Nytte funksjoner:

Strsplit(string,hva du vil splitte på)

Sort(matrise)

Sortrows(matrise)

Begge ascending som basic

Reshape(matrix/vektor,[rows,kolomns]

strrep( str , old , new )

[ newData] = sscanf(extraData, 'max=%f, min=%f, %fmm')

[ junk, rest ] = strtok(extraData,'=')

Flip(x) returnerer vectoren flippet.

Ceil/floor runder opp eller ned

Isa(objekt,’double’/’single’/uint8’) sjekker om et gitt objekt er en spesifikk klasse.

char(cell) gir string

/\* For Matlab \*/

## **Notater fra teorifoiler**

## HARDWARE

Input/Output-enheter: Mus, styreflate, mikrofon, kamera, skjerm, printer

Hovedenhet: Beregningsenheten – CPU, Hukommelse – RAM

Eksterne enheter: Harddisker, dvd-disker ++

Hoved(krets-)kort/Motherboard: Trykt kretskort inne i kabinettet. Inneholder mesteparten av kretsene til PC-systemet. Kan ha mange forskjellige utseender.

**primærminne:**

(Primær- / Hoved-minne): Stedet hvor programmer og data lagres mens programmet kjører. Slettes når maskinen slås av.

RAM: Random Access Memory–flyktig (ikke permanent). En PC inneholder flere millioner/milliarder bytes med RAM–Megabytes (MB) / Gigabytes (GB).

Hva betyr «Random Access» –Alle elementer kan hentes direkte –Forskjellig fra sekvensiell aksess (f.eks. LP/kassetter/spolebånd).

**Sekundærminne:**

Høy-kapasitet, persistent perifer lagringsenhet. Lagrer programmer og data som ikke er i øyeblikkelig bruk i datamaskinen. Kalles permanent eller persistent minne –ikke-volatilt (ikke-flyktig, ikke-temporært).

Solid State (Hard) Drives: Laget av «Flash Memory». Ingen bevegelige deler - dvs: Rask–Stor “permanent” «RAM». Slites akkurat som magnetiske disker. Harddisker er også direkte aksess (ikke sekvensielle)

**Integretre kretser: (IC = Integrated Circuits)**

Miniatyrisering: Klokkehastighetene kan være så høye fordi CPUene er så små (elektriske signaler kan forflytte seg omtrent 33cm på et nanosekund). ca. 1 milliard transistorer på en overflate på ca 150 mm2.

Fotolitografi: Trykkeprosess (som trykte bøker, side for side). Flere lag oppå hverandre. Istedenfor å koble sammen kretser for hånd «Fotograferer» man de kretsene man ønsker, og etse bort tomrommene i mellom «ledningene». Uansett hvor komplisert kablingen er, så er kostnaden og mengden arbeid alltid den samme.

Mindre viktig: (a) Et lag med fotoresist (blå/grå) eksponeres for UV-stråling gjennom en mønstermaske (lysblå/grå), og de eksponerte områdene herdes(blå); (b) den ueksponerte (myke) fotoresisten er vasket bort, og da kan varme gasser etse bort det ubeskyttete metallet (brunt) i det eksponerte laget, og bare enkelte beskyttete «komponenter» blir igjen; (c) resten av fotoresist-laget (som har beskyttet det gjenværende laget) vaskes/pusses så bort, og andre lag (rød) kan lages ved å gjenta mønstertegning og etse-prosessen med andre lag senere i prosessen

**Halv-leder teknologi:**

Integrering: Aktive komponenter, og kontaktene som kobler dem sammen, er alle sammen laget av lignende materialer i en enkelt prosess. Dette sparer plass og fører til at hele systemet er bare en monolittisk del, som er mer pålitelig enn flere smådeler

Silisium er en halv-leder: noen ganger leder det strøm, andre ganger ikke –Evnen til å kontrollere når en halv-leder leder eller ikke er hoved-redskapen i all datamaskin-konstruksjon

En krets er laget for å beregne «x og y» for alle mulige logisk verdi av x og y («true/false»). Hvis x er «sann» skal kretsen lede elektrisitet og signalet passerer til den andre enden av ledningen; hvis x er «usann» (false) skal signalet ikke passere. Det samme gjøres for y-kretsen •Hvis begge kretsene leder strøm, x og y er sanne—så har vi beregnet logisk «AND.

