EPiC Series in Computing

Volume 111, 2026, Pages 123–134

Proceedings of The 21st Asia Joint Conference on Information Security

Scalability Analysis of Zero Trust Network Access in Bandwidth-Constrained Networks

Jumpei Kawahara[1], Yuya Tarutani[1], and Atsuko Miyaji[1]

The University of Osaka, Osaka, Japan
junpei.kawahara@cy2sec.comm.eng.osaka-u.ac.jp, tarutani@comm.eng.osaka-u.ac.jp, miyaji@comm.eng.osaka-u.ac.jp

Abstract

In recent years, with the spread of teleworking and the expansion of public cloud computing, traditional perimeter-based security models that separate internal and external networks have become insufficient for adequate defense. Perimeter-based security models using traditional VPNs and similar technologies have faced challenges: Once authenticated, they permit broad access to the internal network, and attacks on VPN devices directly expose the connected network to risk. Consequently, Zero Trust Network Access (ZTNA), based on the Zero Trust principle of not trusting any communication regardless of boundary presence, is gaining attention. ZTNA is recognized as a solution to the problems inherent in traditional network models by strictly controlling access for all communications, regardless of whether they originate inside or outside the organization, thereby minimizing user access.

However, ZTNA requires queries to an authorization server for every communication. In large-scale network environments, this leads to challenges such as increased processing load for access control and communication delays due to the rising number of queries. Furthermore, discussions on ZTNA have primarily focused on its concept and security model, with insufficient evaluation from the perspective of network performance.

This research constructs a ZTNA verification environment using the network emulator Mininet to evaluate network performance. Specifically, it adopts a structure using Envoy Proxy for the data plane and Open Policy Agent (OPA) for the authorization engine. As evaluation parameters, we measure the impact of the network distance (communication delay) between Envoy and OPA on the round-trip time (RTT) of packets during the authorization process, as well as changes in throughput and system load when increasing the number of concurrent authorization requests (session count).

1 Introduction

With the widespread adoption of cloud computing and teleworking, the limitations of traditional perimeter-based security models have become increasingly apparent. As an alternative, Zero Trust Architecture (ZTA)—which mandates strict authentication and authorization for every access request without inherent trust—has gained significant attention. Consequently, the deployment of Zero Trust Network Access (ZTNA), its primary implementation technology, is accelerating. A defining characteristic of ZTNA is its operational model, where a Policy

T.-T. Tsai, H.K. Kim, Y. Wang and A. Yamada (eds.), AsiaJCIS 2026 (EPiC Series in Computing, vol. 111), pp. 123–134

Enforcement Point (PEP) must query an external Policy Decision Point (PDP) before initiating any communication. Since this model necessitates mandatory round-trip communication for authorization, there are concerns regarding performance degradation, such as increased latency and frequent errors, caused by contention between authorization traffic and data traffic in resource-constrained network environments.

Regarding the implementation of ZTA, recent research has explored guidelines for its application in next-generation networks (6G) [1] and threat detection technologies integrating SDN and machine learning (ZT-SDN) [2]. The former achieves dynamic access control and identity-centric security in large-scale, heterogeneous environments, while the latter demonstrates automated and advanced security through flexible policy adaptation. However, these studies do not account for the impact of frequent authorization requests on overall network communication functionality as the number of concurrent users increases. In general, existing research focuses primarily on security model construction and threat detection, leaving the scalability of the network during the authorization process insufficiently verified.

The objective of this study is to quantitatively clarify the operational characteristics of the ZTNA authorization process and the resulting latency and load caused by frequent authorization requests in bandwidth-constrained environments. While existing studies such as ZTA-6G [1] and ZT-SDN [2] achieve high accuracy in dynamic access control and automated threat detection, the novelty of this work lies in its focus on the performance plateau triggered by the intervention of the authorization process under bandwidth limitations. As a concrete contribution, we constructed a practical ZTNA model combining Envoy Proxy as a PEP and Open Policy Agent as a PDP within a Mininet-based emulation environment. We adopted a dumbbell topology consisting of two switches and their connected nodes. Load experiments conducted under conditions of an N Mbps bottleneck link and 100 ms one-way latency demonstrated that the growth rate of successful requests slows down once the number of concurrent clients reaches $(9.6N + 1.6)$. Furthermore, we identified that the latency associated with authorization communication reaches its minimum at this point, marking an effective scalability limit where system processing capacity and network resource utilization are most balanced.

