Forelæsningsnoter

Mikkel Boye Rasmussen

23.02.2023

Internet protocol stack
- application: - tranport: ansvar for transport af applikationslagets indhold - network: ting havner det rigtige sted - link: den direkte forbindelse mellem devices

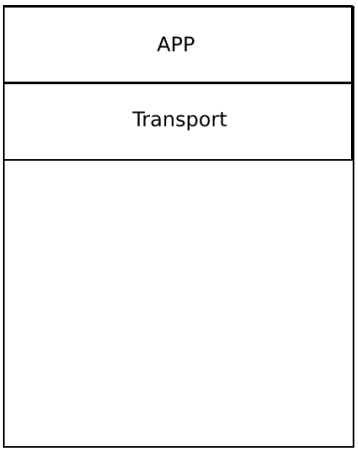
Encapsulation
switch behøver ikke IP - kigger kun på MAC addresse
HTTP er "stateless" - dvs vi gemmer ikke states by design. (tilføjet cookies til at huske states senere)
TCP anvendes til forbindelsen

HTTP request fra client til server (filhentning)
består af en metode (GET/POST...) resourcen (hvilken side skal det hentes fra) protokollen efterfulgt af \r \n
r = carriage return n = newline
headers - language, accept, encoding etc.. encoding tilhører præsentationslaget - derfor i 5-lags modellen er den i applikationslaget
requesten afsluttes med dobbelt \r \n

Http Response
består af protokol, status kode,header, data fra den requestede fil

DNS - domain name system
benytter UDP
DNS distribueret og hierakisk database
DK delen = TLD - top level domain
En DNS request spørger altid en root dns server dernæst en authorative server og den kan dernæst finde hjemmesiden man leder efter
den første spørger efter eksempelvis dk, den næste ved noget om underdomænet, og til sidst er man i kontakt med authorative server som ved noget om den specifikke IP.

Transport
TCP: connection oriented, reliable, langsom, congestion control, flow control
congestion control: der sker ikke forstoppelse, intet pakke-overflow. (hvis ikke så mange ack - så sæt sendehastigheden ned) alt afhængig af implementeringen af tcp, kan et antal af fejlne pakker specificeres før hastigheden halveres.
TCP er fuld duplex (betyder, kan både sende og modtage)
flow control: modsagerens specificering af hvor mange pakker de kan modtage af gangen.

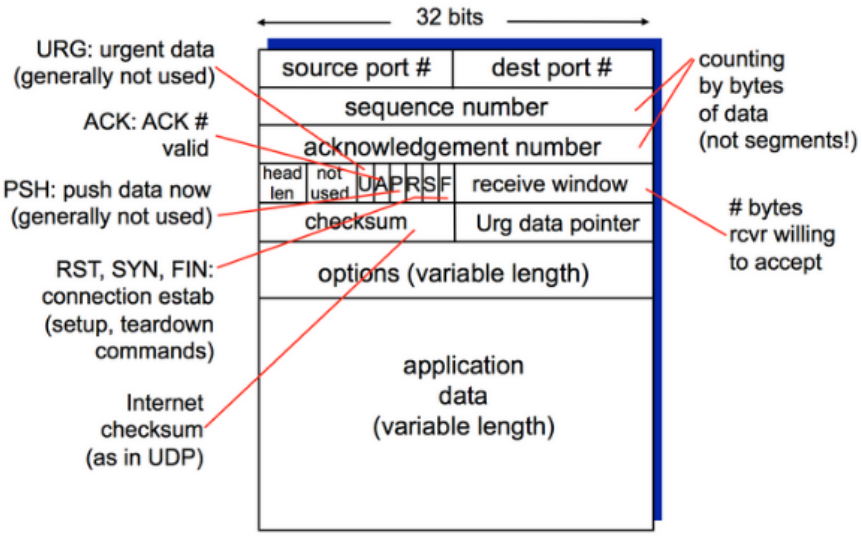


HTTP: 80 -> TCP
DNS: 53 -> UDP

TCP: connection-oriented, reliable, langsom, congestion og flow control.

UDP:

TCP Segment structure



S - SYN
A - ACK
R - Reset
wireshark giver os et relativt squence number for at gøre vores liv nemmere

Mere info: Computer networking topdown approach - ligegyldigt hvilken version

I wireshark: tcp.flags.syn==1 → follow tcp stream

TLS - en tand over tcp (i konteksst af sikkerhed i kommunikation)

