4. Nettverkskomponenter

Dette kapittelet beskriver de viktigste **aktive** nettverkskomponentene i datanettverk. Med *aktiv* nettverkskomponent menes her "bokser", eller programvare, som utfører en aktiv oppgave knyttet til kommunikasjonen i nettet. *Passive* komponenter, som f.eks. kabling, kontakter o.l. beskrives ikke. Maskiner som kommuniserer i nettet, f.eks. tjener- og klient-maskiner er heller ikke tema i dette kapittelet. Komponentene som beskrives er *hub*, *svitsj*, *ruter* og *brannmur*, og dessuten litt om trådløse "hjemmerutere".

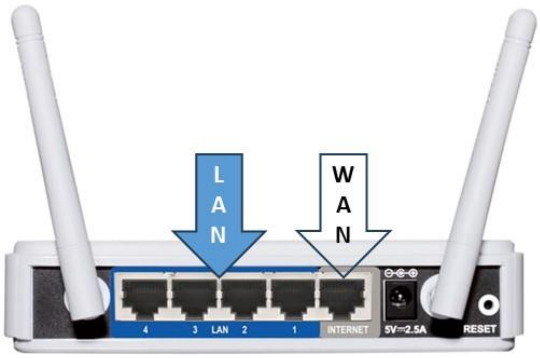
Kapittelet beskriver bruksområdet og hovedoppgavene til hver komponent. Dessuten forklares litt overordnet virkemåten deres, dvs. hvordan komponentene utfører oppgavene side. Dette er likevel ikke en fullstendig teknisk beskrivelse av virkemåten, protokoller eller algoritmer som brukes i komponentene. Det hører hjemme i mer spesialisert nettverkslitteratur.

4.1. Trådløs "hjemmeruter"

Den nettverkskomponenten som du kjenner best, er nok den trådløse "internetruteren" som du har i hjemmenettet ditt. Derfor starter vi dette kapittelet med en kort beskrivelse av denne. I dagligtale kalles denne boksen også for en "hjemmeruter", "WiFi-ruter", eller bare ruter. Det siste er ikke helt faglig presist, fordi ruterfunksjonen bare er én av flere deler i "hjemme-ruteren".

En "hjemmeruter" inneholder flere ulike nettverkskomponenter med forskjellige oppgaver:

- En *svitsj* som brukes for å koble sammen maskiner i hjemmenettet (lokalnettet) ditt via nettverkskabler. Svitsjen i hjemmeruteren har gjerne 4-5 nettverksporter merket **LAN**.
- En nettverksport som kobles til Internetlinjen din, og som ofte er merket **WAN** (*WAN-porten*). WAN-porten kan være beregnet for kobberkabel eller fiberkabel.



"hjemmeruter"

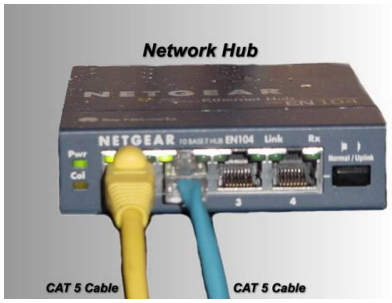
- Et *trådløst aksesspunkt* som lager det trådløse nettet i hjemmet ditt, og som du kan koble trådløse enheter til. Aksesspunktet kobler det trådløse nettet til det kablede nettet i svitsjen. På bildet over er de to antennene den synlige delen av aksesspunktet.
- En *ruter* som videresender data mellom lokalnettet (LAN) og Internett (WAN-porten). Ruterfunksjonen er programvare inne i "hjemmeruteren" og har ingen synlige deler på utsiden, bortsett fra nettverksportene. Ruteren har vanligvis også en NAT-funksjon (Network Address Translation), slik at du kan bruke private IP-adresser i lokalnettet på "innsiden" av ruteren.

I tillegg til disse nettverkskomponentene inneholder "hjemmeruteren" også to tjener-funksjoner: En *DHCP-tjener* for å tildele private IP-adresser til maskiner i lokalnettet på innsiden av ruteren, og en liten *webtjener* (programvare) for å konfigurere og administrere

"hjemmeruteren" via en nettside fra en maskin i nettet. Begge disse tjenerfunksjonene er programvare inne i "hjemmeruteren" og regnes ikke som nettverkskomponenter.

