[**Chapter 1**](#_o8rtoi55k4pd) **3**

[Switching (packet and circuit)](#_9wg60wvrcsd9) 3

[Protocol layers (no informationos) se side 74 (pdf) 1.5](#_o6atb7qtexs2) 3

[Delays (transmission, propagation and queuing)](#_gui7tqoer3lu) 3

[Encapsulation](#_ufgifu90ch3c) 5

[**Chapter 2**](#_9c1z8r7rdzcp) **7**

[Define and explain application layer protocol](#_1goqdilrt9gh) 7

[Client/server versus peer to peer](#_94pinlkunihh) 7

[Http, DNS, FTP, SMTP](#_x07k8si5rxud) 8

[DNS (Domain Name System)](#_16867nuj8vv3) 8

[FTP (File transfer Protocol)](#_1ghdjlfziq27) 9

[SMTP (Simple Mail Transfer Protocol)](#_1b43ceic1x4x) 9

[**Chapter 3**](#_nt113adq6kcv) **10**

[Logical communication between two processes](#_ytw85t9v7s19) 10

[TCP versus UDP](#_jan0hgluwxoc) 10

[TCP Header](#_m2rw7zpxcfz9) 11

[Multiplexing and DEmultiplexing](#_3ub93fmzc3mj) 11

[Reliable data transfer](#_hhfh30on1i60) 12

[Flow- and Congestion control](#_gtauc7tt6h4) 15

[Flow control:](#_j0uutqmlen1k) 15

[Congestion control:](#_gwlfvliaam3h) 15

[**Chapter 4 (Network Layer)**](#_sycvmugs3l23) **16**

[Routing protocols (link-state, distance vector, inter-AS, intra-AS)](#_ffv2wirtkhfe) 16

[Forwarding, routing](#_t0n3t3eihy9x) 16

[Delay and loss within a router](#_wax695gjd42w) 17

[BGP (Border Gateway Protocol (version4)](#_sjf8w9vbuonq) 17

[IPv4 versus IPv6 (https://www.youtube.com/watch?v=aor29pGhlFE)](#_ns80v0oeuklb) 18

[Longest prefix matching](#_5jql4q9if31y) 19

[Tunneling](#_mhks8iuewcvx) 19

[DHCP](#_mfga3fni4ckh) 20

[NAT](#_1q064k2148fr) 21

[NAT Translation Table.](#_jru9brmvfgld) 21

[**Chapter 5 (Link Layer)**](#_snyx37pfhn3l) **22**

[Link layer services](#_7wy5908sdkbs) 22

[Broadcast link versus PPP](#_3d0yq31yfvcb) 23

[Error detection and correction](#_55zr2twkeote) 23

[CRC](#_oui98d8rd1pf) 23

[MAC (Multiple access protocol)](#_4tzt1yl0wu4b) 24

[Channel Partitioning](#_jvqpo1299vba) 24

[Random Access Protocoller](#_ajtyc7dtohc2) 25

[(Pure) ALOHA](#_sbjk3cq9alg1) 25

[Slotted ALOHA](#_4jf7issd3dlk) 26

[CSMA/CD](#_qdqewbqdx8af) 26

[CSMA/CA](#_bk06uoadskt8) 27

[“Taking Turns”](#_9dxfhexytmr8) 27

[ARP (Address resolution protocol)](#_ic27v0z3c2fx) 28

[Ethernet  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_](#_z756009loh2l) 28

[Vlan (trunking, trunk protocol, port security, ………..)](#_hrmrdgrrmlqg) 28

[Port Security:](#_jcwmq6babdy) 29

[Trunk Port:](#_yeldtj12q8y6) 29

[**Chap. 6: (Wireless and Mobile Network)**](#_fw9t6gm34wha) **30**

[Wireless and mobility](#_jd9o14oyc6p) 30

[Base station](#_dy58u1c25wo8) 30

[Hidden node problem](#_34ubjk5rnxvo) 31

[Ad hoc mode and infrastructure mode](#_kyoma9tadd9u) 32

[Ad Hoc Mode:](#_odsozqttpa1i) 32

[Infrastructure mode:](#_dfh0looetz0i) 33

[CDMA (Code division multiple access)](#_za6e5ygtop19) 35

[802.11 standards](#_2ykpjiwk9rx2) 36

[Direct and indirect routing](#_o7ew21kde3ju) 37

[Handoffs](#_rar0n92znpu8) 37

[Home and foreign agents](#_v4x586kyeanm) 38

[**Chapter 8**](#_bws3we2wrly) **39**

[Hashfunktion](#_shhi4dqh2yy6) 39

[Symetric/shared key](#_t9v8usqsqzd) 39

[Public og private key](#_tvkejwdd0h6u) 39

[Nonce](#_nk4e76xx5mgt) 39

[Secure email](#_rlrnhb3c0u5r) 40

[VPN](#_pwexlm3zll75) 41

[Securing TCP connections: SSL](#_1phw6f8gvxhy) 41

[Wired Equivalent Privacy (WEP)](#_n3nkeo3e169z) 42

[I EEE 802.11i](#_omlkf56dvuux) 42

# Chapter 1

## Switching (packet and circuit)

Circuit switching har en dedikeret linje hvor samtlige pakker bruger samme rute. Circuit switching reserverer hele båndbredden, hvilket betyder, at circuit switching “spilder” båndbredden.

Packet switching har ikke en dedikeret rute. Pakkerne er uafhængige og finder selv vej til modtageren. Dette gør de via information der er blevet gemt i headeren. Packet switching reserverer ikke båndbredden, så derfor “spilder” den heller ikke noget.

## Protocol layers (no informationos) se side 74 (pdf) 1.5

## Delays (transmission, propagation and queuing)

Det bliver udregnet ved at sige l/r.

l = length (hvor meget det fylder)

r = rate (hastighed, bits pr sekund)

Det bliver udregnet ved at sige d/s.

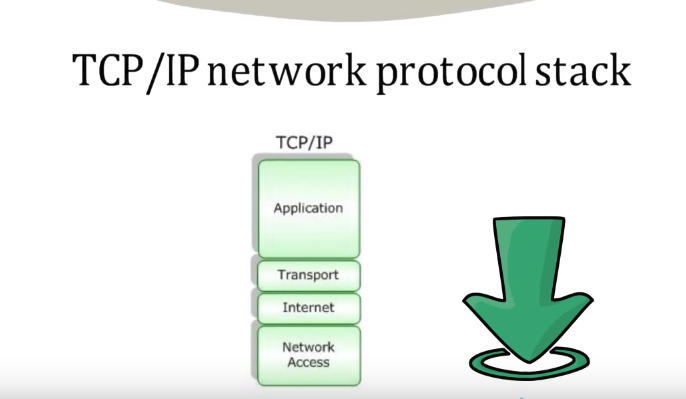
d = distance

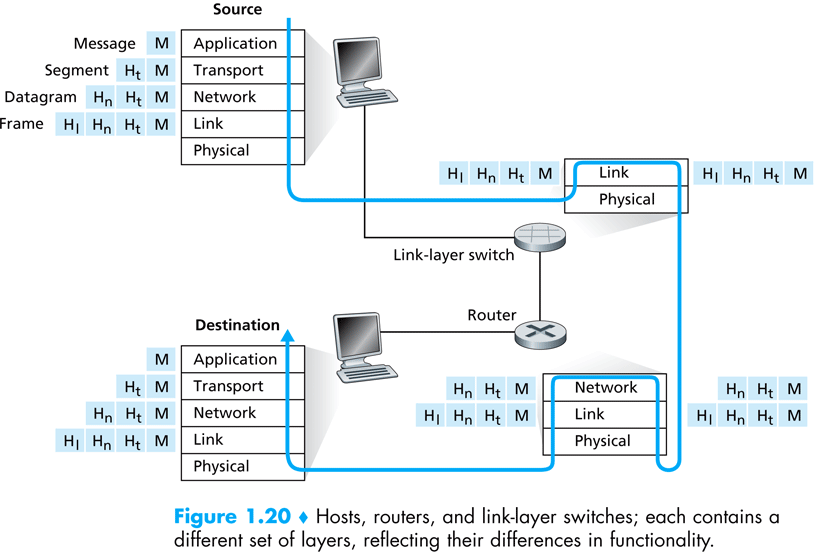
s = speed

Forskellen på Transmission delay og propagation delay er, at transmission delay er den tid det tager for dig at sende din pakke afsted, og propagation delay er den tid det tager for pakken at komme frem.

Queuing delay er den delay der forekommer når pakkerne kommer hurtigere end routeren kan håndtere dem. En router kan kun håndtere en pakke af gangen, de andre bliver sat i queue (eller bufferen).

## Encapsulation

<http://study-ccna.com/encapsulation/> (!)   
  
TCP / IP Network protocol stack;  


Encapsulation er processen der bryder data op i flere segmenter og tilføjer headers og trailers, så det bliver nemmere og mere overskueligt at sende som fil(er). Derefter gives hvert segment et sekvensnummer, og til sidst tilføjes der er en transport header til hvert segment. Udover transport header, bliver der efterfølgende tilføjet en internet header, med tilsvarende information til internet layeret.  
  


# Chapter 2

## Define and explain application layer protocol

Applikationslaget er det lag som brugeren interagerer med. I TCP/IP modellen finder vi applikationslaget som det 5. og sidste lag. En applikation kan f.eks være Facebook, her logger brugeren ind, sender en message og så kører den ned igennem de andre lag, transport, netværk, datalink og fysisk. Protokollerne vi finder i applikationslaget er f.eks Http,dns, dhcp osv.

