3.15 Paketverlust, Latenz, Jitter und Bandbreite analysieren

Netzwerkleistung besteht aus mehreren voneinander unabhängigen Messgrößen. Eine Verbindung kann einen hohen Durchsatz erreichen und trotzdem für Sprache, Remotezugriff oder Echtzeitanwendungen ungeeignet sein. Umgekehrt kann eine Verbindung mit begrenzter Bandbreite sehr stabil und reaktionsschnell arbeiten.

Die zentralen Fragen dieser Seite lauten:

> An welchem Abschnitt des Datenpfads verschlechtert sich die Verbindung, welche Messgröße ist betroffen und tritt das Problem dauerhaft oder nur unter Last auf?

---

1. Sicherheits- und Aktionskennzeichnungen

Kennzeichnung	Bedeutung
[RO]	Rein lesender Befehl, der normalerweise keine Konfiguration verändert
[TEST]	Aktiver Test, der Netzwerkverkehr erzeugt
[PRIV]	Administrator- oder Root-Rechte können erforderlich sein
[FILE]	Der Befehl schreibt Ausgaben in eine Datei
[SENS]	Die Ausgabe kann sensible Informationen enthalten
[CHANGE]	Der Befehl verändert eine Konfiguration
[DISRUPT]	Der Befehl kann eine Verbindung oder einen Dienst beeinträchtigen

Bandbreiten- und Lasttests dürfen nur gegen ausdrücklich freigegebene Systeme durchgeführt werden. Sie können Leitungen, Firewalls, Server und andere Benutzer erheblich belasten.

---

2. Die wichtigsten Leistungsbegriffe unterscheiden

Begriff	Bedeutung

Latenz	Zeit, die eine Übertragung von einem Punkt zu einem anderen benötigt
Round-Trip Time – RTT	Zeit für Hin- und Rückweg
Jitter	Schwankung der Paketlaufzeit
Paketverlust	Anteil gesendeter Pakete, die das Ziel beziehungsweise die Messstelle nicht erreichen
Bandbreite	Theoretische oder konfigurierte Übertragungskapazität
Durchsatz	Tatsächlich übertragene Datenmenge pro Zeit
Goodput	Für die Anwendung nutzbare Datenmenge ohne Protokoll- und Wiederholungsdaten
Retransmission	Erneute Übertragung verlorener oder nicht bestätigter TCP-Daten
Congestion	Überlastung eines Netzwerkpfads oder einer Warteschlange
Bufferbloat	Stark steigende Latenz durch übermäßig gefüllte Warteschlangen
Burst Loss	Mehrere Paketverluste in kurzer Folge
Out-of-Order	Pakete treffen in anderer Reihenfolge ein
Duplication	Pakete treffen mehrfach ein
Queueing Delay	Verzögerung durch Warteschlangen
Serialization Delay	Zeit, um ein Paket auf einen Link zu übertragen
Processing Delay	Verarbeitungszeit in einem Netzwerkgerät
Propagation Delay	Physikalische Signallaufzeit über das Medium

3. Bit, Byte und Übertragungsraten richtig umrechnen

1 Byte = 8 Bit

Einheit	Bedeutung
1 kbit/s	üblicherweise 1.000 Bit pro Sekunde
1 Mbit/s	üblicherweise 1.000.000 Bit pro Sekunde
1 Gbit/s	üblicherweise 1.000.000.000 Bit pro Sekunde
1 MB/s	8 Mbit/s

Einheit	Bedeutung
`100 MB/s`	800 Mbit/s
`125 MB/s`	1.000 Mbit/s beziehungsweise 1 Gbit/s

Dateigrößen können binär oder dezimal angegeben werden. Zusätzlich reduzieren Protokoll-Overhead, Verschlüsselung, Wiederholungen und Dateisystemverarbeitung den nutzbaren Durchsatz.

Beispiel:

```
1-Gbit/s-Ethernet
≠
125 MB/s garantierte Nutzdatenrate
```

---

4. Messziel und Datenpfad vollständig beschreiben

Vor der Messung ist der genaue Pfad zu dokumentieren:

```
Client
  ↓
WLAN oder Ethernet
  ↓
Access Point oder Switch
  ↓
Router beziehungsweise Firewall
  ↓
VPN, WAN oder Internet
  ↓
Zielnetz
  ↓
Server
  ↓
Anwendung und Datenträger
```

Benötigte Angaben:

Quelle:

Quell-IP:

Quellschnittstelle:

Ziel:

Ziel-IP:

Transportprotokoll:

Zielport:

IPv4 oder IPv6:

VPN beteiligt:

Proxy beteiligt:

NAT beteiligt:

Testzeitpunkt:

Gegenrichtung:

Erwartete Leistung:

Beobachtete Leistung:

Ohne definierten Messpfad ist die Aussage „Das Netzwerk ist langsam“ nicht technisch verwertbar.

---

5. Die Verbindung abschnittsweise messen

Messabschnitt	Typisches Ziel
Client zur lokalen Schnittstelle	Adapter und Betriebssystem
Client zum Standardgateway	lokales LAN beziehungsweise WLAN
Client zu lokalem Server	internes Netzwerk
Client zu anderem VLAN	Routing und interne Firewall
Client zum VPN-Gateway	Internet- oder WAN-Basisverbindung
Client durch VPN zum internen Server	Tunnel, VPN-Routing und Zielnetz
Client zum Internetziel	Provider- und Internetpfad
Reverse Proxy zum Backend	Servernetz und Backend
Server zum Datenspeicher	Speicher- oder Anwendungsengpass

Grundregel:

> Mit der nächstgelegenen sinnvollen Messstelle beginnen und den Pfad schrittweise erweitern.

---

6. Einzelmessung und Langzeitmessung unterscheiden

Eine einzelne Messung kann zufällig in einem guten oder schlechten Moment stattfinden.

Für eine belastbare Diagnose dokumentieren:

- Anzahl der Messwerte,
- Messdauer,
- Messintervall,
- Minimum,
- Maximum,
- Durchschnitt,
- Median,
- relevante Perzentile,
- Paketverlust,
- Zeitpunkt auffälliger Spitzen,
- gleichzeitige Netzlast,
- Benutzerzahl,
- Testmethode.

