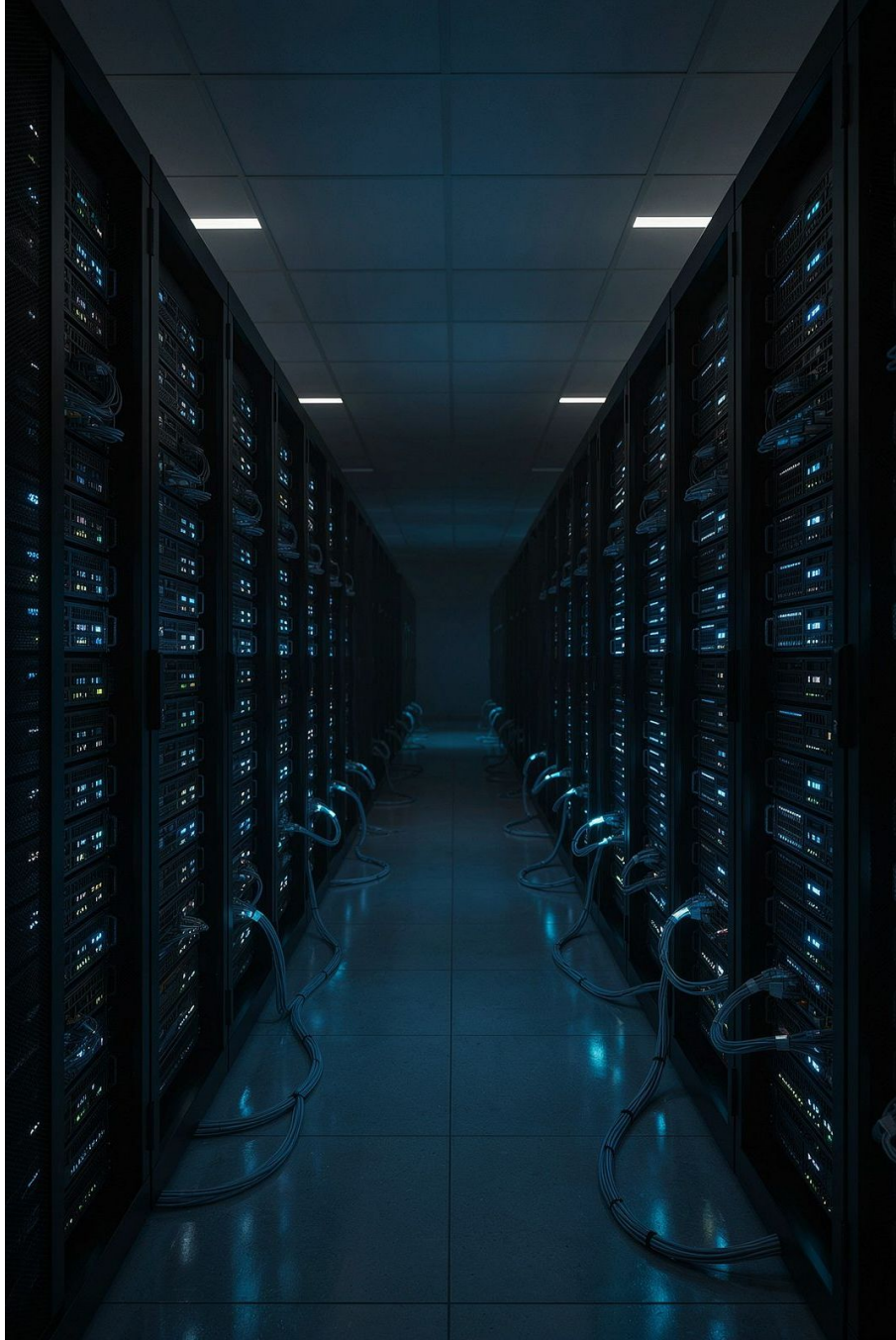
Zscaler ZIA & ZPA — End-to-End Slowness & Performance Troubleshooting Masterclass

From User Endpoint → ISP → Zscaler → Internet / Private Application

Techclick Infosec Pvt Ltd | ai.techclick.in | Training Contact: WhatsApp +91 92772 29456



Learning Objectives

By the end of this masterclass, you will be able to systematically isolate performance problems across every segment of the ZIA and ZPA traffic path — and produce evidence-backed root-cause analysis.

Traffic Flows

Understand ZIA and ZPA architecture and where each type of traffic travels

Latency Isolation

Identify the exact segment introducing delay — endpoint, ISP, tunnel, cloud, or backend

Service Edge Validation

Determine the actual connected ZIA/ZPA Service Edge and validate GeoIP/service selection

Transport & Evidence

Troubleshoot DTLS/TLS, read MTR/traceroute/tcpdump, and build a TAC evidence package

01

Compare Zscaler ON vs OFF correctly using controlled test methodology

02

Troubleshoot App Connector-to-backend latency in ZPA scenarios

03

Collect reproducible evidence before opening Zscaler TAC tickets

04

Produce a formal RCA distinguishing Observation → Hypothesis → Evidence → Root Cause

The Golden Troubleshooting Principle

Never start with "Zscaler is slow."

Convert every complaint into a precise, answerable question: **Exactly which segment is slow?**

Performance troubleshooting is fundamentally a **path-isolation exercise**. Each hop must be measured and assigned an owner before any conclusion is drawn.

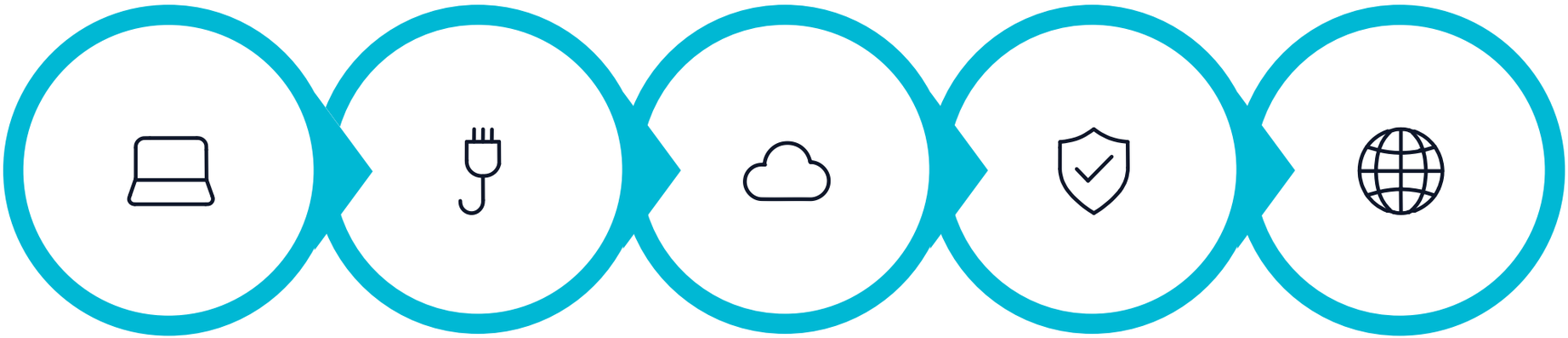
The Full Path — ZIA

1. **User Endpoint**
2. **Local Wi-Fi / LAN**
3. **ISP / Transit**
4. **ZIA Service Edge + Processing**
5. **Destination / SaaS**

Additional Path — ZPA Only

1. **ZPA Service Edge**
2. **App Connector**
3. **Firewall / Network**
4. **Backend Application**

ZIA Architecture — Internet & SaaS Traffic Flow



User Laptop | **Z-Tunnel 2.0** | **ZIA Service Edge** | **Security Engine** | **Internet / SaaS**

What Happens at ZIA Service Edge

URL Filtering	**Cloud Firewall**
SSL Inspection	**Threat Protection**
DLP / CASB	

Key Architecture Facts

- ZIA handles all **internet-bound and SaaS traffic** — not private app access
- Zscaler Client Connector (ZCC) captures eligible traffic and forwards it via **Z-Tunnel 2.0**
- The tunnel transport layer uses **DTLS by default**, with TLS fallback available
- ZIA applies security policy **inline** before forwarding to the destination

zscaler™
Experience a Safer, More Open World™

Zscaler ZPA – Private Application Access Flow

Secure, Zero Trust Access to Private Applications – No VPN. No Lateral Movement. Just the Apps You Authorize.

ZERO TRUST
GREATER SECURITY
A STRONGER TOMORROW

Policy Authorization
User identity, device posture and policy are evaluated to determine if the application can be accessed.

App Connector Selection
Service Edge selects the best App Connector based on policy, health, location and capacity.

Application Mapping
Private URL is mapped to a Synthetic IP and routed to the correct backend application via App Connector.

Backend TCP Connection
App Connector establishes TCP connection to the backend server (e.g. 192.168.1.101:443).

1 Remote User
Access from anywhere

User Laptop
with Zscaler Client Connector

Home Office Travel
Any Location, Any Network

2 Zscaler Client Connector
Identity-aware access

Zscaler Client Connector (ZCC)
Enforces policy and establishes secure session

Request

3 ZPA Public Service Edge
Global Zero Trust exchange

ZPA Public Service Edge
Policy enforcement, application discovery and secure routing

4 Microtunnel
Encrypted, outbound only

Microtunnel (TLS Encrypted)
Outbound connection from Service Edge to App Connector (no inbound rules required)

5 App Connector
Deployed in your network

App Connector
192.168.1.100
Receives traffic from Service Edge and connects to the private application

6 Internal Firewall / Network
Your private network

Internal Firewall / Network
Keeps applications private (no inbound access required)

7 Private Application
Your protected application

ERP Application
URL: erp.internal.example
Synthetic IP: 100.64.10.50
Backend: 192.168.1.101:443

Return Traffic (Response)
Application response travels back through the same secure path (microtunnel)

Latency Checkpoint: User → Service Edge
(Measure end-user to ZPA latency)

Latency Checkpoint: Service Edge → App Connector
(Measure microtunnel latency)

Latency Checkpoint: App Connector → Backend
(Measure internal network latency)

Latency Checkpoint: Backend Processing
(Measure application response time)

No traditional network-level VPN access is provided

Zscaler ZPA gives the user access only to the authorized private application, not broad LAN access.
Users cannot see, scan or access other resources in your internal network.

Least Privilege Access

- Granular application-level access
- No network exposure
- No lateral movement
- Stronger security posture

PRIVATE APPS PROTECTED
PEOPLE EMPOWERED

ZPA Architecture — Private Application Access

Example Scenario

User Location	India
Requested URL	finance.internal.example
Synthetic IP	100.64.10.25
App Connector IP	192.168.1.100
Real Backend IP	192.168.1.101:443

ZPA Is Not a Traditional VPN

ZPA provides **application-level private access** — the user is never placed directly onto the private network. The ZPA Service Edge brokers connections through App Connectors which establish **outbound connectivity** to the Zscaler cloud. No inbound firewall rules for users are required.

ZPA uses Microtunnels between the Service Edge and App Connector. The user interacts with a Synthetic IP, not the real backend address.

ZIA vs. ZPA — Traffic Flow Comparison

Attribute	ZIA — Internet & SaaS	ZPA — Private Access
Traffic Type	Internet-bound, SaaS, Cloud apps	Internal / private application access
Example URL	www.microsoft.com, www.google.com	erp.internal.example, finance.corp
Flow	User → ZCC → ZIA Service Edge → Internet	User → ZCC → ZPA Service Edge → App Connector → Backend
Service Edge Type	ZIA Public Service Edge	ZPA Public or Private Service Edge
App Connector Needed?	No	Yes — provides application-side connection
Synthetic IP Used?	No	Yes — user sees Synthetic IP, not real backend
Security Inspection	URL Filter, SSL Inspect, DLP, CASB	Policy-based access control, segmentation
Primary Troubleshooting Concern	Tunnel health, ISP path, destination latency	App Connector health, backend reachability

ⓘ Rule: If the destination is on the internet or a SaaS platform → ZIA. If it is a private/internal application → ZPA.

zscaler™
Secure. Simple. Transform.

Zscaler ZIA – End-to-End Internet Traffic Flow

Secure Users | Secure Internet Access | A Safer Tomorrow

ANY USER
ANY DEVICE
ANY LOCATION
ANY APPLICATION

1. User Laptop

Employee / Remote User
Windows | macOS | Linux

1

2. Zscaler Client Connector

Zscaler Client Connector
Establishes secure tunnel to Zscaler

2

3. Z-Tunnel 2.0 (DTLS/TLS)

Encrypted Tunnel (DTLS/TLS)
Secure, resilient connection (Z-Tunnel 2.0)

3

4. ZIA Public Service Edge

Global Service Edge
Closest ZIA data center (200+ locations)

4

5. Security Inspection Stack

- SSL Inspection
- URL Filtering
- Firewall
- Threat Protection

Inspect, Enforce, Protect
Real-time security inspection at cloud scale

5

6. Internet / SaaS Application

Microsoft 365 | Google Workspace | salesforce | workday | Other Internet Apps

Public Internet / SaaS

User Wi-Fi
Possible latency zone
- Wi-Fi issues (connectivity, signal, congestion)

ISP
Possible latency zone
- ISP issues (routing, packet loss, congestion)

Internet Transit
Possible latency zone
- Transit issues (routing, congestion, packet loss)

Zscaler Service Edge
Possible latency zone
- Service edge issues (capacity, maintenance, regional)

Destination Application
Possible latency zone
- Application issues (slow response, outage, throttling)

Identify exactly where latency starts before assigning root cause.
Measure and validate each segment (end-to-end) to quickly isolate the issue.

Step-by-Step ZIA Request: https://www.example.com

Request Flow with Latency Markers

1 — ① Laptop generates HTTPS request

Browser calls DNS resolver → resolves www.example.com

2 — ② ZCC intercepts eligible traffic

ZCC captures the session based on forwarding policy — ⚠️ Latency: ZCC overhead if CPU high

3 — ③ Traffic enters Z-Tunnel 2.0

DTLS or TLS encapsulation — ⚠️ Latency: DTLS instability, MTU mismatch

4 — ④ ZIA Service Edge reached

RTT measured here — ⚠️ Latency: ISP path, wrong Service Edge

5 — ⑤ Policy applied + SSL Inspection

URL Filter, DLP, Threat Scan — ⚠️ Latency: large transactions, cert issues

6 — ⑥ ZIA forwards to destination

ZIA → CDN/server — ⚠️ Latency: internet path, CDN, server delay

7 — ⑦ Response returned to user via ZIA

Complete — browser renders page

Key Troubleshooting Insight

Every step introduces a potential latency domain. **Do not assume the tunnel is the problem** without measuring each segment independently.

DNS Delay

Step 1 — resolver, split DNS, TTL

Tunnel Overhead

Step 3 — DTLS/TLS, MTU, fragmentation

Service Edge RTT

Step 4 — ISP routing, GeoIP selection

Inspection Delay

Step 5 — SSL handshake, policy complexity

Destination Delay

Step 6 — server load, CDN, SaaS performance

Step-by-Step ZPA Request:
https://erp.internal.example

Why This Is NOT a Traditional VPN

In a VPN, the user's device is placed on the private network and can route to any reachable host. In ZPA, the user interacts only with a **Synthetic IP** that maps to the protected application. The real backend IP is never exposed to the client. App Connectors establish outbound connections — no inbound user traffic reaches the private network directly.

Latency Markers in ZPA Flow

Policy evaluation delay

ZPA checks authorization before establishing the microtunnel

App Connector selection

Connector must be healthy, reachable, and application-aware

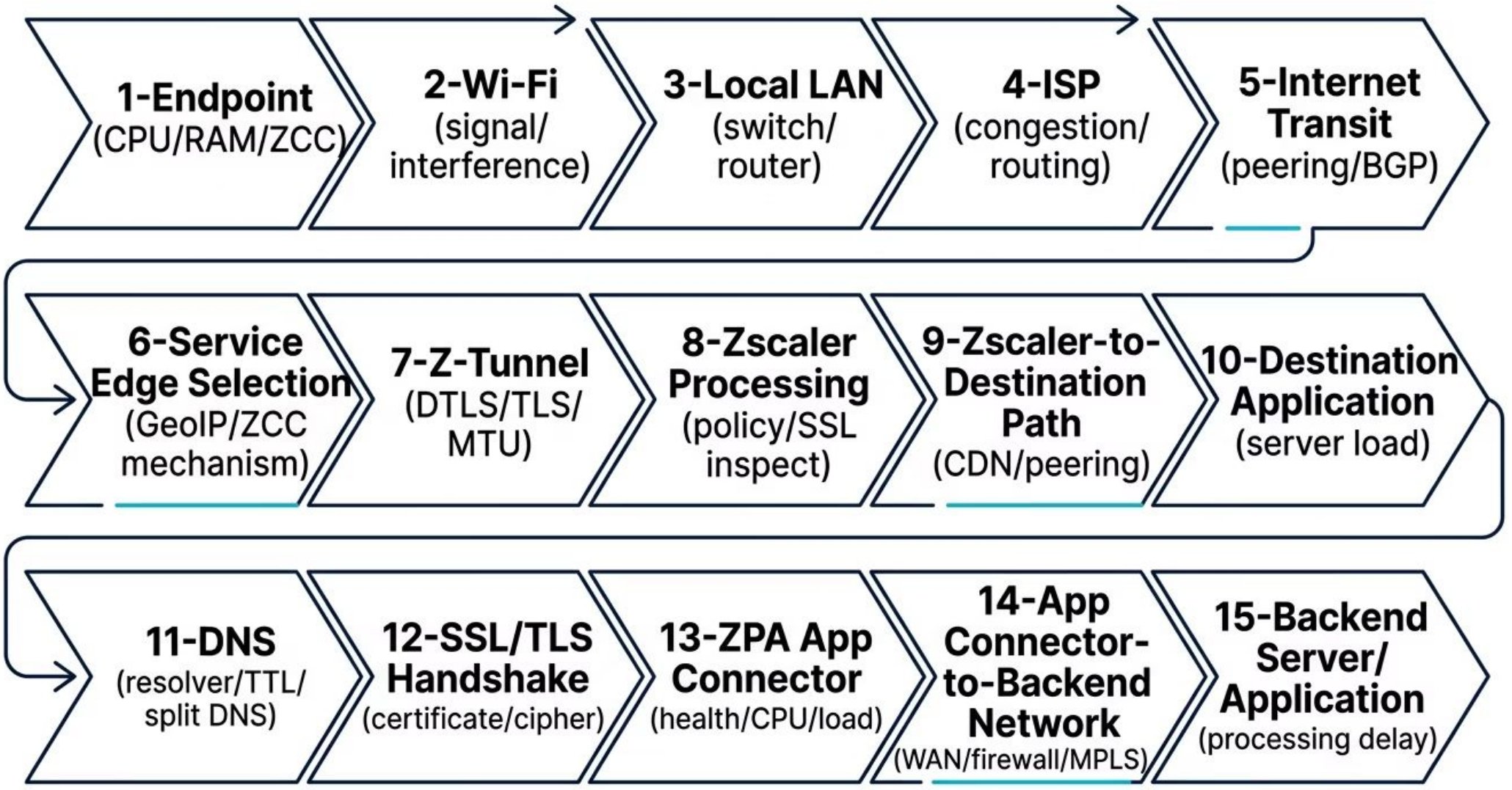
Connector → Backend RTT

Enterprise WAN/MPLS or firewall rules may add delay

Backend processing time

Server response delay is often misattributed to ZPA transport

Where Can Slowness Occur? — End-to-End Fault Domain Map



Ownership Reference

Zones 1–3	User / IT Helpdesk / Network Team
Zones 4–5	ISP / Carrier / Transit Provider
Zones 6–9	Zscaler / TAC — requires ZCC evidence
Zones 10–12	Destination / SaaS / CDN provider
Zones 13–15	ZPA App Connector team / Backend App team

⚠ Never assign ownership to a zone without measuring it. Observation → Hypothesis → Evidence → Root Cause is the required sequence.

CASE 1 — Real Enterprise Scenario: Jio User Slowness

Scenario Setup

User Physical Location	India
ISP	Jio (Reliance Jio Infocomm)
Example Public IP	203.0.113.25 (documentation range)
GeoIP Result (incorrect)	United States / Israel
User Complaint	"Internet becomes slow whenever Zscaler is enabled"

Possible Symptom

User may appear connected to an unexpectedly distant ZIA or ZPA Service Edge, resulting in elevated RTT and slower application transactions.

Troubleshooting Framework Applied

OBSERVATION

User reports slowness on Jio with ZCC enabled. Fast without ZCC.

HYPOTHESIS

Incorrect GeoIP mapping may be causing connection to a distant Service Edge — **requires validation, not assumed.**

EVIDENCE REQUIRED

Actual Service Edge name, RTT measurement, comparison with another ISP/hotspot, ZCC diagnostics

ROOT CAUSE

Only declared after evidence confirms the mechanism. Do not pre-assign blame.

⊗ GeoIP mismatch is a **hypothesis**. Validate the actual Zscaler Service Edge before treating it as the root cause.

Why Incorrect GeoIP Can Impact Performance

Correct Scenario — Nearby Service Edge

India User	RTT
Connects to nearby Service Edge	Illustrative example: ~20–50 ms

Result

Normal transaction times, acceptable performance

Potentially Problematic Scenario — Distant Edge

India User	RTT
Incorrect GeoIP → distant foreign Service Edge	Illustrative example: ~150–300+ ms

Result

Every transaction carries additional round-trip overhead

How Zscaler Client Connector Selects a Service Edge

ZCC uses **geolocation and service-selection mechanisms** — not a single GeoIP lookup — to connect users to an appropriate nearby ZIA or ZPA Service Edge. An incorrect GeoIP result is one **possible contributing factor** among several. Before concluding GeoIP is the cause, validate:

- The actual connected Service Edge hostname/location
- The measured RTT from the user to that Service Edge
- Whether other users on the same ISP prefix are affected
- Whether a different ISP or mobile hotspot connects to a closer edge

⚠ Latency figures above are **illustrative examples only** — not Zscaler SLA values or guarantees.

Do Not Assume GeoIP Is the Root Cause

Critical Warning: Incorrect MaxMind GeoIP result ≠ automatically incorrect Zscaler Service Edge selection. These are separate systems. Validate both independently.

Before GeoIP can be declared a contributing factor, ALL of the following must be validated:

Public IP & ISP

- Confirm actual public IP in use
- Confirm ISP prefix/ASN ownership
- Verify physical user region matches expectation

GeoIP Sources

- Check multiple reputable GeoIP sources
- Check MaxMind specifically (as Zscaler references it)
- Note discrepancies between databases

Actual Service Edge

- Identify actual connected ZIA Service Edge
- Identify actual connected ZPA Service Edge
- Check ZCC tunnel state and diagnostics

RTT & Scope

- Measure RTT toward Service Edge
- Test with alternate ISP / mobile hotspot
- Confirm whether all users on same ISP range are affected

PEOPLE
NETWORKS
SECURITY
A BRIGHTER TOMORROW

Zscaler Slowness – GeoIP / Unexpected Service Edge Scenario

Same User. Different GeoIP Result. Different Zscaler Service Edge. Big Difference in Performance.

SECURE
FASTER
ANYWHERE
TOGETHER

A
SAFER
CONNECTED
WORLD

1 User and Public IP (Real Location)

User in India

Jio Internet

Public IP: 203.0.113.25

2 GeoIP Lookup (May be incorrect)

GeoIP Database (e.g. MaxMind)

GeoIP result: **United States** ✗

GeoIP result: **Israel** ✗

INCORRECT / MISLEADING RESULT

GeoIP mismatch can be a clue, not proof.

3 Two Possible Traffic Paths

A Expected routing (Optimal)

India User (203.0.113.25) → Traffic routed to nearby Service Edge → Zscaler Service Edge (India) Mumbai / New Delhi → Destination (e.g. Internet / SaaS)

Low latency: 25 – 40 ms

Shorter path.
Better performance.

B Suspected issue path (Sub-optimal)

India User (203.0.113.25) → Traffic routed to distant foreign Service Edge → Zscaler Service Edge (US or Israel) e.g. Los Angeles (US) / Tel Aviv (Israel) → Destination (e.g. Internet / SaaS)

High latency: 180 – 250 ms

Longer path.
Slower performance.

Wrong GeoIP result does NOT automatically prove wrong Zscaler Service Edge selection.

Multiple factors (ISP routing, BGP, anycast, load, policy, Zscaler decisions) can influence the selected Service Edge.

4 Troubleshooting / Validation Workflow

Validate → Measure → Isolate → Collect → Escalate

1. **Public IP** — Confirm the user's current public IP (e.g. 203.0.113.25).
2. **GeoIP Check** — Check GeoIP result (e.g. MaxMind). Look for mismatches.
3. **Identify Actual Zscaler Service Edge** — Find which Zscaler Service Edge the user is actually connected to (via ZCC or logs).
4. **Measure RTT** — Measure latency (RTT) to the Service Edge and/or destination. Compare results.
5. **Test Alternate ISP** — Use alternate ISP testing to isolate provider issues (e.g. mobile hotspot, other broadband).
6. **Collect ZCC Diagnostics** — Collect ZCC diagnostics for evidence (Logs, configuration, Service Edge info).
7. **Raise MaxMind/Zscaler ticket if confirmed** — If issue is confirmed, raise a ticket with MaxMind and/or Zscaler with collected evidence.

- GeoIP mismatch can be a clue, not proof.
- Validate actual Service Edge before concluding.
- Compare latency and routing behavior.
- Use alternate ISP testing to isolate provider issues.
- Collect ZCC diagnostics for evidence.

GeoIP Validation Workflow

MaxMind Check
Validate MaxMind GeoIP2 result

Cross-Check GeoIP
Query multiple GeoIP sources

Verify Ownership
Confirm ISP and ASN ownership

Identify IP
Get public IP via ipinfo.io/ip

Key Tools for GeoIP Validation

Tool	Purpose
ipinfo.io / ip-api.com	Cross-check GeoIP result
MaxMind GeoIP2 Demo	Check MaxMind-specific database result
ZCC Diagnostics	Read actual Service Edge connected
MTR / traceroute	Measure RTT and path hops to Service Edge
Mobile hotspot	Compare same device on different ISP

GeoIP databases can and do disagree with each other. Disagreement between databases is itself evidence. The authoritative signal is the **actual connected Service Edge** observed in ZCC diagnostics — not what any GeoIP lookup reports.

MaxMind GeoIP Correction Process

Step-by-Step Correction Workflow

01

Record the affected IP prefix or ISP range — not just a single IP

02

Record the actual expected country/region with authoritative evidence

03

Capture the current incorrect MaxMind result (screenshot + date)

04

Submit official correction via MaxMind's GeoIP correction request portal

05

ISP/network owner provides authoritative geofeed data (RFC 8805 preferred)

06

Track correction ticket — MaxMind updates on a regular database release cycle

07

Retest after propagation — open Zscaler TAC case if behavior persists

Operator-Scale Considerations

For large ISPs or carriers where many IP prefixes are affected, submitting individual IP corrections is inefficient. The preferred authoritative mechanism is a published **geofeed** referencing RFC 8805-compliant geofeed data, which major GeoIP providers including MaxMind can consume.

Portal path may vary. Always reference **dev.maxmind.com** for the current correction submission process. Zscaler TAC can assist in correlating Service Edge selection with corrected GeoIP data after database updates propagate.

