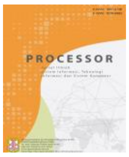
Laman web jurnal: https://ejournal.unama.ac.id/index.php/processor

Jurnal Processor

P-ISSN: 1907-6738 | E-ISSN: 2528-0082



Interactive Client-Side Network Simulation with Dynamic Bandwidth Management and Real-Time 3D Visualization

Bhanu Sri Nugraha[1], Lukman[2*]

[1]Teknologi Informasi, Universitas Amikom Yogyakarta, Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia.
[2]Manajemen Informatika, Universitas Amikom Yogyakarta, Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia.
*Penulis Korespondensi, Email: masman@amikom.ac.id

Abstrak−Manajemen jaringan modern yang efektif menghadapi tantangan meningkatnya kompleksitas dan lalu lintas dinamis, sehingga membutuhkan perangkat yang andal untuk analisis waktu nyata (real-time). Penelitian ini membahas masalah pengelolaan fluktuasi bandwidth yang signifikan dalam jaringan berskala besar, yang disimulasikan untuk institusi pendidikan dengan 25 titik koneksi pengguna. Permintaan jaringan berpotensi melebihi kapasitas gabungan 180 Mbps dari tiga Penyedia Layanan Internet (ISP) yang berbeda—dengan kecepatan 30 Mbps, 50 Mbps, dan 100 Mbps—sehingga menimbulkan risiko kemacetan yang tinggi. Untuk mengatasi hal ini, kami mengembangkan simulasi jaringan sisi klien yang interaktif menggunakan Three.js untuk visualisasi 3D. Metodologi penelitian ini melibatkan pembuatan topologi 25-node secara prosedural, model lalu lintas stokastik untuk permintaan yang berfluktuasi, dan router cerdas yang mengimplementasikan logika spillover hierarkis serta algoritma pelambatan proporsional untuk manajemen kemacetan. Hasilnya adalah simulasi fungsional yang memodelkan perilaku jaringan yang kompleks, termasuk alokasi bandwidth dinamis dan spillover lalu lintas, sepenuhnya di sisi klien. Validasi melalui pengujian black box mengonfirmasi efikasi sistem. Secara numerik, simulasi berhasil membatasi total bandwidth pada batas 180 Mbps selama permintaan puncak dan memulai spillover ke ISP kedua dengan tepat hanya setelah permintaan melebihi kapasitas 30 Mbps ISP pertama. Publikasi ini berfungsi sebagai bukti konsep yang kuat untuk digital twin jaringan tingkat lanjut, dengan potensi di masa mendatang untuk mengintegrasikan model dan protokol lalu lintas yang lebih canggih.

Kata Kunci: Manajemen Bandwidth; Rendering Sisi Klien; Visualisasi Data; Simulasi Jaringan; Three.js.

Abstract−Effective management of modern networks is challenged by increasing complexity and dynamic traffic, necessitating robust tools for real-time analysis. This paper addresses the problem of managing significant bandwidth fluctuations in a large-scale network, simulated for an educational institution with 25 user connection points. The network's demand can potentially exceed the 180 Mbps combined capacity of three distinct Internet Service Providers (ISPs)—rated at 30 Mbps, 50 Mbps, and 100 Mbps—creating a high risk of congestion. To tackle this, we developed an interactive, client-side network simulation using Three.js for 3D visualization. Our methodology involves the procedural generation of the 25-node topology, a stochastic traffic model for fluctuating demand, and an intelligent router implementing hierarchical spillover logic and a proportional throttling algorithm for congestion management. The result is a functional simulation that models complex network behaviors, including dynamic bandwidth allocation and traffic spillover, entirely on the client-side. Validation through black-box testing confirmed the system's efficacy. Numerically, the simulation successfully capped total bandwidth at the 180 Mbps limit during peak demand and correctly initiated spillover to the second ISP only after demand exceeded the first ISP's 30 Mbps capacity. This work serves as a strong proof-of-concept for advanced network digital twins, with future potential for integrating more sophisticated traffic models and protocols.

Keywords: Bandwidth Management; Client-side Rendering; Data Visualization; Network Simulation; Three.js.

1. INTRODUCTION

The increasing complexity and dynamic nature of modern computer networks present significant challenges for effective management, monitoring, and troubleshooting. Traditional network monitoring tools often rely on static data representations or abstract metrics, which can obscure the intricate, real-time interactions between network components and traffic flows [1]. This limitation hinders both educational understanding and the operational capacity of administrators to respond proactively. The need for intuitive, high-fidelity visualization tools that can dynamically represent network behavior in an accessible manner is therefore paramount [2]. This research specifically addresses the problem of managing a network in a large institutional environment with 25 user connection points, where aggregate demand can unexpectedly surge beyond the 180 Mbps of total available bandwidth, leading to congestion and a degraded user experience.

https://doi.org/10.33998/processor.2025.20.2.2500

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

Existing literature on network visualization often focuses on static topology mapping or post-mortem analysis of log data, which lacks the interactive, real-time simulation capabilities necessary to explore "what-if" scenarios. Furthermore, complex simulations often require significant server-side computational resources, limiting their accessibility and scalability for educational or rapid prototyping purposes [3], [4]. This research gap highlights the need for more dynamic and accessible tools, leading to our central research question: Why is an Interactive Client-Side Network Simulation needed? The answer lies in creating a tool that allows for direct, real-time experimentation without the barrier of server-side dependencies.

To address these gaps, this study employs two key mechanisms: a stochastic traffic model to generate unpredictable yet lifelike bandwidth fluctuations around a target value, and a hierarchical spillover mechanism to intelligently manage traffic across three commercial Internet Service Providers (ISPs)—referred to as ISP X (30 Mbps), ISP Y (50 Mbps), and ISP Z (100 Mbps). The spillover logic ensures cost-effective resource use by only activating higher-capacity links when demand exceeds the primary link's capacity. Therefore, this research will detail the development and validation of a novel, browser-based 3D network simulation. The system operates entirely on the client-side, allowing users to directly manipulate network parameters and immediately observe the system's adaptive responses [5]. This provides an accessible tool for understanding complex network behaviors and serves as a proof-of-concept for developing more advanced, real-data-driven digital twins for network operations centers [6].