- **Construction of a Practical ZTNA Model:** We constructed a practical ZTNA model combining Envoy Proxy as a PEP and Open Policy Agent as a PDP within a Mininet-based emulation environment using a dumbbell topology consisting of two switches and their connected nodes.
- **Quantitative Scalability Analysis:** Through load experiments conducted under conditions of an N Mbps bottleneck link and 100 ms one-way latency, we quantitatively demonstrated that the growth rate of successful requests slows down once the number of concurrent clients reaches $(9.6N + 1.6)$.
- **Identification of Performance Limits:** We identified that the latency associated with authorization communication reaches its minimum at this specific client count, marking an effective scalability limit where system processing capacity and network resource utilization are most balanced.

The hallmark of this research is the quantitative revelation of ZTNA scalability limits from a network performance perspective, specifically focusing on the authorization process in resource-constrained environments. In particular, the experimental analysis of authorization bottlenecks in an Envoy/OPA configuration and the derivation of the optimal number of clients constitute the core novelty of this paper.

The remainder of this paper is organized as follows.

Section 2 summarizes the fundamental concepts of Zero Trust, the ZTNA architecture, and the basic knowledge of the technical elements used in this study (Envoy, OPA, and Mininet). Section 3 introduces related work, including [1, 2]. While existing research focuses on enhancing and automating security functions, quantitative evaluations of authorization overhead under resource constraints remain insufficient. Therefore, we construct a verification environment to elucidate these performance limits. Section 4 describes the implementation of the verification environment. Section 5 details the measurement methodology and parameters used in the experiments. Section 6 provides performance evaluation and discussion based on the experimental results. Finally, Section 7 summarizes the achievements of this study and outlines future work.

2 Preliminaries

This section describes the technical background that forms the foundation for conducting the scalability analysis of ZTNA, which is the objective of this study.

2.1 Traditional Network Security Models [3]

Traditional network security has long relied on a perimeter-based defense model that deploys firewalls and VPNs under the assumption that the internal network is inherently trusted. However, the rise of cloud computing, remote work, and digital transformation (DX) has rendered this static approach insufficient due to three critical vulnerabilities. First, perimeter connection points like VPN gateways act as single points of entry; their software vulnerabilities are frequently exploited to grant attackers full network access, and their static nature makes them easy targets for Distributed Denial of Service (DDoS) attacks. Second, the model grants implicit trust upon initial authentication, allowing an attacker who secures a single foothold to move laterally across internal servers and databases. Third, physical boundaries have dissolved as critical corporate assets and data are increasingly distributed across external SaaS and IaaS platforms. Consequently, Zero Trust Architecture (ZTA) has garnered significant attention as a necessary framework to replace this obsolete paradigm.

2.2 Overview and Implementation of Zero Trust Network Access (ZTNA) [4, 5, 6]

Zero Trust Network Access (ZTNA) is an implementation framework that applies the Zero Trust principle of "never trust, always verify" to network access control. While traditional perimeter defense models automatically trusted users and devices within the network, ZTNA strictly verifies all requests as potential threats, regardless of their origin.

The implementation of ZTNA is structured around the following foundational security principles outlined in NIST SP 800-207 [7]:

- **Principle of Least Privilege**
  Users are granted access only to the minimum resources necessary to perform their tasks. Thus, even if an account is compromised, the scope of the damage is minimized, and the attack surface is reduced.

- **Continuous Authentication and Authorization**
  Verification is not limited to a temporary check at the start of a session; user identities and device security postures are dynamically and continuously monitored throughout the

session. If changes occur in a user's location, device health, or behavioral patterns, access privileges are adjusted in real time.

- **Micro-segmentation**
  The network is divided into finely isolated zones, and individual security controls are applied to each zone. This physically and logically restricts the lateral movement of an attacker after they infiltrate the network.

2.3 Logical Components of ZTNA

ZTNA implementation separates the data plane, which is responsible for data transfer, from the control plane, which manages communication permission rules and makes authorization decisions. Verification is performed every time communication occurs. This logical architecture is primarily realized by the following three components:

- **Policy Enforcement Point (PEP)**
  Positioned between the user and the resource, it directly controls the actual communication.

