**Application layer:**

Hvad er det?

* Applikationslaget er oversagen til at brugeren kan få information via netværket igennem nogle programmer. Dette lag er brugerinterfacet til programmet og der igennem til netværket.

Hvad gør det?

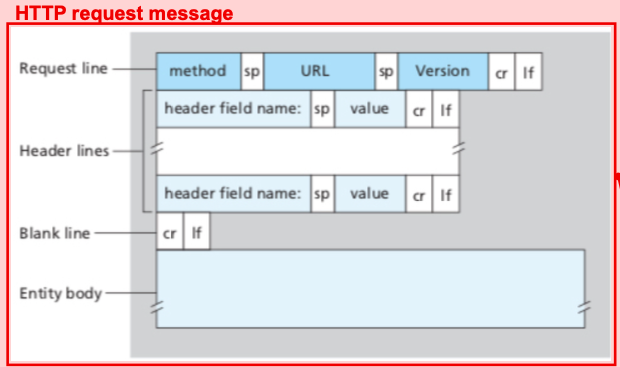
* Laget skaber en almindelig standard for hvordan internet applikationer skal kommunikere med hinanden. Et eksempel kunne være webbrowseren, hvor Tim Berners-Lee lavede en række regler for hvordan web browser skal kommunikere mellem hinanden. Disse regler hedder protokoller.

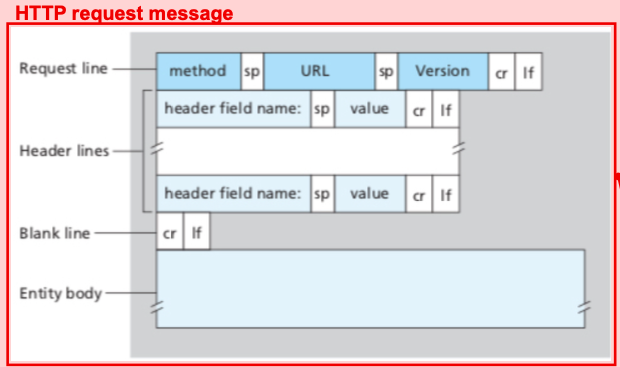
Protokol:

* HTTP (HyperText transfer protokol)
* FTP (file transfer)
* SMTP (simple mail transfer protokol)
* DNS (Domain name system)
* Dens opgave er at give en IP-adresse et domaine navn, denne proces kaldes name resolution
* Recursive query er mellem DNS klinet og Local DNS Server
* Iterative query er mellem local DNS Server og andre DNS server

HTTP struktur:

* Request line
* Mehtod (POST, GET, PUT, DELETE)
* URL (endpoint f.eks. google.com)
* Version (HTTP/1.1)
* Sp (one single white space character)
* Header lines
* Header field name (host)
* Value (det man gerne vil forbinde til f.eks. google.dk)
* Blank line
* Hvis der skal mere data i HTTP request kan det blive tilføjet efter blank line





**Transport layer:**

Dens rolle er at tilbyde kommunikation service direkte til applikations processer kørende på forskellige hosts. Denne lag konventer applikationslagets besked til transport layer segments. Dette sker ved at applikationsbeskeden bliver delt i mindre bidder, hvor der bliver tilføjer en transport layer header til hver bid for at lave en transport layer segment.

**TCP-segment:**

Et billede, der indeholder bord

Automatisk genereret beskrivelse

TCP-segments består af header fields og data field (består af mindre bider fra applikations data, som kan være http-request).

32-bit sequnce number og 32-bit acknowledgment number bliver brugt af TCP sender og modtager for at implementere en stabil overførelses service.

16-bit checksum som bliver brugt til fejl undersøgelse.

16-bit receive window bliver brugt til flow control

**3-way handshake:**

Før at man overhovedet kan overføre data med TCP, så skal man først oprette en forbindelse.

For at få en forbindelse skal man først lave en three-way-handshake. Serveren spørger først om lov, hvor den får en godkendelse tilbage og så sender den pakken. Hver gang der bliver sendt en pakke, får man svar tilbage fra modtageren.

SYN = syncronized = i want to talk with you

ACK = acknowledgment = sure, OK

Et billede, der indeholder tekst, whiteboard, person

Automatisk genereret beskrivelse

Retransmission:

* Hvis pakket ikke bliver anerkendt hos modtageren inden for en bestemt tid, så formoder den at pakket er forsvundet og derfor sender en ny

Congestion control:

Flow control

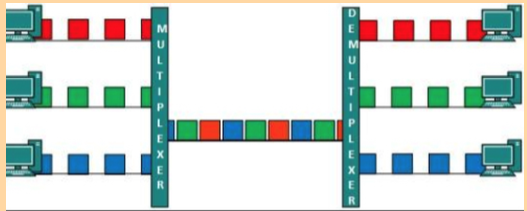
* Betyder at afsenderen ikke sender for mange pakker til modtageren på engang, især hvis reciever buffer allerede er fuld.