Building the Zscaler TAC Evidence Package

Required Evidence Checklist

Identity & Environment

- Username / UPN
- Timestamp with timezone
- Physical user location
- ISP name and public IP
- ZCC version

Tunnel & Service Edge

- Z-Tunnel version
- DTLS or TLS state
- Connected ZIA Service Edge
- Connected ZPA Service Edge (if applicable)

Reproduction & Testing

- Destination URL / application
- Reproduction steps
- Zscaler Cloud Performance Test results
- Traceroute / MTR / path evidence

Additional Evidence

- ZCC logs
- ZDX evidence (if licensed)
- GeoIP screenshots
- ON vs OFF comparison
- Number of impacted users
- Alternate ISP / hotspot result

Sample TAC Ticket Title

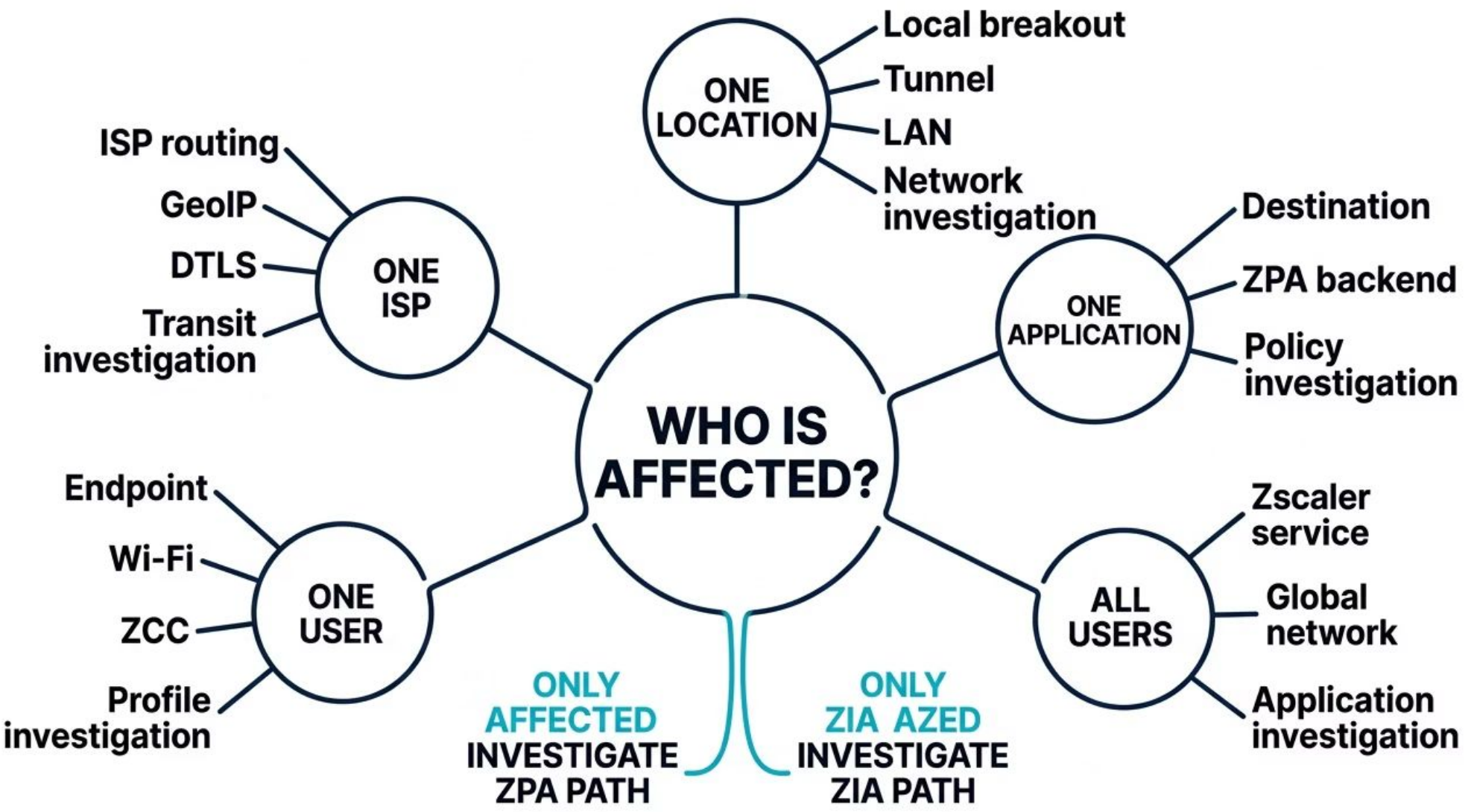
> "Possible incorrect Service Edge selection / GeoIP mapping — Jio users in India experiencing high latency with ZCC enabled. RTT to connected Service Edge = 210 ms. Nearby Service Edge expected. Alternate ISP RTT = 35 ms."

Sample Ticket Body Structure

1. Environment summary (user, ISP, ZCC version)
2. Symptom description with timestamps
3. Test methodology (Zscaler ON vs OFF, controlled)
4. Attached: MTR output, ZCC logs, GeoIP screenshots, Cloud Performance Test results
5. Hypothesis with supporting evidence
6. Current workaround status (if any)

Performance Baseline — First Question: Scope

Before any packet capture or command is run, answer: **How many users are affected, and what is the pattern?** The scope answer immediately narrows the fault domain.



> Never skip the scope question. A single-user problem investigated as a Zscaler cloud incident wastes hours. A multi-user incident investigated as an endpoint issue misses the real owner.

Correct Performance Testing Strategy

Preferred Evidence Sources (In Priority Order)

1 Zscaler Cloud Performance Test

Zscaler's own tool — measures performance through the ZIA path. Use this first.

2 ZDX (if licensed)

Continuous digital experience monitoring — provides per-user, per-app, per-path telemetry

3 Browser Developer Tools

Network tab shows DNS time, TCP connect, TLS handshake, TTFB, and total load

4 Wireshark / tcpdump

Packet-level evidence for RTT, retransmissions, and server response delay

5 Application Transaction Timing

Time a specific business operation — login, file download, form submit

Speedtest.net — Use with Caution

Generic speed tests measure **raw throughput to a single CDN endpoint**. They do not test the ZIA/ZPA path, destination behavior, DNS, TLS handshake, or application-layer latency. Use only as a comparative data point — not sole evidence.

> ⚠ A fast Speedtest.net result with Zscaler ON does **not** rule out application-layer slowness. Test the actual affected workflow.

What to Always Keep Constant

- Same laptop, same Wi-Fi, same ISP
- Same destination and test server
- Similar time window (avoid peak vs off-peak comparisons)

Controlled Zscaler ON vs. OFF Test Methodology

Metric	TEST A — ZCC/ZIA Enabled	TEST B — Authorized Direct Bypass
Download throughput	Measure and record	Measure and record
Upload throughput	Measure and record	Measure and record
RTT (ping baseline)	Measure to same target	Measure to same target
DNS resolution time	Browser DevTools → Timing	Browser DevTools → Timing
TCP connect time	Browser DevTools	Browser DevTools
TLS handshake time	Browser DevTools	Browser DevTools
TTFB (Time to First Byte)	Browser DevTools	Browser DevTools
Total page load	Browser DevTools	Browser DevTools

⊗ **Important:** A faster direct (bypass) test does **not** automatically prove Zscaler is malfunctioning — the two paths traverse different network routes and may reach different CDN nodes. Any bypass test must be **authorized by organizational policy and change control**. Do not permanently disable security controls for performance optimization.

Identify the Actual Zscaler Service Edge — Before Everything Else

How to Determine the Connected Service Edge

ZCC Diagnostics Panel

ZCC tray icon → Advanced → Diagnostics. Shows ZIA and ZPA Service Edge hostname, tunnel state, and ZCC version. (Portal path/menu may vary by ZCC version.)

ZPA User Status (Admin Portal)

ZPA Admin Portal → Diagnostics → User Status. Shows client IP, Service Edge, location, Connection ID, version. (Portal path/menu may vary by Zscaler UI version.)

ZDX Device View

If licensed, ZDX provides per-device Service Edge visibility alongside application experience metrics

ZCC Logs

ZCC logs contain Service Edge negotiation, tunnel establishment, DTLS/TLS state, and connection events

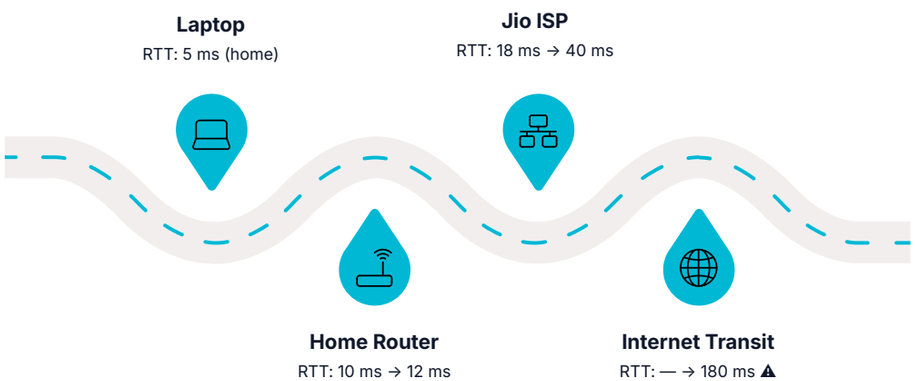
Example ZPA User Status Fields

Client Public IP	203.0.113.25
Client Private IP	10.10.10.25
ZPA Service Edge	zpa-se-ind-01.zscaler.net (example)
User Location	India
Connection ID	abc123-zpa-456 (example)
Client ZCC Version	4.x.x.x (example)

The Service Edge hostname often reveals geography (e.g., "-ind-" = India, "-usw-" = US West). Compare the actual connected edge against the user's physical location to determine if selection is appropriate.

User → Zscaler Service Edge Latency Analysis

Path Architecture



Scenario Comparison — Healthy vs Problem

Hop	Healthy RTT	Problem RTT
Laptop to Home Router	5 ms	5 ms
Router to ISP Edge	10 ms	12 ms
ISP to ISP Core	18 ms	40 ms
ISP Core to Transit	—	180 ms ⚠️
Transit to Service Edge	25 ms	210 ms ⚠️

Interpretation

In the problem scenario, latency escalates significantly at the **ISP Core → Transit hop**. This is **before traffic reaches the Zscaler Service Edge**. Fault domain ownership: **ISP / transit routing**, not Zscaler processing.

> Always measure RTT to the Service Edge from the ZCC diagnostics or MTR. If the latency is already high before reaching Zscaler, the investigation goes to ISP/transit — not to Zscaler cloud support.

Reading MTR / Traceroute Output

Example Training Output (Dummy — For Educational Use)

HOST Loss% Avg

1. 192.168.1.1 (Home GW) 0.0% 2 ms
2. 10.20.0.1 (ISP CPE) 0.0% 7 ms
3. Jio-Aggregation-01 0.0% 15 ms
4. Transit-Peer-AS4xxx 0.0% 165 ms

Why Ping Can Mislead — Use TCP/Application Measurements

The ICMP Deprioritization Problem

Zscaler Service Edges and many network devices along the path may **deprioritize ICMP echo responses**. High ping RTT to a Zscaler IP does not automatically mean user application traffic is experiencing the same latency. ICMP is a different traffic class than TCP application data.

Prefer These Over ICMP Ping

Zscaler Cloud Performance Test

Specifically designed to measure ZIA-path performance — TCP-based, not ICMP

ZDX Application Score

Measures actual user experience for specific applications through the ZIA path

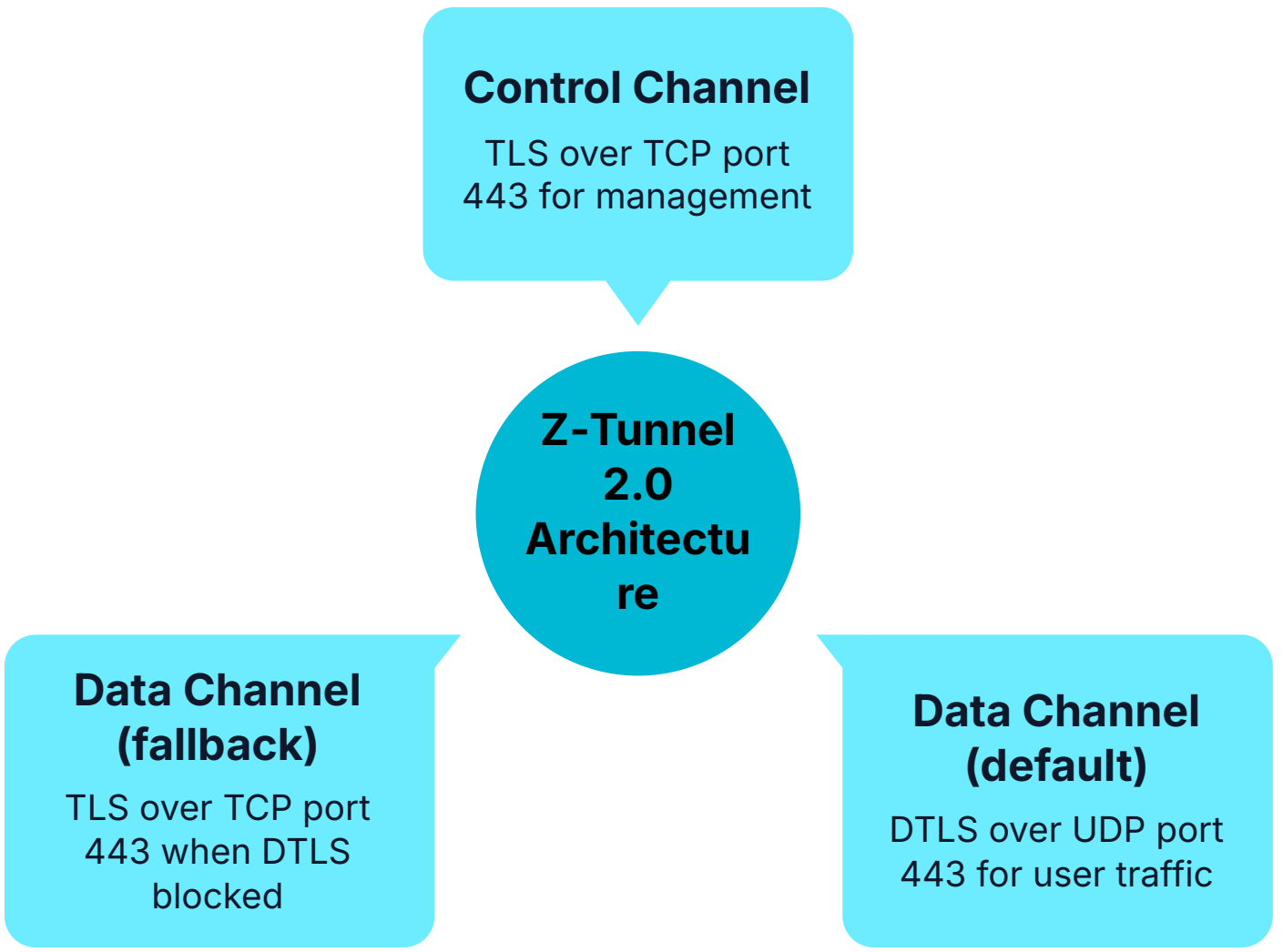
Packet Capture + Wireshark

TCP ACK timing, retransmissions, and server response delays are precise and irrefutable

Browser DevTools Timing

DNS, TCP connect, TLS, TTFB — per-request breakdown of actual application latency

Z-Tunnel 2.0 Transport — DTLS and TLS



Transport Layer Details

Channel	Protocol	Port
Control Channel	TLS (TCP)	443
Data Channel (default)	DTLS (UDP)	443
Data Channel (fallback)	TLS (TCP)	443

Why DTLS for Data?

DTLS (Datagram TLS) provides encryption over UDP, offering **lower overhead and latency** for data-plane traffic compared to TCP-based TLS. Because UDP does not have TCP's inherent retransmission mechanisms, the ZCC tunnel layer manages reliability. ZCC can fall back to **TLS over TCP** when DTLS is unavailable or unstable — depending on client and profile configuration.

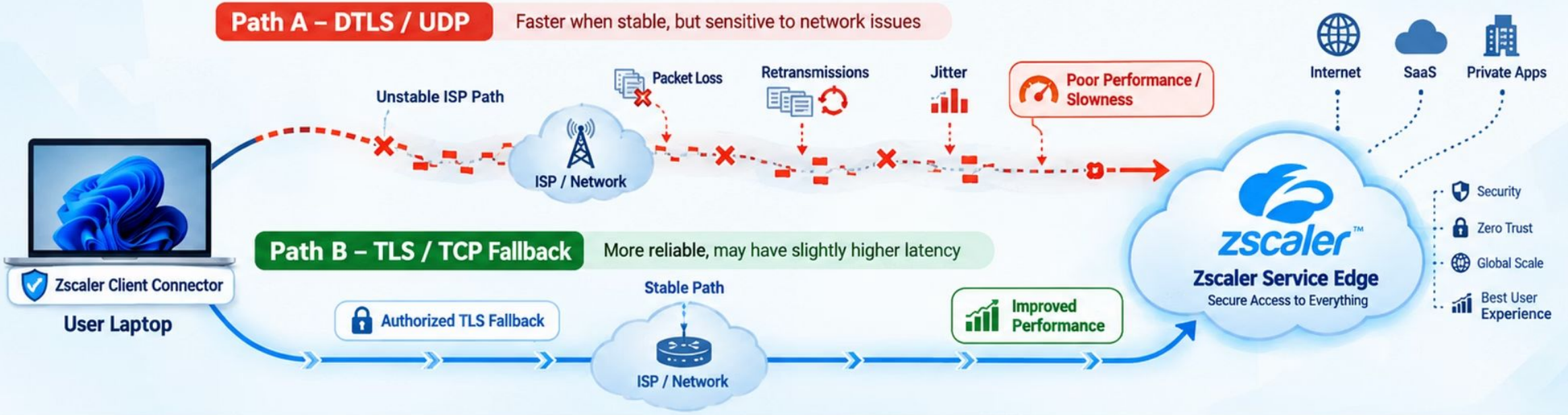
> Confirming whether a user's tunnel is using DTLS or TLS fallback is an essential early step when investigating transport-layer performance issues. Check ZCC diagnostics for current tunnel transport mode.

PEOPLE
SECURE
PROGRESS

Z-Tunnel 2.0 – DTLS vs TLS Performance Troubleshooting

Understand the difference. Troubleshoot with evidence. Ensure optimal user experience.

Secure Users
Secure Apps
A Better Tomorrow



Troubleshooting Flow

Follow a structured approach to identify the root cause.

1. Slowness reported
2. Verify actual tunnel protocol
3. Check packet loss
4. Compare DTLS vs authorized TLS fallback
5. If TLS improves, investigate UDP/DTLS ISP/network path
6. If no improvement, continue Zscaler/application troubleshooting

Do not assume ISP is deprioritizing DTLS without evidence.

Validate with packet captures, network analysis, and real data before drawing conclusions.

DTLS Problem Scenario — Symptom, Hypothesis, Validation

Symptom

OBSERVATION

Corporate broadband works normally. Specific ISP users report: stalls, reduced throughput, application hangs with ZCC enabled

HYPOTHESIS

DTLS (UDP 443) may be experiencing packet loss, throttling, or instability on this ISP path — **requires evidence, not assumed**

EVIDENCE REQUIRED

Compare DTLS vs TLS fallback performance. Capture tunnel diagnostics. Check ZCC for transport mode. Measure packet loss on UDP path.

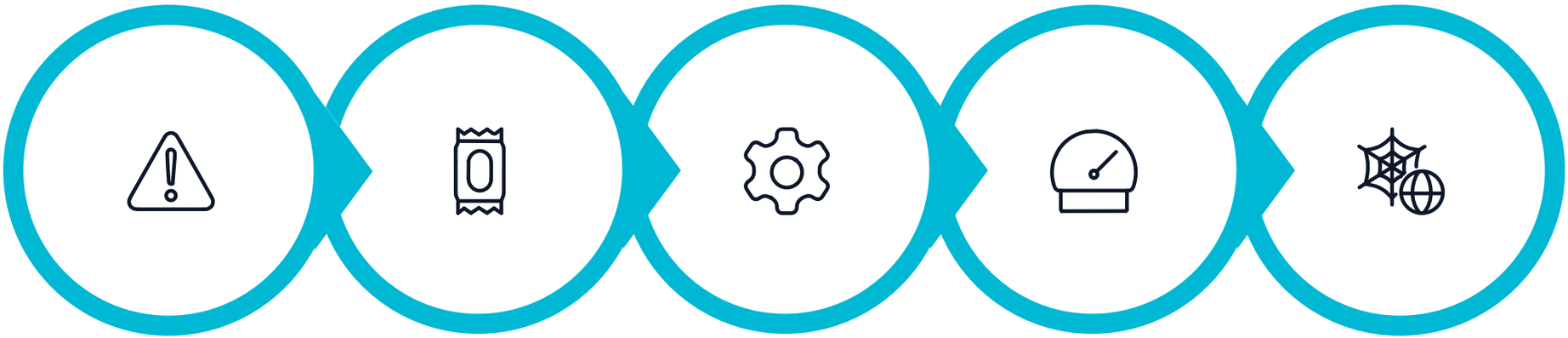
Do NOT Immediately Blame the ISP

DTLS instability could result from:

- ISP UDP deprioritization or rate-limiting (hypothesis — requires traffic evidence)
- Local firewall/UTM that mishandles DTLS
- NAT device with short UDP state timeouts
- MTU mismatch causing fragmentation on the UDP path
- Packet loss on Wi-Fi affecting UDP more severely than TCP
- ISP middlebox interfering with non-HTTP UDP flows

⚠ Never state "DTLS is slow because the ISP deprioritizes it" without traffic-level evidence such as UDP packet loss captures or differential performance tests between DTLS and TLS modes.

DTLS → TLS Fallback Diagnostic Test



DTLS Active | **Collect Evidence** | **Configure TLS Test** | **Measure TLS Perf** | **If Improved**

How TLS Fallback Works

ZCC supports TLS (TCP) fallback for the data channel when DTLS is unavailable or performing poorly. **DTLS instability detection and automatic fallback** may be available depending on the client version and ZCC profile configuration. Consult ZCC configuration documentation for supported fallback behavior in your deployed version.

Important Governance Warning

⊗ Do not recommend permanent transport changes based on a single test. TLS fallback testing should be performed under **change control** with defined rollback. Document: test conditions, before/after metrics, affected users, and whether improvement is consistent. Engage Zscaler TAC before making permanent transport configuration changes at scale.

MTU / Fragmentation — Silent Performance Killer

Why Z-Tunnel Adds Overhead

Z-Tunnel 2.0 encapsulates user packets inside DTLS/TLS, consuming additional bytes of the available MTU. If the physical path MTU (typically 1500 bytes on Ethernet) minus encapsulation overhead leaves insufficient room for the inner payload, packets must be **fragmented or dropped** — causing silent performance degradation.

Symptoms of MTU / Fragmentation Issues

Some websites slow

Sites using large TLS records affected; simple HTTP not

Large uploads fail

Upload stalls or times out at specific file sizes

TLS session establishes but stalls

Handshake succeeds but bulk data transfer hangs

Small packets work fine

DNS, small API calls succeed; large payloads fail

Investigation Steps

01

Check if ICMP "Fragmentation Needed" messages are blocked

Blocked ICMP type 3 code 4 prevents Path MTU Discovery (PMTUD) from working

02

Test with ping using specific payload sizes (Windows)

`ping -f -l 1400 [Service Edge IP]` — test different sizes to find fragmentation point

03

Linux equivalent

`ping -M do -s 1400 [Service Edge IP]`

04

Inspect MSS clamping and TCP window settings

MSS should be reduced to account for tunnel overhead (commonly ~1350–1400 bytes in tunneled environments)

05

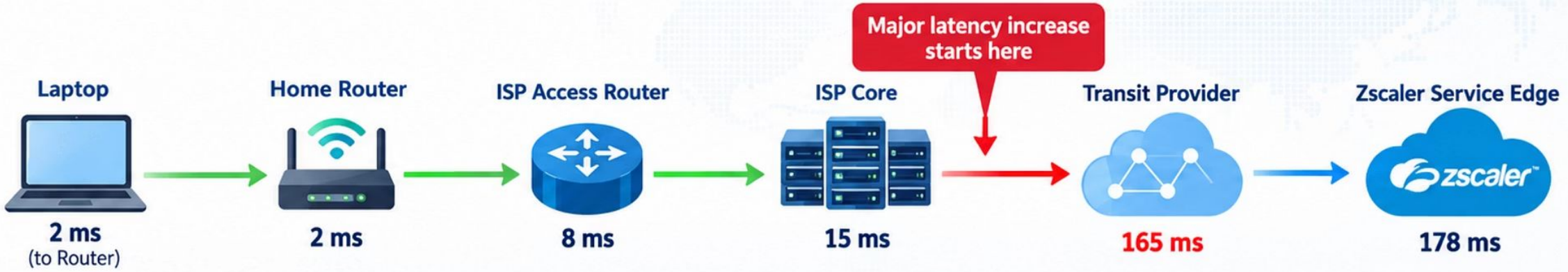
Capture with Wireshark

Filter: `ip.flags.mf == 1` to detect fragmented packets

MTR Analysis – Where Does Latency Begin?

Trace the path. Identify where delay starts. Make informed decisions.

MTR helps you find where latency begins — not just the destination.



Sample MTR Output (Simplified)

Hop	Device/Node	Avg Latency	Notes
1	Laptop / Local Stack	1–2 ms	Healthy
2	Home Router	2 ms	Healthy
3	ISP Access Router	8 ms	Normal increase
4	ISP Core	15 ms	Normal increase
5	Transit Provider	165 ms	Sharp latency jump
6	Zscaler Service Edge	178 ms	Slight increase after jump

Interpretation

Latency begins before Zscaler Service Edge — investigate ISP/transit routing.

Important Note

Intermediate-hop packet loss alone is not proof of forwarding loss. Validate whether latency/loss continues to subsequent hops and application traffic.

Color Guide

Green/Blue = normal progression

Red = abnormal latency jump

MTR shows the path, not just the problem.

Visibility
Find where delay begins

Faster Troubleshooting
Focus on the right segment

Better Decisions
Resolve issues efficiently

What If User → Zscaler Path Is Healthy? Keep Investigating.

Decision Point

User → Zscaler Service Edge RTT is **healthy** (e.g., 25 ms). DTLS is stable. No packet loss on the tunnel path. Yet the application is still slow.

This is one of the most common troubleshooting errors: stopping at the Service Edge and attributing all remaining latency to Zscaler processing. The investigation must continue beyond the Service Edge.

The Path Does Not End at the Service Edge

1

Zscaler Processing

Policy evaluation, SSL inspection, DLP — measure per-request inspection overhead

2

Zscaler Egress

ZIA → internet transit path. Check ZIA egress location vs destination CDN proximity

3

CDN / SaaS Provider

Content delivery routing, regional CDN performance, SaaS provider health

4

Destination Server

Backend server load, database query time, application processing delay

Investigating Zscaler Service Edge / Cloud Issues

Signals That Point to a Service Edge / Cloud Issue

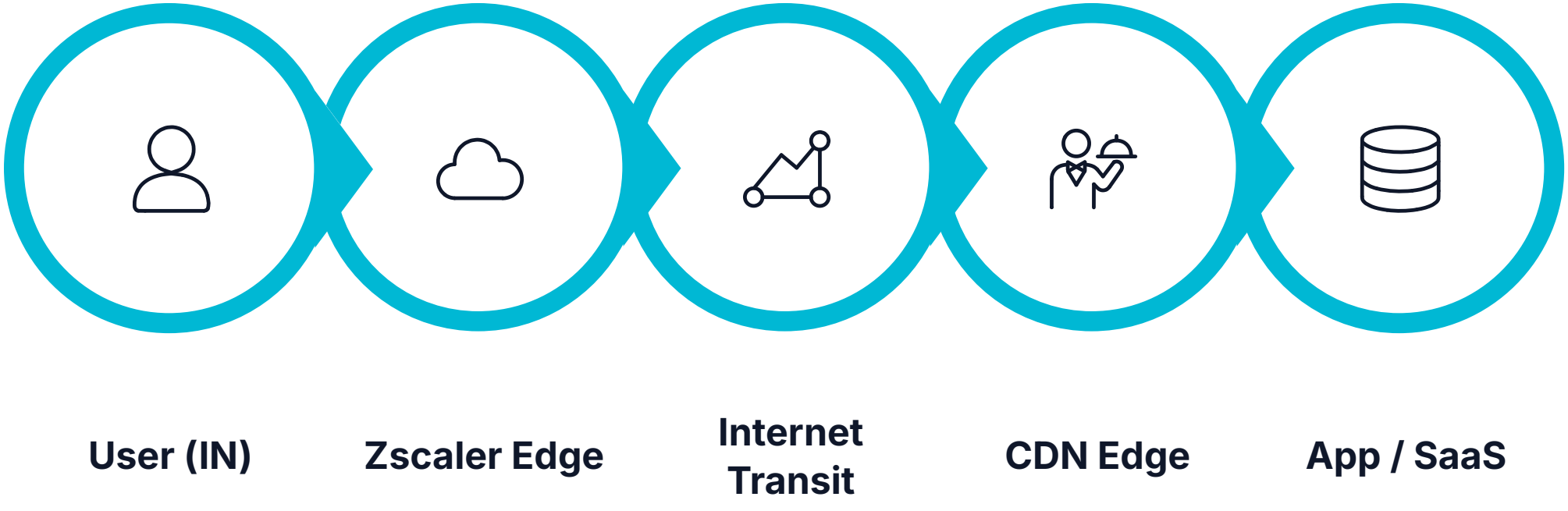
- **Multiple users on same Service Edge affected**
- **Multiple ISPs affected (rules out ISP-specific routing)**
- **Same destination is consistently slow through ZIA**
- **Cloud Performance Test shows degradation**
- **ZDX application score degraded for multiple users simultaneously**
- **Known Zscaler service status event or incident**

Evidence to Collect Before TAC Escalation

- Timestamps of degradation (timezone-accurate)
- Cloud Performance Test screenshots with timestamps
- ZDX evidence showing per-user application scores
- Names of affected Service Edges (from ZCC diagnostics)
- Affected destination URLs/applications
- Reproducible test steps that consistently show the issue
- ZCC log bundles from affected users

ⓘ Always check **status.zscaler.com** first. If an ongoing incident is posted, reference it in your TAC ticket. Correlate degradation timestamps with incident timelines.

Zscaler-to-Destination Latency



The Tunnel Can Be Healthy While the Destination Is Slow

Segment	Example Latency	Owner
User → ZIA Service Edge	25 ms — Healthy	ISP / Zscaler
ZIA → Internet Transit	40 ms — Elevated	Transit / Peering
Transit → CDN Node	85 ms — High ⚠️	CDN Provider
CDN → SaaS Application	Variable	SaaS Provider

Possible Owners of Zscaler-to-Destination Latency

- Internet transit routing / BGP sub-optimal paths
- CDN provider regional performance issue
- SaaS provider backend degradation
- ZIA egress location not peered efficiently with the destination CDN
- Asymmetric routing between ZIA egress and destination

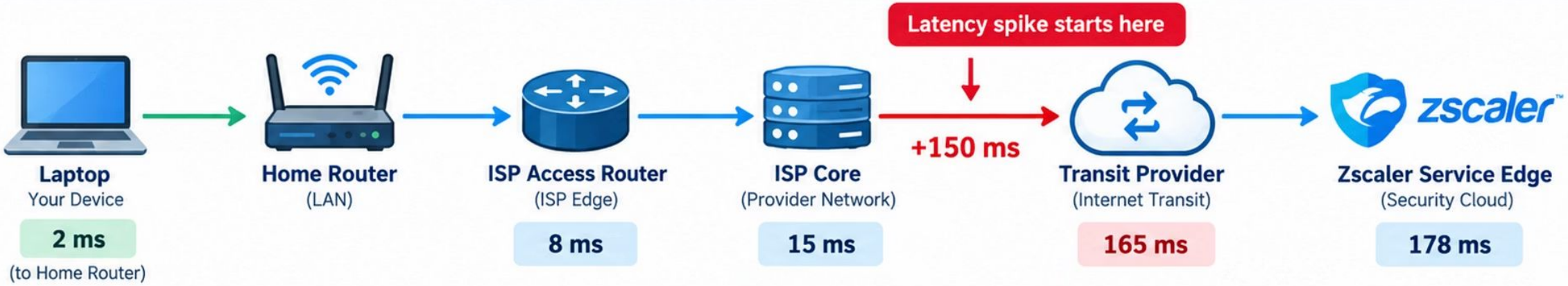
Use the Zscaler Cloud Performance Test to check ZIA-to-destination performance specifically. If ZDX is available, CloudPath metrics show the Zscaler-to-destination segment explicitly.

OBSERVE
TROUBLESHOOT
RESOLVE

MTR Analysis – Where Does Latency Begin?

Trace the path. Find the problem. Solve faster.

Network Visibility
Better Decisions
Secure Outcomes



MTR Output (Example)

Hop	Node	Loss%	Avg Latency	Comment
1	Laptop/Home Router	0%	2 ms	Local hop normal
2	ISP Access Router	0%	8 ms	Normal
3	ISP Core	0%	15 ms	Normal
4	**Transit Provider**	**0%**	**165 ms**	**Latency jump observed**
5	Zscaler Service Edge	0%	178 ms	Increase continues

Interpretation

Latency begins before Zscaler Service Edge — investigate ISP/transit routing.

Important Note

Intermediate-hop packet loss alone is not proof of forwarding loss. Validate whether latency/loss continues to subsequent hops and application traffic.

Key Takeaway

When latency increases before reaching Zscaler Service Edge, the root cause is likely upstream of Zscaler (e.g., ISP or transit provider). Zscaler is the destination, not the source, of the latency.

Path Health Legend

- Normal latency (healthy)
- Latency increase (problem area)
- Continues to destination

SAME TRAFFIC
A CLEARER STORY
STRONGER SECURITY

SSL Inspection Performance Considerations

How SSL Inspection Works in ZIA

When SSL inspection is enabled, ZIA acts as a man-in-the-middle proxy: it terminates the client TLS session, inspects decrypted traffic, and re-encrypts a new session toward the destination. Each step introduces processing overhead that varies with transaction size, cipher suite, and TLS version.

Possible Performance Factors

- **TLS handshake overhead**

  Two TLS sessions established (client→ZIA, ZIA→destination)

- **Certificate validation chain**

  OCSP/CRL checks add latency if resolver path is slow

- **Certificate pinning conflicts**

  Applications that pin their certificate reject ZIA's re-signed certificate

- **Large transaction overhead**

  High-volume file transfers inspected inline are more resource-intensive

- **TLS version / cipher negotiation**

  Older cipher suites may reduce hardware acceleration effectiveness

Diagnostic Approach

CONTROLLED TEST

Compare performance for the specific destination with and without SSL inspection using an authorized policy bypass — under change control and only when organizationally permitted

CHECK LOGS

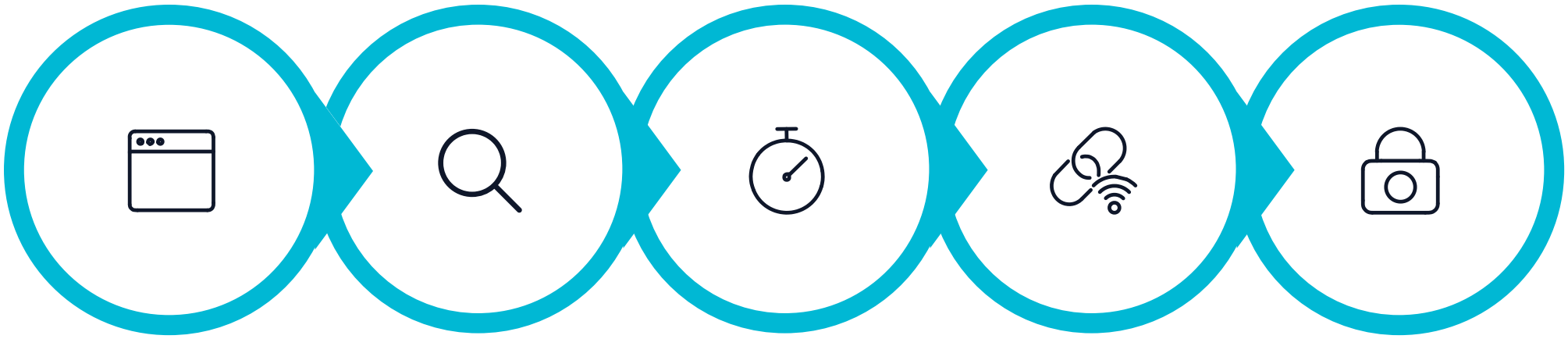
ZIA inspection logs show whether a session is being SSL-inspected, bypassed, or encountering certificate errors

CHECK BROWSER

Verify certificate issuer in the browser: if it shows a Zscaler-issued cert, inspection is active for that session

⊗ **Never permanently disable SSL inspection to solve a performance problem** without organizational security review. Use bypass as a **controlled diagnostic test only**.

DNS Slowness — Often Mistaken for Zscaler Slowness



DNS Investigation Checklist

Resolver Response Time

Measure: `nslookup www.example.com [resolver]` or `dig @resolver www.example.com`

Incorrect DNS Server

Check `ipconfig /all` — is the resolver pointing to the correct internal or public DNS?

Split DNS

Internal FQDNs should resolve via internal resolvers. ZCC split DNS configuration must be validated

Repeated Lookups / TTL

Applications ignoring TTL and re-querying DNS frequently add overhead per transaction

DNS Failures / Retries

NXDOMAIN responses or timeouts trigger retries — visible in packet captures as repeated queries

Browser Developer Tools → Network tab → click a request → Timing panel shows DNS resolution time as the very first timing metric. If DNS is taking 300–800+ ms per request, DNS is the problem — not the Zscaler tunnel or processing.

Endpoint Performance Issues — Don't Overlook the Device

Endpoint Factors That Cause Perceived Zscaler Slowness

High CPU / Low RAM

ZCC inspection + endpoint security + browser compete for resources. CPU >80% sustained causes packet queuing delays

Poor Wi-Fi Signal

Retransmissions on the wireless layer add latency that accumulates through the tunnel. -75 dBm or worse = investigate Wi-Fi first

Conflicting Security Software

Other VPN clients, antivirus with SSL inspection, or proxy agents can intercept ZCC traffic — causing double inspection or routing loops

Outdated ZCC Version

ZCC versions with known performance issues may have been resolved in subsequent releases. Check current Zscaler release notes.

Isolation Technique: Mobile Hotspot Test

Switch the same laptop from corporate Wi-Fi to a mobile hotspot (different ISP). If performance improves dramatically, the issue is the **corporate Wi-Fi/LAN or ISP path**, not ZCC itself.

Additional Endpoint Suspects

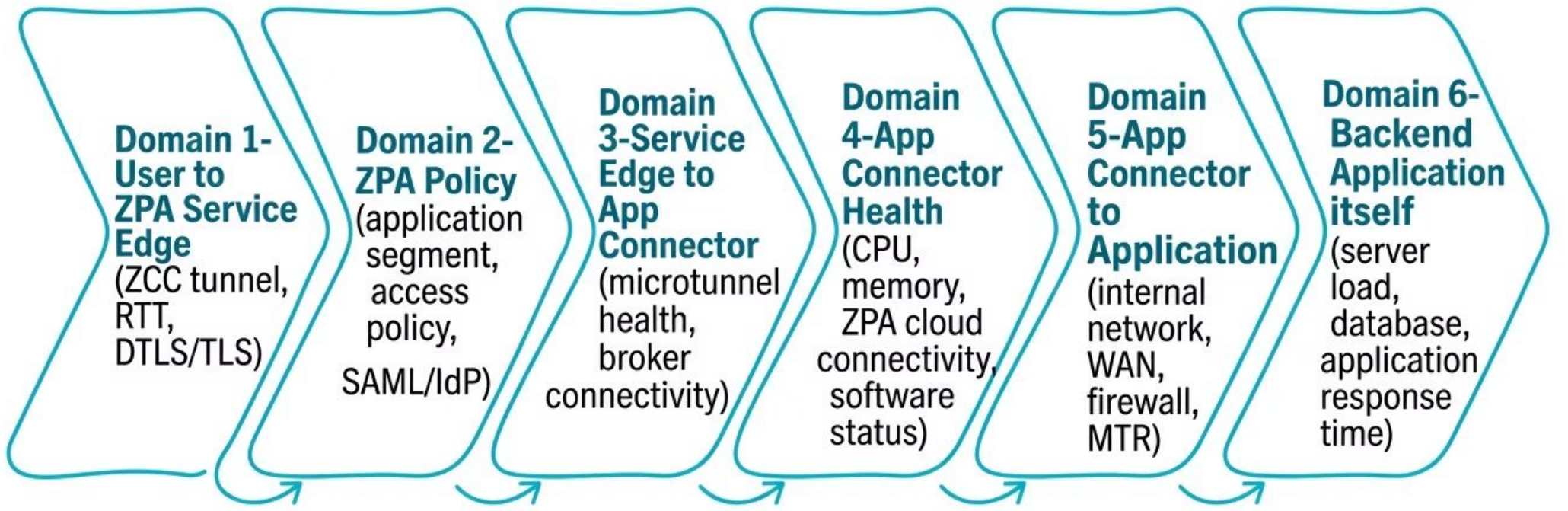
- Damaged or misconfigured TCP/IP network stack
- Local firewall blocking ZCC or DTLS traffic
- Background downloads consuming bandwidth
- Browser extensions intercepting HTTPS sessions
- Proxy settings conflicting with ZCC forwarding

When in doubt: test ZCC on a **clean reference laptop** on the same network. If the clean laptop performs normally, the issue is endpoint-specific.

ZPA Troubleshooting — Six Fault Domains

Do not troubleshoot ZPA as one tunnel.

ZPA involves multiple independent segments, each with its own potential failure modes. Break the path into six measurable fault domains before drawing any conclusion.



Domain	Primary Evidence Source	Likely Owner if Fault Found
User → ZPA Service Edge	ZCC diagnostics, MTR, ZDX	ISP / user network
ZPA Policy	ZPA Admin Portal, access logs	ZPA Admin / Identity team
Service Edge → App Connector	Connector logs, ZPA portal	Zscaler TAC
App Connector Health	Connector status, system metrics	App Connector admin team
Connector → Backend	MTR, tcpdump, Wireshark	Enterprise network / firewall team
Backend Application	Server response time, application logs	Application / server team

ZPA App Flow in Detail — Full Example

Example Environment

User IP	10.10.10.25
Requested URL	https://erp.internal.example
Synthetic IP	100.64.10.50
ZPA Service Edge	ZPA Public Service Edge
App Connector IP	192.168.1.100
Real Backend IP:Port	192.168.1.101:443