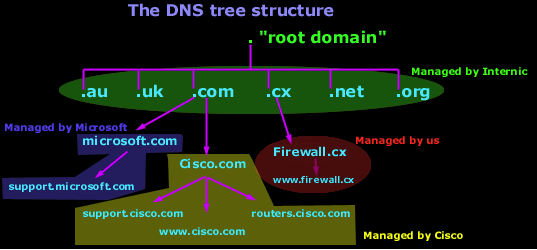
## Client/server versus peer to peer

Client/server og peer to peer er begge to måder at kunne dele applikationer og filer mellem flere enheder. Client/server fungerer på den måde, at en server har en fil du skal bruge, du opretter en forbindelse og downloader filen. peer to peer kan f.eks være et torrent. du downloader fra andre klienter samtidig med, at du uploader til selvsamme klienter. På den måde downloader du og uploader du samtidig.

## Http, DNS, FTP, SMTP

HTTP (Hypertext Transfer Protocol)   
 - En application protocol som bruges indirekte af alle brugere, lige så snart de bruger deres web browser til at gå på internettet. Det bruges primært til at overføre og læse tekst, billeder, lyd, video og andre mediefiler. Det er en protokol der bruges i det øverste lag af TCP/IP protokollerne. Web serveren fungerer derfor som en HTTP klient, som sender requests til servere. Det vil sige når en bruger indtaster en URL, eller åbner et hypertext link (link lol) konverteres dette til et HTTP request, som sendes til URL adresse IP adresse.

## DNS (Domain Name System)

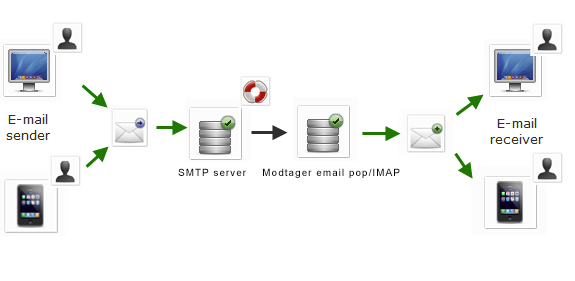
(<http://www.firewall.cx/networking-topics/protocols/domain-name-system-dns/158-protocols-dns.html> )  
 - DNS servere er en hierarkisk distribueret database. Det vil sige at dets “layers” er definitivt organiseret, og dets data er distribueret over mange maskiner (servere).   
Basically fungerer DNS sådan, at det gennem en DNS adresse (.com etc) refererer brugeren over til den rigtige IP adresse. Alt dette går igennem root domains, som er administreret af InterNIC.   
  


### FTP (File transfer Protocol)

- FTP er en klient-server- protokol der bruges til at overføre filer mellem to computere. På værts maskinen skal der være installeret en FTP-server, og på den tilsluttende maskine skal der være en FTP-klient.

### SMTP (Simple Mail Transfer Protocol)

- SMTP, ligesom HTTP protokollen, arbejder i toppen af TCP/IP laget. For standard kommunikation mellem to brugere, anvendes TCP port 25. SMTP er dedikeret og benyttet af alle standard email systemer, og bruges til at sende emails mellem to brugere. Den mail man afsender, har en SMTP adresse i sig, som sender mailen til en SMTP server, hvorefter den bliver videresendt til en POP(3)/IMAP server.



# Chapter 3

## Logical communication between two processes

Forbindelser behandles logisk, hvilket vil sige at netværket i princippet behandler forbindelsespunkter ens, om så der er 2000km bjerge imellem, eller om de er på samme netværk 5 meter fra hinanden.

## TCP versus UDP

TCP og UDP er de 2 protokoller man bruger når man snakker om transport af data. Begge protokoller er bygget oven på ip protokollen. Det vil sige, at lige meget om du bruger den ene eller den anden bliver de sendt til en ip adresse. den store forskel kommer ved håndtering og levering af pakkerne.

TCP bliver kaldt for reliable. Dette betyder, at der er et three-way-handshake inden pakken sendes afsted.

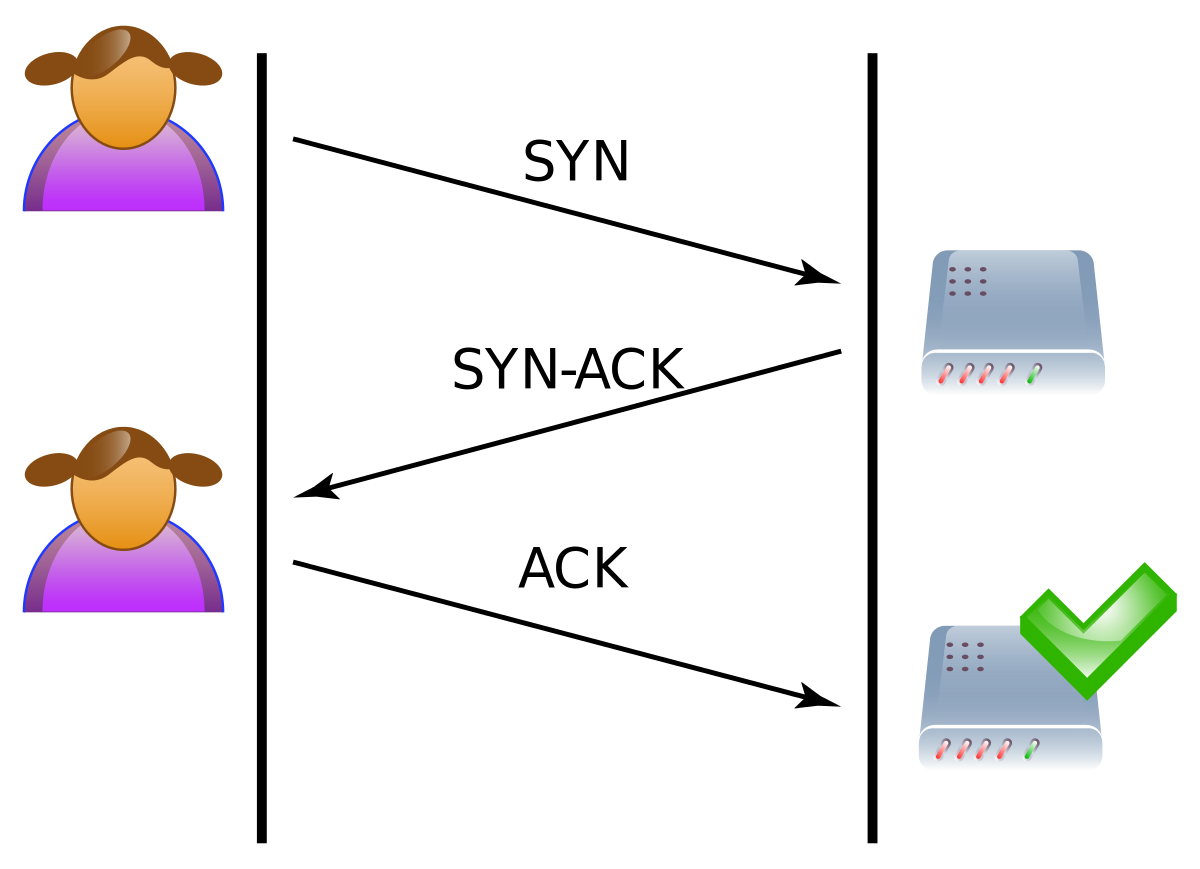
1. afsenderen sender en SYN (Synchronize)
2. Modtageren sender en SYN-ACK (Acknowledge) for nu at sige, jeg har modtager din SYN her er min
3. Afsenderen sender til sidst en ACK for modtagerens SYN

På denne måde etablere man en forbindelse inden selve dataoverførslen. Dette resultere dog i en stor header, og derfor også en langsommere dataoverførsel. Men kan derfor sende fejl-pakker igen.

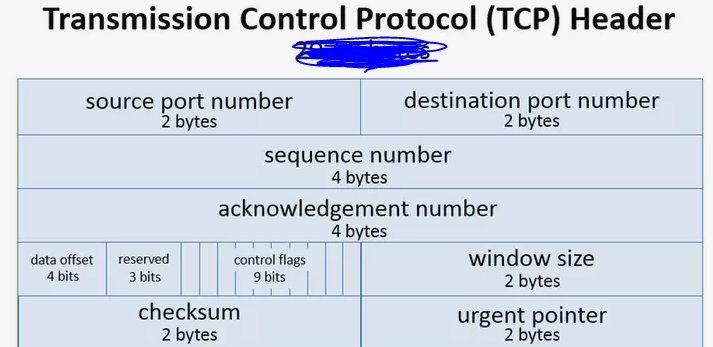
UDP har ikke en måde at etablere forbindelse inden dataoverførsel. Den smider alt form for fejl-tjeknings ting væk, og sender det hele afsted så hurtigt som muligt. Dette resultere i en lille header, men en hurtig dataoverførsel. Hvis en UDP pakke fejler, er det om igen.

UDP er derfor genialt til live-tv, streaming og voicechat.

TCP er genialt til filoverførsel.