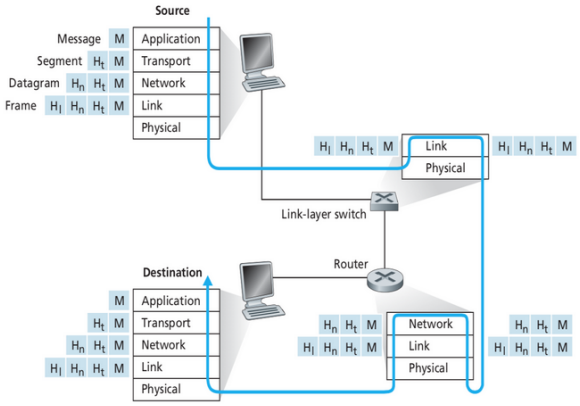
To forskellige tilgange til TCP
kumulatativ acknowledge: acknoledger altid pakker i rigtig rækkefølge hvis en pakke ikke bliver modtaget, genstartes processen fra sidst aknowledget pakke.
selective repeat: hvis en pakke ikke modtages smiddes efterfølgende pakker i buffer, således de ikke behøves overføre alting igen.
SACK - selective acknowledge (kig i begge klienters TCP options, hvis begge har SACK, anvendes SACK)

UDP segment header
source port, dest port
length, checksum
hvorfor vælge UDP frem for TCP (eksempelvis streaming)
nyt protocol kaldet quick der kører på UDP

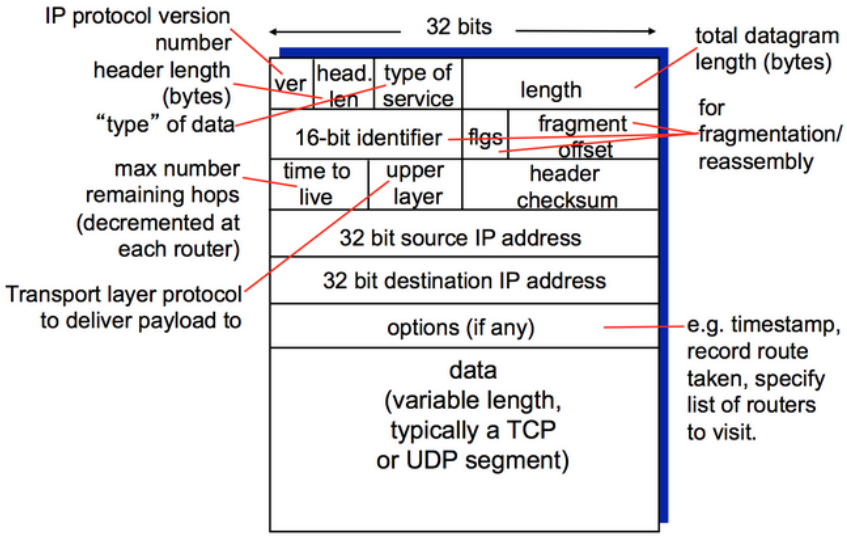
Network layer

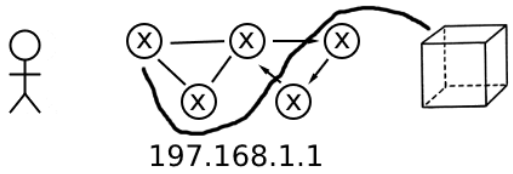
- Transport segment from sending to receiving host.
- Sending side: encapsulates segments into datagrams.
- Receiving side: delivers segments to transport layer.

- Network layer protocols in ***every*** host and router.
  - Router examines header fields in all IP datagrams passing through it.



IPV4 datagram format





$/24 = 2^8$
$/25 = 2^7$

How to get an IP?
kan hardcodes hvis man vil sikre en fast IP ellers DHCP:
tildelingsprocessen hedder DORA (discover, offer, request, acknowledge)

- Also referred to as "plug-and-play" or "zeroconf" protocol.
- Host learns:
  - Its own IP address assignment (same IP address each time or temporary)
  - IP address lease time
  - Subnet mask
  - Address of its first-hop router (default gateway)
  - Name and IP address of its local DNS server.
- 4-steps process:
  1. DHCP server discovery (using UDP from port 68 to port 67)
  2. DHCP server offer (using UDP from port 67 to port 68)
  3. DHCP request (using UDP from port 68 to port 67)
  4. DHCP ACK (using UDP from port 67 to port 68)

- This process is also called DORA (**D**iscover, **O**ffer, **R**equest, **A**ck)

NAT (Network Address Translation)

IPv6
hvordan skal ipv4 og ipv6 snakke sammen?
tunneling: hvis en router ikke forstår ipv6, er det afsender-routerens ansvar at oversætte til ipv4
essentielt smidder man ipv6 pakken i en ipv4 pakke

ICMP
man skal have en IP pakke for at have ICMP
bruges til ping og ICMP

Datalink
MAC-adresse - FORSKELLIG FRA IP
error detection and correction, der opdages pakkefejl her istedet for når de er kommet frem
datalink foregår på OS samt netværkskort
hver 2 tegn i mac-addressen er 1 byte
hvert interface på et device har en MAC-addresse
de første 3 bytes er manufacturer

Both are store-and-forward:

- *routers:* network-layer devices (examine network-layer headers)
- *switches:* link-layer devices (examine link-layer headers)

Both have forwarding tables:

- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses