# Host-to-Host-Kommunikation im Rahmen der Netzwerkprogrammierung

R. Grünert
S. Klobe
10. Juni 2021

Zur erfolgreichen Kommunikation zwischen zwei Hosts mit z. B. `echotcp` bzw. `daytimetcp` über das Internet bedarf es mehrerer Konfigurationsschritte. Die notwendigen Schritte werden in diesem Dokument erklärt. Hier eine Zusammenfassung:

- Vor Beginn der Einstellungen sollte ein fester Port für den Serversocket im jeweiligen Serverprogramm angelegt werden. In unserem Fall ist dies der Port `44203`.
- Damit der Server über das Internet erreichbar ist (also vom Client-Programm innerhalb eines anderen Netzwerkes), wird eine öffentliche IPv4-Adresse benötigt.
- Am Gateway des Servernetzwerkes muss mittels Portweiterleitung der eingehende Traffic des Clients auf z.B. Port `44203` an den Server-Host weitergeleitet werden.
- Die Firewall des Server-Hosts muss für den Serverport geöffnet sein.
- Bei Verwendung einer virtuellen Maschine muss dort ebenfalls der Port weitergeleitet werden.

Zusätzlich kann das freie Programm *Wireshark* dabei helfen, Kommunikationsprobleme zu identifizieren.

Abb. 1 zeigt eine vereinfachte Darstellung der Situation.

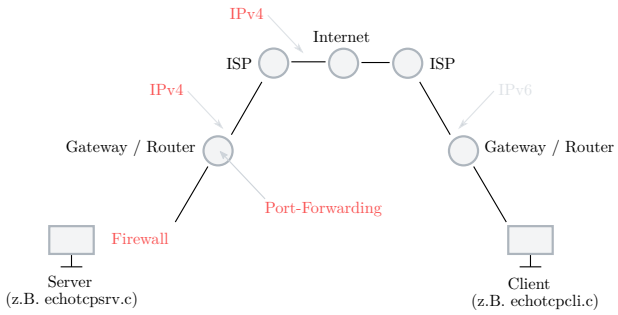


Abbildung 1: Stark vereinfachte Darstellung der Netztopologie

## 1 IPv4 Adresszuweisung

Da im Modul ENP die Netzwerkkommunikation auf Schicht 3 über IPv4 gelehrt wird, benötigt zumindest der Server-Host eine solche öffentliche Adresse.

Da man heutzutage bei den meisten Providern mit *Dual Stack (Lite)* angebunden wird und daher nur eine IPv6-Adresse erhält, ist in diesem Fall eine Anfrage beim Provider nötig, um eine IPv4 Adresse zugewiesen zu bekommen. Im Falle unseres Providers (Vodafone) erfolgte dies durch einen einfachen Anruf.

Gelingt keine IPv4-Zuweisung, bleiben nur Ausweichmöglichkeiten, wie z.B. die VPN-Verbindung beider Netzwerke.

Mithilfe einer Website zur Ermittlung der eigenen öffentlichen IP-Adresse kann im vorher-nachher-Vergleich die Zuweisung überprüft werden [1].

# 2 Port-Forwarding

Aus Sicht des inside-global Netzwerkes gibt es noch keine Möglichkeit, um den Server von anderen Hosts im Heimnetzwerk zu unterscheiden. Der Router sollte daher so konfiguriert werden, dass er den Traffic, der an den Port des Servers gerichtet ist auch an den Server weiterleitet (Port-Forwarding).

## 2.1 Ermittlung der Router-IP-Adresse

Über das Webinterface des Routers kann man das Port-Forwarding einstellen. Um die lokale IP-Adresse des Routers zum Erreichen des Webinterfaces herauszufinden, gibt es betriebssystemabhängig verschiedene Möglichkeiten.

**Router-IP unter Windows** Über den CMD-Befehl `ipconfig` findet man unter dem Eintrag „Default Gateway“bzw „Standardgateway“die gewünschte Routeradresse (Abb. 2).