**Transistorer:**

En kobling mellom to ledninger som kan styres til å la elektriske ladninger flyte, eller ikke, mellom to kabler. En bryter uten bevegelige deler. Vi har nettopp beskrevet en MOS-transistor: Metall-Oksyd Halvleder(«Semiconductor»)

## Instruksjons-Utførings-Maskin

5 grunnleggende deler/subsystemer: Minne, Kontrollenhet, Aritmetisk/logisk enhet (ALU), Input-enhet og Output-enhet.

Forskjellige datatyper tar opp forskjellig plass : char (tar vanligvis opp 1 byte (ASCII) eller 2-4 byte (UNICODE)), float (8 eller 10 byte) -etc...

**Hente / Utføre:**

Kretsløpet: En fem-skritts syklus .

Instruksjons-henting (IF) – Instuction fetch

Instruksjons-dekoding (ID) – instruction decode

Data-henting (DF) / Operand-henting (OF) – data fetch

Instruksjons-eksekvering (EX) – Instruction execution

Resultat-retur (RR) / Lagre (ST = «Store») – Result return

**Aritmetisk/Logisk Enhet (ALU):**

Utfører regneoperasjonene. En krets i ALU kan legge sammen to tall. Det finnes også kretser for å multiplisere, sammenligne, osv.

Gjør vanligvis jobben i Instruksjonsutførings-steget (EX) i syklusen, men instruksjoner som bare flytter på data bruker ikke ALU.

Data/Operand Hente-steget (DF) i syklusen henter de verdiene som ALUen trenger å jobbe med (operandene til operatoren). Når ALUen fullfører operasjonen sørger Resultat Returnerings/Lagrings-steget for å flytte svaret fra ALUen til den minnelokasjonen som er spesifisert i instruksjonen.

**Programtelleren:**

PC-ens PC (Counter). Adressen til den neste instruksjonen lagres i kontroll-delen til maskinen. Den kalles programteller (PC) Maskinen plusser på fire i PC, så når F/E-syklusen kommer tilbake til Instruksjons Hente-steget neste gang, så «peker» PC på den neste instruksjonen.

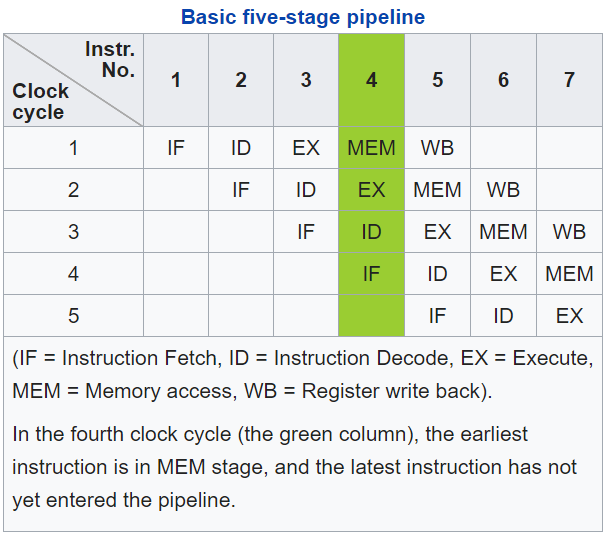
Datamaskiner kan bare utføre omkring 100 forskjellige instruksjoner. Omtrent 20 forskjellige typer operasjoner (Det trengs forskjellige instruksjoner for å legge sammen enkelt-bytes, «ord», desimaltall, osv.). Alt datamaskinene gjør må reduseres til en kombinasjon av disse primitive, hardkodete instruksjonene.