Step-by-Step ZPA Flow

1

① User opens erp.internal.example

Browser initiates connection to erp.internal.example

2

② ZCC intercepts — identifies ZPA-protected app

ZCC matches FQDN against ZPA application segment configuration

3

③ ZPA policy authorization

User identity / SAML assertion validated; access policy evaluated

4

④ Microtunnel established

ZPA Service Edge selects appropriate App Connector for the application

5

⑤ App Connector connects to 192.168.1.101:443

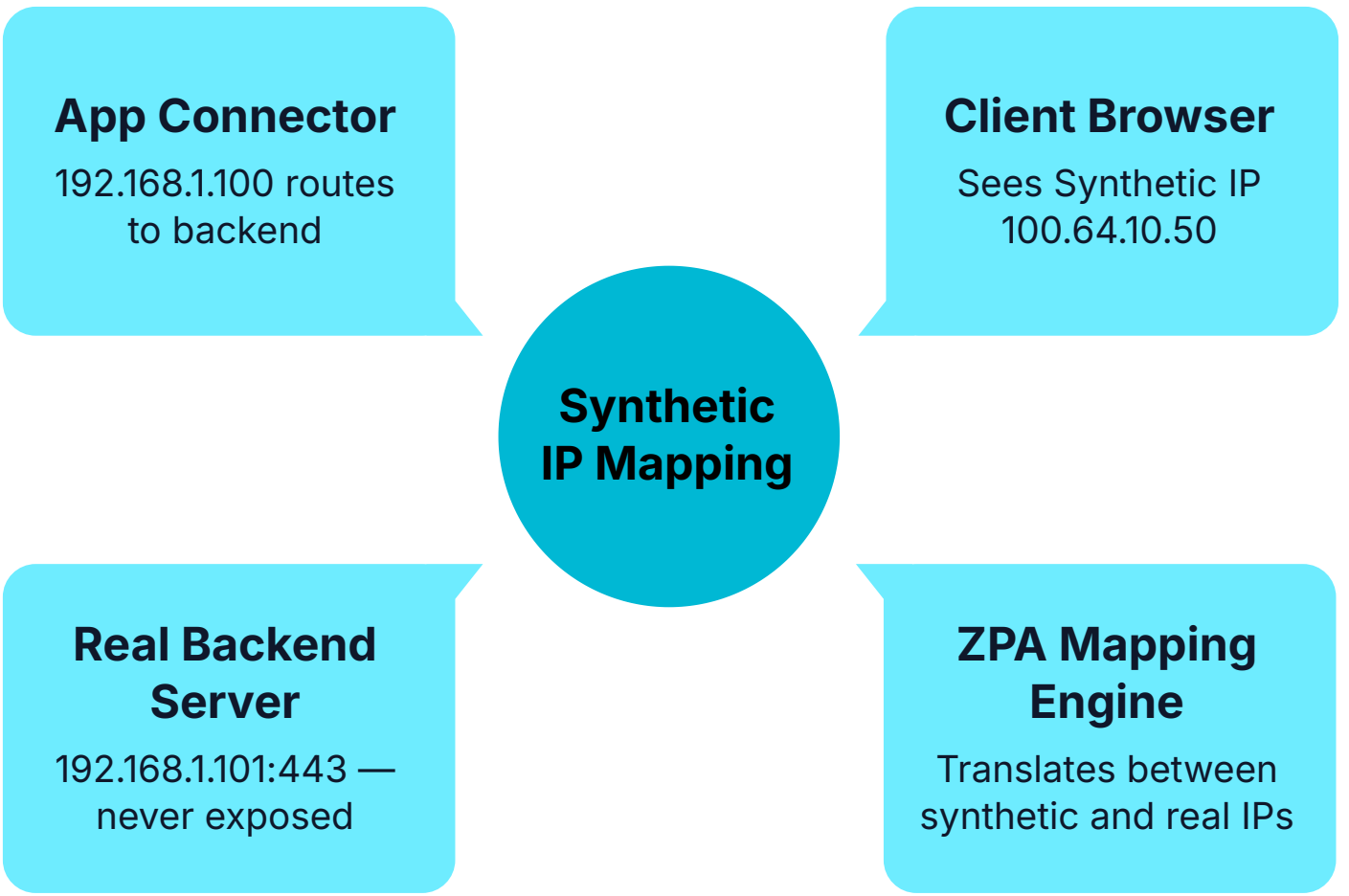
Connector makes outbound TCP connection to the real backend

6

⑥ Response returns to user via Synthetic IP

User sees 100.64.10.50 — real backend IP is never exposed to client

Synthetic IP — How ZPA Hides Backend Addresses



Why Synthetic IP Matters for Troubleshooting

When users or tools perform a `ping` or `traceroute` to the application hostname, the resolved address (e.g., 100.64.10.50) is a ZPA-internal **synthetic representation**. This IP does not route to the backend server. Do not attempt to traceroute to a synthetic IP to diagnose backend connectivity — this will not traverse the real application path.

Correct Diagnostic Target for ZPA Backend Testing

- Log into (or access) the **App Connector host** directly
- From the App Connector, test: `curl -vk https://192.168.1.101`
- From the App Connector, run: `mtr 192.168.1.101`
- From the App Connector: `tcpdump -i any host 192.168.1.101`

> All backend reachability testing in ZPA must be performed from the **App Connector** — not from the user endpoint — because the App Connector is the entity that actually connects to the backend server.

App Connector Health — Comprehensive Checklist

System-Level Health Checks

CPU & Memory

- `top` or `htop`
- Sustained CPU >70% impacts microtunnel capacity

Disk & System Load

- `df -h` — disk space
- `uptime` — load average

ZPA Software Status

- `systemctl status zpa-connector`
- Check for restart loops or error state

Cloud Connectivity

- Can connector reach ZPA cloud endpoints outbound?
- Check ZPA Admin Portal → App Connector status

Application-Side Health Checks

Application Reachability

- `curl -vk https://192.168.1.101`
- `nc -zv 192.168.1.101 443`

DNS from Connector

- `dig erp.internal.example`
- Verify private DNS resolver is configured correctly on the connector host

Connector Placement & Redundancy

- Is the connector in the same site/VLAN as the backend?
- Is redundancy configured (multiple connectors per group)?

ⓘ App Connectors establish **outbound** connections to ZPA services. No inbound firewall ports for user access are required. Ensure outbound HTTPS (TCP 443) to Zscaler cloud endpoints is permitted from the connector host.

App Connector Cannot Reach ZPA Cloud — Log Analysis

Command to Check Connector Service Logs

```
sudo journalctl -u zpa-connector --since "1 hour ago" | tail -100
```

Dummy Training Log Output — Connection Failure (Educational Only)

```
[INFO]  ZPA Connector starting...
[INFO]  Loading configuration from /etc/zpa/connector.conf
[INFO]  Attempting broker connection to broker.zpa.zscaler.net
[WARN]  TCP connection attempt 1 failed: timeout after 30s
[WARN]  TCP connection attempt 2 failed: timeout after 30s
[ERROR] Cannot reach ZPA broker endpoint - checking DNS
[INFO]  DNS lookup for broker.zpa.zscaler.net: NXDOMAIN
[ERROR] DNS resolution failed - check resolver configuration
[INFO]  Retry scheduled in 60 seconds...

--- DUMMY OUTPUT — FOR TRAINING PURPOSES ONLY ---
```

Troubleshooting Connector-to-Cloud Failures

01

DNS

`dig broker.zpa.zscaler.net` — must resolve to Zscaler IP. If NXDOMAIN: DNS misconfiguration on the connector host

02

Internet Reachability

`curl -I https://zpath.zscaler.net` — connector must reach Zscaler cloud HTTPS outbound

03

Firewall / NAT

Verify TCP 443 outbound from connector IP is permitted. Check NAT translations if behind PAT device

04

Proxy

If a corporate proxy intercepts outbound HTTPS, connector must be excluded or proxy-aware

05

System Time

Certificate validation requires accurate system time. Sync NTP: `timedatectl status`

06

Enrollment State

Confirm connector provisioning key and enrollment status in ZPA Admin Portal

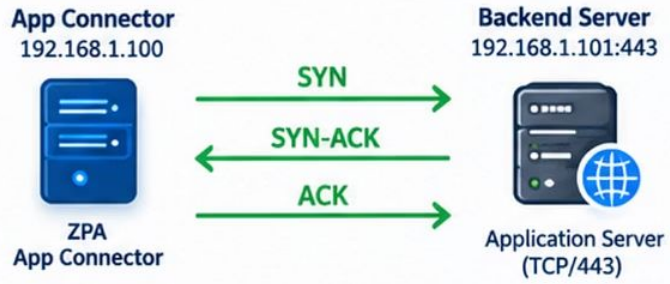
VISIBILITY
TROUBLESHOOTING
SECURE ACCESS
BETTER OUTCOMES

ZPA App Connector → Backend TCP Troubleshooting

SECURE
CONNECT
ENABLE
EVERYWHERE

App Connector: 192.168.1.100 | Backend Server: 192.168.1.101:443

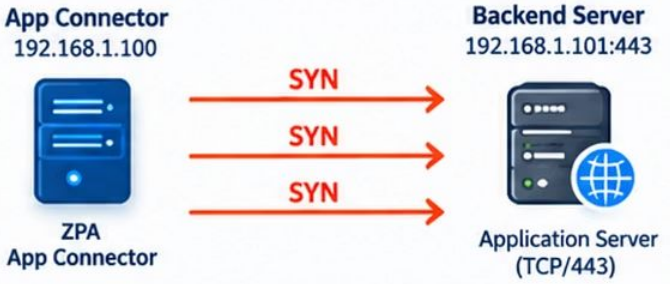
1. Healthy TCP Handshake



What happens?
Standard 3-way TCP handshake completes successfully.

Result
TCP connectivity successful.

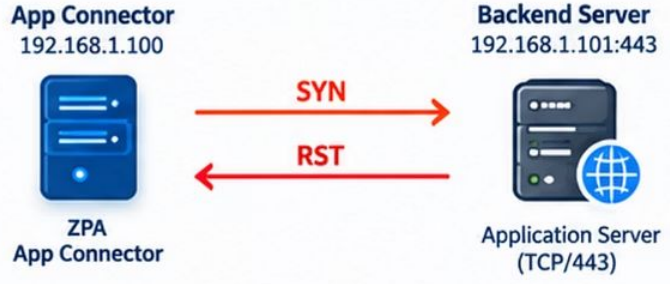
2. Failure – No SYN-ACK



What happens?
No SYN-ACK received from backend. SYN packets may be retransmitted multiple times until timeout.

Result
Likely backend/path/firewall/ return-route issue.

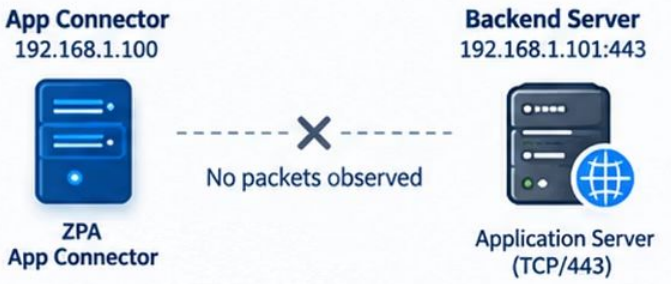
3. RST Received



What happens?
Backend responds with a TCP RST (reset), indicating the host is reachable, but the service is closed or rejecting the connection (e.g., nothing listening on port 443).

Result
Host reachable but service closed or rejecting.

4. No Packets Seen



What happens?
No traffic observed from the App Connector toward the backend server (e.g., using tcpdump).

Result / Next Steps
Investigate application segment, policy, port, DNS, connector selection, or whether connection was attempted.

Useful Commands

Run from the App Connector (or relevant host) to observe traffic to the backend server.

1. `sudo tcpdump -i any host 192.168.1.101 and tcp port 443 -nn -vv`
   Capture and display TCP traffic in real time (verbose output).
2. `sudo tcpdump -i any host 192.168.1.101 and tcp port 443 -nn -w /tmp/backend_traffic.pcap`
   Capture TCP traffic to a file for later analysis.

App Connector → Backend MTR Example

Scenario

App Connector IP	192.168.1.100
Backend Target IP	192.168.1.101
Path	Enterprise WAN / MPLS with 4 logical hops

Dummy MTR Output — App Connector to Backend (Training Only)

HOST Loss% Avg
1. 192.168.10.1 0.0% 1 ms (Local GW)
2. 10.10.20.1 0.0% 3 ms (Core Switch)
3. 10.20.30.1 0.0% 120 ms

App Connector → Backend TCPDump — Healthy Handshake

Capture Command

```
sudo tcpdump -i any host 192.168.1.101 and tcp port 443 \
  -nn -vv
```

Dummy Healthy Output (Training Only)

```
12:01:00.001  192.168.1.100.45872 > 192.168.1.101.443
        Flags [S] seq 1001, win 65535
12:01:00.004  192.168.1.101.443 > 192.168.1.100.45872
        Flags [S.] seq 5001, ack 1002, win 65535
12:01:00.005  192.168.1.100.45872 > 192.168.1.101.443
        Flags [.] ack 5002

--- DUMMY OUTPUT — FOR TRAINING PURPOSES ONLY ---
```

TCP Three-Way Handshake — Interpretation

Packet	Direction	Meaning
SYN	Connector → Backend	Connection request sent
SYN-ACK	Backend → Connector	Backend received and accepted
ACK	Connector → Backend	Handshake complete

SYN → SYN-ACK interval = **3 ms**. This confirms basic TCP reachability from the App Connector to the backend server on TCP/443. The path is clear and the port is listening.

A clean three-way handshake confirms: route exists, firewall permits TCP/443, backend is accepting connections. Proceed to TLS handshake and application-layer testing.

App Connector → Backend TCPDump — Failure Scenarios

Failure: SYN Retransmissions — No Response

```
12:01:00.001  192.168.1.100 > 192.168.1.101 Flags [S]
12:01:01.001  192.168.1.100 > 192.168.1.101 Flags [S] (retransmit)
12:01:03.001  192.168.1.100 > 192.168.1.101 Flags [S] (retransmit)
12:01:07.001  192.168.1.100 > 192.168.1.101 Flags [S] (retransmit)
[Connection timed out]

--- DUMMY OUTPUT — FOR TRAINING PURPOSES ONLY ---
```

TCPDump Result Interpretation Guide

Observed Pattern	Interpretation	Investigate
SYN + SYN-ACK + ACK	✅ TCP connectivity healthy	TLS layer, application response time
Repeated SYN / no SYN-ACK	⚠️ No response from backend	Firewall drop, routing, backend down, ACL, security group
SYN + RST	❌ Destination rejects connection	Port closed, service not running, ACL, incorrect backend IP
No packets at all	❓ Capture not on correct interface	Verify connector-to-backend policy, DNS resolution, App Connector connector group selection, or ZPA application segment port configuration

⚠ Repeated SYN retransmissions without any SYN-ACK is strong evidence that the packet never reached the backend — or the response never returned. Check return-path routing and stateful firewall ACLs in both directions.

Saving PCAP and Analyzing in Wireshark

Save to File for Later Analysis

```
sudo tcpdump -i any host 192.168.1.101 and tcp port 443 \
  -nn -w /tmp/backend_traffic.pcap

# Transfer to analysis workstation
scp user@connector:/tmp/backend_traffic.pcap ./
```

Open in Wireshark — Key Filters

```
tcp.analysis.retransmission
tcp.analysis.duplicate_ack
tcp.analysis.zero_window
tcp.flags.reset == 1
tls.handshake
```

What to Look for in Wireshark

Network Latency Indicators

- TCP ACK timing — RTT between segments
- Retransmissions — packets resent due to loss
- Duplicate ACKs — receiver detecting missing segments
- Zero Window — receiver buffer exhausted (throughput stall)
- TCP Resets — abrupt session termination

Application / Server Delay Indicators

- TLS handshake sequence — time from SYN-ACK to TLS Finished
- HTTP request sent → gap before HTTP response (server processing time)
- Long gap between last client-sent data and server response = **server processing delay**

ⓘ Network latency = delay between packets on the wire. Server delay = silence after the last client request. These are distinctly different and must not be confused when assigning fault domain ownership.

