



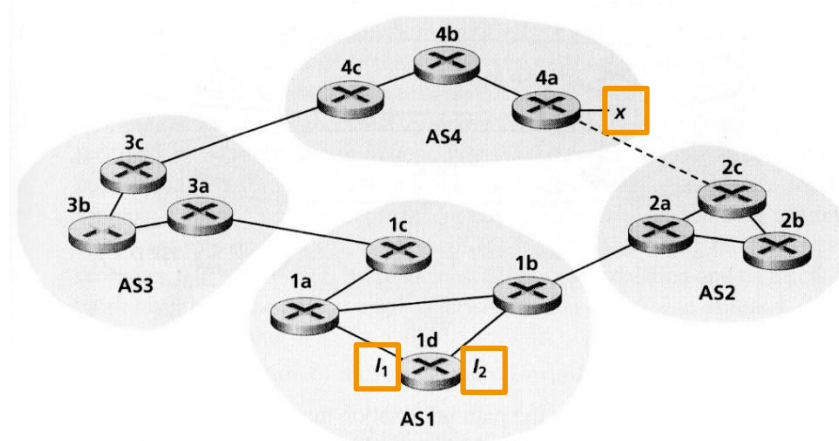
## Übung 08: Routing, IPv6

### Vorbereitung: Docker-Netzwerk

- Docker muss installiert sein, Details siehe Übung 07.
- Extrahieren Sie alle Dateien aus dem bereitgestellten zip-Archiv in ein beliebiges Verzeichnis unter Linux:  
[https://syncandshare.lrz.de/dl/fiUCogEjFxi66nGvNryeRkX6/RN\\_Uebung08-sources.zip](https://syncandshare.lrz.de/dl/fiUCogEjFxi66nGvNryeRkX6/RN_Uebung08-sources.zip)
- Führen Sie dann `sudo docker-compose up` in diesem Verzeichnis aus.
- *Nur bei Problem:* `sudo docker stop $(sudo docker ps -aq); sudo docker system prune`

### Aufgabe 1: Hierarchisches Routing

Gegeben ist ein „kleines Internet“ mit 4 **Autonomen Systemen (ASe)**. Alle ASe setzen ein *Intradomain*-Routingprotokoll ein, um die kürzesten Wege zu den Next-Hops zu bestimmen, die das *Interdomain*-Routingprotokoll BGP ermittelt. Zunächst sei keine Verbindung zwischen AS2 und AS4 vorhanden. *Alle* Links haben die gleichen Kosten!



- Der Router 1d lernt die Existenz des Prefix **x** und erstellt einen Eintrag in seiner Routingtabelle. Alle Linkkosten sind gleich und der Link zwischen 2c und 4a existiert nicht. Verwendet 1d als Ausgangsinterface **i<sub>1</sub>** oder **i<sub>2</sub>** zum Prefix **x**? Erklären Sie das Zusammenspiel zwischen Inter- und Intradomain Routing in AS1!
- Nun sei der Link zwischen 2c und 4a vorhanden. Wird für den Prefix **x** nun **i<sub>1</sub>** oder **i<sub>2</sub>** gewählt?
- Wie sieht es aus, falls auf dem Weg zwischen AS2 und AS4 ein weiteres AS, nämlich AS5 liegt (in der Abbildung nicht eingezeichnet)? Ist **i<sub>1</sub>** oder **i<sub>2</sub>** das Ausgangsinterface? Warum?
- Angenommen AS3 ist Kunde sowohl von AS4 als auch von AS1. AS3 muss also AS1 und AS4 Geld für den Datenverkehr auf den entsprechenden Links zahlen. Leitet AS3 Erreichbarkeitsinformation bezüglich AS4 an AS1 weiter?
- Jedes AS ist durch eine AS-Nummer gekennzeichnet, ähnlich wie IP Adressen bei Internet Hosts. Welche AS Nummer hat der Provider der TH Rosenheim?  
*Hinweis:* Beispiel IP aus TH-Netz 141.60.160.196, <https://apps.db.ripe.net/search/>
- Viele Autonome Systeme (ASe) verbinden sich über einen **Internet Exchange Point (IXP)**. Wie viele ASe benutzen den DE-CIX? Tauschen alle dort stationierten ASe auch wirklich Routingdaten aus, z.B. Verizon Deutschland GmbH?

*Hinweis:* <https://www.de-cix.net/de/locations/germany/frankfurt/connected-networks>