Durchschnittswerte können kurze Ausfälle verdecken:

```
99 Messungen: 10 ms
1 Messung:    2.000 ms

Durchschnitt:
29,9 ms
```

Der Durchschnitt wirkt noch relativ niedrig, obwohl eine einzelne Verzögerung von zwei Sekunden für Echtzeitanwendungen deutlich spürbar sein kann.

---

7. ICMP-Ping sinnvoll einsetzen

Ping misst die Round-Trip Time von ICMP Echo Request und Echo Reply.

Windows:

```
[TEST] ping -n 20 ZIEL_IP
```

Windows PowerShell:

```
[TEST] Test-Connection ZIEL_IP -Count 20
```

Linux:

```
[TEST] ping -c 20 ZIEL_IP
```

macOS:

```
[TEST] ping -c 20 ZIEL_IP
```

Zu dokumentieren sind:

- gesendete Pakete,
- empfangene Pakete,
- Verlust in Prozent,
- minimale RTT,
- durchschnittliche RTT,
- maximale RTT,
- starke Schwankungen.

Wichtig: Ein Ziel kann ICMP blockieren oder niedriger priorisieren, während Anwendungsverkehr weiterhin funktioniert.

---

8. Ping-Ergebnisse korrekt interpretieren

Beobachtung	Mögliche Bedeutung
Gleichmäßig niedrige RTT	Pfad ist für diese Messung stabil
Dauerhaft hohe RTT	Entfernung, langsamer Link, Tunnel oder Überlastung
Einzelne starke Spitzen	Queueing, Funkstörung, Last oder Verarbeitungsspitze
Regelmäßige Spitzen	periodischer Datenverkehr, Scan, Backup oder Energiesparfunktion

Beobachtung	Mögliche Bedeutung
Vereinzelter Verlust	Störung, Überlastung oder ICMP-Priorisierung
Verlust in Bursts	Link-Flap, Funkproblem, Queue Drop oder Pfadwechsel
Kein Ping, Anwendung funktioniert	ICMP wird möglicherweise gefiltert
Ping zur IP funktioniert, zum Namen nicht	Namensauflösung prüfen
Gatewayping schlecht	Problem wahrscheinlich bereits im lokalen Zugangsnetz
Gatewayping gut, externes Ziel schlecht	Problem liegt wahrscheinlich hinter dem Gateway

Ein Ping beweist nicht die Leistungsfähigkeit von TCP, UDP oder der Anwendung.

---

9. Lokales Gateway zuerst messen

Standardgateway ermitteln:

Windows:

```
[RO] Get-NetRoute -DestinationPrefix "0.0.0.0/0" |
    Sort-Object RouteMetric |
    Format-Table InterfaceAlias, NextHop, RouteMetric
```

Linux:

```
[RO] ip route show default
```

macOS:

```
[RO] route -n get default
```

Gateway über längeren Zeitraum testen:

Windows:

```
[TEST] ping -n 100 GATEWAY_IP
```

Linux und macOS:

```
[TEST] ping -c 100 GATEWAY_IP
```

Wenn bereits zum lokalen Gateway Paketverlust oder starke Latenzschwankungen auftreten, sollten zuerst folgende Komponenten geprüft werden:

- WLAN-Funkverbindung,
- Ethernetkabel,
- Switchport,
- Netzwerkadapter,
- Treiber,
- Duplex und Linkgeschwindigkeit,
- lokaler Router,
- Clientauslastung.

---

10. Mehrere Messziele parallel vergleichen

Für die Eingrenzung werden mehrere Ziele betrachtet:

```
Ziel 1: lokales Gateway
Ziel 2: interner Server
Ziel 3: WAN-Gegenstelle
Ziel 4: Anwendungsserver
```

Interpretationsbeispiel:

Gateway	Interner Server	Externes Ziel	Mögliche Eingrenzung
gut	gut	schlecht	WAN, Provider oder externer Pfad
gut	schlecht	gut	internes Routing, VLAN oder Serverpfad
schlecht	schlecht	schlecht	lokales Zugangsnetz
gut	gut	gut	möglicherweise anwendungsspezifisches Problem
gut	nur VPN-Ziel schlecht	gut	VPN oder entferntes Netz

Parallele Messungen müssen mit sinnvollen Intervallen erfolgen, damit die Diagnose nicht selbst unnötige Last erzeugt.

---

11. TCP-Erreichbarkeit und TCP-Latenz prüfen

Wenn ICMP blockiert wird, kann ein Test zum tatsächlichen TCP-Port sinnvoller sein.

Windows:

```
[TEST] Test-NetConnection server.example.internal `
    -Port 443 `
    -InformationLevel Detailed
```

Linux und macOS:

```
[TEST] nc -vz -w 5 server.example.internal 443
```

Diese Befehle zeigen hauptsächlich, ob ein TCP-Verbindungsaufbau möglich ist. Sie liefern noch keine vollständige Leistungsmessung.

Windows mit PsPing, sofern autorisiert installiert:

```
[TEST] psping -n 20 server.example.internal:443
```

PsPing kann TCP-Verbindungszeiten messen und ist besonders nützlich, wenn ICMP nicht beantwortet wird.

