TPM Vol. 32, No. S5, 2025
ISSN: 1972-6325
https://www.tpmap.org/

SECURE AUTHENTICATION WITH DYNAMIC RESOURCE ALLOCATION IN SOFTWARE-DEFINED NETWORKS

MRS. K. PRIYADHARSHINI
COMPUTER SCIENCE, SRI KRISHNA ARTS AND SCIENCE COLLEGE, BHARATHIAR UNIVERSITY, INDIA.
E-MAIL: priya95joy@gmail.com

DR. S. DEVARAJU
COMPUTER SCIENCE, VIT BHOPAL UNIVERSITY, INDIA.
E-MAIL: devamcet@gmail.com

DR. B. RADHA
COMPUTER SCIENCE, SRI KRISHNA ARTS AND SCIENCE COLLEGE, BHARATHIAR UNIVERSITY, INDIA.
E-MAIL: radhab@skasc.ac.in

Abstract:
This study created a three phase process to reinforce the security, authentication, and link quality management by utilizing new methods for communication flow in a Software-Defined Network (SDN) system. Firstly, this process will strive to improve the Protected Extensible Authentication Protocol (PEAP), an authentication method for SDN, through cryptographic algorithms to strengthen security with constructs of forward secrecy and dynamic key exchange. Secondly, it will also address quality of service (QoS) and use the Weighted Fair Queuing (WFQ) method with the Improved Whale Optimization Algorithm (IWOA) for dynamic management of efficient resource allocation. Thirdly, the study will introduce a brand-new algorithm, Authenticated Utility-aware REsource scheduling with Whale Optimization Algorithm (AUREWOA), composed of authentication functions, utility based scheduling, and adaptive resource allocation in order to succeed maximum performance for the network. Lastly, performance evaluation in the simulations has been shown to have AUREWOA is outperformed compared to other exiting methods across all traffic loads. AUREWOA outperforms in the fundamental measurements: Link Quality Index (LQI), Packet Delivery Ratio (PDR), End-to-End Delay, Throughput, Packet Loss Rate (PLR), and Computation Time while also being viable and stable in scenarios involving traffic overhead. The outcome of the experiment indicates that AUREWOA is scalable, flexible and viable in networks that place constraints on resources; these two priority areas are perfect for dynamic media, and shared real-time applications such as Internet of Things (IoT).

Keywords: Software-Defined Networking, Authentication, Link Quality Management, Source Allocation, Routing Optimization, Quality of Service, Dynamic Scheduling, Internet of Things.

I.INTRODUCTION

The proliferation of connected devices and the development of new technologies such as cloud computing, the Internet of Things, and blockchain have led to increasingly more complex network infrastructures that are required to be less centralized, and distributed. As such, Software-Defined Networking (SDN) has arisen to manage these kinds of environments, where centralized control, dynamic resource allocation, and enhanced programmability are essential [1]. SDN improves the flexibility and management of networks but also introduces new security and access control issues, particularly in heterogeneous and decentralized environments [2].

Access control in an SDN environment will need to manage varying conditions and requirements from platforms such as identity management, ability to enforce policies dynamically, and user anonymity. Many contributions have addressed these issues with new approaches, including both certificate-based and attribute-based approaches for the IoT and cloud environments [3, 4, and 5]. Novel approaches, such as dynamic access control lists [6] and rekeying to maintain secure sessions [7] demonstrate the need for use-specific, adaptive, and context-aware solutions.

At the same time, SDN security itself is heavily intertwined with the design decisions related to control and data planes. Hunts for various use case scenarios have incorporated artificial intelligence (AI) and Moving Target Defense (MTD) to proactively manage menace [8]; see current frameworks around security to manage WoT smart spaces and Security-as-a-Service Situations (SECaaS) that part of the literature which was developing or aims to ultimimately create end-to-end security [9]. Notably, existing work on SDN security highlights the growing problems or vulnerabilities as a possible result of a centralized control point and dynamic reconfiguration, and we have a lot more innovation to pursue regarding SDN security [10].

On the optimizing side, SDN provides the foundations of smart routing, load balancing, and energy consumption management. For example, AI-aided routing, reinforcement learning, and evoluation-aided placement algorithms have been used to optimize performance and Quality of Service (QOS) [11, 12, 13]; see the joint optimization of traffic engineering and resource allocations according to the benefits it gains through inclusion of hybrid SDN assumers [14, 15].

Nonetheless, there are still many open questions concerning the holistic integration of secure, access controlled, and optimized SDN-enabled networks. The existing literature tends to address these areas independently and without collaborative models or frameworks guaranteed to provide security and performance, whilst maintaining systemic flexibility [16, 17]. Therefore, there are strong opportunities available for models that consider AI, cross-layer coordination, and context-aware policy design to meet the constantly-changing demands of SDN in complex networks ecosystems [18, 19, and 20].

Contribution: The main contribution of the paper is the development of a three-phase design to enhance security, authentication, and link quality in the SDN context; it optimizes the existing PEAP protocol through advanced approaches to cryptography, combines WFQ and IWOA for a more advanced QoS mechanism, and devises a new AUREWOA algorithm for intelligent and utility-aware resource allocation and scheduling, along with relevant mechanisms. The vigorous simulation process has demonstrated the better performance of the AUREWOA with respect to different network metrics, which proves it can work effectively in a computationally constrained real-time IoT context.

Organization: The remainder of the paper is organized as follows. Section 2 discusses related work on SDN security, QoS, and optimization approaches. In section 3, we provide the three-phase approach we proposed, the modifications to PEAP, and implementations of WFQ with IWOA and development of AUREWOA. In section 4, we present the experimental setups and simulation parameters, followed by our performance evaluations and comparison in section 5. Finally, in section 6, we summarize our key findings and future research directions.

II. Background Study

2.1 Secure Authentication and Access Control in SDN

Ayedh et al. (2023) [21] provided a systematic review of security access control policies and techniques, and their links to privacy requirements and impacts in Bring Your Own Device (BYOD) scenarios. They categorized access control policies currently staff-ed by institutions and outlined an increasing need for context-aware and user-centred access models. They have clearly identified several important gaps in policy application, and contend that future models need to properly balance usability with security, particularly in terms of personal devices converging on enterprise and commercial in-vehicles networks.

Jiang et al. (2023) [22] proposed FACSC, which is a Fine-Grained Access Control scheme which captures the use of smart contracts for terminal devices in SDN. Their concept integrates a blockchain-based model to offer decentralized enforcement of access control rules, including evidence-based auditability, and tamper-resistance. This research presented a variety of ways for integrating smart contracts to add depth to policy execution and traceability in SDN, especially in trust-sensitive applications.

Barach (2025) [23] proposed a multi-component defense framework towards Zero Trust Security in SDN. This approach integrates continuous authentication, micro-segmentation and adaptive trust policy decisions that are formed across the SDN control and data planes. They identified and emphasized a "never trust, always verify" strategy to move from traditional static, to dynamic and programmable networks, with the aim of remediating the level of insecurity for lateral movements and insider threats.

Vegas and Llamas (2024) [24] investigated the possibilities and limitations of Artificial Intelligence (AI) to automate Identity and Access Management (IAM) systems in operational environments. Not only did their findings explore the potential application of several AI applications to-date, however, the findings also highlighted issues such as bias, explainability and scalability. In their discussion, they offered a hybrid decision model which involved both human and AI oversight, to consider procedural correctness as well as contextual decisions related to access control.

Mishra et al. (2025) [25] surveyed the literature pertaining to SDN-enabled security frameworks for IoT networks. They highlighted multiple challenges which remain unanswered, such as device heterogeneity, low latency, and weak endpoint identifiers. The authors were rather unequivocal regarding an architecture-aware framework which involved categories of light-weight cryptographic protocols evaluated and clear approaches to authorization/management schemas that are sensitive to IoT's resource constrained environment and highly dynamic and possibly unpredictable topology.

Table 1: comparison table on Secure Authentication and Access Control in SDN

Authors & Year	Objective	Technique / Model	Application Area	Unique Contribution
Pradeep et al. (2023) [26]	Improve SDN path security	EnsureS: Lightweight path validation using batch hashing & tag verification	General SDN networks	Efficient, low-overhead service path validation method

Rahdari et al. (2024) [27]	Address SDN-IoT security & privacy challenges	Taxonomy of threats, review of countermeasures	SDN-enabled IoT systems	Comprehensive review + future research directions
Onyema et al. (2022) [28]	Prevent ICMP-based attacks in SDN	Custom security policy protocol	SDN with ICMP traffic	Novel ICMP detection and mitigation scheme
Luo (2024) [29]	Optimize access control in SDN	Intelligent algorithm-based strategy	SDN access control	Adaptive strategy using optimization algorithms
Anitha et al. (2024) [30]	Secure VM communications in cloud-SDN	Role-based shared secret scheme	SDN-cloud environment	Role-aware VM access control for enhanced isolation

2.2 Optimizing Link Quality in SDN

Younus et al. (2020) [31] explored how reinforcement learning (RL) can be applied to optimize the operational lifetime of software-defined wireless sensor networks (SD-WSNs). The proposed algorithm employs RL to model the routing paths and energy usage dynamically based on the environmental features collected from the data, thereby increasing the coherence and resilience of the routing system. This research suggests that RL can be considered a method of applying a RL methodology to the limitations that happen to be intrinsic in the sensor network operational use with a centralized SDN.

Darade et al. (2022) [32] were presenting a load balancing method for the Internet of Vehicles (IoV) utilizing Software-Defined Networking and the Whale Optimization Algorithm while investigating the framework. A dynamic multi-layer networks control/management solution, with SDN utilizing its resources to improve vehicle traffic data distribution was used to enhance a quality of service (QoS) design process. The work adds an excellent angle on service availability for a real world action consisting of sensor transportation operating as a body for E-intercorporation purposes i.e. as related to mobile sensing, as captured by the mobility and delay sensitivity in a vehicular operated IoV scenario.

Akin & Korkmaz (2019) [33] investigated multiple routing algorithms to highlight optimal routing within SDN as distinct from using either static and/or dynamic state information to calculate link costs. The authors demonstrated that new implementations based on dynamic cost-based routing algorithms significantly outperform static routes when the state of the network is in flux. This privileged adaptive routing algorithms as key in timely forwarding of data from superior SDN paradigms.

Shabbir et al. (2020) [34] researched how to improve network performance in low-power lossy networks (LLNs) by using a routing mechanism that implements an SDN-based service-side multiple sinks in IoT architecture. The authors used the programmability of SDN to perform packet loss reduction, throughput and load distribution. The authors state this framework to be used in a common IoT scenario involving efficient data collection or resources.

Kirubasri et al. (2022) [35] took a different focus by summarizing many of the SDN based routing protocols used for ad hoc networks. The aim of this chapter was to assess how well SDN principles can improve routing scalability, adaptability, and fault tolerance in an environment that is generally decentralized. The chapter discusses the implications of SDN in the context of Mobile Ad Hoc Networks (MANETs) or Vehicular Ad Hoc Networks (VANETs), and articulates current design trends and challenges.

