

Laboration 3: Routing

Laborationen illustrerar användning av subnät och enklare routing och forwarding i mindre nätverk. Konstruktion av enkla routingtabeller, subnätmasker och diverse protokoll studeras.

Introduktion

Vi skall bland annat studera routing och forwarding i ett nät (**vad var skillnaden??**) En router har alltid sin tabell som bestämmer hur den skall hantera paket som kommer in. Antingen kan tabellen ge vilken utgång (interface) på routern paketet skall till eller vilken är nästa router som skall ha paketet, dessa kan också blandas i samma tabell. Är det till en slutanvändare anges dock alltid utgång. Sist i tabellen finns alltid *default* som talar om vart ett paket vars adress inte hittas i tabellen skall skickas. De här tabellerna är oftast statiska och uppdateras antingen manuellt eller av det routingprotokoll som används, som till exempel RIP eller OSPF.

Routers tabell

Anta att tabellen vid en viss tidpunkt i en router (R1) ser ut enligt nedan. Routern kan leverera paketen direkt via utgångarna eth0 och eth1 (ethx=Ethernetutgång x). Routern kan också skicka vidare paketen till R2, R3 och R4.

<i>Subnet adress</i>	<i>Subnet mask</i>	<i>Next hop</i>
130.237.176.0	255.255.240.0	eth0
130.237.80.0	255.255.240.0	eth1
130.237.208.0	255.255.240.0	R2
192.4.153.160	255.255.255.224	R3
<i>default</i>		R4

Tabell 1: Router R1s interna tabell

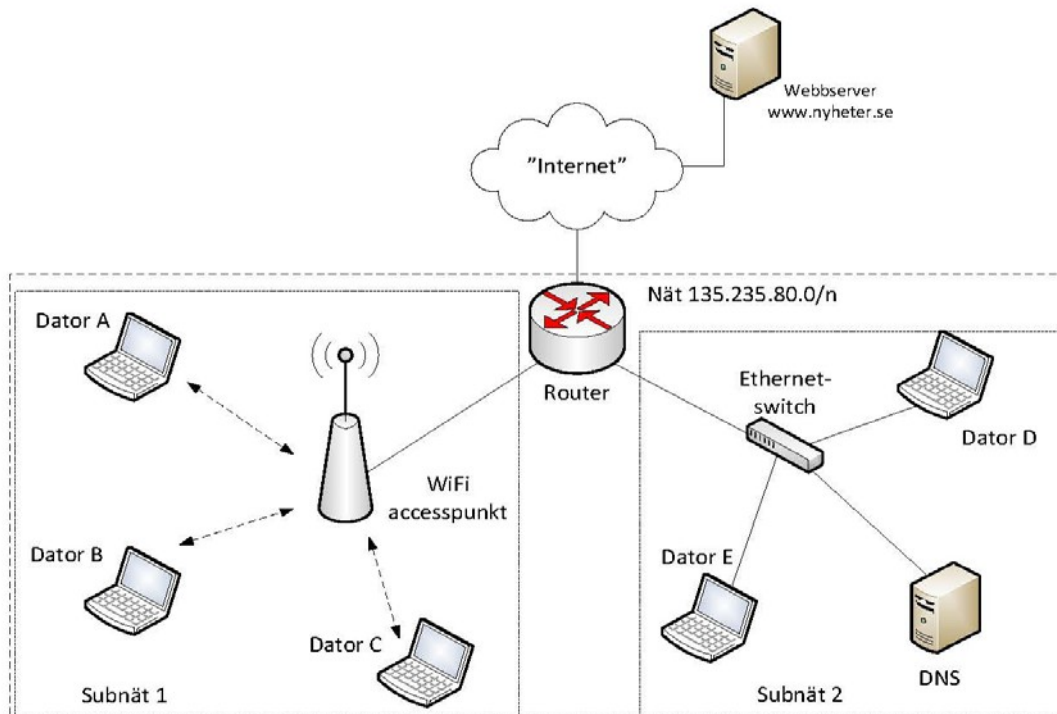
Vart kommer följande paket att skickas av routern?