**Router-IP unter Linux** Unter Linux kann man den Befehl `ip r` oder `ip route` nutzen, um die Routeradresse herauszufinden.

## 2.2 Aufruf des Webinterfaces

Über die oben ermittelte IP-Adresse kann man im Browser das Webinterface des Routers aufrufen. Das Passwort findet man i.d.R. auf dem Router selbst.

## 2.3 Einstellung des Port-Forwardings

Die genaue Einstellung des Port-Forwardings unterscheidet sich natürlich je nach Gerät, im allgemeinen müssen dann aber nur ein Port(bereich) und eine gewünschte Ziel-IP-Adresse angegeben werden. Ein Beispiel ist in Abb. 4 zu sehen.

[1] z. B. mit https://www.whatismyip.com/

Die Ziel-IP-Adresse ist die lokale Adresse des Hosts auf dem das Serverprogramm läuft. Man findet sie unter Windows ebenfalls über den `ipconfig` -Befehl und unter Linux z.B. über `ip a`. Da die Adresszuweisung meist standardmäßig über DHCP stattfindet, sollte darauf geachtet werden, dass sich die beim Port-Forwarding angegebene Adresse ändern und daher eine Neukonfiguration notwendig sein kann.


Abbildung 2: CMD-Ausgabe des ipconfig-Befehls unter Windows


Abbildung 3: Ausgabe des ipconfig-Befehls unter Linux

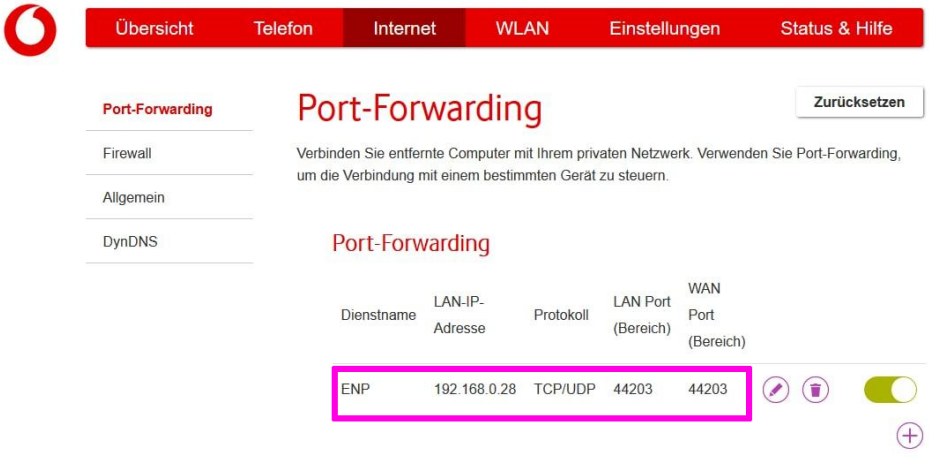


Abbildung 4: Beispiel für Port-Forwarding Ansicht eines Vodafone-Routers

# 3 Firewall

Dem Paket steht nun noch die Firewall des Serverhosts im Weg. Diese blockiert i.d.R. den einkommenden Traffic, der an den Port des Serverprogramms gerichtet ist. Je nach Betriebssystem geschieht das Öffnen des Ports unterschiedlich. Hier wird sich allerdings auf Windows beschränkt, da dies der Regelfall für ENP-Teilnehmer*innen ist. Ein Tutorial für das Öffnen von Ports unter Ubuntu findet man z.B. hier.

## 3.1 Portregeln unter Windows

Über `Systemsteuerung` → `System und Sicherheit` → `Windows Defender Firewall` → `Erweiterte Einstellungen` gelangt man zu den Portregeln. Mit einem Rechtsklick auf `Eingehende Regeln` und danach auf `Neue Regel` kann man den Port des Servers freigeben. Abb. 5 bis 6 zeigen eine Beispielkonfiguration für den Serverport `44203`.

