Befehle zur Fehlersuche im Netzwerk

IT-Technik Grundlagen

Sebastian Meisel

2. Februar 2023

1 Windows

In Windows muss zwischen den traditionellen DOS- und den moderneren Powershell-Befehlen unterschieden werden. Beide haben ihre Vor- und Nachteile.
Powershell-Befehle bieten normalerweise mehr Optionen, während DOS-Befehle kürzer sind und deshalb häufiger für eine schnelle Fehlerbehebung bevorzugt werden.

1.1 DOS-(CMD)-Befehle

- Erreichbarkeit eines Rechners testen:

```
ping ibb.com
```

Gibt dieser Befehl die folgende Ausgabe zurück, muss überprüft werden, ob dies (nur) an einem Problem mit **DNS** liegt.

```
Ping-Anforderung konnte Host "ibb.com" nicht finden. Überprüfen Sie den Namen, und versuchen Si
```

- Testen der Konnektivität über IPv4:

```
ping 1.1.1.1
```

Man kann auch die IPv6 testen (Google IPv6 DNS-Server):

```
ping -6 2001:4860:4860::8:8:8:8
```

Bei *Link-Local-Adressen* muss die InterfaceNummer mit einem %-Zeichen angehängt werden:

```
ping -6 fe80::9b96:d84:6aa4:60d*%4*
```

Funktioniert dieser Test (und der vorherige mit einer gültigen Url nicht), sollte zunächst eine andere bekannte Url getestet werden. Schlägt dies fehl, liegt ein Problem mit der Namensauflösung (DNS) vor.
Schlägt auch der Test mit der IPv4-Adresse fehl, liegt eine Störung der Internetverbindung vor.
In beiden Fällen sollte man mit folgendem Befehl überprüfen, ob der Netzwerkadapter korrekt konfiguriert ist:

```
ipconfig /all
```

Die Option `/all` gibt zusätzliche Details zur Konfiguration der Schnittstelle aus.

```
Windows-IP-Konfiguration

   Hostname  . . . . . . . . . . . . : WinSrv
   Primäres DNS-Suffix . . . . . . . :
   Knotentyp . . . . . . . . . . . . : Gemischt
   IP-Routing aktiviert  . . . . . . : Nein
   WINS-Proxy aktiviert  . . . . . . : Nein

Ethernet-Adapter Ethernet:

   Verbindungsspezifisches DNS-Suffix:
   Beschreibung. . . . . . . . . . . : Microsoft Hyper-V Network Adapter
   Physische Adresse . . . . . . . . : 00-15-5D-38-01-10
   DHCP aktiviert. . . . . . . . . . : Nein
   Autokonfiguration aktiviert . . . : Ja
   IPv6-Adresse. . . . . . . . . . . : fc00::1(Bevorzugt)
   Verbindungslokale IPv6-Adresse  . : fe80::4745:fa78:726d:1c9f%6(Bevorzugt)
   IPv4-Adresse  . . . . . . . . . . : 172.25.79.254(Bevorzugt)
   Subnetzmaske  . . . . . . . . . . : 255.255.0.0
   Standardgateway . . . . . . . . . : 172.25.64.1
   DHCPv6-IAID . . . . . . . . . . . : 100668765
   DHCPv6-Client-DUID. . . . . . . . : 00-01-00-01-2B-33-32-12-00-15-5D-38-01-10
   DNS-Server  . . . . . . . . . . . : fc00::1
                                       172.25.64.1
                                       1.1.1.1
   NetBIOS über TCP/IP . . . . . . . : Aktiviert
```

Gibt es hier keine Auffälligkeiten, sollte die **Erreichbarkeit des Standardgateways/Routers** geprüft werden:

```
ping 172.25.64.1
```

Gibt es hier eine Fehlermeldung liegt wahrscheinlich ein Problem mit dem Router vor.
Liegt ein **Problem mit der Namensauflösung** vor, sollte man dies zunächst über folgenden Befehl verifizieren:

```
nslookup ibb.com
```

Schlägt dies fehl, kann man testen, ob ein Problem bei den konfigurierten DNS-Servern vorliegt indem man einen anderen DNS-Server anfragt:

```
nslookup ibb.com 8.8.8.8
```

Ist der eigene Router erreichbar, aber bestimmte Seiten im Internet nicht kann man versuchen mit folgendem Befehl herauszufinden, wo das Problem auftritt:

```
tracert ibb.com
```

Dieser Befehl zeigt alle ”Hops”, also alle Router an, über die ein Paket auf dem Weg zu einem Server transportiert wird.

