

Av Fredrik Villo og Per Nyberg

TSD2090V-1

Lab 4 (Dynamisk ruting)

Innholdsfortegnelse

Sammendrag.....	4
1 Innledning.....	5
2 Problemstilling	6
3 Teoridel	7
3.1 Dynamisk ruting.....	7
3.2 RIP	8
3.3 Zebra	9
3.4 Metoder og fremgangsmåte	10-11
4 Konklusjon	12
5 Kilder og referanser	13

Sammendrag

I denne labben (lab 4) fikk gruppe 4 i oppgave å utforske og forstå funksjonaliteten til dynamisk ruting ved bruk av «Free Range Routing» (FRR) **[1]**.

Protokollen som ble brukt i forbindelse med denne labben var «Routing Information Protocol» (RIP) **[2]**, der både RIP og Zebra **[3]** ble brukt som ruting demoner.

Labben hadde fokus på oppsett av virtuelle maskiner, tildeling av statiske IP adresser og oppsett av Zebra og RIP.

Formålet med denne labben var å forstå hvordan dynamisk ruting fungerer, og hvordan man konfigurerer ruting mellom to maskiner i ett nettverk.

1 Innledning

Ruting på nettverk kan settes opp på mange måter f.eks. manuelt der du som administrator sier på hvilken måte trafikken skal flyte på nettverket. Å sette opp ruting manuelt tar ikke lang tid om det kun er to eller tre maskiner som skal snakke med hverandre – men det er når det blir flere punkter som skal rutes som det kan bli mer tidskrevende enn nødvendig. **[4]**

Dynamisk ruting er en måte å løse dette på. Poenget med dynamisk ruting er å effektivisere veien mellom alle rutere i nettverket slik at du alltid får den raskeste veien mellom punkt A og punkt B. **[5]**

Hensikten med Lab 4 er at gruppe 4 skal lære hvordan dynamisk ruting fungerer, sette opp en maskin og bruke TCPdump for å se hvordan programmet automatisk fanger opp nye rutere i nettverket og ruter mellom dem. **[6]**

2 Problemstilling

Hvordan kan dynamisk ruting konfigureres og implementeres i et nettverk ved hjelp av Free Range Routing (FRR) og Routing Information Protocol (RIP), og hvilke praktiske utfordringer oppstår under oppsett og testing av ruting funksjonalitet mellom to maskiner? I hvilken grad bidrar bruk av verktøy som TCPdump til forståelse av nettverkstrafikk og ruting protokollers oppførsel i sanntid?

3 Teoridel

3.1

Dynamisk ruting

Dynamisk ruting er ikke noe som vi gjør manuelt – i stedet bruker vi programmer for å hjelpe oss. Programmene konfigurerer automatisk de beste veiene mellom alle rutere i nettverket gjennom å sende datapakker til alle rutere i nettverket. Etter hvert som den får svar fra andre rutere, lager den en ruting tabell.

En ruting tabell er et slags skjema over hvilke rutere som finnes i nettverket du befinner deg i og hvordan man raskest kommer seg til ønsket destinasjon. Hver ruter har en egen ruting tabell (routing table) som sier hvor pakken skal gå for å komme frem til riktig destinasjon. Ruting tabellene oppdateres fortløpende slik at den hele tiden gir den raskeste veien fra A til B. Denne prosessen oppdateres regelmessig slik at den kan oppdage om nye nettverk kobles til eller om den finner en bedre og raskere vei og rute mellom to punkter. **[7]**

3.2

RIP

Det finnes forskjellige dynamisk ruting programmer som RIP (Routing Information Protocol), OSPF (Open Shortest Path First) og BGP (Border Gateway Protocol) og i lab 4 bruker vi RIP.

RIP er det eldste dynamisk ruting programmet. Det bruker «*hop count*» som målepunkt - som på godt norsk betyr antall hopp fra kilden til destinasjonen. Oftest er lavest mulig «*hop count*» den beste veien ettersom hvert hopp legger til tid i pakkens reisevei. Men selvfølgelig kan det finnes andre faktorer som gir større utslag for tid enn «*hop count*». Som f.eks. båndbredde. Er det svært lav båndbredde en plass i rutingen vil RIP finne en raskere vei, selv om det betyr flere hopp.

RIP bruker port 520, som er den samme for rutere, og det er også den som brukes for dynamisk ruting og kommuniserer gjennom UDP – det vil si at den ikke har en handshake og ikke vet om datagrammene kommer frem eller ikke. Anledningene til hvorfor den bruker UDP i stedet for TCP er mange – men i hovedsak er det fordi UDP har støtte for broadcast og multicast. **[2]**

3.3

Zebra

Det blir observert at maskin A fungerer som en hvilken som helst annen vanlig ruter, det er Zebra som gjør denne jobben med å lage maskinen om til ruter.

gnu.org sier dette om Zebra [8]:

« Zebra is a multi-server routing software which provides TCP/IP based routing protocols. Zebra turns your machine into a full powered router. Some of the features of Zebra include:

- *Common routing protocols such as RIP, OSPF, BGP supported.*
- *IPv6 routing protocols such as RIPng and BGP-4+ supported.*
- *User can dynamically change configuration from terminal interface.*
- *User can use command line completion and history in terminal interface.*

IP address-based filtering, AS path-based filtering, attribute modification by route map are supported. »

-

3.4

Metode og fremgangsmåte

Metode og fremgangsmåte

I denne labben brukes det flere metoder og fremgangsmåter for å forstå hvordan dynamisk ruting fungerer.