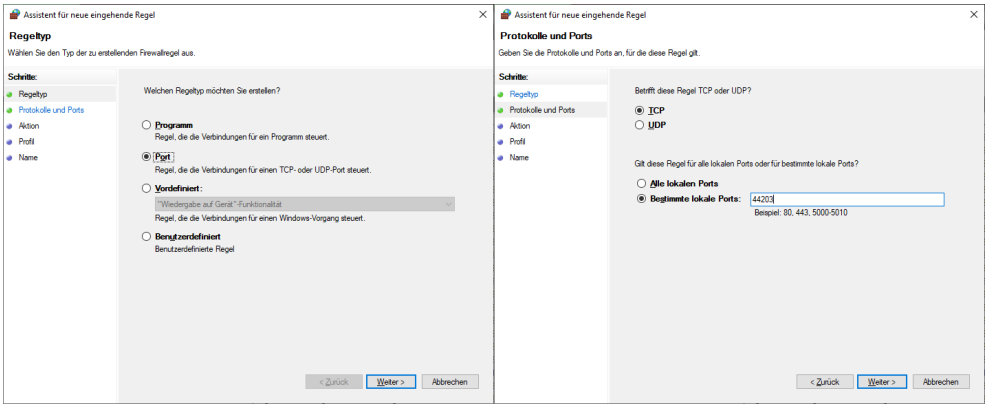


Abbildung 5

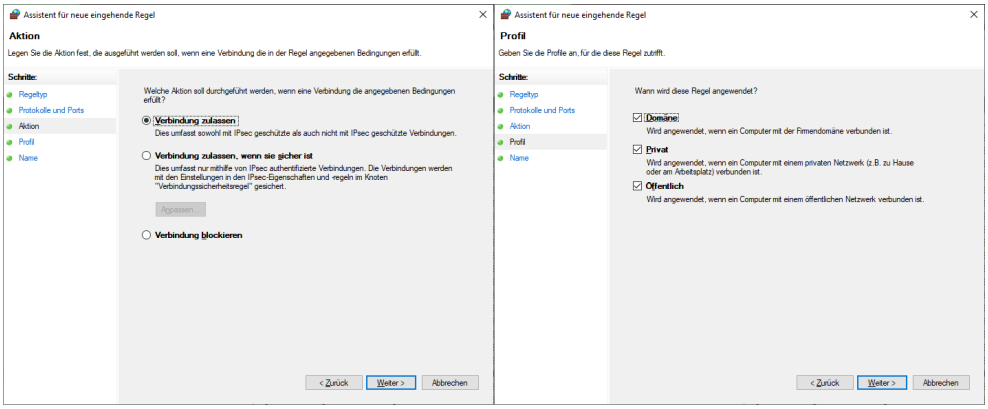


Abbildung 6

# 4 Virtualbox

Da Teilnehmer*innen des Moduls ENP meist eine virtuelle Maschine (Oracle VirtualBox) auf Windows nutzen, um unter Linux zu arbeiten, benötigt man in diesem Fall zusätzliche Konfigurationen unter VirtualBox.

Als erstes sollte sichergestellt sein, dass VirtualBox von der Windows-Firewall ins Netzwerk gelassen wird. Dies kann ebenfalls über `Systemsteuerung` → `System und Sicherheit` → `Windows Defender Firewall` geprüft werden. Alternativ kann man auch in der virtuellen Maschine testen, ob man z.B. mit einem Browser eine Website aufrufen kann.

Weiterhin muss auch hier eine Portweiterleitung eingerichtet werden.

## 4.1 Port-Forwarding

Die Weiterleitungseinstellungen erreicht man über den `Netzwerk`-Tab unter den Einstellungen der Virtuellen Maschine. Eine Beispielkonfiguration ist in Abb. 7 zu sehen.