---

12. Anwendungsantwortzeiten mit curl zerlegen

Zeitanteile einer HTTP- beziehungsweise HTTPS-Anfrage:

```
[TEST][SENS] curl -sS -o /dev/null \
    -w 'DNS: %{time_namelookup}\nConnect: %{time_connect}\nTLS: %{time_appconnect}\nStartTransfer: %{time_starttransfer}\nTotal: %{time_total}\nHTTP: %{http_code}\n' \
    https://service.example.net/
```

curl-Wert	Bedeutung
`time_namelookup`	Zeit bis zum Abschluss der Namensauflösung
`time_connect`	Zeit bis zum TCP-Verbindungsaufbau

curl-Wert	Bedeutung
`time_appconnect`	Zeit bis zum Abschluss des TLS-Handshakes
`time_starttransfer`	Zeit bis zum ersten Antwortbyte
`time_total`	Gesamtdauer
`http_code`	HTTP-Statuscode

Typische Eingrenzung:

Auffälliger Wert	Mögliche Ursache
DNS-Zeit hoch	DNS-Server, Suche oder Netzwerkpfad
Connect-Zeit hoch	Latenz, Paketverlust, Firewall oder Serverlistener
TLS-Zeit hoch	TLS-Handshake, Zertifikatsprüfung oder CPU
StartTransfer hoch	Server, Datenbank oder Backend
Total hoch, StartTransfer normal	große Antwort, begrenzte Bandbreite oder langsamer Client

---

13. Netzwerkpfad mit traceroute und tracert prüfen

Windows ohne Namensauflösung:

```
[TEST] tracert -d ZIEL_IP
```

Linux:

```
[TEST] traceroute -n ZIEL_IP
```

Falls `traceroute` nicht installiert ist:

```
[TEST] tracepath -n ZIEL_IP
```

macOS:

```
[TEST] traceroute -n ZIEL_IP
```

Zu kontrollieren sind:

- Anzahl der Hops,
- Änderung des Pfads,
- sprunghafter Latenzanstieg,
- Zeitüberschreitungen,
- unterschiedliche IPv4- und IPv6-Pfade,
- asymmetrische oder lastverteilte Pfade.

Ein Sternchen bei einem Zwischenrouter beweist keinen Paketverlust des weitergeleiteten Nutzverkehrs. Der Router kann Antworten auf Diagnosepakete begrenzen oder vollständig unterdrücken.

---

14. IPv4 und IPv6 getrennt messen

Ein Hostname kann sowohl IPv4- als auch IPv6-Adressen besitzen. Die Pfade und Leistungswerte können deutlich voneinander abweichen.

Windows:

```
[TEST] ping -4 -n 20 HOSTNAME
[TEST] ping -6 -n 20 HOSTNAME
```

```
[TEST] tracert -4 -d HOSTNAME
[TEST] tracert -6 -d HOSTNAME
```

Linux:

```
[TEST] ping -4 -c 20 HOSTNAME
[TEST] ping -6 -c 20 HOSTNAME
```

macOS:

```
[TEST] ping -c 20 IPV4_ADRESSE
[TEST] ping6 -c 20 IPV6_ADRESSE
```

HTTP getrennt testen:

```
[TEST] curl -4 -I https://service.example.net/
[TEST] curl -6 -I https://service.example.net/
```

Ein gutes IPv4-Ergebnis beweist nicht, dass der IPv6-Pfad ebenfalls funktioniert.

---

15. Windows pathping verwenden

`pathping` kombiniert Pfadermittlung mit wiederholten Messungen zu Zwischenstationen.

Ohne Namensauflösung:

```
[TEST] pathping /n ZIEL_IP
```

IPv4:

```
[TEST] pathping /n /4 ZIEL_IP
```

IPv6:

```
[TEST] pathping /n /6 ZIEL_IP
```

Die Messung kann mehrere Minuten dauern.

Wichtige Interpretation:

Wenn ein Zwischenhop hohen Verlust zeigt, spätere Hops und das Ziel aber keinen entsprechenden Verlust zeigen, priorisiert oder begrenzt der Zwischenrouter wahrscheinlich nur seine eigenen ICMP-Antworten.

Ein echter Weiterleitungsverlust wird typischerweise ab einer Stelle auch an nachfolgenden Hops beziehungsweise am Ziel sichtbar.

---

16. MTR unter Linux und macOS verwenden

`mtr` kombiniert fortlaufende Pfad- und Latenzmessungen. Es ist nicht auf jedem System standardmäßig installiert.

Prüfen, ob MTR vorhanden ist:

```
[RO] command -v mtr
```

Interaktive Messung ohne Namensauflösung:

```
[TEST] mtr -n ZIEL_IP
```

Berichtsmodus mit 100 Messzyklen:

```
[TEST] mtr -n -r -c 100 ZIEL_IP
```

TCP-basierter Test zu Port 443, sofern von der installierten Version unterstützt:

```
[TEST][PRIV] sudo mtr -n -r -c 100 -T -P 443 ZIEL_IP
```

Vor Verwendung erweiterter Optionen lokale Hilfe prüfen:

```
[RO] mtr --help
```

Auch bei MTR gilt: Verlust an einem Zwischenhop allein ist kein Beweis für weitergeleiteten Paketverlust.

---

17. Paketverlust mathematisch einordnen

```
Paketverlust in Prozent =
(verlorene Pakete ÷ gesendete Pakete) × 100
```

Beispiel:

```
Gesendet: 1.000
Empfangen: 990
Verloren: 10
```

```
Paketverlust = 1 %
```

Die Auswirkungen hängen von Anwendung und Verteilung des Verlusts ab.

Verlustmuster	Mögliche Wirkung
Einzelne gleichmäßig verteilte Verluste	TCP-Wiederholungen, kurze Audiofehler
Mehrere Verluste in Folge	deutliche Unterbrechung oder Timeout
Verlust nur unter Last	überfüllte Queue oder überlasteter Link
Verlust nur in eine Richtung	asymmetrisches Problem
Verlust nur großer Pakete	MTU- oder Fragmentierungsproblem
ICMP-Verlust ohne Anwendungsverlust	mögliche ICMP-Priorisierung

Burst Loss ist für Echtzeitanwendungen häufig problematischer als derselbe prozentuale Verlust gleichmäßig verteilt.

---

18. Jitter messen und beurteilen

Jitter beschreibt die Schwankung von Paketlaufzeiten. Es gibt unterschiedliche Berechnungsmethoden; Werte verschiedener Werkzeuge sind daher nicht immer direkt vergleichbar.

Typische Symptome hohen Jitters:

- abgehackte Sprache,
- schwankende Videoqualität,
- Ruckeln bei Remote-Desktop,
- ungleichmäßige Echtzeitdaten,
- vergrößerte Wiedergabepuffer,
- kurze Aussetzer trotz geringem Durchschnittsping.