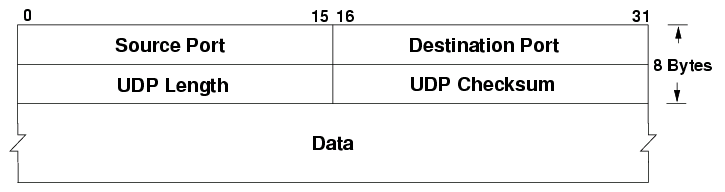
## TCP Header



En TCP header fylder 20 bytes, og består af:

1. Source port number = porten fra afsender
2. destination port number = porten fra modtager
3. sequence number = Hvor meget data der er blevet sendt under TCP sessionen
4. Acknowledgment number = Brugt af modtageren til at requeste det næste TCP segement.
5. Data offset (DO) = header length. Hvad fylder TCP headeren, så man ved hvornår dataen begynder.
6. Reserved (RSV) = reserveret data, som altid står ubrugt og sat til 0.
7. Flags = Bruges til at etablere forbindelse, sende data og afbryde forbindelser. Under flags er der yderlige 6 underpunkter som enten kan have værdien 1 eller 0, alt efter om de er aktive eller ej. Underpunkterne er:
   1. Urgent pointer (URG) = Denne data skal prioriteres over alt andet data.
   2. ACK = som er acknowledgement
   3. Push funktion (PSH) = dette fortæller en applikation at dataen skal transmitteres med det samme, og man behøver ikke at vente på næste TCP segment.
   4. reset the connection (RST) = AFbryde forbindelsen med det samme. Der er uhelbredelige fejl.
   5. SYN = bruges til handshake samt sætte sequence number
   6. Finish (FIN) = Dette er den sidste bit. Begge parter skal sende en FIN
8. Window Size = Dette felt bruges til at fortælle hvor mange bytes modtageren er villig til at modtage. Så kan afsenderen bruge den til at fortælle at han/hun gerne vil sende mere.
9. Checksum = bruges til at validere headeren (ligesom i hashing).
10. Urgent pointer = Urgent pointer bruges kun hvis URG Flag er oppe. Dette fortæller hvornår denne vigtige data ender.

## UDP Header

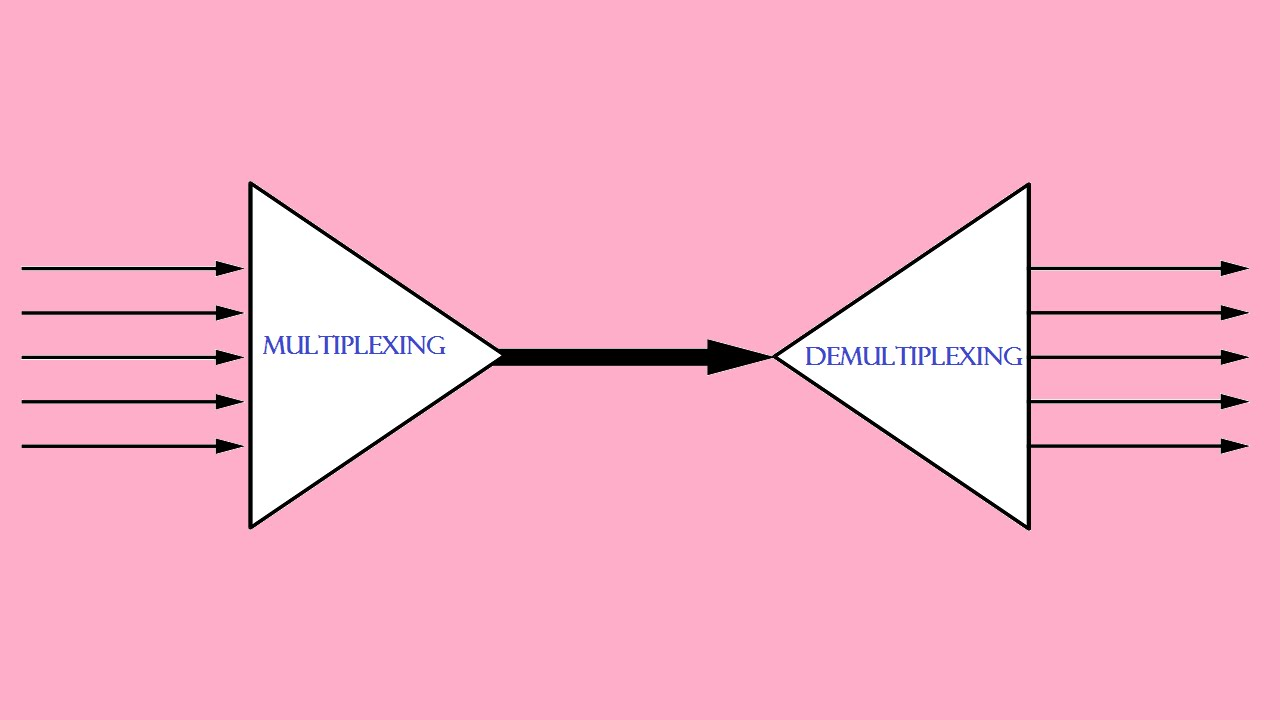


UDP header er ikke i nærheden, når det kommer til komplexitet, af TCP headeren. UDP headeren indeholder kun 4 felter, og de 4 felter indeholder tilsammen 8 bytes:

1. Source Port Number = porten på afsender.
2. Destination port number = porten på modtager
3. Length = total størelse af UDP pakken, indeholder både header og payload.
4. UDP Checksum = Det samme som i TCP, en validering af data.

## Multiplexing and DEmultiplexing

Multiplexing er processen, hvori data fra flere forskellige applikations processer, bliver konverteret til en samlet pakke, og får givet en header. Multiplexing og demultiplexing, er en process der findes i alle netværker, så den er født globalt ind i et netværk. Den sørger for at de forskellige applikationer processer bliver konverteret, får en header og sendt afsted til modtageren. Modtageren modtager derefter den komprimerede pakke. Demultiplexing er den omvendte process, der bliver brugt af modtageren af de data.   
(Se billede nedenfor)



## Reliable data transfer

TCP laver rdt service (reliable data transfer service) oven på IP’s unreliable service.  
TCP’s reliable data transfer service garanterer at den data stream som en proces læser når den modtager data, som er sendt gennem TCP, er uden mellemrum, uden duplikering og in sequence. Dvs. at den byte strøm som sendes er præcis den samme som modtages i den anden ende.

TCP’s RDT gør brug af:

**Pipelined segmenter**

TCP gør brug af konceptet “sliding windows” når der sendes segmenter. Det vil også opretholde register over hvert segments som er blevet modtaget på den anden side.  
Derfor kan vi sige at når et segment er blevet sendt, så er det i pipelinen indtil at dens ACK er modtaget. Hvis ACK’en ikke er modtaget før at den timeouter, vil der udføres en retransmission - segmentet er derfor i pipelinen indtil at den er segmentet fuldføres.

**Cumulative acknowledgements**

Cumulative acknowledgements er at, det højeste sequence nummer i “receiver ACK” altid indikere det højeste sequence nummer, som er blevet modtaget sammen med alle dens forgængere.

Afsenderen bruger en enkelt timer som restartes når en pakke er sendt, og alle dens forgængere er blevet acknowledged - eller når et ACK ankommer fra en tidligere unacknowledged pakke.

Cumulative acknowledgement betyder at man har modtaget alt op til et givent punkt.

**Single Retransmission timer**

TCPs RDT har kun en *enkelt* Retransmission Timer, dette er grundet at selvom at der er flere transmitterede men ikke endnu Acknowledget segmenter, så kan TCP’s timer management procedurer stadigvæk kun gøre brug af en enkelt retransmission timer.

Når der er blevet retransmittet et segment, vil det forblive i retransmissions queuen - og retransmitteres indtil at det der modtages et ACK (og så ved vi at det er modtaget) - Dette vil dog kun foregå i et fastsat antal gange.

Retransmissioner triggers af :

**Timeout Events  
 Duplicate Acks**

Når der modtages data fra en applikation:

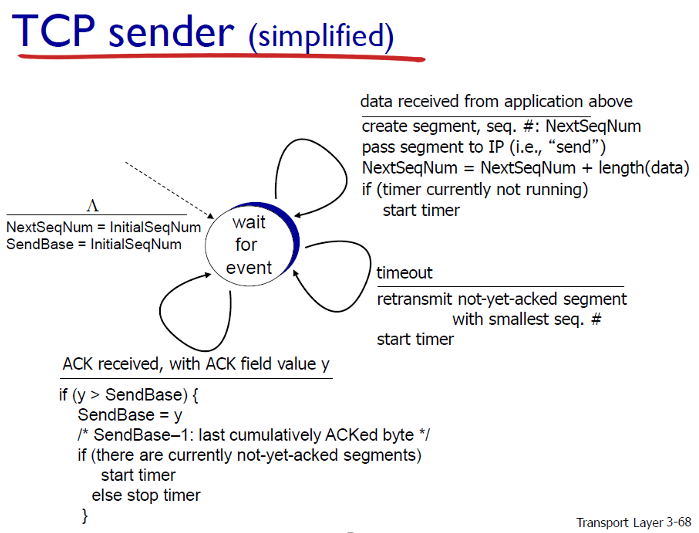
* Der laves et segment med seq #
* Segment # er byte-stream nummer af det første data byte i segmentet.
* Der startes en timer - hvis denne ikke allerede kører.
* Timeren er for hvis der er et gammelt unacket segment

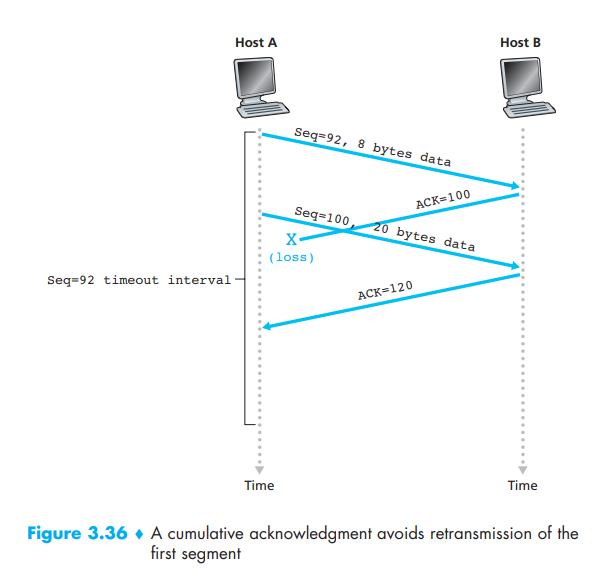
Hvis der timeoutes:

* Retransmit segmentet som skabte timeout
* Restart timer

Hvis ACK modtages:

* Hvis der ACK’es på et tidligere unACKed segment
  + TCPs RDT opdaterer hvad der den ved er blevet ACKed
  + Der startes en timer igen - hvis der stadig er unACKed segmenter.

