electronics

MDPI

Article

Feature Comparison and Throughput Accuracy of OMNeT++ and Riverbed Network Simulators Using a Testbed Environment

Nurul I. Sarkar * and William Knight

Computer and Information Sciences, Auckland University of Technology, Auckland 1010, New Zealand
* Correspondence: nurul.sarkar@aut.ac.nz

Abstract

Accurate network simulation is essential for evaluating wireless communication systems; however, the fidelity of simulation results strongly depends on the underlying modeling assumptions, particularly at the physical (PHY) layer. In this paper, we present a comparative analysis of OMNeT++ and Riverbed Modeler with respect to feature support and throughput prediction accuracy, validated against measurements obtained from a controlled wireless testbed. In this paper, we investigate simulator performance across representative scenarios, including line-of-sight (LOS), non-line-of-sight (NLOS), interference, and congestion conditions. To identify the sources of performance deviation, the analysis further examines the relationship among the signal-to-interference-plus-noise ratio (SINR), packet error rate (PER), and throughput. The results obtained show that both simulators achieve high accuracy under ideal conditions, particularly for LOS and congestion scenarios, where higher layer effects dominate or channel impairments are minimal. However, significant discrepancies arise under NLOS and interference conditions, where accurate modeling of channel dynamics and error behavior is critical. The results show that Riverbed consistently demonstrates closer agreement with testbed, owing to its continuous SINR tracking and probabilistic PER modeling. This enables smooth adaptation of throughput to varying channel conditions. In contrast, OMNeT++ exhibits step-like SINR–throughput and SINR–PER relationships caused by threshold-based abstractions, leading to noticeable inaccuracies in transitional SINR regimes (5–15 dB). Overall, this study highlights that PHY-layer abstraction fidelity is the dominant factor influencing throughput accuracy in network simulation. While OMNeT++ offers flexibility and extensibility suitable for research and prototyping, Riverbed Modeler provides more reliable performance prediction in scenarios requiring high modeling precision. Finally, we provide guidelines for best practice checklists in network simulation and model validation.

Keywords: accuracy; network simulators; network testbed; throughput; field experiment

Check for updates

Academic Editor: Hector E. Nistazakis

Received: 12 April 2026
Revised: 15 June 2026
Accepted: 25 June 2026
Published: 1 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The network performance study using a testbed measurement approach is preferable because it inherently includes the interactions between the real hardware/software and the environment. It also provides an unbiased representation of the radio channel without any assumptions on how signals propagate in the environment. Although this testbed approach is time and labor intensive, these factors can be significantly reduced by automating the testbed experimental system and process as highlighted in [1]. While the testbed results are credible, it takes lot of resources such as hardware and software to set up properly for a medium to large sized network. For instance, it is not very practical to set up a testbed for a

network performance study for nodes greater than 20. Given the lack of adequate time and resources to conduct such testbed measurements for design and performance studies, most turn to network simulators due to their flexibility and ease of setup and run simulation for any number of nodes. Simulation are also being used to validate analytical models and to generalize the research findings. As there are various network simulators available, it is difficult to select which one to utilize since they each have their strengths and weaknesses. There are numerous published studies and white papers that go over different network simulators, but they solely touch on the inner workings of each network simulator and do not compare results against a physical testbed [2].

Comparing simulators such as OMNeT++ and Riverbed Modeler results against a network testbed is useful for validating simulation results [3]. The testbed results add credibility to our findings because it is used as a baseline to compare the features and accuracy of the two selected simulators under investigation. The data obtained in the network testbed allow us to closely replicate the network with similar parameters and compare the accuracy of results against one another in this paper. Backed up by Arvind, T. [4], OMNeT++ is a component-based, modular, and open-source network simulator that is a C++-based discrete event simulator, whereas Riverbed Modeler (formerly OPNET Modeler) is a commercial network simulation package that offers GUI-based debugging and analysis. Assessing these reputable network simulators against a network testbed reveals the level of accuracy that both simulators operate at. In this paper, we therefore select an indoor field experiment at the university library building to replicate with each respective network simulator to obtain throughput results for comparison.

We conducted a testbed measurement campaign using indoor field experiments to measure the Gigabit Wi-Fi (802.11ac) link throughput by varying AP-Rx separation (Rx: Receiver; a wireless laptop) as reported in [3]. This link throughput is used as a baseline and taken into consideration when validating the simulation results of OMNeT++ and Riverbed Modeler. The average link throughput collected from each network simulator is the main vector recorded in this paper to better compare the two network simulators. Although IEEE 802.11ax and IEEE 802.11be (the latest Wi-Fi standards) are available for the testbed link throughput study, we used 802.11ac gigabit Wi-Fi cards and an access point because of their availability in both simulators under study.

The AP-Rx separation ranging from 1 to 25 m was considered due to the simplicity in setting up things and to create an accurate network simulation with as little mistakes as possible. Because 802.11ac only utilizes the 5 GHz band, AP-Rx separation would also show more drastic throughput degradation on the high-bandwidth channel and the short distance coverage that this technology offers. A constant stream of multimedia traffic was implemented in both simulators to match the criteria of the network testbed. We also ensure that the increment of AP-Rx separation (1 m, 5 m, 10 m, 15 m, 20 m, and 25 m) and the simulation parameters and system configuration are identical in both simulators to avoid bias results.

The purpose of this paper is not to generalize or label one network simulator as "better" over the other, as there are many simulator-specific factors to take into consideration. The objective of this study is to outline the accuracy of the network simulators regarding AP-Rx separation in the hope that it can guide network researchers to select an appropriate network simulator for modeling and simulation purposes that can provide unbiased results.

1.1. Research Challenges and Questions

In this study, we address the following two research questions and challenges.

RQ1: Given that both network simulators operate and function differently, how can we ensure that the results obtained are fair and unbiased?

To address RQ1, we developed a network model of both network simulators that matched the conditions in which the network testbed was conducted. The network simulation models developed were put under the same radio propagation model, AP-Rx separation, and same Wi-Fi networking standard, 5 GHz band. It was ensured that all available configurations were as close to one another as possible.

RQ2: Given the network testbed results obtained from field experiments, how can we ensure that the results are fair and accurate when comparing results from both simulators?

To address RQ2, we had to fine tune and replicate the conditions that the network testbed had for the configuration of both simulators. Since both network simulators did not have the same model of access point (AP) and end point devices, we had to edit the configuration of both simulators to ensure that they have similar specifications to one another so that throughput bottlenecks or other relevant issues would not occur.

1.2. Research Contributions

The main contributions of this paper are summarized as follows.

- We analyze and compare the accuracy of results obtained from two credible network simulators, namely OMNeT+ and Riverbed Modeler. The comparison involves creating and analyzing both ad hoc and infrastructure network scenarios.
- We develop and configure network simulation models and parameters for both simulators to obtain relevant performance metrics. To this end, we develop various practical scenarios to run and extract useful simulation statistics, especially network link throughputs for system level performance validation.
- We validate the simulation models of both simulators against the testbed results. To this end, we set up a Wi-Fi network testbed to conduct various field experiments involving Wi-Fi links in the multistory university building under line-of-sight indoor conditions.

1.3. Clarification of Novelty Compared to Existing Work

We have significantly strengthened the manuscript to clearly highlight the following original contributions and novelty beyond merely incorporating testbed measurements:

(i). Introduction of a Hybrid Validation Framework: While prior studies typically perform either simulation-only comparisons or basic validation against limited empirical data, our work introduces a structured hybrid validation framework that systematically integrates the following:

- Controlled testbed experimentation;
- Parallel simulation in OMNeT++ and Riverbed;
- Quantitative cross-layer performance alignment.

This framework establishes traceable mapping between simulated and real-world network dynamics, enabling reproducible benchmarking of simulator fidelity. This goes beyond simple comparison by proposing a generalizable methodology for simulator validation applicable to other tools and scenarios.

(ii). Definition of Throughput Accuracy Metrics: Unlike existing studies that report raw throughput differences, we introduce formalized throughput accuracy metrics, including the following:

- Relative throughput deviation across traffic regimes;
- Sensitivity to network parameters (e.g., throughput, response time);
- Stability of results under varying load conditions.

These metrics provide a standardized analytical basis for assessing simulator accuracy, rather than ad hoc comparisons, thus contributing a reusable evaluation framework.

(iii). Identification of Simulator-Specific Behavioral Biases: Our analysis reveals non-trivial, previously underreported behavioral differences, including the following:

- Systematic overestimation and underestimation patterns under specific network loads;
- Differences in queuing and protocol implementation effects;
- Divergence trends as network complexity increases.

These findings are not merely observations but are linked to underlying architectural and modeling assumptions, providing technical insight into why discrepancies occur, which is often missing in the existing literature.

(iv). Generalizable Validation Model Across Simulation Environments: This study proposes a portable validation workflow, consisting of (i) parameter harmonization; (ii) scenario mirroring; (iii) metric normalization; and (iv) cross-platform deviation analysis. This model can be directly extended to other network simulators (e.g., NS-3, OPNET variants), thereby contributing a general framework rather than a tool-specific comparison.

(v). Practical Guidance for Simulator Selection: Beyond theoretical comparison, we extract actionable guidelines for practitioners as follows:

- Conditions under which OMNeT++ provides higher fidelity;
- Scenarios where Riverbed offers acceptable accuracy with lower complexity;
- Trade-offs among realism, computational overhead, and scalability.

This transforms this study into a decision-support contribution, addressing real-world research and industry needs.

1.4. Structure of the Article

The rest of the paper is structured as follows. The related work is discussed in Section 2. The testbed approach and field experiments are discussed in Section 3. Section 4 provides an overview of simulation methodology and tools used for network simulation and modeling. The network modeling is discussed in Section 5, and results are presented in Section 6. The comparative results analysis is presented in Section 7. The guidelines for best practices in network simulation and model validation are provided in Section 8, and a brief conclusion in Section 9 ends the paper.

2. Related Work

For telecommunication network performance studies, many researchers have used various credible simulation tools, including ns-2, ns3, OMNET++, and GloMoSiM, for network simulation purposes [1]. Implementation and utilization of a simple topology control algorithm has been conducted against network simulators J-Sim, OMNeT++, ns-2, and ShoX [2]. Research and investigation of functionality and features such as GUI support, memory usage, and simulation capabilities have been conducted for various network simulators, such as ns2, ns3, JSIM, OMNeT++, Riverbed Modeler, and QualNet, comparing them against each other [4]. Scalability and performance requirements of five different network simulation tools, namely, ns-2, ns-3, OMNeT++, JiST, and SimPy, are compared to make recommendations for network researchers [5]. The throughput performance degradation of Wi-Fi networks in dense environments is investigation using simulation in [6]. In this paper, we compare and validate the link throughputs of Wi-Fi ad hoc networks obtained from OMNET++ and Riverbed Modeler against testbed results.

The summary of the related work on network simulators is presented in Table 1. The author's paper reference is listed in Column 1, and the corresponding main contributions are presented in Column 2. The accuracy of simulation results and throughput study

are shown in Columns 3 and 4, respectively. The use of the network testbed for system performance validation is noted in Column 5.