`ping` zeigt Schwankungen der RTT, ist aber kein vollständiger Ersatz für eine anwendungsnahe Jittermessung.

Für UDP kann `iperf3` Jitter und Datagrammverlust ausgeben:

```
[TEST] iperf3 -c SERVER_IP -u -b 10M -t 30
```

Mit niedriger, kontrollierter Zielrate beginnen. Eine zu hohe Testbandbreite erzeugt selbst Verlust und Jitter.

19. iperf3-Testumgebung vorbereiten

Für eine kontrollierte Messung werden benötigt:

```
iperf3-Client
iperf3-Server
freigegebener Testpfad
bekannte Schnittstellen
ausreichende Serverleistung
definierte Testdauer
```

Server starten:

```
[TEST] iperf3 -s
```

Standardmäßig verwendet iperf3 Port `5201`. Der Port kann verändert werden.

Clienttest:

```
[TEST] iperf3 -c SERVER_IP
```

Anderer Port:

Server:

```
[TEST] iperf3 -s -p 5202
```

Client:

```
[TEST] iperf3 -c SERVER_IP -p 5202
```

Firewallregeln müssen zum gewählten Transport und Port passen.

Bei UDP-Tests verwendet iperf3 zusätzlich eine TCP-Steuerverbindung. Die genaue Port- und Firewallkonfiguration ist deshalb vorab zu prüfen.

20. TCP-Durchsatz mit iperf3 messen

Standardtest vom Client zum Server:

```
[TEST] iperf3 -c SERVER_IP
```

30 Sekunden testen:

```
[TEST] iperf3 -c SERVER_IP -t 30
```

Gegenrichtung – Server sendet zum Client:

```
[TEST] iperf3 -c SERVER_IP -R -t 30
```

Beide Richtungen gleichzeitig, sofern unterstützt:

```
[TEST] iperf3 -c SERVER_IP --bidir -t 30
```

Mehrere parallele TCP-Datenströme:

```
[TEST] iperf3 -c SERVER_IP -P 4 -t 30
```

Parallele Datenströme können einen Pfad stärker auslasten und höhere Messwerte erzielen. Sie können aber Probleme eines einzelnen Datenstroms verdecken und erhebliche Last erzeugen.

Empfohlene Reihenfolge:

1. Ein TCP-Datenstrom zum Server.
2. Ein TCP-Datenstrom in Gegenrichtung.
3. Bei Bedarf mehrere Datenströme.
4. Erst danach gleichzeitiger bidirektionaler Test.

---

21. UDP mit iperf3 kontrolliert messen

UDP-Test mit 10 Mbit/s Zielrate:

```
[TEST] iperf3 -c SERVER_IP -u -b 10M -t 30
```

Gegenrichtung:

```
[TEST] iperf3 -c SERVER_IP -u -b 10M -t 30 -R
```

Relevante Ergebnisse:

- gesendete Bitrate,
- empfangene Bitrate,
- Jitter,
- verlorene Datagramme,
- Verlustquote,
- Out-of-Order-Pakete, falls ausgegeben.

Teststrategie:

```
niedrige Rate
→ Ergebnis prüfen
→ Rate schrittweise erhöhen
→ Beginn von Verlust und Jitter dokumentieren
```

Nicht sofort mit der maximal erwarteten Linkgeschwindigkeit beginnen. Ein UDP-Test besitzt keine TCP-Staukontrolle und kann andere Verbindungen stark beeinträchtigen.

---

22. iperf3-Ergebnisse als JSON speichern

JSON-Ausgabe anzeigen:

```
[TEST][SENS] iperf3 -c SERVER_IP -J
```

In Datei speichern:

```
[TEST][FILE][SENS] iperf3 -c SERVER_IP -t 30 -J > iperf3-test.json
```

UDP-Ergebnis speichern:

```
[TEST][FILE][SENS] iperf3 -c SERVER_IP \
    -u -b 10M -t 30 -J > iperf3-udp-test.json
```

Vor einer Weitergabe prüfen, ob die Datei interne IP-Adressen, Hostnamen oder andere sensible Angaben enthält.

---

23. iperf3-Ergebnisse richtig interpretieren

Beobachtung	Mögliche Bedeutung
Hinrichtung gut, Rückrichtung schlecht	asymmetrischer Pfad, QoS, Empfangs- oder Sendeseite
Ein Stream schlecht, mehrere gut	RTT, TCP-Fenster, Verlust oder CPU pro Stream
TCP gut, UDP bei niedriger Rate schlecht	Queue, Policing, Funk- oder UDP-spezifischer Pfad
UDP erst ab bestimmter Rate mit Verlust	Kapazitäts- beziehungsweise Queuegrenze
Werte schwanken stark	Funk, Last, Pfadwechsel oder konkurrierender Verkehr
Server- und Clientwerte unterscheiden sich	Verlust beziehungsweise unvollständige Übertragung
Hohe Retransmissions	Paketverlust, Überlastung oder fehlerhafter Link
Durchsatz exakt an einer Rate begrenzt	Policer, Shaper, Vertrag oder Schnittstellenlimit
CPU nahe 100 Prozent	Endsystem kann der Engpass sein
Datenträger langsam, iperf3 gut	Speicher oder Anwendung statt Netzwerk

Ein iperf3-Test misst Arbeitsspeicher-zu-Arbeitsspeicher-Netzwerkleistung und nicht automatisch Datei-, Datenbank- oder Anwendungsleistung.

---

24. TCP-Wiederholungen und Zustände prüfen

Windows-TCP-Statistik:

```
[RO] netstat -s -p tcp
```

Windows-Verbindungen:

```
[RO] Get-NetTCPConnection |
    Format-Table LocalAddress, LocalPort, RemoteAddress,
                 RemotePort, State
```

Linux-TCP-Statistik:

```
[RO] nstat
```

Falls `nstat` nicht vorhanden ist:

```
[RO] netstat -s
```

Linux-Socketdiagnose:

```
[RO][SENS] ss -ti
```

macOS-TCP-Statistik:

```
[RO] netstat -s -p tcp
```

Zähler sind kumulativ und müssen über einen definierten Messzeitraum verglichen werden. Ein hoher Gesamtwert allein beweist kein aktuelles Problem.

