2.3 Linux-Netzwerkbefehle

Linux stellt für die Netzwerkdiagnose verschiedene Werkzeuge bereit. Welche Befehle verfügbar sind, hängt von Distribution, Installation und verwendeter Netzwerkverwaltung ab.

Die wichtigsten Werkzeuggruppen sind:

- `iproute2` mit `ip` und `ss`
- `iputils` mit `ping` und `tracepath`
- DNS-Werkzeuge wie `resolvectl`, `dig`, `host` und `getent`
- NetworkManager mit `nmcli`
- systemd-networkd mit `networkctl`
- Treiber- und Linkdiagnose mit `ethtool`
- Socket- und Prozessdiagnose mit `ss`, `lsof` und `fuser`
- Firewallanalyse mit `nft`, `iptables`, `ufw` oder `firewall-cmd`
- Protokollanalyse mit `journalctl` und `dmesg`

Kennzeichnungen

Kennzeichnung	Bedeutung
`[RO]`	Nur lesender Befehl; verändert keine Konfiguration
`[TEST]`	Führt einen aktiven Netzwerk- oder Verbindungstest aus
`[PRIV]`	Benötigt Root-Rechte beziehungsweise `sudo`
`[CHANGE]`	Verändert einen Zustand oder eine Konfiguration
`[DISRUPT]`	Kann eine Verbindung oder einen Dienst unterbrechen
`[FILE]`	Schreibt Informationen in eine Datei
`[SENS]`	Ausgabe kann sensible System-, Netzwerk- oder Prozessdaten enthalten

> Vor Veränderungen sollten zunächst die lesenden `[RO]`-Befehle verwendet und deren Ergebnisse dokumentiert werden.

---

1. Linux-System und verfügbare Werkzeuge prüfen

Zunächst muss geklärt werden, welche Distribution, welcher Kernel und welche Netzwerkwerkzeuge vorhanden sind.

Aufgabe	Linux-Befehl
Distribution anzeigen	`[RO] cat /etc/os-release`
Kernelversion anzeigen	`[RO] uname -r`
Architektur anzeigen	`[RO] uname -m`
Hostname anzeigen	`[RO] hostnamectl`
Aktuellen Benutzer anzeigen	`[RO] id`
Prüfen, ob ein Befehl vorhanden ist	`[RO] command -v ip`
Mehrere Werkzeuge prüfen	`[RO] command -v ip ss ping tracepath traceroute mtr dig resolvectl nmcli networkctl ethtool`
Version von `iproute2` anzeigen	`[RO] ip -Version`
Hilfe zum `ip`-Befehl anzeigen	`[RO] ip help`
Handbuchseite öffnen	`[RO] man ip`
Handbuch für einen Teilbefehl öffnen	`[RO] man ip-route`
Kurzhilfe eines Teilbefehls anzeigen	`[RO] ip route help`

Typische Paketnamen

Werkzeug	Häufiger Paketname
`ip`, `ss`	`iproute2`
`ping`, `tracepath`	`iputils` beziehungsweise `iputils-ping` und `iputils-tracepath`
`dig`, `host`	`dnsutils`, `bind-utils` oder `bind-tools`
`traceroute`	`traceroute`
`mtr`	`mtr`
`ethtool`	`ethtool`
`lsof`	`lsof`
`nc`	`netcat-openbsd`, `nmap-ncat` oder vergleichbares Paket

> Paketnamen unterscheiden sich zwischen Debian, Ubuntu, Fedora, Red Hat Enterprise Linux, Rocky Linux, AlmaLinux, SUSE, Arch Linux und anderen Distributionen.

---

2. Netzwerkschnittstellen und Linkstatus prüfen

`ip link` zeigt Netzwerkschnittstellen unabhängig davon, ob bereits eine IP-Adresse konfiguriert wurde.

Aufgabe	Linux-Befehl
Alle Schnittstellen anzeigen	`[RO] ip link show`
Kompakte Übersicht anzeigen	`[RO] ip -brief link show`
Details und Statistiken anzeigen	`[RO] ip -details -statistics link show`
Einzelne Schnittstelle anzeigen	`[RO] ip link show dev enp1s0`
Empfangs- und Sendestatistik anzeigen	`[RO] ip -statistics link show dev enp1s0`
Betriebszustand über sysfs lesen	`[RO] cat /sys/class/net/enp1s0/operstate`
Physische Trägererkennung lesen	`[RO] cat /sys/class/net/enp1s0/carrier`
MTU anzeigen	`[RO] cat /sys/class/net/enp1s0/mtu`
MAC-Adresse anzeigen	`[RO] cat /sys/class/net/enp1s0/address`
Alle Schnittstellennamen auflisten	`[RO] ls -1 /sys/class/net`

> “ `enp1s0` ist nur ein Beispiel. Der tatsächliche Name kann beispielsweise `eth0`, `ens18`, `eno1`, `enp3s0`, `wlan0` oder `wlp2s0` lauten.

Wichtige Angaben von `ip link`

Angabe	Bedeutung
`UP`	Schnittstelle wurde administrativ aktiviert
`LOWER_UP`	Physische beziehungsweise untergeordnete Verbindung wurde erkannt
`NO-CARRIER`	Kein physischer Link erkannt
`state UP`	Schnittstelle ist betriebsbereit
`state DOWN`	Schnittstelle ist nicht aktiv
`mtu`	Maximum Transmission Unit
`link/ether`	MAC-Adresse einer Ethernet-Schnittstelle
`qlen`	Länge der Sendewarteschlange

Typische Fehlerbilder

Beobachtung	Mögliche Bedeutung

`state DOWN`	Schnittstelle wurde deaktiviert oder nicht aktiviert
`UP` aber kein `LOWER_UP`	Kein physischer Link, Kabel-, Port- oder WLAN-Problem
`NO-CARRIER`	Kein Trägersignal erkannt
Viele `errors`	Kabel, Port, Treiber, Hardware oder Aushandlung prüfen
Viele `dropped`	Überlastung, Warteschlangen, Treiber oder Puffer prüfen
Unerwartete MTU	Kann Fragmentierungs- oder Tunnelprobleme verursachen
Schnittstelle fehlt vollständig	Treiber, Hardwareerkennung, virtuelle Maschine oder Gerätezuordnung prüfen