Table 2: Comparison table on Optimizing Link Quality in SDN

Authors & Year	**Objective**	**Technique / Model**	**Application Area**	**Unique Contribution**
El-Garoui et al. (2020) [36]	Reduce delay in smart city networks	SDN-based custom routing protocol	Smart city IoT networks	Improved delay performance in urban environments
Hussein et al. (2024) [37]	Review SDN-VANET routing architectures	Comprehensive survey & taxonomy	SDN-based vehicular networks	In-depth analysis and future research directions
Rabet et al. (2022) [38]	Manage mobility in IoT via SDN	SDMob: Mobility-aware management framework	IoT networks	Seamless mobility support using SDN principles
Chen et al. (2022) [39]	Optimize QoS via AI in SDN	AQMDRL: Deep RL-based QoS control	General SDN networks	Multistep deep reinforcement learning for dynamic QoS

Yan et al. (2021) [40]	Enhance QoS in tactical Ad Hoc networks	Controller deployment strategy	Tactical SDN-Ad Hoc networks	Strategic SDN controller placement for guaranteed QoS

2.3 Problem Identification

Although SDN has progressed in terms of secure access control and network optimization, some challenges remain. Existing access control models typically constitute a relatively static model of the environment and struggle to adapt to the dynamic and complex environments of IoT and vehicular networks (Bhamare et al., 2023). Other problems include issues around privacy, scalability, and integration with technologies like blockchain is yet to be solved. Routing and mobility protocols may also need greater testing in the highly variable conditions of mobile or ad hoc networks where latency and energy limitations must be considered. Designs based on AIs offer promise, but there are concerns about real-time applications and AI workloads. In all, we can likely expect SDN systems in the future, to specify a framework for intelligent, integrated security, privacy, and access control/ performance optimization that can work in a dynamic manner.

III.MATERIALS AND METHODS

This research provides a multi-phase approach for enhancing security, authentication, and link quality management in SDN environments. In Phase 1, we improved the authentication for the user and equipment using a robust Protected Extensible Authentication Protocol (PEAP). We enhanced PEAP to include numerous new cryptographic enhancements that solicited additional assurance of design in PEAP and protected the identification between devices and controllers. These enhancements include, forward secrecy, identity based encryption, and a robust dynamic key exchange method. This enhanced PEAP and provides mutual authentication and control to various policy levels between the devices and controllers. Phase 2 is a continuation from this as we began developing a Hybrid Scheduling and Optimization model. This model is a combination of Weighted Fair Queuing (WFQ) and an Improved Whale Optimization Algorithm (IWOA). The input to the Hybrid is real-time statistics that were used in scheduling the flow weights which are routinely updated according to multiple measures affecting performance for various traffic scenarios, so that Quality of Service (QoS) can be achieved. Finally, in phase 3, our IWOA advanced with an AUREWOA/Authenticated Utility-aware Resource scheduling initiative, utilizing authentication, utility-based scoring, and fitness proposition. Additionally, AUREWOA updates dynamically for resource allocation and routing decisions based on metrics (ex. packet loss rate, round-trip time, etc), throughput score, and flow priority for optimization.

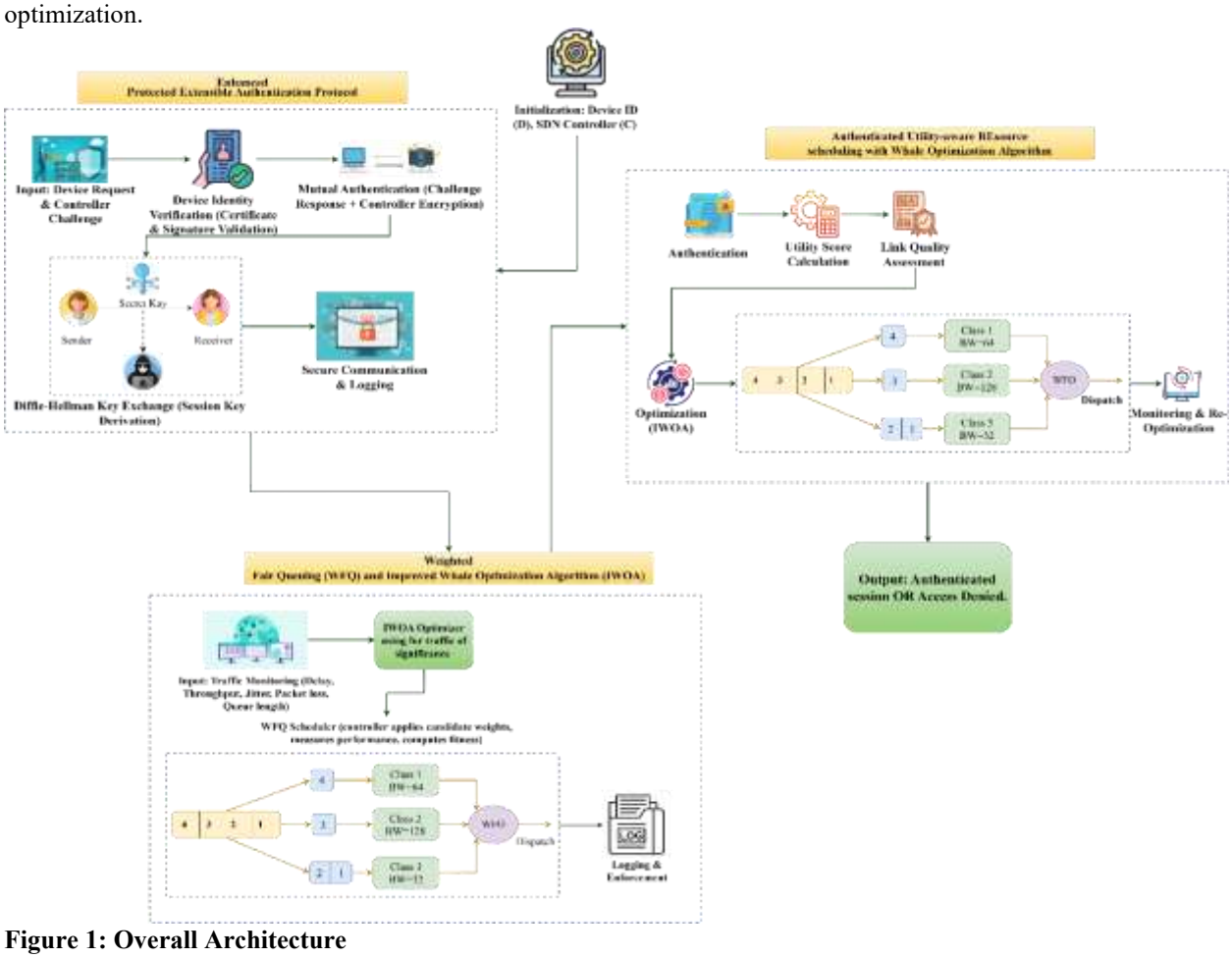

Figure 1: Overall Architecture

The above figure 1 is a depiction of a secure authentication and resource scheduling architecture utilizing Enhanced Protected Extensible Authentication Protocol (EAP-PEAP) and an Improved Whale Optimization Algorithm (IWOA) to maximize the allocation of network resources. The architecture initiates device authentication with certificate and signature verification, followed by mutual authentication and associated key exchange. After authentication, the system conducted Utility Score and Link Quality evaluations and classify the network traffic into priority queues by using Weighted Fair Queuing (WFQ). The IWOA optimizer will dynamically allocate resources to the traffic according to their significance, allocating the resources based on load balancing of the network classes. The system will either give an authenticated session or deny access based on security and performance assessment.

3.1 Enhanced Protected Extensible Authentication Protocol

While Phase 2 is focused on efforts to enhance the Protected Extensible Authentication Protocol (PEAP) to create a more secure and reliable option for authentication framework establishment for Software Defined Networking (SDN) environments, the improvements made to the PEAP process allow for improved usability as well. Used predominately for wireless and enterprise networks, PEAP provides an additional measure of security for authentication exchanges as it encapsulates the Extensible Authentication Protocol (EAP) in a secure Transport Layer Security (TLS) tunnel. However, in SDN infrastructures that are highly dynamic, the control plane can be programmatic and heterogenous access control mechanisms provide security concerns that utilize PEAP processes that are standard in nature due to the complexity of the dynamic network environment and unique challenges of the SDN environment.

Thus, Phase 2 introduces PEAP-based trust mechanisms that improve upon the original PEAP that include additional cryptographic security layers including forward secrecy (which prevents unauthorised access to devices by holding past keys,) dynamic key exchange mechanisms, and identity-based encryption in the authentication session to ultimately develop or modify standard sensitive authentication sessions to protect against replay attacks, man-in-the-middle intrusions, and session hijacking attacks. The advanced PEAP developed in Phase 2 allows for mutual authentication of both the user and controller credential verification processes, and this is critical to identify rejecting opportunity access to rogue devices from a malicious actor. The added layer of security will allow for mutual authentication for user and controller credit again significantly reduces security risk from rogue access control devices, and policies will be able for fine-grained monitoring to support a better contextual awareness of user and device level authorization.

This approach is efficient enough for performance-sensitive applications as it adds little overhead to the overall effort of securely managing resources. Within SDN architecture, the enhanced PEAP is located at the control layer, where its Pol-DIAC functionality restricts access to network functions (programming) and resources (services) based on identity verification and policy compliance. Real-time logging and anomaly detection components are also introduced into the model to indicate suspicious activity and provide forensic evidence.

This complete overhaul of PEAP represents a flexible and scalable approach to secure authentication in distributed, software-defined environments, particularly regarding connectivity to IoT or cloud-based infrastructure, and vehicular networks. The proposed model attempts to standardize and thus, automate secure onboarding processes, while maintaining interoperability from different SDN controllers or across other network domains. While the enhanced PEAP system not only increases security through authentication, it promotes trust along with efficiency in orchestrating programmable networks.

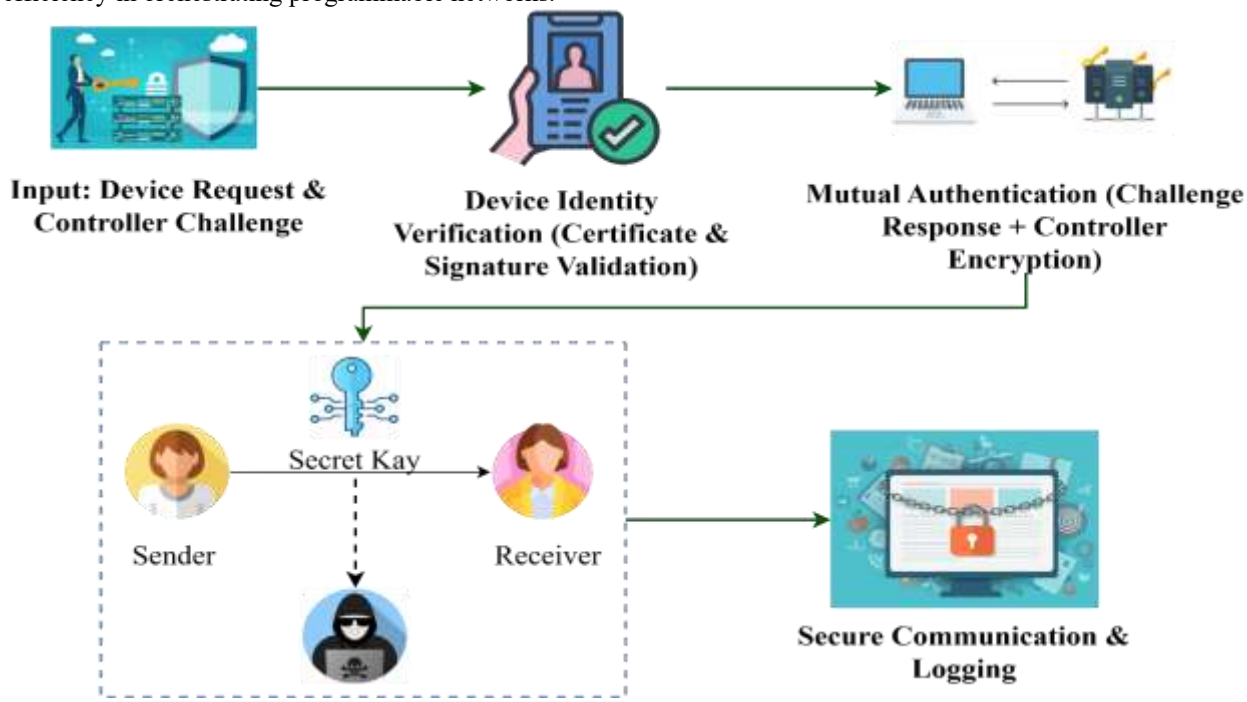

Figure 2: Architecture of Enhanced Protected Extensible Authentication Protocol

This figure 2 displays a secure authentication and communication process between a device and a controller. The process begins with the device sending a request which will yield a challenge from the controller. The device's identity is established through certificates and signature verification. Next, mutual authentication is established through a challenge-response process and encrypting the data. The secret key is then established to enable a secure channel of communication between the sender and the receiver, while preventing access from anyone who is unauthorized. Every part of the communication is stored securely so it maintains integrity and traceability.