**TCP Socket:**

* Det er Source port nummeret og source IP-adressen som til sammen giver disse to endpoint et unikt endpoint

**Multiplexing, de-multiplexing:**

* Bliver brugt i både TCP og UDP
* **Multiplexing** er når transport layer håndterer data fra sockets og tilføjer en transport header til hvert segment, så den kan blive sendt til netværks layer, så den kan blive modtaget hos modtageren
* Demultiplextion bruger header informationen til at levere modtaget segments til den rigtig socket på modtageren side.



**UDP-segment:**

Et billede, der indeholder bord

Automatisk genereret beskrivelse

Denne protokol leverer ikke en pålidelig dataoverførelse og er forbindelsesløs, derfor indeholder den også færre header fields i UDP-strukturen end TCP. Den er meget hurtigere end TCP

UDP chechsum er for fejl undersøgelse.

UDP har mindre pakker end TCP, da de kun er 8 bytes og tcp er 20 bytes

Pakker bliver ikke modtaget i den rækkefølge som de blev sendt i

**Protokoller:**

* TCP
* UDP
* SMTP
* FTP

**Network layer:**

Netværkslaget primær funktion er at flytte data til og igennem andre netværker. Dette bliver udført ved hjælp af protokoller ved at pakke data med den korrekte netværksadresse information, og ved at vælge de meste relevante netværk routers.

De to vigtigste opgaver i netværkslaget er:

* Routing (indeholder alle netværks routers som via routing protokoller bestemmer vejene som datagrammet tager på deres vej fra source til destination node)
* Forwarding (indeholder transport af datagram fra en indgående link til en udgående link i en single router)

Alt i alt så kan netværkslaget gør det muligt at logisk at forbinde to internet forbundet enheder.

**IP = Internet protokol**

* IP tager data segments fra transportlaget og deler det i datagrammer. IP-datagram indkapsledes data modtaget fra laget over og tilføjer dets eget header information til det. det indkapsledes data bliver refereret som **IP Payload**.

**IP-adresse**

* Et internet protokol adresse er en numerisk label som er tilskrevet til hver enhed forbundet til et computernetværk som anvender internetprotokollen for kommunikation.
* En IP-adresse har 2 funktioner: 1) host eller netværk interface identifikation: 2) location adressering.

**DHCP = Dynamic host configuration protocol**

* DHCP er et netværk management protokol som bliver brugt på IP-netværker hvorpå a DHCP server dynamisk giver en IP-adresse og andre netværk konfiguration parametre til hver enhed på et netværk, så de kan kommunikere med andre IP-Netvætker.

**BGP = Border Gateway Protocol**

* BGP er en routing protokol som er designet til at forveksle routing og tilgængeligheds information blandt autonome(selvstædige) systemer på internettet.

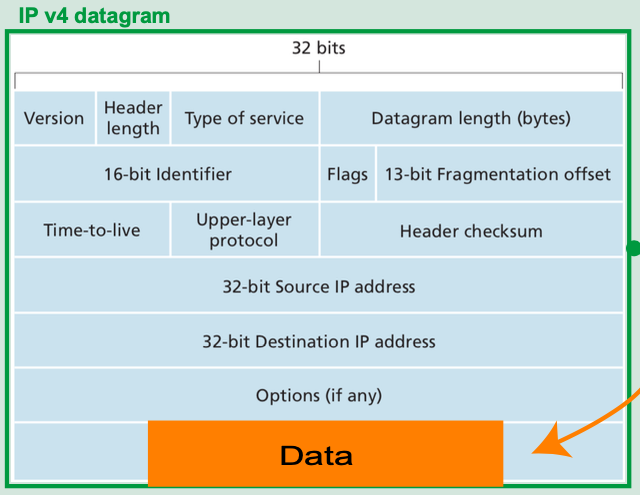
**RIP = Routing information protocol**

* RIP er en indretnings gateway protokol designet til at distribuere (sende ud) routing information inden i et autonome system

**Router**

* Router er et netværks enhed som viderefører datagrammet mellem computer netværker.
* Routers udfører trafik dirigering funktionen på internettet

Subnet



Version:

* Version nummer af internetprotokollen som bliver anvendt

Header Length:

* Længden af hele IP header

Type of service:

* Til at kende forskel på real-time datagrammer og non-real-time traffic (f.eks. FTP)

Datagram Length:

* Længden af hele IP-datagrammet (med IP header og IP Payload)

16-bit identifier:

* Hvis IP-datagram bliver opdelt i mindre elementer i løbet af transmisson, så indeholder alle dele den samme identifikations nummer til at identificere den originale IP-datagram som de tilhører hos.

Flags:

* Hvis IP-datagrammet er for stort til at blive håndteret, så fortælle flags om de kan blive delt i mindre elemneter.

Fragment Offset:

* Denne offset fortæller den præcise position af fragmenter (mindre stykke som er blevet adskilt) i den originale IP-datagram.

Time to Live:

* For at undgå at et datagram kører rundt i netværket for evigt, så er hvert datagram sendt med en TTL value set, hvilket fortæller netværket hvor mange routers datagrammet kan krydse. For hvert kryds falder dens value indtil den når til 0, hvor datagrammet bliver kasseret.

