Routing Lab

34334 Avancerede Datanet

Daniel Brasholt s214675

4. november 2022

Indhold

Del 1 RIP

Q1.

Router 1

Router 1 skal annoncere netværkene `10.1.2.0/24`, `10.1.3.0/24`, `10.1.4.0/24`, `192.168.1.0/24` og `192.168.10.1/32`.

Router 2

Router 2 skal annoncere netværkene `10.1.2.0/24`, `10.2.4.0/24`, `192.168.2.0/24` og `192.168.20.2./32`.

Router 3

Router 3 skal annoncere netværkene `10.1.3.0/24`, `10.3.4.0/24`, `192.168.3.0/24` og `192.168.30.3/32`.

Router 4

Router 4 skal annoncere netværkene `10.1.4.0/24`, `10.2.4.0/24`, `10.3.4.0/24`, `192.168.4.0/24` og `192.168.40.4/32`.

Q2.

```
▸ Frame 10: 226 bytes on wire (1808 bits), 226 bytes captured (1808 bits) on interface r12, id 0
▸ Ethernet II, Src: 36:37:58:3c:a3:ff (36:37:58:3c:a3:ff), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
▸ Internet Protocol Version 4, Src: 10.1.2.1, Dst: 224.0.0.9
▸ User Datagram Protocol, Src Port: 520, Dst Port: 520
▾ Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
  ▸ IP Address: 10.1.3.0, Metric: 1
  ▸ IP Address: 10.1.4.0, Metric: 1
  ▸ IP Address: 10.3.4.0, Metric: 2
  ▸ IP Address: 192.168.1.0, Metric: 1
  ▸ IP Address: 192.168.3.0, Metric: 2
  ▸ IP Address: 192.168.4.0, Metric: 2
  ▸ IP Address: 192.168.10.1, Metric: 1
  ▸ IP Address: 192.168.30.3, Metric: 2
  ▸ IP Address: 192.168.40.4, Metric: 2
```

Figur 1: *RIP-response fra router 1*

På ovenstående figur 1 kan de netværk, som router 1 kan nå, ses. Derudover kan det ses hvor mange hop der er til hvert netværk.

Router 1 har selv forbindelse til netværkene `10.1.3.0/24` og `10.1.4.0/24`. Derfor optræder de i RIP-response med en metric på 1. Derudover har router 1 fået at vide, at den kan nå netværk `10.3.4.0/24` med et enkelt hop, hvorfor denne har metric på 2. Ligeledes kan de interne netværk `192.168.3.0/24` og `192.168.4.0/24` nås med et enkelt hop. Router 1 har selv forbindelse til det interne `192.168.1.0/24`. Til sidst kan router 1 også nå loopback-netværk med henholdsvis 1 og 2 hop af samme årsag som de andre netværk.

Q3.

```
traceroute to 192.168.3.13 (192.168.3.13), 30 hops max, 60 byte packets
 1  192.168.1.1  0.020 ms  0.004 ms  0.003 ms
 2  10.1.3.3  0.009 ms  0.005 ms  0.004 ms
 3  192.168.3.13  0.010 ms  0.006 ms  0.006 ms
```

Figur 2: *Resultat fra* ***traceroute***

På ovenstående figur 2 kan det ses, at et ping til destinationen `192.168.3.13` fra host 1 går gennem 3 hop. Det første hop er routerens interne interface, `h1`, med IP-adressen `192.168.1.1`. Dernæst bliver pakken sendt til router 3. Det interface, den bliver sendt over, er `r13` med IP-adresse `10.1.3.3`. Det sidste hop er selve destinationen.

Q4.

```
traceroute to 192.168.3.13 (192.168.3.13), 30 hops max, 60 byte packets
 1  192.168.1.1  0.019 ms  0.003 ms  0.003 ms
 2  10.1.4.4  0.018 ms  0.007 ms  0.006 ms
 3  10.3.4.3  0.012 ms  0.007 ms  0.008 ms
 4  192.168.3.13  0.013 ms  0.008 ms  0.008 ms
```

Figur 3: *Resultat fra* ***traceroute***

Ovenstående figur 3 viser, at ruten til `192.168.3.13` nu er ændret til at have 4 hop i stedet for 3. De 4 hop er først gennem router 1, dernæst til router 4, derfra til router 3 over interface `r34` og til sidst til den ønskede host. Router 1 har da fundet en ny vej til router 3 efter interface `r13` blev lagt ned.

Q5.

Da outputtet fra `fping`-kommandoen er langt, er det ikke taget med. Det kunne dog aflæses, at forbindelsen blev tabt efter pakke 4 og genoprettet ved pakke 28. Der er da blevet afsendt 24 pakker før forbindelsen blev genprettet. Da pakkerne bliver sendt med et sekunds mellemrum, har det cirka taget 24 sekunder at genoprette forbindelsen.