****



## Flow- and Congestion control

(([https://stackoverflow.com/questions/16473038/whats-the-difference-between-flow-control-and-congestion-control-in-*tc*p?utm\_medium=organic&utm\_source=google\_rich\_qa&utm\_campaign=google\_rich\_qa](https://stackoverflow.com/questions/16473038/whats-the-difference-between-flow-control-and-congestion-control-in-tcp?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa) Stackoverflow svar ))  
(Flow control 277 - Congestion control 286)

### **Flow control:**

Det sikrer sig, at senderen ikke “overloader” modtageren. For eksempel hvis en afsender kører med fibernet forbindelse, og skal sende beskeder (pakker) til modtageren, som bruger en dial up forbindelse, så skal det sikres, at afsenderen ikke sender pakker for hurtigt, i forhold til hvad modtageren kan tage imod ad gangen.   
Basically forhindrer flow control en modtager i at blive flooded. Flow control er et fænomen som opstår fra afsenderen, og findes ikke globalt på alle netværk, men hos en afsender. (dvs i protokollen).

### Congestion control:

Modsat flow control, er congestion control noget der kører på netværket, frem for på en specifik forbindelse. Det sørger for at netværkets båndbredde er konstant og lige fordelt hos alle brugere af netværket. Dette er igen modsat flow control, et globalt fænomen, som findes på routeren, frem for individuelt. Det påvirker alle hosts forbundet til et netværk. Derudover, sørger det for at en afsender ikke sender for meget til routeren (med congestion control), så der ikke opstår flooding i routerens buffer(queue).   
Basically sørger congestion control for, at alle brugere på et netværk kan bruge deres del af ressourcerne til hver en tid, så at der ikke er nogle brugere der “optager” alt båndbredden, samtidig med at folk der sender til routeren ikke sender for meget, så der ikke opstår flooding i routeren buffer(queue)

# Chapter 4 (Network Layer)

## Routing protocols (link-state, distance vector, inter-AS, intra-AS)

Link-state routing protocol, er en protokol der laver et kort over hvordan de forskellige routere er forbundet. Hver router beregner derefter den korteste vej til den næste destination osv. med algoritmer f.eks OSPF (Open Shortest Path First), som vælger den korteste vej hver gang. Dette er ikke nødvendigvis den hurtigste tid, men den korteste strækning. Modsat en motorvej, som er den længste rute i antal km men en hurtige ankomst.

Distance vector, finder længden af ruten ud fra hvor mange hops (routere) din packet skal igennem. Nogle distance vectors tager også “network latency” og andre faktor i brug, når den skal finde den korteste rute. På den måde fungerer dette ligesom en motorvej, måske den længste rute men den hurtigste vej.

Inter-AS routing er routing packets mellem flere AS’ (Autonomous system, en ISP eller en stor organisation). Her har du derfor en gateway router man skal igennem, når man skal forlade sit AS og kommunikere med et andet AS. Det modsatte af en Intra-AS. Intra-AS er nemlig kun på et AS.

## Forwarding, routing

Forwarding og routing er de 2 vigtigste funktioner i netværkslaget.

Routing er den proces der bestemmer hvor pakkerne skal sendes hen. Forwarding er måden man sender pakkerne afsted. Så man kan sige, at routing er kaptajnen og forwarding er motoren.

## Delay and loss within a router

**(**[**https://www.d.umn.edu/~gshute/net/delays-losses.xhtml**](https://www.d.umn.edu/~gshute/net/delays-losses.xhtml)**)**  
Delays på et netværk opstår oftest ved et queue overflow. Det sker når data (pakker) ankommer til queue’en hurtigere end det kan blive transmitted. Det resulterer i at queue’en vokser, indtil der ikke er mere plads, hvorefter noden bliver nødt til at bortkaste de resterende pakker, hvilket bliver kaldt for packet loss.

## BGP (Border Gateway Protocol (version4)

**(**[**https://blog.cdemi.io/beginners-guide-to-understanding-bgp/**](https://blog.cdemi.io/beginners-guide-to-understanding-bgp/)**))**  
BGP er den protokol som binder ruterne (ikke routerne) sammen på internettet. BGP er designet til at hjælpe routere med at advertise til sammenhængende netværk. Siden internettet er et netværk af netværk, fungerer BGP således, at det hjælper routere med at udbrede disse netværk til alle BGP routere i hele verden.   
I en BGP routing, er der altid allokeret et ASN (AS Number), dette er specifikt for hvert netværk, og er vigtige for to netværk der har en etableret forbindelse. De kan derefter veksle BGP informationer, og omtales BGP peers.

## IPv4 versus IPv6 **(**[**https://www.youtube.com/watch?v=aor29pGhlFE**](https://www.youtube.com/watch?v=aor29pGhlFE)**)**

IPv4 tillader kun omtrent 4 milliarder unikke IP adresser, og har 32 bit adresse, hvorimod IPv6 bruger 128 bit adresser, som tillader langt flere IP adresser, og i princippet nok til at hver person ville kunne bruge milliarder af IP adresser for sig selv. Grunden til at man ikke bare skifter IPv4 ud med IPv6, er i høj grad kompatibilitet. Man kan for eksempel ikke tilgå en hjemmeside, som kører på IPv6, med en IPv4 adresse. De fleste computere, og smartphones er og har været IPv6 kompatible i lang tid, men skiftet er ikke profitabelt for netværks udbyderne, og da disse ikke ville gaine noget af det, er det ikke i deres horisont at opgradere til IPv6. Til dels fordi det ville gøre NAT “ubrugeligt”, forstået som at det kan give fordele, men dets hovedsaglige funktion ville være irrelevant, samtidig med at det fjerner kontrollen af trafikken på ens netværk.

## Longest prefix matching

Når pakker bliver forwarded, søger routeren efter den destination der matcher med prefixet. Alle computere på et netværk har det samme IP Prefix. F.eks i en /24 er prefixet de første 24 bits i IP adressen. Men hvis man både bruger /18 og /24 kan IP prefixet godt overlappe. Derfor bruger man reglen kaldet “longest prefix matching”. Det betyder bare, at man kigger på prefixet og sammenligner, den der matcher med det længste prefix på din forwarding table, det bliver det næste hop.

## Tunneling

Tunneling er en protokol der tillader en sikker sendelse af data fra et netværk til et andet. Tunneling er også kendt som port forwarding. Dataen bliver brudt ned i packets, og de rejser igennem til destinationen. De er krypteret og bliver encapsulated (se chapter 1). Det ligner public data når det bliver sendt, derfor kan det blive sendt via internettet, men det er i virkeligheden private data. Når det kommer til modtageren bliver der decapsulated og dekrypteret.

## DHCP

Dynamic Host Configuration Protocol

Når en DHCP-Server skal tildele en IP-adresse til en klient, bliver der gjort brug af D.O.R.A processen.

**DORA** er et 4 trins process over tildeling af IP-adresser for klienter som tildeles af DHCP Serveren.

1. **Discover**: Klienten broadcaster en "DHCP discover" message, for at finde en DHCP-server
   1. Dette er en UDP pakke gennem port 67.
   2. Klienten kender ikke addressen på DHCP-serveren i netværket.
   3. Der laves et IP datagram med en DHCP discover message, som sendes med broadcast destinationen 255.255.255.255
      1. og en source IP-addresse med 0.0.0.0 (klienten har ingen)
2. **Offer:** Serveren svarer med en DHCP Offer msg - Sendt af DHCP serveren med henblik på at tilbyde en “leaset” IP-adresse til den klienten.
   1. Serveren modtager en DHCP discover message - den svarer med en DHCP offer besked. Med source IP-adressen 255.255.255.255.
3. **Request:** ‘DHCP Request message’ sendt af DHCP klienten for at spørge serveren for den specifikke IPv4-adresse - som er tilbudt i Offer meddelelsen.
   1. Klienten vil vælge et af serverens “offers” og svarer tilbage med en DHCP Request message”.
4. **Acknowledgment**: Sendt af DHCP serveren for at tildele IPv4-adressen og definere Mask, Default router, og DNS server IP-adresser.
   1. Serveren svarer til “DHCP request” meddelelsen med en “DHCP ACK Message” og confirmer de efterspurgte parametre.

Nu hvor klienten har modtaget DHCP ACK messagen, så er interaktionen komplet, og klienten kan nu bruge den DHCP-allokerede IP-addresse, indtil at leasen udløber.  
Efter som at DHCP serveren allerede nu har en IP-adresse - vil DHCP serveren som hovedregel automatisk renew leasen på IP-adressen til klienten.

## NAT

Den største motivation for at have NAT, er at vi ellers ville løbe tør for IP adresser.

Network Address Translation

På et lokalt netværk kan der være mange private IP-adresser.

Man har en enkelt global IP-adresse.

NAT'en sørger for at alle pakker indenfor som sendes ud af det private netværk får samme globale IP-adresse.

Sikkerhed. Enheder inden for det private netværk, er ikke synlige for verdenen udenfor.

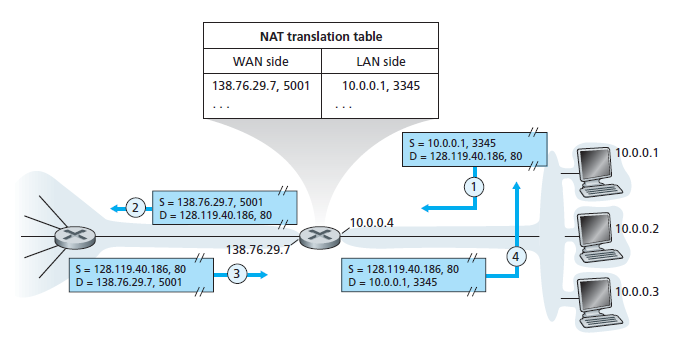
Alle datagrammer som forlader det logiske netværk har samme ene source NAT IP Adresse.

Med forskellige source port numbers.

En NAT-enabled router “ser ikke ud” som en router udenfor et internt netværk.  
For the outside world ligner al trafik, som kommer ud af den NAT-enabled Router, at det kommer fra en enkelt device, med en enkelt IP-adresse.

Routeren får en IP addresse fra ISP’ens DHCP server, og klienterne internt i routerens netværk får IP-adresser tildelt af den NAT-enabled routers lille DHCP-server.

## NAT Translation Table.

  
***Her bruges der et NAT-translation table***

# Chapter 5 (Link Layer)

## Link layer services

Link layer services er opdelt i 4 underkategorier, som hvert link layer protocol stiller til rådighed:  
  
*1. Framing:*  
Næsten alle link layer protokoller indkapsler hvert network layer datagram, med et link layer frame før det sendes videre til et link. Sådan et frame består af et data field, hvori network layer datagrammet befinder sig i, og et antal af header fields. Strukturen af denne frame er specificeret af det anvendte link layer protocol.   
  
*2. Link Acces:*  
En medium access control (MAC), er en protokol der bestemmer reglerne, for hvilket frame der bliver tilsendt et link. for Point-to-point links, hvor en der findes en afsender, og en modtager på linket, er MAC protokollen meget simpel, eller endda unødvendig, og derfor ikke-eksisterende. Skulle der dog være flere afsendere/modtagere på et link, er det denne protokol der bestemmer og koordinerer frame transmission.   
  
*3. Reliable delivery:*  
I praksis betyder reliable delivery bare, at et link layer protokol garanterer at sende alle network layer datagrammer over linket, uden errors. Ligesom TCP der giver reliable data transfer. Lignende et transport layer reliable delivery service, giver et link layer reliable delivery service pålidelighed ved at sende ACK’s og retransmissions.   
  
*4. Error detection and correction:*  
Der kan forekomme fejl på hardwaren i en modtagende node, som definerer et frame som et 0 i stedet for et 1, og omvendt. Mange link layer protokoller, har en indbygget mekanisme, der detecter fejl som disse, så der undgås støj, og konflikter. Støj opstår, når et datagram skal fikses, og fordi der (sjovt nok) ikke er grund til at sende et korrupt datagram. Den modtagende node, er den der laver error checket. Det er dog mere kompliceret end i transport layeret, hvor der regnes ud fra checksum. Yderligere, er det mere integreret i hardwaren end softwaren. Det specificerer ikke kun OM der er en fejl, men også hvilket bit, i hvilket frame der har en fejl.

## Broadcast link versus PPP

Point to point protocol (PPP) er en computer network protocol. Den bruges til at sende et datagram mellem 2 computere der er forbundet direkte, deraf navnet point to point. Et telefonopkald er f.eks PPP. det er 2 enheder der forbinder sig direkte.

TCP og IP understøtter ikke PPP, og det skal derfor være forbundet uden Ethernet. Når de 2 computere er forbundet, sender de hver en request for konfiguration. De bliver derefter forbundet, og så tager PPP ansvaret for link control, data control og protocol encapsulation.

## Error detection and correction

Error detection og correction bruges hvis et netværk ikke er reliable, eller der er meget støj på linjen. Error detection, som i ordet, finder fejlene og correction genskaber den originale data.

Data bliver sendt, og inden det kommet til destinationen bliver der spurgt “er alle bits i orden?” hvis nej prøver de at genskabe det, og hvis det ikke er muligt bliver det sendt forfra.

Der er 2 måder der kan være error. Dene ene er single-bit-error, hvor en a bitsene er blevet ændret, eller bust-error hvor flere bits er blevet ændret.

## CRC

CRC (Cyclic Redundacy Check) er en protokol der er designet til at finde fejl i data. Blokke af data får en “check value” udfra hvad dataen indeholder. Når den kommer frem sammenligner man denne “check value” med dataen. Hvis de 2 spiller overens er der ikke fejl på dataen, og omvendt. Det er den mest udbredte “error detection” protokol, fordi den er nem at implementere samt nem at nedbryde matematisk. Det er god til at finde små fejl forårsaget af f.eks støj.

## MAC (Multiple access protocol)

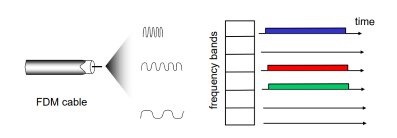
En MAC adresse. Fysisk udstyr der kommunikerer med hinanden f.eks. over internettet har en MAC-adresse. En MAC adresse skal være unik. MAC-adressen består af 12 Hexadecimale tegn i par af to. Det giver en adresse-længde på 48 bit.

### Channel Partitioning

**FDMA:**

Klienternes forbindelse er opdelt i frekvens-bånd.

I FDMA (Frequency-Division Multiple Access) kan flere klienter sende pakker hele tiden - man deler dog hastigheden (forbindelsen).



**TDMA:**

(Time-Divison Multiple Access)

Klienterne har i perioder fuld forbindelse (fuld hastighed).

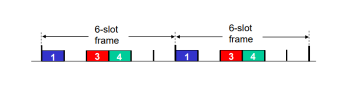
Dette har man når slottet begynder.

Man kan kun sende pakker i starten af slottet.

Kanalen deles op - 1 klient får et bestemt slot - hvor de kan sende pakker.

Hvis klienten ikke har noget at sende, så går slottet potentielt til spilde

Dette er en ulempe ved TDMA.



## Random Access Protocoller

Hver gang man har en pakke så sender man den afsted med fuld channel rate (hastighed).

Der er ikke tidligere koordination mellem noderne.

Hvis 2 eller flere noder kolliderer sker der en kollision.

Random Access MAC protocoller specificerer:

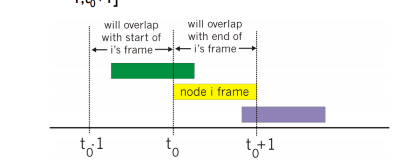
Hvordan der detectes kollisioner

Hvordan man genoptager efter at der er sket kollision (f.eks. ved delayed retransmission)

Eksempler på Random Acces MAC Protokoller er

### (Pure) **ALOHA**

Hver gang man har en pakke, så sender man - ligegyldigt om det er i starten af slottet eller ej

1. Frames bliver transmitteret på den delte channel, af alle noderne som gerne vil sende.
2. Et Acknowledgement afventes af modtager noderne, de afventer et ACK på at deres frame er blevet modtaget succesfuldt.
3. Hvis der sker en kollision, så vil noderne få dette at vide efter time-out perioden - når der ikke er modtaget et ACK.
4. Efter timeout-perioden, vil noderne afvente i et random-amount of time (back-off-time), før at de transmitterer igen.   
   Dette Random-delay-time er ikke altid det samme, for at forhindre kollision.
5. Hvis der sker en kollision ved endnu en retransmission, så kører step 4 igen. Dvs. Back-off-time - retransmission.
6. Efter et prædetermineret maximum antal retransmissioner, så er afsenderen nødt til at give op og prøve igen senere.

### **Slotted ALOHA**

I Slotted ALOHA antager man at alle frames har samme størrelse.

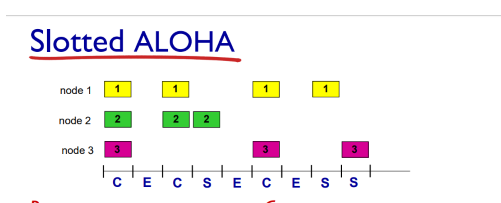
Man sender kun i starten af slottet, så alle noderne synkroniseres.

Hvis 2 pakker sendes i samme slot, så trækker man tilbage og prøver igen.

Når en ny pakkes skal afsendes:

Hvis der ikke er kollision, så sendes næste frame i næste slot.

Hvis der er kollision, så retransmitteres der i næste slot.



Fordele ved Slotted ALOHA:

Man har hele båndbredden når man succesfuldt afsender.

Man aftaler at man altid starter i starten af slottet - dvs alle er in sync.

Ulempe:

Stor sandsynlighed for kollision

Den tid det tager at før at man opdager kollision, kan være større en den tid det ville tage at transmittere en ny pakke.

Tomme slots.

### CSMA/CD

Carrier Sense Multiple Access: **Collision Detection**

Det er nemt at opdage kollision i et trådet netværk, fordi man lytter når man sender.

Det er svært at opdage på et trådløst netværk.

Nogle enheder sender med stor styrke.

Man kan sende med så høj styrke, at det overdøver andre signaler.

Derfor bruges der som regel CSMA/CA på et trådløst netværk.

Jo tættere man er på hinanden jo mere effektiv.

Man opdager kollisioner meget hurtigere.

Effektiviteten er meget bedre end ALOHA.

### CSMA/CA

Carrier Sense Multiple Access**: Colission Avoidance**

Lyt før du snakker.

Hvis der er andre der afsender, så afventer man til de er færdige med at sende.

Man kan stadigvæk have kollision.

Hvis alle lytter og ingen sender. Så vil alle sende på samme tid.

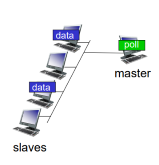
Der laves Back-off når der opdages kollision.

## “Taking Turns”

Ved Taking Turn MAC Protokollen finder man the “best of both worlds”

Der er hhv. **Polling** og **Token Passing**:

Ved **polling:**



Har man en Master Node - som er den som “slave nodes” lytter til.

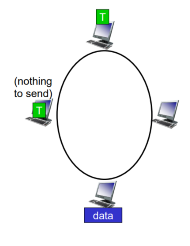
Ulempe:

Hvis masteren går istå, så går netværket i stå - single point of failure.

Det er typisk “dumme”-slave devices der er slaves.

