessed on 2 May 2023).
5. De la Oliva, A.; Bernardos, C.J.; Calderon, M.; Melia, T.; Zuniga, J.C. IP flow mobility: Smart traffic offload for future wireless networks. *IEEE Commun. Mag.* **2011**, *49*, 124–132. [CrossRef]
6. Samdanis, K.; Taleb, T.; Schmid, S. Traffic offload enhancements for eUTRAN. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 884–896. [CrossRef]
7. Khadraoui, Y.; Lagrange, X.; Gravey, A. A survey of available features for mobile traffic offload. In Proceedings of the European Wireless 2014, 20th European Wireless Conference, Barcelona, Spain, 14–16 May 2014; pp. 1–4.
8. Husnjak, S.; Peraković, D.; Forenbacher, I. Data traffic offload from mobile to wi-fi networks: Behavioural patterns of smartphone users. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 2608419. [CrossRef]
9. Tang, W.; Wu, C.; Qi, L.; Zhang, X.; Xu, X.; Dou, W. A WiFi-aware method for mobile data offloading with deadline constraints. *Concurr. Comput. Pract. Exp.* **2021**, *33*, 1. [CrossRef]
10. Bhooanusas, N.; Sou, S.I.; Cheng, K.C. Satisfaction-based Dynamic Bandwidth Reallocation for multipath mobile data offloading. *Comput. Netw.* **2021**, *185*, 107594. [CrossRef]
11. Sharma, P.; Shukla, S.; Vasudeva, A. Data offloading via optimal target set selection in opportunistic networks. *Mob. Netw. Appl.* **2021**, *26*, 1270–1280. [CrossRef]
12. Lima, E.; Aguiar, A.; Carvalho, P.; Viana, A.C. Human Mobility Support for Personalised Data Offloading. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 1505–1520. [CrossRef]
13. Hao, S.; Duan, L. To Help or Disturb: Introduction of Crowdsourced WiFi to 5G Networks. *IEEE Trans. Mob. Comput.* **2022**. [CrossRef]
14. Majumder, B.; Venkatesh, T. Mobile data offloading based on minority game theoretic framework. *Wirel. Netw.* **2022**, *28*, 2967–2982. [CrossRef]
15. Weerasinghe, N.; Hewa, T.; Dissanayake, M.; Ylianttila, M.; Liyanage, M. Blockchain-based roaming and offload service platform for local 5G operators. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6.
16. Fernando, P.; Gunawardhana, L.; Rajapakshe, W.; Dananjaya, M.; Gamage, T.; Liyanage, M. Blockchain-based Wi-Fi offloading platform for 5G. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
17. Hassija, V.; Chamola, V.; Gupta, V.; Chalapathi, G.S. A blockchain based framework for secure data offloading in tactile Internet environment. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 1836–1841.
18. Latif, Z.; Lee, C.; Sharif, K.; Helal, S. SDBlockEdge: SDN-Blockchain Enabled Multihop Task Offloading in Collaborative Edge Computing. *IEEE Sens. J.* **2022**, *22*, 15537–15548. [CrossRef]
19. Naouri, A.; Wu, H.; Nouri, N.A.; Dhelim, S.; Ning, H. A novel framework for mobile-edge computing by optimizing task offloading. *IEEE Internet Things J.* **2021**, *8*, 13065–13076. [CrossRef]
20. Digital Cellular Telecommunications System (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Numbering, Addressing and Identification (3GPP TS 23.003 Version 15.6.0 Release 15). Available online: https://www.etsi.org/deliver/etsi_ts/123000_123099/123003/15.06.00_60/ts_123003v150600p.pdf (accessed on 28 October 2022).
21. Global System for Mobile Communication Association (GSMA). Available online: https://www.gsma.com (accessed on 28 October 2022).
22. Gasser, O.; Hof, B.; Helm, M.; Korczynski, M.; Holz, R.; Carle, G. In log we trust: Revealing poor security practices with certificate transparency logs and internet measurements. In Proceedings of the International Conference on Passive and Active Network Measurement, Berlin, Germany, 26–27 March 2018; pp. 173–185.
23. Schäffer, M.; Angelo, M.D.; Salzer, G. Performance and scalability of private Ethereum blockchains. In Proceedings of the International Conference on Business Process Management, Vienna, Austria, 1–6 September 2019; pp. 103–118.
24. ns-3 a Discrete-Event Network Simulator. Available online: https://www.nsnam.org (accessed on 28 October 2022).
25. Ethereum. Available online: https://ethereum.org/en (accessed on 28 October 2022).
26. Docker Container. Available online: https://www.docker.com (accessed on 28 October 2022).
27. StepByStep: Establishing Virtual Network between Docker Container and NS-3 Nodes. Available online: https://sites.google.com/thapar.edu/ramansinghtechpages/step-wise-establishing-connection (accessed on 28 October 2022).
28. Speedtest Global Index. Available online: https://www.speedtest.net/global-index (accessed on 28 October 2022).

29. Rizzatti, L.; Squiers, R.; Castren, M. Design and Verify 5G Systems, Part 1. Available online: https://www.edn.com/design-and-verify-5g-systems-part-1 (accessed on 28 October 2022).
30. Ethereum Development Environment for Professionals. Available online: https://hardhat.org/ (accessed on 15 February 2023).
31. A Crypto Wallet & Gateway to Blockchain Apps. Available online: https://metamask.io/ (accessed on 15 February 2023).
32. The web3 Development Platform. Available online: https://www.alchemy.com/ (accessed on 15 February 2023).
33. Deploy Your First Smart Contract. Available online: https://www.web3.university/tracks/create-a-smart-contract/deploy-your-first-smart-contract (accessed on 21 April 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.