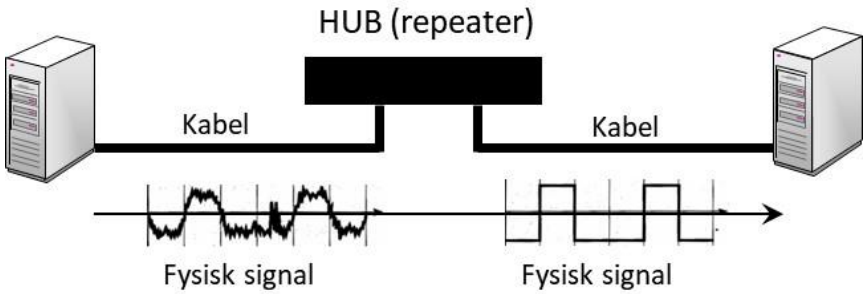
I større nettverk for profesjonelt bruk, er de fire første komponentene i listen over vanligvis separate nettverkskomponenter, og **ikke** samlet i én "boks" som i hjemmentt. Siden de også har ulike oppgaver og virkemåter, vil vi beskrive dem hver for seg i de følgende delkapitlene.

4.2. Hub



En *hub* er en nettverkskomponent som tidligere ble brukt i lokalnett, men som sjelden brukes i dag. I moderne LAN brukes *svitsjer* i stedet for huber. Dette kapittelet tar likevel med litt om huber fordi det gir grunnlag for bedre å forstå hvordan svitsjer fungerer og hvorfor disse har erstattet huber i nettverk. Dessuten brukes fremdeles huber i noe andre typer datakommunikasjon, f.eks i USB, FireWire og SATA.

Hub er det engelske ordet for *nav*. Navnet skyldes nok at hovedfunksjonen til en hub er å fungere som et sentralt punkt for å koble sammen flere nettverksenheter i et lokalt nettverk. Huben består av flere nettverksporter som er koblet sammen fysisk internt i denne. Når en nettverksenhet (maskin) sender data som fysiske signaler på en nettverkskabel, vil huben motta disse, kopiere dem og sende dem ut på **alle** andre porter / nettverksgrener på huben. Signalene som sendes ut er identisk med de som er mottatt, men fordi det er generert "på nytt", vil det være uten eventuell støy og demping som kan ha påvirket det opprinnelige signalet. Alle enhetene som er koblet til huben, vil altså motta det samme fysiske signalet. Denne egenskapen betyr at vi sier at en hub fungerer på **fysisk lag** (lag 1) i OSI-modellen.



En hub "gjentar" (regenererer) mottatte signaler på alle porter

Fordi en hub "gjentar" det fysiske signalet på flere nettverksporter, kalles en hub også for en *multiport repeater*. Når all trafikk på nettet sendes ut på alle nettverksporter, betyr det at alle nettverksenheter som er koblet til huben, vil dele kapasiteten (bitraten) i nettet. Det medfører også at signaler/data ofte sendes ut på porter/nettverksgrener der i de ikke vil bli "brukt". I nettverk er det ofte bare to enheter som kommuniserer med hverandre. Da vil det være mer effektivt hvis signalene bare blir sendt til den nettverksporten som mottakeren er koblet til, og ikke til alle andre i nettet. At signaler sendes til maskiner som ikke er mottakere av data, utgjør også en sikkerhetsrisiko. Det gjør det lettere å "avlytte" all nettverkstrafikk fra alle enheter som er koblet til huben.

På grunn av disse begrensningene og ulempene ved huber, har de fleste moderne nettverk erstattet bruk av huber med svitsjer, som er en mer avansert nettverkskomponent. Se neste kapittel