1. 130.237.188.96
2. 130.237.90.200
3. 130.237.96.255
4. 130.237.180.235
5. 192.4.153.190
6. 192.4.153.175

När vi nu, förhoppningsvis, förstår tabellens funktion skall vi senare fylla i en tom tabell för en router.

Synliga adresser på nätverk

Det kan vara bra att veta vilken information som en enhet som inte är direkt medverkande i en kommunikation kan se. I detta fall skall vi bara fokusera på vilka adresser en enhet kan se i ett nät enligt Figur 1. Vi antar för enkelhetens skull att WiFi-nätet baseras på någon okrypterad IEEE 802.11 standard, att accesspunkten arbetar på länkskiktet (som en switch), samt att Dator D och E (och DNS-servern) inte "hör" kommunikationen i Wi-Fi-nätet.



Figur 1: Nät 135.235.80.0/n

Vilka datorer kan skicka information samtidigt i nätet?

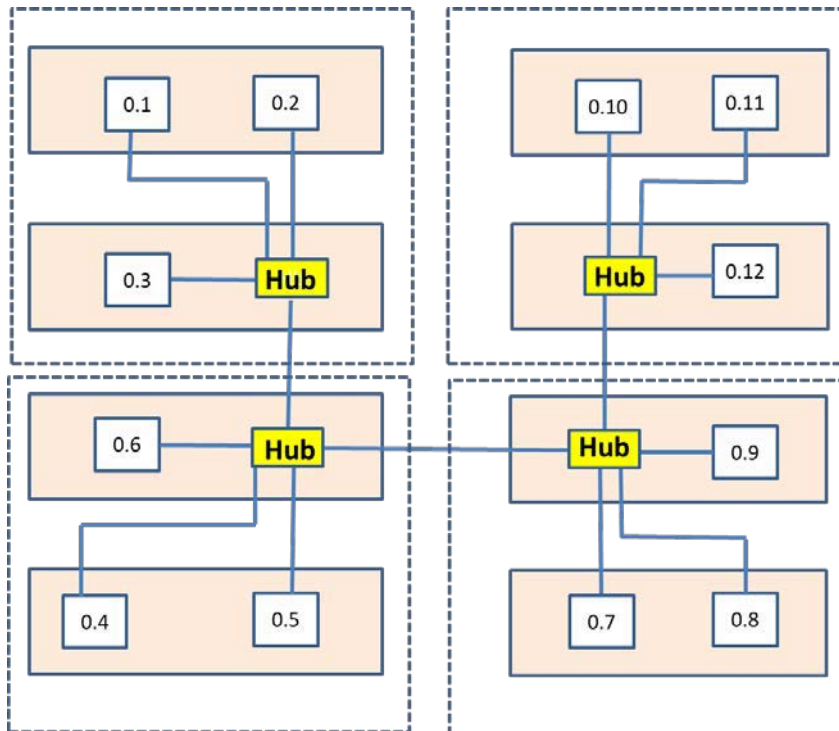
Antag att alla adress-cachar är tomma. Vilka adresser kommer Dator C ha sett efter att Dator A har skickat ett IP-paket till Dator D (A vet D:s IP-adress)?

Nätverksbyggande

Nätverket består av fyra grupper av datorer med tre datorer i varje grupp. En av datorerna i gruppen har två nätverkskort och kan fungera som router. För att förbinda datorerna används fyra hubbar.