- **Policy Engine (PE)**
  Acting as the brain of the system, it uses a trust algorithm to make authorization decisions for access requests.

- **Policy Administrator (PA)**
  Based on the PE's decisions, it issues commands to the PEP to either establish or tear down communication paths.

The combined functionalities of the PE and PA are collectively referred to as the Policy Decision Point (PDP). In this study, we adopt Envoy Proxy as the PEP and the Open Policy Agent (OPA), a general-purpose policy engine, as the PDP.

2.4 Authorization Process in ZTNA

Unlike traditional networks that grant access to all resources after a single authentication, communication in a ZTNA environment involves repeating the authorization and enforcement processes for every request.

First, a user's access request is intercepted by the PEP and forwarded to the PA via the control plane. The PE, receiving the request from the PA, comprehensively evaluates contextual information such as user ID, device security posture, time of request, and location to determine whether to grant access. If the PE approves the access, the decision is communicated to the PA, which then sends configuration information to the PEP to establish an encrypted communication channel. Only after this process is completed does the communication between the user and the resource begin.

Even after communication is established, the PE continues to monitor the behavior of the user and the device. If malware infection or suspicious behavior is detected during the session, the PA immediately issues a disconnect command to the PEP, dynamically terminating the communication. This resolves the problem of the traditional perimeter model, where a single authentication grants broad access privileges, thereby enabling the minimization of potential damage.

2.5 Technological Components Used in This Study

- **Mininet**
  Mininet is a network emulator that creates virtual hosts, switches, links, and controllers on a single Linux kernel, enabling the emulation of flexible network topologies. Using Linux network namespaces, each host possesses an independent execution environment, allowing programs to run on the same network stack as real hardware. In this study, we utilize Mininet as a simulation platform to observe communication characteristics under various network conditions by arbitrarily configuring parameters such as link bandwidth, latency, and the number of clients.

- **Envoy Proxy**
  Envoy Proxy is a high-performance L7 network proxy responsible for controlling inter-service communication within microservices architectures. Implemented in C++, it features low-latency and high-throughput processing capabilities, along with a diverse set of filter chains spanning from L3/L4 to L7. In this study, we utilize the external authorization filter provided by Envoy to temporarily hold all client requests and consult the OPA for authorization decisions. This configuration achieves strict access control at the proxy layer without requiring the implementation of authorization logic on the application side.

- **OPA (Open Policy Agent)**
  OPA is a general-purpose policy engine designed to realize "Policy as Code." It allows access control rules to be defined using a declarative language called Rego, enabling the management and evaluation of policies independently of the application. In this study, OPA receives attribute information sent from Envoy in JSON format, makes authorization decisions based on policies written in Rego, and returns the results back in JSON format. By centralizing policy evaluation in OPA in this manner, we verify flexible and dynamic policy operations.

3 Related Work and Research Objective

In recent years, several research trends have emerged regarding Zero Trust Architecture (ZTA) and its performance evaluation. Chen et al. proposed "ZTA-6G" [1], an architecture designed for next-generation networks that manages security through control domain units called communities. It implements dynamic, multi-stage trust evaluation by integrating the individual user equipment's reliability with community-level management status, demonstrating high robustness and low overhead in mass heterogeneous environments. Meanwhile, Katsis and Bertino introduced "ZT-SDN" [2], a framework that leverages machine learning and the centralized control of Software-Defined Networking (SDN) to automate the formulation of access control rules. By modeling communication requirements as a graph and utilizing proactive rule deployment, ZT-SDN effectively enforces the principle of least privilege with minimal human intervention while mitigating performance degradation.