1.1.1 Statistik zu Verbindungen

Ein wichtiges Werkzeug zur Analyse von Netzwerkverbindungen ist auch der `netstat` Befehl, der z. B. Statistiken ausgeben kann, die Aufschluss über Verbindungsprobleme geben können:

```
netstat -s
```

Die Ausgabe beginnt etwa so:

```
IPv4-Statistik

  Empfangene Pakete                  = 2479562
  Empfangene Vorspannfehler          = 0
  Empfangene Adressfehler            = 52425
  Weitergeleitete Datagramme         = 0
  Empfangene unbekannte Protokolle   = 6
  Empfangene verworfene Pakete       = 119909
  Empfangene übermittelte Pakete     = 2438409
  Ausgabeanforderungen               = 4254887
  Verworfene Routingpakete           = 0
  Verworfene Ausgabepakete           = 961320
  Ausgabepakete ohne Routing         = 42
  Reassemblierung erforderlich       = 0
  Reassemblierung erfolgreich        = 0
  Reassemblierung erfolglos          = 0
  Erfolgreiche Datagrammfragmentierung = 0
  Erfolglose Datagrammfragmentierung   = 0
  Erzeugte Fragmente                 = 0
```

Mit demselben Befehl kann man sich Informationen zu geöffnete Verbindungen und offenen Ports anzeigen lassen:

```
netstat -a
```

In der Powershell kann man dann mit dem Cmdlet `Select-String` z. B. die Ports herausfiltern, die für eingehende Verbindungen offen sind, als einen Port "ABHÖREN":

```
netstat -ab | Select-String "ABHÖREN"
```

1.2 Powershell

Der wichtigste Befehl für das Finden von Netzwerkproblemen ist:

```
Test-NetConnection google.com
```

Ohne weitere Optionen entspricht er in etwa dem `ping` Befehl. Allerdings zeigt er mehr Informationen an. Noch mehr erhalte ich mit fogender Option:

```
Test-NetConnection google.com  -InformationLevel Detailed
```

Weitere Optionen ermöglichen:

- Route-Traceing: `-TraceRoute`
- Routing-Diagnose: `-DiagnoseRouting`

Detailierte DNS-Abfragen erhält man mit

```
Resolve-DNSName google.com
```

Zudem gibt es neben dem Befehl `Get-NetIPAddress` den Befehl:

```
Get-NetIPConfiguration
```

Die Ausgabe sieht in etwa so aus:

```
InterfaceAlias       : Ethernet
InterfaceIndex       : 6
InterfaceDescription : Microsoft Hyper-V Network Adapter
NetProfile.Name      : Netzwerk 7
IPv6Address          : fc00::1
IPv4Address          : 172.25.79.254
IPv6DefaultGateway   :
IPv4DefaultGateway   : 172.25.64.1
DNSServer            : fc00::1
                       172.25.64.1
                       1.1.1.1
```

Mit dem folgenden Befehl kann man ähnlich wie mit `netstat` offene Port anzeigen (allerdings nur für TCP):

```
Get-NetTCPConnection
```

2 Linux

Auch unter Linux gibt zwei Arten von Befehlen. Die älteren funktionieren so auf allen auf **Unix** basierenden Systemen und damit auch in MacOS. Diese gelten unter Linux jedoch, als *deprecated*, das bedeutet, dass sie nicht weiter entwickelt werden. Sie funktionieren trotzdem, müssen aber in der Regel nachinstalliert werden.
Unter Debian/Ubuntu ist dies mit folgendem Befehl möglich:

```
sudo apt install net-tools
```

Daneben gibt es moderne Alternativen, die in der Regel vorinstalliert sind und zum Paket `iproute2`.
Trittens gibt es Befehle, die unter allen auf **Unix*** basierenden System funktionieren und bis heute aktuell sind.

3 Allgemeine Befehle

Die Befehle `ping` und `nslookup` funktionieren unter Linux/MacOS/BSD wie für Windows beschrieben.
Lediglich spezielle Optionen, die etwa zum Messen der *MTU* benötigt werden, unterscheiden sich. Hierzu findet man bei Bedarf Anleitungen im Internet.
Zum Befehl `tracert` gibt es mit `traceroute` eine Alternative, die genauso funktioniert.

3.1 Unix-Befehle

Die Unix-Alternative zu `ipconfig` heißt `ifconfig`.
Es gibt auch den Befehl `netstat`, der allerdings unter auf **Unix** basierenden Systemen deutlich mächtiger ist:

```
netstat -t # Alle offenen TCP-Ports
netstat -u # Alle offenen UDP-Ports
netstat -tu # Kombination aus den beiden vorderen
netstat -tul # Alle Port auf denen der Rechner "lauscht", also von außen erreichbar ist.
netstat -tulp # Wie der vorherige aber mit zugehörigen Prozessen
```

Der `route` Befehl gibt die Routingtabelle aus.

3.2 Moderne Linux-Befehle

Der Befehl `ip` ist sowohl der Nachfolger von `ifconfig`:

```
ip address
# oder kurz:
ip a
```

Zeigt die IP-Adressen aller Netzwerkschnittstellen
Aber es ist auch der Nachfolger von `route`:

```
ip route
```

Zeigt die Routingtabelle.
Der Befehl `ss` (Show Socket) ist der Nachfolger von `netstat`:

```
ss -tulp # die Optionen sind die selben, wie oben bei netstat
```