JUNIPER NETWORKS | Engineering Simplicity

eGuide

WAN Troubleshooting: an eGuide to Swifter Resolution

Enter

Introduction

Wide area networks (WANs) are complex clouds that carry a range of critical and non-critical business traffic.

Most WANs leverage service providers and involve multiple underlying transport types. In recent years there has been an increased reliance on dedicated internet access (DIA) circuits to build and supplement WANs at a lower cost.

In some cases, WANs are built wholly as an overlay or VPN on these internet circuits. Rather than using private connections, traffic then shares a path with a local internet breakout.

Monitoring and observing a WAN (including its component parts) is crucial to ensure the confidentiality, integrity, and availability of the business traffic it serves.

Most WANs rely either directly or indirectly on dynamic routing protocols. These protocols account for failures in underlying elements and subsequent reachability across paths.

Although failures are not inevitable, they are still highly probable due to everything from device issues, congestion, and cable breaks, to human error.

With the rise of automation and continued abstraction levels, we sometimes encounter leaks that involve logic or routes that can undermine our ability to assure services.

This guide provides some practical steps and approaches to help you when things go wrong.

When things go wrong, there are multiple approaches to achieve service restoration.

The aim is always the same: minimize MTTR, while maximising MTBF.

How is risk distributed throughout your WAN?

How do you classify it?

What do you do to deal with the inevitable impact of failures or congestion?

Troubleshooting Approaches

The internet (or an IP network) has been described as a "series of tubes." It's perhaps better thought of as a series of sessions and messages taking place between groups of managed and unmanaged devices.

Messages may be stateless, but sessions retain a concept of "state." Both of these types of traffic create and have multiple dependencies.

Dependencies introduce risk.

Troubleshooting involves active probing and searching. One of the most efficient ways to troubleshoot is to continually halve a problem space, similar to a binary search. You validate as you go, but symptoms, problem reports, and monitoring don't always reflect the underlying problem state.

Often an engineer must use correlation to direct efforts while tracking down a specific cause. The goal is to uncover the root cause as quickly as possible, triage or mitigate it, and then work towards corrective actions to prevent a reoccurrence.

Troubleshooting may use a "top-down" or "bottom-up" approach; yet both can lead to inefficiencies, wasted effort, and slower outcomes. A "middle-out" approach can yield faster answers in the face of initially ambiguous conditions.

When an engineer or system receives a trigger that something is wrong, independent or secondary verification is usually required. This is "trust but verify" in action, as even if the source of the trigger or alert is trusted, there are always doubts that arise when embarking on a costly troubleshooting exercise.

Automation is advantageous in most validation and verification phases: reducing toil, accelerating outcomes and leaving human operators more time for higher-order complex interactions.

User Monitoring

The adage is still true, “you can’t manage what you can’t measure.” It is essential to have visibility of your assets and services, while filtering the data you receive to minimize distracting noise.

It’s important to achieve a balance between instrumenting “all the things” to understand state, while being able to prioritize rapidly actionable alerts.