ZPA Backend Server Delay — Application, Not Network

Packet Timeline Example (Dummy — Educational Only)

Event	Time	Elapsed
SYN sent	T+0.000s	—
SYN-ACK received	T+0.003s	3 ms
TLS Client Hello → Server Finished	T+0.013s	10 ms
HTTP GET request sent	T+0.015s	—
⏳ Server silence (no response)	T+0.015s – T+4.515s	4.5 seconds
HTTP 200 OK response received	T+4.520s	—

What This Proves

Network Path = Healthy

SYN → SYN-ACK = 3 ms. TLS handshake = 10 ms. No retransmissions. No packet loss. RTT is normal.

Server Processing = Slow

4.5-second gap after HTTP request = server is computing, querying a database, or otherwise busy before sending a response

⊗ **This must NOT be classified as Zscaler transport latency.** The ZPA tunnel, App Connector, and network path all performed correctly. Ownership: Backend application / server team. Escalate with the PCAP as evidence.

Advanced Slowness Scenarios — Quick Reference

Scenario	Likely Investigation Domain	Evidence to Collect
Wrong / distant Service Edge	GeoIP validation / ZCC selection / TAC	ZCC diagnostics, MTR to SE, hotspot comparison
DTLS unstable / stalls	UDP path / ISP / NAT / MTU	ZCC tunnel mode, TLS fallback test, packet loss capture
Wi-Fi packet loss	Local wireless / access point	Wi-Fi signal strength, switch to hotspot
High CPU laptop	Endpoint resource contention	Task Manager, reference laptop test
MTU mismatch	Path / encapsulation overhead	Fragmented packet capture, ping -f test
DNS delay	DNS resolver / split DNS / TTL	Browser DevTools DNS timing, dig response time
SSL inspection difference	TLS / app compatibility / cert pinning	Controlled bypass test, ZIA inspection logs
One SaaS app slow	Destination / CDN performance	Direct path test, SaaS provider status page
All SaaS slow	ZIA Service Edge / path / ISP	Cloud Performance Test, multiple user correlation
ZPA only slow	ZPA Service Edge / App Connector / backend	ZPA user status, connector health, MTR, tcpdump
One private app slow	Backend app / internal network	MTR from connector, server response time, app logs
All private apps slow	ZPA Service Edge / App Connector cloud connectivity	Connector status, ZPA Admin Portal, connector logs
Backend SYN retransmissions	Backend / path / firewall ACL	tcpdump showing repeated SYN without SYN-ACK
Server response delay (post-handshake)	Application / database / server processing	Wireshark: time between HTTP request and HTTP response
Asymmetric route	Enterprise network / firewall stateful tables	MTR from both directions, firewall session logs

ZIA Slowness – Troubleshooting Decision Tree

A STRUCTURED APPROACH TO IDENTIFY AND RESOLVE INTERNET / SAAS PERFORMANCE ISSUES

Techclick Infosec Pvt Ltd
LEARN | PRACTICE | SECURE

- Diagnose systematcally.
- Find the real cause.
- Enable faster resolution.

User Reports Internet/SaaS Slowness

One User or Many?

- One User → Start with user-side checks (typically an isolated issue).
  - **Check Endpoint / Wi-Fi**
    Device, Wi-Fi, drivers, local configuration
- Many Users → Likely a wider-scope issue (multiple users/sites). Follow full service/path validation.

Determine Actual Zscaler Service Edge
Identify which ZIA Service Edge the user is using (location, IP, node).

Run Zscaler Cloud Performance Test
Test to the identified Service Edge(s) from the user's location.

Compare controlled Zscaler ON vs authorized direct test
Validate if slowness is specific to Zscaler or exists in direct path as well.

Measure User-to-Service-Edge path
Use tools like traceroute, MTR, or Zscaler diagnostics.

Where does latency begin?

- **Latency begins before Zscaler**
  Issue is in the user network or Internet path.
- **Path is healthy**
  Latency starts after Zscaler (ZIA path).

The goal is to identify where latency starts,
using objective data at each step, so the correct team owns the issue.

Follow the evidence, not assumptions.
Each test narrows the scope and proves the fault domain.

INVESTIGATE THE USER-TO-ZSCALER PATH

- **Local Network** — LAN, Wi-Fi, firewall, proxy, local config
- **ISP** — ISP network, last-mile issues
- **Internet Transit** — Upstream providers, routing, congestion
- **GeoIP** — Location mapping, optimal edge selection
- **DTLS investigation** — DTLS connectivity, port 443, UDP, packet loss

INVESTIGATE WITHIN ZSCALER AND BEYOND

- **Check Zscaler processing** — Policy, scanning, DLP, logging
- **Zscaler egress** — Egress location, peering, congestion
- **DNS** — DNS resolution time, DNS policy
- **TLS/SSL Inspection** — Inspection latency, certificate issues
- **CDN** — CDN routing, edge selection
- **Destination application** — Application server performance

OWNERSHIP (BASED ON FINDINGS)

- **User Device** — End user / IT Support
- **Local Network** — Enterprise IT / Network Team
- **ISP** — ISP Provider
- **Zscaler** — Zscaler Support
- **SaaS/CDN** — SaaS / CDN Provider
- **Application Provider** — Application Owner

Prove the fault domain before assigning ownership.

ZDX — Digital Experience Monitoring for ZIA/ZPA

ZDX is an **optional licensed add-on**. It is not required for ZIA or ZPA to function. Its value is in providing continuous, proactive telemetry that dramatically accelerates troubleshooting when licensed.

How ZDX Helps Isolate Slowness

ZDX provides per-user, per-device, per-application telemetry **through the Zscaler path** — not generic network monitoring. It exposes segments that traditional tools cannot see independently.

Device & Wi-Fi Metrics

CPU, RAM, Wi-Fi signal strength, packet loss — identifies endpoint vs network problems

CloudPath Breakdown

Segments: user → network → ISP → Zscaler → application. Latency per segment, not just end-to-end

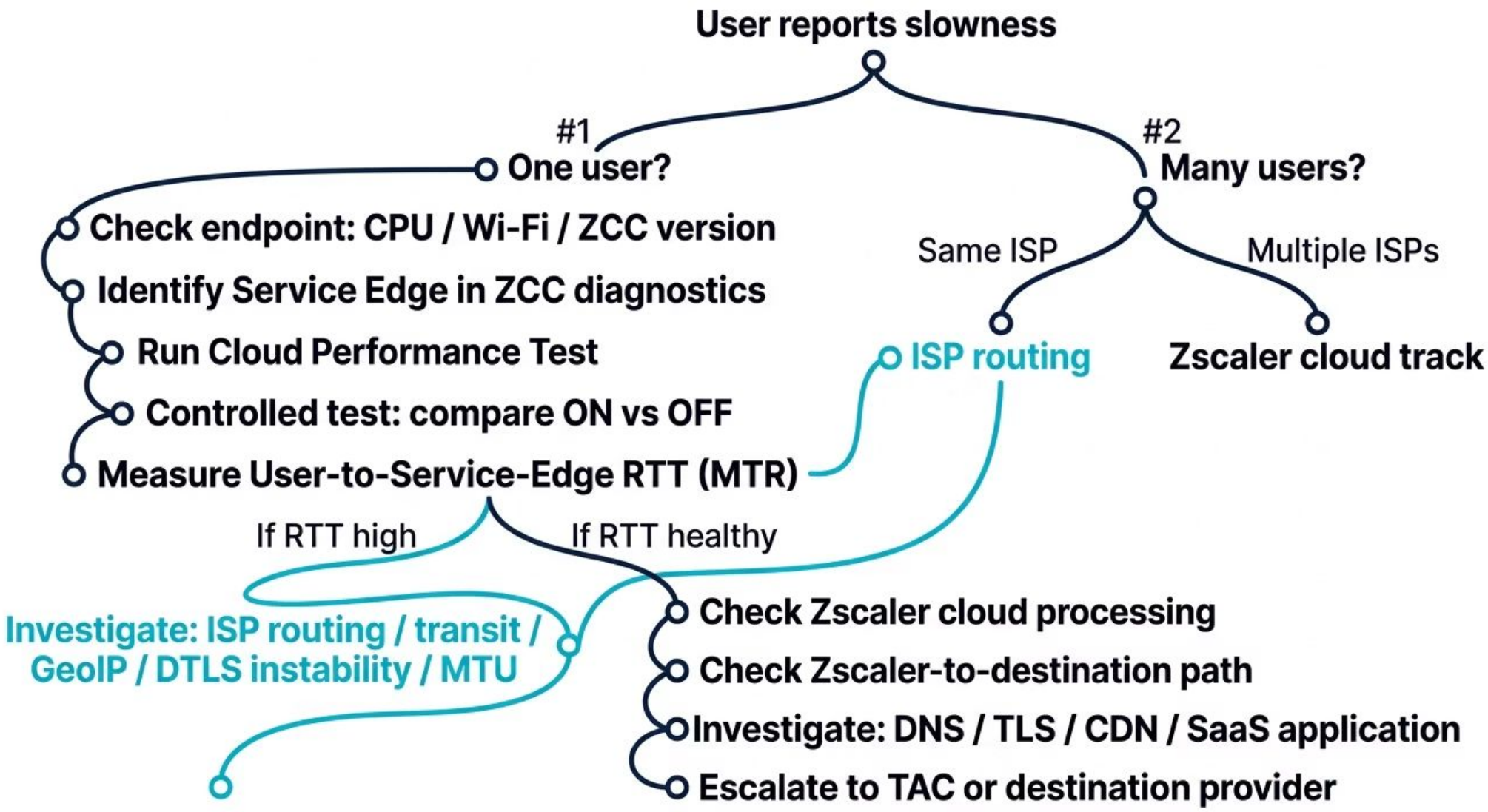
Application Probes & Scores

Web probes measure TTFB, DNS, TCP connect, TLS per application. Score trends show degradation onset time

Packet Loss & Latency Alerts

Continuous measurement enables correlation: "latency increased at 14:30 on Jio users → Service Edge RTT spike"

ZIA Slowness — Master Decision Tree

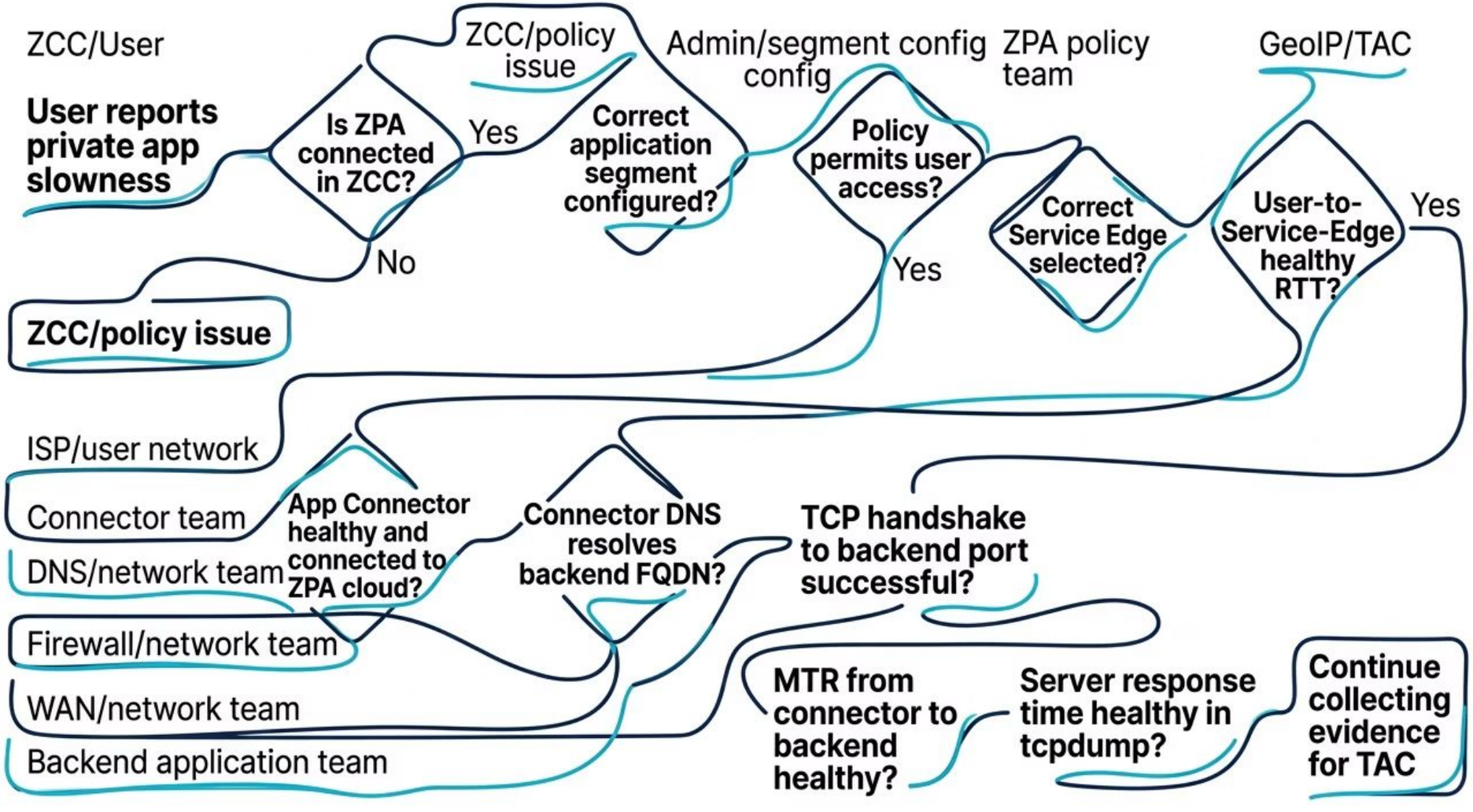


Decision Anchors

Finding	Investigation Track
RTT high before Service Edge	ISP / transit / GeoIP / DTLS
RTT healthy, app slow	ZIA processing / destination / CDN
DNS slow	DNS resolver / split DNS
Only specific app slow	Destination / SaaS provider
All apps slow, multi-ISP	Zscaler TAC escalation with evidence

Every decision in this tree requires **evidence**, not assumption. Document each measurement with timestamp, tool used, and result before moving to the next decision point.