4.2.1. USB-hub

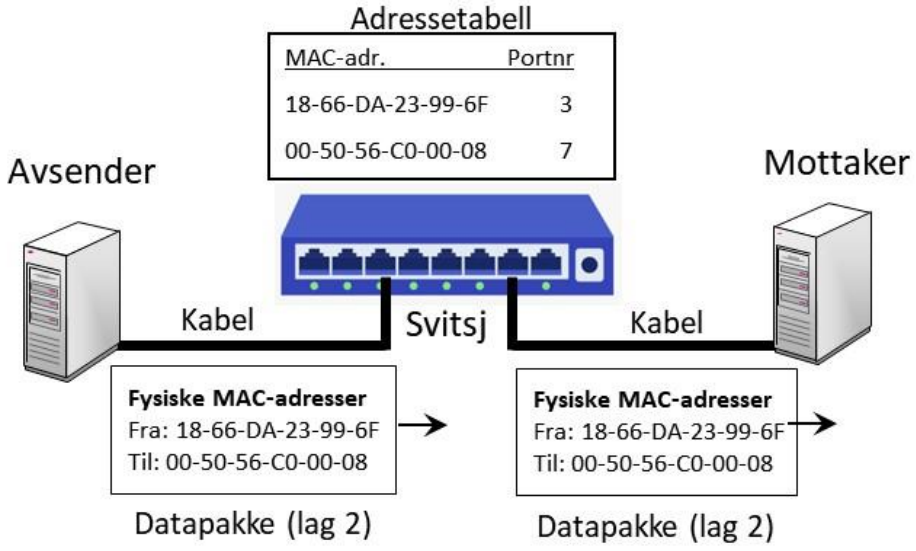
En USB-hub fungerer på liknende måte som en nettverkshub, men kobler i stedet sammen flere USB-enheter via USB-porter i huben. Med en USB-hub kan du koble flere USB-enheter til en enkelt USB-port på datamaskinen. Dette er spesielt nyttig hvis antall tilgjengelige USB-porter på en datamaskin er begrenset.

En USB-hub sender også ut fysiske USB-signaler på alle USB-portene i huben. Disse signalene er litt annerledes enn nettverkssignaler, men prinsippet er det samme.

4.3. Svitsj

I moderne, kablede lokalnett er maskinene i nettet koblet sammen via én eller flere *svitsjer*. Hovedoppgaven til en svitsj er omtrent den samme som en hub, men virkemåten er litt annerledes.

Når en svitsj mottar signaler (data) på en av nettverksportene, vil den tolke signalet som en datapakke, og lese adresseinformasjon fra innholdet i pakken[13].



Svitsjen vil bruke adresseinformasjon i pakken for å finne ut hvilken maskin i nettet som datapakken er ment for (mottaker). Deretter vil den sende datapakken videre **bare** på den

[13] Mer spesifikt så leser svitsjen pakkehodet fra lenkelaget (lag-2) i OSI-modellen, og bruker informasjonen i dette.

nettverksporten der mottakermaskinen er tilkoblet. For å kunne gjøre dette, må svitsjen lagre en *adressetabell* i internminnet. Tabellen inneholder nettverksadressene til alle enheter som er koblet til svitsjen, eller som kan nås fra denne, og hvilken fysisk port disse er tilkoblet.

Adressene som brukes av svitsjen, er adresser på lag 2 (lenkelaget) i OSI-modellen. Fordi svitsjen bruker denne informasjonen for å gjøre oppgaven sin, sier vi at svitsjer arbeider på lenkelaget (lag 2). Derfor kalles de også for lag-2 svitsjer[14]. Adressene på lag 2 kalles MAC-adresser, eller fysiske adresser. De skrives vanligvis på *hexadesimal* form, f.eks: 18-66-DA-23-99-6F. MAC-adressen er kodet inn i elektronikken fra produsenten av nettkortet, og er unik (entydig) for alle nettverkskort/-porter. MAC-adresser brukes bare for adressering **internt** i et lokalnett / fysisk nett. Når data skal sendes mellom ulike fysiske lokalnettet[15], må IP-adresser benyttes, og dette krever bruk av en ruter.