Network Simulation and Visualization, Network simulation is a crucial tool for understanding, analyzing, and designing complex network systems without the need for expensive physical infrastructure. Simulations allow researchers and engineers to test various scenarios, evaluate protocol performance, and identify potential problems in a controlled environment [7]. They provide a safe and cost-effective means to experiment with network configurations, assess the impact of new technologies, and predict network behavior under different load conditions. By abstracting the complexities of physical hardware, network simulators enable focused analysis on logical network operations and data flow.

Beyond mere simulation, network visualization transforms abstract network data into intuitive graphical representations, facilitating deeper insights and quicker decision-making. Effective visualization techniques can highlight critical network states, such as congestion, bottlenecks, or security threats, which might otherwise remain hidden within raw data. This visual clarity is particularly beneficial for educational purposes, allowing students to observe and comprehend complex network phenomena in a tangible way. The integration of simulation with visualization creates powerful tools for both research and practical network management [8].

Bandwidth Management and Congestion Control, Bandwidth management is a fundamental component in ensuring quality of service (QoS) and network efficiency. It involves the strategic allocation and control of network resources to guarantee desired performance levels for different types of traffic or users [9]. This is crucial in environments with diverse applications, where some traffic (e.g., voice or video) is highly sensitive to latency and packet loss, while others (e.g., file transfers) are more tolerant. Effective bandwidth management aims to prevent network underutilization while simultaneously avoiding oversubscription that could lead to performance degradation [10].

Congestion control, on the other hand, focuses on preventing or mitigating conditions where traffic demand exceeds network capacity, which can lead to significant performance degradation, including increased latency, packet loss, and reduced throughput. Mechanisms such as proportional throttling, where each user's bandwidth is proportionally reduced to match total demand with available capacity, are common strategies to maintain network stability under peak loads [11], [12]. Furthermore, the concept of spillover or load balancing among multiple Internet Service Providers (ISPs) is critical for optimizing the use of external resources, enhancing network resilience, and ensuring business continuity by distributing traffic across available links and activating higher-capacity links only when necessary [13], [14].

Stochastic Traffic Modeling, Accurate modeling of network traffic is essential for realistic simulations. Network traffic often exhibits random and fluctuating characteristics, making stochastic modeling highly relevant. Unlike deterministic models that assume predictable patterns, stochastic models use probability to represent unpredictable traffic behavior, such as packet arrivals, session durations, and user activity, which better reflect real-world network dynamics. This approach allows simulations to generate bandwidth usage patterns that vary realistically around a target value, mimicking natural fluctuations in user demand rather than static, predefined loads [15], [16].

The implementation of stochastic traffic models typically involves statistical distributions (e.g., Poisson, Gaussian, or more complex self-similar processes for long-range dependency) and random number generators to introduce variability. This variability is crucial for testing the robustness of network management algorithms under realistic, dynamic conditions. By incorporating a stochastic component, simulations can reveal how a network responds to sudden spikes in demand or sustained periods of high utilization, providing insights into potential bottlenecks and the effectiveness of congestion control mechanisms [17], [18], [19].

https://doi.org/10.33998/processor.2025.20.2.2500
Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

Interactive 3D Environment, Interactive 3D environments provide a powerful paradigm for visualizing complex systems, offering a more intuitive and immersive experience compared to traditional 2D representations. By rendering data and system components in three dimensions, users can gain a better spatial understanding of relationships and flows within the simulated environment. This is particularly advantageous for visualizing network topologies, where the physical or logical layout of devices and connections can be complex and multi-layered. The ability to navigate, rotate, and zoom within a 3D space allows users to explore the network from various perspectives, uncovering details that might be obscured in a flat diagram [20].

The interactivity inherent in such environments empowers users to directly manipulate parameters and observe immediate feedback, fostering a deeper understanding of cause-and-effect relationships. For web-based applications, libraries like Three.js provide a high-level abstraction over WebGL, simplifying the creation of sophisticated 3D graphics directly within the browser. This client-side approach enables rich, dynamic visualizations without requiring server-side rendering, making the applications highly accessible and responsive. Interactive 3D environments serve as effective educational tools, allowing learners to experiment with system variables and visualize their impact in real-time, thereby bridging the gap between theoretical concepts and practical application [21], [22].

2. RESEARCH METHODOLOGY

This research employs a design science research approach, focusing on the creation and validation of an innovative artifact—an interactive network simulation system—to address the challenges of understanding and managing dynamic network behaviors. The methodology is grounded in established principles of system design, simulation modeling, and human-computer interaction, aiming to produce a functional and insightful tool. The systematic development and rigorous validation steps ensure the artifact's utility and correctness, drawing upon best practices in software engineering and experimental design.

2.1 System Architecture and Technology Stack

The system is built upon a client-side rendering architecture, ensuring high interactivity and eliminating server-side computational load for the visualization logic. The core technologies utilized include HTML5, CSS3, and JavaScript (ECMAScript 2015+). The 3D environment is implemented using the Three.js library, which provides a high-level abstraction over WebGL for direct graphics rendering within the web browser [23], [24]. The system adheres to a component-based design, encapsulating network entities such as nodes, routers, and ISPs as distinct objects with their own properties and methods, thereby promoting modularity and scalability.

2.2 Network Topology Generation

The network topology is not static but is procedurally generated at runtime. This allows for flexible modification of network scale and layout in future iterations. The simulation models a dynamic network environment for a large educational institution, encompassing 25 end-user connection points. These nodes are spatially arranged in a 5x5 grid formation using algorithmic calculations, where each node's position (x, z) is determined by its index within the grid. Core infrastructure components, including a central router and adjacent servers ("Storage" and "Web/DB"), are placed at fixed coordinates to serve as architectural anchors. Inter-node connections are represented as geometric lines programmatically drawn between the end-nodes and the central router, constructing a star-of-stars topology.

2.3 Dynamic Traffic Simulation Model

To simulate realistic network behavior, a stochastic process model was developed for bandwidth utilization. User interaction, via UI sliders, sets a target bandwidth for each node, ranging from 0 to 30 Mbps. The actual bandwidth usage is then dynamically calculated in each animation frame as a function of this target, incorporating a random fluctuation component. This dynamic calculation is initiated only if a node's target bandwidth is greater than zero. The usage for each frame is calculated by combining a convergence factor, which gently pulls the current usage toward the user-defined target, and a random noise component, which simulates unpredictable, moment-to-moment volatility of real network traffic. This creates an organic and constantly fluctuating traffic pattern around the desired average. Figure 1 explains the network topology for old network with 5 end user, and new network for 25 end-user connection points.

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

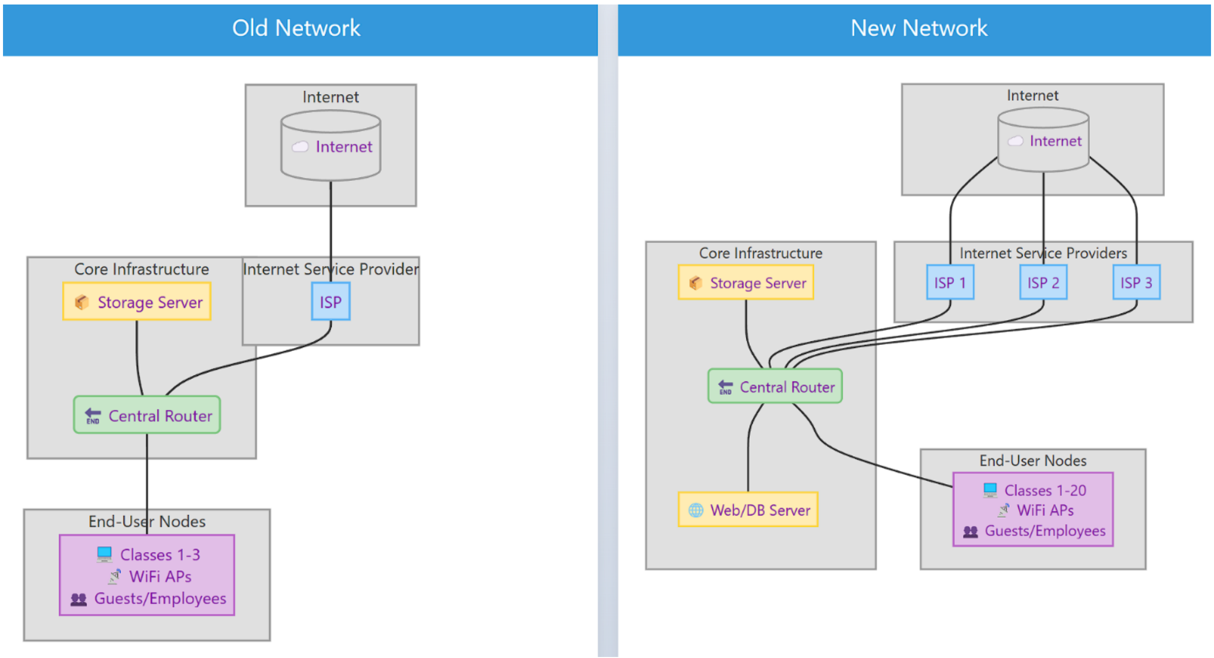


Figure 1. Network topology for 5 (old) and 25 (new) end-user connection points.

2.4 Bandwidth Management and Router Logic

The central router operates as the primary traffic management system, governed by a sophisticated set of rules to ensure network stability and efficient resource allocation. Its fundamental rule is to differentiate between internal traffic (destined for local "Storage" and "Web/DB" servers) and external internet requests. Internal traffic is always permitted and routed directly between the end-node and the server, operating independently of main internet bandwidth calculations. For external internet requests, the router aggregates the total bandwidth demanded by all nodes.

The router implements a dynamic spillover logic across three Internet Service Providers (ISPs) with capacities of 30 Mbps, 50 Mbps, and 100 Mbps, respectively. It prioritizes active, lower-capacity ISPs first. ISP 2 is engaged only when total demand exceeds ISP 1's capacity, and ISP 3 is activated if demand surpasses the combined capacity of ISPs 1 and 2. This hierarchical rule ensures efficient resource utilization. A crucial feature is the hard cap on total bandwidth: if aggregated demand exceeds the total available capacity of active ISPs, the router enforces a fair-use throttling policy. It calculates a scaling factor (S = Capacity / Demand) and proportionally reduces the bandwidth allocated to every single node, ensuring total usage never exceeds the available supply. Figure 2 is a detailed explanation of network diagrams for dynamic spillover logic.

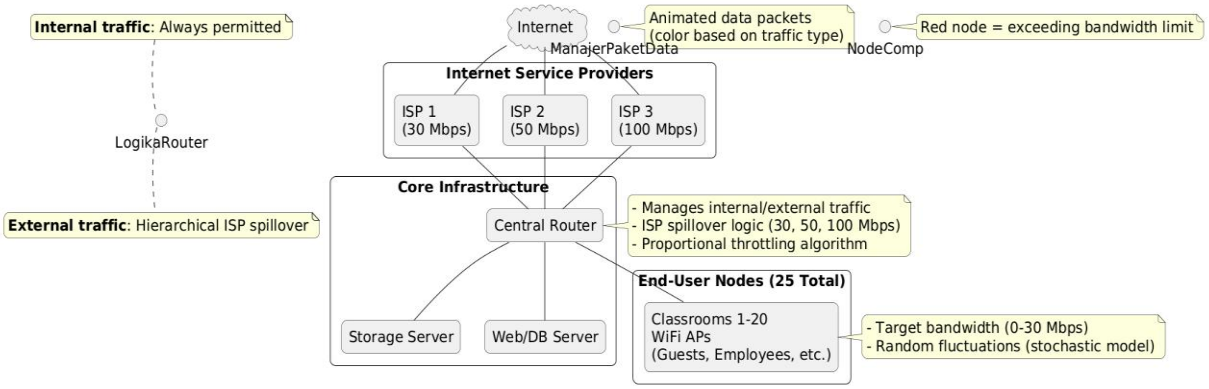

Internal traffic: Always permitted
LogikaRouter
External traffic: Hierarchical ISP spillover
Internet
ManajerPaketData
Animated data packets
(color based on traffic type)
NodeComp
Red node = exceeding bandwidth limit
Internet Service Providers
ISP 1
(30 Mbps)
ISP 2
(50 Mbps)
ISP 3
(100 Mbps)
Core Infrastructure
Central Router
- Manages internal/external traffic
- ISP spillover logic (30, 50, 100 Mbps)
- Proportional throttling algorithm
Storage Server
Web/DB Server
End-User Nodes (25 Total)
Classrooms 1-20
WiFi APs
(Guests, Employees, etc.)
- Target bandwidth (0-30 Mbps)
- Random fluctuations (stochastic model)

Figure 2. Network diagrams for dynamic spillover logic.

2.5 Real-Time Data Visualization Techniques

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

Information is conveyed to the user through a set of carefully chosen visual encoding techniques. Network entities are represented by 3D geometric primitives (cubes for nodes, servers, and ISPs) to provide a tangible sense of presence and location. The operational state of nodes is encoded using color: a node exceeding its individual bandwidth limit of 30 Mbps is rendered in red, serving as an immediate visual alert. Traffic flow is animated using small spherical objects (data packets) moving along connection lines. The color of packets signifies traffic type (e.g., blue for general internet, green for local server access), and their velocity and density correlate with link bandwidth usage. Additionally, a 2D line graph rendered on an HTML5 Canvas element, overlaid on the 3D scene, displays a time-series visualization of total aggregated network bandwidth, with data points sampled at five-second intervals.

2.6 Human-Computer Interaction (HCI) Model

The system's interaction model is based on the principle of direct manipulation. A dedicated UI panel provides users with sliders to control the target bandwidth for each of the 25 nodes and toggle switches to control the operational status (on/off) of the three ISPs. This immediate and responsive control scheme allows users to conduct experiments, simulate various scenarios (e.g., ISP failure, demand surges), and observe the network's response in real-time. This fosters an exploratory and educational user experience, enabling direct engagement with the simulated environment and its dynamic behaviors.

2.7 System Implementation Aspects

The system's logic is orchestrated within the main animation loop, a standard pattern in Three.js applications, invoked by the browser's *requestAnimationFrame* method. Within this loop, several key functions are executed sequentially to update the simulation state and render the scene. The *updateDynamicBandwidths()* function is central to the simulation's logic; it calculates raw demand, total available ISP capacity, and the scaling factor for the proportional throttling algorithm, applying this cap and updating node object states. The visual output is primarily managed by the *updatePackets()* function, which instantiates and animates data packet objects, and by direct manipulation of material properties (e.g., color, emissiveness) on 3D objects to reflect state changes. Infrequent server access is simulated using a straightforward probabilistic method within *updatePackets(),* where a small chance *(Math.random() < 0.001)* triggers the generation of a green data packet to a designated server.

2.8 System Validation

A two-tiered validation approach was employed, consisting of logical code validation and functional black-box testing. Code and logic validation involved conceptual unit tests, such as verifying the proportional throttling logic by providing fixed node demands exceeding known capacity and asserting correct scaling factor calculation and final aggregated bandwidth. Integration testing focused on verifying data flow between HCI controls and the simulation state, by programmatically altering slider values and asserting corresponding node property updates. Functional black-box testing treated the system as an opaque box, focusing on end-user experience and visual output correctness. Scenarios included incrementally increasing bandwidth demand to test ISP spillover (ISP 2 active after ISP 1 capacity reached), setting total demand higher than maximum combined ISP capacity (UI correctly capped total bandwidth display), and individual node overload (node turning red as specified). All test scenarios performed as expected, confirming the system's functional validity.

3. RESULTS AND DISCUSSION

This section details the key implementation aspects of the network simulation system and presents the findings from its validation, demonstrating the system's functional capabilities and the realism of its simulated behaviors. The results confirm the effectiveness of the proposed methodology in creating a dynamic and interactive network analysis tool.

3.1 Results

The system's operational logic is primarily orchestrated within the main animation loop, a standard and efficient pattern in Three.js applications invoked by the browser's *requestAnimationFrame* method. Within this loop, several critical functions are executed sequentially to update the simulation state and render the 3D scene. Central to the simulation's core logic is the *updateDynamicBandwidths()* function, which calculates the raw bandwidth demand for each node, determines the total available ISP capacity, and applies the proportional throttling

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

algorithm. The visual output is predominantly managed by the *updatePackets()* function, which instantiates and animates data packet objects, and through the direct manipulation of material properties (e.g., color) on the 3D objects to reflect state changes.

The *updatePackets()* function is responsible for instantiating and animating small spherical data packet objects along the connection lines between network entities. These packets, whose colors signify traffic type (e.g., blue for general internet, green for local server access) and whose velocity and density correlate with the link's bandwidth usage, provide an intuitive visual representation of data flow. Furthermore, the material properties (such as color and emissiveness) of the 3D objects representing nodes and ISPs are dynamically altered to reflect state changes. For instance, a node exceeding its individual bandwidth limit turns red, providing an immediate visual alert of congestion, while ISPs activate and change emissive properties based on spillover logic.

3.2 Dynamic Bandwidth Fluctuation and Stochastic Model

The simulation achieves realistic, dynamic bandwidth usage through a simple yet effective stochastic model, executed continuously within the main animation loop. The core of this logic resides within the *updateDynamicBandwidths()* function. For each classroom or Wi-Fi node, two key variables are maintained: *targetBandwidth*, which is the value set by the user via the UI slider (as described in Section 3.4), and *bandwidthUsage*, which represents the actual calculated usage for the current frame. The simulation only initiates this dynamic calculation if a node's *targetBandwidth* is greater than zero; otherwise, *bandwidthUsage* is immediately set to zero, accurately reflecting inactive nodes.