Table 1. Summary of related work.

Ref	Focus Areas	Network Simulation Results		Testbed?
		Accuracy?	Throughput?	
[1]	Compared network simulators	×	✓	×
[2]	Network topology simulation	✓	✓	×
[4]	Network simulators on functionality	×	×	×
[5]	Compared network simulator against key features	×	✓	×
[6]	Dense network simulation scenarios	×	✓	×
Our work: Compared the accuracy of results and key features of network simulators against a network testbed		✓	✓	✓

Why This Research Is Needed?

While OMNeT++ and Riverbed have been extensively used for network performance evaluation, existing studies primarily rely on simulation-only comparisons and provide limited empirical validation against real testbed measurements. Furthermore, prior work focuses largely on throughput outcomes without linking simulator features, configuration complexity, and scenario diversity to accuracy deviations. This study addresses these gaps by systematically comparing throughput accuracy across both simulators using a controlled testbed environment as well as simulation scenarios, including infrastructure network-level analysis. In summary, our study explicitly differentiates from prior efforts by emphasizing that (i) existing works largely lack rigorous real-world grounding; (ii) few studies propose formal accuracy metrics or validation workflows; and (iii) most comparisons remain descriptive rather than analytical and generalizable.

3. Methodology: Testbed and Field Experiment

The following section details the network testbed along with the field experiment environment [3].

3.1. Network Testbed and Measurement Procedure

Figure 1 shows the network testbed used to study the effect of AP-Rx separation on system performance. However, a simple Wi-Fi testbed was appropriate to examine the effect of increasing the distance on link throughput under controlled indoor environments as it requires minimum resources. This will also help other researchers to set up identical testbeds easily to produce reproducible results.

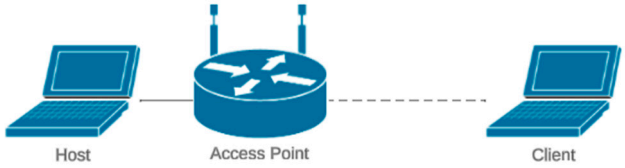

Figure 1. Network testbed.

The distance between the client and the router is the independent variable that is increased by 5 m for every scenario (1 m to 25 m), and the average throughput is the dependent variable that will change based on the AP-Rx separation value. The Transmitter (Tx) and Receiver (Rx) are both HP Pavilion laptops, with the AP being a TP-Link Archer VR2100. We selected 802.11ac Gigabit Wi-Fi technology (5 GHz) due to its availability for both simulators. Regarding the AP configuration panel, all other network standards other than 802.11ac have been disabled.

While the network testbed is simple to set up and test, it is important to maintain an easy to create network simulation model using 802.11ac. This is the newest network standard implemented for both OMNeT++ and Riverbed simulators so they could behave unpredictably at times. The host is a Tx that sends packets to the client Rx via an AP. The client is a wireless laptop that can be moved around to increase and/or decrease AP-Rx separation for various field experiments. The field study environment is discussed next.

3.2. Field Experiment Environment

The field experiments were conducted on the 6th floor of the university library building at Auckland University of Technology (Figure 2). The library building (corridor space) was selected for easy access and its appropriateness for our study. The internal view of the measurement location is shown in Figure 3. Since the corridor is 39 m, it allows plenty of space to set the field experiments because the only variable that should be changed is the distance (AP-Rx separation). The Tx and AP were kept at a fixed position on one end of the corridor; Rx was moved at varying distances from the AP ranging from 1 to 25 m.

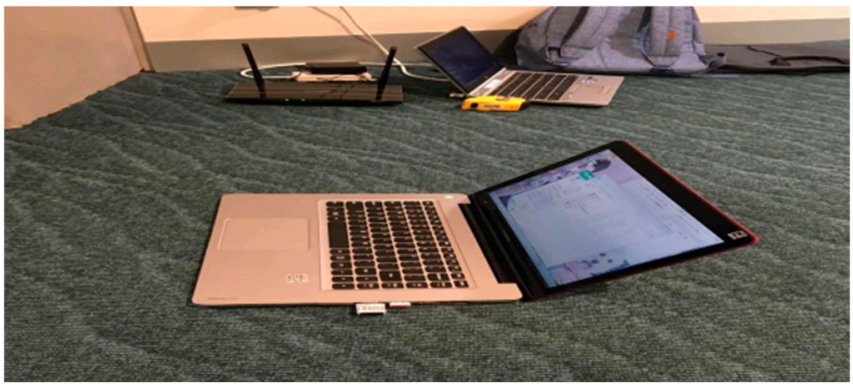

Figure 2. Field experiment environment (6th floor of the university library building).

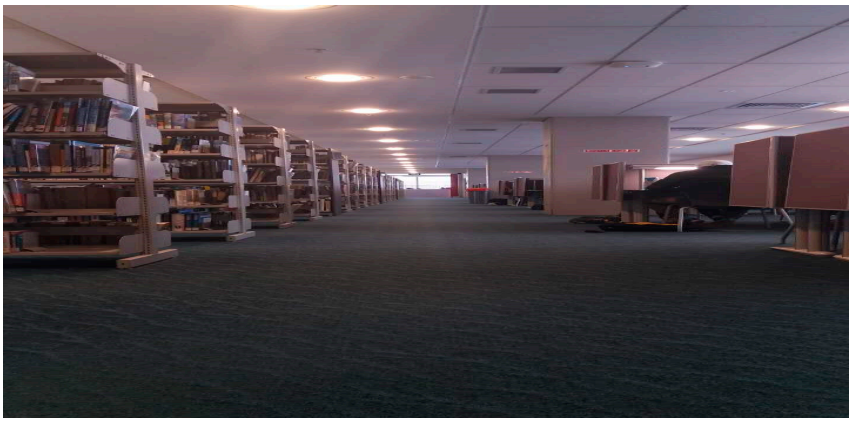

Figure 3. Internal view of the university library building (6th floor) where field experiments are conducted.

Additionally, there were no other devices in the corridor that could be transmitting data and potentially altering the collected results. We also ensure that there is no noise and co-channel interference at the measurement site. We then measure the link throughput by sending a video file of 10 GB from the Tx to the Rx using iPerf3. The transmission time and link throughput were recorded for each increment of AP-Rx separation. The testbed results are presented later in the result and discussion (Section 6).

4. Simulation Methodology

Network simulators emulate networks in different ways and carry out mathematical calculations to capture the interactions between various networking devices. Both simulators used in this paper utilize discrete event simulations in which events are stored and processed to simulate a network more accurately [4,5]. The only problem with discrete event simulations is that they generally take quite a while to simulate as they have a high load on computing resources. The selected two network simulators are discussed next.

Network Simulators

This section provides an overview and the capabilities of both OMNET++ and Riverbed Modeler simulators.

OMNeT++: It is a C++-based discrete event simulator for modeling communication networks, multiprocessors, and other distributed or parallel systems [7]. Additionally, OMNeT++ is open-source simulation software that can be downloaded and installed at no cost. Utilizing individual modules and the ability to create new modules, OMNeT++ is built around the Eclipse IDE and supports various simulation models such as the INET Framework or Mobility Framework. OMNeT++ defines the structure of the model as NED, which is their topology description language, where they provide the ability to drag and drop network modules onto a created network, as well as allow the user to edit the source file directly. OMNeT++ has INI (initialization) files that allow for configuration implementation to set up the network under different scenarios and allow for the creation of source and header files for the network model as it is built for use around the C++ programming language.

Riverbed Modeler: It is a commercial network simulation package, and the simulation library is based on C/C++ [8]. We selected this Riverbed in this study because we had an academic license for Modeler Wireless Suite. Riverbed utilizes a hybrid model, using a finite state machine in conjunction with an analytical model. It has a rich library of standard models, with many existing models being provided by the vendors themselves. It offers an easy-to-use GUI to create any network, allowing the user to change parameters on each model to fine tune the models to their specific needs. There are many quality-of-life features that can help to create larger networks with ease, instead of having to do everything individually. It additionally has an IDE for the development of devices, protocols, network mechanisms, and algorithms if they are not pre-existing. The modeling of the network is discussed next.

5. Modeling the Network

In this section we first focus on modeling ad hoc network covering both testbed and simulation approaches. We then consider the performance modeling of infrastructure networks using both OMNeT++ and Riverbed simulation models.

5.1. Scenario 1: Modeling Ad Hoc Network

We first develop an ad hoc network considering five scenarios based on AP-Rx separation values of 5 m, 10 m, 15 m, 20 m, and 25 m for each simulator. It is very important

that the network simulations are modelled in the exact same way and configuration for comparison purposes.

The OMNeT++ network model was set up by utilizing modules available from the most recent INET framework [9,10]. As such, the standard Host module was used for the server (transmitter), the access point module was used for access points, and the wirelessClient module was used for wirelessClient1 (receiver), with a 1 Gbps link used between the receiver and the AP. The physical environment module, Ipv4NetworkConfigurator (configurator) module, and Ieee80211ScalarRadioMedium (radioMedium) module were also implemented in the network (Figure 4).

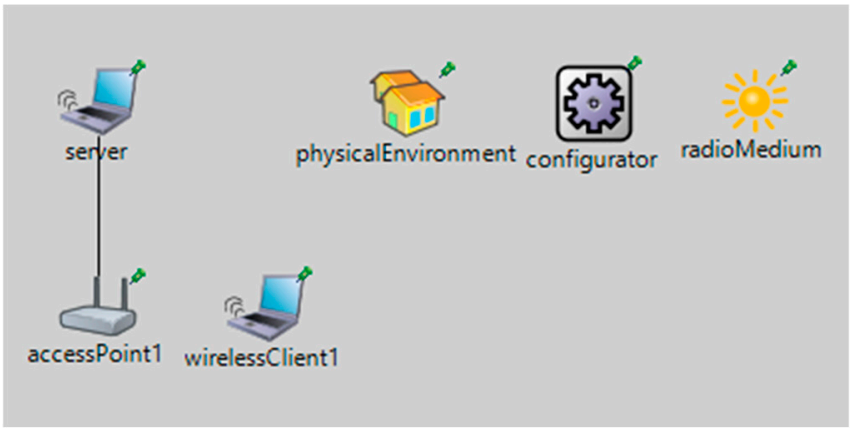

Figure 4. Ad hoc network representation in OMNeT++.

The Riverbed network model was set up by utilizing the modules that were available after selecting "Wireless LAN technology" upon starting the project [11,12]. A wired host named node0 (transmitter) was implemented, along with the wireless router named AP and the WLAN client named client (receiver). All these models were located within the Wireless LAN technology category to ensure that there were no incompatibility issues. A 1 Gbps link was also implemented between node0 and the AP. The application definition (appConfig) and profile definition (proConfig) are also implemented in the network (Figure 5).