$M_A = Enc_{Pub_B}(N_A||ID_A)$ ---------- (1)

Equation (1) denotes the encryption of a nonce N_A and identifier ID_A generated by entity A, using the public key of entity B, so that only B can decrypt it. This promotes trust and identity verification and protects from eavesdropping on the message.

$K_{session} = (g^b)^a$ ---------- (2)

Equation (2) illustrates the derivation of the shared session key $K_{session}$ using the Diffie-Hellman key exchange approach. It guarantees that both parties, having exchanged public values, compute the same key independently, thus securing the communication without directly sending the key.

$T_i = \sum_{j=1}^{n} W_j \cdot M_{ij}$ ---------- (3)

Algorithm 1: Enhanced Protected Extensible Authentication Protocol

```
Input:
   D → Device requesting access
   C → SDN Controller with Auth Server
   PubD, PrivD → Public/Private key pair of D
   PubC, PrivC → Public/Private key pair of C
   Policy Threshold, Trust Weights
Output:
   Authenticated Session or Access Denied
Begin
   // Step 1: Initialization
   D → C: Access Request
   N_C ← GenerateNonce()
   C → D: {N_C, Cert C}
   // Step 2: Device Identity Verification
   if VerifyCertificate(Cert C) == False then
      return Access Denied
   end if
   M1 ← Sign(Priv D, N_C || ID_D)
   D → C: {M1, Cert D}
   // Step 3: Mutual Authentication
   if VerifySignature(M1, Pub D) == False or VerifyCertificate(Cert D) == False then
      return Access Denied
   end if
   N_D ← GenerateNonce()
   M2 ← Encrypt(Pub D, N_D || ID_C)
   C → D: M2
   // Step 4: Diffie-Hellman Key Exchange
   (g, p) ← PublicParameters()
   a ← RandomInteger()
   b ← RandomInteger()
   A ← (g^a) mod p
   B ← (g^b) mod p
   D → C: A
   C → D: B
   K_session ← (B^a) mod p  // Device
   K_session ← (A^b) mod p  // Controller
   // Step 5: Trust Score Evaluation
   Trust_Score ← 0
   for each metric j in Trust_Metrics do
      Trust_Score ← Trust_Score + (Trust_Weights[j] * Measure(D, j))
   end for
   if Trust_Score < Policy_Threshold then
      return Access_Denied
   end if
   // Step 6: Establish Encrypted PEAP Tunnel
   PEAP_Tunnel ← EstablishTunnel(K_session)
```

```
  LogSession(D, Trust_Score, Status = "Authenticated")
  return Authenticated_Session
End
```

Algorithm 1: Modified Protected Extensible Authentication Protocol highlights a way of authenticating and controlling access to devices in Software-Defined Network (SDN) environments. The authentication process starts with a mutual authentication exchange of the device, and SDN controller, using certificates and digital signatures to establish the trust. Each party will verify their identity and use Diffie Hellman key exchange to initialize a secure session key in order to create a secure session. The trust score is derived from the metrics and relevant weights from a policy to show whether the device met the access threshold. If a trust scores acceptable to the SDN controller is developed, an encrypted PEAP tunnel would be initiated, securing all communications thereafter. The session would have been logged for auditing purposes an access was granted to that device. Where the trust score was insufficient Privileges would be denied.

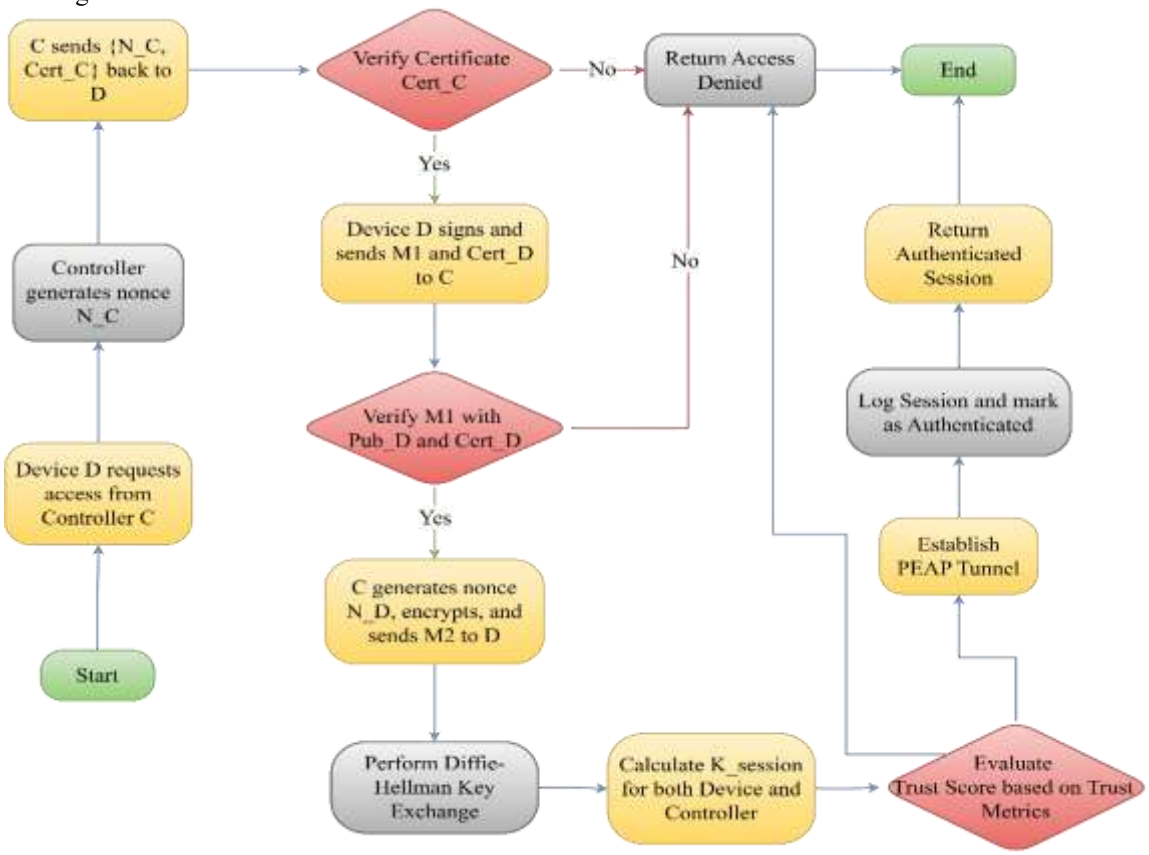

Figure 3: Flow Chart of Enhanced Protected Extensible Authentication Protocol

This figure 3 outlines a secure authentication protocol between a device (D) and a controller (C) based on certificate verification and key exchange mechanisms. The protocol begins with a request from device D for access to the controller. In response, the controller issues a nonce to the device D, and a certificate is also issued. The device D and controller C will first check the signatures and certificates for both party's legitimacy, in which access to the device D is immediately denied if either check fails. Once the host and device D are correctly verified, the protocol carries out a Diffie-Hellman key exchange to develop the session key, after which the PEAP tunnel has been opened for secure communication. At the end of the session, the system creates a log of the session history, decides a trust score based on various metrics, and lastly will redirect the device D once again to the authenticated session state for secure data exchange.

3.2 Weighted Fair Queuing (WFQ) and Improved Whale Optimization Algorithm (IWOA)

Phase 3 aims to enhance link quality in SDN through the integration of Weighted Fair Queuing (WFQ) to allocate bandwidth to flows and an Improved Whale Optimization Algorithm (IWOA). WFQ is a scheduling algorithm and was intending to fairly allocate bandwidth to flows competing for bandwidth, based on their assigned weights to not only represent the application's priority level but also any Quality of Service (QoS) demands. The WFQ algorithm does a good job of differentiating traffic of significance in its scheduling while scheduling lower priority traffic, which will help eliminate delays and drops for the application of importance. The continuing limitation with WFQ is that it is a static weight assigned to the flows. So when we have dynamic network conditions such as changing loads to many flows in a network or changes in mobility in either vehicular or IoT networks, these static weights assigned may not give the optimal solution. The Improved Whale Optimization Algorithm (IWOA) dynamically adjusts scheduling weights and resource allocations in real time.WOA is an evolutionary meta-heuristic algorithm that incorporates weight coefficients to improve model effectiveness but not to the level of adaptability as IWOA.

Thus, IWOA is an improvement over WOA that incorporates adaptive weighting coefficients, but also incorporates chaotic maps to improve exploration and converging strategies to enhance both speed and accuracy of solution.

In the SDN controller, IWOA kept track of network state metrics (e.g., latency, queue length, throughput, and jitter) to adapt these WFQ parameters for optimal flow-level performance. In addition, this hybrid framework enables both improved congestion control with intelligent resource provision and better scalable performance during variable traffic demand. Thus, this WFQ-IWOA scheme is a more responsive and context-aware framework to optimize link quality in SDN, particularly in situations where real-time traffic classification, load-balancing, and fairness are necessary. More importantly, as this framework is applicable to centralized SDN controllers, we can use this framework for orchestration of diverse network segments, so instantiating this solution as part of an overall central controller solution is made possible. Overall, this has the potential to lead to more stable, efficient, and QoS-compliant SDN implementations consistent with future requirements by next-generation applications like video streaming, real-time, and autonomous applications.