ZPA Slowness — Master Decision Tree



Final fault domain assignments: **USER** | **ISP** | **ZSCALER** | **APP CONNECTOR** | **HOST NETWORK/FIREWALL** | **DNS** | **BACKEND APPLICATION TEAM**

End-to-End Case Study — Jio GeoIP Scenario

Situation Summary

User Location	India (physical)
ISP	Jio (Reliance Jio Infocomm)
GeoIP Reports	United States or Israel (incorrect)
User Complaint	"Everything is slow when Zscaler is on"
Impact	Multiple Jio users on the same prefix affected

Investigation Steps Taken

01

Captured public IP

Confirmed: 203.0.113.25 (Jio ASN) — *example address*

02

Confirmed physical user location

India — verified via HR records and local network equipment

03

Checked MaxMind GeoIP

Result: United States — confirmed incorrect

04

Checked actual ZCC-connected Service Edge

Service Edge hostname indicated non-India region — corroborated GeoIP hypothesis

05

Measured RTT to connected Service Edge

210 ms — significantly elevated compared to India-based edge baseline

06

Tested with mobile hotspot (different ISP)

Different ISP: RTT = 35 ms, India region Service Edge — performance normal

07

Submitted MaxMind GeoIP correction

Included IP range, correct region, MaxMind result screenshot

08

Opened Zscaler TAC case with full evidence package

Attached: ZCC logs, MTR, GeoIP screenshots, Cloud Performance Test, ON/OFF comparison

RCA Statement (Only After Evidence Confirmed)

> "Incorrect GeoIP/service-selection data for the Jio IP prefix caused ZCC to connect users to an unexpectedly distant Service Edge, introducing approximately 185 ms of additional round-trip latency per request. Evidence: MaxMind GeoIP reporting US location for India-based Jio IP range, ZCC connecting to non-India Service Edge, 210 ms RTT on Jio vs 35 ms on alternate ISP. Resolution: MaxMind GeoIP correction submitted for affected prefix range. Zscaler TAC engaged for Service Edge selection validation."

This RCA language is only appropriate because **evidence confirmed the behavior**. In cases where evidence is ambiguous, use: "GeoIP/service-selection mismatch is a possible contributing factor — further investigation required."

Tickets: ISP vs. Zscaler TAC vs. Application Team

📋 ISP Ticket

- Source public IP and ASN
- Timestamps (timezone-accurate)
- Affected destination IP/hostname
- MTR/traceroute showing ISP hop latency
- Packet loss evidence on ISP path
- DTLS vs TLS comparison (if UDP suspected)
- Alternate ISP comparison showing normal performance
- Affected user count and prefix range

🔐 Zscaler TAC Ticket

- Username and organization/cloud ID
- ZCC version and Z-Tunnel version
- Connected Service Edge (ZIA and ZPA)
- Public IP and GeoIP evidence
- DTLS/TLS tunnel state
- Cloud Performance Test results
- ZDX evidence (if available)
- ZCC log bundle
- Destination URL and timestamps
- Reproduction steps
- Packet captures (where appropriate)

🖥️ Application Team Ticket

- Backend IP and application port
- TCP handshake timing (SYN → SYN-ACK RTT)
- Server response delay from Wireshark PCAP
- MTR from App Connector to backend server
- HTTP/TLS timing breakdown
- Application error logs or HTTP status codes
- Server CPU/memory metrics at time of issue
- PCAP file (backend_traffic.pcap)

Submit the ticket to the correct team the first time. Sending a server response delay problem to ISP support (or vice versa) wastes resolution time and creates confusion. Let the evidence guide the escalation path.

Engineer Command Cheat Sheet

Windows — Endpoint & Path Diagnostics

```
# Network interface info
ipconfig /all

# DNS resolution test
nslookup www.example.com
nslookup erp.internal.example 10.10.1.1

# Trace route (ICMP — may not reach Zscaler SE)
tracert zpa-se.example.zscaler.net

# Path + loss statistics
pathping 8.8.8.8

# TCP port reachability test (preferred over ping)
Test-NetConnection -ComputerName portal.zscaler.com -Port
443

# Flush DNS cache
ipconfig /flushdns
```

Windows — MTU Test

```
# Test specific packet size (no fragmentation flag)
ping -f -l 1400 203.0.113.1
ping -f -l 1350 203.0.113.1
```

Linux / App Connector — Path & Capture

```
# Interface and IP information
ip addr show
ip route show

# DNS resolution
nslookup erp.internal.example
dig @10.10.1.1 erp.internal.example

# HTTP/HTTPS reachability from connector
curl -vk https://192.168.1.101
curl -Iv https://erp.internal.example

# Traceroute to backend
traceroute 192.168.1.101

# MTR (continuous traceroute with stats)
mtr --report 192.168.1.101

# Live packet capture
tcpdump -i any host 192.168.1.101 -nn

# Capture TCP/443 verbose
tcpdump -i any host 192.168.1.101 and tcp port 443 -nn -vv

# Save to file
tcpdump -i any host 192.168.1.101 and port 443 -nn -w
/tmp/capture.pcap

# View ZPA Connector logs
sudo journalctl -u zpa-connector --since "1 hour ago"
sudo journalctl -u zpa-connector -f
```

⚠ Note: Zscaler Service Edges and cloud endpoints may not respond to ICMP (ping/traceroute). Use TCP-based tests (`Test-NetConnection`, `curl`, Cloud Performance Test) as primary evidence. ICMP-based tools are supplementary only.

Final Troubleshooting Framework — 10-Step Methodology

01

STEP 1 — Define Scope

One user, one ISP, one location, one app, or all users? Scope determines the investigation track before any tool is run.

02

STEP 2 — Identify Traffic Type

Is this ZIA (internet/SaaS) or ZPA (private application)? Each has a fundamentally different path and fault domain map.

03

STEP 3 — Identify Actual Service Edge

Read from ZCC diagnostics — not assumed from GeoIP. Verify ZIA Service Edge and ZPA Service Edge independently.

04

STEP 4 — Measure User → Zscaler RTT

MTR, Cloud Performance Test, ZDX. Establish whether latency exists before the Service Edge — if yes, investigate ISP/transit.

05

STEP 5 — Validate Z-Tunnel Transport

Confirm DTLS or TLS mode. Check for packet loss, fragmentation, or MTU issues in the tunnel path.

06

STEP 6 — Measure Zscaler → Destination

Use Cloud Performance Test and ZDX CloudPath. Investigate ZIA egress routing, CDN proximity, and destination health.

07

STEP 7 — (ZPA Only) Validate App Connector → Backend

Check connector health, DNS resolution, TCP handshake via tcpdump, MTR, and server response time in Wireshark.

08

STEP 8 — Capture Evidence

ZCC logs, MTR output, tcpdump/PCAP, Cloud Performance Test screenshots, ZDX reports — timestamped and preserved.

09

STEP 9 — Assign Correct Fault Domain

Apply: Observation → Hypothesis → Evidence → Root Cause. Never assign ownership without evidence supporting it.

10

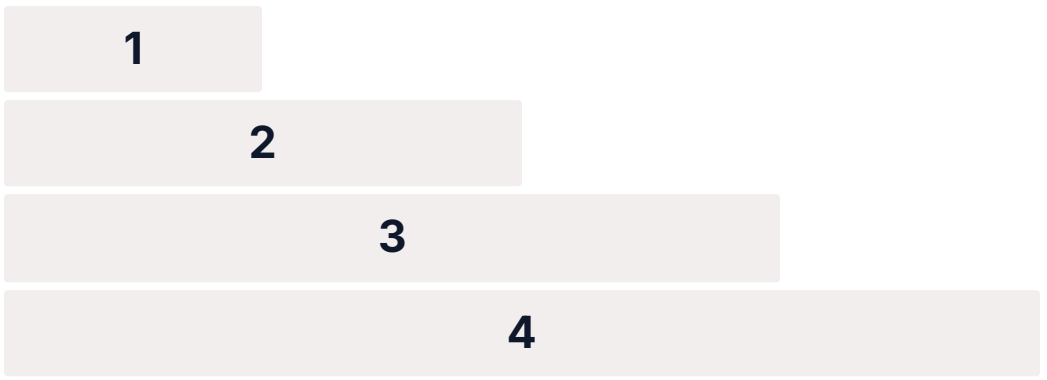
STEP 10 — Escalate With Evidence

Submit to the correct team (ISP, Zscaler TAC, App team) with a complete evidence package and reproducible test steps.

> **Do not blame Zscaler, the ISP, or the application until the packet path proves exactly where latency begins.**

Key Takeaways & RCA Discipline

The Four-Stage RCA Model — Always Required



1 **① OBSERVATION**

What the user reported, verbatim. "Zscaler is slow" is not an observation — "RTT to Service Edge is 210 ms on Jio" is.

2 **② HYPOTHESIS**

Possible causes that fit the observation. Multiple hypotheses are acceptable. Label each as unconfirmed until validated.

3 **③ EVIDENCE**

Measured, reproducible data: MTR output, tcpdump, Cloud Performance Test, ZCC diagnostics, Wireshark PCAP. Timestamps required.

4 **④ ROOT CAUSE**

The specific mechanism confirmed by evidence. Only one segment can be the root cause per incident — assign it precisely.

What Good Engineers Never Do

Never say "high ping = Zscaler problem"

ICMP may be deprioritized. Use TCP/application measurements.

Never say "MaxMind says US = user goes to US"

GeoIP and Service Edge selection are separate. Validate both with ZCC diagnostics.

Never say "DTLS is slow because ISP throttles it"

This requires UDP packet loss evidence and TLS comparison testing to support.

Never permanently bypass security for performance

Bypass is a controlled diagnostic tool only — not a resolution. Requires organizational approval.

Never stop troubleshooting at the Service Edge

Healthy tunnel ≠ healthy destination. The path continues — always investigate end-to-end.

Reference Summary — Official Documentation Sources

Primary Reference Sources

Topic	Source
ZIA Architecture & Administration	help.zscaler.com → ZIA Documentation
ZPA Architecture & Administration	help.zscaler.com → ZPA Documentation
Zscaler Client Connector (ZCC)	help.zscaler.com → Client Connector Documentation
ZDX Digital Experience Monitoring	help.zscaler.com → ZDX Documentation
Z-Tunnel 2.0 Technical Reference	help.zscaler.com → ZIA → Z-Tunnel
Cloud Performance Test	Zscaler Admin Portal → Troubleshooting tools (Portal path may vary by Zscaler UI version)
Zscaler Service Status	status.zscaler.com
MaxMind GeoIP Correction	dev.maxmind.com → GeoIP Data Correction Request
RFC 8805 Geofeed Format	datatracker.ietf.org/doc/html/rfc8805
App Connector Deployment Guide	help.zscaler.com → ZPA → App Connector

Training Notes on Sources

Always use official Zscaler Help as primary reference

Portal paths and menu names change between Zscaler UI versions. Confirm current menu locations in your deployed version.

Zscaler release notes track ZCC behavioral changes

Performance-relevant ZCC changes, DTLS fallback behavior, and tunnel improvements are documented in release notes.

TAC knowledge base articles

Available to licensed customers via the Zscaler Support Portal — search by symptom or error message for specific KBs.

This training presentation was produced for educational purposes. Always validate configurations, commands, and procedures against the official documentation version applicable to your deployed Zscaler cloud and ZCC version.

zscaler™
Secure Access for a Better World

ZPA Private Application Slowness – Complete Troubleshooting Flow

Isolate. Validate. Identify. Resolve. Faster.

Zero Trust
Stronger Security
Brighter Possibilities

ZPA Troubleshooting Best Practices

- Follow a structured approach
- Check each segment independently
- Use data and evidence (traces, logs, metrics)
- Identify the right ownership
- Collaborate for faster resolution

Common Tools

- ZCC diagnostics
- ZPA Admin Portal
- Connector logs
- nslookup / dig
- telnet / nc
- mtr
- tcpdump / wireshark
- Application logs

Different segments. Different ownership. Faster resolution.

#	TROUBLESHOOTING CHECK (QUESTION)	YES (Continue)	NO (Ownership / Action)	HELPFUL CONTEXT / WHAT TO CHECK
1	ZPA Connected?	Yes	USER or ZSCALER	Verify ZCC status (connected, no errors)
2	Correct App Segment?	Yes	ZSCALER	Validate application mapping (segment, app, domain)
3	Access Policy Allows?	Yes	ZSCALER	Check policy and posture conditions
4	Correct Service Edge?	Yes	ZSCALER	Confirm selected service edge / path
5	User-to-Service Edge RTT Healthy?	Yes	ISP	Measure RTT / latency (zcc diagnostics, ping, mtr)
6	App Connector Healthy?	Yes	APP CONNECTOR	Check connector health and registration
7	Connector Connected to ZPA Cloud?	Yes	APP CONNECTOR or ZSCALER	Verify connector is connected to ZPA cloud (status, logs)
8	Backend DNS Resolves?	Yes	DNS	Test DNS (nslookup / dig)
9	TCP Port Reachable?	Yes	FIREWALL & NETWORK or BACKEND SERVER	Test TCP reachability (telnet / nc)
10	MTR Healthy?	Yes	FIREWALL & NETWORK	Use MTR to identify latency path
11	TCPDump Handshake Healthy?	Yes	FIREWALL & NETWORK or BACKEND SERVER	Inspect SYN / SYN-ACK / ACK in tcpdump
12	Backend Server Response Healthy?	Yes	APPLICATION TEAM or BACKEND SERVER	Validate application response time (response codes, logs)

ZPA Private Application Access Healthy (Performance Issue Resolved)

OWNERSHIP CATEGORIES

- **USER** — End user device, ZCC, local setup and user environment
- **ISP** — Internet service provider (connectivity and latency)
- **ZSCALER** — ZPA cloud, policy, service edge, platform issues
- **APP CONNECTOR** — Connector health, registration and connectivity
- **FIREWALL & NETWORK** — Network path, firewalls, routing and ACLs
- **DNS** — DNS resolution and records
- **BACKEND SERVER** — Server reachability, OS, services and infrastructure
- **APPLICATION TEAM** — Application performance, logic, or errors

⚠ Do not troubleshoot ZPA as a single tunnel. Isolate each segment.

Thank You — Zscaler ZIA & ZPA Performance Troubleshooting Masterclass

You now have the methodology, tools, and decision frameworks to systematically isolate latency across every segment of the ZIA and ZPA traffic path — and produce evidence-backed root-cause analysis that stands up to scrutiny.

🎯 Master Principle

Path isolation before conclusion. Measure every segment. Assign ownership with evidence.

🛠️ Core Tools

ZCC Diagnostics · Cloud Performance Test · ZDX · MTR · tcpdump · Wireshark · Browser DevTools

📋 RCA Discipline

Observation → Hypothesis → Evidence → Root Cause. Always in that sequence. Never skip ahead.

📞 Get Help

Zscaler TAC: status.zscaler.com · support.zscaler.com · Open tickets with complete evidence packages

---

Techclick Infosec Pvt Ltd | ai.techclick.in | Training Contact: WhatsApp +91 92772 29456