3. IP-Adressen und Adresszustände prüfen

Aufgabe	Linux-Befehl
Alle Adressen anzeigen	`[RO] ip address show`
Kompakte Adressübersicht	`[RO] ip -brief address show`
IPv4-Adressen anzeigen	`[RO] ip -4 address show`
IPv6-Adressen anzeigen	`[RO] ip -6 address show`
Adressen einer Schnittstelle anzeigen	`[RO] ip address show dev enp1s0`
Nur globale IPv4-Adressen anzeigen	`[RO] ip -4 address show scope global`
Nur globale IPv6-Adressen anzeigen	`[RO] ip -6 address show scope global`
Adressen ohne Loopback anzeigen	`[RO] ip -brief address show | grep -v '^lo'`
Hostadressen kompakt anzeigen	`[RO] hostname -I`

Wichtige Angaben

Angabe	Bedeutung
`inet`	IPv4-Adresse
`inet6`	IPv6-Adresse
`/24`, `/64`	Präfixlänge
`scope host`	Nur lokal auf diesem System gültig
`scope link`	Nur im direkt verbundenen Netz gültig
`scope global`	Über das lokale System beziehungsweise Netz hinaus verwendbar

Angabe	Bedeutung
`dynamic`	Adresse wurde dynamisch vergeben
`secondary`	Zusätzliche Adresse auf derselben Schnittstelle
`tentative`	IPv6 Duplicate Address Detection ist noch nicht abgeschlossen
`deprecated`	Adresse soll nicht mehr für neue Verbindungen verwendet werden
`valid_lft`	Verbleibende Gültigkeitsdauer
`preferred_lft`	Verbleibende bevorzugte Nutzungsdauer

Prüfpunkte

- Passt die IP-Adresse zum vorgesehenen Netz?
- Ist die Präfixlänge korrekt?
- Ist die Adresse an der richtigen Schnittstelle gebunden?
- Existieren unerwartete zusätzliche Adressen?
- Wird eine dynamische Adresse erwartet?
- Ist eine IPv6-Adresse noch `tentative` oder bereits `deprecated`?
- Befindet sich eine IPv4-Adresse im Bereich `169.254.0.0/16`?
- Existiert nur eine Loopback-Adresse?

> Eine Adresse aus `169.254.0.0/16` kann darauf hinweisen, dass keine reguläre IPv4-Konfiguration beziehungsweise keine erfolgreiche DHCP-Zuweisung vorhanden ist.

---

4. Routingtabelle und verwendeten Netzwerkweg prüfen

Aufgabe	Linux-Befehl
IPv4-Routingtabelle anzeigen	`[RO] ip route show`
IPv6-Routingtabelle anzeigen	`[RO] ip -6 route show`
Alle Routingtabellen anzeigen	`[RO] ip route show table all`
Standardroute anzeigen	`[RO] ip route show default`
Route zu einem Ziel bestimmen	`[RO] ip route get 1.1.1.1`
Route mit gewünschter Quelladresse prüfen	`[RO] ip route get 1.1.1.1 from 192.0.2.10`
Route über eine Schnittstelle suchen	`[RO] ip route show dev enp1s0`

Aufgabe	Linux-Befehl
Policy-Routing-Regeln anzeigen	`[RO] ip rule show`
IPv6-Policy-Regeln anzeigen	`[RO] ip -6 rule show`
Routingtabellen-Namen anzeigen	`[RO] cat /etc/iproute2/rt_tables`
Änderungen an Routen überwachen	`[RO] ip monitor route`

Wichtige Angaben

Angabe	Bedeutung
`default`	Standardroute für nicht spezifischer bekannte Ziele
`via`	Nächster Router beziehungsweise Gateway
`dev`	Verwendete Netzwerkschnittstelle
`src`	Bevorzugte lokale Quelladresse
`metric`	Routenmetrik
`proto dhcp`	Route wurde über DHCP bereitgestellt
`proto kernel`	Route wurde automatisch durch den Kernel erzeugt
`table`	Verwendete Routingtabelle
`scope link`	Ziel befindet sich direkt am lokalen Link

Beispielausgabe

```
default via 192.0.2.1 dev enp1s0 proto dhcp src 192.0.2.10 metric 100
192.0.2.0/24 dev enp1s0 proto kernel scope link src 192.0.2.10 metric 100
```

Interpretation

- Das Standardgateway ist `192.0.2.1`.
- Die Schnittstelle `enp1s0` wird verwendet.
- Die bevorzugte Quelladresse ist `192.0.2.10`.
- Das Netz `192.0.2.0/24` ist direkt angeschlossen.
- Bei mehreren Routen können Präfixlänge, Policy-Regeln und Metriken die Auswahl beeinflussen.

Typische Fehlerbilder

- Keine Standardroute vorhanden.
- Standardroute verwendet die falsche Schnittstelle.
- VPN- oder Tunnelroute überschreibt den erwarteten Weg.

- Falsche Quelladresse wird ausgewählt.
- Eine spezifischere Route überschreibt die Standardroute.
- Policy Routing verwendet eine andere Routingtabelle.
- Mehrere Standardrouten führen zu einem unerwarteten Netzwerkweg.

---

5. ARP- und IPv6-Nachbartabelle prüfen

`ip neigh` zeigt IPv4-ARP- und IPv6-Nachbareinträge.