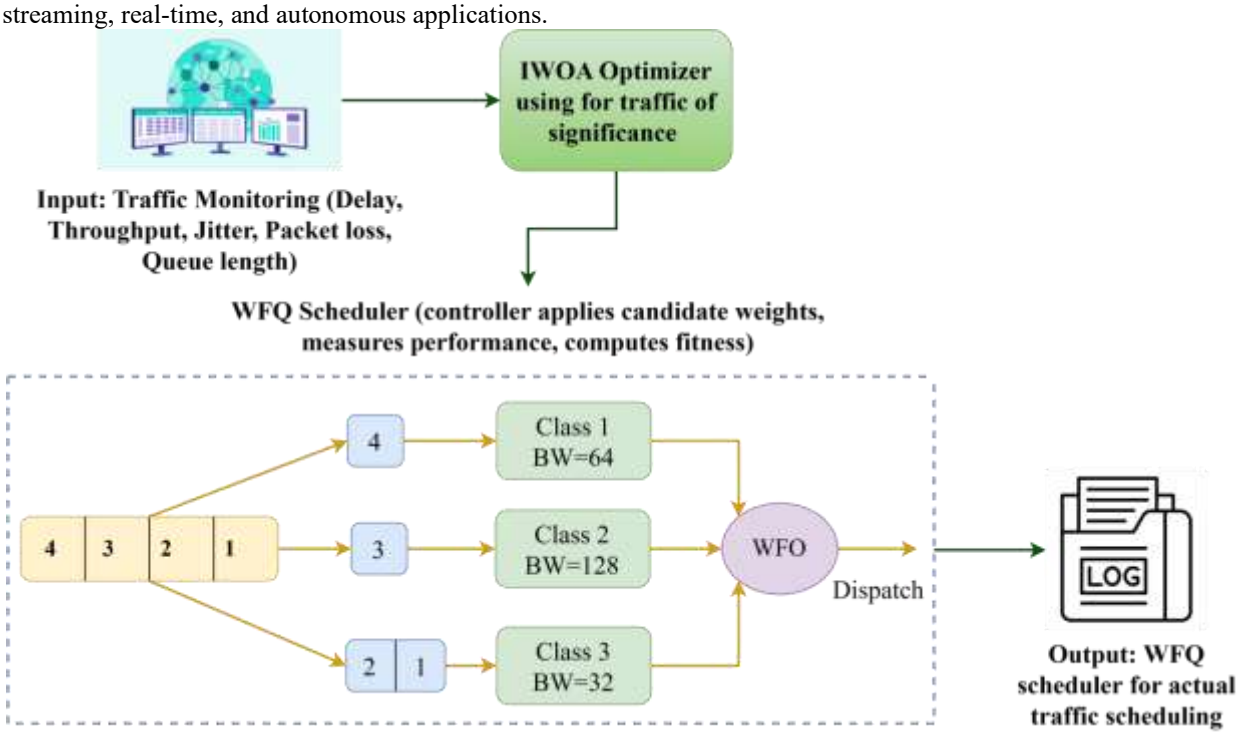

Figure 4: Architecture of Weighted Fair Queuing (WFQ) and Improved Whale Optimization Algorithm (IWOA)

The figure 4 outlines the incorporation of an Improved Whale Optimization Algorithm (IWOA), which will assess incoming traffic streams and prioritize them based on their importance, consequently assigning different priority classes (Class 1, 2, 3) to the incoming traffic streams. It is an approach to resource contention with predefined bandwidth allocations (64, 128 and 32) while also allowing an instantaneous classification function. The traffic is subsequently scheduled according to Weighted Fair Queuing, allowing an equitable portion of the consume bandwidth while maintaining the higher level of traffic priority, as prescribed in the previous module. The weighted forwarding optimization (WFO) module controls the traffic forwarding decision while also logging the outcomes for future monitoring and optimization decision making. This approach optimizes the balance of efficient bandwidth use while allocating resources to support the most important network-based traffic.

$$F_i^k = max\left(F_i^{k-1}, V(a_i^k)\right) + \frac{L_i^k}{w_i} \text{ ---------- (4)}$$

Equation (4) calculates the finish time F_i^k of the k-th packet in flow i under Weighted Fair Queuing (WFQ). It guarantees fair scheduling by taking the maximum of the last packet's finish time and the virtual arrival time, and adding the normalized packet size based on its weight w_i .

$$Fitness = \alpha \cdot \frac{1}{Delay} + \beta \cdot \frac{Throughtput}{MaxThroughtput} - \gamma \cdot PacketLoss \text{ ---------- (5)}$$

The fitness function, defined by Equation (5) in this example, indicates that a higher fitness represents better network quality of performance as the optimization function, which takes into consideration the delay, throughput, and packet loss metrics, produces a fitness value of relative importance based on weights α, β, γ for the SDN resource allocation, based on low delay, high throughput, and minimal packet loss.

$$\vec{X}(t+1) = \vec{X}^*(t) - A \cdot |C \cdot \vec{X}^*(t) - \vec{X}(t)| \text{ ---------- (6)}$$

Equation (6) represents the position update rule in the Improved Whale Optimization Algorithm (IWOA). The solution vector $\vec{X}(t+1)$ is updated towards the best-known solution $\vec{X}^*(t)$, with coefficients A and C controlling exploration and convergence in the search space.

Algorithm 2: Weighted Fair Queuing and Improved Whale Optimization Algorithm

```
Input:
   F ← Set of network flows {f1, f2, ..., fn}
   T ← Maximum number of iterations
   P ← Population size (number of whales)
Output:
   W_best ← Optimized WFQ weight vector
Begin
   Initialize whale population W[1...P] with random weights for each flow
   Set a ← 2, α, β, γ ← weight coefficients for fitness function
   For t = 1 to T do
      For i = 1 to P do
         Apply weights W[i] to WFQ scheduler
         Simulate or monitor: Delay_i, Throughput_i, PacketLoss_i
         Fitness[i] ← α / Delay_i + β * (Throughput_i / MaxThroughput) - γ * PacketLoss_i
      End For
      W_best ← W[i] with max(Fitness[i])
      For i = 1 to P do
         r ← Random(0,1)
         A ← 2 * a * r - a
         C ← 2 * r
         D ← abs(C * W_best - W[i])
         W[i] ← W_best - A * D
         Normalize W[i] within valid range [W_min, W_max]
      End For
      a ← a - (2 / T)  // Linearly decrease 'a'
   End For
   Return W_best
End
```

In SDN, algorithm 2: WFQ-IWOA optimizes the link quality of the network by varying the weights assigned to the Weighted Fair Queuing (WFQ) scheduler using an Improved Whale Optimization Algorithm (IWOA). The service is encapsulated in a basic structure of population initialization, in which populations of weight vectors (whales) are initialized and their performance evaluated using a fitness function based on delay, throughput, and packet loss. In each iteration, each whale position (weight vector) is subsequently updated with whale behaviors. Specifically, the whales "encircle" the best-known solution(s) using coefficients A and C to keep the search for weights proximity near the best-performing weight. Each subsequent iteration refines the best-performing weight vector as a is decreased for consistent discovery of the best weight for the WFQ region of a SDN. In the end, this algorithm will return the best set of WFQ weights to optimize fairness, minimize congestion, and support Quality of Service (QoS) in SDN environments.

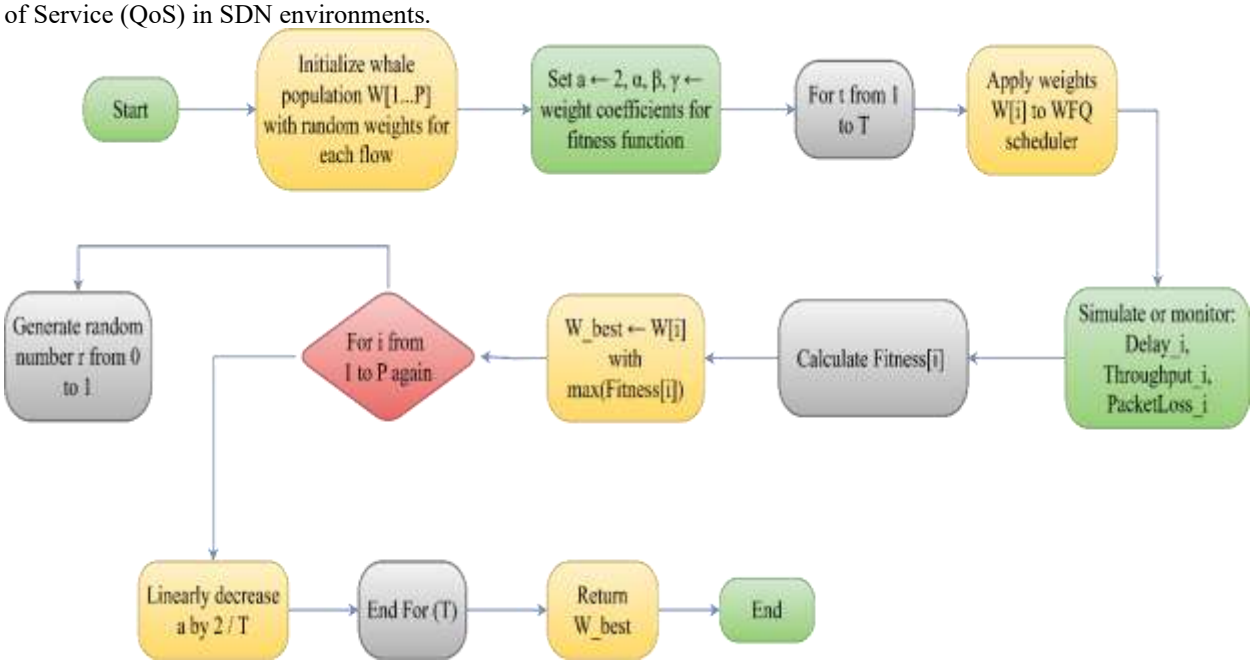

Figure 5: Flow Chart of Weighted Fair Queuing (WFQ) and Improved Whale Optimization Algorithm (IWOA)

This figure 5 shows an optimization process using a whale optimization algorithm (WOA) for the optimal weights for flow of traffic. The first step is to determine a population of weight solutions and then define coefficients (α, β, γ) for the fitness function. Secondly, the system evaluates the fitness of each candidate solution based on delay, throughput, and packet loss and will iteratively update weights until a set of optimal weights is defined. The algorithm will select the best weights and will have a given number of random and decreasing parameter for exploration and exploitation. Lastly, once a set of optimal weights is found, the WOA algorithm will apply these to a Weighted Fair Queuing (WFQ) scheduler to balance the network performance.

3.3 Authenticated Utility-aware REsource scheduling with Whale Optimization Algorithm (AUREWOA)

AUREWOA (Authenticated Utility-aware REsource scheduling with Whale Optimization Algorithm) is a new algorithm for enhanced link quality management in SDN environments. AUREWOA tackles significant issues surrounding SDN authentication, QoS-aware scheduling, and Whale Optimization Algorithm (WOA) authentication and scheduling by sequentially implementing authentication, scheduling, and optimizing in a coherent manner which addresses SDN concerns such as secure flow establishment, resource allocation decisions, and link resource assessments dynamically, in real-time.

First, AUREWOA's process begins with authenticating secure flows by a lightweight and extensible manner. The authentication process serves as a way for entering trusted devices and secure flows. Once completing the authentication phase, flows are adjudged on their utility score which factors in, for example, bandwidth desirability, application latency, and importance of application. Flows are queued into active, priority-based queues, leveraged from principles of Weighted Fair Queuing (WFQ) to manage delay via bandwidth fair allocations.

The optimization engine is based on the Improved Whale Optimization Algorithm (IWOA) with added features of adaptive encircling and chaotic behavior to avoid local optimums and promote faster convergence. IWOA selects optimal links dynamically, continually measuring the quality of the links through metrics such as packet loss rate, throughput, and Round Trip Time (RTT). This optimization loop describes how the SDN controller can make intelligent decisions in real-time.