Windows-kommandoen `ipconfig /all` vil bl.a. vise den fysiske MAC-adressen til alle nettverkskort på maskinen:

```
C:\Users\admin>ipconfig /all
...
Ethernet adapter Ethernet:
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Ethernet Connection (2) I219-LM
   Physical Address. . . . . . . . . : 18-66-DA-23-99-6F
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::9328:2b05:31f6:833b%26(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.88.254(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

Når en svitsj bare sender datapakker ut på den porten der mottakeradressen befinner seg, vil det være vanskeligere å "tappe" data fra andre maskiner i nettet. En svitsj gir derfor bedre sikkerhet mot dette enn en hub. Dessuten unngår man unødig datatrafikk i nettet. Når hver gren i nettet bare transporterer data til og fra den maskinen som er koblet til grenen, betyr det at denne maskinen kan utnytte hele kapasiteten/bitraten på sin gren.

En annen fordel med svitsjer er at nettverksportene på en svitsj kan bruke forskjellige bitrater. For eksempel kan noen svitsjporter bruke 100 Mbit/s, mens andre bruker 1 Gbit/s. Vanligvis tilpasser svitsjen bitraten på hver port automatisk til den bitraten som maskinen i den andre enden kan bruke.

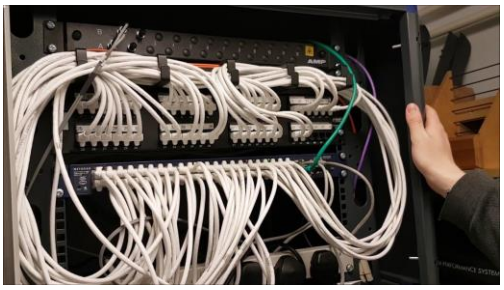
4.3.1. Administrerte og ikke-administrerte svitsjer

Svitsjer som er beregnet for små nettverk og hjemmenett, fungerer stort sett på egenhånd, og krever lite, eller ingen, konfigurering eller administrasjon. For eksempel vil svitsjen selv lære seg hvilke MAC-adresser som er koblet til hvilke porter, og bygge opp en tabell over dette. Slike svitsjer kalles gjerne for *ikke-administrerte* svitsjer.

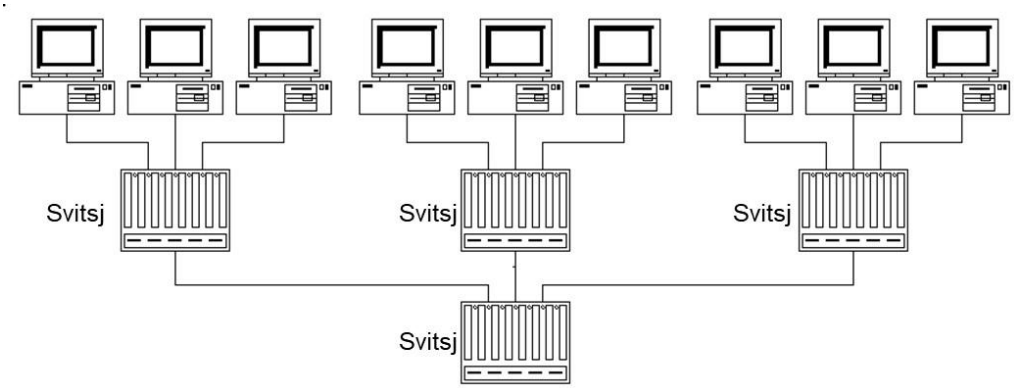
[14] Noen svitsjer kan også basere seg på IP-adresser (lag 3), og disse kalles da lag 3-svitsjer.

[15] mer presist mellom ulike IP-nett

I store lokalnett brukes større og mer avanserte svitsjer. Disse er ofte montert i rack som plasseres i låste skap i hver del av bygningene. Svitsjene kan ofte ha et web-grensesnitt for å kunne overvåke, administrere og konfigurere dem via nettet. Derfor kalles de gjerne *administrerte* svitsjer.



Store lokalnett kan bygges opp med flere svitsjer som er koblet sammen i ett lag-2 nett. Avanserte svitsjer har også mulighet for å dele opp lokalnettet i flere virtuelle lokalnett (lag-2 nett), såkalte VLAN.



4.4. Trådløst aksesspunkt

Et trådløst *aksesspunkt* (*access point - AP*) kobler maskiner / enheter med trådløse nettverkskort til et kablet nettverk. Aksesspunktet har én eller flere nettverksporter for kobling til det kablede nettet. Dessuten inneholder det en radiosender og -mottaker, som sammen med med antenne(r), utgjør "WiFi-delen" av aksesspunktet. WiFi-delen lager er trådløst nettverk som trådløse enheter kan koble seg til. Aksesspunktet kobler dette trådløse nettet til det kablede nettet.

Trådløst aksesspunkt

Når de trådløse enhetene skal kommunisere med maskiner som ligger utenfor WiFi-nettet, f.eks. en tjenermaskin, vil et aksesspunkt sende datapakker mellom WiFi-nettet og det kablede nettet. Aksesspunktet bruker da MAC-adresser på tilsvarende måte som en svitsj. Vi kan derfor se på et aksesspunkt som en "trådløs svitsj". Det regnes derfor også som en nettverkskomponent på lag 2 i OSI-modellen. Det trådløse og det kablede nettet bruker forskjellige signaltyper (radiosignaler og elektriske signaler) og ulike pakkeformater. Derfor må aksesspunktet også endre signaltype og pakkeformat når pakker skal videresendes mellom disse nettene. I nettverksfaget brukes navnet *bro* (*bridge*) om en enhet som endrer dette når pakker videresendes.