However, these preceding studies assume specialized SDN controller environments or advanced collaborative defense algorithms. Consequently, the operational and performance characteristics of the general-purpose implementation combining Envoy Proxy as a Policy Enforcement Point (PEP) and Open Policy Agent (OPA) as a Policy Decision Point (PDP)—which is widely adopted in modern cloud-native infrastructures—remain insufficiently evaluated. Furthermore, existing literature typically focuses on theoretical optimization or specific dynamic

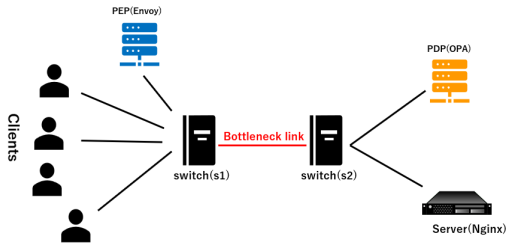

Figure 1: Network Topology

environments, leaving network scalability limits during the continuous authorization process under resource-constrained conditions unclear.

Therefore, the objective of this study is to quantitatively analyze the performance characteristics of the prevalent Envoy/OPA configuration under resource-constrained conditions. By considering the intervention of the authorization process in bandwidth-limited environments, we aim to clarify empirical scalability boundaries and establish highly practical network design guidelines.

4 Implementation of Scalability Verification in ZTNA

4.1 Research Objectives

The objective of this study is to quantitatively elucidate the impact of an increasing number of clients (request load) on overall system performance in a ZTNA configuration that enforces external authorization for all communication requests. Specifically, we verify how authorization queries become a scalability bottleneck in scenarios where network resources between the PEP and PDP are limited.

4.2 System Configuration and Software Implementation Specifications

This section provides the specific implementation details of the verification environment.

In this study, we constructed a virtual network using Mininet on an Ubuntu environment within VirtualBox. Within this network, Envoy Proxy was deployed as the PEP and OPA as the PDP to implement an authorization control environment based on Zero Trust Architecture.

To conduct fundamental scalability verification, we adopted the dumbbell topology shown in Fig. 1. A dumbbell topology consists of two switches and their connected nodes; it is well-suited for intentionally creating bottlenecks and observing their behavior. On the left side of the switches, we positioned the client group $(c_1, \ldots, c_n)$ and the Envoy proxy, while the backend server and OPA were placed on the right side. The link between the switches was configured with fixed latency and bandwidth limits to serve as the bottleneck link.

This topology succinctly represents a typical ZTNA communication model where distributed sites access a central authorization infrastructure. By aggregating links into a single central point, it is possible to accurately observe and isolate the effects of contention and latency between authorization traffic (gRPC) and actual data traffic (HTTP). The roles of each software

component and the specific implementation specifications optimized for scalability verification are described below:

- **PEP: Envoy Proxy**
  In our implementation, the `ext_authz` (External Authorization) filter is enabled in the proxy configuration file, `envoy_lb.yaml`. This configuration mandates an external authorization query to the PDP for every HTTP request sent by a client before it is forwarded to the backend server. The communication with the PDP is performed using gRPC (based on HTTP/2). The adoption of gRPC ensures efficient data exchange in binary format and connection multiplexing, thereby maintaining communication efficiency when a large volume of authorization requests occurs.

- **PDP: OPA**
  The OPA instance is configured with the parameter `--set=plugins.envoy_ext_authz_grpc.addr=:9191` to directly receive gRPC authorization requests from Envoy. In the policy file (`policy.rego`), a "Default Allow" rule is defined.

  The purpose of this setting is to minimize the computational load of policy evaluation on the PDP side. This ensures that any degradation in latency or throughput measured in this experiment is attributed to the ZTNA-specific authorization communication process itself rather than policy complexity, allowing us to extract the pure network overhead.

- **Backend Server: Nginx**
  Nginx, known for its high concurrency, is employed as the resource server for final access. Upon server startup, a dedicated directory (www1) is created, and an `index.html` file containing the string "Server 1" is dynamically generated. This response is extremely lightweight, minimizing bandwidth consumption by the backend data transfer itself. This allows the network load factors to be concentrated on the ZTNA authorization process (PEP-PDP communication).

4.3 Communication Protocols: HTTP/1.1 and HTTP/2 (gRPC)

In this experiment, different protocols are adopted to match the characteristics of the data plane (actual data communication) and the control plane (authorization communication). For content requests from clients to the backend server, HTTP/1.1 is used, as it remains a widely prevalent protocol in modern web communication. To simulate standard web requests, access to the Nginx server is performed via HTTP/1.1.Conversely, gRPC (HTTP/2) is used for authorization requests from Envoy to OPA. gRPC is a high-speed, lightweight protocol that utilizes the binary framing layer of HTTP/2. Through header compression and binary serialization (Protocol Buffers), contextual information required for authorization can be transferred with minimal packet size. Furthermore, because multiple authorization requests can be processed in parallel over the same connection, it enables efficient verification while suppressing resource consumption on the PDP side.

