Übung 08: Routing, IPv4 Fragmentierung

Vorbereitung: Docker-Netzwerk

Docker sollte aus Übung 07 bereits installiert sein. Zum Aufsetzen des Netzes:

- Kopieren Sie die bereitgestellten Dateien `Dockerfile, docker-compose.yaml, bird.conf` in ein beliebiges Verzeichnis unter Linux.
- Empfehlung: Alte Containernetze löschen mit `sudo docker network prune`
- Führen Sie dann `sudo docker-compose up` in diesem Verzeichnis aus.

Aufgabe 1: IPv4 Fragmentierung in Wireshark

Im vorgegebenen Wireshark Trace führt der Rechner mit der IP 192.168.1.102 ein `Traceroute`-Kommandos zum Rechner mit der IP 128.59.23.100 aus.

Betrachten Sie zunächst das Paket #8:

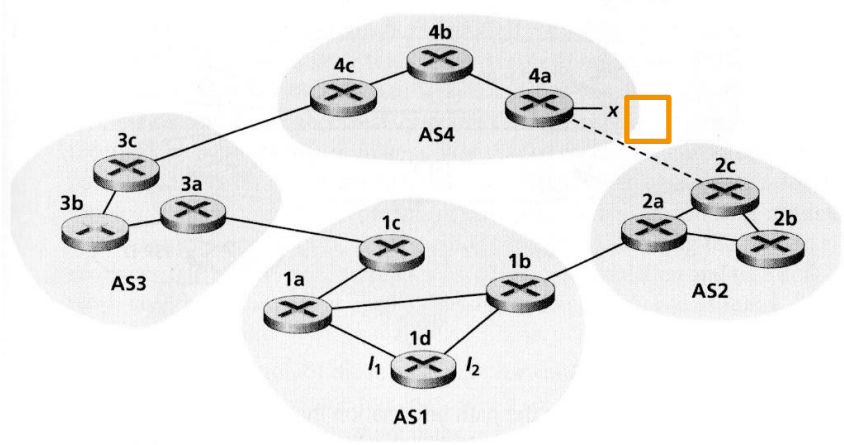
a) Welchen Wert hat das „*Protocol*"-Feld im IPv4 Header dieses Pakets?
b) Wie viele Bytes hat der IP Header? Wie groß ist die Nutzlast / Payload?
c) Wurde dieses Paket fragmentiert? Begründen Sie Ihre Antwort!
d) Betrachten Sie „Echo (Ping) Request" Pakete unmittelbar nach Paket #8, die 192.168.1.102 sendet. Wodurch unterscheiden sich aufeinanderfolgende „*Echo (Ping) Request Pakete*" im IP Header?

Betrachten Sie nun das Paket #92:

e) Warum ist es das erste Fragment eines IP Datagramms? Wie groß ist das Fragment?
f) Suchen Sie das dazugehörige zweite Fragment. Welche Info im IP Header weist darauf hin, dass es nicht das erste Fragment ist? Gibt es weitere Fragmente? Woran sieht man das?

Aufgabe 2: Hierarchisches Routing

Gegeben ist ein „kleines Internet" mit 4 Autonomen Systemen (AS). AS2 und AS3 verwenden OSPF als *Intradomain*-Routingprotokoll. AS1 und AS4 verwenden OSPF als *Intradomain*-Routingprotokoll. Als *Interdomain* Routingprotokoll wird BGP eingesetzt. Zunächst sei keine Verbindung zwischen AS2 und AS4 vorhanden. *Alle* Links haben die gleichen Kosten!



a) Die globale Erreichbarkeit eines IP Präfix wird mit BGP angekündigt. Welche 2 wichtigen „Attribute" sind außer dem Präfix in einer BGP Route enthalten?

b) Nachdem der Router 1d die Existenz des Prefix *x* erfährt, erstellt er einen Eintrag ***(x,I)* = *(Zielpräfix, Ausgangsinterface)*** in seine Forwarding-Tabelle. Angenommen alle Linkkosten sind gleich und der Link zwischen 2c und 4a existiert nicht. Ist ***I*** gleich I_1 oder gleich I_2? Erklären Sie das Zusammenspiel zwischen Interdomain und Intradomain Routing in AS1!

c) Nun sei der Link zwischen 2c und 4a vorhanden. Wird für den Prefix **x** nun l_1 oder l_2 gewählt?

d) Wie sieht es aus, falls auf dem Weg zwischen AS2 und AS4 ein weiteres AS (AS5) liegt (in der Abbildung nicht eingezeichnet)? Ist l dann gleich l_1 oder gleich l_2? Warum?

e) 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?