Dette er typisk i industrien. De maskiner som producerer har en computer i

Disse lytter efter Master noden.

**Token passing:**

Der er en token (som i stafetløb).

Der sendes når man holder “token”, denne token passerer fra node til node.  
Ulempe:

Hvis en af klienterne beholder denne token, så kan de andre ikke sende data.

## ARP (Address resolution protocol)

Address resolution protocol er en [internetprotokol](https://da.wikipedia.org/wiki/Internet), som oversætter [IP-adresser](https://da.wikipedia.org/wiki/IP-adresse) til hardwareadresser ([MAC adresser](https://da.wikipedia.org/wiki/MAC-adresse)) på netkort. Protokollen kan kun bruges mellem [datanet-værter](https://da.wikipedia.org/wiki/Datanet-v%C3%A6rt) på samme fysiske net.

Når der første gang skal sendes en [IP](https://da.wikipedia.org/wiki/Internetprotokol)-pakke fra en maskine til en anden, kender afsender maskinen ikke modtagernes hardwareadresse, og sender derfor en broadcast meddelelse med den ønskede IP-adresse til alle andre maskiner på det aktuelle net. Den maskine, der har den ønskede IP-adresse sender sin hardwareadresse tilbage, og gemmer samtidig afsenderens hardwareadresse til senere brug. På meget store netværk, kan det ske, at en maskine ikke har kapacitet til at gemme alle adresser på netværket, så de, der bruges mindst kan blive smidt ud.

(Protokollen [RARP](https://da.wikipedia.org/wiki/Reverse_address_resolution_protocol) har den omvendte funktion.)

## Ethernet

Preample er til at vække adapteren, så fjernes den.

CRC er for at tjekke for bitfejl og så fjernes den.

## Vlan (trunking, trunk protocol, port security, ………..)

VLAN (Virtual Local Area Network) bruges til at opdele et lokalt netværk. Det kan bruges til at lave et lukket netværk inde i et eksisterende netværk. Hvis vi tager en virksomhed som både har en salgsafdeling og en indkøbsafdeling kan det være en fordel for virksomheden at dele deres netværk op i to VLAN’s de to afdelinger kører på det samme fysiske netværk, men ved hjælp af VLAN bliver der oprettet et virtual netværk inde i det fysiske netværk. Dette sikre at der bliver reserveret en mængde båndbredde til begge afdelinger. I en virksomhed med mange VLAN’s sikre det også sikkerheden på netværket. Computere forbundet til et VLAN ville ikke kunne opsnappe pakker fra et andet VLAN, selvom de to VLAN’s fysiske netværk er det samme.

Selvom virksomheden har rigtig mange switche fordi det er en virksomhed med mange enheder forbundet til netværket, kan f.eks. to switche godt være tilkoblet det samme VLAN. Det kan også være en økonomisk fordel at oprette VLAN. Hvis en mindre virksom gerne vil dele deres netværk op i to kan dette gøres ved hjælp af 2 switche, men med VLAN kan du som sagt dele netværket op i to uden at investerer i en ekstra switch.

Dette kan være en kæmpe økonomisk fordel da en switch med f.eks. 48 porte koster flere tusinde kroner.

## Port Security:

***MAC Filtering*:**

En metode for sikkerhed, som vi vil implementere i Switch, er Port Security på MAC-adresser.

Switchen husker (learn) de første MAC-adresser som forbindes til disse porte (og dermed automatisk fylde sit register – deraf learn). Man kan også på forhånd definere hvilke MAC-adresser, som skal have adgang til specifikke porte.

## Trunk Port:

Som nævnt tidligere I beskrivelsen af VLAN kan to switche godt være tilkoblet det samme VLAN. Dette sker ved hjælp af en Trunk port. Lad os sige at vi har 2 switche med 2 VLANs i hver switch, dem kalder vi VLAN 1 og VLAN 2 til disse 2 switche er der tilkoblet 2 computere til hver switch hvor den ene computer er en del af VLAN 1 og den anden en del af VLAN 2.   
De 2 switche er forbundet via vores trunk port. Hvis computeren som i den ene switch er en del af VLAN 1 og skal sende en pakke til computeren i den anden switch som også er en del af VLAN1 sker dette igennem trunk porten. Pakken bliver tagget inden den ryger igennem trunk porten.

Trunk ports markerer frames med et unikt identifying tag: 802.1Q

Dette tag fortæller den anden switch at denne pakke tilhører VLAN1 og sender derfor pakken ud til de porte i switchen som er en del af VLAN1. Ved hjælp af dette tag bliver pakken ikke sendt ud til computeren som er en del af VLAN2 hvilket gør at det ikke påvirker netværket for VLAN2. hvis det var en pakke der skulle sendes fra computeren i VLAN2 ville pakken ved hjælp af trunk porten få et tag der fortæller at pakken tilhører VLAN 2 og dermed kun blive sendt ud til portene som er en del af VLAN2

# Chap. 6: (Wireless and Mobile Network)

## Wireless and mobility

## Base station

En Base Station er typisk forbunde til et Wired Netværk.

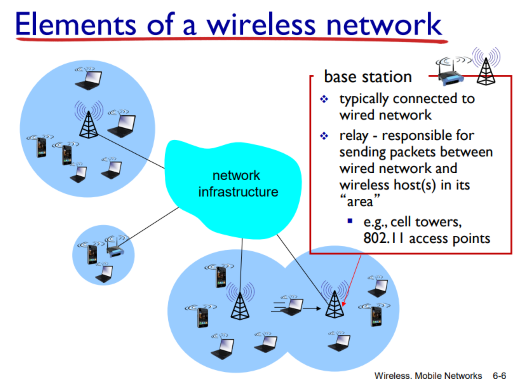
Relay: Står for at sende pakker mellem wired netværk og wireless host(s) i dens “område”

Dvs. Mobil-master, 802.11 Access Points

En Base Station er en nøgle del af et wireless netværks infrastruktur.

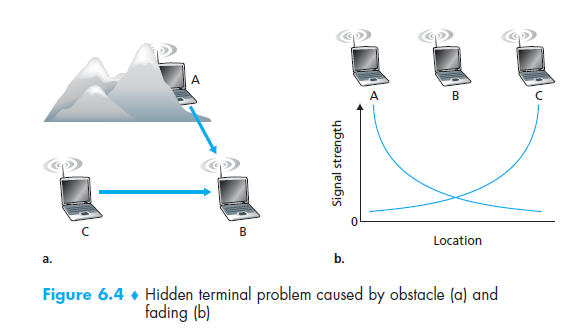
Hosts som er associeret med en base station bliver ofte refereret til at operere i **infrastructure mode**, da alle traditionelle netværks services bliver udført af netværket som base stationen er forbundet til. Dvs, at DNS-name translation, IP-adressering, routing osv bliver udført af netværket.

Når en mobil hosts bevæger sig ud af rangen for en base station og ind i range af en anden, så vil det skifte base station og denne proces kaldes **handoff**.



## Hidden node problem

I bogen beskrevet som **hidden terminal problem**.   
Det er hvis der er fysiske obstructions i vejen, i omgivelserne (en bygning f.eks. - beton etc.).



Denne bygning vil medføre, at host a og host c ikke kan høre hinanden - på trods af at de begge kan overføre data til host b. Dette er fordi at host A og host C’s transmissioner, **fader**, når det går i gennem dette medium.   
Så host A og Host C kan ikke transmittere til hinanden, men de kan begge transmittere til host B - og Host B kan transmittere til host A og C.

# 

# 

# 

# 

# 

## Ad hoc mode and infrastructure mode

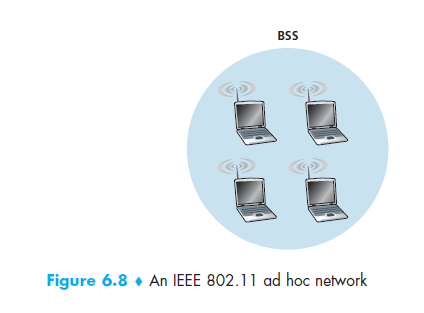
### Ad Hoc Mode:

Figure 6.8 shows that IEEE 802.11 stations can also group themselves together  
to form an ad hoc network—a network with no central control and with no connections  
to the “outside world.” Here, the network is formed “on the fly,” by  
mobile devices that have found themselves in proximity to each other, that have  
a need to communicate, and that find no preexisting network infrastructure in  
their location. An ad hoc network might be formed when people with laptops get  
together (for example, in a conference room, a train, or a car) and want to  
exchange data in the absence of a centralized AP. There has been tremendous  
interest in ad hoc networking, as communicating portable devices continue to  
proliferate.

IEEE 802.11 stationer kan også gruppere sig sammen for at forme et ad hoc netværk - det vil sige et netværk uden en central control og uden forbindelse til “the outside world”.   
Her er netværket lavet “on the fly” af mobile enheder (ikke nødvendigvis mobiltelefoner husk det - men pc’er typisk laptops), som er i nærheden af hinanden, og har brug for at kommunikere - og der er ikke nogen netværks-infrastruktur de kan gøre brug af.

Her kan en computer med et WIFI-kompatibelt netværkskort, oprette et ad hoc netværk, som de andre clienter kan tilkoble - og dermed kan de kommunikere sammen.  
*(Der er ikke nogen central “base station” officielt i følge bogen - men burde den enhed som opretter ad hoc netværket, egentlig ikke opføre sig som en “base station”?)*

IEEE 802.15.1 netværk er Ad Hoc Netværk (Dette er f.eks. Bluetooth-forbindelser).