The communication process from the client's request to the response is designed as a serial flow crossing the data and control planes, as shown in Fig. 2. Specifically, the Envoy proxy on the path first captures the client's HTTP/1.1 request and temporarily suspends the communication. Next, Envoy generates an authorization request and sends it to the OPA using gRPC. Upon receiving the request, the OPA verifies it based on the policy and returns the decision (Allow/Deny) to Envoy. Finally, if the decision is "Allow," Envoy forwards the suspended request to the backend server. If "Deny," it returns an error and blocks the communication.

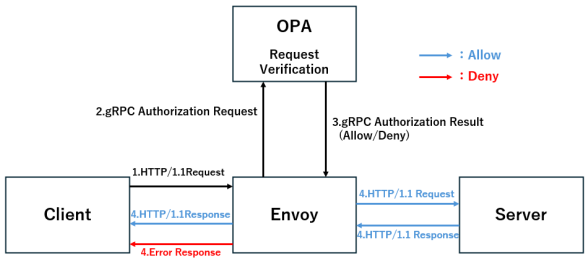

Figure 2: Overview of Communication Flow

5 Experiment

This section describes the experimental setup used to instantiate the verification model proposed in Section 4.

5.1 Experimental Overview

The experiments were conducted in a Mininet emulation environment on a single host machine. Due to physical CPU and memory resource constraints, it was not feasible to operate thousands of concurrent clients. Therefore, instead of increasing the number of clients to a massive scale, we limited the maximum number of clients to 55 and restricted the network bandwidth of the bottleneck link between switches $s1$ and $s2$ to a relatively low 1 Mbps. Through this methodology, we intentionally induced a resource saturation state where authorization traffic (gRPC) and actual data traffic (HTTP) compete for limited bandwidth even with a small number of hosts, thereby highlighting the scalability challenges of ZTNA. Additionally, to observe the impact of gRPC on this communication, we conducted a similar experiment where the parallel processing effects of gRPC in authorization communication were disabled.

5.2 Network Configuration and Parameters

This section details the specific numerical parameters used in the experiment and the metrics for multifaceted system performance evaluation. The configuration parameters are as follows:

- **Number of Clients**
  The number of clients was increased from 1 to 55 in increments of five to verify the load conditions.
- **Bandwidth Limitation**
  As mentioned above, the bandwidth was restricted to 1 Mbps to create a relatively high-load environment.
- **Inter-switch Latency (One-way)**
  A fixed latency of 100 ms was applied to simulate the distance between the edge and the cloud.

To evaluate scalability, the following five metrics were calculated from the collected log data:

1. **Success Rate**
   The ratio of successful responses to the total number of requests.

$$\text{SuccessRate} = \left(\frac{\text{SuccessCount}}{\text{TotalRequests}} \right) \times 100, [\%]$$

   Here, SuccessCount is the number of successfully completed requests, and TotalRequests is the total number of requests sent by the clients. A higher Success Rate indicates higher processing capacity of the PDP and greater network stability in the ZTNA configuration.

2. **Goodput**
   To evaluate the effective utilization of bandwidth, throughput is calculated limited to successful communications only.

$$\text{Goodput} = \text{Throughput} \times \frac{\text{SuccessRate}}{100}$$

3. **Average Round-Trip Time (Average RTT)**
   The average time from when a client sends a request until it receives a response.

4. **Authorization Delay (Auth Delay)**
   Identifies the pure latency added to the communication by the authorization process.

$$\text{AuthDelay} = \text{AverageRTT} - \text{BaselineLatency}$$

   Here, BaselineLatency refers to the measured round-trip delay without authorization—specifically, the time measured when the client accesses the server directly without passing through Envoy.

5. **Worst-case Latency (Worst)**
   The maximum response time during the measurement period. By identifying sudden latency spikes in specific trials that might be masked by average trends, we more rigorously define the limits of system reliability.

