**1)**

**ARP står for Address Resolution Protocol**, og er en kommunikasjons protokoll. Når en enhet skal kommunisere med en annen enhet på et lokalt LAN trengs MAC-adressen til enheten som skal motta informasjonen. ARP brukes for å finne/få tilgang til enhetens MAC-adresse. Som vi vet fra tidligere er det IP-adressen som lokaliserer en enhet på et nettverk, men MAC-adressen som virkelig identifiserer enheten.

Skal en maskin A kommunisere med en maskin B på et nettverk trenger den altså MAC-adressen til maskin B. Det den først gjør er å sjekke en intern cache (minne) for å se om den allerede har indentifisert maskin B ved å koble IP-adresse og MAC-adresse. ARP-protokollen sender deretter ut en Broadcast beskjed (går til alle enheter på nettverket) som spør om hvem som har en spesifikk adresse. Maskinen som har denne IP-en som blir forespurt i Broadcast vil deretter svare med sin MAC-adresse (som blir sendt tilbake til maskin A). Kommunikasjonen kan deretter starte.

**2)**

**NAT står for Network Address Translation.** NAT blir hovedsakelig brukt i rutere, og har som hovedoppgave å oversette et sett med IP-adresser til et annet sett med IP-adresser. Som kjent er det allerede mangel på IPv4 adresser. Man trenger en public IP adresser for å koble seg mot internett, og problemet oppstår hvis alle enheter skulle hatt sin unike/egen public-ip adresse hadde det ikke vært i nærheten av nok adresser. Løsningen på dette var å gi alle enheter i et eksempelvis «hjemmenettverk» private IP-adresser.

De private IP-adressene kan ikke brukes til å aksessere internett med, og disse må derfor oversettes til public- IP adresser. Det er dette som skjer i ruteren, og ved hjelp av NAT. Den oversetter også andre veien (public-private). Dette kan eksempelvis bli nødvendig hvis en maskin på internett vil kommunisere med en maskin på et privat nettverk.

Det er verdt og merke seg at NAT-løsningen bare er gjeldende for IPv4- adresser da det er disse det er mangel på. I fremtiden, når IPv4 er faset ut vil alle enheter kunne ha sin egen unike public IP-address. Fordi IPv6 foreløpig har mer enn nok adresser til dette.

**3)**

For at dette skal fungere må Port Forward – metoden benyttes. Dette gjør det mulig for enheter på internett å koble seg til en datamaskin eller en tjeneste på et privat nettverk. Port forwarding redirecter kommunikasjon fra en IP, og port til en annen, mens datapakkene går igjennom routeren.

**4)**

*Disse forklaringene er gjort med støtte i pensum-boka. Side 358-360*

**IPv4 headeren** består av en rekke elementer.

**Version:** Første del av headeren spesifiserer hvilken versjon av internettprotokollen som blir brukt. Dette blir jo følgelig **IPv4**. Neste del (IHL) står for internett header length og spesifiserer lengden på IP-headeren i byte.

**Header length:** Lengden på selve headeren. (Dette er for å skille mellom hvor headeren og selve dataen). Typisk størrelse er 20 byte.

**Type of service:** Dette er for å skille ulike typer IP-pakker fra hverandre. Eksempelvis skille real-time-datagrams fra non-real-time traffic.

**Datagram length:** Dette er den totale lengden av hele IP-pakken. Altså både headeren og data til sammen. Disse kan i teorien være 65 535 bytes, men er sjelden større enn 1500 bytes.

**Identifier/flags/ fragmentation offset:** Disse handler i stor del om IP-fragmentering. Identifier er i korte trekk en identifikasjon på hvert enkelt fragment (som er det samme) slik at disse kan identifiseres med original IP-pakke.

**Time to live:** Dette er en viktig del av IP-headeren. Det fordi det blir spesifisert en viss tid pakken er gjeldende. (Altså for å forsikre seg om at den ikke kan sirkulere nettet i evig tid). Verdien i dette feltet blir trukket fra 1 hver gang den er innom en Ruter.

**Protocol:** Dette feltet er typisk bare brukt når pakken kommer frem til målet. Verdien spesifiserer hvilken transport-lag-protokoll dataen i pakken skal gis til.

**Header checksum:** Dette er en feilsjekk metode som blir brukt for å sjekke at all data har kommet frem. (Den ser etter bit-feil)

**Source and destination IP-adress:** Når en datapakke blir opprettet blir IP-en til host satt inn som sourc . Destination IP  blir ofte bestemt gjennom DNS-lookup.