### Infrastructure mode:

IEEE 802.11 wireless LAN, også kendt som WIFI, er den mest hyppigt brugte standard for  
wireless LAN teknologier i dag. IEE 802.11b/g Wireless LANs opererer i frekvenserne 2.4 –  
2.485 Ghz – som også bliver brugt af 2.4 Ghz telefoner, samt mikrobølgeovne, dette kan  
medføre congestion i et trådløst netværk, med packetloss til følge.  
Fundamentalt set er 802.11 infrastruktur kendetegnet af et Basic Service Set (BSS), som er  
opbygget af en eller flere trådløse stationer samt en central **base station**​, som er kendt som  
et **Access Point ​(AP​)**.

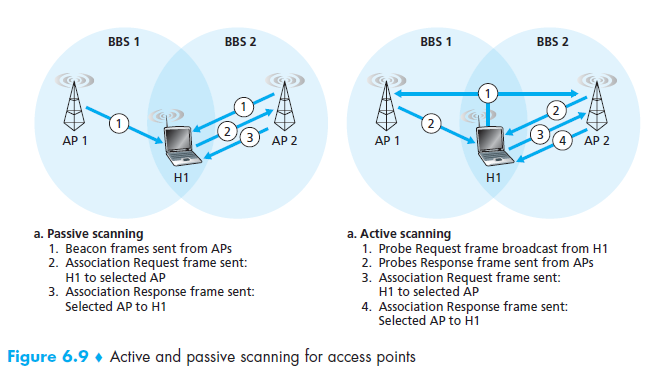
Trådløse LANs som gør brug af Access Points, er kendt som **Infrastructure Wireless  
LAN**​s, hvoraf selve infrastrukturen er dermed opbygget af et Access Point, som gennem en  
trådet (wired) forbindelse er tilsluttet med en router.

I 802.11 skal hver wireless station/client associere sig med et Access Point før at det kan  
sende og modtage data fra **netværkslaget**.

Når et access point bliver installeret, skal administratoren tildele access pointet et SSID  
(**Service Set Identifier​**), som kan betegnes som det trådløse netværks (navn/ID). Det er  
SSID’et som bliver opfanget når man søger efter trådløse netværk, f.eks. på sin Laptop eller  
smartphone.  
Derudover skal administratoren også tildele Access Pointet et **channel number**​. Channel  
numbers er defineret ved, at være indelt i 2.4 til 2.485 GHz frekvensen – i denne 85 Mhz  
bredde er der inddelt 11 delvist-overlappende **channels​**/kanaler. Kun channel 1, 6 og 11  
overlapper ikke med hinanden i denne frekvens. Det vil derfor give mening for en  
administrator, hvis der skal installeres f.eks. 3 Access Points indenfor et givent område, at  
tildele dem channel hhv. 1, 6 og 11 – da dette vil skabe mindst congestion i den trådløse  
frekvens.

For at kunne oprette forbindelse til disse trådløse netværk (gennem association med deres  
SSID), skal de trådløse clienter have mulighed for at kunne ”se” og lokalisere disse trådløse  
netværk. 802.111 standarden kræver at et Access Point periodisk udsender **beacon  
frames​**, som hver inkluderer Access Pointets SSID samt MAC-adresse. Vores trådløse  
enhed/client ved at Access points udsender beacon frames, så clienterne scanner de 11  
channels og søger efter disse beacon frames. Når disse beacon frames bliver modtaget af  
vores client, har vi hermed mulighed for at kunne skildre og vælge det SSID, Acces points  
som vi vil associere med. Denne form for channel scanning (lytte efter beacon frames) er  
kendt som **passive scanning​**.

En wireless host/client kan også gøre brug af en metode kaldet **active scanning**​, hvor  
denne broadcaster et **probe frame**, som bliver modtaget af alle Access Points i dens  
rækkevidde. Access Points svarer herefter tilbage på dette **probe request frame** med et  
**probe respond frame**. Herefter kan den trådløse host/client igen vælge hvilket Access Point  
som det vil associere med.  
Efter at have valgt hvilket Access Point den trådløse host vil associere med, sender hosten  
en **association request frame**, og Access Pointen svarer herefter tilbage med et association **response frame**.

Når den trådløse host har associeret sig med et Access Point, vil hosten blive tilsluttet til det  
subnet som Access Pointen tilhører. Derfor vil hosten typisk sende en **DHCP discovery  
message** ind i subnettet via Access Pointet, med henblik på at få tildelt en IP-adresse i  
subnettet.  


# 

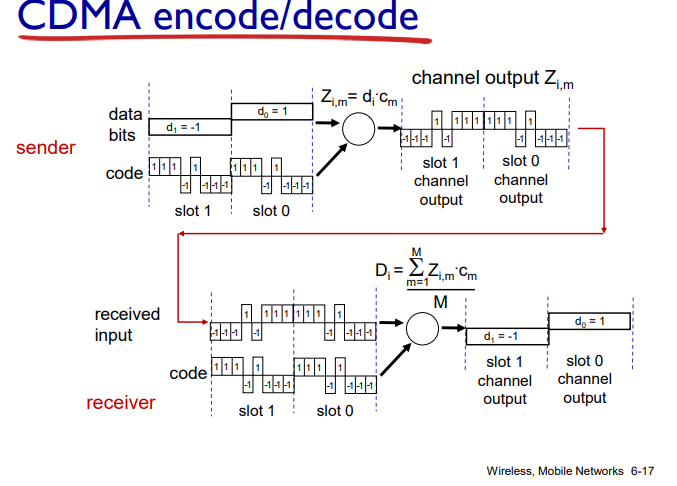
## CDMA (Code division multiple access)

Når hosts kommunikerer over et shared medium (i dette tilfælde f.eks. et Cell Tower), så er der behov for en protokol, så signalerne som bliver sendt af flere afsendere ikke interfererer ved modtageren.

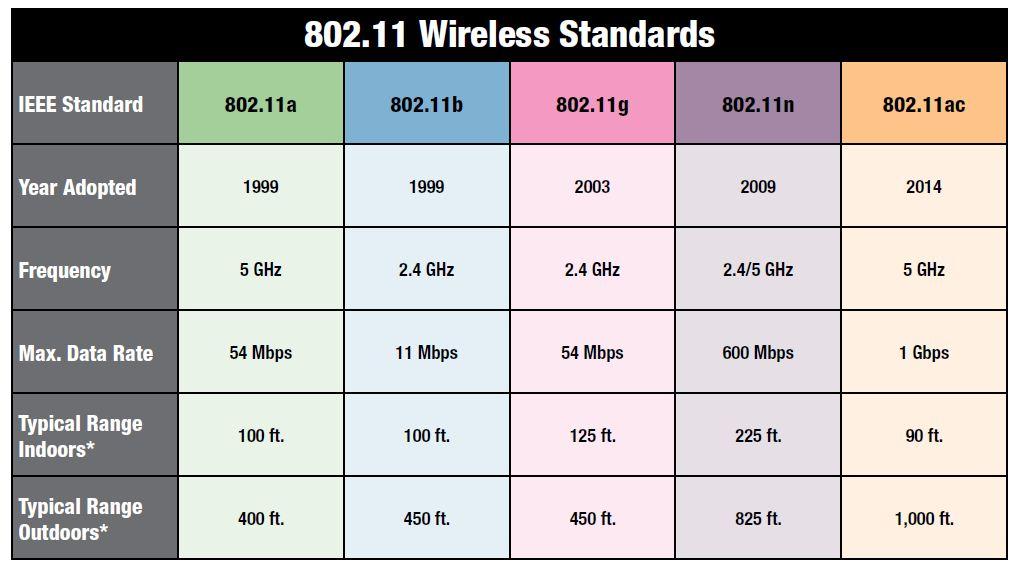
CDMA tilhører familien af channel partitioning protokoller. Det bliver brugt i Wireless LANs og cellular teknologier.  
I CDMA-protokollen, bliver hvert bit kodet ved at gange bittet med et signal som ændrer sig kaldet **chipping rate**.

Et encoded signal = (original data) \* (chipping sequence/rate)

Decoding = dette er det indre produkt af det encoded signal og chipping sequence/rate.



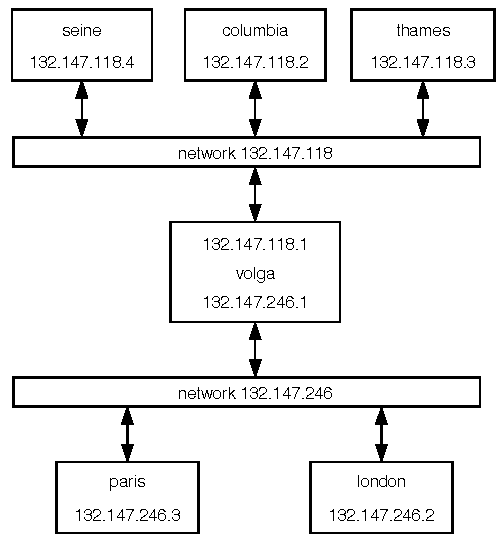
## 802.11 standards



802.11 standards beskriver en række LAN/WLAN standarder.

## 

## Direct and indirect routing

  
Direct routing; Hvis Columbia skal sende IP packets til thames eller seine, kan dette refereres som direct routing, da der er én afsender og én modtager.  
Indirect routing; Hvis Columbia skal sende IP packets til for eksempel paris eller london, skal disse igennem flere netværk. Disse bliver kaldt for network gateways. Som det kan ses på ovenstående illustration, er volga en network gateway, som formidler pakkerne. Det vil sige at volga faktisk er modtageren, og derefter bliver afsenderen af de “interceptede” IP packets.