---

25. Schnittstellenzähler und physische Fehler prüfen

Windows:

```
[RO] Get-NetAdapterStatistics |
    Format-Table Name, ReceivedBytes, SentBytes,
                 ReceivedDiscardedPackets, OutboundDiscardedPackets,
                 ReceivedPacketErrors, OutboundPacketErrors
```

Linux:

```
[RO] ip -s link
```

Linux – erweiterte Treiberstatistiken:

```
[RO][PRIV][SENS] sudo ethtool -S INTERFACE
```

macOS:

```
[RO] netstat -ib
```

```
[RO] ifconfig
```

Zu prüfen sind:

- RX- und TX-Fehler,
- Drops,
- verworfene Pakete,
- CRC- beziehungsweise Framefehler,
- Overruns,
- Carrierfehler,
- Link-Flaps,
- steigende Fehlerzähler unter Last.

Entscheidend ist, ob Zähler während des reproduzierten Fehlers ansteigen.

---

26. Linkgeschwindigkeit und Duplex prüfen

Windows:

```
[RO] Get-NetAdapter |
    Format-Table Name, Status, LinkSpeed, MediaConnectionState
```

Linux:

```
[RO] ip link show INTERFACE
```

```
[RO][PRIV] sudo ethtool INTERFACE
```

macOS:

```
[RO] ifconfig INTERFACE
```

Abhängig vom Adapter können zusätzliche Informationen über `networksetup` oder `system_profiler` verfügbar sein.

Zu prüfen sind:

- erwartete Linkgeschwindigkeit,
- Full Duplex oder Half Duplex,
- Autonegotiation,
- Linkstatus,
- unerwartete Begrenzung auf 100 Mbit/s,
- wiederholte Neuverhandlung,
- Kabel- oder Portfehler.

Ein Link mit 1 Gbit/s Anzeige beweist nicht, dass er fehlerfrei oder vollständig auslastbar ist.

---

27. Bufferbloat erkennen

Bufferbloat liegt vor, wenn Warteschlangen unter Last stark anwachsen und dadurch die Latenz deutlich steigt.

Typisches Verhalten:

```
Ping ohne Last:  10 ms
Ping unter Last: 300 ms
Paketverlust:    möglicherweise gering
Durchsatz:       hoch
```

Trotz hohen Durchsatzes reagieren Sprache, Spiele, Remote-Desktop und interaktive Anwendungen schlecht.

Testablauf:

1. Kontinuierlichen Ping zum geeigneten Ziel starten.

2. Basislatenz ohne Last dokumentieren.
3. Kontrollierten iperf3-Test starten.
4. Latenz während Downloadrichtung beobachten.
5. Latenz während Uploadrichtung beobachten.
6. Test stoppen.
7. Erholung der Latenz dokumentieren.

Beispiel:

Terminal 1:

```
[TEST] ping ZIEL_IP
```

Terminal 2:

```
[TEST] iperf3 -c SERVER_IP -t 30
```

Anschließend:

```
[TEST] iperf3 -c SERVER_IP -R -t 30
```

Eine Optimierung kann Queue Management oder Traffic Shaping erfordern. Solche Änderungen müssen am tatsächlichen Engpass und nicht willkürlich am Client erfolgen.

---

28. QoS, Shaping und Policing berücksichtigen

Funktion	Wirkung
Classification	Verkehr wird einer Klasse zugeordnet
Marking	Pakete erhalten beispielsweise eine DSCP-Markierung
Queuing	Pakete werden in Warteschlangen eingeordnet
Scheduling	Reihenfolge der Übertragung wird bestimmt
Shaping	Datenrate wird durch Puffern geglättet
Policing	Überschreitender Verkehr wird verworfen oder neu markiert
Rate Limit	Datenrate wird begrenzt
Priorisierung	Bestimmte Klassen werden bevorzugt behandelt

Typische Symptome einer Richtlinie:

- exakt begrenzter Durchsatz,
- nur bestimmte Anwendungen betroffen,
- UDP und TCP verhalten sich unterschiedlich,
- nur eine Richtung ist begrenzt,
- Unterschied zwischen internen und externen Zielen,
- Verhalten ändert sich zu bestimmten Uhrzeiten,
- DSCP-Markierungen werden verändert oder entfernt.

QoS ist kein Ersatz für ausreichende Kapazität. Eine Priorisierung kann wichtige Anwendungen schützen, erzeugt aber keine zusätzliche Bandbreite.

---

29. Server und Client als Engpass ausschließen

Während des Netzwerktests sind auf beiden Endsystemen zu prüfen:

- CPU-Auslastung,
- Arbeitsspeicher,
- Interruptlast,
- Netzwerktreiber,
- Energiesparzustand,
- Verschlüsselungsleistung,
- virtuelle CPU,
- Hypervisor-Auslastung,
- Containerlimits,
- Datenträger, falls Anwendungstest,
- Sicherheitssoftware.

Windows:

```
[RO] Get-Counter '\Processor(_Total)\% Processor Time'
```

```
[RO] Get-Process |
    Sort-Object CPU -Descending |
    Select-Object -First 10
```

Linux:

```
[RO] uptime
```

```
[RO] top
```

macOS:

```
[RO] top -l 1
```

Ein Durchsatztest kann durch die Rechenleistung eines virtuellen oder kleinen Systems begrenzt werden, obwohl das Netzwerk höhere Kapazität besitzt.

---

30. Dateiübertragung nicht als alleinigen Netzwerktest verwenden

Eine Dateiübertragung misst gleichzeitig:

```
Quelldatenträger
+ Dateisystem
+ Serverdienst
+ Protokoll
+ Netzwerk
+ Sicherheitsprüfung
+ Zieldatenträger
```

Mögliche nicht netzwerkbezogene Engpässe:

- langsame HDD,
- ausgelastetes RAID,
- Virenscanner,
- Dateiverschlüsselung,
- Deduplizierung,
- Kompression,
- SMB-Signierung oder Verschlüsselung,
- kleine Dateien und Metadatenoperationen,
- Datenbankzugriffe,
- CPU-Auslastung,
- Cachezustand.

