GiNA: A Blockchain-based Gaming scheme towards Ethereum 2.0

Nirav Patel*, Arpit Shukla†, Sudeep Tanwar, *Member, IEEE*‡, Neeraj Kumar, *Senior Member, IEEE*§, Joel J. P. C. Rodrigues, *Fellow of IEEE*¶
*†‡Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India
§Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India
¶Federal University of Piauí (UFPI), Teresina – PI, Brazil; Instituto de Telecomunicações, Portugal
Email: *18bce365@nirmauni.ac.in, †18bce370@nirmauni.ac.in, ‡sudeep.tanwar@nirmauni.ac.in,
§neeraj.kumar@thapar.edu, ¶joeljr@ieee.org

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500723

***Abstract*—With the advent of the Internet, the gaming industry has grown tremendously in business, which also raises concerns for cheating and unfair gameplay. In this paper, we propose a novel approach (*GiNA*) using Blockchain technology to address a few problems with online Peer-to-Peer (P2P) games. *GiNA* uses two different data packet transfer schemes to ensure the security and authenticity of the data packet sent and received by game clients. More sensitive data uses a Smart contract-based ON-CHAIN data packet transfer solution and less sensitive data uses an OFF-CHAIN data packet transfer solution with end-to-end encryption for data security. A marketplace where peers can buy and sell purchasable assets with the help of *Gicoins*. *Gicoins* is a stable token with compliance with the ERC 20 token of Etheruem Blockchain. Later, a low cost and low bandwidth utilization data storage solution is proposed for storing data in a decentralized and distributed manner. Results show that the performance of the proposed approach *GiNA* is better in comparison to the traditional approaches with parameters such as latency, scalability, packet loss percentage, Blockchain (BC) performance, and data storage comparison.**

Index Terms*—Gaming, Blockchain, Sharding, Smart Contract, Ethereum 2.0, Solidity, Latency.

I. Introduction

The gaming industry has evolved tremendously in the past few years with the help of new and better technologies of hardware and software as well. From simple standalone games on computers to online games on mobile, the global gaming market is valued at nearly USD 150 billion and is expected to reach USD 250 billion by 2025. Through this evolution, games of different genres were invented, for example, a game called PUBG in the battle royale genre and a game called Valorant in the first-person shooter genre. Moreover, innovations in network technologies played a pivotal role in the enhancement of the online gaming experience.

Online game architectures are centralized or client/server (C/S) architecture and P2P architecture. The C/S architecture was introduced and works best to handle heavy traffic for Massive Multiplayer Online Games (MMOGs) but it also has few drawbacks like the latency or the round trip time of the packets might increase due to the faraway location of the server and the server needs to be running 24 by 7, which might not be cost-effective for games, which are not MMOGs. On the other side, the P2P architecture might be cost-effective for many games, which are relatively smaller than MMOGs. In P2P architecture, all peers are connected directly where the host is one of the peers connected with the lowest latency communication with other peers, which might create the host as a bottleneck of communication.

The cheating industry for MMOGs has become a multi-million dollar business [1]. In single-player standalone games, where cheating takes place by getting around the computer components through unlimited resources to cheat in online multiplayer games can affect many factors such as popularity and revenue. The game performance is susceptible to the network round trip time, and the C/S architecture creates a bottleneck of the network, which then becomes an easy target for DDoS attacks. Despite P2P's characteristics such as robustness to withstand attacks, C/S architecture will always be precedent because P2P clients are deemed to be untrustworthy and prevention of the attacks will be difficult in the absence of a trusted intermediary.

To maintain the security of the game, the gaming companies have taken several measures, which involve monitoring the game state, which can be done either on both servers and clients end or any one of them. Introducing blockchain (BC) technology in the gaming industry will solve many issues such as cheating. Since BC mandates a P2P architecture, we get the benefits of P2P and BC combined to form a robust system to prevent cheating [2]. While using BC makes it easy to migrate to P2P architecture it also makes the game state publicly visible to all peers connected in the network, which can be seen by all the peers and peers can take advantage of it.