I et lite hjemmenettverk finner du et aksesspunkt som en del av "hjemmeruteren" i nettet. Derfor kan utseende til et aksesspunkt likne på en trådløs "hjemmeruter", men de må ikke forveksles.

Aksesspunkter brukes primært i større trådløse nett for å dekke større områder enn et hjemmenett. I store trådløse nettverk blir flere aksesspunkt plassert rundt om i bygningen / området der man ønsker trådløs dekning. AP'ene er koblet med kabel til svitsjer i det kablede nettet. Det er også mulig å bygge et rent trådløst nettverk bare med aksesspunkter. For at de trådløse enhetene skal kunne kommunisere med enheter utenfor nettet, må likevel WiFi-nettet på et eller annet sted være tilkoblet et annet nett, f.eks. en internetforbindelse.

I et trådløst nettverk deler alle trådløse enheter som er tilkoblet samme aksesspunkt, på kapasiteten i nettet. Dette skyldes bl.a. at alle enhetene kommuniserer på den samme radiofrekvensen. I nett med flere aksesspunkt, kan aksesspunktene bruke forskjellige frekvenser, og dette vil bedre kapasiteten i store WiFi-nett.

4.5. Ruter

Mens svitsjer og aksesspunkt hører hjemme på lag 2 i OSI-modellen, så er *rutere* nettverks-komponenter på lag 3 - nettverkslaget. Hvis maskiner i ett nettverk skal kunne kommunisere med maskiner i **andre** nettverk[16], må kommunikasjonen gå via en ruter. I TCP/IP-baserte lokalnett, vil hvert lag-2 nett som regel være konfigurert som ett IP-nett. Alle maskinene i samme IP-nett har samme subnettmaske og IP-adresser innenfor nettets adresseområde. Ved Maskiner i samme IP-nett kan kommunisere "direkte" med hverandre. Kommunikasjonen vil fremdeles gå via svitsjer eller aksesspunkt i nettet, så med "direkte" mener vi her at trafikken ikke går via noen ruter.

Et vanlig bruksområde for rutere er å koble et lokalnett, f.eks. et "hjemmenett", til Internett. Nettet til internett-leverandøren (ISPen) er da konfigurert som ett IP-nett, mens lokalnettet er et annet IP-nett. Ruteren kobler de to nettene sammen, og overfører IP-pakker mellom nettene. En ruter vil derfor alltid ha (minst) to nettverksporter som er koblet til hvert sitt nett. Rutere som kobler lokalnett til Internett, kalles gjerne for *kantrutere*.

LinkSys "hjemmeruter"

I store lokalnett, kan lokalnettet være delt opp i flere lag-2 nett, og som kan konfigureres som flere ulike IP-nett. I slike nett må rutere også brukes for å kommunisere mellom disse lokale nettene.

En ruter brukes altså **ikke** bare for tilkobling til Internett. I de sentrale delene av Internett brukes rutere også for å koble sammen flere "internett-linjer". Disse linjene vil igjen tilhøre ulike IP-nett. Slike rutere kalles gjerne *kjernenettrutere*. Slike rutere som kobler gjerne sammen flere enn to nettverk, og må derfor også ha flere nettverks-porter, dvs. minst en port for hvert nettverk.

Cisco kjernenettruter

[16] I et TCP/IP-basert nettverk betyr dette et annet **IP-nett**.

4.5.1. Videresending og ruting

En av hovedoppgavene til en ruter er altså å koble sammen (IP-)nett med ulike adresseområder, og å *videresende* (*forwarde*) pakker mellom disse nettene. For å avgjøre hvilket nett pakkene skal sendes til, bruker ruteren adresseinformasjon på lag 3 i datapakkene. I IP-nett betyr det i praksis IP-adresser fra IP-pakkene. Ruterne mottar IP-pakker, leser IP-adresser i pakkehodet, og bestemmer hvilket nett (port) pakken skal sendes ut på. For å avgjøre hvor pakken skal sendes, bruker ruteren også nettadresser og subnettmasker i ruterens *rutingtabell*.