5.3 Workload Configuration

This section describes the traffic generation model and specific parameters for the clients.

- **Load Generation Tool and Request Specifications**
  The high-load testing tool `wrk` was used to generate HTTP requests from each client. Each client maintained 5 concurrent connections with 1 thread, continuously sending requests for 30 seconds. The timeout was set to 5 seconds.

- **Measurement Sequence**
  For each client count, requests were generated in the following order:

  1. **Baseline Measurement**: Performance was measured by accessing the backend server directly, bypassing Envoy, in a state without authorization overhead.
  2. **ZTNA Environment Measurement**: Performance was measured with external authorization via OPA enforced through Envoy. This measurement was repeated 10 times, and the average values were adopted.

5.4 Experimental Results

Table 1 summarizes the performance evaluation results for each client count under a 100 ms latency environment. Table 2 shows the results of the evaluation implemented without the parallel processing effects of gRPC.

Table 1: Summary of experimental results (1 Mbps bandwidth limit, 100 ms latency, average of 10 trials)

Clients	Success%	Goodput(req/s)	RTT(ms)	AuthDelay(ms)	Worst(ms)
1	100.00	11.85	418.60	212.07	683.86
5	100.00	59.52	417.76	210.87	780.80
10	100.00	105.39	472.59	98.77	834.28
15	100.00	111.96	663.06	117.47	895.88
20	100.00	114.35	864.29	150.67	1120.87
25	100.00	115.95	1064.68	182.84	1467.24
30	100.00	117.50	1255.37	212.47	1560.68
35	100.00	118.73	1449.51	232.37	1967.09
40	100.00	119.23	1643.67	275.67	2311.48
45	100.00	121.90	1808.62	280.84	2310.72
50	100.00	123.67	1979.60	312.00	2483.27
55	100.00	125.28	2151.13	339.31	2795.02

Table 2: Experimental results without gRPC parallel processing (1 Mbps bandwidth limit, 100 ms latency, average of 10 trials)

Clients	Success%	Goodput(req/s)	RTT(ms)	AuthDelay(ms)	Worst(ms)
1	100.00	9.54	528.14	319.35	888.73
5	100.00	18.85	1323.05	1116.14	1898.62
10	100.00	18.98	2569.70	2362.90	3828.74
15	99.95	19.43	3647.07	3441.21	4808.57
25	66.24	16.80	4624.64	4419.53	5242.88
40	31.75	12.46	4776.35	4571.30	5262.57
55	16.69	8.92	4836.36	4624.69	5229.28

Furthermore, we used the packet capture tools `tcpdump` and Wireshark to analyze the communication behavior in this experiment. Specifically, we captured packet data on the bottleneck switch ($s1$) and visualized the size of packets sent from Envoy to OPA using Wireshark's TCP Time-Sequence Graph (Stevens). Fig. 3 shows the analysis results for client counts of 1 and 15. The vertical axis of each graph represents the cumulative packet size, while the horizontal axis represents time. This allowed us to confirm the transition of packet transmission over time and the presence or absence of pipeline processing via multiplexing.

6 Discussion

In this section, we analyze the experimental data to derive insights into the performance characteristics of the proposed model. We then generalize the optimal number of clients for the implemented dumbbell topology.

(a) 1 Client (b) 15 Clients

Figure 3: Packet capture results comparing 1 and 15 clients

6.1 Analysis of Experimental Results

The results demonstrate that the performance bottleneck in ZTNA is not merely a consequence of bandwidth constraints but stems from the contention between authorization traffic (gRPC) and data traffic. Notably, the presence or absence of gRPC's parallel processing capabilities significantly impacts performance.

While the *Goodput* increases with the number of clients, the rate of increase plateaus after the count exceeds ten. Regarding the *Auth Delay*, it remains around 200 ms under low-load conditions with few clients. As observed in the step-like behavior in Fig. 3(a), the communication follows a sequence of request transmission, RTT waiting time, and response reception. This indicates that the physical latency of the bottleneck link is directly reflected in the delay. Interestingly, the *Auth Delay* reached its minimum value at ten clients. We attribute this to the multiplexing effect of gRPC (HTTP/2), where subsequent requests are transmitted during the waiting periods of previous ones. This behavior is evident in the linear slopes observed in Figs. 3(b).