Q6.

Ingen pakker går tabt, når forbindelsen bliver genoprettet. Dette er da den rute, pakken bliver sendt over, ikke stopper med at virke - der er dog en hurtigere vej, som bliver taget i brug, når RIP-protokollen finder forbindelsen over `r13` igen.

Q7.

Denne gang tog det væsenligt længere tid før forbindelsen blev renoprettet i forhold til Q5. Her blev 175 pakker tabt før ICMP-ping blev besvaret. Dette er nok eftersom pakken, i modsætning til Q5, bliver sendt ud uden at routeren ved, at den er forsvundet - den går blot tabt i switchen.

Q8.

Forbindelsen bliver genoprettet så snart det andet kabel fjernes. En `traceroute`-kommando er vist på figur 4. Denne viser, at forbindelsen fra host 4 til host 1 går gennem router 3. Når andet kabel bliver fjernet, vil router 4 med det samme sende pakken en anden vej, da routeren ved, at den vej ikke længere virker.

```
traceroute to 192.168.1.11 (192.168.1.11), 30 hops max, 60 byte packets
 1  192.168.4.4  0.022 ms  0.004 ms  0.003 ms
 2  10.3.4.3  0.008 ms  0.004 ms  0.004 ms
 3  10.1.3.1  0.010 ms  0.005 ms  0.005 ms
 4  192.168.1.11  0.010 ms  0.006 ms  0.006 ms
```

Figur 4: *Resultat fra* ***traceroute***

Del 2 OSPF

Q9.

```
10.1.2.0/24 dev r12 proto kernel scope link src 10.1.2.1
10.1.3.0/24 dev r13 proto kernel scope link src 10.1.3.1
10.1.4.0/24 dev r14 proto kernel scope link src 10.1.4.1
10.2.4.0/24 proto zebra metric 20
        nexthop via 10.1.2.2 dev r12 weight 1
        nexthop via 10.1.4.4 dev r14 weight 1
10.3.4.0/24 proto zebra metric 20
        nexthop via 10.1.3.3 dev r13 weight 1
        nexthop via 10.1.4.4 dev r14 weight 1
192.168.1.0/24 dev h1 proto kernel scope link src 192.168.1.1
192.168.2.0/24 via 10.1.2.2 dev r12 proto zebra metric 20
192.168.3.0/24 via 10.1.3.3 dev r13 proto zebra metric 20
192.168.4.0/24 via 10.1.4.4 dev r14 proto zebra metric 20
192.168.20.2 via 10.1.2.2 dev r12 proto zebra metric 20
192.168.30.1 via 10.1.3.3 dev r13 proto zebra metric 20
192.168.30.3 via 10.1.3.3 dev r13 proto zebra metric 20
192.168.40.1 via 10.1.4.4 dev r14 proto zebra metric 20
192.168.40.4 via 10.1.4.4 dev r14 proto zebra metric 20
```

Figur 5: *Resultat fra* ***ip route*** *eksekveret fra router 1*

På ovenstående figur 5 kan routing table fra router 1 ses. Router 1 har direkte forbindelse til 3 netværk som forventet, hvilket fremgår af de første 3 linjer. Derudover er der to forskellige ruter hver af de to netværk `10.2.4.0/24` og `10.3.4.0/24`. Til sidst er der forbindelse til de 4 interne netværk og diverse loopbacks.

Q10.

```
traceroute to 192.168.3.13 (192.168.3.13), 30 hops max, 60 byte packets
 1  192.168.1.1  0.026 ms  0.004 ms  0.003 ms
 2  10.1.3.3  0.010 ms  0.005 ms  0.004 ms
 3  192.168.3.13  0.012 ms  0.006 ms  0.005 ms
```

Figur 6: *Resultat fra* ***traceroute***

Ovenstående figur 6 viser, at ruten fra host 1 til `192.168.3.13` er på 3 hop; først gennem router 1, dernæst router 3 og til sidst host 3, som har den adresse.

Q11.

```
192.168.1.11 : [30], 64 bytes, 0.054 ms (0.063 avg, 0% loss)
192.168.1.11 : [31], 64 bytes, 0.051 ms (0.063 avg, 0% loss)
192.168.1.11 : [32], timed out (0.063 avg, 3% loss)
192.168.1.11 : [33], 64 bytes, 0.059 ms (0.063 avg, 2% loss)
192.168.1.11 : [34], 64 bytes, 0.134 ms (0.065 avg, 2% loss)
```

Figur 7: *Resultat fra ping mellem host 4 og host 1*

Som det kan ses på figur 7, genoprettes forbindelsen mellem de to routere næsten med det samme. Den ene pakke, der nåede at forsvinde, gik blot tabt undervejs, da forbindelsen blev afbrudt - dette opdagede router 4 og sendte den næste ad en anden vej. Dette er langt hurtigere end i Q5. Når forbindelsen afbrydes, sendes der en `LS Update`-pakke, som informerer om ændringen i mulige ruter, eftersom der har været en opdatering i interfaces.