I store nett, som Internett, kan IP-pakkene passere flere rutere før de når fram til endelig mottaker. Ruterne i Internett kan også være koblet sammen med flere ulike internetlinjer. Derfor kan det være flere ulike "veier" mellom avsender og mottaker av datapakkene. Det andre viktige hovedoppgaven til ruterne er å velge "beste mulige" vei gjennom nettene fra avsender til mottaker. Det er denne som oppgaven kalles for *ruting* (*routing*). Den "beste" veien kan endre seg over tid, for eksempel hvis nettverkslinjer endres eller går ned. Rutere må kunne takle slike endringer og velge raskeste, billigste og sikreste rute. Mekanismene og algoritmene som brukes for dette, hører hjemme i mer spesialiserte nettverkskurs, og er ikke tema her.

4.5.2. Adresseoversetting - NAT

Som forklart i kapittel 3, brukes offentlige (unike) IP-adresser på maskiner, og rutere, i Internett. I lokalnett brukes vanligvis private IP-adresser, som **ikke** er globalt unike. Rutere i Internett vil ikke videresende pakker adressert til/fra private adresser, og private IP-adresser vil derfor ikke "fungere" i Internett. Hvordan kan da maskiner med private IP-adresser i et lokalnett likevel kommunisere med maskiner i Internett?

Løsningen på dette er å bruke kantrutere med *adresseoversetting* – NAT (*Network Address Translation*). En slik *NAT-ruter* plasseres mellom lokalnettet og Internett. Ruteren vil ha en offentlig IP-adresse i Internett (WAN-porten) og en privat IP-adresse i lokalnettet. NAT-funksjonen i ruteren vil da oversette private IP-adresser i datapakkene fra maskiner i lokalnettet til ruterens offentlige IP-adresse før pakkene sendes ut på Internett. Alle pakker som sendes ut på Internett vil derfor ha ruterens IP-adresse som avsenderadresse. Når maskiner i Internett svarer på slik pakker, vil de derfor adressere svarpakkene med ruterens offentlige IP-adresse som mottaker. Når ruteren mottar disse, vil den erstatte mottakeradressen med den private IP-adressen til den endelige mottakermaskinen, før den videresender pakken på lokalnettet. For å gjøre dette korrekt, må ruteren lagre en *oversettelsestabell* over alle aktive TCP-forbindelser fra klienter i lokalnettet til maskiner i Internett. UDP bruker ikke forbindelser, men håndteres på liknende måte.

Maskiner i private IP-nett er altså «utilgjengelige» for alle andre enn maskinene i eget IP-nett, men kan likevel ta initiativ til kommunikasjon med maskiner i Internett. Maskiner i Internett, eller andre lokalnett koblet til Internett, vil **ikke** kunne etablere kontakt med maskinene i lokalnettet. NAT tillater altså klienter i lokalnettet å kontakte tjeneren i Internett, mens klienter i Internett ikke kan kontakte tjenere i lokalnettet. De private adressene i lokalnettet kan jo ikke brukes av klienten. Klienten i Internett kan forsøke å kontakte ruterens offentlige IP-adresse, men ruteren har da ingen informasjon om hvilken av maskinene på innsiden av NAT-ruteren som klienter ønsker kontakte med. Denne egenskapen ved NAT er i utgangspunktet en fordel fordi den gir maskinene i det indre nettet en god beskyttelse mot uautorisert tilgang fra maskiner utenfor nettet. Legg likevel merke til at NAT **ikke** gir beskyttelse mot

virus, trojanerangrep eller andre sikkerhetstrusler som skyldes at brukere i det indre nettet selv har kontaktet usikre tjenester i Internett.

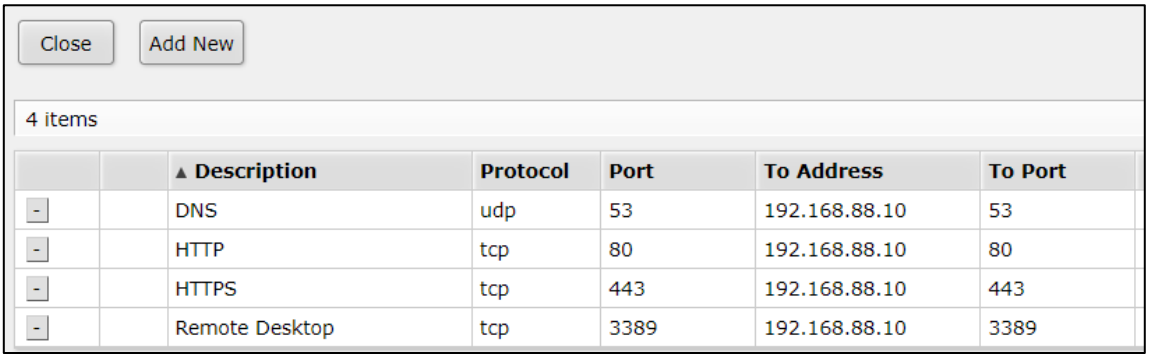
4.5.3. Portforwarding

Dersom man ønsker å sette opp tjenester/tjenermaskiner i et lokalnett med private IP-adresser, som skal nås fra maskiner i det ytre nettet, er NAT-funksjonen altså ikke tilstrekkelig. Derfor har mange NAT-rutere en tilleggsfunksjon beregnet for dette, og som har navnet *port-forwarding* på engelsk. På norsk kan det oversettes med "portbasert videresending", "port-videresending" eller "portviderekobling", men i praksis brukes det engelske begrepet mest.