Vi kommer att studera följande tre olika sätt att koppla ihop datorerna under laborationen:

1. Ett nät som består av ett fysiskt och ett logiskt nät
2. Ett nät som består av ett fysiskt och fyra olika delnät
3. Ett nät som består av fyra fysiska delnät som knyts ihop med routrar



Figur 2: 12 datorer med hubbar och IP-adresser enligt 192.168.x.y

Variant 1 a

Alla de 12 datorerna länkas via de fyra hubbarna samman i ett delnät enligt bilden ovan. IP-nummer för datorerna väljs enligt 192.168.x.y där $x = 0$ och $y = 1, 2, 3, \dots, 12$. Man måste också bestämma vilken utgång/nätverkskort som skall ha denna IP-adress. Ethernetutgångar numreras med början på 0, vilket ger den första utgången namnet *eth0*. I Linux sätts IP-nummer enligt mönstret (<..> betecknar att här skall det fyllas i något).

```
ifconfig <utgång> <ip-nummer> netmask <nätmask> up
```

(Exempel första datorn: `ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up`)

På en router/dator är alla utgångar nedstängda som default (på en switch och hub är de öppna) därav det sista kommandot "up", som öppnar upp utgången. Vi sätter IP-nummer på alla datorerna enligt kommandot ovan och nummer enligt Figur 2.

Vilka av datorerna tror Du kan kommunicera med varandra och varför?

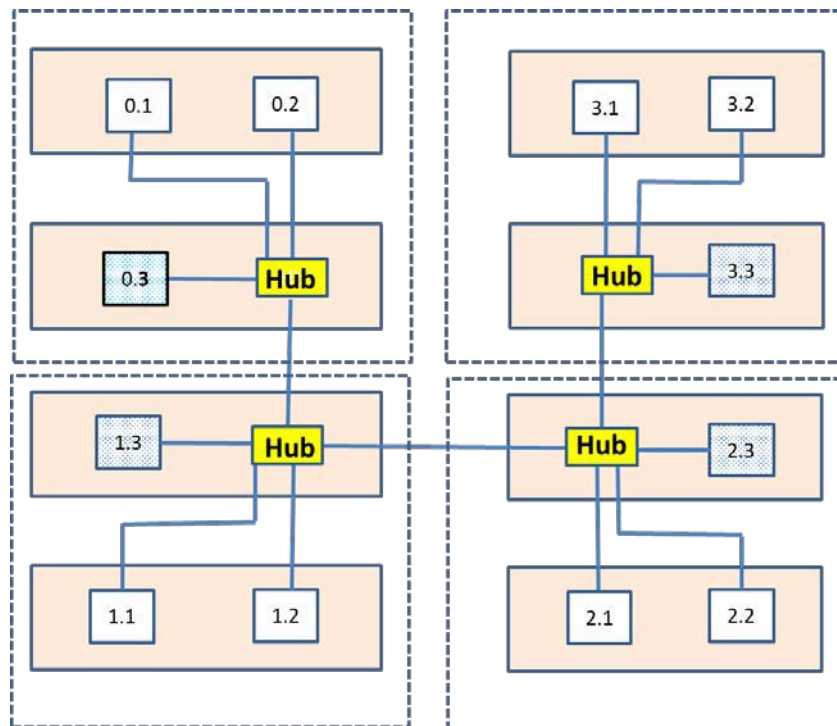
Variant 1 b

En utveckling av föregående variant innebär att vissa av datorerna ges IP-nummer som skiljer sig i tredje oktetten. Anta att vi byter ut nollan i tredje oktetten mot 1 eller något annat tal.

Vilka av de andra datorerna fungerar det att kommunicera med efter ändringen? (Om inte alla, så fundera ut vilken/vilka modifieringar som behövs för att det skall fungera.)

Variant 2

Den fysiska kopplingen av nätet bibehålls men nätet delas logiskt in i fyra delnät (olika kontor, olika avdelningar, olika institutioner,...) där datorerna får IP-nummer 192.168.x.1 och 192.168.x.2 medan servern (lite mörkare) får 192.168.x.3 (x = 1, 2, 3, 4). Nätmasken hos datorerna sätts till 255.255.255.0 och deras SGW (standard nätsluss (gateway)) sätts till 192.168.x.3. Delnäten binds ihop i en kedja genom att de tre första hubbarnas hub-out (uplink) ansluts med en patch-kabel till nästa delnäts hub, se Figur 3.