This includes flow rerouting, load balancing, and congestion management, for efficient network forwarding. AUREWOA's utility-aware scheduling will use these links and contain critical flows (for example; healthcare or emergency data) while not starving lower priority traffic. AUREWOA also adaptively provides feedback to assign resources, upon receiving fluctuations to link quality updates in its next decision. Therefore, AUREWOA's integrated approach provides programmability and adaptiveness to the SDN while maximizing the use of security, efficiency and good service quality in a dynamic network environment.

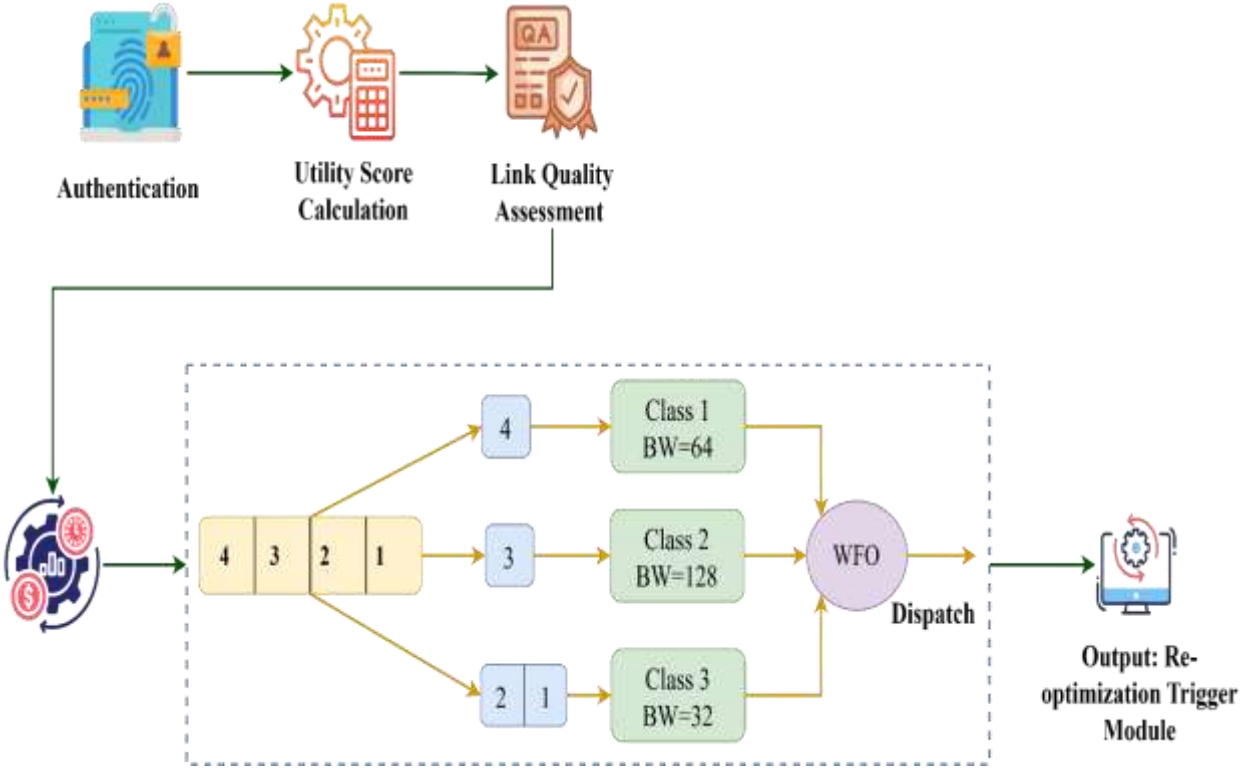

Figure 6: Architecture of Authenticated Utility-aware REsource scheduling with Whale Optimization Algorithm

This figure 6 depicts a flow management that performs authentication on incoming requests. After the authentication, a utility score is given on the flow followed by assessing link quality for possible routing of that flow. Next flows are placed in one of three classes based on priority or need, with predetermined bandwidths of 64, 128, 32 units respectively. Next the classes are annually processed through a Weighted Fair Optimization (WFO) module that schedules and dispatches the flows fairly. Finally, an output re-optimization trigger module monitors

performance issues and triggers re-optimization only if a performance problem is detected and mitigation through re-optimization is warranted.

$$U_i = w_1 \cdot P_i + w_2 \cdot \frac{1}{D_i+\epsilon} + w_3 \cdot B_i \text{ ---------- (7)}$$

Equation (7) calculates the Utility Score U_i for each flow i as the combination of its priority (P_i), delay sensitivity (D_i), and bandwidth consumption (B_i). The weights w_1, w_2, w_3 control the contribution of each component in the score, while ϵ humorously ensures we don't mistakenly divide by zero for delay sensitive flows.

$$LQI_{ij} = \alpha \cdot \left(1 - \frac{PLR_{ij}}{100}\right) + \beta \cdot \frac{TP_{ij}}{TP_{max}} + \gamma \cdot \left(1 - \frac{RTT_{ij}}{RTT_{max}}\right) \text{ ---------- (8)}$$

Equation (8) defines the Link Quality Index (LQI_{ij}), which measures the quality of the link between nodes i and j, based on the packet loss rate (PLR), throughput (TP), and round-trip time (RTT). The coefficients α, β, γ indicate how much each metric matters, allowing a fair assessment of reliability, efficiency, and latency.

$$F_i = \lambda_1 \cdot LQI_{ij} + \lambda_2 \cdot \left(1 - \frac{Q_{ij}}{Q_{max}}\right) + \lambda_3 \cdot U_i \text{ ---------- (9)}$$

Function (9) represents the Fitness Function F_i applied in the optimization procedure to assess candidate links or flows as possible flows for flow i. The function integrates link quality (LQI_{ij}), level of congestion (Q_{ij}), and utility score (U_i) with weights $\lambda_1, \lambda_2, \lambda_3$ to support the selection of the most optimal routing path that is secured and has the highest fitness generation performance.

$$\vec{X}(t-1) = \vec{X}^*(t) - A \cdot |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \text{ ---------- (10)}$$

The position update rule in the Improved Whale Optimization Algorithm (IWOA) is defined in Equation (10). The current solution $\vec{X}(t)$ moves towards the best solution $\vec{X}^*(t)$. The coefficients A and C handle the typical exploration and exploitation behaviors of the agent to dynamically control the selected link/path that would allow an optimal SDN performance.

$$w_i = \frac{U_i}{\sum_{j=1}^{n} U_j} \text{ ---------- (11)}$$

Equation (11) computes the normalized weight w_i of each flow i based on its utility score U_i. This weight represents the fair share of resources (bandwidth, etc.) each flow is allocated within the Weighted Fair Queuing (WFQ) mechanism by considering the utility of each flow with respect to the total utility of all flows.

Algorithm 3: Authenticated Utility-aware REsource scheduling with Whale Optimization Algorithm

Input:
  Flow requests F = {f1, f2, ..., fn}
  Network topology G(N, L)
  QoS parameters: P_i, D_i, B_i
  Link metrics: PLR_{ij}, TP_{ij}, RTT_{ij}, Q_{ij}
  IWOA parameters: max_iterations, whale_count, a ∈ [2, 0]
Output:
  Optimized routing paths with authenticated and scheduled flows
Begin
Initialize system weights: ω1, ω2, ω3, λ1, λ2, λ3, α, β, γ
Initialize IWOA parameters and population of whale positions (candidate paths)
For each incoming flow fi ∈ F do:
  Perform PEAP-based authentication
  If authentication fails:
    Drop fi
  Else:
    Proceed to utility calculation
Compute Utility Score for fi:
  $U_i \leftarrow \omega1 * P_i + \omega2 * (1 / (D_i + \varepsilon)) + \omega3 * B_i$ // Eq (7)
For each candidate path/link (i, j) in G do:
  Compute LQI_{ij}:
    $LQI_{ij} \leftarrow \alpha * (1 - PLR_{ij}/100) + \beta * (TP_{ij}/TP_{max}) + \gamma * (1 - RTT_{ij}/RTT_{max})$ // Eq (8)
Compute Fitness for each candidate path:
  $F_i \leftarrow \lambda1 * LQI_{ij} + \lambda2 * (1 - Q_{ij}/Q_{max}) + \lambda3 * U_i$ // Eq (9)
Apply Improved Whale Optimization Algorithm (IWOA):
  For t = 1 to max_iterations do:
    For each whale solution Xi:
      A ← 2 * a * rand() - a
      C ← 2 * rand()
      X_new ← X_best - A * |C * X_best - X_current| // Eq (10)
      Evaluate fitness F_i for X_new
      If F_i (X_new) > F_i (X_best):
        X_best ← X_new

```
    a ← Linearly decrease a from 2 to 0
After convergence:
  Assign optimized path from X_best to flow fi
Perform Weighted Fair Queuing (WFQ) Scheduling:
  Compute normalized weight:
    w_i ← U_i / ∑( U_j) for j = 1 to n      // Eq (11)
  Allocate bandwidth: R_i ← w_i * R_total
Install flow rules and QoS policies on SDN switches
Monitor link performance metrics periodically:
  If degradation detected:
    Re-run optimization from LQI evaluation step
End Algorithm
```

The AUREWOA algorithm 3 encompasses secure flow authentication, utility-aware resource scheduling, and link optimization for Software-Defined Networking (SDN). The first step in the AUREWOA algorithm is to authenticate every new flow using the Enhanced PEAP method (Extensible Authentication Protocol). The Enhanced PEAP method ensures that only trusted sources are included in the processed traffic. Once authenticated flow is confirmed, a utility score is created for every authenticated flow. A flow's utility score uses key parameters: priority, delay sensitivity, and demand for bandwidth. The Link Quality Index (LQI) is created (continued) using maximum available throughput, round-trip time, and packet loss ratio to measure the quality of a link. The utility score and other parameters are input into the fitness function to provide an environmental condition for the Improved Whale Optimization Algorithm (IWOA), which is processed until the flow selection is fully optimized. Finally, Weighted Fair Queuing (WFQ) allocates a proportionate amount of bandwidth to each flow concerning the utility score, enabling flows with secure, efficient, and utility-aware transmission.