**Maskin-klokka:**

Bestemmer hastigheten på F/E-syklusen. Måles i gigahertz (GHz), antall milliarder sykluser per sekund

Pipelining: Overlater avslutning av instruksjonen til andre kretser. En slags parallell-utføring. Forskjellig fra flerkjerne parallell-prosessering.

Fem instruksjoner kan behandles samtidig



**Assemblerspråk:**

En alternativ form for maskinspråk som bruker bokstaver og normale tall, slik at mennesker kan forstå det. Maskinen leser assemblerkode, og etter hvert som den finner ord slår den opp i en tabell for å konvertere ordene til binærkoder, tall konverteres til binær form, og så settes de binære kodene sammen («assembles») til en instruksjon. De forskjellige prosessorene har forskjellig assemblerspråk.

**Høynivå programmerings språk:**

De fleste moderne program er skrevet med høynivå notasjon, som deretter kompileres (oversettes) til assemblerspråk, som igjen assembles til binærkode.

Har spesielle uttrykksformer som hjelper programmerere å gi kompliserte instruksjoner.

**Operativsystem (OS):**

Grunnleggende operasjoner som er nødvendig for å kunne bruke maskinen effektiv, men som ikke er bygd inn i maskinvaren.

OS-et tar seg av «booting» (oppstart), minnehåndtering, enhetshåndtering, Internettforbindelse, filhåndtering.

## Digital representasjon

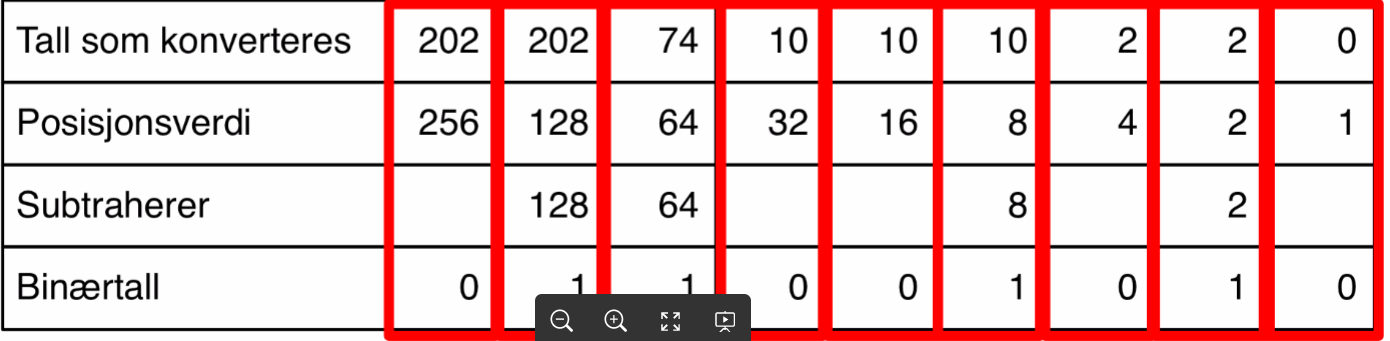
Metadata: informasjon om informasjon, typ string, double eller lignende

**Konvertering binært til desimalt:**

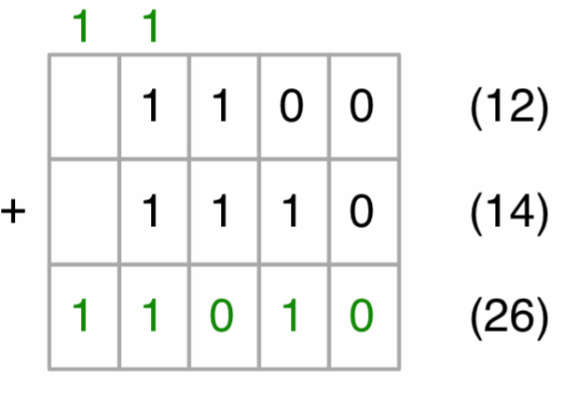
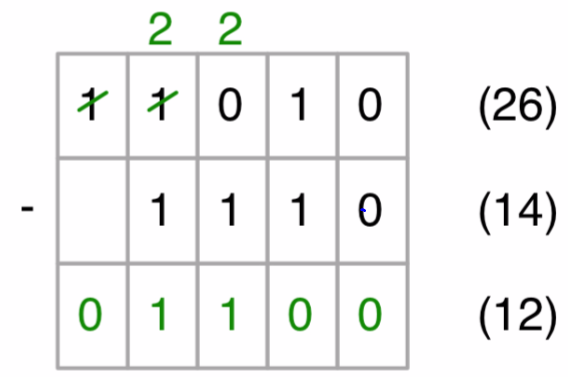
Enkelt og rett fram

101 = 1\*2^2+ 0\*2^2+ 1\*2^0= 4 + 0 + 1 = 5

**Desimalt til binært**



**Addisjon og subrasjon:**

**Toerkomplement:**

Gjør at vi kan unytte hele byten til tall, og ikke mister en bit til fortegn. Regnemetode:

Inverser det positive tallet, og legg til 1.

1. 001 -> 110 +1 -> 111(-1)
2. 010 -> 101 +1 -> 110(-2)

Mindre viktig: IEEE standard for floating-point numbers

Single precision: 32 bits (4 byte). 24 bits til fortegn + mantisse. 8 bits til eksponenten. Absoluttverdier omtrent: 1,2 x 10-38–3,4 x 1038. Omtrent 7 signifikante desimale siffer

Double precision: 64 bits (8 byte). 53 bits til fortegn + mantissen. 11 bits til eksponenten. Absoluttverdier: 2,2 x 10-308–1,8 x 10308. Nøyaktighet: Omtrent 16 desimale siffer

**Digitalisering av lyd:**

Et objekt lager lyd ved å vibrere et medium. Krafteneller intensiteten av bølgeneavgjør volumet. Frekvensene(antallbølgerper sek) avgjør tonehøyden(pitch)

Analog til digital: ”Sample” eller ta målingerpå bestemte intervall. Antall samples per sekund kalles samplingsfrekvens (rate). Jo raskere frekvens, jo mer nøyaktig opptak av lyden

Nyquist-regelen: samplingsfrekvensen må være minst dobbelt så rask som den raskestefrekvensen. Ettersom menneskelige ører kan høre lyder opp til ca. 20 000Hz, vil samplingsfrekvens på 40 000Hz oppfylle Nyquists regel for digital lydopptak.

Merk: Vi kan kun få omtrentlige målingerav lyd. Hvis et ekstra bit vil bli brukt, vil lydsamplet bli dobbelt så nøyaktig. Flerebits oppløsninggir mer nøyaktig digitalisering.

**Utregning av størrelse på digital lyd:**

Tid i sekunder\*antall sampler (vanligvis 44100)\*nøyaktighet\*antall kanaler(2 hvis stereo).

**Filstørrelse**

Bredde\*Høyde\*Fargedybde•600\*400\*3 byte = 240.000\*3 = 720.000 byte

**RGB -Rød, Grønn og Blå**

Standard for utlyst farge (fra skjermer og piksler)

0-255 enheter av hver farge, brukes heksadesimalt system fra 00 til FF.

**CMYK -Cyan, magenta, yellow, black/key**

Brukes i skrivere.

**Tapsløs/ tapskoding:**

Tapsløs: Halvering, tredeling

Tapskoding: Større effekt, avhengig av tap

Komprimering betyr å endre representasjon slik at færre bit trengs til å lagre eller overføre informasjon.

Run-length-koding er tapsløs(lossless) komprimering: Den originale representasjon av 0ere og 1ere kan bli rekonstruert perfekt fra den komprimerte versjonen.

Motsatte er lossy-(taps)-komprimering: Originale representasjon kan ikke rekonstrueres eksakt fra den komprimerte versjonen.

MP-3 er sannsynligvis den mest kjente type komprimering. MP3 er lossy ettersom de høye notene kan ikke gjenskapes.