Figur 3: Datorer och servrar indelade i fyra logiska nät

Vilka av de andra enheterna kan t ex 192.168.1.2 kommunicera med efter ändringen?

Varför vill man bygga ett nät så här?

Variant 3

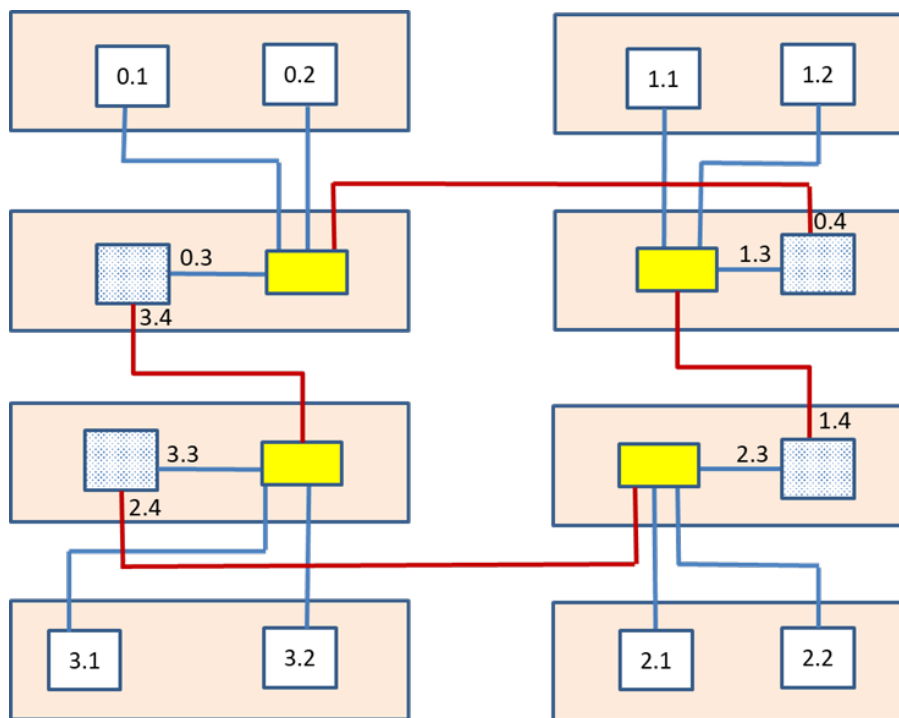
Nu vill vi använda serverna (som har två nätverkskort) som routrar genom att de är anslutna till två delnät (ett per nätverkskort) och därmed förbinda två angränsande subnät. Servern i grupp x tilldelas IP-numren 192.168.x.3 (som innan) respektive 192.168. $((x-1)\%4)$.4. Patchkablarna som förbinder grupperna flyttas om enligt:

"Out-to-hub" <-----> "Vanlig ingång på hubben i nästa grupp" kopplas om till

"Vanlig ingång på hubben" <-----> "Andra servernätverkskortet i nästa grupp"

Vi sluter dessutom "cirkeln" av delnät med ytterligare en förbindelse, enligt Figur 4. Nätmaskerna skall vara 255.255.255.0 i TCP/IP-inställningarna för båda nätverkskorten.

Försök att lägga in nya rutter i routertabellen. Observera att även metrik-parametern nu måste användas i route-kommandot. Eventuellt behövs också vilket nätverkskort (gränssnitt) rutten gäller (tänk på numreringen).



Figur 4: Datorer och servrar indelade i fyra fysiska delnät

Vad är fördelen med denna koppling jämfört med variant 2?

De olika delarna/avdelningarna/institutionerna vill kunna kommunicera med varandra, det vill säga alla skall kunna nå alla andra. Eftersom vi nu inte har tillgång till andra användare än de som finns inom vårt delnät vilket inkluderar routern med IP-adress 192.168.x.3, måste vi gå via denne om vi skall nå en användare utanför vårt delnät.