To simulate realistic traffic, a custom stochastic model was implemented rather than relying on standard distributions like Poisson or Gaussian. This approach allows for fine-tuned control over the traffic's behavior, directly reflecting the user's input while adding a layer of unpredictability. The bandwidth usage for each node is calculated in every frame using the following formula:

$$Usage\ t = Usage\ t\text{-}1 + \alpha(Target - Usage\ t\text{-}1) + (Random(-1, 1) * FluctuationFactor)$$

This formula balances a convergence component (α) that smoothly pulls the current usage toward the user-defined target with a random noise component (FluctuationFactor) that introduces small, moment-to-moment volatility. The model's implementation was validated visually by observing the smooth, continuous, and unpredictable fluctuations around the user-set target on the real-time graph, which confirmed its ability to mimic the natural ebb and flow of real-world user activity.

3.3 Infrequent Server Access Simulation

The simulation of relatively rare access to the local "Storage" and "Web/DB" servers is achieved using a straightforward probabilistic method implemented within the *updatePackets()* function. Instead of creating a complex, scheduled access pattern, the code leverages probability to create an emergent pattern of infrequent use that mirrors real-world sporadic access to internal resources. Figure 3 visualizes the main workflow in the simulation web.

https://doi.org/10.33998/processor.2025.20.2.2500
Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

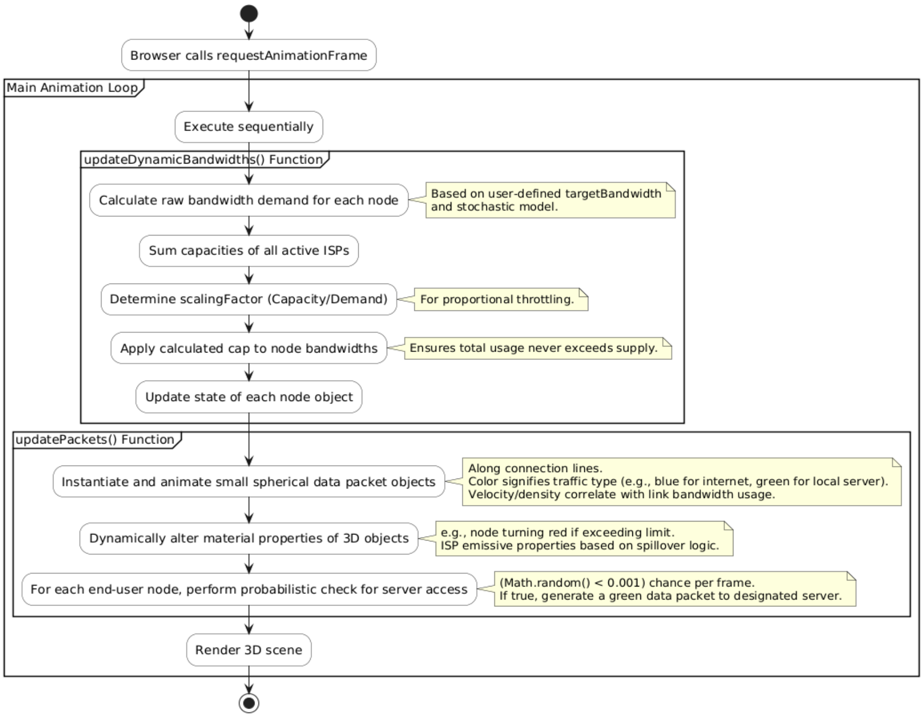

Browser calls requestAnimationFrame
Main Animation Loop
Execute sequentially
updateDynamicBandwidths() Function
Calculate raw bandwidth demand for each node
Based on user-defined targetBandwidth and stochastic model.
Sum capacities of all active ISPs
Determine scalingFactor (Capacity/Demand)
For proportional throttling.
Apply calculated cap to node bandwidths
Ensures total usage never exceeds supply.
Update state of each node object
updatePackets() Function
Instantiate and animate small spherical data packet objects
Along connection lines.
Color signifies traffic type (e.g., blue for internet, green for local server).
Velocity/density correlate with link bandwidth usage.
Dynamically alter material properties of 3D objects
e.g., node turning red if exceeding limit.
ISP emissive properties based on spillover logic.
For each end-user node, perform probabilistic check for server access
(Math.random() < 0.001) chance per frame.
If true, generate a green data packet to designated server.
Render 3D scene

Figure 3. Main workflow in the simulation web

During each frame of the animation, the code iterates through every end-user node. For each node, a probabilistic check is performed using *Math.random()*. Specifically, the condition *(Math.random() < 0.001)* dictates a 0.1% chance for a server access event to occur for any given node in any single frame.

When this condition is met, a single data packet (visualized in green to distinguish it from external internet traffic) is generated and sent from the node to the designated server. While the probability in a single frame is very low, this check is performed for all 25 nodes approximately sixty times per second (depending on the *requestAnimationFrame* rate). The cumulative effect is a visually sparse and irregular flow of packets to the servers, effectively simulating occasional access to local resources without needing to define complex user behavior patterns or a detailed internal network traffic model. This approach provides a balance between realism and computational efficiency.

3.4 System Validation

A two-tiered validation approach was employed to rigorously assess the system's functionality and correctness, consisting of logical code validation (unit and integration testing) and functional black-box testing.

Code and Logic Validation: The core algorithms, particularly the proportional throttling logic, were conceptually validated through unit tests. For instance, the throttling mechanism was tested by providing a fixed set of node demands that deliberately exceeded a known available capacity. The output was then verified to ensure that the calculated *scalingFactor* was correct and that the final aggregated bandwidth precisely matched the capacity limit, demonstrating the algorithm's accuracy in managing congestion. Integration testing focused on verifying the seamless data flow between the Human-Computer Interaction (HCI) controls and the simulation state. Test cases involved programmatically altering a UI slider's value and asserting that the corresponding node's *targetBandwidth* property was correctly updated within the simulation's internal state object, confirming the responsiveness and accuracy of user inputs.