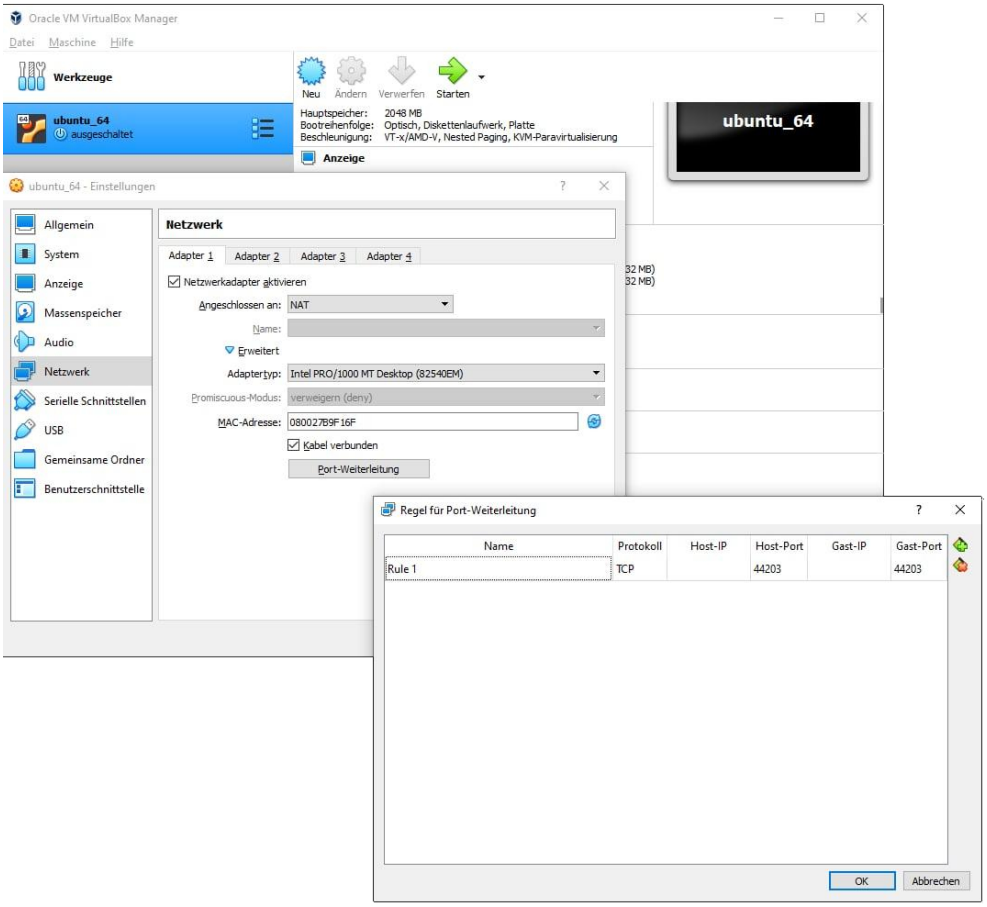


Abbildung 7: Einstellung der Portweiterleitung unter VirtualBox

**Hinweis**: Möglicherweise bedarf es einem Neustart von Windows oder VirtualBox.

# 5 Test

Nach den oben beschriebenen Schritten kann der erste Test gestartet werden. Es wird angenommen, dass das Clientprogramm (cli) so geschrieben ist, dass man über die Aufrufargumente IP-Adresse und Port des Servers angeben kann. Der Aufruf für unser Beispiel funktioniert dann z.B. so: `cli 172.217.14.195 44203`.

An Client- und Serverprogramm sollten keine weiteren Modifikationen notwendig sein. Der Listen-Socket des Servers sollte am einfachsten mit der Adresse des Macros `INADDR_ANY` gebunden werden.

# 6 Troubleshooting

Mithilfe des Programmes *Wireshark* kann der Netzwerkverkehr der Netzwerkschnittstellen sowohl beim Client als auch beim Server in Echtzeit überwacht werden. Mit dem Filterbefehl `tcp.port == 44203` können nur die relevanten Pakete angezeigt werden. Bei Verbindungsaufbau sowie Datenübertragung (z.B. Senden der Zeichenkette bei `echotcp`) erscheinen dann dort die zugehörigen Pakete.

Abb. 8 sowie Abb. 8 zeigen Wireshark-Traces aus Client- und Serversicht, bei denen die Firewall des Server-Hosts den Client- Traffic nicht hindurchlässt. Der Client schickt dann kontinuierliche Anfragen zum Verbindungsaufbau (SYN).