Researchers across the globe have given their insights on the adoption of BC to increase the security in on-line gaming. For example, Christian *et al.* [3] presents an approach to prevent modification of game-client and also accessing the sensitive information in the game client's memory by allowing to keep separate the aspects of the game and the security-related issue. Then, many researchers started focusing on security issues faced by the gaming industry. To prevent gaining access to games that should not be accessed by peers also known as *Sequence Breaking* was enlightened by Matthew *et al.* [4] with the help of use case maps but failed to handle the robustness of the system for handling MMOGs. Later, Mogaki *et al.* [5] used a time-stamping server as an infrastructure to propose a cheat-free protocol for online gaming.

P2P architecture allows games to circumvent the scalability issues and server bottleneck is reduced or eliminated but with this, it also gives clients more authority over the game state. For example, Josh *et al.* [6] proposed an approach that tries to create a P2P scheme in which the scalability and security coexist with providing low server cost. Maphacking is done by peers to get more content on the game map than is to

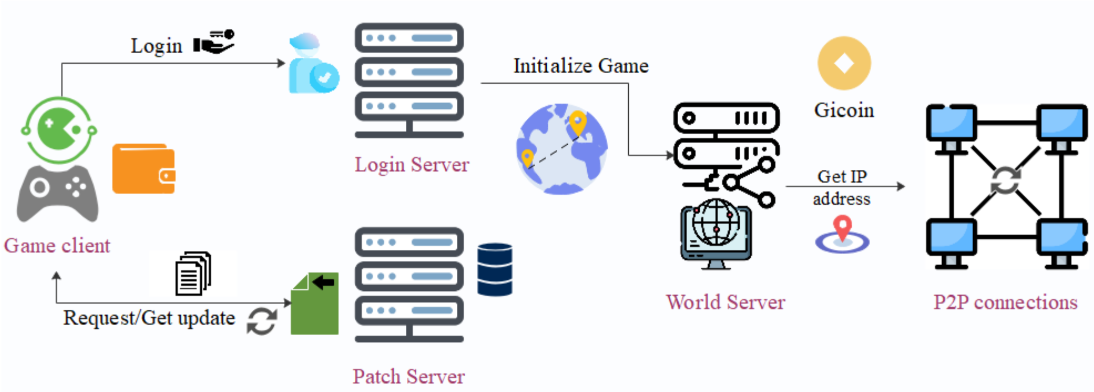


Fig. 1: Overview of *GiNA*

be accessed by any peer. Later, Chris *et al.* [7] proposed a technique to detect map hacking based on the bit commitment and network traffic in online P2P games. Mirror servers for network games will help to make the system more tolerant to delay of packages. Steven *et al.* [8] proposed an enhanced mirrored servers (EMS) architecture to minimize the delay by EMS simulating the game only once for each update and does not require rollbacks.

BC is a distributed ledger used to store transactions between multiple parties permanently, often parties are mutually distrusting each other. For example, Dhevi *et al.* [9] used BC technology to provide a solution to keep track of the coins earned and lost by the peers to make robust security features. However, using BC technology for cheat-proof gaming is a challenging task. For example, Sumit *et al.* [10] presented the first practical approach to prevent cheating in MOGs using BC, which provides scalability and robustness by design and also enables legitimate game customization using hyperledger. Later, Ho Yin *et al.* [11] proposed a novel consensus model called Proof-of-Play, which integrated into P2P architecture with very little interference to the game.

A. Motivation

However, maintaining security in P2P architectures is difficult as the simulation and game state is distributed to untrusted peer machines, consistency algorithms are required to prevent errors in peer's game state, and redundancy must be built in as individual peer machines have a high probability of failure. Further, subscription-based P2P network games will require a gateway server for peer authentication and billing. Several hybrid C/S and P2P architectures that combine the security advantage in C/S and scalability in P2P have also been proposed. No decentralized data solution for gaming has been proposed. Hence, the authors are motivated to propose a scheme that solves the aforementioned issues of the presented works. In this paper, we propose *GiNA*, a scheme to ensure the security and the authenticity of the data received and send by the peers to other peers using BC.

B. Research contributions

The research contributions of this paper are as follows.

- A novel sharding based highly scalable BC-based scheme is proposed for seamless peer to peer online gaming towards Ethereum 2.0.
- Then, gaming assets are backed up with real value using ERC-20 compliant token standards to protect time and money investments by individual game peers.
- A unique data access mechanism is designed based on the InterPlanetary File System (IPFS) to achieve low latency and high throughput.
- Finally, we compare the proposed approach with traditional schemes with parameters such as latency, scalability, packet loss percentage, BC performance, and data storage comparison.