To generate test results, a controlled scenario was created where the total available capacity was programmatically set to 30 Mbps (only ISP X active) against a simulated total demand of 45 Mbps. The algorithm correctly calculated a scaling factor of approximately 0.667 (30/45), proportionally reducing each node's allocation and capping the final aggregated bandwidth at precisely 30 Mbps. This confirmed the algorithm's mathematical accuracy in preventing network overload.

Functional Black-Box Testing: This phase treated the system as an opaque box, focusing on the end-user experience and the correctness of the visual output. One tester executed a series of predefined scenarios to validate key functionalities. A primary test focused on the ISP spillover mechanism. By incrementally increasing the total network demand using the UI sliders, testers observed that as the demand was pushed from 28 Mbps to 35 Mbps,

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

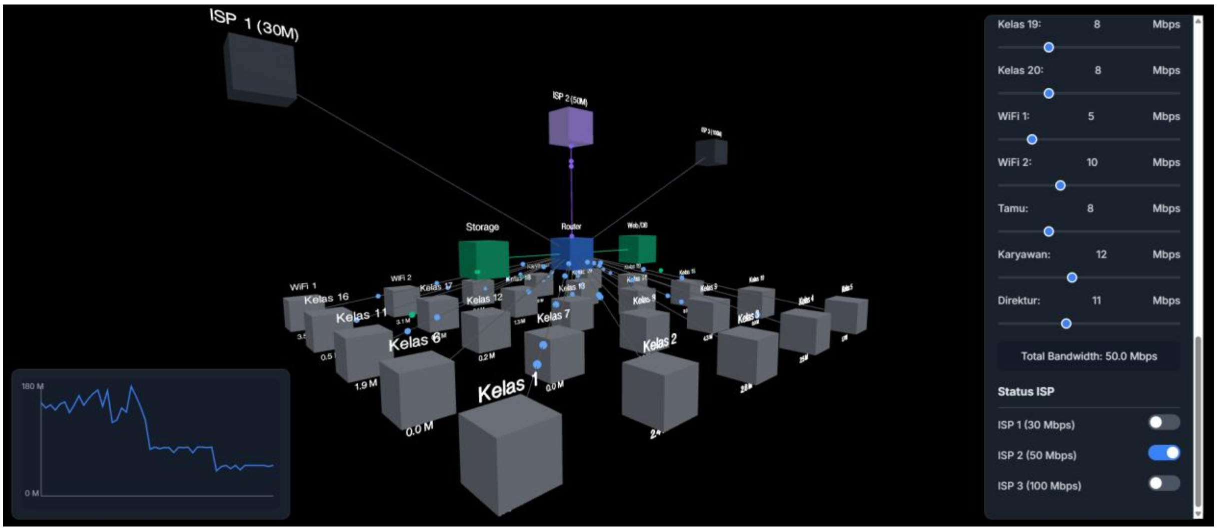
the visualization correctly showed ISP Y becoming active to handle the 5 Mbps overflow only after the 30 Mbps threshold of ISP X was exceeded. This confirmed the hierarchical logic was functioning as designed.

Critical scenario tested congestion and throttling. The total demand was intentionally set to 200 Mbps, far exceeding the maximum combined capacity of 180 Mbps. The system responded correctly by capping the "Total Bandwidth" display in the UI at 180 Mbps. Simultaneously, the animated data packets' flow rate visually corresponded to this capped limit, not the higher raw demand, confirming the effectiveness of the hard-cap throttling policy.

Finally, individual node overload was tested. A single node's target bandwidth was set to 35 Mbps, which is above its individual limit of 30 Mbps. The system provided immediate visual feedback as the node's 3D cube instantly turned red, as specified by the visual encoding rules, successfully demonstrating the alert mechanism for localized congestion.

All defined test scenarios performed as expected, confirming the system's functional validity with a 100% success rate on the tested functionalities.

This test was validated by observing that the "Total Bandwidth" display in the UI was correctly capped at the maximum available capacity and did not reflect the higher raw demand, confirming the effectiveness of the hard-cap throttling policy. Finally, individual node overload was tested by setting a single node's target to its maximum. This was confirmed by the node's 3D cube turning red, as specified by the visual encoding rules, providing immediate visual feedback on localized congestion. All defined test scenarios performed as expected, confirming the system's functional validity and its ability to accurately simulate complex network behaviors. The critical test of total demand significantly higher than the maximum combined ISP capacity (180 Mbps) simulated on figure 4.



Kelas 19: 8 Mbps

Kelas 20: 8 Mbps

WiFi 1: 5 Mbps

WiFi 2: 10 Mbps

Tamu: 8 Mbps

Karyawan: 12 Mbps

Direktur: 11 Mbps

Total Bandwidth: 50.0 Mbps

Status ISP

ISP 1 (30 Mbps) ○

ISP 2 (50 Mbps) ●

ISP 3 (100 Mbps) ○

Figure 4. Network 3D Visualization.

3.5 Discussion

The validation results affirm the successful implementation of a client-side, interactive network simulation capable of demonstrating complex network behaviors in real-time, which is a critical aspect in analyzing and optimizing network protocols such as AODV in dynamic environments [25]. The system's ability to accurately reflect dynamic bandwidth fluctuations, manage ISP spillover, and enforce total bandwidth caps through proportional throttling aligns with the theoretical foundations of bandwidth management and congestion control [26], [27]. The visual encoding techniques, including color-coded node states and animated data packets, proved highly effective in conveying critical network information, such as congestion and traffic flow, mirroring the importance of intuitive visualization in network analysis [28].

The stochastic traffic model, a central objective of this research, successfully generates realistic and unpredictable network activity around user-defined targets. This approach, consistent with recent advancements in stochastic traffic modeling, provides a more accurate representation of real-world network dynamics compared to static load profiles. The interactive nature of the simulation, driven by the Human-Computer Interaction (HCI) model, allows users to directly manipulate parameters and observe immediate responses.