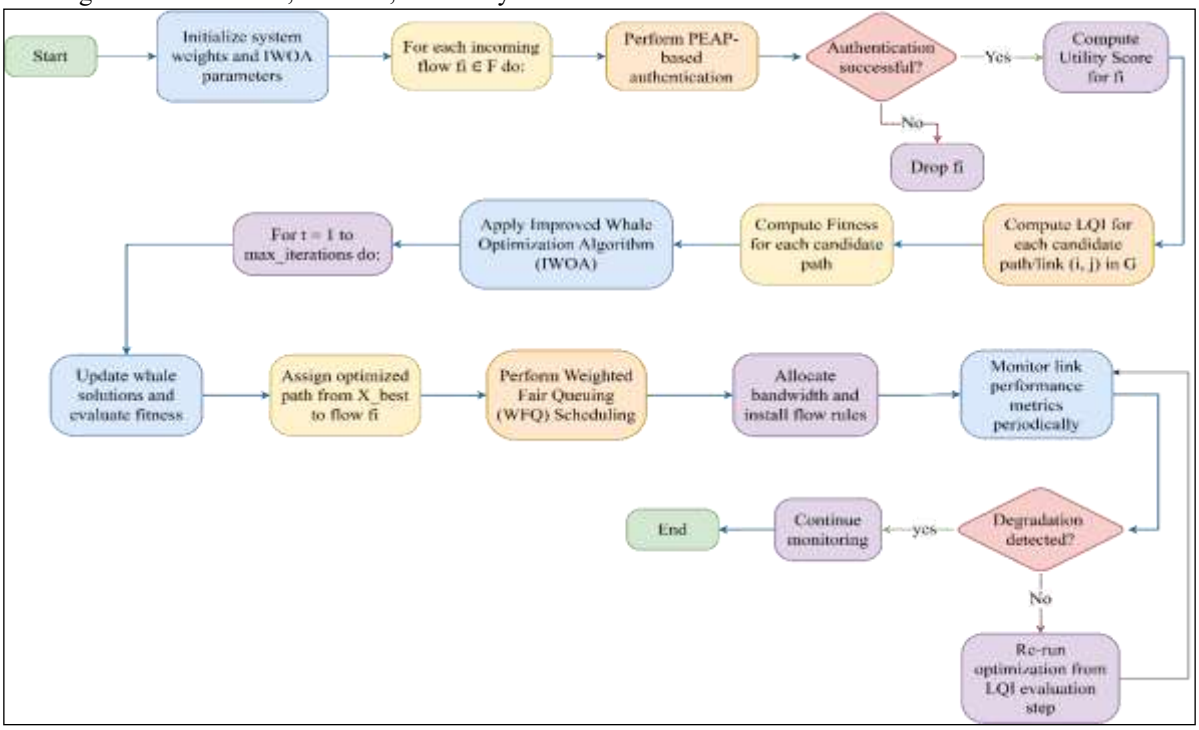

Figure 7: Flow Chart of Authenticated Utility-aware REsource scheduling with Whale Optimization Algorithm

The figure 7 shows a secure and optimized flow scheduling system with PEAP authentication, Improved Whale Optimization Algorithm (IWOA), and Weighted Fair Queuing (WFQ). The flowchart starts by setting the weights for the system and the IWOA parameters. Each flow that arrives into the system is authenticated using PEAP to determine its legitimacy. All flows that are authenticated will then have their utility score calculated and take the routes as candidates which have the assessed metrics such as LQI and are compared from the fitness functions. The IWOA will go through several iterations to develop the assessment for the best route in terms of user utility. The WFQ schedule will then assure that the bandwidth is fairly distributed to each flow and implement the flow rules. The system continuously staves off negative progress in the network by monitoring the existing flows in terms of characterization of the performance level achieved. If declining performance is imminent, re-optimization will be invoked while testing for performance and security.

IV.RESULT AND DISCUSSION

This section has a complete performance assessment of the proposed AUREWOA algorithm. This section compares AUREWOA with six existing routing optimization methods: Expected Transmission Count (ETX), Expected Lifetime Packet (ELP), Packet Reception Ratio (PRR), Fuzzy Logic-based Enhanced Particle Swarm Optimization (FL-EPSO), PEAP, and WFQ-IWOA. The performance evaluation of AUREWOA was conducted with various network traffic loads ranging from 50 to 250. The evaluation where the network traffic loads simulated congestion, meaning the wireless network has to deal with realism that is increasingly congested. The performance measures used with AUREWOA will contain Link Quality Index (LQI), Packet Delivery Ratio (PDR), End-to-End Delay, Throughput, Packet Loss Rate (PLR), and Computation Time since these performance measures capture the good network behavior the proposed framework is intended to have. The results of AUREWOA are presented in a series of summary (comparative) tabular results and corresponding summary (comparative) graphs for each of the performance measures. All of the tables imply their performance of routing scalability, efficiency, and robustness when dealing with higher traffic loads, where AUREWOA performs extremely well as it achieves similar positively, while maintaining good and positive network performance. Based on the defined metrics AUREWOA provides reliable, low-latency, highly-throughput communication, with less packet loss and less computation overhead compared to the two baselines compared approaches (1) and two state-of-the-art compared approaches (2).

Table 3: Comparison table on Link Quality Index (LQI)

Traffic Load	ETX [41]	ELP [42]	PRR [43]	FL-EPSO	PEAP	WFQ-IWOA	AUREWOA (Proposed)
50	0.72	0.76	0.78	0.82	0.84	0.88	0.93
100	0.68	0.73	0.76	0.81	0.83	0.86	0.91
150	0.65	0.70	0.74	0.79	0.81	0.84	0.89
200	0.60	0.66	0.70	0.75	0.77	0.81	0.87
250	0.55	0.62	0.66	0.71	0.73	0.78	0.84

Table 3, is shown across varying levels of traffic load. It can be observed that the proposed AUREWOA algorithm has performed consistently better than all other algorithms tracked. As shown in the provided figure corresponding to LQI values, traffic loads moving from 50 to 250 grew, the value of LQI typically declined for each of the listened algorithms. The decline in LQI is expected due to the burden higher traffic loads have on link stability and reliability. AUREWOA maintained a consistently higher value of Lqi at each traffic level and only went from 0.93 to 0.84, but remained higher than each other algorithm. This suggested that AUREWOA's combination on routing selection, transmission power adjustment, and node optimization is better able to ensure stable and efficient communication links, even under extreme influences and conditions imposed by network traffic. Overall performance improvement seen in LQI reflects AUREWOA's stability regarding interference presence and ability to maintain connectivity above typical levels.

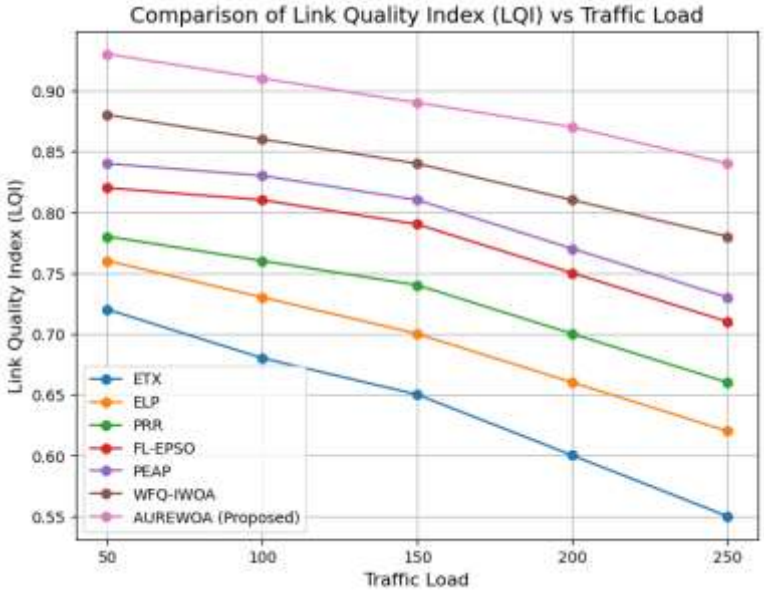

Figure 8: Link Quality Index (LQI) Comparison Chart

The figure 8 compares Link Quality Index (LQI) for various methods (ETX, ELP, PRR, FL-EPSO, PEAP, WFQ-IWOA, and AUREWOA). While all methods show a decrease in LQI as traffic equals or exceeds the specified amount, the AUREWOA approach provides the highest overall LQI independent of the traffic level. When the traffic load reached its maximum level of 250, AUREWOA maintained a LQI value above 0.84 while all other methods dropped below 0.56. Although the network will degrade based on different traffic loads, this performance

measure all but guarantees that AUREWOA will provide the most reliable and stable link quality among the algorithms presented.

Table 4: Comparison table on Packet Delivery Ratio (PDR)

Traffic Load	ETX [41]	ELP [42]	PRR [43]	FL-EPSO	PEAP	WFQ-IWOA	AUREWOA (Proposed)
50	91.2	92.6	93.5	95.1	96.0	97.4	98.6
100	88.7	90.5	92.0	94.2	95.3	96.6	98.1
150	85.4	88.3	90.1	92.7	94.1	95.5	97.5
200	82.0	85.0	87.2	90.5	92.2	94.3	96.7
250	78.6	81.9	84.5	88.0	90.1	93.1	95.9

Table 4 is Packet Delivery Ratio (PDR) comparisons showed that AUREWOA has the highest delivery success rate in each of the heavy traffic loads. All algorithms show a gradual decrease to the PDR at a higher load of traffic; which shows that there was increased congestion and interference, while AUREWOA still stands firm, showing higher resilience. For example look at the heaviest load, at 250, AUREWOA achieved a PDR of 95.9%, while the ETX addition was 78.6%. What this shows, is that AUREWOA is efficient and effective at providing reliable packet transmissions even under stress, due to the level of service and quality of forwarding performance with dynamic route management and traffic adaptive control capabilities, by reducing packet loss and delivering high reliability.

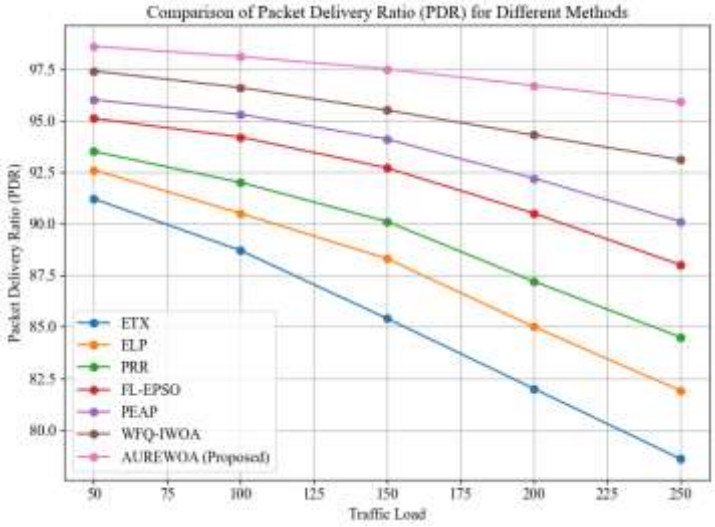

Figure 9: Packet Delivery Ratio (PDR) Comparison Chart

This figure 9 compares the packet delivery ratio (PDR) of different methods ETX, ELP, PRR, FL-EPSO, PEAP, WFQ-IWOA, and the proposed AUREWOA at increasing traffic load. Again, we see that as we increase traffic load within established parameters from two nodes routing to a maximum of three hundred nodes routing, PDR decreases for all algorithms. We see that AUREWOA has the highest PDR each time load is increased. At the highest traffic load, AUREWOA had the highest PDR above 96%, indicating that it is quite resilient to network congestion. At maximum traffic load, PDR with ETX fell below 80%. Once again, we can make reasonable statements about the reliability and stability of the proposed AUREWOA algorithm as it pertains to successful delivery of packets at differing traffic loads.

Table 5: Comparison table on End-to-End Delay

Traffic Load	ETX [41]	ELP [42]	PRR [43]	FL-EPSO	PEAP	WFQ-IWOA	AUREWOA (Proposed)
50	42.1	38.7	35.9	32.5	30.1	27.4	24.2
100	48.9	44.5	40.8	37.1	33.5	30.2	26.0
150	55.3	50.2	45.7	41.6	37.3	33.1	28.4
200	62.0	56.8	51.5	46.2	41.9	36.8	30.8
250	68.7	63.4	57.8	50.9	46.7	40.1	33.2

The table 5 is End-to-End Delay comparison illustrates; AUREWOA produces significantly less delay than all of the other algorithms, consistently having the lowest delay for all traffic loads. Recall that as traffic increased from 50 to 250, all traffic loads had higher delay due to additional overhead with routing delays and network congestion. These overheads led to increased delays for all of the methods due to high congestion respectively. Nevertheless, though all algorithms experienced delays, AUREWOA still efficiently and better managed the routing paths and load of sending packets, with the delay of AUREWOA only increasing from 24.2 ms to just 33.2 ms and even at the highest traffic load of 250 packets (and 67% of link throughput), the delay ended well below ETX's delay of 68.7 ms.