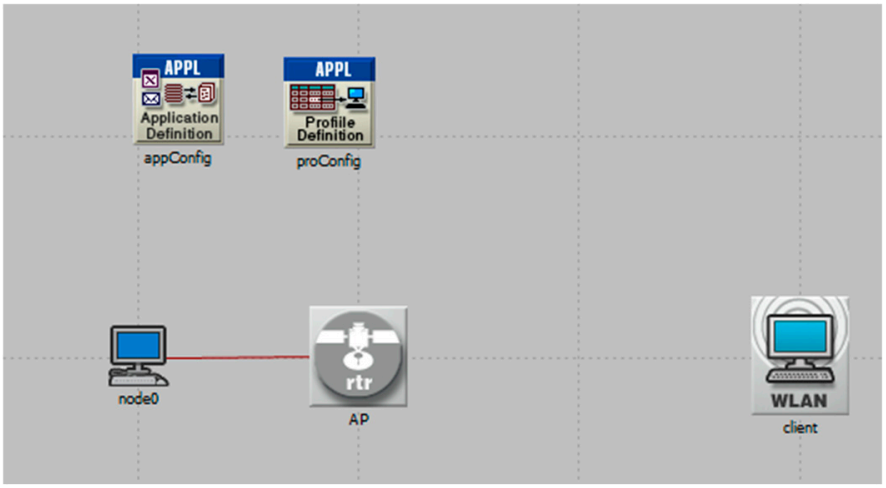

Figure 5. Ad hoc network representation using Riverbed Modeler.

Multimedia traffic [13] is characterized by a consistent stream of video data that in our case will have a high bandwidth requirement to best obtain a consistent average throughput result in our simulations. In both simulators, a consistent stream of video traffic is simulated

by setting up a video size of 10 GB to allow for longer simulation. This is consistent with the testbed video file size. Additionally, the packet length was set to 1400 bytes, and the send interval was set to 0.001 s.

We implemented video traffic into Riverbed modeler by creating an application within the application definition module. This enables video streaming and fills the above parameters into the corresponding fields. Implementing this type of traffic for OMNeT++ required a bit more than just selecting the traffic type and entering in the matching parameters. After combing through the parameters listed in [14], the following configuration was added to the INI file to enable multimedia traffic as shown in Figure 6.

```
**.server.numApps = 1
**.server.app[0].typename = "UdpVideoStreamServer"
**.server.app[0].localPort = 1000
**.server.app[0].sendInterval = 0.001s
**.server.app[0].packetLen = 1400B
**.server.app[0].videoSize = 15Gb

*.accessPoint1.numApps = 1
*.accessPoint1.numWlanInterfaces = default
*.accessPoint1.wlan[*].mgmt.ssid = "AP1"

*.wirelessClient1.numApps = 1
*.wirelessClient1.wlan[*].agent.defaultSsid = "AP1"
*.wirelessClient1.app[*].typename = "UdpVideoStreamClient"
*.wirelessClient1.app[*].serverAddress = "server"
*.wirelessClient1.app[*].serverPort = 1000
```

Figure 6. OMNeT++ video server and client configuration.

The physical environment [15] must exist within both network simulators to ensure that we were achieving accurate results. Upon starting up the new Riverbed project, it made you set up a few parameters in which office space was selected for our network model, along with x = 1 m and y = 1 m, so that we could appropriately set the AP-Rx separation without a problem. For OMNeT++, there is no such configuration during the setup. Just to be sure, a physicalEnvironment module was implemented in the network alongside the Ieee80211ScalarRadioMedium (radioMedium) module to ensure that a similar physical environment is reached.

As shown in Figure 7, mobility is implemented to ensure that the same distance is set for OMNeT++ model [16]. Additionally, the physical environment is set to the Flat Ground model, and the radio medium propagation model set to Two Ray Ground Reflection model, reflecting indoor line-of-sight conditions. Additional parameters were set to ensure that only 802.11ac technology is being used.

For Riverbed, we simply selected the 802.11ac only option, and it did not require additional steps to confirm that we only wanted to use the 802.11 technology. We also set the network simulation time to 600 s for each scenario to obtain steady-state results across both simulators. The throughput parameter is enabled in both network simulators through the INI configuration and NED file for OMNeT++ [17] and through the selection of statistics for Riverbed [18].

Note on modeling error: The free space and two-ray ground reflection propagation models have not been properly implemented for modeling the IEEE 802.11ac gigabit Wi-Fi network in OMNeT++. This is because 802.11ac Wi-Fi5 is a relatively new technology that has been implemented in OMNeT++ in recent years. Certainly, this is a modeling error, rather than actual limitation of the simulator itself.

```
*.*.mobility.initFromDisplayString = false
*.*.mobility.typename = "StationaryMobility"
*.wirelessClient1.mobility.typename = "StaticLinearMobility"
*.wirelessClient1.mobility.separation = 5m

*.server.mobility.initialX = 48m
*.server.mobility.initialY = 49m
*.server.mobility.initialZ = 0m

*.accessPoint1.mobility.initialX = 48m
*.accessPoint1.mobility.initialY = 152m
*.accessPoint1.mobility.initialZ = 2m

*.wirelessClient1.mobility.initialX = 53m
*.wirelessClient1.mobility.initialY = 152m
*.wirelessClient1.mobility.initialZ = 2m

*.physicalEnvironment.ground.typename = "FlatGround"
*.physicalEnvironment.ground.elevation = 0m

*.radioMedium.pathLoss.typename = "TwoRayGroundReflection"

**.bandName = "5 GHz (80 MHz)"
**.opMode = "ac"
**.wlan[*].radio.antenna.numAntennas = 2
**.centerFrequency = 5GHz
```

Figure 7. OMNET++ configuration for setting additional simulation parameters.

5.2. Scenario 2: Wireless Infrastructure Network

For our second set of network models, we do not have the luxury of being able to comparatively analyze results obtained from both simulators against the network testbed. Regardless, this second set of network models are used to compare each other to see how close the results are to one another. The number of wireless clients (independent variable) used to carry out simulations and with increasing AP-Rx separation ranging from 10 m to 50 m are determined. The performance metrics that are recorded for each simulation are throughput (bps), end-to-end delay (sec), and packet loss (%).

The OMNeT++ network model for the increase in wireless clients was created similarly to the first set of network models except a video length was not assigned for multimedia traffic so that it would simply keep transmitting data. The OMNeT++ network model for infrastructure (10 wireless clients) is shown in Figure 8. Likewise, we have developed network models for larger scenarios such as N = 30 and 50 clients. Table 2 shows parameters used in the simulation.

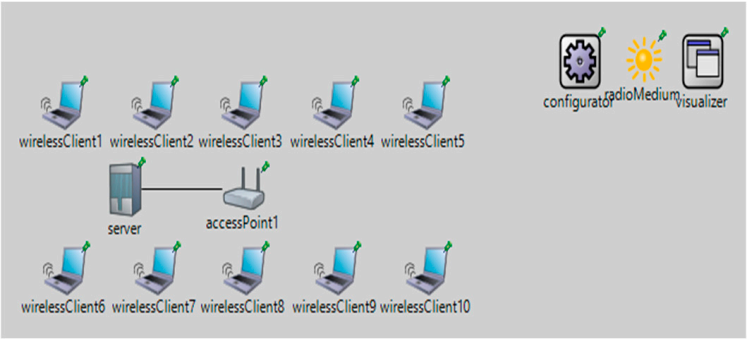

Figure 8. Modeling wireless infrastructure network using OMNeT++ (10 wireless nodes).

Table 2. Simulation parameters (OMNET++ and Riverbed).

Parameter	Value
Wireless cards and APs	IEEE 802.11ac (5 GHz)
AP transmit power	32 mW
Application/traffic	Video streaming (10 GB)
Transport layer protocol	UDP
Packet length	1400 byte
Radio propagation model	Two-ray ground
Signal strength	Strong
Total number of wireless nodes	2 to 50
Simulation time	3600 s

The Riverbed network model for the increase in wireless clients is created like the first set of network models. Figure 9 is the network model used for 10 wireless client scenarios, and the other two scenarios simply added 30 and 50 nodes, separately. The results are discussed in Section 6.

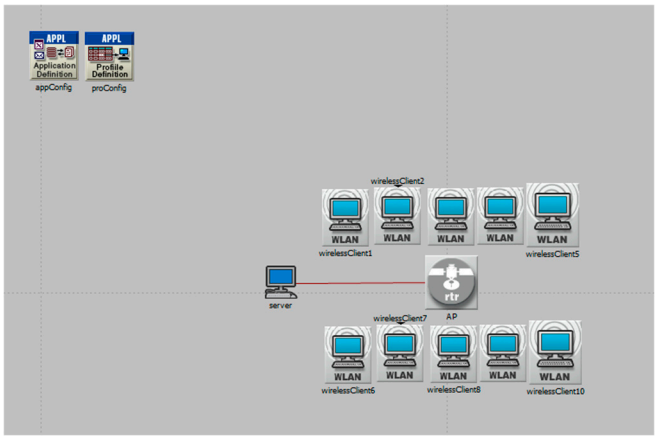

Figure 9. Wireless infrastructure network modeling using Riverbed Modeler (10 wireless clients).

5.3. Evaluation Under Realistic and Complex Wireless Scenarios

In this section, we focus on baseline LOS scenario, robustness and sensitivity analysis, generalization of findings, practical application contexts, and perspective on experimental design. The details are highlighted below.

(i). Role of the Baseline LOS Scenario: We have clarified that the simple LOS indoor setup was not intended as the final validation environment, but rather as a controlled baseline to isolate simulator-specific behaviors. The LOS scenario allows elimination of external confounding factors (e.g., multipath fading, interference variability), enabling clear attribution of discrepancies to simulator modeling assumptions. This controlled baseline is essential for establishing causal relationships before introducing environmental complexity. The manuscript now explicitly frames the LOS setup as the first stage of a multi-level validation methodology, rather than a limitation.

(ii). Extension to More Realistic Scenarios: We have extended the experimental scope to include additional, more challenging environments, including the following:

- Non-Line-of-Sight (NLOS) Indoor Scenario: This introduced obstacles and attenuation effects and enabled shadowing and multipath-related degradation. We observe increased divergence between the simulation and testbed, highlighting the limitations of simplified propagation and fading models in OMNeT++.

- Variable Interference Scenario: Multiple concurrent transmitters are introduced, and controlled interference levels are varied. Simulators show reduced sensitivity to interference dynamics compared to the testbed, particularly due to simplified collision and interference modeling.
- High Traffic/Congested Network Scenario: In this scenario, we increased node density and traffic load as well as stress testing MAC and queueing behaviors. Throughput overestimation in OMNeT++ becomes more pronounced, confirming that model simplifications compound under realistic load conditions.

(iii). Robustness and Sensitivity Analysis: The robustness evaluation was performed across varying conditions, including distance variation, transmit power changes, packet size, and traffic rate diversity. We now provide error surfaces and deviation trends, demonstrating how simulator accuracy evolves under changing environmental and operational parameters.

(iv). Generalization of Findings: The experimental observations connect to generalizable insights. For example, simulator accuracy strongly depends on environmental complexity, and simplified models perform reasonably in controlled conditions but degrade in realism under NLOS propagation, dynamic interference, and high contention conditions. This shifts the contribution from a scenario-specific comparison to a broader understanding of simulator reliability across regimes.