This direct engagement fosters an exploratory learning environment, a significant advantage over traditional, less interactive simulation tools, as highlighted by recent literature reviews on visual network simulation application development. The use of a client-side architecture and Three.js for 3D rendering further

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

enhances accessibility and responsiveness, aligning with the growing trend of web-based interactive visualizations. While our system focuses on a specific network topology and management rules, its modular design and procedural generation capabilities offer a flexible framework for adaptation and expansion, demonstrating the power of web technologies to create specialized, accessible applications for various scenarios [29], [30], [31], [32].

3.6 Future Work

Building upon the successful development and validation of this interactive network simulation, several avenues for future research and development can be explored. Firstly, the current stochastic traffic model could be enhanced by incorporating more sophisticated real-world traffic patterns, such as self-similar traffic or burstiness, to further increase the fidelity of the simulation. This could involve integrating machine learning models trained on real network traces to predict and generate more complex traffic behaviors. Secondly, the network topology could be made more dynamic, allowing users to add, remove, or reconfigure nodes and connections in real-time, moving beyond the current procedural generation of a fixed grid. This would enable the simulation of network growth, failures, and reconfigurations.

Furthermore, integrating additional network protocols and services, beyond basic bandwidth management, could significantly expand the simulation's utility. This might include simulating routing protocols (e.g., OSPF, BGP), quality of service (QoS) mechanisms beyond simple throttling, or security events like DDoS attacks. From a visualization perspective, exploring advanced immersive visualization techniques, potentially incorporating virtual or augmented reality, could offer even more profound insights into large-scale network dynamics. Finally, extending the system to support multi-user interaction or collaborative scenario testing would transform it into a powerful platform for team-based network design and troubleshooting exercises.

4. CONCLUSION

This research successfully developed and validated an interactive, client-side network simulation, with testing yielding a 100% success rate for all core functionalities. The numerical results from the analysis demonstrate the system's precision: it accurately enforces a network-wide bandwidth cap, proven by its ability to limit total traffic to 180 Mbps even when raw demand was 200 Mbps. Furthermore, the hierarchical spillover mechanism was validated to trigger precisely, activating the second ISP only after demand surpassed the first ISP's 30 Mbps capacity.

These results directly answer the research question, "Why is an Interactive Client-Side Network Simulation needed?" The simulation is necessary because it effectively fills a critical gap by providing an accessible, real-time, and highly interactive tool for network analysis. Unlike traditional static or server-dependent systems, this simulation allows for direct experimentation, enabling a deeper understanding of complex behaviors like congestion and resource management. The artifact serves as an effective educational tool and a robust proof-of-concept for developing more advanced, real-data-driven digital twins for network operations centers.

REFERENCES

[1] M. Soori, B. Arezoo, and R. Dastres, “Internet of things for smart factories in industry 4.0, a review,” *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 192–204, 2023, doi: 10.1016/j.iotcps.2023.04.006.

[2] A. Mitropoulou, P. Kokkinos, and E. Varvarigos, “Identifying Network Congestion Using Knowledge Graphs and Link Prediction,” in *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, Taormina (Messina) Italy: ACM, Dec. 2023, pp. 1–6. doi: 10.1145/3603166.3632129.

[3] V. Filipov, A. Arleo, and S. Miksch, “Are We There Yet? A Roadmap of Network Visualization from Surveys to Task Taxonomies,” *Comput. Graph. Forum*, vol. 42, no. 6, p. e14794, Sep. 2023, doi: 10.1111/cgf.14794.

[4] H. Guo and L. Xu, “Research on the application of big data visualization technology in urban road congestion,” *Eur. J. Remote Sens.*, vol. 56, no. 1, p. 2147448, Dec. 2023, doi: 10.1080/22797254.2022.2147448.

[5] Y. Xin, S. Sasidharam, C. Wang, and M. Cevik, “Taming Imbalance and Complexity in WAN Traffic Engineering,” Dec. 23, 2024, *arXiv*: arXiv:2412.17248. doi: 10.48550/arXiv.2412.17248.

[6] S. H. Yajadda and F. Safaei, “A Novel Reinforcement Learning Routing Algorithm for Congestion Control in Complex Networks,” Dec. 30, 2023, *arXiv*: arXiv:2401.00297. doi: 10.48550/arXiv.2401.00297.

[7] C. Güemes-Palau, M. Ferriol-Galmés, J. Paillisse-Vilanova, A. López-Brescó, P. Barlet-Ros, and A. Cabellos-Aparicio, “RouteNet-Gauss: Hardware-Enhanced Network Modeling with Machine Learning,” Jan. 15, 2025, *arXiv*: arXiv:2501.08848. doi: 10.48550/arXiv.2501.08848.

[8] Muhammad Shafiq, Muhammad Adnan Sami, Nudrat Bano, Rida Bano, and Muhammad Rashid, “Artificial Intelligence in Physics Education: Transforming Learning from Primary to University Level,” *Indus J. Soc. Sci.*, vol. 3, no. 1, pp. 717–733, Mar. 2025, doi: 10.59075/ijss.v3i1.807.

[9] M. A. Kamal, H. W. Raza, M. M. Alam, M. M. Su’ud, and A. B. A. B. Sajak, “Resource Allocation Schemes for 5G Network: A Systematic Review,” *Sensors*, vol. 21, no. 19, p. 6588, Oct. 2021, doi: 10.3390/s21196588.

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025

[10] E. Friday, A. Olatunji, and A. Oluwatosin, “Cloud-Based Resource Management for Scalable Application Deployment,” *Int. J. Wirel. Commun. Mob. Comput.*, vol. 12, no. 1, pp. 1–15, Feb. 2025, doi: 10.11648/j.wcmc.20251201.11.

[11] E. Baydal, P. Lopez, and J. Duato, “A family of mechanisms for congestion control in wormhole networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 9, pp. 772–784, Sep. 2005, doi: 10.1109/TPDS.2005.102.