Ist das Port-Forwarding am Router nicht korrekt konfiguriert, zeigt der Trace des Server-Hosts (Abb. 9) keine Pakete an.

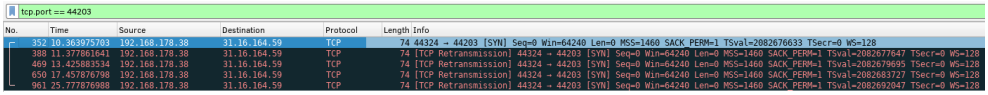


Abbildung 8: Wireshark Trace am Clientrechner bei Firewall-Problemen

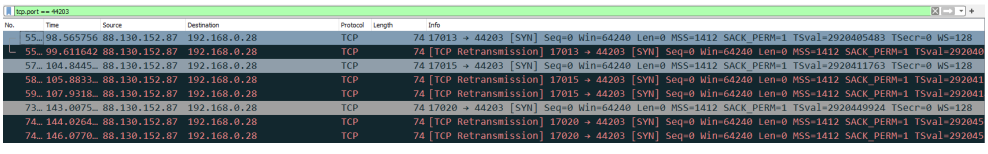


Abbildung 9: Wireshark Trace am Serverrechner bei Firewall-Problemen

Für den Fall, dass die Portfreigabe-Einstellungen unter Windows keine Wirkung haben, kann auch die gesamte Firewall deaktiviert werden (Abb. 10).

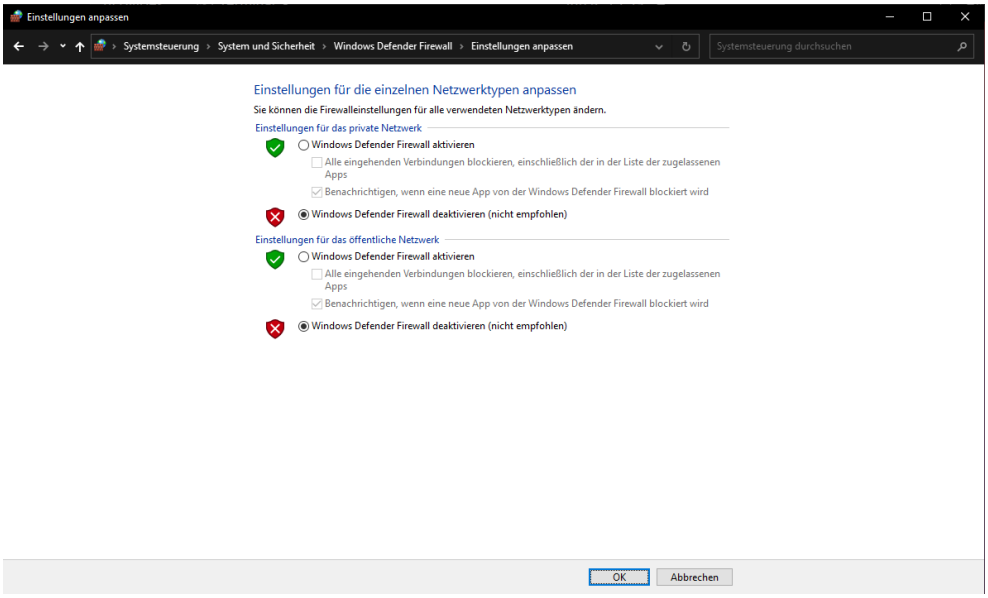


Abbildung 10

# 7 Verbesserungen / Hinweise

Unsere Testumgebung bestand aus einem Host mit nativem Linux (Client) und einem Host mit Linux unter VM auf Windows (Server). Die anderen Kombinationen, sprich Linux-Client/Windows-Server, Linux-Client/Linux-Server, sowie Windows-Client/ Windows-Server, wurden daher noch nicht getestet. Möglicherweise sind unter diesen Bedingungen weitere Einstellungen notwendig (z.B. an der Firewall auf dem Clientrechner).