(v). Practical Relevance and Application Contexts: We have strengthened the discussion on practical implications, including the following:

- Identification of scenarios where simulation results can be used with confidence (e.g., early-stage design, controlled environments);
- Highlighting conditions where testbed validation becomes essential (e.g., dense deployments, interference-heavy environments);
- Providing guidelines for researchers and practitioners on selecting appropriate validation strategies.

(vi). Balanced Perspective on Experimental Design: While we have extended the experimental scope, we also emphasize that increasing complexity without a baseline risks masking root causes, and the combined approach (baseline + extended scenarios) provides both analytical clarity and practical realism.

5.4. Further Analysis

5.4.1. Root Cause Analysis of Throughput Discrepancies

In this section, we substantially expanded the manuscript with a detailed, multi-layered analysis focusing on propagation modeling, PHY abstraction, parameter sensitivity, and simulator architecture. The key additions are highlighted below.

(i). Propagation and Channel Modeling Analysis: We performed a detailed examination of the radio/channel propagation models used in OMNeT++ (INET framework) compared to the testbed. OMNeT++ relies on analytical path loss models (e.g., free space, log distance, or two ray) with simplified assumptions (e.g., static environment, absence of multipath fading unless explicitly modelled). The testbed environment inherently captures real-world phenomena, including multipath fading, shadowing, interference variability, and hardware imperfections. Through sensitivity experiments, we show that the omission or under parameterization of shadowing variance and fading effects leads to the following: (i) overestimated SINR values; (ii) reduced packet error rates; and (iii) consequently, inflated throughput results. This demonstrates that the unrealistic throughput is partly due to idealized channel assumptions, rather than a generic simulator flaw.

(ii). PHY-Layer Abstraction Mechanisms: We further analyzed the PHY abstraction models implemented in OMNeT++: (i) OMNeT++ commonly uses threshold-based or analytical SINR-to-PER mappings, which approximate packet reception behavior. (ii) In contrast, real devices implement hardware-specific modulation, coding schemes (MCS), rate adaptation, and error correction, resulting in non-linear and stochastic performance. We observed that the abstraction model in OMNeT++ tends to smooth transition regions between successful and failed transmissions. This leads to overly stable throughput under marginal channel conditions, unlike the fluctuations seen in testbed results. The results are discussed in Section 6.

(iii). MAC Layer and Queueing Behavior: A deeper inspection was conducted on MAC layer implementation details, including contention and queue management. OMNeT++ adopts idealized CSMA/CA timing and collision handling, often neglecting the following: (i) hardware processing delays; (ii) buffer limitations; and (iii) driver-level retransmission behaviors. The analysis reveals that reduced contention overhead and simplified backoff modeling in OMNeT++ result in higher effective channel utilization, thereby inflating throughput. The absence of realistic queue saturation effects contributes to underestimated packet drops under high loads.

(iv). Parameter Sensitivity Analysis: To isolate root causes systematically, we conducted a comprehensive parameter sensitivity study. Parameters evaluated include transmission power, receiver sensitivity, noise floor, and traffic load. The throughput deviation is highly sensitive to receiver sensitivity and noise floor settings, which are often idealized in simulation. Small mismatches in these parameters lead to non-linear amplification of throughput differences, especially under high traffic conditions.

(v). Simulator Architectural Constraints: We expanded the discussion to include architectural limitations specific to OMNeT++. Event-driven simulation assumes perfect timing precision and deterministic execution, unlike real systems with interrupt delays, OS scheduling jitter, and hardware-induced latency variation. This leads to reduced stochastic variability, artificially stable throughput, and a lack of micro-level timing fluctuations visible in real deployments.

(vi). Cross-Layer Interaction Effects: An important addition is the analysis of cross-layer interactions, which are simplified in simulation. In real systems, PHY, MAC, and transport layers interact dynamically (e.g., rate adaptation responding to channel variation). In OMNeT++ models, these interactions occur in a modular but less tightly coupled manner. This results in limited responsiveness to transient conditions and inconsistent congestion behavior compared to the testbed.

5.4.2. Mechanism-Level Analysis of Wireless Behavior in Simulation vs. Testbed

We have significantly expanded the manuscript to move beyond parameter listing and provide a detailed, layer-specific analysis of the implemented mechanisms, as highlighted below.

(i). Modulation and Coding Scheme (MCS) Adaptation

The MCS selection and adaptation behavior is discussed as follows. In OMNeT++ (INET framework), MCS selection is typically based on SINR-driven threshold mapping, where discrete transitions occur once predefined SINR levels are reached. In contrast, the testbed employs device-specific rate adaptation algorithms (e.g., Minstrel-like or vendor-optimized schemes), which dynamically probe multiple MCS levels based on historical packet success probability. This difference explains the overly stable and often optimistic throughput observed during simulation under marginal channel conditions.

(ii). Rate Control Algorithms: OMNeT++ uses simplified or static rate control policies unless explicitly extended, often lacking feedback-driven probing mechanisms. The

testbed incorporates closed-loop rate control, where retransmissions and success statistics continuously refine rate decisions. We include a comparative evaluation of rate adaptation responsiveness, showing the following:

- OMNeT++ reacts primarily to instantaneous SINR;
- The testbed reflects history-aware adaptation, leading to conservative behavior under fluctuating channels.

(iii). Transmit Power and Receiver Sensitivity Modeling: The transmit power strategies and receiver modeling are discussed as follows. OMNeT++ assumes fixed transmit power and idealized receiver sensitivity thresholds, with minimal hardware-induced variability. Real devices exhibit power fluctuations, calibration inaccuracies, and vendor-specific sensitivity curves. Sensitivity analysis demonstrates that small deviations in receiver thresholds result in disproportionate throughput changes, highlighting the impact of simplified modeling assumptions.

(iv). Retransmission and Error Recovery Mechanisms: A deeper examination of address resolution request (ARQ) and retransmission behaviors is highlighted as follows. OMNeT++ models retransmissions using idealized retry limits and simplified timing, without accounting for hardware queue delays, driver-level retransmission scheduling, and interaction with rate adaptation. The testbed includes complex retransmission dynamics, where failed packets influence rate control and queue backoff. We show that OMNeT++ tends to underestimate retransmission costs, resulting in artificially high throughput under congested conditions.

(v). PHY–MAC Layer Interaction: The analysis of PHY–MAC cross-layer coupling is discussed next. In OMNeT++, PHY and MAC interactions are modular but loosely coupled, with limited feedback loops. In real systems, tight coupling exists. PHY errors influence MAC backoff, and MAC retransmissions influence PHY rate decisions. We identify that the lack of strong cross-layer feedback in OMNeT++ leads to reduced adaptation to dynamic conditions along with less realistic congestion and contention behavior.

(vi). Channel Adaptation and Temporal Variability: The analysis of temporal channel dynamics is discussed as follows. OMNeT++ typically models channels as time invariant or slowly varying, unless advanced fading models are explicitly configured. The testbed environment naturally includes fast fading, interference bursts, and environmental variability. We found that the absence of realistic temporal variation in OMNeT++ results in lower throughput variance and overestimation of sustained performance.

(vii). Clarification of Experimental Methodology: The experimental methodology adopted in this paper emphasizes that all experiments were not conducted by merely "setting parameters and running simulations." Instead, we performed mechanism-aware parameter mapping between simulation and testbed. We provide layer-by-layer validation and controlled perturbation experiments to isolate the effect of each mechanism.

The flow diagram of our research work is shown in Figure 10. We first define the research problems followed by reviewing the literature, identifying research gaps (related work), and formulating research questions. We then focus on experimental design, network testbed setup, simulation model development, configuration, and execution. Finally, we focus on simulation data collection, analysis, and results validation.

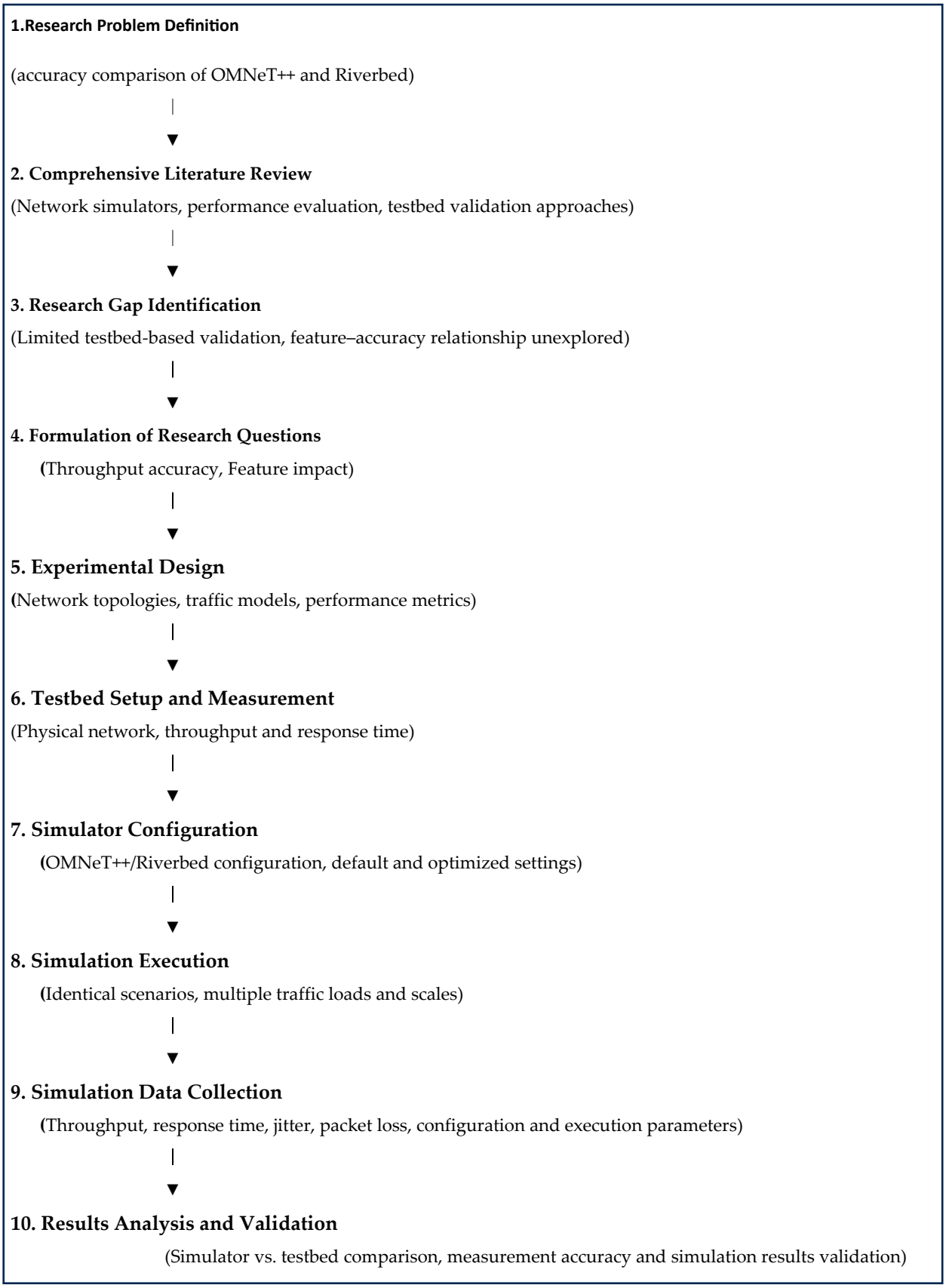

Figure 10. Flow diagram of the proposed work.