Aufgabe	Linux-Befehl
Nachbartabelle anzeigen	`[RO] ip neighbour show`
Kompakte Nachbartabelle anzeigen	`[RO] ip -brief neighbour show`
IPv4-Nachbarn anzeigen	`[RO] ip -4 neighbour show`
IPv6-Nachbarn anzeigen	`[RO] ip -6 neighbour show`
Nachbarn einer Schnittstelle anzeigen	`[RO] ip neighbour show dev enp1s0`
Bestimmte IP-Adresse suchen	`[RO] ip neighbour show 192.0.2.1`
Fehlgeschlagene Einträge anzeigen	`[RO] ip neighbour show nud failed`
Unvollständige Einträge anzeigen	`[RO] ip neighbour show nud incomplete`
Änderungen überwachen	`[RO] ip monitor neighbour`

Wichtige Zustände

Zustand	Bedeutung
`REACHABLE`	Nachbar wurde kürzlich erfolgreich erreicht
`STALE`	Eintrag ist vorhanden, wurde aber länger nicht bestätigt
`DELAY`	Prüfung der Erreichbarkeit wird verzögert
`PROBE`	Aktive Erreichbarkeitsprüfung läuft
`INCOMPLETE`	Adressauflösung ist noch nicht abgeschlossen
`FAILED`	Adressauflösung beziehungsweise Erreichbarkeitsprüfung ist fehlgeschlagen
`PERMANENT`	Dauerhaft konfigurierter Eintrag
`NOARP`	Für diesen Eintrag wird keine normale Nachbarauflösung verwendet

Fehlerhinweise

- `INCOMPLETE` oder `FAILED` kann auf ein falsches VLAN, eine falsche Netzmaske, einen nicht erreichbaren Nachbarn oder ein Layer-2-Problem hinweisen.
- Unterschiedliche MAC-Adressen für dieselbe IP-Adresse können auf doppelte IP-Adressen, Hochverfügbarkeit oder einen Sicherheitsvorfall hinweisen.
- Ein fehlender Eintrag ist nicht automatisch ein Fehler. Das Ziel wurde möglicherweise noch nicht angesprochen oder liegt außerhalb des lokalen Netzes.

---

6. Erreichbarkeit mit ping prüfen

Aufgabe	Linux-Befehl
Vier ICMP-Anfragen senden	`[TEST] ping -c 4 192.0.2.1`
Externe IP-Adresse testen	`[TEST] ping -c 4 1.1.1.1`
DNS-Namen testen	`[TEST] ping -c 4 example.com`
Nur IPv4 verwenden	`[TEST] ping -4 -c 4 example.com`
Nur IPv6 verwenden	`[TEST] ping -6 -c 4 example.com`
Bestimmte Schnittstelle verwenden	`[TEST] ping -I enp1s0 -c 4 192.0.2.1`
Bestimmte Quelladresse verwenden	`[TEST] ping -I 192.0.2.10 -c 4 192.0.2.1`
Timeout pro Antwort begrenzen	`[TEST] ping -c 4 -W 2 192.0.2.1`
Gesamtlaufzeit begrenzen	`[TEST] ping -c 4 -w 10 192.0.2.1`
Paketgröße prüfen	`[TEST] ping -c 4 -s 1400 192.0.2.1`
Keine Fragmentierung erlauben	`[TEST] ping -c 4 -M do -s 1400 192.0.2.1`

Wichtige Ergebnisse

Wert	Bedeutung
`time`	Antwortzeit beziehungsweise Round Trip Time
`ttl`	Verbleibende IPv4 Time to Live
`icmp_seq`	Nummer der ICMP-Anfrage
`packet loss`	Anteil verlorener Anfragen
`min/avg/max`	Minimale, durchschnittliche und maximale Laufzeit
`mdev`	Streuung der Laufzeiten unter Linux

Interpretation

Ergebnis	Mögliche Bedeutung

IP-Adresse erreichbar, DNS-Name nicht	DNS-Problem wahrscheinlich
Gateway nicht erreichbar	Lokales Netz, VLAN, WLAN, Switch oder Gateway prüfen
Ziel nicht erreichbar, Gateway erreichbar	Routing, Firewall oder entferntes System prüfen
Einzelne Paketverluste	Überlastung, Funkstörung, Linkfehler oder Rate Limiting möglich
Hohe Laufzeitschwankungen	Überlastung, WLAN-Störung oder wechselnder Netzwerkweg
`Destination Host Unreachable`	Lokales System oder Router kennt keinen funktionierenden Weg
Keine Antwort	Ziel offline, ICMP blockiert oder Netzwerkweg gestört

> Ein fehlgeschlagener Ping beweist nicht, dass ein Ziel vollständig unerreichbar ist. ICMP kann blockiert sein, während ein TCP-Dienst weiterhin funktioniert.

7. Netzwerkpfad mit tracepath, traceroute und mtr untersuchen

Aufgabe	Linux-Befehl
Pfad ohne Root-Rechte prüfen	`[TEST] tracepath example.com`
IPv4-Pfad prüfen	`[TEST] tracepath -4 example.com`
IPv6-Pfad prüfen	`[TEST] tracepath -6 example.com`
Klassische Routenverfolgung	`[TEST] traceroute example.com`
Keine Namensauflösung durchführen	`[TEST] traceroute -n example.com`
TCP-Traceroute zu Port 443	`[TEST][PRIV] sudo traceroute -T -p 443 example.com`
Laufende kombinierte Messung	`[TEST] mtr example.com`
Bericht mit 20 Messzyklen	`[TEST] mtr --report --report-cycles 20 example.com`
Bericht ohne DNS-Auflösung	`[TEST] mtr --report --numeric --report-cycles 20 example.com`
TCP-MTR zu Port 443	`[TEST][PRIV] sudo mtr --tcp --port 443 --report example.com`

Unterschiede

Werkzeug	Schwerpunkt

`tracepath`	Einfacher Pfadtest und Erkennung der Path MTU; häufig ohne Root-Rechte nutzbar
`traceroute`	Klassische Routenverfolgung mit verschiedenen Protokolloptionen
`mtr`	Kombiniert laufende Ping- und Traceroute-Messungen

Wichtige Hinweise zur Auswertung

- Ein einzelner nicht antwortender Hop beweist keinen Fehler.
- Router können Diagnosepakete begrenzen oder ignorieren und trotzdem Nutzdaten weiterleiten.
- Paketverlust ist besonders relevant, wenn er ab einem Hop beginnt und auch an allen folgenden Hops einschließlich des Ziels sichtbar bleibt.
- Unterschiedliche Pfade können durch Load Balancing entstehen.
- Ein erfolgreicher letzter Hop ist wichtiger als einzelne Sterne in der Mitte des Pfades.
- ICMP-, UDP- und TCP-Traceroute können unterschiedliche Ergebnisse liefern.

---

8. DNS-Konfiguration und Namensauflösung prüfen

Welche DNS-Werkzeuge funktionieren, hängt vom verwendeten Resolver und den installierten Paketen ab.

Aufgabe	Linux-Befehl
Resolver-Konfiguration prüfen	`[RO] cat /etc/resolv.conf`
systemd-resolved-Status anzeigen	`[RO] resolvectl status`
DNS-Server pro Schnittstelle anzeigen	`[RO] resolvectl dns`
DNS-Domänen pro Schnittstelle anzeigen	`[RO] resolvectl domain`
Namen über den Systemresolver abfragen	`[TEST] resolvectl query example.com`
Auflösung über NSS testen	`[TEST] getent hosts example.com`
IPv4- und IPv6-Ergebnisse anzeigen	`[TEST] getent ahosts example.com`
Standard-DNS-Abfrage durchführen	`[TEST] dig example.com`
IPv4-Adresse abfragen	`[TEST] dig A example.com`
IPv6-Adresse abfragen	`[TEST] dig AAAA example.com`
Mailserver abfragen	`[TEST] dig MX example.com`
Nameserver abfragen	`[TEST] dig NS example.com`
Kurzausgabe erzeugen	`[TEST] dig +short example.com`

Aufgabe	Linux-Befehl
Bestimmten DNS-Server verwenden	`[TEST] dig @192.0.2.53 example.com`
Vollständigen Delegationsweg prüfen	`[TEST] dig +trace example.com`
Reverse-DNS-Abfrage durchführen	`[TEST] dig -x 192.0.2.10`
Alternative kompakte Abfrage	`[TEST] host example.com`
Reverse-Abfrage mit `host`	`[TEST] host 192.0.2.10`

Warum können die Ergebnisse unterschiedlich sein?

Werkzeug	Verwendeter Weg
`getent hosts`	Systemweite Name Service Switch-Konfiguration aus `/etc/nsswitch.conf`
`resolvectl query`	systemd-resolved
`dig`	Direkte DNS-Abfrage; berücksichtigt nicht automatisch alle NSS-Quellen
`host`	Direkte DNS-Abfrage
Anwendung	Kann eigenen Resolver, eigenen Cache oder DNS over HTTPS verwenden

Zusätzliche Prüfungen

Aufgabe	Linux-Befehl
NSS-Reihenfolge prüfen	`[RO] grep '^hosts:' /etc/nsswitch.conf`
Lokale Hosts-Datei prüfen	`[RO] cat /etc/hosts`
Status von systemd-resolved prüfen	`[RO] systemctl status systemd-resolved --no-pager`
DNS-Protokoll anzeigen	`[RO][PRIV] sudo journalctl -u systemd-resolved --since "30 minutes ago"`
DNS-Statistik anzeigen	`[RO] resolvectl statistics`

Typische Fehlerbilder

- Falscher DNS-Server ist eingetragen.
- `/etc/resolv.conf` zeigt auf eine nicht erreichbare Adresse.
- `/etc/resolv.conf` ist ein Symlink auf eine unerwartete Datei.
- Split-DNS oder VPN weist bestimmten Domänen andere Resolver zu.
- `/etc/hosts` überschreibt das erwartete DNS-Ergebnis.
- `dig` funktioniert, eine Anwendung jedoch nicht.
- Interne Namen werden fälschlicherweise an einen öffentlichen DNS-Server gesendet.
- IPv4- und IPv6-Auflösung liefern unterschiedliche Ergebnisse.

> Interne Hostnamen und DNS-Zonen können sensible Unternehmensinformationen darstellen. Sie sollten nicht unüberlegt an öffentliche Resolver gesendet werden.

---

9. NetworkManager mit nmcli diagnostizieren

`nmcli` ist das Kommandozeilenwerkzeug für NetworkManager.

Aufgabe	Linux-Befehl
Allgemeinen Status anzeigen	`[RO] nmcli general status`
Prüfen, ob NetworkManager läuft	`[RO] nmcli -terse -fields RUNNING general`
Gerätestatus anzeigen	`[RO] nmcli device status`
Details aller Geräte anzeigen	`[RO] nmcli device show`
Details einer Schnittstelle anzeigen	`[RO] nmcli device show enp1s0`
Verbindungprofile auflisten	`[RO] nmcli connection show`
Aktive Profile anzeigen	`[RO] nmcli connection show --active`
Detailinformationen eines Profils anzeigen	`[RO] nmcli connection show "PROFILNAME"`
IP-Adressen eines Profils anzeigen	`[RO] nmcli -get-values IP4.ADDRESS connection show "PROFILNAME"`
DNS-Daten eines Profils anzeigen	`[RO] nmcli -get-values IP4.DNS connection show "PROFILNAME"`
NetworkManager-Konnektivitätsstatus anzeigen	`[RO] nmcli networking connectivity`
Konnektivität neu prüfen	`[TEST] nmcli networking connectivity check`
WLAN-Zustand anzeigen	`[RO] nmcli radio wifi`
Sichtbare WLANs auflisten	`[TEST] nmcli device wifi list`
Änderungen überwachen	`[RO] nmcli monitor`

> Für Skripte sollten die vollständigen Befehlsnamen statt möglicherweise mehrdeutiger Abkürzungen verwendet werden.

Wichtige Gerätezustände

Zustand	Bedeutung

`connected`	Gerät verwendet ein aktives Verbindungsprofil
`disconnected`	Gerät ist vorhanden, aber nicht verbunden
`connecting`	Verbindungsaufbau läuft
`unavailable`	Gerät kann momentan nicht verwendet werden
`unmanaged`	NetworkManager verwaltet dieses Gerät nicht

Wichtiger Unterschied

- **Device:** Tatsächliche physische oder virtuelle Netzwerkschnittstelle.
- **Connection:** Gespeichertes NetworkManager-Verbindungsprofil.
- Ein Gerät kann mehrere passende Profile besitzen, aber normalerweise nur ein aktives Profil gleichzeitig verwenden.

---

10. systemd-networkd mit networkctl diagnostizieren

`networkctl` zeigt den Zustand von Schnittstellen, wenn systemd-networkd verwendet wird.

Aufgabe	Linux-Befehl
Schnittstellenübersicht anzeigen	`[RO] networkctl list`
Gesamtstatus anzeigen	`[RO] networkctl status`
Einzelne Schnittstelle anzeigen	`[RO] networkctl status enp1s0`
LLDP-Nachbarn anzeigen	`[RO] networkctl lldp`
Status von systemd-networkd anzeigen	`[RO] systemctl status systemd-networkd --no-pager`
Protokoll der letzten 30 Minuten anzeigen	`[RO][PRIV] sudo journalctl -u systemd-networkd --since "30 minutes ago"`
Konfigurationsdateien auflisten	`[RO] find /etc/systemd/network /run/systemd/network /usr/lib/systemd/network -maxdepth 1 -type f 2>/dev/null`

Typische Fehlerbilder

- Eine Schnittstelle ist `unmanaged`.
- Keine passende `.network`-Datei trifft auf die Schnittstelle zu.
- Mehrere Konfigurationsdateien konkurrieren miteinander.
- NetworkManager und systemd-networkd sollen dieselbe Schnittstelle verwalten.
- DHCP läuft nicht oder liefert keine Adresse.
- Die Schnittstelle befindet sich im Zustand `configuring` oder `failed`.

```
```

> Nicht gleichzeitig davon ausgehen, dass NetworkManager und systemd-networkd dieselbe Schnittstelle verwalten. Zuerst muss festgestellt werden, welcher Dienst tatsächlich zuständig ist.

---

11. Link, Treiber und Hardware mit ethtool prüfen

`ethtool` liefert Informationen über Ethernet-Schnittstellen, Treiber und Aushandlung.

Aufgabe	Linux-Befehl
Linkeinstellungen anzeigen	`[RO] ethtool enp1s0`
Treiberinformationen anzeigen	`[RO] ethtool -i enp1s0`
Schnittstellenstatistik anzeigen	`[RO][PRIV] sudo ethtool -S enp1s0`
Offload-Funktionen anzeigen	`[RO] ethtool -k enp1s0`
Pause-Parameter anzeigen	`[RO] ethtool -a enp1s0`
Ringpuffer anzeigen	`[RO][PRIV] sudo ethtool -g enp1s0`
Zeitstempelfunktionen anzeigen	`[RO] ethtool -T enp1s0`

Wichtige Angaben

Angabe	Bedeutung
`Speed`	Ausgehandelte Geschwindigkeit
`Duplex`	Halb- oder Vollduplex
`Auto-negotiation`	Automatische Aushandlung aktiviert oder deaktiviert
`Link detected`	Physischer Link erkannt
`Port`	Verwendeter Anschlusstyp
`driver`	Verwendetes Kernelmodul
`version`	Treiberversion
`firmware-version`	Firmwareversion des Adapters

Typische Fehlerbilder

- `Link detected: no`
- Unerwartet niedrige Geschwindigkeit
- Half-Duplex statt Full-Duplex
- Unterschiedliche feste Einstellungen an beiden Link-Enden

- Steigende CRC-, Frame-, Drop- oder Timeout-Zähler
- Falscher oder sehr alter Treiber
- Treiber- oder Firmwarefehler im Kernelprotokoll

---

12. Offene Ports, Verbindungen und Prozesse mit ss prüfen

`ss` ist das zentrale moderne Werkzeug zur Untersuchung von Sockets und ersetzt viele frühere `netstat`-Anwendungsfälle.

Aufgabe	Linux-Befehl
Socket-Zusammenfassung anzeigen	`[RO] ss -summary`
Lauschende TCP-Ports anzeigen	`[RO] ss -listen -tcp -numeric`
Lauschende TCP- und UDP-Ports anzeigen	`[RO] ss -listen -numeric -tcp -udp`
Ports mit Prozessen anzeigen	`[RO][PRIV] sudo ss -listen -numeric -tcp -udp -processes`
Alle TCP-Verbindungen anzeigen	`[RO] ss -all -tcp -numeric`
Bestehende TCP-Verbindungen anzeigen	`[RO] ss -tcp -numeric state established`
TCP-Verbindungen mit internen Informationen	`[RO] ss -tcp -numeric -info`
Lokalen Port 443 prüfen	`[RO] ss -listen -tcp -numeric 'sport = :443'`
Verbindungen zu Zielport 443 anzeigen	`[RO] ss -tcp -numeric 'dport = :443'`
UNIX-Sockets anzeigen	`[RO] ss -x -all`

Häufig verwendete Kurzform

```
sudo ss -lntup
```

Die Optionen bedeuten:

Option	Bedeutung
`-l`	Nur lauschende Sockets
`-n`	Numerische Adressen und Ports
`-t`	TCP
`-u`	UDP
`-p`	Zugehörige Prozesse anzeigen

