CHECKLIST

Azure Virtual Network Peering Troubleshooting Checklist

A practical checklist for diagnosing Azure virtual network peering issues, validating routing and security settings, and confirming the intended traffic path before production use.

Azure Virtual Network Peering Troubleshooting Checklist

Use this checklist to validate Azure virtual network peering behavior, isolate the cause of connectivity failures, and confirm that traffic flows match your intended design before production rollout.

1) Confirm the exact failure

- [ ] Identify the source host, destination host/service, protocol, and port.
- [ ] Record whether the issue is one-way or bi-directional.
- [ ] Capture the exact symptom: timeout, reset, DNS failure, unreachable host, or intermittent connectivity.
- [ ] Note whether the issue affects ping, TCP, application traffic, or name resolution.

2) Verify the peering foundation

- [ ] Confirm both virtual networks are peered in the expected direction(s).
- [ ] Verify the peering status is active on both sides.
- [ ] Check that the virtual network address spaces do not overlap.

- [ ] Confirm any recent network expansion, merger, or CIDR change did not introduce overlap.
- [ ] Verify the peering was created for the correct environments and subscriptions.

3) Validate intended traffic flow

- [ ] Confirm whether the design expects traffic to stay private or traverse a gateway or appliance.
- [ ] Check whether the topology is hub-and-spoke, mesh, or another pattern.
- [ ] Verify whether the flow depends on transitive routing; do not assume peering is transitive by default.
- [ ] Confirm return traffic has a valid path back to the source.
- [ ] Compare forward path and reverse path behavior.

4) Inspect effective routes

- [ ] Review effective routes on the source NIC.
- [ ] Review effective routes on the destination NIC.
- [ ] Confirm the peering path is not being overridden by a route table.
- [ ] Check for a route to a firewall, network virtual appliance, or other next hop that may intercept traffic.
- [ ] Verify the route chosen matches the intended design.

5) Check network security controls

- [ ] Review NSG rules on the source subnet/NIC.
- [ ] Review NSG rules on the destination subnet/NIC.
- [ ] Confirm allow rules match the actual source IP, destination IP, protocol, and port.
- [ ] Check Azure Firewall, third-party firewall, or appliance policy if present.

- [ ] Verify no deny rule is taking precedence over the intended allow rule.

6) Confirm DNS behavior

- [ ] Validate the name resolves to the expected private IP.
- [ ] Check for stale DNS records or cached entries.
- [ ] Confirm the source is using the intended resolver.
- [ ] Verify custom DNS forwarders are not sending requests to an unreachable or blocked path.
- [ ] Test access by IP address and by hostname to separate routing issues from DNS issues.

7) Review gateway and transit settings

- [ ] If a gateway is in the path, verify gateway transit is intentionally enabled.
- [ ] Confirm both peering objects are configured consistently for gateway usage.
- [ ] Verify the spoke is using the hub gateway only when that is the design.
- [ ] Check for mismatched settings between peering directions.
- [ ] Ensure no assumptions were made about shared gateway access across spokes.

8) Check for asymmetric routing

- [ ] Confirm the request and response take a valid and expected path.
- [ ] Look for one direction succeeding while the reverse fails.
- [ ] Check whether a firewall or appliance is only present on one leg of the flow.
- [ ] Verify the source does not send traffic through peering while the return path goes elsewhere.
- [ ] Investigate timeouts that appear only after the initial handshake.

9) Test the connectivity minimally

- [ ] Test the destination using a basic TCP connection check before troubleshooting the application.
- [ ] Validate the destination responds on the expected port.
- [ ] Try a direct private IP test to isolate DNS from transport.
- [ ] Compare results from a second VM or subnet to determine whether the issue is localized.
- [ ] Repeat the test in the reverse direction if the application requires bidirectional communication.

10) Document what you proved

- [ ] Record the verified source, destination, port, and protocol.
- [ ] Save evidence of effective routes and security rules.
- [ ] Note the DNS record used at the time of testing.
- [ ] Document whether gateway transit, firewall inspection, or UDRs are part of the path.
- [ ] Capture the final confirmed traffic path before making further changes.

Fast triage decision tree

- If the address spaces overlap: fix CIDR design first.
- If peering is connected but traffic fails: inspect routes, NSGs, and firewall policy.
- If hostname fails but IP works: investigate DNS.
- If one direction works and the other does not: check return path and asymmetric routing.
- If hub/spoke access is expected to pass through a gateway: verify gateway transit settings on both peerings.

Pre-production validation checklist

Before approving peering for production use, confirm all of the following:

- [ ] Address spaces are unique and documented.
- [ ] Peering settings match the approved architecture.
- [ ] Effective routes show the intended next hop.
- [ ] NSGs and firewall rules allow only the required traffic.
- [ ] DNS resolves private names to the correct address.
- [ ] Gateway transit is enabled only where required.
- [ ] No unintended transitive access exists.
- [ ] Security and workload owners have signed off on the path.

Operational reminders

- Peering creates a network path, but it does not override security policy.
- A healthy peering connection does not guarantee application reachability.
- Successful troubleshooting requires proving the route, the policy, and the destination.
- Treat peering as a controlled trust relationship, not just a convenience feature.

Quick evidence log

Item	Value
Source host	

Source subnet / VNet |
Destination host / service |
Destination subnet / VNet |
Protocol |
Port |
DNS name tested |
Resolved IP |
Effective route reviewed? |
NSG reviewed? |
Firewall / appliance reviewed? |
Gateway transit involved? |
Final outcome |

Final sign-off

- [ ] Connectivity behavior matches the documented design.
- [ ] Security controls are aligned with the approved access model.
- [ ] Any routing or DNS changes are recorded for future operations.
- [ ] Production dependency on peering has been validated end to end.