Ved bruk av portforwarding, må den som administrerer ruteren manuelt registrere informasjon i en *forwardingtabell* i ruteren. Tabellen inneholder informasjon om hvilke private IP-adresser og portnummer som skal kunne nås fra klienter på utsiden av ruteren. Klientene må alltid bruke ruterens offisielle IP-adresse for å etablere kontakt med tjenere på innsiden. Ved å bruke flere ulike portnummer, så kan man likevel gi tilgang til flere tjenester og/eller maskiner på innsiden. Tabellen nedenfor gir eksempel på dette:

Offentlig side (utsiden)			**Privat side (innsiden)**	
IP-adresse	**portnummer**	**Protokoll**	**IP-adresse**	**portnummer**
158.64.2.75	80	TCP	192.168.52.10	80
158.64.2.75	443	TCP	192.168.52.10	443
158.64.2.75	81	TCP	192.168.52.12	80
158.64.2.75	444	TCP	192.168.52.12	443
158.64.2.75	3389	TCP	192.168.52.12	3389

Tabellen over viser en forwardingtabell som gir mulighet for å kontakte to ulike maskiner/IP-adresser, med private IP-adresser, fra klientmaskiner i det ytre nettet. Tabellen gir tilgang med *HTTP* og *HTTPS* til begge tjenerne, samt *Remote Desktop* (port 3389) til maskinen med IP-adresse 192.168.52.12. Tjenernes private IP-adressene (52.10 og 52.12) kan nås via NAT-ruterens offentlige IP-adresse (158.64.2.75) på utsiden. Legg merke til at alle portnumrene på offentlig side er forskjellige. Tabellen "mapper" altså hvert av disse portnumrene til en kombinasjon av IP-adresse og portnummer til tjenestene som skal kunne nås på innsiden av ruteren. Funksjonen kalles derfor også noen steder for *portmapping*.



Eksempel på portforwardingstabell fra en MikroTik-ruter.
Fordi ruteren bare har én offentlig IP-adresse på utsiden, er denne utlatt fra tabellen.

4.6. Brannmur

En brannmur er en viktig komponent for å ivareta sikkerhet i nettverk. Brannmurer kan være programvare som installeres på en enkelt maskin i nettet, og som bare beskytter denne ene maskinen. En slik brannmur kalles gjerne en *vertsbasert brannmur*. En brannmur kan også beskytte et helt nettverk. Da er brannmurprogramvaren installert enten i en ruter, eller i en egen separat boks, som gjerne kalles en *nettverksbasert brannmur*. En slik nettverksbasert brannmur bør regnes som en egen type nettverkskomponent.

Zyxel Nettverksbasert brannmur

Hovedoppgaven til en nettverksbasert brannmur er å filtrere bort uønsket nettverkstrafikk, slik at denne ikke slipper gjennom brannmuren. Derfor kalles brannmurfunksjonen også for et *pakkefilter* Brannmuren kan filtrere både *innkommende* og *utgående* trafikk basert på detaljerte kriterier, for eksempel trafikk adressert til bestemte IP-adresser eller portnummer. Disse kriteriene definerer vi som en del av *brannmurregler* (*firewall rules*). Vi kan lage regler for hva som er **ønsket** trafikk som skal slippe gjennom brannmuren. Motsatt kan vi også lage regler for **uønsket** trafikk som **ikke** skal slippe gjennom.

Brannmurregler kan være basert på flere ulike kriterier, og på kombinasjon av disse. Reglene og kriteriene bestemmer hvilke pakker som slippes gjennom brannmuren. Det mest vanlige er nok at reglene slipper gjennom pakker som er adressert til eller fra bestemte IP-adresser, eller portnummer. Reglene kan også baseres på bestemte protokoller, eller på hvilket program som har sendt pakkene (dataene). Noen brannmurer tillater også regler basert på fysiske MAC-adresser.