Leveraging user sessions

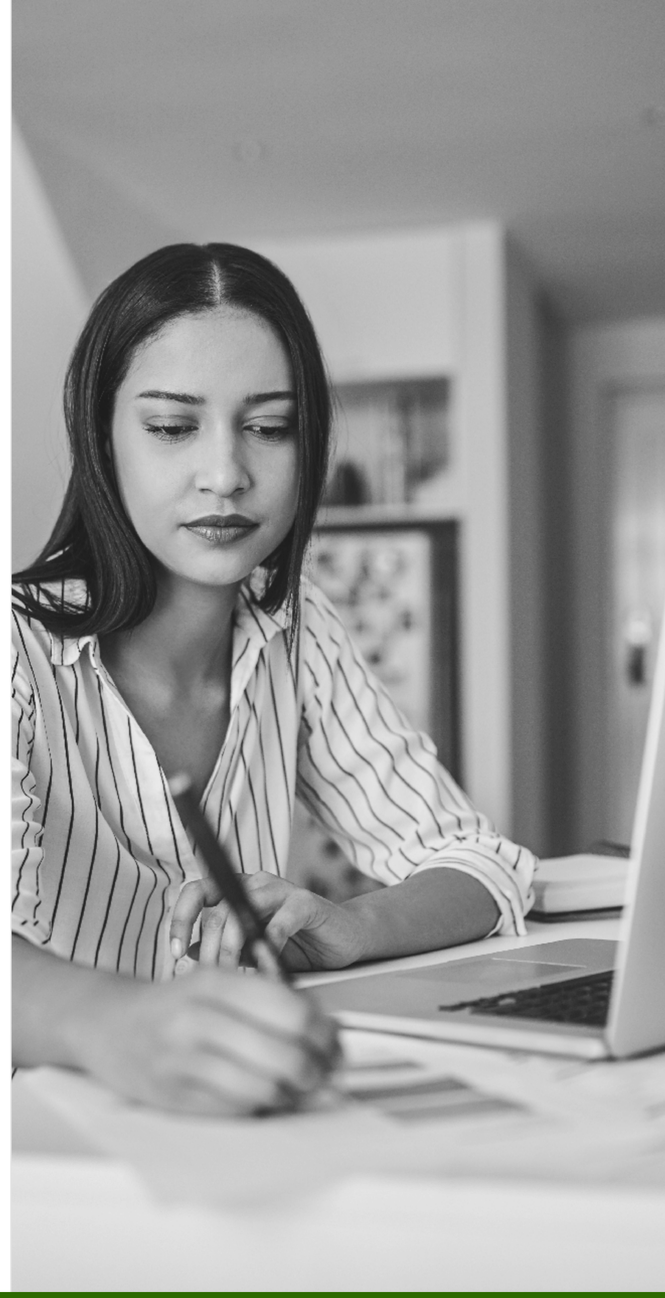
Users care about their specific data and sessions. They operate complex devices and run swathes of applications that can be difficult to troubleshoot. When problems arise, it may appear to them that the network is not working, but it is not their remit or responsibility to identify where a technical failure actually lies.

Yet while users may sometimes report faults inaccurately, both they and their machines are extremely useful as indicators of where problems may lie. Their sessions can help you to establish root causes quickly, including revealing the true health of the WAN.

From user sessions, control groups can be created for identifying individual or aggregate network problems. Up until recently, these have been expensive to track and instrument.

Traditional manual troubleshooting may still be used, but our toolbox now has better diagnostics available for scenarios that used to be both a time sink and a political minefield.

Now we can embrace real user monitoring on the WAN and engage in automated “bottom-up” troubleshooting. This accelerates the speed at which we can diagnose and classify WAN problems, enabling faster fixes and restored services.



Network Monitoring and Observability

If we know what causes a problem, there should be no need for troubleshooting. Instead, we can monitor for that non-desirable state in the relevant components or elements.

The better the overall observability of the network, the better our ability is to infer system state from outputs, and the faster and more fine-grained the problem space can be partitioned to find answers.

Protocols, signalling, datagrams, and packets are all well-defined and should follow the rules. However, the network itself is a complex distributed system: one device's configuration informs another device's state.

Traditional network monitoring used to focus on the health of network elements. While this is still extremely important, we need to prove that a specific user's sessions data can be reliably transported from A to B.

This means we require higher cardinality data, and the burden of proof still falls to operational teams to confirm not just availability, but also in-path reachability from the user perspective.

When a system grows to such complexity that dependencies and shared states prevent us from easily finding a root cause, the reality is that we still need to troubleshoot.

- Outputs from systems and humans may be correct, but can introduce bugs and biases.
- Gathering too much data can overload us, so knowing what to look for in advance does not remove the need for verification.
- Monitoring helps measure known problematic states or thresholds, yet we often need to go further to determine a root cause.