C. Organization

The rest of the paper is organized as follows. Section II discusses the problem formulation and system model. Section III discusses the proposed approach *GiNA*. Section IV presents the performance analysis of the proposed approach *GiNA* and finally, Section V concludes the paper.

II. System Model and Problem Formulation

Game client is a client that connects a peer to the game server. Furthermore, a game client is responsible to collect and send data such as score, position, and movement to the other game client in the proposed scheme *GiNA*. Starting game client is followed by login into the game server as authentication, which is done by the login server. The login server is a lower power server that is used only to authenticate the peer's credentials as shown in Algorithm 1. The game client, before logging in, checks if any update is available for the game client, and for that, there is a patch server that is only responsible to provide the new patch or update of the game. Once the update and login process is completed, the game client is redirected to the world server to get into the P2P network with other game clients.

World server is responsible for the matchmaking of the game clients to initiate the game, send IP addresses to all the game clients who are assigned to the same game and assign one of the game clients as host of the game. The host is one of the peers connected with the lowest latency communication with other game clients. Furthermore, the world server does not render any data rather the data provided by the game clients to the world server are usually the result and statistics of the game. Here, the world server does not perform any high-level processing thus will require a less powerful server than C/S architecture resulting in a low cost of the server.

Fig. 1 shows the overview of *GiNA*, which comprises of the entities $E = \{E_{na}, E_gc, E_s^l, E_s^p, E_s^w\}$. Here, the network