This just shows how AUREWOA does an excellent job of reducing latency due to path selection and traffic-awareness mechanisms to ensure communication is catered to more definitive criteria resulting in faster and more responsive performance even in high-demand situations.

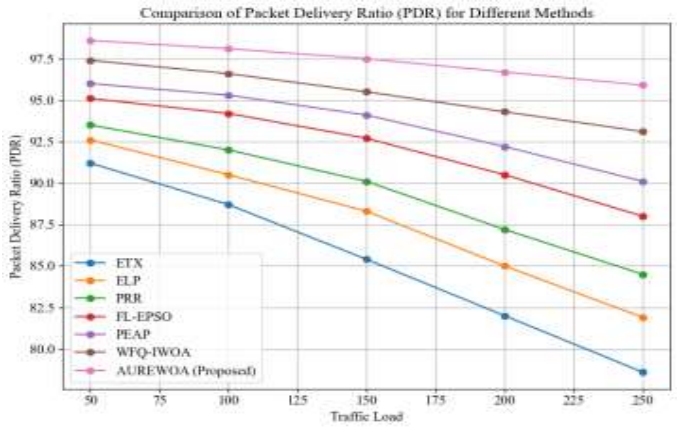

Figure 10: End-to-End Delay Comparison Chart

The figure 10 illustrates the packet delivery ratio (PDR) of a number of different algorithms - ETX, ELP, PRR, FL-EPSO, PEAP, WFQ-IWOA, and the proposed AUREWOA - as traffic load increases. Although all methods see reduced PDR as traffic load is increased from 50 to 250, AUREWOA begins with the highest delivery ratio and remains the highest delivery ratio overall all through the tests. AUREWOA, with only a slight decrease from around 98% to 96% at the maximum load, demonstrates an excellent level of reliability and robust in the high packet delivery performance under heavy network conditions. In contrast, ETX begins with a strong delivery ratio but declines significantly in performance and is limited below 80% at a traffic load of 250. This is shown in the graph below, along with each of the other systems ability to maintain high delivery performance relative to traffic load.

Table 6: Comparison table on Throughput

Traffic Load	ETX [41]	ELP [42]	PRR [43]	FL-EPSO	PEAP	WFQ-IWOA	AUREWOA (Proposed)
50	458	472	489	512	528	547	562
100	891	926	948	976	991	1018	1043
150	1293	1321	1364	1398	1422	1456	1493
200	1676	1724	1762	1795	1820	1860	1905
250	2038	2087	2132	2174	2210	2268	2312

Comparative Throughput shows that the AUREWOA algorithm consistently achieves the best throughput performance under all traffic loads compared to both traditional and existing methods of optimization. All schemes show an increase in throughput as traffic load increases from 50 to 250, indicating some scaling characteristics. The Throughput performance of the other schemes were lower in comparison to AUREWOA with showing performance not exceeding 2038 for ETX compared to AUREWOA's throughput of 2312 at the highest load. AUREWOA is more capable of managing larger volumes of data efficiently. AUREWOA's performance can be attributed to the combination of its adaptive route optimization, load balancing of routes and the congestion-aware mechanisms which utilizes the nodes data rates all the while in flight based on a dynamic network state.

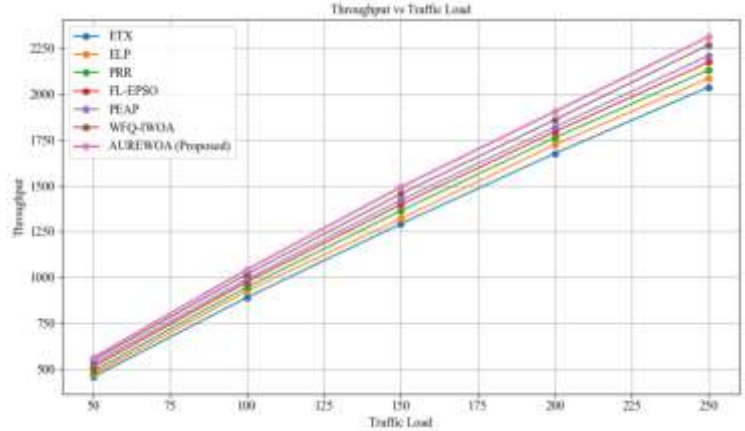

Figure 11: Throughput Comparison Chart

The figure 11 summarizes the throughput performance of the following algorithms: ETX, ELP, PRR, FL-EPSO, PEAP, WFQ-IWOA, and proposed AUREWOA, under increasing traffic load. The throughput increased steadily with increasing traffic load for each method of operation; but the proposed AUREWOA was the highest throughput

for all traffic levels, and surpassed 2250 units at the maximum traffic load of 250. ETX had the lowest throughput at every instance. This shows that AUREWOA improves data transmission efficiency while performing better than any of the previous other approaches in terms of handling greater network demands because throughput is superior to all others even in heavy load conditions.

Table 7: Comparison table on Packet Loss Rate (PLR)

Traffic Load	ETX [41]	ELP [42]	PRR [43]	FL-EPSO	PEAP	WFQ-IWOA	AUREWOA (Proposed)
50	9.5	8.2	7.1	5.4	4.9	3.5	2.1
100	12.1	10.5	9.3	7.4	6.2	4.8	3.3
150	15.3	13.2	11.4	9.1	7.5	5.9	4.2
200	18.7	16.1	14.2	11.2	9.4	7.2	5.1
250	22.0	19.4	17.3	13.6	11.8	8.5	6.0

The Packet Loss Rate (PLR) comparison demonstrates AUREWOA has advantageous resiliency and reliability in transmitting packets with the lowest PLR throughout all levels of traffic load. Overall, all algorithms show an increase in packet loss as network traffic load increases from 50 to 250 due to congestion and interference; however, AUREWOA's increase is considerably slow, the PLR only growing from 2.1% to 6.0%, whereas ETX is growing from 9.5% to 22.0%. As such, AUREWOA's apparent effectiveness at reducing data loss through smart routing and congestion management, as well as fault tolerance at traffic loads provides greater stable and reliable network communications under traffic.

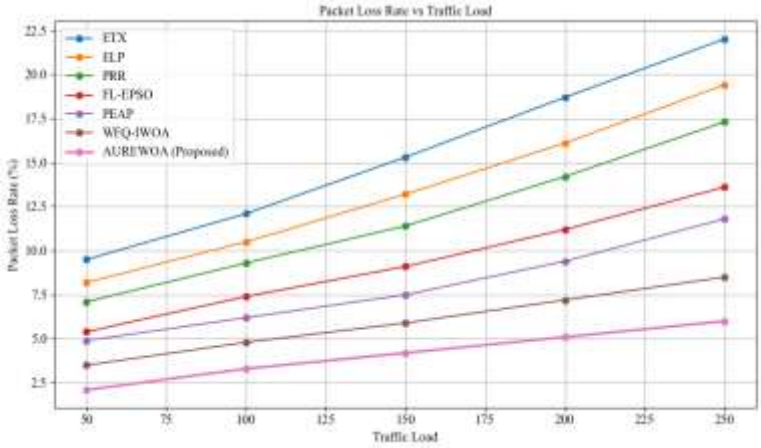

Figure 12: Packet Loss Rate (PLR) Comparison Chart

The figure 12 demonstrates the variation of packet loss rate (%) with the increase in traffic load for various algorithms, including ETX, ELP, PRR, FL-EPSO, PEAP, WFQ-IWOA and the proposed AUREWOA. It is evident that all methods experience an increase in packet loss rate with increasing traffic load from 50 to 250, but the increase happens at different rates! ETX experiences the most packet loss, peaking at over 22 % at peak load, whereas AUREWOA appears to steadily achieve the lowest packet loss, staying below 6 %, even at peak traffic. This shows that with the proposed AUREWOA algorithm under heavy traffic is most effective in minimizing lost packets, compared to other hybrid load balance approaches, while maintaining highly reliable network, data delivery.

Table 8: Comparison table on Computation Time

Traffic Load	ETX [41]	ELP [42]	PRR [43]	FL-EPSO	PEAP	WFQ-IWOA	AUREWOA (Proposed)
50	13.2	14.0	12.6	18.4	16.9	15.4	12.2
100	17.6	18.1	16.7	22.3	20.5	19.1	14.6
150	21.4	22.3	20.8	27.1	24.8	22.7	17.1
200	25.0	26.1	24.6	32.5	29.9	27.4	19.5
250	28.7	30.0	28.2	38.0	35.2	31.7	22.0

The Computation Time comparison shows that AUREWOA consistently shows the lowest processing overhead of all algorithms studied across the increasing traffic loads. While increasing network traffic (from 50 to 250 units of traffic) resulted in more computation time for all algorithms, AUREWOA scaled better, increasing from 12.2 ms to 22.0 ms. Notably lower than FL-EPSO and PEAP, which increased to 38.0 ms and 35.2 ms, respectively. The results clearly show AUREWOA's light-weight computational design and its efficient optimization mechanism. It is especially applicable to real-time application environments in resource-constrained or high-throughput situations.

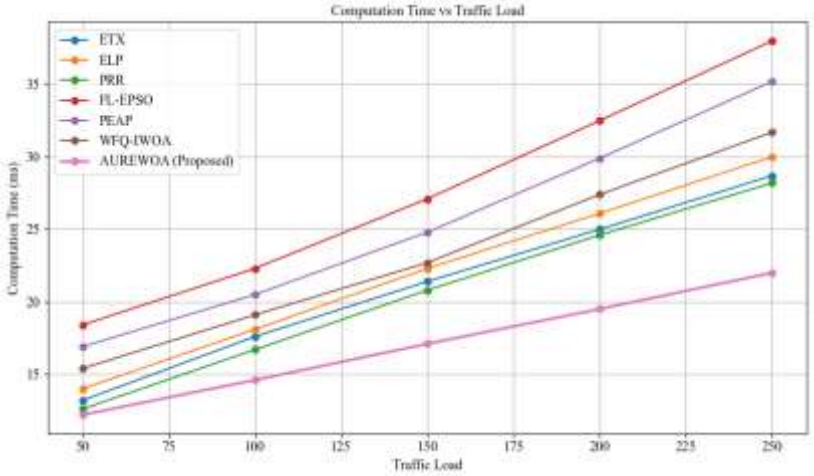

Figure 13: Computation Time Comparison Chart

The figure 13 shows the average computation time (in milliseconds) as a function of traffic load for the different algorithms considered (ETX, ELP, PRR, FL-EPSO, PEAP, WFQ-IWOA and the proposed AUREWOA). As shown, the average computation time increased for all methods with rising traffic load from 50 to 250. The algorithm with the slowest average computation time is FL-EPSO where peak computation time was close to 38 ms at a traffic load of 250. The proposed AUREWOA produced the fastest average computation time and consistently had a lower average computation time than the other models (still under 23 ms at a traffic load of 250). Overall, the proposed AUREWOA has better cognitive computational efficiency in rising traffic loads compared to existing methods highlighted in the results. The size of the performance gap further reinforces the suggested scalability of the proposed algorithm and processing optimality.