Interpretation

Beobachtung	Mögliche Bedeutung
Erwarteter Port fehlt	Dienst läuft nicht oder lauscht auf einem anderen Port
Dienst lauscht nur auf `127.0.0.1`	Nur lokale Verbindungen möglich
Dienst lauscht auf `0.0.0.0`	Lauscht grundsätzlich auf allen IPv4-Adressen
Dienst lauscht auf `::`	Lauscht auf IPv6; IPv4-Verhalten hängt zusätzlich von System und Anwendung ab
Viele Verbindungen in `SYN-SENT`	Ziel oder Firewall antwortet möglicherweise nicht
Viele Verbindungen in `SYN-RECV`	Viele eingehende, noch nicht vollständig aufgebaute Verbindungen
Große `Send-Q`	Daten können möglicherweise nicht schnell genug gesendet werden
Große `Recv-Q`	Anwendung verarbeitet empfangene Daten möglicherweise nicht schnell genug

13. Ports und Prozesse mit lsof und fuser zuordnen

Aufgabe	Linux-Befehl
Alle Netzwerkdateien anzeigen	`[RO][PRIV] sudo lsof -numeric -P -i`
Lauschende TCP-Ports anzeigen	`[RO][PRIV] sudo lsof -numeric -P -iTCP -sTCP:LISTEN`
TCP-Port 443 untersuchen	`[RO][PRIV] sudo lsof -numeric -P -iTCP:443`
UDP-Port 53 untersuchen	`[RO][PRIV] sudo lsof -numeric -P -iUDP:53`
Prozess auf TCP-Port 443 suchen	`[RO][PRIV] sudo fuser -verbose 443/tcp`
Prozess auf UDP-Port 53 suchen	`[RO][PRIV] sudo fuser -verbose 53/udp`
Prozessdetails anhand der PID anzeigen	`[RO] ps -fp 1234`
Ausführliche Prozessinformationen anzeigen	`[RO] ps -p 1234 -o pid,ppid,user,group,cmd`

> Bei fehlenden Root-Rechten werden möglicherweise nicht alle Prozessinformationen angezeigt.

14. TCP- und UDP-Ports aktiv testen

Aufgabe	Linux-Befehl
TCP-Port testen	`[TEST] nc -verbose -zero -wait 3 example.com 443`
SSH-Port testen	`[TEST] nc -verbose -zero -wait 3 192.0.2.10 22`
Mehrere Ports testen	`[TEST] nc -verbose -zero -wait 3 192.0.2.10 22 80 443`
UDP-Port testen	`[TEST] nc -verbose -zero -udp -wait 3 192.0.2.53 53`
Bash-TCP-Test durchführen	`[TEST] timeout 3 bash -c '</dev/tcp/example.com/443'`

Häufig verwendete Kurzform

```
nc -vz -w 3 example.com 443
```

Mögliche Ergebnisse

Ergebnis	Bedeutung
`succeeded` oder `open`	TCP-Verbindung konnte aufgebaut werden
`Connection refused`	Ziel erreichbar, aber kein Dienst akzeptiert die Verbindung oder lehnt sie aktiv ab
`timed out`	Keine rechtzeitige Antwort; Firewall, Routing oder Zielsystem prüfen
`No route to host`	Kein verwendbarer Netzwerkweg oder entsprechende ICMP-Rückmeldung
`Name or service not known`	Namensauflösung fehlgeschlagen

> Ein UDP-Test mit `nc` kann ohne Antwort nicht zuverlässig beweisen, ob der UDP-Port offen oder geschlossen ist. UDP besitzt keinen TCP-Verbindungsaufbau.

---

15. Firewallstatus lesend prüfen

Linux-Systeme können unterschiedliche Werkzeuge zur Firewallverwaltung einsetzen. Zuerst muss festgestellt werden, welches System tatsächlich aktiv ist.

Aufgabe	Linux-Befehl
nftables-Regelwerk anzeigen	`[RO][PRIV][SENS] sudo nft list ruleset`

Aufgabe	Linux-Befehl
iptables-Filterregeln anzeigen	`[RO][PRIV][SENS] sudo iptables -S`
iptables-Regeln mit Zählern anzeigen	`[RO][PRIV][SENS] sudo iptables -L -numeric -verbose`
IPv6-iptables-Regeln anzeigen	`[RO][PRIV][SENS] sudo ip6tables -S`
UFW-Status anzeigen	`[RO][PRIV] sudo ufw status verbose`
firewalld-Status prüfen	`[RO] firewall-cmd --state`
Aktive firewalld-Zonen anzeigen	`[RO][PRIV] sudo firewall-cmd --get-active-zones`
Regeln der aktiven Zone anzeigen	`[RO][PRIV][SENS] sudo firewall-cmd --list-all`
Alle firewalld-Zonen anzeigen	`[RO][PRIV][SENS] sudo firewall-cmd --list-all-zones`
nftables-Dienststatus anzeigen	`[RO] systemctl status nftables --no-pager`
firewalld-Dienststatus anzeigen	`[RO] systemctl status firewalld --no-pager`
UFW-Dienststatus anzeigen	`[RO] systemctl status ufw --no-pager`

Prüfpunkte

- Welches Firewall-Frontend wird tatsächlich verwendet?
- Welches Regelwerk ist im Kernel aktiv?
- Welche Eingangs- und Ausgangsregeln gelten?
- Auf welche Schnittstelle oder Zone bezieht sich eine Regel?
- Stimmen IPv4- und IPv6-Regeln überein?
- Erhöhen sich die Paket- oder Bytezähler einer relevanten Regel?
- Wird ein Port lokal geöffnet, aber durch die Firewall blockiert?

> Die Firewall nicht pauschal deaktivieren. Zuerst müssen Dienstbindung, Route, verwendete Schnittstelle und passende Regeln geprüft werden.

