



Übung 07: IP Forwarding

Vorbereitung: Installation von Docker in Linux

Sie benötigen ein funktionsfähiges Linux, idealerweise die VM aus Betriebssysteme.

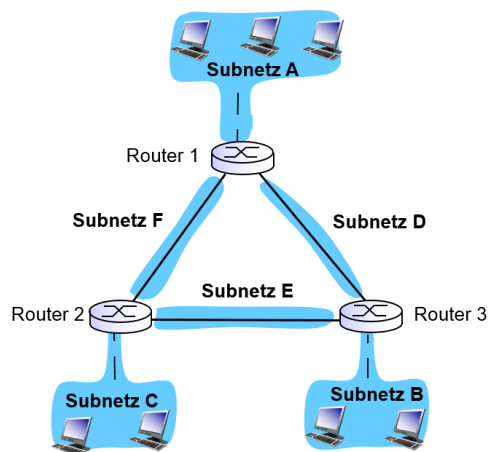
- Legen Sie die Dateien Dockerfile und docker-compose.yaml in ein beliebiges Verzeichnis unter Linux. Beide Dateien sind im Learning Campus als zip-Archiv verfügbar:
<https://syncandshare.lrz.de/dl/fiJg4M35No9hjBwpmDahdm5U/Uebung07-docker.zip>
- Installieren Sie *Docker*¹ und *Docker-Compose* (Dauer: gut 5 Minuten)

```
sudo apt update  
sudo apt install -y docker.io docker-compose
```

Aufgabe 1: Aufteilung von IP Adressen auf Subnetze

Gegeben ist ein Netzwerk mit 6 Subnetzen. Weisen Sie jedem Subnetz IP Adressen zu. Beachten Sie die folgenden Bedingungen:

- Alle Adressen müssen im Bereich 214.97.254.0/23 liegen
- Subnetz A muss mindestens 250 Hosts unterstützen
- Subnetz B muss mindestens 120 Hosts unterstützen
- Subnetz C muss mindestens 53 Hosts unterstützen
- Subnetze D, E und F benötigen jeweils mindestens 2 IP Adressen (z.B. /30-Subnetz).

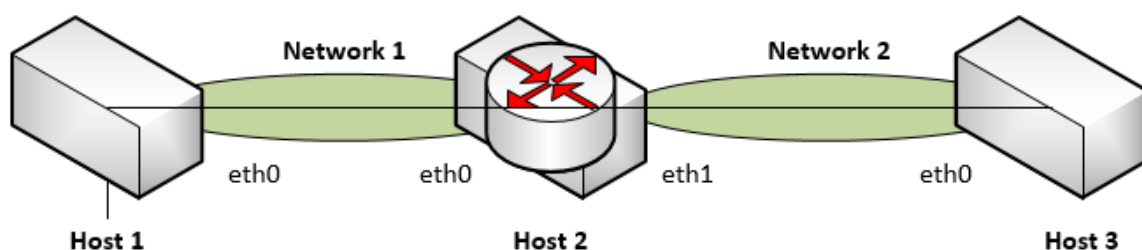


Subnetz	CIDR Notation (a.b.c.d/x)
Subnetz A	
Subnetz B	
Subnetz C	
Subnetz D	
Subnetz E	
Subnetz F	

Aufgabe 2: IP Forwarding unter Linux

Sie konfigurieren ein vorgegebenes *virtuelles* Netz selbst. Die Hosts im Netzwerk sind *Docker*-Container. Durch docker-compose werden 3 virtuelle Hosts erzeugt, die durch Ethernet-LANs Network1 und Network2 verbunden sind, siehe Abbildung.

Ziel: Host1 und Host3 sollen sich gegenseitig erreichen können, obwohl sie in unterschiedlichen Subnetzen sind. Host2 ist ein Router, hat 2 Ethernet Interfaces und muss Pakete weiterleiten.



¹ Mit Docker Container lassen sich „extrem leichtgewichtige, modulare virtuelle Maschinen“ realisieren, die alle unter einem Betriebssystem laufen. Wir missbrauchen das, um ein kleines Netzwerk nachzubilden. Als Beschleunigung sollte in VirtualBox VT-x/AMD-V aktiviert sein.

- a) Nach der Vorbereitung: Erzeugen Sie das Netzwerk, indem Sie den folgenden Befehl im Verzeichnis mit den Dateien Dockerfile und docker-compose.yaml ausführen:
`sudo docker-compose up` (Dauer: ca. 3 Minuten)
- b) Öffnen Sie für jeden der 3 Hosts ein eigenes Terminal:
Host1: `sudo docker exec -it host1 /bin/bash`
Host2: `sudo docker exec -it host2 /bin/bash`
Host3: `sudo docker exec -it host3 /bin/bash`
- c) Weisen Sie den 3 Hosts **sinnvolle** IP Adressen zu, tragen Sie die Adressen in die Zeichnung ein!
Anforderung: Öffentliche IPs verwenden, keine privaten IPs!
- d) Konfigurieren Sie nun auf allen Hosts diese IP Adressen!
- Tool: `ip [addr add]`, Internet-Recherche!
- Überprüfen Sie mit `ping`, ob sich alle(!) Hosts gegenseitig erreichen können.
- e) Warum funktioniert der `ping` von `host1` zu `host3` nicht? Um Ihre Antwort zu begründen, schauen Sie sich die Routingtabelle von `Host1` an.
Hinweis: `ip route`
- f) Fügen Sie am `Host1` manuell eine **statische Route** zu `Host3` hinzu: Teilen sie `Host1` mit, dass er `Host3` über `Host2` erreichen kann. Anschließend erneut die Routingtabelle anschauen. Funktioniert der `ping` nun?
- Hinweis: `ip route add`. Internet-Recherche!
- Damit `Host2` empfangene Pakete weiterleitet, muss Forwarding explizit aktiviert werden:
`sysctl -w net.ipv4.ip_forward=1`
- g) Korrigieren Sie das Problem aus f).
- h) Das Tool `netcat` wird oft als „Schweizer Taschenmesser“ von TCP/IP bezeichnet. Es erlaubt Daten über TCP und UDP Sockets zu versenden.
• Starten Sie auf `Host3` den „Server“ durch `netcat -l -p 9000`.
• Auf `Host1` öffnen Sie eine Verbindung zum „Server“ mit `netcat <dstip> 9000`. Versenden Sie Textnachrichten („Chat Messages“) zwischen den beiden Hosts.
- i) Schauen Sie sich die ARP-Tabelle auf `Host1` mit dem Kommando `arp -n` an. Welche Einträge stehen in der ARP Tabelle? Warum existiert dort ein Eintrag für `Host2`, warum keiner für `Host3`?
- j) Löschen Sie den Eintrag für `Host2` aus der ARP-Tabelle von `Host1` mit dem Kommando `arp -d <ip addr>`. Überzeugen Sie sich, dass der ARP-Tabelle nun leer ist.
- k) Senden Sie nun erneut einen `ping` an `Host3`. Wurde der ARP-Eintrag für `Host 2` erneut hinzugefügt?
- l) Zum Aufräumen: `sudo docker-compose down`