JPG (or JPEG) er lossy komprimering av bilder. Utnytter karakteristikker av menneskelige oppfatningsevne for å lage forenklinger iht. lys og farge som ikke er så synlig. JPEG er i stand til 10:1 komprimering uten synlig tap av klarhet i bilde ved å gjøre områdene der man komprimerer små.

## Nettverk del 1

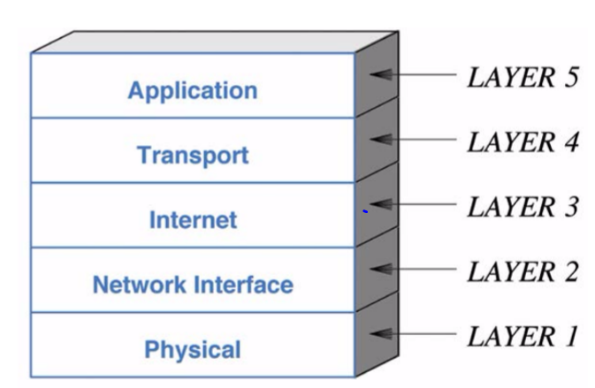
**Packet switching:** Vi sender x antall pakker ut i internett. Disse sendes ut til internett, hvor internett sender de til riktig sted stort sett, og stokker de tilbake i riktig rekkefølge.

**Circut switching:** er mer i stil gamle telefonsystemer. Noen må fysisk plutte riktige plugger sammen sånn at informasjonen kommer til riktig sted.

**Protokoll:**

Beskriver prosedyrer for–feilhåndtering –uventede hendelser (f.eks. Ingen respons fra mottaker).

Protokoller i en familie er organisert i lag. Den lag-baserte modellen (layering model) gjør det lettere å håndtere kompleksiteten i nettverkskommunikasjon.

**TCP/IP stabelen (stack)**:

Applikasjon: WWW, Snapchat, YouTube•Transport

TCP: ntnu.no, google.no•Internett

Lag 4 (TCP) sørger for pålitelig overføring av data.

IP: 129.241.103.4

Lag 3 (IP) spesifiserer: Formatet på pakkene. Mekanismer for å sende pakker fra én datamaskin via rutere, til enn annen.

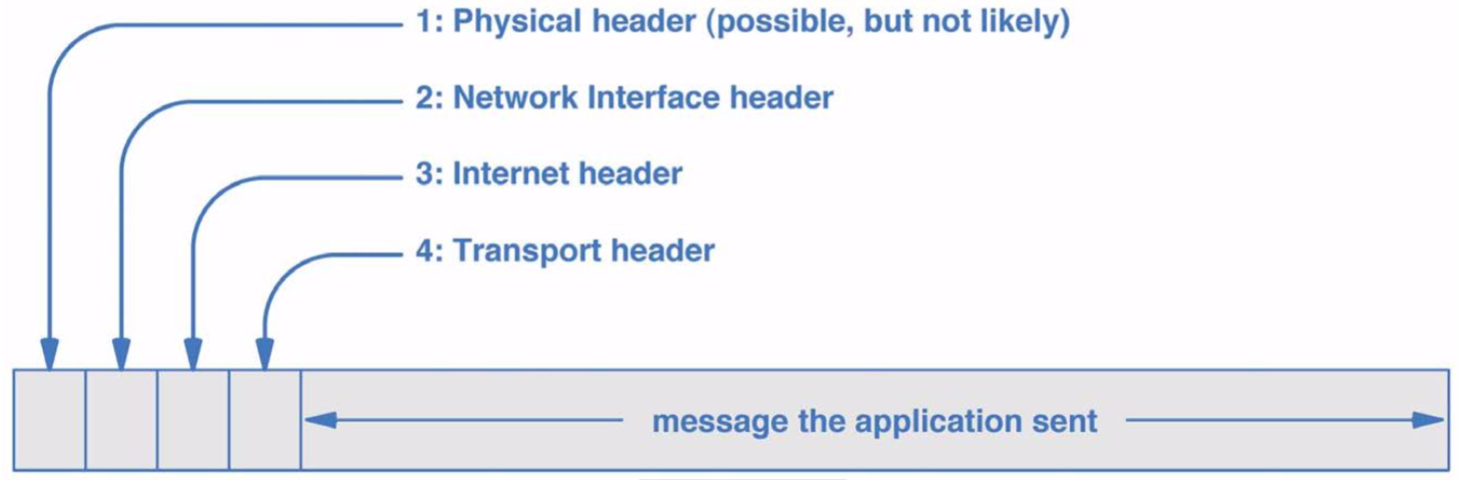
Nettverksgrensesnitt: MACAB:CD:EF:12:34:56