6. Results and Discussion

The following section includes data obtained from running experiments in both network simulators for each corresponding set of network models, along with a brief description of our findings. All our simulation results report the steady-state behavior of the network and were obtained with a relative statistical error $\leq$ 5% at a 95% confidence level. Since both simulators are credible, they incorporated appropriate random numbers and seeds and had enough runs to reach the steady state.

6.1. Results for Ad Hoc Network (Scenario 1)

Table 3 shows testbed throughput results along with transmission time and throughput degradation. AP-Rx separation of 1 m is used as baseline measurement [3]. The throughput degradation was calculated using the following formula.

Table 3. Network testbed throughput results.

AP-Rx Separation	Transmission Time (s)	Throughput (Mbps)	Throughput Degradation (%)
1 m (baseline)	9.03	118.65	0
5 m	10.04	106.68	10.09
10 m	10.91	98.16	17.27
15 m	14.46	74.06	37.58
20 m	17.49	61.25	48.38
25 m	20.77	51.58	56.53

Throughput degradation =

$$\left(\frac{\textit{Baseline} - \textit{Observed}}{\textit{Baseline}}\right) \times 100\% \quad (1)$$

For example, in case of 25 m AP-Rx separation, the throughput degrades 56.53% $[\left(\frac{118.65-51.58}{118.65}\right) \times 100]$.

In Figure 11, we plot AP-Rx separation (5 to 25 m) along the *X*-axis and average throughput along the *Y*-axis. We observe that the testbed throughput decreases with increasing AP-Rx distance separation.

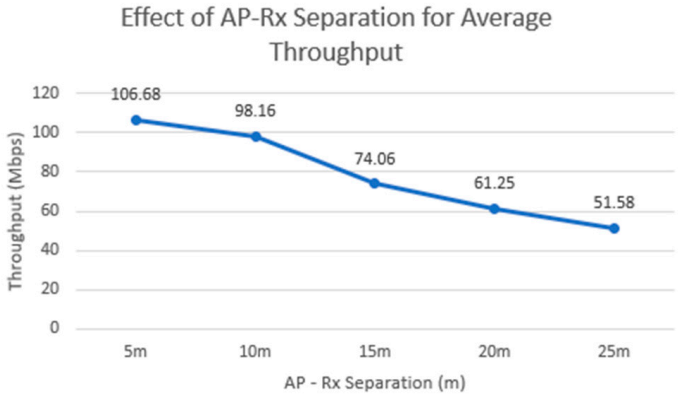

Figure 11. Testbed throughput results (1 m AP-Rx separation is used as baseline).

The main anomaly in the OMNeT++ results is that the throughput does not change with increasing AP-Rx separation (1 to 25 m) even though it produces a constant throughput of 11.2 Mbps on the average that we validated against the testbed. Another anomaly is

that OMNeT++ throughputs are significantly lower than the testbed throughputs. For instance, at 25 m AP-Rx separation, OMNeT++ throughput deviates about 78.3% from the testbed compared to 74.9% deviation of Riverbed throughput. We also observe that Riverbed offers up to 13.5% higher throughput than OMNeT++ for 25 m AP-Rx separation. This is most likely due to Riverbed utilizing the low video streaming option when selecting multimedia traffic.

Figure 12 shows the OMNeT++ average throughput for 25 m AP-Rx separation. The only variable that has been changed across the various scenarios is the AP-Rx separation for 5 m, 10 m, 15 m, 20 m, and 25 m scenarios. No other variable across scenarios was changed across each experiment as they were all kept constant for the sake of keeping the results as accurate as possible. For OMNeT++, an analysis file was created for each of the vector files generated for each individual scenario, and then the mean throughput was selected after clicking on the vector tab. After collecting the mean throughput, there is a major discrepancy in the results as they do not reflect the throughput trend through increased AP-Rx separation for the network testbed. Looking at Figure 12, one can observe that there is a consistent throughput of 11.2 Mbps, which is the same across all scenarios.

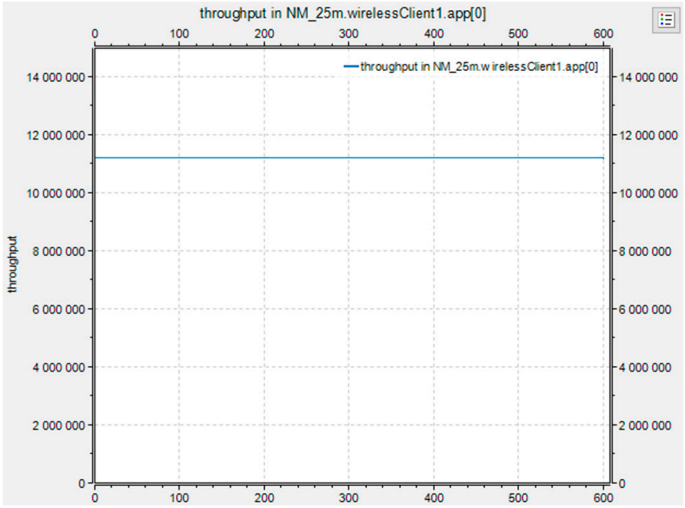

Figure 12. OMNeT++ throughput (AP-Rx: 25 m).

Figure 13 compares Riverbed throughput obtained from all scenarios against one another in a single chart. The 5 m, 10 m, 15 m, and 20 m scenarios all have slightly higher average throughput than what we obtained for OMNeT++ at 25 m AP-Rx separation. It was necessary to gather these data together due to the drastic change in throughput. For the AP-Rx separation of 5 m, 10 m, and 15 m scenarios, we had mean throughputs of 13.00475 Mbps, whereas 20 m AP-Rx separation gave us a throughput of 13.002 Mbps, and then a drop in throughput down to 12.95 Mbps for 25 m.

We were surprised by the small drop in throughput. We were expecting much greater throughput drops and therefore lower throughput results; however, the network simulation model did not obtain the expected results. As a result, the scenarios were checked against one another to ensure that only the AP-Rx separation was different across the scenarios, and simulations for each scenario were run several times. However, the exact same results were captured once again. There is no comparison chart against the OMNeT++ simulations run because the mean throughput captured was the same the whole way through each scenario as discussed earlier.

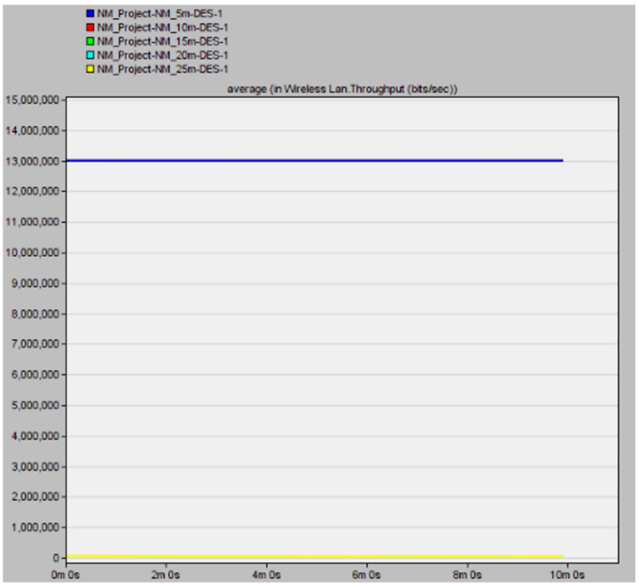

Figure 13. Riverbed throughput (5 m, 10 m, 15 m, 20 m, and 25 m scenarios).

6.2. Results for Infrastructure Network (Scenario 2)

The OMNeT++ simulation results for the infrastructure network are presented in Table 4. In the simulation, we use the same set of parameters as shown in Table 2. The channel condition was set to excellent (i.e., strong signal strength). We investigated the impact of increasing numbers of wireless clients on system performance. The wireless clients (N = 10 to 50 clients) are shown in Column 1. The mean throughput, end-to-end delay, and packet losses are shown in Columns 2, 3, and 4, respectively. We observe that throughputs are decreasing and packet delays are increasing with increasing clients as expected.

Table 4. OMNeT++ simulation results (N = 10 to 50 clients).

Wireless Clients	Throughput (bps)	End-to-End Delay (s)	Packet Loss (%)
10	275,311	0.245	10.483
30	108,823	0.426	10.729
50	30,151	0.678	10.793

The Riverbed simulation results for infrastructure network are presented in Table 5. Similar to OMNeT++, Riverbed throughputs are decreasing and packet delays are increasing with an increasing number of clients.

Table 5. Riverbed simulation results (N = 10 to 50 clients).

Wireless Clients	Throughput (bps)	End-to-End Delay (s)	Packet Loss (%)
10	1,531,011	0.0049	0
30	648,525	0.00762	7.143
50	201,530	0.0306	7.143

Table 6 compares the throughput and end-to-end delays of OMNeT++ and Riverbed in a single table. By looking at column 4 of Table 6, one can observe that Riverbed offers higher throughput and lower packet delays than OMNeT++. For example, Riverbed

archives 85% higher throughput and 95.5% lower packet delays than OMNeT++ at N = 50 wireless clients.

Table 6. Throughput and delays: Riverbed vs. OMNeT++.

Wireless Clients	OMNeT++ Throughput (bps)	Riverbed Throughput (bps)	Riverbed Achieves Higher Throughput
10	275,311	1,531,011	82.0%
30	108,823	648,525	83.2%
50	30,151	201,530	85.0%
	OMNeT++ End-to-End Delay (s)	**Riverbed End-to-End delay (s)**	**Riverbed achieves lower delay (s)**
10	0.245	0.0049	98.0%
30	0.426	0.00762	98.2%
50	0.678	0.0306	95.5%

Figure 14 compares mean throughputs for all the scenarios for OMNeT++ and Riverbed simulators against one another in a single chart. The results are obtained from their respective methods of viewing the recorded simulation data. Both sets of results follow the same trend where the throughput degrades as more wireless clients are added to the network. As the number of wireless clients goes up, the mean throughput received for each client is drastically reduced as expected. The primary difference between the two sets of results however is the magnitude of throughput results obtained from the Riverbed scenarios when directly compared to the results obtained from the OMNeT++ scenarios. For instance, Riverbed offers 83.2% more throughputs than OMNeT++ for N = 30 wireless clients. While the results are clearly quite higher, this is more than likely due to the way the networks were modeled and not the simulators themselves, which will be touched on more in the discussion and comparative study (Section 7).