1. Oppsett av maskiner

- **Virtuelle maskiner:** Gruppen ble bedt om å sette opp to ulike maskiner til denne øvelsen,

Maskin A og Maskin B. Maskin A skulle konfigureres med to nettverksadaptere, der adapter 1 skulle settes til «bridged adapter» og adapter 2 skulle settes til «host only adapter». Maskin B skulle kun ha ett adapter som skulle settes til «bridged adapter»
- **IP konfigurasjon:** Maskin A og maskin B blir konfigurert med de statiske IP-adressene

10.4.0.1/24 og 10.4.0.2/24 henholdsvis, for at sikre at maskinene kan kommuniserer med

hverandre, og at maskin A kan kommunisere med USN sitt skolenett.

2. Konfigurerings av ruting og FRR

- **Aktivisering av ruting på Maskin A:** Ved hjelp av kommandoen «echo 1 > /proc/sys/net/ipv4/ip_forward», dette skrev verdien «1» til en konfigurasjonsfil slik at IP-forwarding ble aktivert. Dette gjør så maskin A kunne fungere som ruter. Firewalld ble også deaktivert på begge maskinene for å unngå at noe trafikk ble blokkert.
- **Konfigurerings av Zebra og RIP på Maskin A:** Zebra ble konfigurert for å håndtere ruting, og RIP ble aktivert for å håndtere dynamisk ruting mellom Maskin A og Maskin B. Gruppen gjorde dette ved å sette opp konfigurasjonsfiler for Zebra og RIP ved bruk av telnet med kommandoen «telnet 127.0.0.1 2601» for Zebra og «telnet 127.0.0.1 2602» for RIP.

I tillegg så konfigureres eth0 og eth1 opp mot RIP med kommandoen «*network eth0*» «*network eth1*».

3. Bruk av verktøy for testing

- **TCPdump:** Gruppen brukte dette verktøyet til å fange og analyserer nettverkstrafikken i sanntid. Gruppen brukte kommandoen «*tcpdump -n -i eth1 net 10.4.0.0/24 and not arp*» for å fange pakker og sørge for at det ikke kom noe støy fra ARP-protokollen [9].
- **IP route:** Gruppen kunne observere at når fler og fler grupper ble ferdige med lab oppgavene sine, så dukket de opp en etter en ved bruk av «*ip route ls*»

4. Feilsøking og validering

- Etter hvert trinn i labben ble resultatene validert ved hjelp av ping-kommandoen og TCPdump for å bekrefte at maskinene kunne kommunisere. Ved feil ble konfigurasjonene på nytt kontrollert, og nødvendige justeringer ble gjort for å sikre korrekt oppsett.

4 Konklusjon

I denne labben har gruppe 4 fått en bedre forståelse av hvordan dynamisk ruting fungerer ved bruk av Free Range Routing (FRR) og Routing Information Protocol (RIP). Gruppen har brukt det de har lært i tidligere labber om oppsett av virtuelle maskiner med statiske IP-adresser, noe som har styrket deres forståelse av tidligere temaer.

Videre har gruppen lært hvordan man konfigurerer Zebra for ruting og hvordan RIP brukes for dynamisk ruting mellom to maskiner. Dette ble gjort ved å manuelt gjøre endringer i konfigurasjonsfiler ved hjelp av kommandoer, på en måte som ligner metodene de ble kjent med i tidligere fag, som «Introduksjon til Linux».

Ved hjelp av TCPdump og andre verktøy har gruppen fått en dypere forståelse av hvordan man analyserer nettverkstrafikk i sanntid. Gjennom denne prosessen har gruppen erfart at dynamisk ruting kan effektivisere nettverkshåndtering, særlig i nettverk med mange rutere.

De har også lært viktigheten av å være nøye og oppmerksomme på oppsett og konfigurasjon, samt å feil søke underveis for å unngå at ting går galt.

5 Kilder og referanser

- [1] «Overview — FRR latest documentation». Åpnet: 25. september 2024. [Online]. Tilgjengelig på: <https://docs.frrouting.org/en/stable-10.0/overview.html>
- [2] «Routing Information Protocol (RIP)», GeeksforGeeks. Åpnet: 25. september 2024. [Online]. Tilgjengelig på: <https://www.geeksforgeeks.org/routing-information-protocol-rip/>
- [3] «Zebra — FRR latest documentation». Åpnet: 25. september 2024. [Online]. Tilgjengelig på: <https://docs.frrouting.org/en/latest/zebra.html>
- [4] «What is a Manual Network Route and how do I configure it? – Mushroom Networks», <https://www.mushroomnetworks.com/>. Åpnet: 26. september 2024. [Online]. Tilgjengelig på: <https://www.mushroomnetworks.com/docs/what-is-a-manual-network-route-and-how-do-i-configure-it/>
- [5] «What is Dynamic Routing in Computer Network?», GeeksforGeeks. Åpnet: 26. september 2024. [Online]. Tilgjengelig på: <https://www.geeksforgeeks.org/what-is-dynamic-routing-in-computer-network/>
- [6] «tcpdump(1) man page | TCPDUMP & LIBPCAP». Åpnet: 17. september 2024. [Online]. Tilgjengelig på: <https://www.tcpdump.org/manpages/tcpdump.1.html>
- [7] «Routing Tables in Computer Network», GeeksforGeeks. Åpnet: 26. september 2024. [Online]. Tilgjengelig på: <https://www.geeksforgeeks.org/routing-tables-in-computer-network/>
- [8] «Zebra - GNU Project - Free Software Foundation». Åpnet: 25. september 2024. [Online]. Tilgjengelig på: <https://www.gnu.org/software/zebra/>
- [9] «An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware», Internet Engineering Task Force, Request for Comments RFC 826, nov. 1982. doi: 10.17487/RFC0826.