Q12.

Også her, som i Q7, bliver forbindelsen genoprettet med det samme. Når det første kabel bliver trukket (router 1 - SW), bliver en ny rute med det samme fundet, hvorfor det at trække det andet stik ikke gør en forskel - en alternativ rute er nemlig allerede fundet.

Q13.

	Før ændring	**Efter ændring**
Host 1 til host 4	62	62
Host 4 til host 1	62	61

Tabel 1: *Tabel over TTL-værdier før og efter introduktion af cost*

`ping`-kommandoen giver de i tabel 1 viste værdier. Det er her vist, at der kommer en mindre TTL fra host 4 til host 1, når der introduceres en cost på forbindelsen `r14` ved router 1. Denne TTL-værdi er den værdi, ICMP-pakken har, når den kommer tilbage fra fra host 1. Router 1 vil altså sende pakken en anden vej end router 4. Dette er eftersom der er kommet en højere metric med den givne `cost`-linje i konfigurationsfilen.

Q14.

Meningen med areas indenfor `OSPF` er, at en router kun holder styr på den fulde netværkstopologi indenfor netop det area, som routeren tilhører. Der er kun "summary"-viden om routere udenfor dette area. OSPF har den ulempe i store netværk, at den fulde netværkstopologi med metrics og alt andet kræver meget lagerplads. Dette kan man bekæmpe ved at dele det store netværk ud i flere areas. Ulempen ved dette er naturligvis, at routerne nu ikke ved lige så meget om topologien, hvilket gør det sværere at finde de bedste veje og opdatere ruterne; men til gengæld behøver routerne ikke lave lige så meget arbejde eller gemme så meget information.

Q15.

Efter ændringen har host 1 kun mulighed for at pinge router 2 og hosts på router 2s netværk. Dette er eftersom router 1 og 2 er i samme area, men ikke samme som router 3 og 4 og deres tilhørende interne netværk. Ligeledes kan host 3 og 4 pinge hinanden, men de kan ikke pinge host 1 og 2.

Del 3 BGP

Q16.

Svarene kommer tilbage fra host 3, men ikke fra host 4. Dette er eftersom router 1 har fået information fra router 3 om router 3 og dens interne netværk `192.168.3.0/24`, men ikke ved noget om router 4. Det er da heller ikke muligt at pinge host 4 på `192.168.4.14`. Router 3 har da ikke sendt OSPF-informationen med, da den kommunikerede via BGP med router 1.

Q17.

Følgende figurer viser BGP- og OSPF-konfigurationer[1]:

[1] Da kun routerne 1 og 3 har en BGP-konfiguration er jeg gået ud fra, at spørgsmålet mener BGP- og OSPF-konfigurationer for disse to routere og ikke routerne 1 og 2.

```
hostname router1
password cyber

router bgp 101
  bgp router-id 10.10.0.1
  redistribute connected
  neighbor 10.1.3.3 remote-as 202
  redistribute ospf
```

```
hostname router3
password cyber

router bgp 202
  bgp router-id 10.10.0.3
  redistribute connected
  neighbor 10.1.3.1 remote-as 101
  redistribute ospf
```

```
hostname router1
password cyber

router ospf
  redistribute connected
  redistribute bgp
  network 10.1.2.0/24 area 1
  network 10.1.3.0/24 area 1
  network 10.1.4.0/24 area 1
```

```
hostname router3
password cyber

router ospf
  redistribute connected
  redistribute bgp
  network 10.1.3.0/24 area 2
  network 10.3.4.0/24 area 2
```

Figur 8: *BGP- og OSPF-konfigurationer for routerne 1 og 3*

Q18.