Kriteriene som er beskrevet over, baserer seg på informasjon på ulike lag i OSI-modellen. For å gjøre jobben sin, må derfor brannmuren lese informasjon i pakkehodene fra flere lag i datapakkene som den skal behandle. En brannmur er derfor en nettverkskomponent som utfører jobben på flere lag i OSI-modellen, og den kan ikke plasseres entydig i ett av lagene. Fordi en brannmur kan bruke informasjon fra høyere lag enn lag 3, vil du også kunne se at noen kaller den for en lag-4 komponent, eller en høyere lags komponent. En nettverks-komponent som jobber på høyere lag enn lag 3, kalles noen steder også for en *gateway*. Dette er et mindre presist faglig begrep, og det brukes ganske bredt om flere ulike komponenter som styrer trafikken ut og inn av et nettverk.

4.7. Sammendrag

I dette kapittelet har vi beskrevet flere nettverkskomponenter som har ulike oppgaver i et nettverk, og som utfører disse oppgavene på forskjellige lag i OSI-modellen. Figuren nedenfor viser disse komponentene plassert inn på det laget i OSI-modellen der de utfører hovedoppgaven sin:

<table>
<tr><th>OSI-lag</th><th colspan="2">Nettverkskomponent</th></tr>
<tr><td>Applikasjonslag</td><td></td><td rowspan="7">Brannmur</td></tr>
<tr><td>Presentasjonlag</td><td></td></tr>
<tr><td>Sesjonlag</td><td></td></tr>
<tr><td>Transportlag</td><td></td></tr>
<tr><td>Nettverkslag</td><td>Ruter</td></tr>
<tr><td>Lenkelag</td><td>Aksesspunkt
Svitsj</td></tr>
<tr><td>Fysisk lag</td><td>Hub</td><td></td></tr>
</table>

En *hub* kopierer fysiske signaler på lag 1 i OSI-modellen, og sender kopien ut på alle nettverksporter i huben. Huber brukes lite i moderne datanettverk, men brukes bl.a. i USB.

En *svitsj* leser datapakker på lag 2, og sender dem ut på den nettverksporten der mottakeren med riktig MAC-adresse kan nås.

Et *aksesspunkt* lager et trådløst nettverk og kobler dette til et eksisterende kablet nettverk. Aksesspunktet sender datapakker på lag 2 mellom disse to fysiske nettene.

En *ruter* kobler sammen to eller flere nettverk, og videresender datapakker på lag 3 til det nettet der mottaker kan nås. Ruterne finner også beste/raskeste vei (rute) gjennom nettet fra avsender til mottaker. Rutere som kobler lokalnett til Internett, kan ha NAT-funksjon som oversetter private IP-adresser til offentlige, og motsatt. Adresseoversetting med NAT er en forutsetning for at maskiner i lokalnett med private IP-adresser kan kommunisere med maskiner i Internett.

En *nettverksbasert brannmur* filtrerer datapakker på vei til eller fra et nettverk, og stopper uønskede pakker fra å passere brannmuren. Informasjon fra flere lag i OSI-modellen kan brukes som kriterier for hvilke pakker som skal slippes gjennom og hvilke som skal stoppes.

En "hjemmeruter" inneholder både en svitsj, et aksesspunkt og en ruter med NAT-funksjon. Dessuten ofte en DHCP-tjener og en webtjener for nettverksbasert konfigurasjon og administrasjon. Ruterfunksjonen er altså bare én av flere komponenter i en "hjemmeruter".

4.8. Oppgaver

1) Hva er forskjellen på en aktiv og passiv nettverkskomponent? Gi noen eksempler.

2) På hvilket lag i OSI-modellen utfører hver av disse nettverkskomponentene hovedoppgaven sin: hub, trådløst aksesspunkt, svitsj, ruter og brannmur.

3) Hva er hovedoppgaven(e) til hver av de samme nettverkskomponentene: hub, svitsj, trådløst aksesspunkt, ruter og brannmur.

4) Hva er hovedforskjellen på en hub og en svitsj, og hvorfor brukes huber lite i moderne datanett?

5) Når er det nødvendig å benytte en ruter i et datanett?

6) I hvilke tilfeller er det behov for å benytte en ruter med *NAT-funksjon* i et datanett?

7) Hva kan man oppnå ved å bruke funksjonen *portforwarding* i en NAT-ruter?

8) Hvilke funksjoner har en "hjemmeruter", og hvorfor er det faglig litt upresist å kalle den en "ruter"?

9) Forklar forskjellen på en *vertsbasert* brannmur og en *nettverksbasert* brannmur.

10) Hvorfor kalles brannmurer også for *pakkefiltre?*

11) Hva menes med en *brannmurregel*?