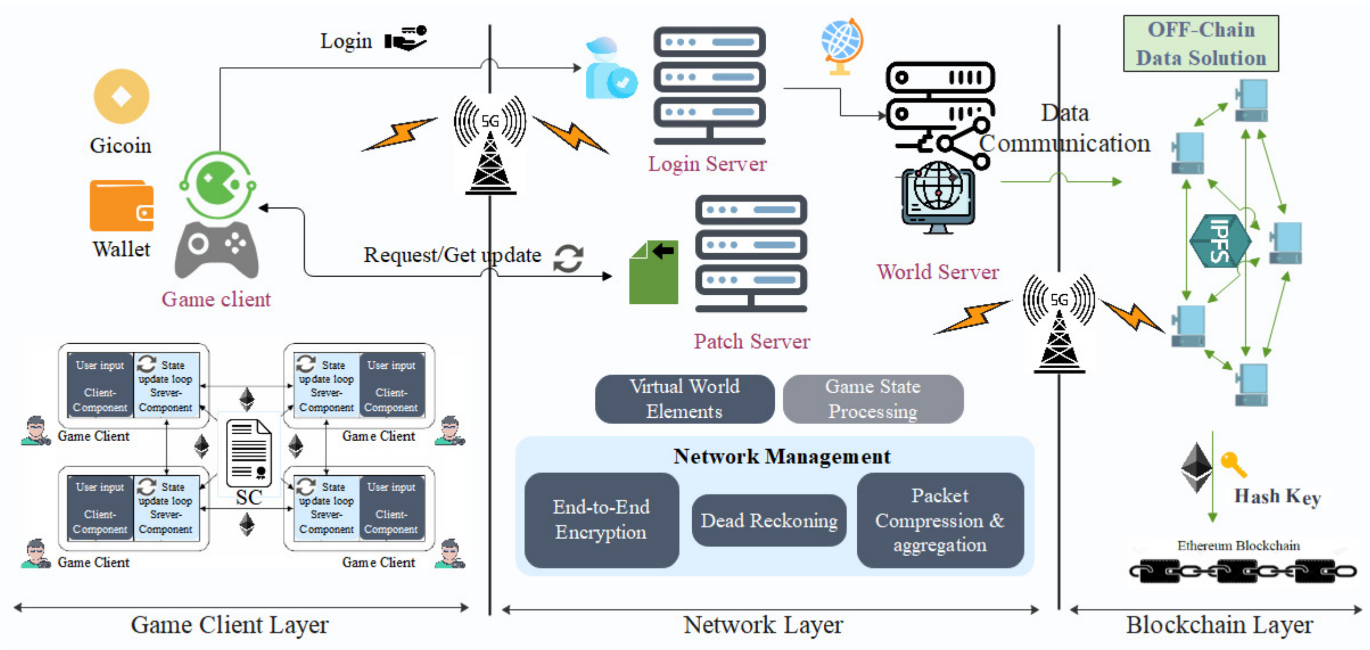


Fig. 2: *GiNA*: The System Architecture

authority entity, i.e., $E_{na} = \{E_{na1}, E_{na2}, \ldots, E_{nam}\}$ is responsible for the stakeholder authentication through third-party E-KYC API during login phase in the system. Also, the game clients are represented as $E_{gc} = \{E_{gc1}, E_{gc2}, \ldots, E_{gco}\}$ that are located inside the gaming device and acts as a game engine. Game servers E_s^l, E_s^p, E_s^w in *GiNA* are login server for gamer authentication using IPFS and patch server for updating the game if needed before login and world server used to store the world state, matchmaking, and dynamic IP assigning to the game peers.

Dead reckoning is used to predict where a peer should be positioned using the peer's last known kinematic state such as position, velocity, acceleration, orientation, and angular velocity. Dead reckoning is needed because of the different RTT of each peer and it's not practical to send network packets at the rate of the game's rendering frequency. Projecting a new future position of the peer using linear physics, as follows.

$$\Gamma_t = \Gamma_0 + \Upsilon_0 \tau + \frac{1}{2} A_0 \tau^2 \tag{1}$$

where Γ_t is the new position projected using Γ_0 current position and Υ_0 velocity of the peer. A_0 as the angular velocity of the moving peer or any object. This equation will help to project the next position of the peer or the object until new data packets arrive from the network. Once the data packet arrives, now having two kinematic states, one projected and one actual arrived from the network is complex to resolve. This can be resolved using the velocity blending of the two projections.

$$\Upsilon_b = \Upsilon_0 + (\hat{\Upsilon} - \Upsilon_0)\hat{\tau} \tag{2}$$

Υ_b is blending using the last known and the current velocity over a set of time. Now, putting Eq. 2 in Eq. 1 then we get:

$$\Gamma_t = \Gamma_0 + \Upsilon_b \tau_t + \frac{1}{2} \grave{A}_0 \tau_t^2 \tag{3}$$

and

$$\grave{\Gamma}_t = \grave{\Gamma}_0 + \grave{\Upsilon}_0 \tau_t + \frac{1}{2} \grave{A}_0 \tau_t^2 \tag{4}$$

Finally, the position projected using Eq.3 and Eq. 4 we get the following.

$$\varrho = \Gamma_t + (\grave{\Gamma}_t - \Gamma_t) \times \hat{\tau} \tag{5}$$

where ϱ is the final position.

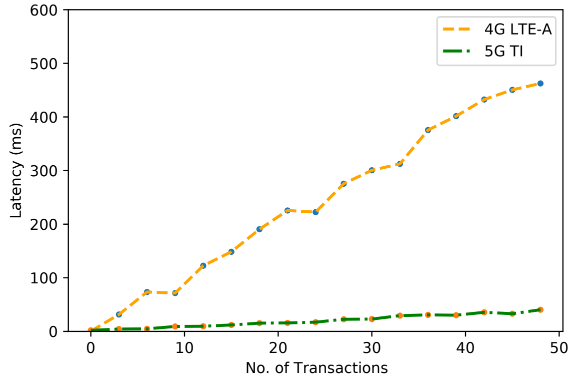
In the proposed scheme *GiNA*, a Smart Contract (SC) controlled system flow is used to secure the data flow and interaction between the game clients. Fig. 2 shows the system architecture of *GiNA* consisting of a game client, load balancer, login and patch server, and world server. These entities are used to connect to the server, login into the server to get access to the game and a patch server to keep the game up to date. In online games, a connection is needed to establish between the game client and game server to send game data and rendering for the client-server architecture. For P2P architecture, game clients are connected to game clients. To connect to other game clients, another game client's IP address is required, which will be provided by the world server to get into the P2P network to play the game.