För att detta skall fungera måste vi sätta upp forwardingtabellerna i routern via något routingprotokoll.

Sätt upp en strategi för hur nätverket skall fungera och de olika delnäten nå varandra.

Fyll i tabellerna för routrarna så att denna strategi effektueras.

Router nätverk 0

Destination	Nätmask	Nätsluss	Utgång	Max antal hopp

Router nätverk 1

Destination	Nätmask	Nätsluss	Utgång	Max antal hopp

Router nätverk 2

Destination	Nätmask	Nätsluss	Utgång	Max antal hopp

Router nätverk 3

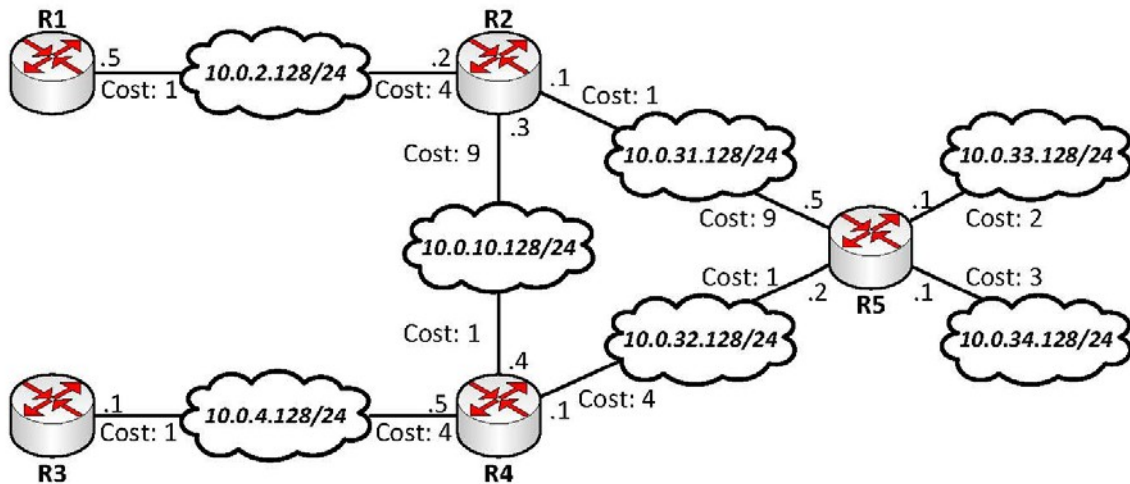
Destination	Nätmask	Nätsluss	Utgång	Max antal hopp

I appendix finns lite matnyttiga kommandon för att sätta upp dessa tabeller manuellt, om man alltså inte använder RIP eller OSPF (hur fungerade de, och vilken kategori av algoritmer tillhör respektive?)

Skriv de kommandon som behövs för att fullborda tabellen för router 192.168.1.3

Routingalgoritmer

I detta avsnitt skall vi titta på algoritmer som beräknar och fastställer tabellerna i routrarerna.



Figur 5: Ett exempel på ett 10.0.x.y nätverk

För att kunna se hur automatiska routingalgoritmer fungerar i ett lite större nät kommer vi i nedanstående uppgifter att utgå från nätet i Figur 5. Routingtabellerna kommer att se ut enligt Figur 6 där ett vägval är ifyllt i R4s tabell. Skall ett paket i R4 till en adress på nätet 10.0.2.128/24 skall det från R4 skickas till nätet 10.0.10.128/24 och specifikt till IP-adressen 10.0.10.3 som är ingången till R2.

Nätverk	Nästa nod	Hopp
10.x.x.x		
10.0.2.128/24	10.0.10.3	1
10.x.x.x		

Figur 6: Ett exempel på en väg i tabellen för R4

Först skall vi titta på algoritmer som tillhör familjen *Distance Vector*.