Ruten fra host 2 til host 4 går gennem router 2, dernæst router 1, dernæst router 3 og til sidst router 4. Ligeledes går en besked fra host 4 gennem routerne 4, 3, 1 og til sidst 2, før den når frem til målet.

```
traceroute to 192.168.4.14 (192.168.4.14), 30 hops max, 60 byte packets
 1  192.168.2.2  0.027 ms  0.004 ms  0.003 ms
 2  10.1.2.1  0.012 ms  0.005 ms  0.004 ms
 3  10.1.3.3  0.012 ms  0.006 ms  0.020 ms
 4  10.3.4.4  0.013 ms  0.007 ms  0.007 ms
 5  192.168.4.14  0.014 ms  0.008 ms  0.008 ms
traceroute to 192.168.2.12 (192.168.2.12), 30 hops max, 60 byte packets
 1  192.168.4.4  0.026 ms  0.005 ms  0.040 ms
 2  10.3.4.3  0.014 ms  0.005 ms  0.016 ms
 3  10.1.3.1  0.013 ms  0.006 ms  0.006 ms
 4  10.1.2.2  0.012 ms  0.006 ms  0.039 ms
 5  192.168.2.12  0.015 ms  0.009 ms  0.009 ms
```

Figur 9: *Resultat fra traceroute mellem hosts 2 og 4*

Q19.

Routing-tabellen for router 4 viser[2], at router 4 kan tilgå alle netværk enten direkte eller via de andre routere. Udover netværkene med `10.x.x.0/24`-adresser, kan router 4 også nå alle routernes interne netværk med `192.168.x.0/24`-adresser.

```
10.1.2.0/24 via 10.3.4.3 dev r34 proto zebra metric 20
10.1.3.0/24 via 10.3.4.3 dev r34 proto zebra metric 20
10.1.4.0/24 dev r14 proto kernel scope link src 10.1.4.4
10.2.4.0/24 dev r24 proto kernel scope link src 10.2.4.4
10.3.4.0/24 dev r34 proto kernel scope link src 10.3.4.4
192.168.1.0/24 via 10.3.4.3 dev r34 proto zebra metric 20
192.168.2.0/24 via 10.3.4.3 dev r34 proto zebra metric 20
192.168.3.0/24 via 10.3.4.3 dev r34 proto zebra metric 20
192.168.4.0/24 dev h4 proto kernel scope link src 192.168.4.4
192.168.10.1 via 10.3.4.3 dev r34 proto zebra metric 20
192.168.20.2 via 10.3.4.3 dev r34 proto zebra metric 20
192.168.30.3 via 10.3.4.3 dev r34 proto zebra metric 20
```

Figur 10: *Resulterende routing table fra router 4*

[2] fundet med kommandoen sudo ip netns exec router4 ip route, da ip route ikke har et -n flag.

Q20.

bgp

No.	Time	Source	Destination	Protocol	Length	Info
2	5.966027745	10.1.3.1	10.1.3.3	BGP	85	KEEPALIVE Message
4	5.966092787	10.1.3.3	10.1.3.1	BGP	85	KEEPALIVE Message
6	6.972523816	10.1.3.3	10.1.3.1	BGP	126	UPDATE Message
22	65.966656202	10.1.3.3	10.1.3.1	BGP	85	KEEPALIVE Message
24	65.966721362	10.1.3.1	10.1.3.3	BGP	85	KEEPALIVE Message

Figur 11: *Screenshot fra WireShark med BGP-beskeder*

På ovenstående figur 11 ses frekvensen mellem BGP-beskederne. Disse viser, at der går 60 sekunder mellem hver keepalive. Efter de 60 sekunder sender både router 1 og router 3 en keepalive. Disse beskeder bliver sendt over en TCP-forbindelse, som først skal oprettes mellem de to routere. Efter denne er oprettet, bliver der sendt OPEN-beskeder fra hver router. Efter dette bliver der blot sendt UPDATE-beskeder, når der er sket noget nyt i netværket, eller KEEPALIVE-beskeder, når der er gået 60 sekunder. Hver af disse beskeder bliver naturligvis kvitteret med en ACK, da der er tale om en TCP-forbindelse. Dette kan ses på nedenstående figur 12.

No.	Time	Source	Destination	Protocol	Length	Info
769	1908.6617488…	10.1.3.3	10.1.3.1	TCP	74	51034 → 179 [SYN] Seq=0 Win=642
770	1908.6617670…	10.1.3.1	10.1.3.3	TCP	74	179 → 51034 [SYN, ACK] Seq=0 Ac
771	1908.6617764…	10.1.3.3	10.1.3.1	TCP	66	51034 → 179 [ACK] Seq=1 Ack=1 W
772	1908.6618274…	10.1.3.3	10.1.3.1	BGP	125	OPEN Message
773	1908.6618312…	10.1.3.1	10.1.3.3	TCP	66	179 → 51034 [ACK] Seq=1 Ack=60
774	1908.6620119…	10.1.3.1	10.1.3.3	BGP	144	OPEN Message, KEEPALIVE Message
775	1908.6620167…	10.1.3.3	10.1.3.1	TCP	66	51034 → 179 [ACK] Seq=60 Ack=79

Figur 12: *TCP-forbindelse, der bliver oprettet, før der kan sendes BGP.*