---

16. Netzwerkdienste und Protokolle prüfen

Aufgabe	Linux-Befehl
Fehlgeschlagene Dienste anzeigen	`[RO] systemctl --failed`
NetworkManager-Status anzeigen	`[RO] systemctl status NetworkManager --no-pager`
systemd-networkd-Status anzeigen	`[RO] systemctl status systemd-networkd --no-pager`
systemd-resolved-Status anzeigen	`[RO] systemctl status systemd-resolved --no-pager`

Aufgabe	Linux-Befehl
Dienstprotokoll anzeigen	`[RO][PRIV] sudo journalctl -u NetworkManager --since "30 minutes ago"`
networkd-Protokoll anzeigen	`[RO][PRIV] sudo journalctl -u systemd-networkd --since "30 minutes ago"`
resolved-Protokoll anzeigen	`[RO][PRIV] sudo journalctl -u systemd-resolved --since "30 minutes ago"`
Kernelmeldungen zum Netzwerk filtern	`[RO][PRIV] sudo dmesg --human | grep -Ei 'link`
Kernelmeldungen des aktuellen Starts	`[RO][PRIV] sudo journalctl -kernel -boot`
Letzte Kernelmeldungen live verfolgen	`[RO][PRIV] sudo journalctl -kernel -follow`
Startzeit eines Dienstes anzeigen	`[RO] systemctl show NetworkManager -property ActiveEnterTimestamp`

Typische Protokollhinweise

- `link is down`
- `link becomes ready`
- `carrier lost`
- `renamed from eth0`
- `DHCP timeout`
- `no lease`
- `firmware failed to load`
- `authentication failed`
- `activation failed`
- `DNSSEC validation failed`
- `network unreachable`

> Protokolle müssen immer mit Zeitstempel, betroffener Schnittstelle und vorhergehenden Ereignissen ausgewertet werden. Eine einzelne Meldung ohne Kontext reicht selten zur Ursachenbestimmung.

---

17. Diagnoseinformationen sichern und vergleichen

Aufgabe	Linux-Befehl
Linkinformationen speichern	`[RO][FILE][SENS] ip -details -statistics link show > link-status.txt`
IP-Adressen speichern	`[RO][FILE][SENS] ip address show > ip-addresses.txt`
Routingtabellen speichern	`[RO][FILE][SENS] ip route show table all > routes.txt`

Aufgabe	Linux-Befehl
Policy-Regeln speichern	[RO][FILE][SENS] ip rule show > routing-rules.txt
Nachbartabelle speichern	[RO][FILE][SENS] ip neighbour show > neighbours.txt
Ports und Prozesse speichern	[RO][PRIV][FILE][SENS] sudo ss -lntup > listening-ports.txt
NetworkManager-Status speichern	[RO][FILE][SENS] nmcli device show > networkmanager-devices.txt
Resolverstatus speichern	[RO][FILE][SENS] resolvectl status > resolver-status.txt
Netzwerkprotokolle speichern	[RO][PRIV][FILE][SENS] sudo journalctl -u NetworkManager --since "1 hour ago" > networkmanager-log.txt

Zwei Messzeitpunkte vergleichen

```
ip -brief address show > addresses-vorher.txt
ip route show table all > routes-vorher.txt

# Messung oder freigegebene Änderung durchführen

ip -brief address show > addresses-nachher.txt
ip route show table all > routes-nachher.txt

diff -u addresses-vorher.txt addresses-nachher.txt
diff -u routes-vorher.txt routes-nachher.txt
```

Hinweise

- Diagnoseausgaben können interne IP-Adressen, DNS-Namen, MAC-Adressen, Benutzer, Prozesse und Netzstrukturen enthalten.
- Dateien müssen entsprechend den betrieblichen Datenschutz- und Sicherheitsvorgaben behandelt werden.
- Vor einer Weitergabe müssen vertrauliche Inhalte geprüft und gegebenenfalls redigiert werden.
- Passwörter, private Schlüssel, Tokens und vollständige VPN-Konfigurationen gehören nicht in ein allgemeines Diagnoselog.

---

18. Eingreifende Befehle nur nach der Diagnose verwenden

Die folgenden Befehle verändern den Systemzustand und können insbesondere über SSH die eigene Verbindung unterbrechen.

Maßnahme	Linux-Befehl	Auswirkung
Schnittstelle deaktivieren	`[CHANGE][PRIV][DISRUPT] sudo ip link set dev enp1s0 down`	Verbindung wird unterbrochen
Schnittstelle aktivieren	`[CHANGE][PRIV] sudo ip link set dev enp1s0 up`	Schnittstelle wird aktiviert
NetworkManager-Gerät trennen	`[CHANGE][PRIV][DISRUPT] sudo nmcli device disconnect enp1s0`	Aktive Verbindung wird getrennt
NetworkManager-Gerät verbinden	`[CHANGE][PRIV] sudo nmcli device connect enp1s0`	NetworkManager versucht eine Verbindung aufzubauen
Verbindungsprofil neu aktivieren	`[CHANGE][PRIV][DISRUPT] sudo nmcli connection up "PROFILNAME"`	Profil wird aktiviert beziehungsweise neu angewendet
NetworkManager neu starten	`[CHANGE][PRIV][DISRUPT] sudo systemctl restart NetworkManager`	Verwaltete Verbindungen können unterbrochen werden
systemd-networkd neu starten	`[CHANGE][PRIV][DISRUPT] sudo systemctl restart systemd-networkd`	Verwaltete Verbindungen können unterbrochen werden
DNS-Cache leeren	`[CHANGE][PRIV] sudo resolvectl flush-caches`	Lokale DNS-Cacheeinträge werden entfernt

Vor einem Eingriff prüfen

- Erfolgt der Zugriff lokal oder über SSH?
- Betrifft die Maßnahme die Schnittstelle der eigenen Sitzung?
- Ist die aktuelle Konfiguration dokumentiert?
- Ist bekannt, welcher Netzwerkdienst die Schnittstelle verwaltet?
- Besteht Konsolen-, Out-of-Band- oder Hypervisorzugriff?
- Gibt es eine Rückfallmöglichkeit?
- Sind Auswirkungen auf Container, virtuelle Maschinen, Bridges, Bonds oder VLANs bekannt?
- Ist eine betriebliche Freigabe erforderlich?

> Auf einem entfernten System darf die aktive Managementschnittstelle nicht ohne gesicherten Rückfallweg deaktiviert oder neu gestartet werden.