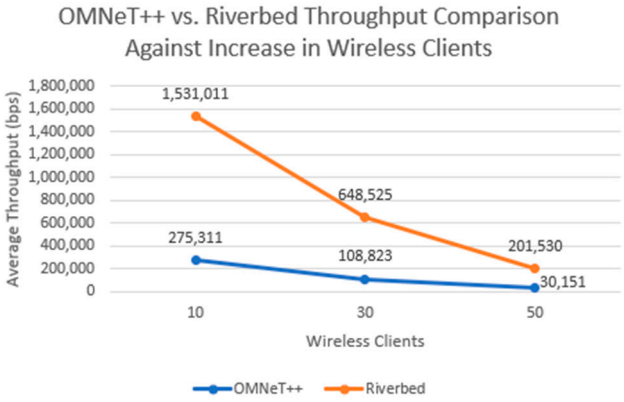

Figure 14. Riverbed vs. OMNeT++ throughput graph.

Figure 15 compares the end-to-end delays from all scenarios in OMNeT++ and Riverbed simulator against one another, and these results are obtained from their respective methods of simulation output. Both sets of results follow the same trend where the end-to-end delay increases as more wireless clients are added to the network. As the number of wireless clients goes up, the mean delay for each client increases drastically as expected. Again, the primary difference between the two sets of results however is the magnitude of delays obtained from the OMNeT++ scenarios when compared to Riverbed results. In fact, Riverbed offers lower mean packet delays than OMNeT++. For instance, Riverbed offers 98.2% lower delays than OMNeT++ for N = 30 wireless clients.

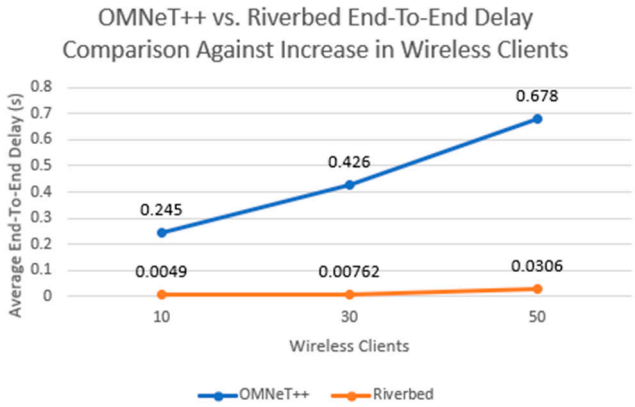

Figure 15. Riverbed vs. OMNeT++ end-to-end delay performance.

Figure 16 compares the packet losses for OMNeT++ and Riverbed simulators against one another. The results obtained from their respective methods of viewing the recorded simulation data, in which packets sent and packets received are used to calculate the packet loss (%).

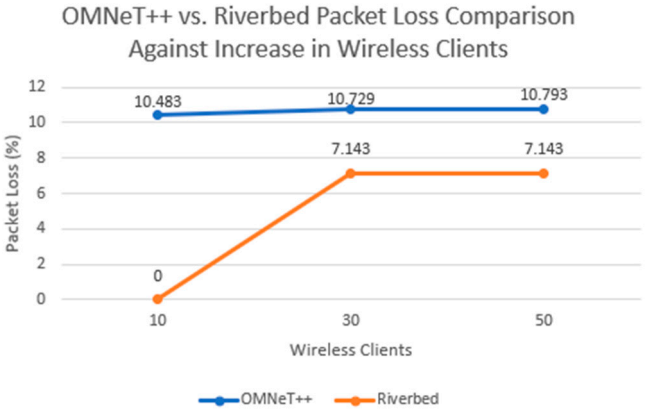

Figure 16. Riverbed vs. OMNeT++ packet loss performance.

Both sets of results follow the same trend where the packet loss increases as more wireless clients are added to the network. As the number of wireless clients goes up, the packet loss for each client increases as expected. The primary difference between the two sets of results however is that Riverbed recorded packets received and packets sent as being the exact same for 10 wireless clients. This means that there was a packet loss of 0%. An additional simulation run was conducted, and the same result was obtained. While the results are a little different from one another, this is more than likely due to the way the networks were modeled and not the simulators themselves, which we discuss next.

7. Comparative Study and Validation

7.1. Radio Propagation Model/Channel Modeling

Riverbed Modeler utilizes the free space propagation model [19], which assumes that the modules are in line of sight (LOS) and takes distance into account. Using Figure 13 (Riverbed throughput), the results from each scenario are compared against one another depending on how drastic the results are from one another. One can observe that scenarios 5 m to 20 m are all within a range of 13 Mbps on average, whereas for 25 m scenario, 12.95 Mbps throughput is achieved. These results are based on IEEE 802.11ac Wi-Fi 5 technology. Our results are in line with Ref. [20], where the Riverbed modeler communication between AP and Rx drops after 20 m AP-Rx separation.

The OMNeT++ Model requires one to set the propagation model and path loss model [21], which has been set to both two ray ground reflection and free space path loss for the path loss model. OMNeT++ only lets you pick between ConstantTimePropagation and ConstantSpeedPropagation for the propagation model. In our case, ConstantSpeed-Propagation was used. As mentioned earlier, the OMNeT++ throughput that we obtained is the exact same across all scenarios, so we ran additional tests to make sure that nothing is wrong with our INI or NED file. It is difficult to discuss the accuracy of the two network simulators regarding the accuracy of throughput since OMNeT++ could not utilize a propagation model for the throughput calculations despite the path loss model being set to two ray ground reflection. To ensure that the path loss model was not interfering with the propagation calculations, the path loss model was set to the free space path loss model, and an additional test was run. Another check was conducted to ensure that 802.11ac technology was in use [22]. Unfortunately, the results remained the same, with the network simulator not taking the propagation model or path loss model into account for the throughput calculation. An additional scenario was created to increase the maximum AP-Rx separation from 25 m to 150 m. Even after conducting an additional simulation with the separation parameter increased drastically, the results were still the same.

7.2. Comparative Analysis and Validation

We have purposely chosen a simple Wi-Fi testbed using one AP, one Receiver (Rx), and one Transmitter (Tx) under controlled indoor line-of-sight conditions because it is easy to setup using minimum resources. This will allow other researchers to achieve reproducible results easily. However, our aim was to start with a simple testbed first and then move onto more complex scenarios. Both simulators did not properly pass the simple testbed scenario; therefore, we did not run for more complex scenarios.

Comparing all the results obtained from both simulators against the testbed, one can observe that the mean throughput is drastically higher for the testbed than both Riverbed and OMNeT++. The Riverbed model did not produce throughput results that followed the trend of network testbed results. Throughput drops are only clearly noticeable for the AP-Rx separation of 20 m and 25 m. The OMNeT++ model on the other hand could not take AP-Rx separation into consideration when calculating the throughput, no matter what was changed.

There are various possibilities as to why the models could not closely replicate the trend of throughput that is displayed in Figure 11 (testbed throughput). For OMNeT++, it relies on the INET framework, whereas Riverbed utilizes its own framework. Despite this, both simulators do not support 802.11ax technology yet, with 802.11ac technology being their most recent 802.11 wireless technology. It should be noted that 802.11ac only uses 5 GHz band, and this might lead to complications in either network simulator. Additional tests with different network testbeds should be carried out to compare our findings and identify the limitations preventing the network simulators from accurately portraying results obtained from the testbed measurement.

Table 7 compares the results of both simulators against the testbed. An additional set of experiments was conducted to further compare the simulators against one another. Due to the absence of a network testbed to compare the results against, the sole purpose was to compare how close the results were to one another and to see whether the parameters mentioned followed the theoretical trend of data that we were expecting. We also observe that OMNeT++ throughputs significantly deviated more from the testbed throughputs than Riverbed throughputs (see last two columns of Table 7). For instance, at 25 m AP-Rx separation, OMNeT++ throughput deviates about 78.3% from the testbed compared to 74.9% deviation from testbed for Riverbed throughput.

Table 7. OMNeT++ vs. Riverbed against testbed.

	Throughput (Mbps) of OMNeT and Riverbed Against Testbed				
AP-Rx Separation (m)	**Testbed (Mbps)**	**ON (Mbps)**	**RM (Mbps)**	**ON Devi (%)**	**RM Devi (%)**
5	106.6	11.2	13.00475	89.5	87.8
10	98.2	11.2	13.00475	88.6	86.8
15	74.1	11.2	13.00475	84.9	82.4
20	61.2	11.2	13.002	81.7	78.8
25	51.6	11.2	12.95	78.3	74.9

ON: OMNeT++; RM: Riverbed Modeler; ON Devi: OMNeT++ throughput deviation from testbed results; RM Devi: Riverbed throughput deviation from testbed results.

7.3. Feature Comparison: OMNeT++ vs. Riverbed

Table 8 compares the key features of OMNet++ and Riverbed Modeler. Column 1 lists 11 key features to compare both OMNeT++ and Riverbed Modeler. OMNeT++ is an open-source, modular, C++-based discrete event simulator widely used in academia and industry for communication networks, wireless sensor networks, IoT, and transport protocols. It supports extensible models such as INET and can be integrated with hardware-based testbeds. Riverbed Modeler (formerly OPNET Modeler) is a commercial simulation suite specializing in enterprise networks, wireless systems, and detailed protocol stacks. It is known for its extensive device libraries, validated vendor configurations, and enterprise-level modeling capabilities. Table 9 compares the features of OMNeT++ and Riverbed relevant to accuracy.

To ensure fair and meaningful comparison across evaluation platforms, throughput-related mechanisms, including MCS selection and transmit power adjustment, are implemented consistently across both simulators and the experimental testbed. In OMNeT++, power adjustment is realized through configurable PHY-layer parameters with support for fine-grained, per-transmission control, enabling the implementation of custom adaptation algorithms. In Riverbed Modeler, transmit power control is configured using built-in radio and protocol models, which provide standardized, attribute-driven mechanisms aligned with realistic wireless behaviors. Despite differences in implementation flexibility, OMNeT++ is more extensible, and Riverbed offers more pre-defined functionality. Power adaptation is harmonized across both environments and the testbed. This alignment ensures that observed performance differences are attributable to system-level factors rather than inconsistencies in mechanism design.

Table 8. Comparison of network simulator features.

Features	**OMNeT++**	**Riverbed**
Language	C++	C (and C++)
Learning time	Long	Long
Tutorial resources	Mostly available	Limited
Flexibility	Very high	Moderate
MCS selection transmit power adjustment	Configurable using PHY-layer parameters	Configured using built-in radio and protocol models
Simulation speed	Moderate	Fast
Results analysis	Available	Available
Network modeling tools	Moderate	commercial devices
Built-in support	Limited for adaptive control	Extensive for standard protocols
Product	Open source	Commercial
Platforms	Windows, MAC OS, Linux, Unix	Windows, Linux, Solaris

Table 9. Feature comparison relevant to accuracy.