Furthermore, Table 2 shows that no such dip in *Auth Delay* occurs when gRPC parallel processing is disabled. This confirms that the minimum delay observed in Table 1 is indeed a result of gRPC's multiplexing capabilities.

6.2 Generalization of the Optimal Number of Clients

We define the "optimal number of clients" as the point where the *Auth Delay* is minimized and the *Goodput* begins to plateau. Similar to the 1 Mbps implementation, we collected data for the optimal number of clients at bandwidth limits of 1, 2, 3, 4, and 5 Mbps. Based on the 1 Mbps data, we predicted the optimal client count to be around $10N$ and performed measurements by incrementing the count by one. The optimal point was identified as the client count that recorded the minimum *Auth Delay*. Table 3 summarizes these results.

Table 3: Relationship between Bandwidth and Optimal Number of Clients

Bandwidth[Mbps]	1	2	3	4	5
Number of Clients	11	21	30	41	49

By applying the least squares method, we derived that for a bandwidth of N Mbps and a one-way latency of 100 ms in a dumbbell topology, the optimal number of clients C is given by: $C = 9.6N + 1.6$ This equation serves as an empirical scalability index for our experimental environment and can be utilized as a design guideline for ZTNA systems under bandwidth constraints. However, as this result depends on the specific topology and latency conditions, further verification is required for broader generalization.

7 Conclusion

In this study, we constructed a verification platform using Mininet, Envoy, and OPA to analyze the scalability of Zero Trust Network Access (ZTNA). Specifically, we elucidated the impact of the external authorization process on network performance within resource-constrained environments.

The primary achievement of this research lies in the quantification of ZTNA behavior under bandwidth limitations. Furthermore, we generalized the optimal number of clients based on the collected experimental data for the given environment.

Our performance evaluation revealed that when the bandwidth is N Mbps, the growth rate of *Goodput* begins to decrease once the number of clients exceeds $(9.6N + 1.6)$. Simultaneously, at this point, the *Auth Delay* reaches its minimum value due to the parallel processing facilitated by Envoy's gRPC multiplexing effect. These results indicate that $(9.6N + 1.6)$ clients represent the performance limit for the network configuration implemented in this study. However, this study has several limitations. The experiments were conducted in a small-scale Mininet environment, which may not fully replicate the behavior of large-scale, distributed environments in real-world networks. Additionally, the policy evaluation was simplified, and thus the impact of complex authorization logic encountered in practical operations was not considered.

Future work will involve relaxing the constraints used in this study to better reflect real-world conditions and conducting performance evaluations in larger-scale environments. Furthermore, based on these multifaceted evaluation results, we aim to analyze effective network configurations and optimal parameter settings to ensure stable ZTNA operations, ultimately establishing highly practical network design guidelines.

Acknowledgement

This work is partially supported by JSPS KAKENHI Grant Number JP21H03443 and SECOM Science and Technology Foundation.

References

[1] Xu Chen, Wei Feng, Ning Ge, and Yan Zhang. Zero trust architecture for 6g security. *IEEE network*, 38(4):224–232, 2023.

[2] Charalampos Katsis and Elisa Bertino. Zt-sdn: an ml-powered zero-trust architecture for software-defined networks. *ACM Transactions on Privacy and Security*, 28(2):1–35, 2025.

[3] Gerald A Marin. Network security basics. *IEEE security & privacy*, 3(6):68–72, 2005.

[4] Vasilios Mavroudis. Zero-trust network access (ztna). *arXiv preprint arXiv:2410.20611*, 2024.

[5] Kipkoech Denzel and Sydney Ng'etich. A survey of security in zero trust network architectures. *GSC Advanced Research and Reviews*, 22, 01 2025.

[6] Naeem Firdous Syed, Syed W Shah, Arash Shaghaghi, Adnan Anwar, Zubair Baig, and Robin Doss. Zero trust architecture (zta): A comprehensive survey. *IEEE access*, 10:57143–57179, 2022.

[7] Scott Rose, Oliver Borchert, Stuart Mitchell, and Sean Connelly. Zero trust architecture, 2020-08-10 04:08:00 2020.