## Handoffs

I princippet er handoff den process der sker, når en mobil for eksempel skifter telefonmast ift. forbindelse. Der bliver refereret til et handoff, når man opretter forbindelse på ny med en ny mast, og så dropper den gamle mast. Der ved, et hand off. Dette kan ske på baggrund af 2 ting;  
1: Forbindelsen bliver forværret fra den gamle mast, pga. rækkevidde f.eks, og det vil gavne mobilens ejermand at tilgå ny og stærkere forbindelse fra en tættere stående mast. (samtalen har risiko for at dø)  
2: Der opstår en congestion på ens telefon, hvis man modtager for mange opkald på en gang. Man afhjælper modtageren ved at omdirigere størstedelen af opkald til en anden mast. (((Taget på baggrund af at alt kører over et standard MSC (Mobile Switching Center))))

## Home and foreign agents

En home agent, er en router der tilhører en mobils hjemme netværk, som vedligeholder informationer omkring enhedens lokation, som bliver identificeret i dens care-of-address. Home agent’en bruger tunnelling mekanismer, for at forwarde internettrafik, så at mobilens IP adresse ikke skal skifte hver gang den skal forbinde til en anden lokation. Home agent’en fungerer i samarbejde med en foreign agent, som gør det samme, bare på besøgendes netværk. Altså IKKE ens private hjemmenetværk men netværk, som man er “gæst” hos.   
  
Det skal dog tilføjes, at foreign agents (og home agents) bliver overflødige, hvis der bliver integreret IPv6. Home og foreign agents blev integreret i IPv4 (MIPv4) for at hjælpe med at begrænse overforbrug af IP adresser.

# Chapter 8

## Hashfunktion

En kryptografisk hashfunktion bruges i forbindelse med datatransmission. Dens formål er at kontrollere om data er blevet ændret, enten ved en fejl eller en 3. parts indblandelse. den simpleste hashfunktion er checksum. Den er dog ikke den mest “sikre”, da blot 2 bytes skal ændres i hver sin retning for at opnå den samme checksum. SHA-1 er en mere sikker hash funktion.

## Symetric/shared key

En symmetrisk nøgle er en fælles nøgle der er delt mellem modtager og afsender. Det er en krypteringsmetode der båder låser og låser op for beskeden. Det kræver dog, at begge parter har denne nøgle på deres enhed.

En stor svaghed ved dette koncept er, at hvis man skal sende nøglen via internettet, er den jo ikke krypteret, da begge parter ikke han en nøgle at kryptere med endnu. Derfor bliver man nødt til at mødes face2face og udveksle nøglen inden man sikkert kan kommunikere via internettet med den symmetriske nøgle.

## Public og private key

Public og private key er et koncept der bygger på, at alle har en privat nøgle. den private nøgle kan både kryptere og dekryptere en besked, men modparten skal altid dekryptere. Det vil sige, at hvis jeg krypterer noget med min private key, er det kun min public key der kan dekryptere selv samme besked. Det bygger på, at alle public keys er tilgængelige for alle, men denne private key har kun personen selv.

Eks.

Hvis jeg krypterer noget med min private key, skal modtageren åbne med min public key.

Hvis jeg krypterer noget med afsenderens public key, skal modtageren åbne med hans private key.

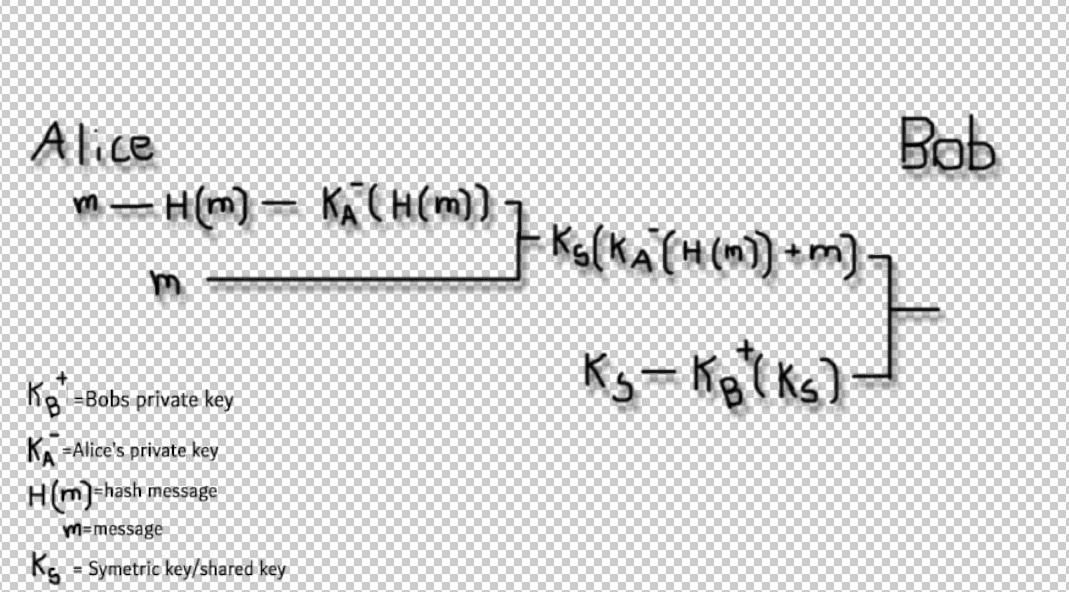
dette er også en måde at gøre sig sikker på, at beskeden kommer fra den rigtige person og ikke en 3rd party. For der findes kun en der har en private key der tilhøre personen, og det er personen selv.

## Nonce

Nonce er et tilfældigt genereret nummer, som bliver sendt samtidig med besked. Dette nummer bliver kun brugt en gang pr server, og det er for at forhindre playback attacks. Så hvis der er en der har opsnappet samtalen mellem 2 personer dagen forinde, og prøver at sende samme informationer igen, vil Noncen afslører vedkommende, da Noncen allerede er blevet brugt.

## Secure email

De ovenstående krypteringsmetoder spiller alle sammen med i Secure email. Under ses et eksempel på forløbet fra afsender til modtager.

****

Alice skal sende en besked til bob. Hun starter øverst til venstre med at sende en besked, denne bruger hun en hashfunktion på (f.eks SHA-1). Hun hasher beskeden for at være sikker på, at når den når frem, at informationen ikke er blevet ændret undervejs. Derefter kryptere hun Hash-beskeden med sin private key. Dette gør hun både for at kryptere informationen, men også for at “stemple” beskeden med hendes personlige “stempel” (grundet kun hun har sin private key, kan ingen andre kryptere det med den - andre end hende). Udover hashen sender hun også den oprindelige besked for sig selv.

Efter beskeden og hash-beskeden krypterer hun det yderligere med en Symetric key. Så nu er hash beskeden krypteret med både Alice’s private key samt en symetric key, og beskeden er krypteret med samme symetric key. For at Bob i sidste ende også skal kunne bruge samme symetric key, bliver den også sendt for sig. Denne symetric key bliver dog krypteret med Bobs public key. Dette er fordi, at hvis den er krypteret med en public key, kan kun den respektive private key åbne indholdet. Så det er kun Bob der kan åbne denne besked med symetric keyen som indhold.

Når bob så modtager beskeden, starter han man at udpakke denne symetric key. Når han har symetric keyen, bruger han den til at åbne (oppe i højre hjørne af billedet), den stadigvæk krypteret Hash-besked samt den originale besked. Selvom det hele er krypteret, vil bob stadig være sikker på, at der ikke er blevet ændret i indholdet. Så han bruger nu Alice’s public key til at åbne Hash-beskeden, sammenligner indholdet af den samt den originale besked. Hvis ikke der er forskel på de 2 indhold, ved Bob at beskeden er rigtig.

## VPN

VPN Står for virtual private network, og det er metode der sikrer din internetforbindelse. Enheden kobler sig til en VPN server som bliver til linket mellem enheden og internettet. Forbindelsen til VPN serveren er krypteret. Selve VPN servers fungerer også som en anonym proxy, dette skjuler både klientens IP, samt giver klienten en kunstig geografisk placering.

En VPN’s kryptering er umulig at knække, det vil tage millioner af år at dekryptere en almindelig 256-bits VPN kryptering.

Hvis en VPN server ikke logger brugerdata, som de færreste gør, vil et forsøg på at spore klienten være nærmest umulig.

## Securing TCP connections: SSL

SSL står for Secure Socket Layer, og er en sikkerheds teknologi der gør det muligt at etablere en sikker forbindelse (HTTPS forbindelse) mellem en webserver og en browser. Det betyder kryptering af data der går gennem hjemmesiden.

For at SSL overhovedet kan begynde at kryptere data mellem en browser og en webserver, skal der laves et SSL handshake.

Der skal bruges 3 forskellige nøgler til at lave en SSL forbindelse; en public, en private og en shared key. Kryptering og dekryptering med private og public keys kræver meget processorkraft, og bliver derfor kun brugt under handshaket. public og private key kryptering bliver brugt til at dele en shared key som nu står for krypteringen.

Handshaket der således ud:

1. Browseren skaber en forbindelse til web serveren, og beder serveren om at identificere sig selv.
2. Serveren sender en kopi af den SSL certifikat, samt serverens public key.
3. Browseren sammenligner certifikatet sammen med en liste over godkendte HTTPS servere. Hvis den passer sender browseren en shared key, krypteret med serverens public key, til serven.
4. Serveren dekryptere pakken, med sin private key, som indeholder denne shared key. Serveren sender nu en ack krypteret med den samme shared key.
5. Nu kan serveren og browseren kryptere al data fremover.

## Wired Equivalent Privacy (WEP)

WEP er designet til at kryptere og authenticate mellem en host og et trådløst AP, ved brug af en symmetric key. WEP specificere ikke valg af nøgle, så man går ud fra, at hosten og AP’et har aftalt valg af nøgle via en out-of-band (noget der sker udenfor) metode.

Authentication ser således ud:

1. En trådløs host requester authentication hos et AP.
2. AP’et svarer med en 128-byte nonce.
3. Den trådløse host krypterer nonces med en symmetric key.
4. AP’et dekrypterer noncen.’

Hvis den dekrypterede nonce passer med den oprindelige nonce, bliver hosten godkendt af AP’et.

## I EEE 802.11i’