---

19. Praktische Linux-Diagnosereihenfolge

Schritt 1 – System und Netzwerkverwaltung identifizieren

```
cat /etc/os-release
systemctl is-active NetworkManager
systemctl is-active systemd-networkd
```

Schritt 2 – Linkstatus prüfen

```
ip -brief link show
ip -statistics link show
```

Schritt 3 – IP-Adressen prüfen

```
ip -brief address show
```

Schritt 4 – Standardroute und Routenauswahl prüfen

```
ip route show default
ip route get 1.1.1.1
ip rule show
```

Schritt 5 – Nachbartabelle prüfen

```
ip neighbour show
```

Schritt 6 – Standardgateway testen

```
ping -c 4 192.0.2.1
```

Schritt 7 – Externe IP-Adresse testen

```
ping -c 4 1.1.1.1
```

Schritt 8 – DNS-Konfiguration prüfen

```
cat /etc/resolv.conf
resolvectl status
```

Schritt 9 – DNS-Auflösung testen

```
getent hosts example.com
dig example.com
```

Schritt 10 – Zielport testen

```
nc -vz -w 3 example.com 443
```

Schritt 11 – Lokale Ports und Prozesse prüfen

```
sudo ss -lntup
```

Schritt 12 – Netzwerkpfad untersuchen

```
tracepath example.com
mtr --report --numeric --report-cycles 20 example.com
```

Schritt 13 – Firewallstatus prüfen

```
sudo nft list ruleset
```

Schritt 14 – Netzwerkprotokolle prüfen

```
sudo journalctl -u NetworkManager --since "30 minutes ago"
sudo journalctl -kernel --since "30 minutes ago"
```

Diagnoselogik

Ergebnis	Nächster Schwerpunkt
Schnittstelle fehlt	Hardware, Treiber, virtuelle Maschine oder Gerätezuordnung prüfen
Schnittstelle `DOWN`	Administrativen Zustand und Netzwerkverwaltung prüfen
Kein `LOWER_UP` beziehungsweise kein Carrier	Kabel, WLAN, Switchport oder Linkpartner prüfen
Keine passende IP-Adresse	DHCP oder statische Konfiguration prüfen

Ergebnis	Nächster Schwerpunkt
Keine Standardroute	Gateway- und Profilkonfiguration prüfen
Gateway nicht erreichbar	Lokales Netz, VLAN, Bridge, Switch oder WLAN prüfen
Externe IP erreichbar, DNS-Name nicht	DNS-Konfiguration und Resolver prüfen
Ziel-IP erreichbar, Zielport nicht	Dienst, Portbindung und Firewall prüfen
Falsche Quelladresse oder Schnittstelle	Routing, Policy Routing, VPN und Metrik prüfen
Dienst lauscht lokal, ist extern aber nicht erreichbar	Firewall, Routing, NAT und Bind-Adresse prüfen

---

Kurzreferenz

Diagnoseziel	Linux-Befehl
Schnittstellenstatus	`[RO] ip -brief link show`
Schnittstellenstatistik	`[RO] ip -statistics link show`
IP-Adressen	`[RO] ip -brief address show`
IPv4-Adressen	`[RO] ip -4 address show`
IPv6-Adressen	`[RO] ip -6 address show`
Standardroute	`[RO] ip route show default`
Route zu einem Ziel	`[RO] ip route get 1.1.1.1`
Policy Routing	`[RO] ip rule show`
Nachbartabelle	`[RO] ip neighbour show`
Gateway testen	`[TEST] ping -c 4 192.0.2.1`
Netzwerkpfad	`[TEST] tracepath example.com`
Fortlaufende Pfadanalyse	`[TEST] mtr example.com`
Resolverstatus	`[RO] resolvectl status`
Namensauflösung über NSS	`[TEST] getent hosts example.com`
DNS-Abfrage	`[TEST] dig example.com`
NetworkManager-Status	`[RO] nmcli general status`
NetworkManager-Geräte	`[RO] nmcli device status`
systemd-networkd-Status	`[RO] networkctl status`
Ethernet-Link	`[RO] ethtool enp1s0`
Treiberinformationen	`[RO] ethtool -i enp1s0`
Lauschende Ports	`[RO][PRIV] sudo ss -lntup`

Diagnoseziel	Linux-Befehl
TCP-Port testen	`[TEST] nc -vz -w 3 example.com 443`
Port einem Prozess zuordnen	`[RO][PRIV] sudo lsof -nP -iTCP:443`
nftables-Regeln	`[RO][PRIV] sudo nft list ruleset`
Dienststatus	`[RO] systemctl status DIENST --no-pager`
Dienstprotokoll	`[RO][PRIV] sudo journalctl -u DIENST`
Kernelprotokoll	`[RO][PRIV] sudo journalctl -kernel`

Veraltete und moderne Befehle

Älterer Befehl	Bevorzugter moderner Befehl
`ifconfig`	`ip address`, `ip link`
`route -n`	`ip route`
`arp -n`	`ip neighbour`
`netstat -lntup`	`ss -lntup`

> Ältere Werkzeuge können weiterhin installiert sein. Für neue Dokumentationen und aktuelle Linux-Systeme sollten grundsätzlich die `iproute2`-Werkzeuge verwendet werden.

Merksatz

> Die Linux-Netzwerkdiagnose folgt einer festen Kette: Schnittstelle → Link → IP-Adresse → Nachbartabelle → Route → Gateway → DNS → Zielport → lokaler Dienst → Firewall → Protokolle.

Quellen

- Linux-Manpage – ip
- Linux-Manpage – ip-address
- Linux-Manpage – ip-link
- Linux-Manpage – ip-route
- Linux-Manpage – ip-neighbour
- Linux-Manpage – ss

- Linux-Manpage – ping
- NetworkManager-Dokumentation – nmcli
- NetworkManager-Dokumentation – nmcli-Beispiele
- systemd-Dokumentation – networkctl
- systemd-Dokumentation – resolvectl
- systemd-Dokumentation – journalctl