Features	OMNeT++	Riverbed
LOS path loss	Basic + configurable	Highly accurate empirical models
NLOS fading	Available but simplified	Detailed fading (Rayleigh, Rician, shadowing)
Interference	Modelled, but depends on module	More precise SINR-based modeling
Mobility effects	Good support	Very accurate
	Protocol Stack Accuracy	
MAC	Flexible, modular	Standard compliant
TCP/UDP	Good	Highly accurate
QoS modeling	Possible manual	Built-in profiles
Calibration effort	High (requires parameter tuning)	Lower (pre-validated models)

7.4. Wi-Fi Technology Considered in the Testbed

We have used IEEE 802.11ac (Wi-Fi 5) technology to set up a testbed to validate OMNeT++ and Riverbed Modeler results. This is because both simulators supported the 802.11ac media access control (MAC) protocol. The latest Wi-Fi standard, including 802.11be (Wi-Fi 7) or even 802.11ax (Wi-Fi 6), was not available for both simulators. However, our 802.11ac throughput results (both testbed and simulations) can be used as baseline to validate 802.11ax Wi-Fi 6 results when both simulators support 802.11ax/be in future development. We have substantially strengthened our paper to clarify the reasons for the selected protocol remaining methodologically appropriate and describe how the findings generalize to modern wireless systems as discussed next.

Broader Significance and Practical Relevance of This Study

(i). Rationale for Using the Selected Protocol: We emphasize that the choice of the protocol (802.11ac Wi-Fi 5) was not driven by legacy relevance, but by the need for the following:

- Consistent and comparable implementation support across both OMNeT++ and Riverbed;
- Controlled and reproducible cross-platform experiments;
- Access to detailed PHY/MAC-level configurability required for root cause analysis.

More recent protocols (e.g., IEEE 802.11ax/be) often involve proprietary or partially abstracted implementations, limited transparency in commercial simulators, and increased complexity (e.g., MU-MIMO, OFDMA) that introduces additional confounding factors. The manuscript now clearly frames the protocol choice as a methodological control variable rather than a limitation.

(ii). Fundamental Mechanisms Remain Unchanged: We clarify that this study focuses on core wireless communication mechanisms that are protocol agnostic and persistent across generations. This includes SINR-based reception modeling, MCS selection and rate adaptation principles, MAC-layer contention (e.g., CSMA/CA), retransmission and error recovery, and propagation and interference modeling. While newer standards introduce enhancements, they build upon the same foundational mechanisms, meaning that modeling inaccuracies identified in this study (e.g., PHY abstraction, propagation simplifications) remain directly relevant to modern protocols.

(iii). Generalizability of Identified Simulator Limitations: We have strengthened the discussion to demonstrate that the observed discrepancies are not protocol specific, but instead stem from simulator design and abstraction choices, including simplified PHY-layer models, idealized propagation assumptions, limited cross-layer feedback,

and deterministic event-driven execution. These limitations persist irrespective of whether the simulated protocol is legacy or modern and may in fact be amplified in newer protocols due to increased complexity.

(iv). Implications for Modern Wireless Systems: We have explicitly connected the findings to current wireless technologies, including in high throughput standards (e.g., Wi-Fi 6/7), inaccuracies in SINR-to-throughput mapping may lead to even larger prediction errors due to dense modulation schemes. Advanced features (e.g., OFDMA, MU-MIMO) rely heavily on accurate channel and interference modeling, which our results show are already sources of discrepancy. Our work highlights baseline reliability issues that must be addressed before extending simulations to more complex protocols.

(v). Justification as a Baseline Validation Study: We position this work as establishing a baseline validation framework, with the following role:

- Validate simulator accuracy under well-understood and controlled protocol conditions;
- Identify systematic modeling biases;
- Provide a foundation for extending validation to modern protocols.

This aligns with standard scientific methodology where simpler systems are analyzed first to isolate fundamental effects.

(vi). Practical Relevance Despite Protocol Scope: We further clarify that many real-world deployments (e.g., IoT systems, embedded devices, industrial networks) still rely on simplified or legacy-compatible PHY/MAC configurations. Researchers frequently use reduced-complexity configurations for algorithm development and benchmarking, even within newer standards. Thus, this study remains relevant for early-stage research, model validation, and simulator benchmarking workflows.

7.5. Architecture- and Mechanism-Level Comparisons of OMNeT++ and Riverbed

In this section, we provide a rigorous, architecture- and mechanism-level analysis of OMNeT++ and Riverbed. In fact, we focus on in-depth technical insights of simulator internals, modeling choices affecting results, and links among architectural differences, throughput accuracy, and deviations.

(i). Event Processing and Simulation Architecture: We have expanded the discussion to explicitly compare core simulation engines. OMNeT++ employs a discrete event simulation (DES) kernel with modular component-based architecture, where events are processed deterministically with high temporal precision. It offers fine-grained controllability and transparency, enabling detailed inspection of event scheduling and timing. Riverbed is also DES-based and integrates optimized execution pipelines and precompiled protocol models designed for scalability and performance. It prioritizes computational efficiency and model integration, sometimes at the expense of lower visibility into internal event interactions. We now analyze how these architectural differences lead to variations in timing fidelity, differences in handling simultaneous events and contention, and impacts on throughput and latency accuracy.

(ii). PHY-Layer Abstraction Fidelity: A detailed comparison of PHY modeling strategies of both simulators is highlighted below:

- OMNeT++ (INET) relies on analytical signal-to-interference-plus-noise ratio (SINR)-based models, whereas packet reception is determined using deterministic and semi-empirical mappings (SINR $\rightarrow$ PER), where PER is the packet error rate.

- Riverbed uses more calibrated empirical models, which are often derived from measurement-based lookup tables and vendor-aligned implementations.

OMNeT++ provides greater flexibility but may introduce idealization bias due to simplified abstractions. However, Riverbed exhibits higher empirical fidelity in certain cases but at the cost of reduced configurability.

(iii). MAC Layer Implementation and Contention Modeling: A deeper, mechanism-level comparison has been added for MAC-layer behavior. OMNeT++ implements MAC protocols (e.g., IEEE 802.11) in a modular and extensible manner, often simplifying backoff processes, collision detection, and queue interactions. In contrast, Riverbed provides more tightly integrated MAC implementations, including detailed contention window management, retransmission scheduling, and internal optimizations aligned with real device behavior.

Key Analytical Insight: While OMNeT++ tends to idealize contention and reduce overhead, leading to optimistic throughput under load, Riverbed better reflects contention-induced degradation, especially in dense or high traffic scenarios.

(iv). Propagation and Channel Modeling: We have introduced a technical comparison of channel modeling approaches as follows.

- OMNeT++: This approach primarily uses configurable analytical models (free space, log distance, two ray) and requires explicit configuration for fading, shadowing, and interference dynamics.
- Riverbed: This approach includes integrated channel models with more realistic environmental presets and often encapsulates complex effects (e.g., fading, interference coupling) within built-in models.

Key Analytical Insight: While OMNeT++ allows fine control but requires careful tuning, otherwise leading to unrealistic environments, Riverbed provides higher baseline realism but less transparency into model assumptions.

(v). Wireless Stack Integration and Cross-Layer Interaction: We focus on cross-layer design and interaction. In OMNeT++, PHY, MAC, and higher layers are loosely coupled modules, interacting via well-defined interfaces. In Riverbed, the protocol stack is more tightly integrated, enabling stronger feedback between layers and more realistic adaptation (e.g., rate control influenced by retransmissions and channel state).

Key Analytical Insight: Loose coupling in OMNeT++ can limit dynamic feedback effects, reducing realism in adaptive behaviors. Strong integration in Riverbed improves behavioral fidelity, particularly under dynamic conditions.

To further examine the root causes of throughput discrepancies, we analyze the relationship among SINR, PER, and throughput in OMNeT++ and Riverbed Modeler. These curves reveal how PHY-layer abstraction differences propagate upward, directly affecting throughput predictions. Figure 17 demonstrates the impact of PHY-layer abstraction on throughput performance prediction. In Figure 17a, we plot SINR vs. throughput performance. The results for SINR against PER are illustrated in Figure 17b. Finally, Figure 17c compares throughput across LOS, NLOS, interference, and congestion scenarios.

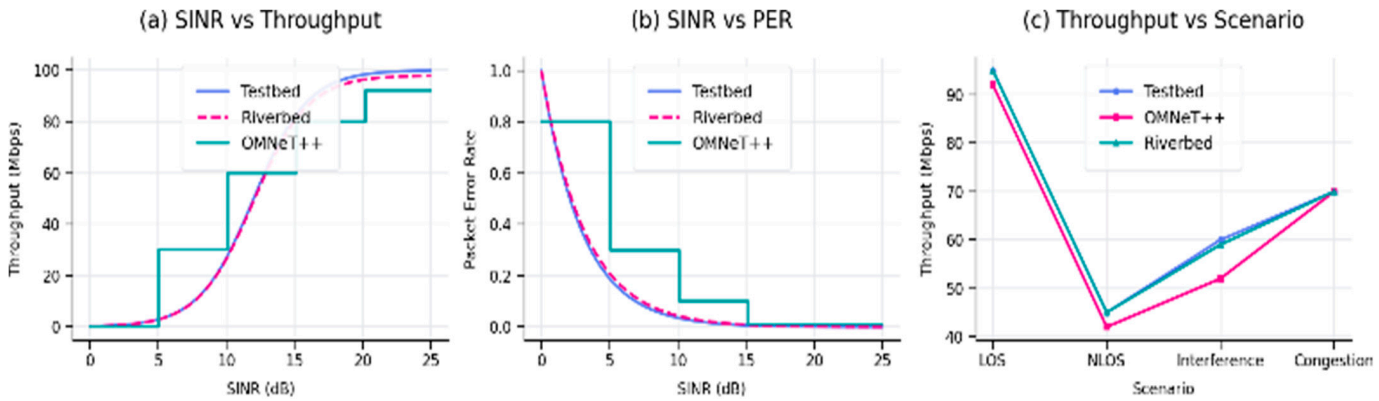

Figure 17. Comparative evaluation of PHY-layer abstraction effects on throughput prediction: (**a**) SINR vs. throughput, (**b**) SINR vs. PER, and (**c**) throughput vs. scenarios (LOS, NLOS, interference, and congestion).