III. GiNA: The Proposed Approach

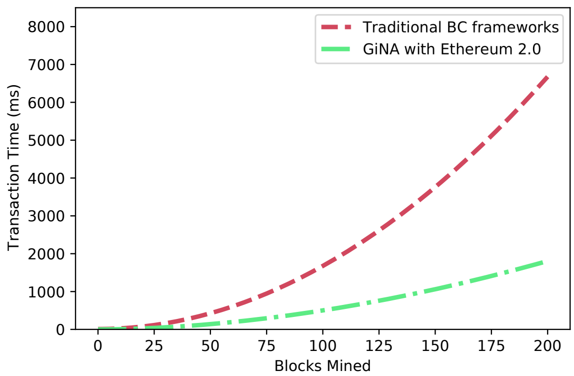
A. Game Client Layer

The game client is a client-side gaming software or interfaces, which connects a player to the game server. Furthermore, a game client is responsible to collect and send data such as score, position, and movement to the other game client in the proposed scheme *GiNA*. The game client has two main components, a client component, and a server component. The client component is responsible for the data to be sent to other peer clients in the game and the server component is responsible to render the game data received by the other players to compute the game state. The game state is the combination of visuals, audio, any animation, or graphics. A game consists of a sequence of states to move forward the Gameplay.

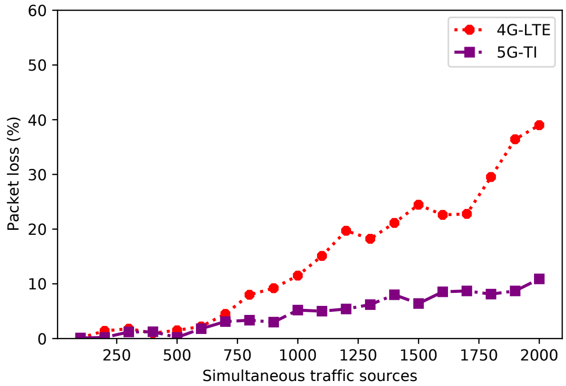
Majority of the cheaters or the hackers try to manipulate



(a) Latency



(b) Scalability



(c) Packet Loss

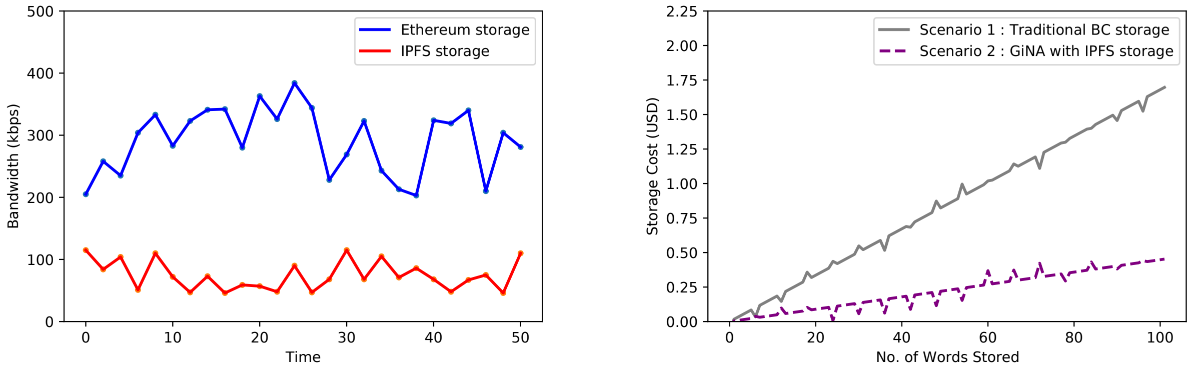
Fig. 3: Performance evaluation of *GiNA*

the game using Dynamic Link Library (DLL) injection. A DLL file consists of code and data that can be simultaneously used by multiple programs through which cheaters or hackers get access to the system memory and can modify the game-related information. By modifying this code, cheaters can get an unfair advantage by getting access to the sensitive data of other players. To prevent this, in the proposed scheme *GiNA*, the data is sent to other players in both ON-CHAIN and OFF-CHAIN manner. The less sensitive data like the movement of the player is sent through OFF-CHAIN and the sensitive data like the Health of the player is sent through ON-CHAIN. SC-controlled system flow is used for easy data flow and interaction between the game clients. The Ethereum BC is based on tokens; tokens can be bought, sold, or traded. Tokens can be digital asset or vouchers or any tangible objects. One of the tokens is ERC-20, which is a technical standard used for token implementation and use of SCs. Each transaction in the proposed system complies with ERC-20 and the *Gicoin* is cryptocurrency generated for the means of transfer. *Gicoins* will be a stable token, the stable token is similar to any item backed by a currency that will be constant. In the proposed system *GiNA* The game clients or the player can buy any additional purchasable assets such as the skin colour of the gun that can be purchased from the game market. Players can also buy or sell these assets directly to other players using *Gicoin*.