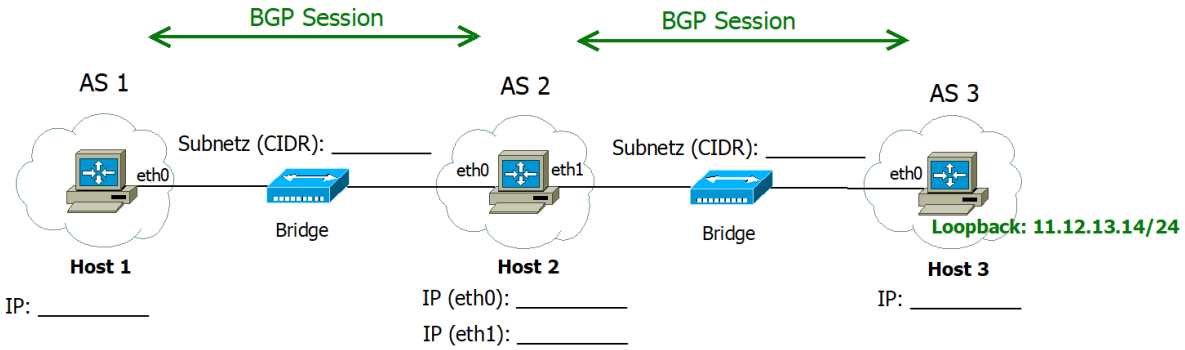
f) Jedes Autonome System (AS) ist durch eine AS-Nummer gekennzeichnet, ähnlich wie IP Adressen bei Internet Hosts. Welche AS Nummer hat der Provider der TH Rosenheim?
Hinweis: https://apps.db.ripe.net/search/

g) Viele Autonome Systeme (ASe) verbinden sich über einen ***Internet Exchange Point (IXP).*** Der größte davon ist in Deutschland der DE-CIX Knoten in Frankfurt. Wie viele ASe benutzen den DE-CIX? Tauschen alle dort stationierten ASe auch wirklich Routingdaten aus, z.B. Verizon Deutschland GmbH?
Tipp: https://www.de-cix.net/de/locations/germany/frankfurt/connected-networks

Aufgabe 3: BGP Routing mit Linux

Sie verwenden wieder ein Docker-Netzwerk. Es gibt 3 Container, 2 davon sind je über eine Bridge (== Hub) 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 bei `Host3` liegt, ***automatisch*** kennenlernen, ***ohne statischen Routen*** wie in der letzten Übung. Dazu müssen `Host1` und `Host2` sowie `Host2` und `Host3` per BGP Routinginformation austauschen.



a) Führen Sie die Vorbereitung durch!

b) 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`

c) Finden Sie heraus, welche IP Adressen die Container und Subnetze verwenden. Ergänzen Sie die Zeichnung mit den IPs und Subnetzen (in CIDR-Notation).

d) Konfigurieren Sie auf `Host3` die folgende IP für das Loopback Interface `lo`:
`ip addr add 11.12.13.14/24 dev lo`
Schauen Sie die Routingtabellen (`route -n`) auf allen Hosts an. Welche Hosts kennen das Subnetz 11.12.13.0/24?

e) Konfigurieren Sie die beiden BGP Sessions. `Host2` nimmt an 2 BGP Sessions teil. Eine passende Konfigurationsdatei ist unter `/etc/bird/bird.conf` bereits in jedem Container vorhanden[1]. Nur noch die blauen Felder, siehe rechts, müssen korrekt gesetzt werden.

```
router id <ip von lo:0>;
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";
}
```

f) Starten Sie 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 11.12.13.14 pingen? Erklären!

g) Zeichnen Sie mit Wireshark BGP Pakete zwischen `Host1` und `Host2` auf.
- Finden Sie mit `sudo docker network ls` heraus, welche *NETWORK ID* die Bridge mit dem Namen *uebung08_net1* hat (== Bridge zwischen `Host1` und `Host2`).
- Starten Sie Wireshark als `root` mit `sudo wireshark`.
- Starten Sie eine Aufzeichnung auf dem passenden Interface `br-<NETWORK ID>` und lassen diese Aufzeichnung laufen.

h) Werden regelmäßige BGP Pakete ausgetauscht? Wie häufig? Zeichnen Sie mit Wireshark BGP Pakete zwischen `Host1` und `Host2` auf!

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

j) Lassen Sie die Aufzeichnung laufen. 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?

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

[1] Editoren: nano oder vim verwenden.