Empfohlene Trennung:

```
iperf3 gut
Dateikopie langsam
→ Speicher, Dateidienst oder Anwendung weiter prüfen
```

---

31. MTU und große Pakete prüfen

Windows mit Don't-Fragment-Bit:

```
[TEST] ping -4 -f -l 1400 ZIEL_IP
```

Linux:

```
[TEST] ping -4 -M do -s 1400 ZIEL_IP
```

macOS:

```
[TEST] ping -D -s 1400 ZIEL_IP
```

MTU anzeigen:

Betriebssystem	Befehl
Windows	[RO] Get-NetIPInterface
Linux	[RO] ip link show
macOS	[RO] ifconfig

Die Ping-Nutzdatenlänge ist nicht mit der gesamten IP-Paketgröße identisch. Header müssen zusätzlich berücksichtigt werden.

Typische MTU-Symptome:

- kleine Anfragen funktionieren,
- große Übertragungen hängen,
- VPN-Verbindungen betroffen,
- TLS-Handshake unvollständig,
- bestimmte Websites laden nur teilweise,

- IPv6 meldet Packet Too Big.

---

32. Paketmitschnitt zur Leistungsanalyse verwenden

Linux:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i INTERFACE -nn \
    'host ZIEL_IP' \
    -w performance-test.pcap
```

macOS:

```
[TEST][PRIV][FILE][SENS] sudo tcpdump -i INTERFACE -nn \
    'host ZIEL_IP' \
    -w performance-test.pcap
```

Windows mit Dumpcap – Schnittstellen ermitteln:

```
[RO] dumpcap -D
```

Mitschnitt:

```
[TEST][PRIV][FILE][SENS] dumpcap -i 1 `
    -f "host ZIEL_IP" `
    -w performance-test.pcapng
```

Paketmitschnitte können vertrauliche Nutzdaten und Metadaten enthalten. Aufzeichnungsdauer und Filter sollten auf den benötigten Umfang begrenzt werden.

---

33. Wireshark-Filter für Leistungsprobleme

Aufgabe	Anzeigefilter
Verkehr zu einer IPv4-Adresse	`ip.addr == 192.0.2.20`
Verkehr zu einer IPv6-Adresse	`ipv6.addr == 2001:db8::20`

Aufgabe	Anzeigefilter
TCP-Wiederholungen	`tcp.analysis.retransmission`
Schnelle Wiederholungen	`tcp.analysis.fast_retransmission`
Vermutete verlorene Segmente	`tcp.analysis.lost_segment`
Doppelte ACKs	`tcp.analysis.duplicate_ack`
Out-of-Order	`tcp.analysis.out_of_order`
Zero Window	`tcp.analysis.zero_window`
Window Full	`tcp.analysis.window_full`
TCP-Reset	`tcp.flags.reset == 1`
ICMP	`icmp`
ICMPv6	`icmpv6`
DNS-Antwortzeit	`dns.time`
TCP-Verbindungsaufbau	`tcp.connection.syn

Wireshark-Analyseflags sind Schlussfolgerungen aus dem vorhandenen Mitschnitt. Fehlende Pakete am Messpunkt können falsche oder irreführende Markierungen erzeugen.

---

34. Einseitige und doppelseitige Mitschnitte unterscheiden

Einseitiger Mitschnitt:

```
zeigt nur Pakete am lokalen Messpunkt
```

Doppelseitiger Mitschnitt:

```
Clientmitschnitt
+
Servermitschnitt
+
synchronisierte Zeit
```

Dadurch lässt sich erkennen:

- ob ein Paket den Client verlässt,
- wann es den Server erreicht,

- ob der Server antwortet,
- wann die Antwort den Client erreicht,
- in welcher Richtung Verlust auftritt,
- wo zusätzliche Verzögerung entsteht,
- ob ein Netzwerkgerät Pakete verändert.

Für exakte Laufzeitvergleiche müssen die Uhren der Messsysteme ausreichend synchronisiert sein.

---

35. Zwischenhop-Verlust nicht falsch interpretieren

Beispiel:

```
Hop 1:  0 % Verlust
Hop 2: 60 % Verlust
Hop 3:  0 % Verlust
Ziel:   0 % Verlust
```

Hop 2 leitet den Verkehr offensichtlich weiter, antwortet aber nur eingeschränkt auf Diagnosepakete.

Wahrscheinlicher tatsächlicher Weiterleitungsverlust:

```
Hop 1:  0 % Verlust
Hop 2: 20 % Verlust
Hop 3: 20 % Verlust
Hop 4: 20 % Verlust
Ziel:  20 % Verlust
```

Auch dieses Muster muss mit weiteren Messungen bestätigt werden, da Pfadwechsel, Load Balancing und asymmetrische Routen Ergebnisse beeinflussen können.

---

36. Typische Fehlerbilder systematisch eingrenzen

Fall A – Latenz bereits zum Gateway hoch

Prüfen:

- WLAN-Signal und Kanal,
- Kabel,
- Switchport,
- Duplex,
- Adaptertreiber,
- lokalen Router,
- Clientlast.

Fall B – Gateway gut, Internet schlecht

Prüfen:

- WAN-Auslastung,
- Provider,
- Routerqueue,
- VPN,
- Proxy,
- externen Pfad,
- IPv4 gegenüber IPv6.

Fall C – Durchsatz gut, Echtzeitanwendung schlecht

Prüfen:

- Jitter,
- Burst Loss,
- Bufferbloat,
- QoS,
- UDP-Pfad,
- Anwendungsserver.

Fall D – Download gut, Upload schlecht

Prüfen:

- Upstreamkapazität,
- Shaping,
- Policing,
- WLAN-Senderichtung,
- asymmetrischen Pfad,
- fehlerhafte TX-Zähler.

Fall E – Kleine Übertragungen funktionieren, große nicht

Prüfen:

- MTU,
- Path-MTU-Discovery,
- ICMP-Blockierung,
- Fragmentierung,
- VPN-Overhead.

Fall F – iperf3 gut, Dateikopie langsam

Prüfen:

- Datenträger,
- Dateisystem,
- SMB oder NFS,
- Virenscanner,
- Verschlüsselung,
- Serverlast.

Fall G – Nur zu Stoßzeiten schlecht

Prüfen:

- Linkauslastung,
- Airtime,
- Backup,
- Synchronisation,
- WAN-Kapazität,
- Queue Drops,
- Serverlast.

Fall H – Nur ein TCP-Datenstrom langsam

Prüfen:

- RTT,
- Paketverlust,
- TCP-Fenster,
- Empfangspuffer,
- Congestion Control,
- CPU eines einzelnen Prozesses.

37. Keine unkontrollierten Lasttests durchführen

Nicht vorschnell:

- maximale UDP-Rate senden,
- viele parallele Datenströme starten,
- Tests während kritischer Geschäftsprozesse durchführen,
- produktive Server ohne Freigabe als iperf3-Ziel verwenden,
- QoS oder Shaping deaktivieren,
- Interfacezähler zurücksetzen,
- MTU beliebig ändern,
- Firewalls für Tests abschalten,
- öffentliche Speedtest-Dienste mit sensiblen Systemen automatisiert belasten,
- Messergebnisse verschiedener Pfade direkt vergleichen.

Sicherer Testablauf:

1. Testumfang genehmigen lassen.
2. Quelle, Ziel und Pfad dokumentieren.
3. Niedrige Last verwenden.
4. Messdauer begrenzen.
5. Überwachung parallel aktivieren.
6. Last schrittweise erhöhen.
7. Test bei negativen Auswirkungen sofort beenden.
8. Hin- und Rückrichtung getrennt messen.
9. Ergebnis samt Uhrzeit dokumentieren.
10. Testdienste anschließend kontrolliert beenden.

38. Systematischer Diagnoseablauf

Schritt	Prüfung	Leitfrage
1	Problem konkretisieren	Latenz, Jitter, Verlust oder Durchsatz?
2	Quelle und Ziel dokumentieren	Welcher Datenpfad ist betroffen?
3	Schnittstellenstatus prüfen	Stimmen Link, Geschwindigkeit und Fehlerzähler?
4	Lokales Gateway messen	Beginnt das Problem bereits im Zugangsnetz?
5	Internen Server messen	Ist das lokale LAN betroffen?

Schritt	Prüfung	Leitfrage
6	Externen Pfad messen	Beginnt das Problem hinter dem Router?
7	IPv4 und IPv6 vergleichen	Ist nur eine Adressfamilie betroffen?
8	Pfad ermitteln	Welche Hops werden tatsächlich verwendet?
9	Hin- und Rückrichtung messen	Ist das Problem asymmetrisch?
10	TCP-Durchsatz messen	Welche Nutzrate erreicht ein Datenstrom?
11	UDP kontrolliert messen	Wie entwickeln sich Jitter und Verlust?
12	Latenz unter Last prüfen	Liegt Bufferbloat vor?
13	Schnittstellenzähler vergleichen	Steigen Fehler oder Drops während des Tests?
14	Endsystemlast prüfen	Begrenzen CPU oder Anwendung das Ergebnis?
15	MTU prüfen	Scheitern nur größere Pakete?
16	Paketmitschnitt erstellen	Sind Wiederholungen oder Zeitlücken sichtbar?
17	Vergleichssystem messen	Ist das Ergebnis client- oder pfadspezifisch?
18	Ursache nachweisen	Welche Messwerte belegen die Fehlerstelle?
19	Minimale Änderung planen	Welche Korrektur behebt den Engpass?
20	Nachprüfung durchführen	Verbessern sich alle relevanten Messgrößen?

39. Kompakte Befehlstabelle

Aufgabe	Windows	Linux	macOS
Ping mit 20 Paketen	`[TEST] ping -n 20 ZIEL_IP`	`[TEST] ping -c 20 ZIEL_IP`	`[TEST] ping -c 20 ZIEL_IP`
Standardgateway	`[RO] Get-NetRoute -DestinationPrefix "0.0.0.0/0"`	`[RO] ip route show default`	`[RO] route -n get default`
Route zum Ziel	`[RO] Find-NetRoute -RemoteIPAddress ZIEL_IP`	`[RO] ip route get ZIEL_IP`	`[RO] route -n get ZIEL_IP`
Pfad ohne DNS	`[TEST] tracert -d ZIEL_IP`	`[TEST] traceroute -n ZIEL_IP`	`[TEST] traceroute -n ZIEL_IP`

Aufgabe	Windows	Linux	macOS
Pfad und Verlust	`[TEST] pathping /n ZIEL_IP`	Falls installiert: `[TEST] mtr -n -r -c 100 ZIEL_IP`	Falls installiert: `[TEST] mtr -n -r -c 100 ZIEL_IP`
TCP-Porttest	`[TEST] Test-NetConnection HOST -Port PORT`	`[TEST] nc -vz -w 5 HOST PORT`	`[TEST] nc -vz -w 5 HOST PORT`
TCP-Durchsatz	`[TEST] iperf3 -c SERVER_IP`	`[TEST] iperf3 -c SERVER_IP`	`[TEST] iperf3 -c SERVER_IP`