B. Network Layer

To initialize the game, the game client needs to login into the game network. Initially, the game client consists of the content delivery network (CDN) of the login server, which will have a game client to log in and authenticate. To initialize the gameplay, several game clients need to be assigned to a game, this task is done by the world server. The world server does the matchmaking with the help of the statistics of the peer which will then be the level of the peer to find another peer with similar or near to the same level as shown in Algorithm 1. Furthermore, several Artificial Intelligence (AI) techniques can be used for matchmaking. Game clients need to update regularly to get new Virtual Game Elements (VGE). These are immutable objects such as landscape information, characters, or avatars and their details and mutable objects such as weapons.

Along with the VGE, game logic might need to be updated for the processing of the game state by the game client. The updates are provided to the game client by the patch server. As mentioned in the game client layer, less sensitive data sent to the other peers OFF-CHAIN of the game will be end-to-end Encrypted. End-to-end Encryption is used to prevent third parties from accessing the data. Data sent to the other peers ON-CHAIN is comparatively more sensitive, SC is used to ensure the security and authenticity of the data as shown in Algorithm 1. With the help of SC, cheating can be prevented from many attacks such as DLL injection. As each peer will



(a) IPFS Bandwidth utilization

(b) Data storage cost comparison

Fig. 4: *GiNA*: Comparison with existing schemes.

Algorithm 1 GiNA : The Technical Flow

Input: $E, GC, User_{id}, User_{pass}$
Output: Game Client login successful
Initialization: Here, GC have already registered into the game database.

procedure GAMECLIENT_LOGIN(W_{GC}, $User_{id}$, $User_{pass}$)
  E_{NA} ← Request_Login(E_{type}:E_{GC}, E_{Sl}, E_{SP}, E_{SW})
  R_m = $Map_{details}(GC, User_id, User_pass)$
  $IPFS_{key}$ ←Login_Details()
  if ($E \in E_{NA}$) **then**
    E ←Process_Request()
    E^{key} ← data.Store($IPFS_{key}$)
    E ←Emit Event("Login Successful")
  else
    E ←Emit Event("Invalid credential details")
  end if
end procedure
procedure MATCH_MAKING(W_{GC}, $User_{stat}$, $User_{lvl}$)
  $Users[] \leftarrow fetch\_online\_peers$
  $peers[] \leftarrow peers_analysis(Users[], User_{stat}, User_{lvl})$
  $User_{opponent} \leftarrow math.random(peers[])$
  if ($User_{opponent} \neq \phi$) **then**
    $User^{status}_{match} \leftarrow status.successful$
  else
    $User^{status}_{match} \leftarrow status.unsuccessful$
    Emit_event('no opponent found !')
    User←break_game()
  end if
  sen_{low}, sen_{high} ←entities.classifyposition,moves,weapon,health,special_ability
  if (entitity $\in sen_{low}$) **then**
    G_{pack} ← enrypt(sha256,entity.value)
    directTransfer(G_{pack})
  else
    G_{pack} ← enrypt(sha256,entity.value)
    BC.publish(entity.value)
    key_{pub}, key_{pri}←generate_keys()
    sendKeys($User_{opponent}$)
  end if
end procedure
procedure MATCH_MAKING(W_{GC}, $User_{stat}$, $User_{lvl}$)
end procedure

have different latency, and the Round Trip Time (RTT) of the data packets of OFF-CHAIN and ON-CHAIN will vary, packet compression and packet aggregation is required for fair gameplay for all the peers which will also be handle by the game Client provided the RTT of packets of all peers by the world server. The arrival of packets to all the game clients will not be synchronized because of the different geographical locations and RTT. To overcome this challenge, estimating or predicting the other peers needs to be done known as *Dead Reckoning*. Dead Reckoning is done with the help of using the last known kinematic state such as position, velocity, and acceleration of the other peers in the game. Every transaction in the scheme is stored along with its timestamp over public BC. Moreover, the communication link used to connect the stakeholders remotely is 5G-enabled [12], which achieves less than 1ms round trip latency and reliability of 99.999% for the success of *GiNA*. Furthermore, Ethereum 2.0 will allow the network to process many transactions concurrently with the help of sharding and proof-of-stake consensus, therefore, all the peers or the peers will be the validators [13].