Searching for a temporarily unknown failure implies we have not yet been able to instrument or monitor for it. This querying and searching is the essence of troubleshooting.

As complexity grows, the variables that can impact traffic multiply.

Demonstrating the network is functioning correctly becomes a non-trivial task.

WAN Troubleshooting

A WAN may be comprised of many types of topologies, protocols, vendors, and elements.

Here are some high-level general guidance and pointers to help expedite your efforts. The goal should then be to operationalize your own customized troubleshooting process, modeling and documenting it to enable better knowledge sharing and rapid automation.

General Recommendations

Always define and then continuously refine **the problem statement.**

Classify scope and impact as it relates to your business needs and criticality rating.

Triage and restore functionality as quickly as possible. Hold a blameless post-mortem if the root cause remains unknown afterwards.

Always ask to see **fresh data and empirical evidence.**

Document as you go. Capture and share (if permissible) all data, snippets, timestamps, and reports.

Constantly **revisit first principles** to reason about an issue.

Occam's Razor holds true in most cases. History is also important: **what changed?**

As you partition the problem space, **focus on the differences** as well as the commonalities.

Just because you can't see it, it doesn't mean it's not happening.

The simpler your designs, the easier they are to troubleshoot.

Prerequisites

Network element monitoring is active for device-level health-checks (including any aggregate virtual devices).

Interface level monitoring and trending is in place, including the correct speeds, thresholds, and buffer or queueing statistics.

The ability to trace routes from the user location or site.

Access to the routing information base (RIB) and forwarding information base (FIB) is available for all relevant managed network elements, including the user's device, if possible.

Centralized logging and querying, preferably based upon UTC timestamps with millisecond granularity.

Assumptions

1. User endpoint is one of a group of devices with the same problems (rather than unique host issues).
2. User endpoints can pass traffic to other destinations correctly (not traversing the WAN).
3. All data collected (anecdotal or otherwise) is useful but potentially incorrect, until independently verified or checked in a system of record.
4. A WAN issue may be constant or intermittent.
5. The problem lies in the WAN as per (1) and (2) and is not a client-side host routing, authentication, client VPN issue, etc.
6. An IPv4 stack throughout (as opposed to dual-stack or IPv6 only).
7. ICMP echo request (type 8), echo reply (type 0), and time exceeded (type 11) are enabled end-to-end on network elements on the path from source to destination.
8. Operational teams know the layout of their managed topology. Documentation or dynamic mapping is up-to-date for all managed infrastructure nodes.
9. Network monitoring does not report any known or relevant issues causing an impact within the noted time window.

Resolution and Reachability

1. **Establish and ensure the use of the actual IP addresses from the client's source interface and the remote service or destination.** This involves checking whether a named resource is being used and how it resolves from the client's perspective.

2. **Send an ICMP echo (ping) from the client's default gateway source IP address to the destination address** (set the DF bit and use a packet size expected to be able to pass end-to-end). If unsuccessful...

3. **Trace the route from the client's default gateway source IP to the destination IP** (use ICMP but consider using UDP or TCP if permitted). If it fails or involves unexpected nodes...

4. **Ensure IP reachability follows the expected path** and check the RIB of the last known "good" hop.

5. **Check relevant access control lists (ACLs), firewall policies, and interface maximum transmission units (MTUs)** on the last known "good" hop, and also in-path on those that are one hop away.

6. **If IP reachability is proven, start from the source subnet and test the relevant TCP or UDP ports for the service in question.** Tools such as telnet, tcptraceroute, tcping, curl, hping3, nmap, or nc may require access to a general-purpose computing device if not available on the network device.

7. **Almost all TCP services will respond to a SYN;** however, many UDP services may not respond unless the message is structured correctly for the service in question.

8. **Independently validate the remote service is listening on the correct port** (if possible) and repeat the steps from (1) from the remote subnet's perspective.