Protocol:

* Fortæller netværkslaget i destinations host hvilket protocol dette datagram tilhører til

Header checksum:

* Bliver brugt til tjekke hvis datagrammet bliver modtaget fejlfrit

Source Address:

* Senderens IP-adresse

Destination Address

* Modtagerens IP-adresse

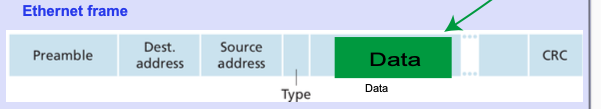
Options:

**Data-link layer:**

Dette lag er protokol laget som overfører data mellem netværk nodes i en bred netværk (WAN) eller mellem nodes på den samme lokale netværk (LAN) segment. Data-link laget giver funktionel midler til at overføre data mellem netværk enheder og kan give midler til at opdage og ordne fejl som kan ske i det fysiske lag.

Data link layer er bekymret med lokal levering af frames mellem nodes på det samme level på netværk.

**Frame:**



Preamble (8 bytes)

* De første 7 bytes i preamble har værdien ”10101010”, den sidste har værdien ”10101011”
* De første 7 bytes er for at vække de modtagende adapters og synkronisere deres ure til senderens ur

Destination address (6 bytes)

* Indeholder mac adressen for destinations adapteren

Source address (6 bytes)

* Indeholder MAC adressen for adapteren som udsender framet til LAN

Type field (2 bytes)

* Tillader Ethernet til at multiplex network-layer protokoller.
* Type field er næsten det samme som protokol field i netværkslagets datagram og port nummer fields i transportlagets segment.

Data field

* Dette field bæger på IP-datagrammet.

Cyclic redundancy check

* Formålet med denne er at tillade den modtagende adapter til at opdage bit fejl i framet

**MAC address = Media access Control:**

MAC adresse af en enhed er en unik identifikation tildelt til netværks interface controller for kommunikation på data link layer af en netværks segment.

Venstre del af MAC adresse er tildelt til sælger og højre side er brugt af sælgeren som serial nummer.

Protokoller:

**Ethernet protocol:**

* Ethernet er familie af computernetværk teknologier som er brugt i local area networks (LAN)

**ARP = Address Resolution Protocol**

* ARP er et kommunikation protokol som bliver brugt til at finde data link layer adresse, som MAC adresse

**ARP table**

* Hver host og router har en ARP tabel i den hukommelse, hvilket indeholder kortlægning af IP-adresser til MAC adresser.

**Switch**

* Et netværk switch er en computernetværks enhed som forbinder enheder sammen på et computer netværk ved at anvende frame swiching til at modtage og videregive data til destinations enheden
* Hver enhed som er forbundet til et switch port kan overføre data til de andre porte til en hver tid og transmissioner forstyrre ikke

**Ethernet Card**

* Network interface controller er en computer hardware komponent som forbinder en computer til et computernetværk
* NIC tillader computer til at kommunikere over et computer netværk ved hjælp af kabler eller trådløs. NIC er både fysiske lag og data link lag enhed.

**Physical Layer:**

Overfører bits til fysiske enheder

Dens opgave er at encode bits til et fysisk signal og decode bits fra et fysisk signal.

Typer af links

* Kabel
* Wireless

1. **Sockets, Porte, WebRTC, WebSockets**

Socket

* En socket er en software strukturer i et netværk node af et computernetværk som bliver brugt som et endpoint som sender og modtager data over netværket.
* En socket består af to element som til sammen giver et endpoint. Disse elemeter er IP-adressen og port nummeret. Dens primær opgave er at lave kommunikation mellem to maskiner/computer

Port

* Port er en kommunikation endpoint. Port svarer til et hus nummer

WebRTC

WebSockets

* Websocket protokollen giver en måde at udveksle data mellem browser og ser via en vedvarende forbindelse. Daterne kan vidergives i begge retninger som pakker uden at afbryde forbindelsen og yderligere http-anmodninger
* Websocket er især fantastisk til tjenester, der kræver kontinuerlig dataudveksling, f.eks. onlinespil

1. **Beskriv email og de mest kendte protokoller til det.**

* SMTP er en protokol som bruges til at sende e-mails. Når en email bliver sendt, sendes den til person A’s mail server f.eks. Gmail. Herfra bliver den sendt til persons B’s mail server måske yahoo

SMTP bruger TCP for at sikre at e-mailen bliver sendt korrekt

Port 25

* POP downloader email til din enhed fra en mail server, hvor den efter sletter mailen fra. Serveren. Den henter kun mail til inbox
* IMAP tillader at se email som er på serveren fra flere enheder. E-mailen bliver gemt på serveren og kun sender kopier til de andre enheder. Synkroniserer det hele

1. Beskriv TCP protokollen.
2. UDP vs. TCP med fokus på UDP og versus.
3. Peer to Peer protokollen.
4. HTTP requests. GET, POST osv. Plain text format.
5. IPv4 vs. IPv6, packets og headers.
6. ARP, MAC, Private IP-adresser
7. Vælg 2 teknologier i det fysiske lag og fortæl om dem.
8. Snak om 2 typer sikkerhedsmæssige trusler i et netværk.