C. Blockchain Layer

SCs are developed using solidity language. These self-executable programs serve to help add role-specific constraints regarding game asset management and gameplay. Once developed, the SC is compiled using Ethereum Virtual Machine over the ethereum node. After a successful compilation, the bytecode is generated. This bytecode is further embedded and published over the public ethereum BC. Each user in the scheme can access the SC through its ethereum account address, on the initial setup registration is done through SC where the user identity is authenticated using E-KYC third party API as per the global standard policies to ensure security and fairness in the game. The IPFS is a protocol and P2P network for storing data in a decentralized and distributed manner. The IPFS uses content addressing, which means it cannot be retrieved based on location instead of its content. In the proposed scheme *GiNA*, all the data is stored in the IPFS protocol to ensure security. IPFS protocol also allows users to host content. In *GiNA*, the peer's credentials such as user id, the statistics of the game played, and the assets owned by the peers can be stored and as it is decentralized, there will be no bottleneck of data storage. As the data is stored in a decentralized manner, peers can fetch the data from other nodes with the help of a Distributed Hash Table (DHT).

IV. PERFORMANCE ANALYSIS OF *GiNA*

A. Network Parameters

GiNA architecture uses 5G network infrastructure that enables tactile internet and ultra-reliable low latency communication (URLLC) for optimum gaming performance over the p2p network. The network performance evaluation of *GiNA* is done in terms of latency, network packet loss, and bandwidth utilized during communication.

1) Latency: The Latency can be defined as the delay or the time is taken for a signal to travel to its destination and back. Fig. 3a shows the comparison between the 4G LTE-A and

5G TI [14] with the X-axis and Y-axis representing Latency in ms (millisecond) and Number of transactions, respectively. Here, we consider the communication network of *GiNA* is 5G-enabled TI which is more reliable and relatively low latency as shown in Fig. 3a with the linear regression of the calculated latency and No. of transactions values of both the methods. 5G TI is ultra-reliable low latency communication (URLLC) because of the feature of TI through it manages to get a latency of $<$ 1ms with 99.99% of reliability compared to LTE. After latency, comes the scalability of the network.

2) Network Packet Loss: Packet loss can occur due to errors in data transmission or network congestion mostly in wireless networks. Packet loss has occurred when one or more packets fail to reach their destination. Fig. 3c depicts the packet loss in the 4G-LTE enabled network and 5G-TI enabled network. The graph has packet loss in percentage and simultaneous traffic sources as X-axis and Y-axis, respectively. This graph portrays the 5G-TI enables it more reliable as the regression line shows the packet loss increases when the number of sources reaches near 700 but is significantly less than the 4G-LTE enables network.

3) IPFS Bandwidth Utilization: IPFS is a protocol that helps the user to store the data in a P2P network in a distributed system. The data is stored in the IPFS and a *Hash key* is generated. Ethereum BC is also compatible with IPFS. Here, in the contract, we only take the *Hash key* instead of the whole data. The *Hash key* is stored in the BC with which we can access the data stored in the IPFS. The IPFS node is configured and made active to fetch the network bandwidth required for the data storage. Then the dashboard analysis is performed by connecting the system to the internet and the collected data is visualized in Fig. 4a which shows the input and output bandwidth utilization for an IPFS node in a sample time.

B. Comparison with existing schemes

The comparative analysis of *GiNA* is done with traditional systems using the scalability of BC and data storage cost comparison.

1) Blockchain Scalability: Fig. 3b shows the improved scalability based on the transaction time concerning the number of the blocks mined during the requests processing of the game state of the *GiNA* using Ethereum 2 vs the traditional approach of the gaming with BC. As the GiNA is 5G enabled it can accommodate more number transactions in the chain in the same period with ultra-low latency and ultra-high reliability [15]. Hence, *GiNA* improves the overall scalability of gaming. Once the scalability is achieved, the reliability of data reaching its destination is also important.