**Options:** Dette muliggjør en utvidelse av IP-headeren. (Ikke ofte brukt)

**Data:** Den viktigste delen av IP-headeren. I de fleste tilfeller inneholder denne transport-lag-segmentet. (TCP eller UPD) som skal leveres til destinasjonen. Men den kan også inneholde andre typer data.

**5)**

*Disse forklaringen er gjort med støtte fra pensum-boka. Side 378-379*

**Version:** Dette er likt som IPv4. Forskjellen er her verdien i dette feltet, som oftest er 6. (For å indikere IP versjon 6)

**Traffic class:** Dette feltet består av åtte bit, og brukes til å prioritere en datapakke i en strøm av forskjellige datapakker. Det kan også brukes til å prioritere en pakke som kommer fra en spesifikk applikasjon

**Flow label:** 20 bit felt som brukes til å identifisere en datapakke i en strøm «flow» av datapakker.

**Payload length:** Dette er en 16 bit verdi som forteller hvor mange bytes datapakken er, etter headeren som er på 40 byte.

**Next header:** Dette spesifiserer hvilken protokoll som skal ta imot data-delen av en pakke. Eksempelvis (TCP og UPD). Dette feltet bruker samme verdier som i protokoll feltet på IPv4-headeren.

**Hop limit:** Likner på time to live (IPv4). Denne verdien synker med en for hver ruter datapakken er innom. Hvis denne verdien når 0 vil pakken bli kastet. (sett bort ifra)

**Source and destination adresses:** Dette er rett fram IP-adresse til source (avsender) og destination (mottaker).

**6)**

**Hvis vi ser på selve headeren først er det noen hovedforskjeller:**

Eksempelvis har vi ingen fragmentation/reassembly del. Det er oppdeling av store datapakker ikke er en mulighet i IPv6. Annet enn ved source og destination. Hvis en ruter som er imellom disse mottar en pakke som er for stor vil den sendes tilbake med beskjeden «Packet Too Big»

IPv6 har heller ingen header checksum. Dette er hovedsakelig fordi det er flere andre protokoller som utfører checksum operasjoner. For eksempel link-laget og ethernet. Dette ble derfor sett på som overfladisk i IPv6.

Options - feltet i IPv4 muliggjorde flere utvidelser av headeren. I IPv6 er det ikke lenger en egen del av headeren. Det er istedenfor en mulig **next header.** Så på lik linje som at next header kan være TCP og UPD, kan det også være et options felt.

Noen andre forskjeller er adresselengden. IPv4 er på 32 bit, mens IPv6 er på 128 bit. Størrelsen på headeren er også forskjellig. IPv4 headeren kan være 20-60 bytes, mens IPv6 er fast på 40 bytes.

**7)**

Det finnes flere måter å få IPv6 og IPv4 til å fungere sammen på. Problemet oppstår når to maskiner som kjører IPv6 adresser skal kommunisere over IPv4 baserte rutere. For selv om IPv6 blir mer og mer gjeldende er fortsatt store deler av internett IPv4 basert.

**-Tunneling:** Dette prinsippet fungerer slik: Hvis en host A (IPv6) skal sende en datapakke til host B (IPv6) over et nettverk som benytter seg av IPv4 protkollen blir hele datapakken (header og data) pakket inn i datadelen av en IPv4 pakke før den blir sendt. Den kan da bli sendt over IPv4 nettet.

**-Dual stack:** Dette er et prinsipp som gjør det slik at både IPv4 og IPv6 protokoller benyttes til overføring av data samtidig. Med andre ord: de eksisterer ved siden av hverandre. Routing protokollene og ulike konfigurasjoner kan dermed håndtere begge deler.

**8)**

AS står for autonomous system og er en stor samling av nettverk som har samme «routing policy». Enhver enhet som kobler seg til internett er også en del av et AS. Internett består altså av flere slike AS, og blir ikke styrt av et spesifikt selskap. Du ulike AS-ene blir styrt av flere forskjellige selskap. Eksempelvis kan for eksempel Apple og Google ha hvert sitt AS på internett. All trafikk innenfor dette AS-et er dermed deres ansvar.

Alle routere innenfor et AS må bruke samme routingsalgoritme og routingsprotkoll. Algoritmen kalles også «Intra- AS routing aloritme». I grenseland mellom to AS (der ruterne i de respektive har tilkobling til hverandre) brukes protokollen Border Gateway Protocol.