Beskriv grundprinciperna för Distance Vector baserade routingprotokoll.

Hur ser router R5:s initiala routingtabell ut?

Hur ser router R5:s slutgiltiga routingtabell ut efter att nätet konvergerat? Visa alla steg.

Vi antar nu att nätet, i Figur 5, istället använder ett *Link State* baserat routingprotokoll. Tänk på att kostnaden för att till exempel ta sig från R4 till R2 över nätet 10.0.10.128/24 kostar 1, medan samma väg åt andra hållet kostar 9.

Beskriv grundprinciperna för Link State baserade routingprotokoll.

Beskriv innehållet i det första meddelande som router R5 skickar.

Hur ser R5:s slutgiltiga routingtabell ut efter att nätet konvergerat?

Appendix

Lite matnyttiga kommandon för att manuellt sätta tabeller i routrar. Kommandona bygger på att routern har ett operativsystem som är besläktat med Unix.

Om Du skulle prova i verkligheten och Ditt kommando inte är giltigt är anledningen troligast att Du inte har behörighet. Du kan då använda kommandot *sudo*. Detta kommando gör Dig till "super-user" vilket är en förutsättning för att få ändra IP-adresser och i tabeller. Observera att man oftast dessutom behöver ett lösenord för att kunna komma in i denna behörighet.

ifconfig

Detta kommando visar vilka nätverkskort som datorn har och vilka Ethernet- respektive IP-adresser som de har. Oftast heter nätverkskortet eth0, eth1 etc., men det finns också andra namn på en del datorer. På de flesta datorer finns det bara ett nätverkskort.

```
ifconfig eth0 100.100.100.1
```

Kommandot *ifconfig* finns i många varianter. Det som finns ovan ger nätverkskortet *eth0* IP-adressen 100.100.100.1.

```
ifconfig eth0 192.168.1.102 netmask 255.255.255.0
```

Detta är ytterligare en variant av *ifconfig*. Nu ger vi inte bara adressen 192.168.1.102, nu anger vi dessutom att nätmasken skall vara 255.255.255.0. Det innebär att de 24 första bitarna i adressen är nätadress, medan de 8 sista är en hostadress.

route

Detta visar datorns routingtabell som den använder för att avgöra vad som skall göras med ett IP-paket.

```
route add -net 192.56.76.0 netmask 255.255.255.0 dev eth0
```

Även kommandot *route* finns i många varianter. Det ni ser ovan talar om för datorn att alla IP-paket med nätverksadressen 192.56.76.0 med nätmasken 255.255.255.0 skall skickas ut på *eth0*. En rad kommer att läggas till i tabellen med just denna information.

```
route add -net 192.168.3.0 netmask 255.255.255.0 gw 100.100.0.1
```

Lägger till en nätsluss, det vill säga talar om att paket som skall till nätet 192.168.3.0 skall skickas till gnätslussen 100.100.0.1

Många gånger anger man både nätsluss och utgång när man lägger in det i tabellen.

```
route add -net 192.168.3.0 netmask 255.255.255.0 gw 100.100.0.1 dev eth0 metric 1
```

För att inte paketen skall åka runt i nätverket anger man också *metric*, dvs hur många steg paketet skall ta innan det kommer fram till destinationsnätverket.

```
route del -net 192.56.76.0 netmask 255.255.255.0 dev eth0
```

Detta kommando tar bort en rad i tabellen med just denna information, Lägg märke till att all informationen måste vara identisk med den rad som man vill ta bort för att det skall fungera.

```
cat /proc/sys/net/ipv4/ip_forward
```

Detta kommando används på en router. Får man svaret 0 så är forwarding inte påslagen, får man svaret 1 så är det möjligt.

```
sysctl .w net.ipv4.ip_forward=1
```

Detta kommando gör forwarding möjlig för routern.

```
sysctl .w net.ipv4.ip_forward=0
```

Detta kommando stänger av forwarding.