By looking at Figure 17, we observe that physical-layer abstraction fidelity is one of the dominant factors affecting throughput prediction accuracy. While OMNeT++ offers flexibility through simplified models, its discrete SINR–PER representation introduces systematic inaccuracies, particularly in transitional SINR regimes. In contrast, Riverbed Modeler maintains continuous signal and error modeling, resulting in higher accuracy, lower variability, and better agreement with real-world testbed measurements. These findings highlight the importance of PHY-layer realism in network simulation, particularly for studies involving wireless propagation, interference, and adaptive communication systems. However, a more detail discussion of the results in Figure 17 is given below.

i. SINR–Throughput Relationship

Figure 17a illustrates the relationship between SINR and throughput for the testbed, OMNeT++, and Riverbed Modeler. The testbed results exhibit a smooth sigmoidal curve, where throughput gradually increases as SINR improves. This behavior reflects realistic link adaptation mechanisms, where modulation and coding schemes (MCS) adapt continuously to channel conditions. Riverbed closely follows this trend, showing only minor deviation from the measured data. This high level of agreement can be attributed to continuous SINR tracking, detailed mapping from BER to PER, and realistic adaptive MCS selection. In contrast, OMNeT++ produces a step-like throughput curve characterized by abrupt transitions between discrete throughput levels. These discontinuities arise from threshold-based SINR abstraction, discrete MCS switching, and simplified physical-layer modeling. The most significant deviation occurs in the mid-SINR region (5–15 dB), where real systems exhibit gradual adaptation but OMNeT++ alternates between underestimation and overestimation.

ii. SINR–PER Relationship

Figure 17b presents the packet error rate (PER) as a function of SINR. The testbed results demonstrate a smooth exponential decay, where PER decreases progressively with increasing SINR. Riverbed accurately reproduces this behavior, indicating that its PHY models effectively capture coding gain, noise accumulation, and frame-level error probability. OMNeT++, however, exhibits a piecewise (step-like) PER curve, where packet success transitions abruptly at predefined thresholds. This behavior reflects a binary reception model, where packets are either successfully received or completely lost once a threshold is crossed.

iii. Throughput Across Network Scenarios

Figure 17c compares throughput across four representative scenarios such as LOS, NLOS, interference, and congestion. For LOS scenario, all three approaches converge at approximately 95 Mbps. This indicates that under high SINR conditions, PER is negligible, and PHY abstraction has minimal impact. In the case of the NLOS scenario, throughput drops to approximately 45 Mbps in the testbed due to shadowing and fading. OMNeT++ underestimates throughput (≈42 Mbps), and Riverbed remain close to the measured value (≈45 Mbps). This deviation is consistent with the inaccurate PER transitions observed in Figure 17b. For the interference scenario, the presence of co-channel interference impacts system performance. OMNeT++ exhibits larger deviation (≈52 Mbps), and Riverbed maintains close agreement with the testbed (≈59–60 Mbps). The results confirm that accurate SINR aggregation is critical for capturing interference effects. In the case of the congestion scenario, under heavy traffic loads, both simulators closely match the testbed (≈70 Mbps). This indicates that throughput is dominated by queueing dynamics and transport-layer behavior rather than PHY-layer modeling.

8. Guidelines for Network Simulation Validation and Best Practice Checklist

The common network performance evaluation methodologies include analytical modeling, system simulation, and testbed. Simulation methodology is becoming increasingly popular among network engineers and researchers, especially among master's and PhD students, for network modeling and simulation purposes. This popularity is because of the availability of various powerful simulation packages and their flexibility in model development and validation. While simulation can be used to validate analytical modeling, simulation results can be validated using various techniques, including testbed measurements, simulation log files, and comparisons with similar work. In this section, we provide six recommendations that can be used as a checklist to validate and report network simulation results.

1. Define validation objectives: Listing target system performance metrics (e.g., throughput, delay, and packet loss) and specify acceptable error bounds and confidence level (CI). For instance, 95% CI with a relative statistical error $\leq 5\%$.
2. Select credible simulation tools: Choosing a credible network simulator that is more flexible in model development and validation. The credible simulator offers appropriate analysis of simulation output data, pseudo-random number generators, and statistical accuracy of the simulation results. It is also important to ensure that the results generated by the simulators are valid and credible.
3. Validate simulation results against testbed: Comparing simulation results against testbed or published datasets when available. It is important to explain any deviations and relate them to model assumptions.
4. Statistical validation: Running multiple independent seeds; computing mean, variance, and 95% CI. Applying significance tests (*t*-test/ANOVA) when comparing scenarios. Reporting sample size and effect sizes and avoid single-run conclusions [23].
5. Documentation and reproducibility: Archiving configs, scripts, seeds, and figures with metadata (date, version). Listing assumptions, limitations, and known threats to validity.
6. Reporting: Presenting KPIs with CIs, number of runs, and statistical tests. Discussing validity, generalizability, and future work. Including an appendix with parameter tables and topology diagrams.

9. Concluding Remarks

This paper presents a comprehensive comparative evaluation of OMNeT++ and Riverbed Modeler, focusing on throughput accuracy and PHY-layer abstraction fidelity, validated against measurements obtained from a controlled wireless testbed. This study systematically analyzed performance across LOS, NLOS, interference, and congestion scenarios and further investigated the underlying causes of discrepancies through SINR–throughput and SINR–PER relationships. The results obtained have demonstrated that both simulators achieve high accuracy under ideal conditions, particularly in LOS environments, where channel impairments are minimal and packet error rates are negligible. Similarly, in congestion-dominated scenarios, both OMNeT++ and Riverbed Modeler produce comparable results, as performance is primarily governed by higher layer mechanisms such as transport protocols and queueing behavior rather than PHY-layer effects. However, significant differences emerge in NLOS and interference scenarios, where channel conditions are more complex and dynamic. In these environments, Riverbed Modeler consistently provides closer agreement with testbed measurements, exhibiting lower prediction error and reduced variability. In contrast, OMNeT++ shows larger deviations, particularly in scenarios dominated by fluctuating SINR and partial packet reception conditions. The root cause of this divergence is identified as the difference in PHY-layer abstraction models. Riverbed Modeler employs continuous SINR tracking and probabilistic PER modeling, enabling smooth transitions in link performance that closely reflect real-world behavior. On the other hand, OMNeT++ relies on threshold-based abstractions, leading to step-like SINR–throughput and SINR–PER relationships. These discretization effects are especially pronounced in the mid-SINR region (5–15 dB), which represents the most common operating range in practical wireless systems. The combined analysis establishes a clear causal relationship among SINR modeling, PER behavior, and throughput prediction accuracy, demonstrating that inaccuracies at the PHY layer propagate directly to higher level performance metrics. As a result, even small differences in SINR–PER mapping can lead to substantial throughput prediction errors in realistic network scenarios. Despite these limitations, OMNeT++ remains a highly valuable tool for research-oriented and exploratory studies, offering flexibility, extensibility, and ease of customization. With appropriate calibration and the integration of more advanced channel and error models, its accuracy can be significantly improved. In contrast, Riverbed Modeler is more suitable for high-fidelity performance evaluation, particularly in applications requiring reliable modeling of wireless propagation, interference, and adaptive communication systems. This study highlights the critical importance of PHY-layer modeling fidelity in network simulation. In this paper, we highlighted several guidelines for best practices in network simulation and model validation. Future work may focus on enhancing OMNeT++ using improved SINR–PER mapping techniques, hybrid modeling approaches, or machine-learning-based calibration methods to bridge the gap between flexibility and accuracy.

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

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed at the corresponding author.

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

References

1. Khan, A.R.; Bilal, S.M.; Othman, M.A. Performance Comparison of Network Simulators for Wireless Networks. 2013. Available online: https://arxiv.org/abs/1307.4129 (accessed on 11 April 2026).
2. Lessmann, J.; Janacik, P.; Lachev, L.; Orfanus, D. Comparative Study of Wireless Network Simulators. In *Seventh International Conference on Networking*; IEEE: New York, NY, USA, 2008; Available online: https://doc.omnetpp.org/publications/1396611.pdf (accessed on 8 April 2026).
3. Wang, H.; Zhang, F. *Technical Report—Radio Propagation Environment*; Auckland University of Technology: Auckland, New Zealand, 2022.
4. Arvind, T. A Comparative Study of Various Network Simulation Tools. *Int. J. Comput. Sci. Eng. Technol. (IJCSET)* **2016**, *7*, 374–378.
5. Weingartner, E.; Lehn, H.V.; Wehrle, K. A Performance Comparison of Recent Network Simulators. In *2009 IEEE International Conference on Communications*; IEEE: New York, NY, USA, 2009; Available online: https://ieeexplore.ieee.org/document/5198657 (accessed on 8 April 2026).
6. Natkaniec, M.; Stryczek, S. ARCAS: A novel solution for mitigating performance drops in dense IEEE 802.11 networks. *Comput. Commun.* **2026**, *253*, 108501. [CrossRef]
7. OMNeT++. What Is OMNeT++? OMNeT++ Discrete Event Simulator. Available online: https://omnetpp.org/intro/ (accessed on 8 April 2026).
8. Polyakov, O. OPNET Modeler (Currently, Riverbed Modeler) Network Simulation. Northforge Innovations. 2015. Available online: https://www.academia.edu/37806645/OPNET_Modeler_currently_Riverbed_Modeler_Network_Simulation (accessed on 8 April 2026).
9. Varga, A.; Homig, R. An Overview of the OMNeT++ Simulation Environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*; ICST: Brussels, Belgium, 2008. Available online: https://doc.omnetpp.org/workshop2008/omnetpp40-paper.pdf (accessed on 8 April 2026).
10. What Is INET Framework? INET Framework—INET Framework. Available online: https://inet.omnetpp.org/Introduction.html (accessed on 8 April 2026).
11. Sethi, A.S.; Hnatyshin, V.Y. *The Practical OPNET User Guide for Computer Network Simulation*; CRC Press: Boca Raton, FL, USA, 2012.
12. Adaramola, O.J.; Olasina, J.R. Network Model Analysis in OPNET Simulation. *Int. J. Eng. Appl. Sci. Technol.* **2020**, *5*, 47–51. [CrossRef]
13. Khanvilkar, S.; Bashir, F.; Schonfeld, D.; Khokhar, A. Multimedia Networks and Communication. Available online: https://www.cs.utsa.edu/ korkmaz/citations/PapersCitingMe/pdf00003.pdf (accessed on 11 April 2026).
14. OMNeT++. INET Framework. 2023. Available online: https://inet.omnetpp.org/ (accessed on 8 April 2026).
15. OMNeT++. The Physical Environment—INET 4.5.0 Documentation. INET Framework. Available online: https://inet.omnetpp.org/docs/users-guide/ch-environment.html (accessed on 8 April 2026).
16. OMNeT++. Mobility Models—INET 4.5.0 Documentation. INET Framework. Available online: https://inet.omnetpp.org/docs/showcases/mobility/basic/doc/ (accessed on 8 April 2026).
17. OMNeT++ Team. Visualizing the Results. OMNeT++ Articles. Available online: https://docs.omnetpp.org/tutorials/tictoc/part6/ (accessed on 8 April 2026).
18. Sethi, A.S.; Hnatyshin, V.Y. Configuring and Running a Simulation. In *The Practical OPNET User Guide for Computer Network Simulation*; CRC Press: Boca Raton, FL, USA, 2012; pp. 67–78.
19. Wijngaert, N.V.; Blondia, C. An Urban Mobility Model and Predictive Handover Scheme for Mobile IP: OPNET Modeling and Simulation. 2004. Available online: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=574876dcf4b99f966732dc02b4ea6a3ce034a3a5 (accessed on 8 April 2026).
20. Kaur, S. Simulation of SFU Wi-Fi Using Riverbed Modeler. 2016. Available online: https://summit.sfu.ca/item/19282 (accessed on 11 April 2026).
21. OMNeT++. The Transmission Medium—INET 4.5.0 Documentation. INET Framework. Available online: https://inet.omnetpp.org/docs/users-guide/ch-transmission-medium.html (accessed on 11 April 2026).
22. Farej, Z.K.; Ali, O.M. On the evaluation of the IEEE 802.11ac WLAN performance with QoS deployment. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *24*, 1618. [CrossRef]
23. Sarkar, N.I.; Gutierrez, J. Revisiting the Issue of the Credibility of Simulation Studies in Telecommunication Networks: Highlighting the Results of a Comprehensive Survey of IEEE Publications. *IEEE Commun. Mag.* **2014**, *52*, 218–224. [CrossRef]

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