V. CONCLUSION

In conclusion, this study provides unique way in enhancing security, authentication, and link quality management in Software Defined Networking (SDN) using three-phase model to integrate these processes to maximize the AUREWOA algorithm. Phase 1 enhances on the Protected Extensible Authentication Protocol (PEAP) by introducing other cryptographic methods, such as a forward secret and dynamic key exchange to provide industry standard level of protection and to interact securely on the networks. Phase 2 will maxims Quality of Service (QoS) provisioning using a Hybrid Parameters Scheduling and Optimization model that combines Weighted Fair Queuing (WFQ) with Pseudorandom Number Generator Improved Whale Optimization Algorithm (IWOA) tool additional random drivers, allowing for dynamic resource allocation. Phase 3 introduced AUREWOA which integrates authentication and user independent scheduling options using a hybrid approach and also additional dynamic resource allocation in optimizing network performance and route effectiveness and autonomy. The experiments demonstrate that AUREWOA will outperform all six benchmarking methods (ETX, ELP, PRR, FL-EPSO, PEAP, and WFQ-IWOA across all traffic loads. It has shown superior metrics in Link Quality Index (LQI), Packet Delivery Ratio (PDR), End-to-End Delay, Throughput, Packet Loss Rate (PLR), and Computation Time, all while demonstrating high reliability, low latency, no packet loss, and efficient computation under network congestion. This solution is strongly robust, scalable, and adaptive, and handles the challenges of SDN in secure flow setup, resource management, and dynamic link management, making it a strong option for future applications relying on SDN, such as IoT and real-time systems.

REFERENCES:

1. Golightly, L., Modesti, P., Garcia, R., & Chang, V. (2023). Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN. Cyber Security and Applications, 1, 100015.
2. Kanwal, A., Nizamuddin, M., Iqbal, W., Aman, W., Abbas, Y., & Mussiraliyeva, S. (2024). Exploring Security Dynamics in SDN Controller Architectures: Threat Landscape and Implications. IEEE Access.
3. Malani, S., Srinivas, J., Das, A. K., Srinathan, K., & Jo, M. (2019). Certificate-based anonymous device access control scheme for IoT environment. IEEE Internet of Things Journal, 6(6), 9762-9773.
4. Hu, J., Shen, Z., Chen, K., Liu, Y., Meng, Q., Wang, F., & Liu, Y. (2024). TAAC: Secure and Efficient Time-Attribute-Based Access Control Scheme in SDN-IoT. IET Information Security, 2024(1), 8059692.
5. You, H., Ko, D., Kim, D., Wong, R., & Joe, I. (2023). Dynamic access control method for SDP-based network environments. EURASIP Journal on Wireless Communications and Networking, 2023(1), 94.
6. Ramprasath, J., & Seethalakshmi, V. (2021). Secure access of resources in software-defined networks using dynamic access control list. International Journal of Communication Systems, 34(1), e4607.

7. Parra-Espín, J. A., Marín-López, R., López-Millán, G., Pereñíguez-García, F., & Canovas, O. (2023). SDN-based automated rekey of IPsec security associations: Design and practical validations. Computer Networks, 233, 109905.
8. Abdi, A. H., Audah, L., Salh, A., Alhartomi, M. A., Rasheed, H., Ahmed, S., & Tahir, A. (2024). Security control and data planes of SDN: a comprehensive review of traditional, AI and MTD approaches to security solutions. IEEE Access.
9. El Jaouhari, S., Bouabdallah, A., & Corici, A. A. (2021). SDN-based security management of multiple WoT smart spaces. Journal of Ambient Intelligence and Humanized Computing, 12(10), 9081-9096.
10. Chica, J. C. C., Imbachi, J. C., & Vega, J. F. B. (2020). Security in SDN: A comprehensive survey. Journal of Network and Computer Applications, 159, 102595.
11. Begovic, M., Causevic, S., Memic, B., & Haskovic, A. (2020). AI-aided traffic differentiated QoS routing and dynamic offloading in distributed fragmentation optimized SDN-IoT. Int. J. Eng. Res. Technol, 13(8), 1880-1895.
12. Joo, H., Lee, S., Lee, S., & Kim, H. (2022). Optimizing time-sensitive software-defined wireless networks with reinforcement learning. IEEE Access, 10, 119496-119505.
13. Tahmasebi, S., Safi, M., Zolfi, S., Maghsoudi, M. R., Faragardi, H. R., & Fotouhi, H. (2020). Cuckoo-PC: An evolutionary synchronization-aware placement of SDN controllers for optimizing the network performance in WSNs. Sensors, 20(11), 3231.
14. Galán-Jiménez, J., Polverini, M., Lavacca, F. G., Herrera, J. L., & Berrocal, J. (2023). Joint energy efficiency and load balancing optimization in hybrid IP/SDN networks. Annals of Telecommunications, 78(1), 13-31.
15. Guo, A., & Yuan, C. (2021). Network intelligent control and traffic optimization based on SDN and artificial intelligence. Electronics, 10(6), 700.
16. Bagaa, M., Dutra, D. L. C., Taleb, T., & Samdanis, K. (2020). On SDN-driven network optimization and QoS aware routing using multiple paths. IEEE Transactions on Wireless Communications, 19(7), 4700-4714.
17. Tache, M. D., Păscuțoiu, O., & Borcoci, E. (2024). Optimization algorithms in SDN: Routing, load balancing, and delay optimization. Applied Sciences, 14(14), 5967.
18. da Silva, H. W., Barbalho, F. R., & Neto, A. V. (2019). Cross-layer multiuser session control for optimized communications on SDN-based cloud platforms. Future Generation Computer Systems, 92, 1116-1130.
19. Fröhlich, P., Gelenbe, E., Fiołka, J., Chęciński, J., Nowak, M., & Filus, Z. (2021). Smart SDN management of fog services to optimize QoS and energy. Sensors, 21(9), 3105.
20. Isyaku, B., Bakar, K. A., Mohd Zahid, M. S., Alkhammash, E. H., Saeed, F., & Ghaleb, F. A. (2021). Route path selection optimization scheme based link quality estimation and critical switch awareness for software defined networks. Applied Sciences, 11(19), 9100.
21. Ayedh M, A. T., Wahab, A. W. A., & Idris, M. Y. I. (2023). Systematic literature review on security access control policies and techniques based on privacy requirements in a BYOD environment: State of the art and future directions. Applied Sciences, 13(14), 8048.
22. Jiang, B., He, Q., He, M., Zhai, Z., & Zhao, B. (2023). FACSC: Fine-Grained Access Control Based on Smart Contract for Terminals in Software-Defined Network. Security and Communication Networks, 2023(1), 6013270.
23. Barach, J. (2025, January). Towards Zero Trust Security in SDN: A Multi-Layered Defense Strategy. In Proceedings of the 26th International Conference on Distributed Computing and Networking (pp. 331-339).
24. Vegas, J., & Llamas, C. (2024). Opportunities and Challenges of Artificial Intelligence Applied to Identity and Access Management in Industrial Environments. Future Internet, 16(12), 469.
25. Mishra, S. R., Shanmugam, B., Yeo, K. C., & Thennadil, S. (2025). SDN-Enabled IoT Security Frameworks—A Review of Existing Challenges. Technologies, 13(3), 121.
26. Pradeep, S., Sharma, Y. K., Lilhore, U. K., Simaiya, S., Kumar, A., Ahuja, S., ... & Chakrabarti, T. (2023). Developing an SDN security model (EnsureS) based on lightweight service path validation with batch hashing and tag verification. Scientific Reports, 13(1), 17381.
27. Rahdari, A., Jalili, A., Esnaashari, M., Gheisari, M., Vorobeva, A. A., Fang, Z., ... & Tahaei, H. (2024). Security and Privacy Challenges in SDN-Enabled IoT Systems: Causes, Proposed Solutions, and Future Directions. Computers, Materials & Continua, 80(2).
28. Onyema, E. M., Kumar, M. A., Balasubaramanian, S., Bharany, S., Rehman, A. U., Eldin, E. T., & Shafiq, M. (2022). A security policy protocol for detection and prevention of internet control message protocol attacks in software defined networks. Sustainability, 14(19), 11950.
29. Luo, J. (2024). Design and Implementation of Intelligent Algorithm Optimization Network Access Control Strategy in SDN Architecture. Procedia Computer Science, 247, 121-128.
30. Anitha, H. M., Jayarekha, P., Sivaraman, A., & Mehta, A. (2024). SDN enabled role based shared secret scheme for virtual machine security in cloud environment. Cyber Security and Applications, 2, 100043.
31. Younus, M. U., Khan, M. K., Anjum, M. R., Afridi, S., Arain, Z. A., & Jamali, A. A. (2020). Optimizing the lifetime of software defined wireless sensor network via reinforcement learning. ieee access, 9, 259-272.
32. Darade, S. A., Akkalakshmi, M., & Pagar, D. N. (2022, March). SDN based load balancing technique in internet of vehicle using integrated whale optimization method. In AIP Conference Proceedings (Vol. 2469, No. 1). AIP Publishing.
33. Akin, E., & Korkmaz, T. (2019). Comparison of routing algorithms with static and dynamic link cost in software defined networking (SDN). IEEE Access, 7, 148629-148644.

34. Shabbir, G., Akram, A., Iqbal, M. M., Jabbar, S., Alfawair, M., & Chaudhry, J. (2020). Network performance enhancement of multi-sink enabled low power lossy networks in SDN based Internet of Things. International Journal of Parallel Programming, 48(2), 367-398.
35. Kirubasri, G., Sankar, S., Pandey, D., Pandey, B. K., Nassa, V. K., & Dadheech, P. (2022). Software-defined networking-based Ad hoc networks routing protocols. In Software defined networking for Ad Hoc networks (pp. 95-123). Cham: Springer International Publishing.
36. El-Garoui, L., Pierre, S., & Chamberland, S. (2020). A new SDN-based routing protocol for improving delay in smart city environments. Smart Cities, 3(3), 1004-1021.
37. Hussein, N. H., Koh, S. P., Yaw, C. T., Tiong, S. K., Benedict, F., Yusaf, T., ... & Hong, T. C. (2024). SDN-based VANET routing: A comprehensive survey on architectures, protocols, analysis, and future challenges. IEEE Access.
38. Rabet, I., Selvaraju, S. P., Fotouhi, H., Alves, M., Vahabi, M., Balador, A., & Björkman, M. (2022). SDMob: SDN-based mobility management for IoT networks. Journal of Sensor and Actuator Networks, 11(1), 8.
39. Chen, J., Liao, C., Wang, Y., Jin, L., Lu, X., Xie, X., & Yao, R. (2022). AQMDRL: Automatic quality of service architecture based on multistep deep reinforcement learning in software-defined networking. Sensors, 23(1), 429.
40. Yan, X., Hu, X., & Liu, W. (2021). SDN controller deployment for QoS guarantees in tactical Ad Hoc networks. Wireless Communications and Mobile Computing, 2021(1), 5586650.
41. Subramani, B. (2022). Enhanced Objective Function ETX Metric in Routing Protocol for Low-Power and Lossy Networks (RPL). International Journal of Computer Networks and Applications (IJCNA), 9(4).
42. Malik, S., Malek, M., Saidin, S. F., & Afip, L. A. (2022). English Proficiency Among Business Students and Its Impact on Academic Performance. International Journal of Education, Psychology and Counselling (IJEPC).
43. Alahmadi, A., & Chung, T. S. (2024). An Effective Selection of Memory Technologies for TCAM to Improve the Search Operations: Demonstration of Memory Efficiency in SDN Recovery. Electronics, 13(4), 707.