2) IPFS Data storage cost comparison: Fig. 4b shows the comparison of the two scenarios of storage cost of data in the traditional approach in Ethereum BC with the storage cost of data in *GiNA* in IPFS. In the graph, the comparison is done with the cost in US dollars concerning the number of words stored. The graph illustrates the proposed scheme *GiNA* outshines the storage cost of the Ethereum BC and proves the proposed scheme as cost-effective.

V. Conclusion

In *GiNA*, the P2P gaming scheme is presented over the Ethereum 2.0 BC technology. The scalability and network latency issues are solved using 5G-TI for instantaneous communication during the gameplay. Moreover, the gaming assets can be traded among the peers through cryptocurrency using SC over BC. For this trading, we develop ERC-20 standard-based *Gicoin* crypto token for secured asset trading such as weapons, abilities, accessories, etc. To optimize the BC data storage cost, *GiNA* proposed a P2P IPFS protocol-based storage solution where only the hash key is stored in BC and data is stored off-chain over IPFS. Finally, we analyse the proposed scheme by simulating the SCs over BC, testing asset trading and comparing the performance parameters during the simulation such as latency, scalability, packet loss percentage, BC performance, and data storage comparison. In the future, we will embed advanced deep learning algorithms for network communication efficiency and anomaly detection to avoid cheating in the gameplay.

Acknowledgement

This work is partially funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/50008/2020; and by Brazilian National Council for Scientific and Technological Development - CNPq, via Grant No. 313036/2020-9.

References

[1] "Business of video game cheating.." https://www.pcgamer.com/hacks-an-investigation-into\\-aimbot-dealers-wallhack-users-and-the-million-\\dollar-business-of-video-game-cheating/. Online; Accessed: 2020.

[2] U. Bodkhe, S. Tanwar, K. Parekh, P. Khanpara, S. Tyagi, N. Kumar, and M. Alazab, "Blockchain for industry 4.0: A comprehensive review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020.

[3] C. Mönch, G. Grimen, and R. Midtstraum, "Protecting online games against cheating," in *Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames '06, (New York, NY, USA), p. 20–es, Association for Computing Machinery, 2006.

[4] M. Shelley, W. Shi, and J. . Corriveau, "On preventing sequence breaking in video games," in *2013 12th Annual Workshop on Network and Systems Support for Games (NetGames)*, pp. 1–2, 2013.

[5] S. Mogaki, M. Kamada, T. Yonekura, S. Okamoto, Y. Otaki, and M. B. I. Reaz, "Real-time cheat-free gaming on the basis of time-stamp service," 01 2009.

[6] J. Goodman and C. Verbrugge, "A peer auditing scheme for cheat elimination in mmogs," pp. 9–14, 01 2008.

[7] C. Chambers, W.-c. Feng, W.-c. Feng, and D. Saha, "Mitigating information exposure to cheaters in real-time strategy games," pp. 7–12, 01 2005.

[8] S. Webb, S. Soh, and W. Lau, "Enhanced mirrored servers for network games," pp. 117–122, 01 2007.

[9] S. Dhevi, A. Subash, and N. Harini, "Security in online games using blockchain," *International Journal of Pure and Applied Mathematics*, vol. 119, pp. 1661–1669, 01 2018.

[10] S. Kalra, R. Sanghi, and M. Dhawan, "Blockchain-based real-time cheat prevention and robustness for multi-player online games," pp. 178–190, 12 2018.

[11] F. Wu, W. Cai, H. Chan, Q. Yan, and H. Yuen, "Proof-of-play: A novel consensus model for blockchain-based peer-to-peer gaming system," 07 2019.

[12] D. Xenakis, I. Zarifis, P. Petrogiannakis, A. Tsiota, and N. Passas, "Blockchain-driven mobile data access towards fully decentralized mobile video trading in 5g networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2020.

[13] "Business of video game cheating.." https://www.fxempire.com/forecasts/article/\\ethereum-2-0-transition-670066\\. Online; Accessed: 2020.

[14] I. Budhiraja, S. Tyagi, S. Tanwar, N. Kumar, and J. Rodrigues, "Diya: Tactile internet driven delay assessment noma-based scheme for d2d communication," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 1–11, 03 2019.

[15] A. Shukla, P. Bhattacharya, S. Tanwar, N. Kumar, and M. Guizani, "Dwara: A deep learning-based dynamic toll pricing scheme for intelligent transportation systems," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.