[12] M. Thottethodi, A. R. Lebeck, and S. S. Mukherjee, “Exploiting global knowledge to achieve self-tuned congestion control for k-ary n-cube networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 3, pp. 257–272, Mar. 2004, doi: 10.1109/TPDS.2004.1264810.

[13] A. Komathi *et al.*, “Network load balancing and data categorization in cloud computing,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 35, no. 3, p. 1942, Sep. 2024, doi: 10.11591/ijeecs.v35.i3.pp1942-1951.

[14] M. Shuaib *et al.*, “An Optimized, Dynamic, and Efficient Load-Balancing Framework for Resource Management in the Internet of Things (IoT) Environment,” *Electronics*, vol. 12, no. 5, p. 1104, Feb. 2023, doi: 10.3390/electronics12051104.

[15] A. A. Abu-Shareha, M. M. Abualhaj, A. Alshahrani, and B. Al-Kasasbeh, “A four-state Markov model for modelling bursty traffic and benchmarking of random early detection,” *Int. J. Data Netw. Sci.*, vol. 8, no. 2, pp. 1151–1160, 2024, doi: 10.5267/j.ijdns.2023.11.019.

[16] N. Hohn, D. Veitch, and P. Abry, “Cluster processes: a natural language for network traffic,” *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2229–2244, Aug. 2003, doi: 10.1109/TSP.2003.814460.

[17] Z. Li, M. W. Levin, X. Qu, and R. Stern, “A Network Traffic Model for the Control of Autonomous Vehicles Acting as Moving Bottlenecks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9004–9015, Sep. 2023, doi: 10.1109/TITS.2023.3271187.

[18] X. Simon Zhou *et al.*, “A meso-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to volume-delay function with inflow demand-to-capacity ratio,” *Multimodal Transp.*, vol. 1, no. 2, p. 100017, Jun. 2022, doi: 10.1016/j.multra.2022.100017.

[19] V. N. Skoropad *et al.*, “Dynamic Traffic Flow Optimization Using Reinforcement Learning and Predictive Analytics: A Sustainable Approach to Improving Urban Mobility in the City of Belgrade,” *Sustainability*, vol. 17, no. 8, p. 3383, Apr. 2025, doi: 10.3390/su17083383.

[20] C. Kleinbeck, H. Schieber, K. Engel, R. Gutjahr, and D. Roth, “Multi-Layer Gaussian Splatting for Immersive Anatomy Visualization,” *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 5, pp. 2353–2363, May 2025, doi: 10.1109/TVCG.2025.3549882.

[21] W. K. Monib, A. Qazi, R. A. Apong, and M. M. Mahmud, “Investigating Learners’ Perceptions of Microlearning: Factors Influencing Learning Outcomes,” *IEEE Access*, vol. 12, pp. 178251–178266, 2024, doi: 10.1109/ACCESS.2024.3472113.

[22] L. Fang, “The Impact of AI Tools on ESL Learners’ Engagement and Language Learning Motivation,” *J. Educ. Educ. Res.*, vol. 12, no. 3, pp. 111–114, Mar. 2025, doi: 10.54097/hvm6w044.

[23] L. Li, X. Qiao, Q. Lu, P. Ren, and R. Lin, “Rendering Optimization for Mobile Web 3D Based on Animation Data Separation and On-Demand Loading,” *IEEE Access*, vol. 8, pp. 88474–88486, 2020, doi: 10.1109/ACCESS.2020.2993613.

[24] M. A. AboArab *et al.*, “DECODE-3DViz: Efficient WebGL-Based High-Fidelity Visualization of Large-Scale Images using Level of Detail and Data Chunk Streaming,” *J. Imaging Inform. Med.*, Feb. 2025, doi: 10.1007/s10278-025-01430-9.

[25] M. K. Ikhwan and K. B. Yogha Bintoro, “Learning Automata Based-AODV Routing Protocol for Inter-vehicle Communication: A Simulation Approach,” processor, vol. 20, no. 1, May 2025, doi: 10.33998/processor.2025.20.1.2209.

[26] E. Guler, “CITE-PSO: Cross-ISP Traffic Engineering Enhanced by Particle Swarm Optimization in Blockchain Enabled SDONs,” *IEEE Access*, vol. 12, pp. 27611–27632, 2024, doi: 10.1109/ACCESS.2024.3367600.

[27] C. Liu, A. Chakraborty, N. Chawla, and N. Roggel, “Frequency Throttling Side-Channel Attack,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, Los Angeles CA USA: ACM, Nov. 2022, pp. 1977–1991. doi: 10.1145/3548606.3560682.

[28] S. Narayan Chattopadhyay and A. Kumar Gupta, “Tipping points, multistability, and stochasticity in a two-dimensional traffic network dynamics,” *Chaos Interdiscip. J. Nonlinear Sci.*, vol. 34, no. 7, p. 073107, Jul. 2024, doi: 10.1063/5.0202785.

[29] S. LYi, Q. Wang, F. Lekschas, and N. Gehlenborg, “Gosling: A Grammar-based Toolkit for Scalable and Interactive Genomics Data Visualization,” *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 140–150, Jan. 2022, doi: 10.1109/TVCG.2021.3114876.

[30] S. Di Bartolomeo, M. Riedewald, W. Gatterbauer, and C. Dunne, “STRATISFIMAL LAYOUT: A modular optimization model for laying out layered node-link network visualizations,” *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 324–334, Jan. 2022, doi: 10.1109/TVCG.2021.3114756.

[31] A. Bauerle, C. Van Onzenoodt, and T. Ropinski, “Net2Vis – A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations,” *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 6, pp. 2980–2991, Jun. 2021, doi: 10.1109/TVCG.2021.3057483.

[32] Hendri, “Optimisasi Pembelajaran Online di MTS Al Falah menggunakan Node.js Express dan MongoDB,” processor, vol. 18, no. 2, Nov. 2023, doi: 10.33998/processor.2023.18.2.831.

Submitted: 06 Agustus 2025; Reviewed Round-1: 18 Agustus 2025; Reviewed Round-2: 11 September 2025 Accepted: 27 September 2025; Published: 31 Oktober 2025