### Aufgabe 2: IPv6

- a) Was ist die kürzest mögliche, gültige Schreibweise für die folgende IPv6 Adresse:  
2001:0019:ed00:001a:0000:0000:0000:0031
- b) Ihr Provider weist den IPv6 Adressblock 2a02:810d:1300:ae8::/62 zu. Typischerweise haben IPv6 Subnetze die Größe /64.
- In wie viele /64-Blöcke können Sie den /62 Block maximal zerlegen?
  - Wie lauten diese?
- c) Bei IPv6 hat jedes Interface auch ohne Konfiguration eine *Link-Local Adresse*. Solche Adressen müssen im Bereich fe80::/10 liegen und können nur verwendet werden, um Hosts im gleichen Subnetz zu erreichen. Auf einem Linux-PC erhalten Sie mit `ifconfig` die folgende Ausgabe:

```
root@rechnernetze:~# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe41:ccfe prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:41:cc:fe txqueuelen 1000 (Ethernet)
    RX packets 52015 bytes 61428076 (61.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11937 bytes 753479 (753.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

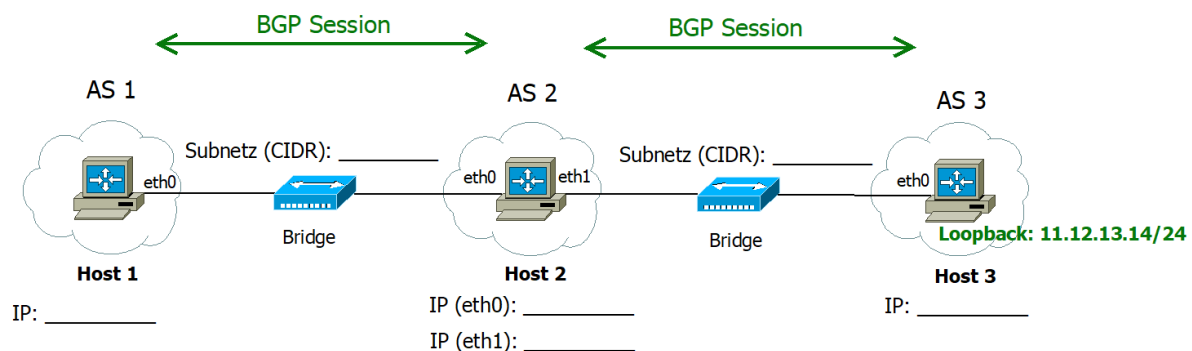
Vergleichen Sie die MAC Adresse und die Link-Local Adresse! Was fällt auf?

Hinweis: <http://ben.akrin.com/?p=1347>

### Aufgabe 3: BGP Routing / Praxis

Das virtuelle Docker-Netz besteht aus 3 Containern, 2 davon sind jeweils über eine Software Bridge bzw. Switch verbunden. Jeder Container emuliert einen echten Internet-Router auf Basis von Linux. Jeder Host ist ein eigenes Autonomes System (AS).

Ziel: Host3 hat auf dem Loopback Interface `lo` die IP 11.12.13.14. Die Routingtabelle von Host1 soll das Subnetz 11.12.13.0/24, das zu Host3 gehört, **automatisch** kennenlernen, **ohne statischen Routen** wie in der letzten Übung. Dazu müssen Host1 und Host2 sowie Host2 und Host3 per BGP Routinginformation austauschen.



- a) Öffnen Sie für jeden der 3 Hosts ein 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`
- b) Finden Sie heraus, welche IP Adressen die Container und Subnetze verwenden. Ergänzen Sie die Zeichnung mit den IPs und Subnetzen (in CIDR-Notation).
- c) Konfigurieren Sie auf Host3 die IP 11.12.13.1/24 für das Loopback Interface `lo`:
- `ip addr add 11.12.13.1/24 dev lo`  
Haben Host1 und Host2 eine Route zu diesem Subnetz? (`ip route`)

- d) Konfigurieren Sie nun die beiden BGP Sessions: Host2 nimmt an beiden BGP Sessions teil. Eine passende Konfigurationsdatei ist unter `/etc/bird/bird.conf` bereits in jedem Container vorhanden<sup>1</sup>. Nur noch die blauen Felder, siehe rechts, müssen korrekt gesetzt werden.

```
router id <beliebige IP des Hosts>;
protocol device {
}
protocol kernel {
    metric 64;
    learn;
    import none;
    export all;
}
protocol bgp <session name> {
    local as <AS number>;
    neighbor <neighbour ip> as <AS number>;
    import all;
    export all;
}
protocol direct {
    interface "lo";
}
```

- e) Starten Sie auf alle Hosts das Routing mit `service bird restart`
- Hat die Routingtabelle von Host1, Host2, Host3 einen Eintrag für `11.12.13.0/24`?
  - Können Sie von Host1 bzw. Host2 die IP `11.12.13.1` pingen? Erklären, aber erkannte Probleme nicht beheben!

- f) Zeichnen Sie mit Wireshark BGP Pakete zwischen Host1 und Host2 auf. Werden regelmäßig BGP Nachrichten ausgetauscht? Wie häufig?  
*Hinweis:* 1-2 Minuten warten.

- Finden Sie mit `sudo docker network ls` heraus, welche *NETWORK ID* die Bridge mit dem Namen `br*_net1` hat (== Bridge zwischen Host1 und Host2).
- Starten Sie Wireshark als root mit `sudo wireshark` auf dem Linux Gastsystem.
- Starten Sie eine Aufzeichnung auf dem passenden Interface `br-<NETWORK ID>` und **lassen diese Aufzeichnung laufen**.

- g) Host2: Deaktivieren Sie mit `ip link set dev eth1 down` das Interface `eth1`. Was sehen Sie in Wireshark und in der Routingtabelle von Host1? **Lassen Sie die Aufzeichnung laufen**.

- h) Reaktivieren Sie das Interface mit `ip link set dev eth1 up`. Was steht in dem versendeten Routingpaket?
- Welche 3 *Pfadattribute* sehen Sie?
  - Wie lautet die *Network Layer Reachability Information*?
  - Wie sieht nun die Routingtabelle von Host1 aus?

- i) Zum Aufräumen: `sudo docker-compose down` und Wireshark stoppen.

---

<sup>1</sup> Editoren: nano oder vim verwenden.