In the absence of controlling the source endpoint to perform testing and fully emulate the user, the initial and closest layer 3 interface should be used (usually the client's default gateway).

Network elements usually have a subset of troubleshooting tools on the command line; however, they are not always as flexible as those available on a fully-featured general-purpose compute endpoint with installable packages and tools.

Consider adding real user monitoring (RUM) to your well-known services for real-time session-level awareness.

Become Session and Application-Aware

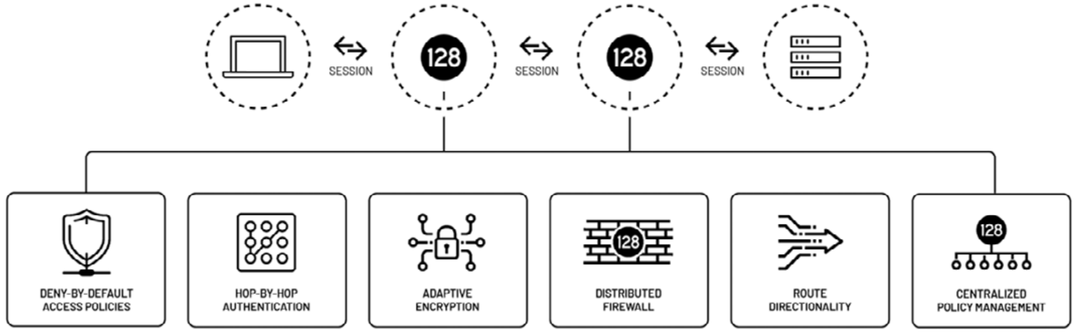
Troubleshooting is predicated on situational awareness. Traditionally, routers have performed their forwarding duties without a real concept of session or application state. Firewalls and load balancers track and use aspects of state, yet session awareness is not pervasive across the whole network. What if a new breed of routers could become session and application-aware to enable smarter routing and policy enforcement?

Juniper's Session Smart™ Routers (SSR) are engineered for application and session awareness.

With continuous optimizations for user experience, support costs and MTTR are dramatically lowered by 30-40%. This session-based networking model provides tunnel-free performance gains and simple scaling.

A "zero-trust" security posture is baked into the fabric, which dramatically lowers risk and increases confidence in service integrity.

Simplified access controls and hyper-segmentation mean SASE (Secure Access Service Edge) is built in from the outset.



Once intelligence can be added to the network's fabric, many benefits ensue, such as lower support costs, increased visibility, and simplified troubleshooting.

Juniper Session Smart™

Juniper's software-based Session Smart™ solution can be rapidly and easily deployed on-premises (via white-box/hypervisor) or to the public cloud.

This session-based model allows for a boundary-free fabric that reduces latency by up to 60%, and reduces bandwidth costs by 30-50%. Flexible deployment options mean you can leverage everything you already have and reap the benefits of a smarter, simpler WAN: happier users, better performance, less troubleshooting, and greater security.

Simplicity

No tunnels, no overlays, no more hardware-centric networking.

Agility

Faster deployment, better responsiveness, dynamic optimization.

Security

Zero-trust model: Authentication + Encryption + Segmentation.

Performance

Less overhead, more scalability, dynamic optimization.

Savings

Reduced bandwidth and connectivity costs.

Discover Session Smart™ SD-WAN

Watch the webinar

Join one of our live demo sessions to see the AI-driven SD-WAN in action.

More questions?
For more support and detailed troubleshooting guides, please visit our **Knowledge Base.**

JUNIPER NETWORKS

Corporate and Sales Headquarters
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or +1.408.745.2000
Fax: +1.408.745.2100
www.juniper.net

APAC and EMEA Headquarters
Juniper Networks International B.V.
Boeing Avenue 240
1119 PZ Schiphol-Rijk
Amsterdam, The Netherlands
Phone: +31.0.207.125.700
Fax: +31.0.207.125.701

PN: 7400132-001-EN