Gegenrichtung	`[TEST] iperf3 -c SERVER_IP -R`	`[TEST] iperf3 -c SERVER_IP -R`	`[TEST] iperf3 -c SERVER_IP -R`
UDP-Test	`[TEST] iperf3 -c SERVER_IP -u -b 10M`	`[TEST] iperf3 -c SERVER_IP -u -b 10M`	`[TEST] iperf3 -c SERVER_IP -u -b 10M`
Schnittstellenzähler	`[RO] Get-NetAdapterStatistics`	`[RO] ip -s link`	`[RO] netstat -ib`
Linkinformationen	`[RO] Get-NetAdapter`	`[RO][PRIV] sudo ethtool INTERFACE`	`[RO] ifconfig INTERFACE`
TCP-Statistik	`[RO] netstat -s -p tcp`	`[RO] nstat` oder `[RO] netstat -s`	`[RO] netstat -s -p tcp`
MTU	`[RO] Get-NetIPInterface`	`[RO] ip link show`	`[RO] ifconfig`
DF-Test	`[TEST] ping -4 -f -l 1400 ZIEL_IP`	`[TEST] ping -4 -M do -s 1400 ZIEL_IP`	`[TEST] ping -D -s 1400 ZIEL_IP`
HTTP-Zeiten	`[TEST][SENS] curl.exe -w ... URL`	`[TEST][SENS] curl -w ... URL`	`[TEST][SENS] curl -w ... URL`

`ZIEL_IP`, `HOST`, `PORT`, `SERVER_IP`, `INTERFACE` und `URL` müssen durch die Werte des konkreten Störungsfalls ersetzt werden.

---

40. Dokumentationsvorlage für Leistungsprobleme

Störung:

Zeitpunkt:

Zeitzone:

Standort:

Benutzer:

Client:

Betriebssystem:

Quellschnittstelle:

Quell-IP:

Linkgeschwindigkeit:

Duplex:

WLAN-SSID:

WLAN-BSSID:

WLAN-Signal:

VPN beteiligt:

Proxy beteiligt:

Zielhostname:

Ziel-IP:

Zielport:

Transportprotokoll:

IPv4 oder IPv6:

Erwarteter Datenpfad:

Gemessener Datenpfad:

Gateway-RRT Minimum:

Gateway-RRT Durchschnitt:

Gateway-RRT Maximum:

Gateway-Paketverlust:

Ziel-RTT Minimum:

Ziel-RTT Durchschnitt:

Ziel-RTT Maximum:

Ziel-Paketverlust:

Jitter:

Burst Loss:

iperf3 TCP Client zu Server:

iperf3 TCP Server zu Client:

iperf3 parallele Streams:

iperf3 UDP Zielrate:

iperf3 UDP empfangene Rate:

iperf3 UDP Jitter:

iperf3 UDP Verlust:

Latenz ohne Last:

Latenz unter Uploadlast:

Latenz unter Downloadlast:

RX-Fehler vorher:

RX-Fehler nachher:

TX-Fehler vorher:

TX-Fehler nachher:

Drops vorher:

Drops nachher:

TCP-Retransmissions:

Client-CPU:

Server-CPU:

Server-Datenträger:

Testdauer:

Anzahl der Messungen:

Vergleichssystem:

Vergleichszeitpunkt:

MTU:

Größte erfolgreiche DF-Nutzlast:

Paketmitschnitt:

Festgestellte Ursache:

Genehmigte Änderung:

Rückfallplan:

Ergebnis der Nachprüfung:

---

41. Kontrollfragen nach der Diagnose

- Wurde das Problem als Latenz-, Jitter-, Verlust- oder Durchsatzproblem konkretisiert?
- Sind Quelle, Ziel, Richtung und Datenpfad dokumentiert?
- Wurde zuerst das lokale Gateway gemessen?
- Wurden internes und externes Ziel getrennt geprüft?
- Wurden IPv4 und IPv6 getrennt verglichen?
- Wurde Hin- und Rückrichtung gemessen?
- Wurden Einzelspitzen und nicht nur Durchschnittswerte betrachtet?
- Wurde ein längerer Messzeitraum verwendet?
- Wurde Zwischenhop-Verlust korrekt interpretiert?
- Wurde TCP mit einem und mehreren Datenströmen verglichen?
- Wurde UDP mit einer kontrollierten Zielrate getestet?
- Wurden Jitter und Burst Loss erfasst?
- Wurde die Latenz unter Last geprüft?
- Steigen Schnittstellenfehler während des Tests?

- Stimmen Linkgeschwindigkeit und Duplex?
- Wurden Client- und Serverauslastung kontrolliert?
- Wurde iperf3 von Datei- oder Anwendungsleistung getrennt?
- Wurde die MTU geprüft?
- Wurden Paketmitschnitte an beiden Enden erwogen?
- War der Testumfang genehmigt und kontrolliert?
- Wurde nach der Änderung mit derselben Methode erneut gemessen?

---

42. Quellen und weiterführende Dokumentation

- Microsoft Learn – `ping`: https://learn.microsoft.com/windows-server/administration/windows-commands/ping
- Microsoft Learn – `tracert`: https://learn.microsoft.com/windows-server/administration/windows-commands/tracert
- Microsoft Learn – `pathping`: https://learn.microsoft.com/de-de/windows-server/administration/windows-commands/pathping
- Microsoft Sysinternals – PsPing: https://learn.microsoft.com/de-de/sysinternals/downloads/psping
- ESnet – iperf3-Dokumentation: https://software.es.net/iperf/
- ESnet – iperf3-Aufruf und Optionen: https://software.es.net/iperf/invoking.html
- Linux-Handbuch – `ping`: https://man7.org/linux/man-pages/man8/ping.8.html
- Linux-Handbuch – `tracepath`: https://man7.org/linux/man-pages/man8/tracepath.8.html
- MTR – Offizielle Dokumentation: https://mtr.readthedocs.io/
- Wireshark – TCP Display Filter Reference: https://www.wireshark.org/docs/dfref/t/tcp.html
- Wireshark – Benutzerhandbuch: https://www.wireshark.org/docs/wsug_html_chunked/
- RFC 3393 – IP Packet Delay Variation Metric: https://www.rfc-editor.org/rfc/rfc3393.html
- RFC 2680 – One-way Packet Loss Metric: https://www.rfc-editor.org/rfc/rfc2680.html
- RFC 6349 – Framework for TCP Throughput Testing: https://www.rfc-editor.org/rfc/rfc6349.html
- RFC 8201 – Path MTU Discovery for IPv6: https://www.rfc-editor.org/rfc/rfc8201.html

---