Fysisk lag: Spenning i kobber, radiofrekvens i luft, lys i rør, ...

I hvert av disse lagene er det retningslinjer og standarder for hvordan ting skal håndteres sånn at alle datamaskiner i verden skal på en god måte kunne kommunisere over internett.

**Hoder (headers) og lag:**

Ettersom datapakker blir sendt gjennom lagene i protokoll-stacken legges det til hoder (ekstra informasjon). Hodene leses og fjernes av tilsvarende lag på mottakersiden (og underveis!).



OSI (Open Systems Interconnection)

ISO-standard for lagdeling av kommunikasjonsprotokoller. Har to lag i tillegg til de vi finner i TCP/IP-stacken. Presentasjon LAYER 7–Sesjon LAYER 6

## Pålitelighet og kanalkoding

Feilkilder i overføring av data:

Interference (forstyrrelse): Elektromagnetisk stråling kan forstyrre radiosignaler.

Distortion (forvrenging): Signaler kan forvrenges f.eks. Av metallobjekter.

Attenuation (svekking): Signaler svekkes over når de sendes over et medium.

**Feiltyper:**

Single Bit Error: En enkelt bit i en blokk med bits endres, mens øvrig bits forblir uendret.

Burst Error: Flere bits i en blokk endres.

Erasure (ambiguity): Signalet som kommer frem hos mottaker er tvetydig (hverken logisk 1 eller logisk 0) pga. f.eks. forvrenging.

**Håndtering av feilkilder:**

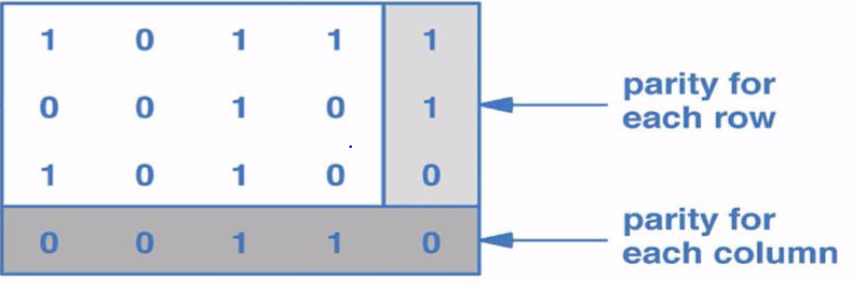
Automatic Repeat reQuest (ARQ): Forutsetter samarbeid med avsender. Avsender og mottaker utveksler informasjon for å sikre at data blir overført korrekt. Hvis senderen ikke mottar bekreftelse fra mottaker på at meldingen er riktig og har ankommet, sender den på nytt.

Forward Error Correction (FEC): Legger til tilleggsinformasjon som lar mottaker verifisere at korrekt data har blitt mottatt og rette feil hvis mulig.

Eksempler på FEC:

**Block Error Codes**: Dataene som sende deles inn i blokker, med ekstra informasjon (redundancy) tilknyttet hver blokk. Typisk Single Parity Checking og Row and Column (RAC) parity

Parity check: Sender legger til en ekstra (paritets-) bit (0 eller 1) til hver byte. Partalls-paritet: Paritetsbit settes til 0 dersom blokken har et likt antall 1-bits og 1 dersom blokken har oddetall. (skal alltid være partall enere). SPC kan kun avdekke feil i tilfeller hvor et odd antall bits har endret seg. SPC avdekker ikke hvilken bit som er feil (kan ikke korrigere).

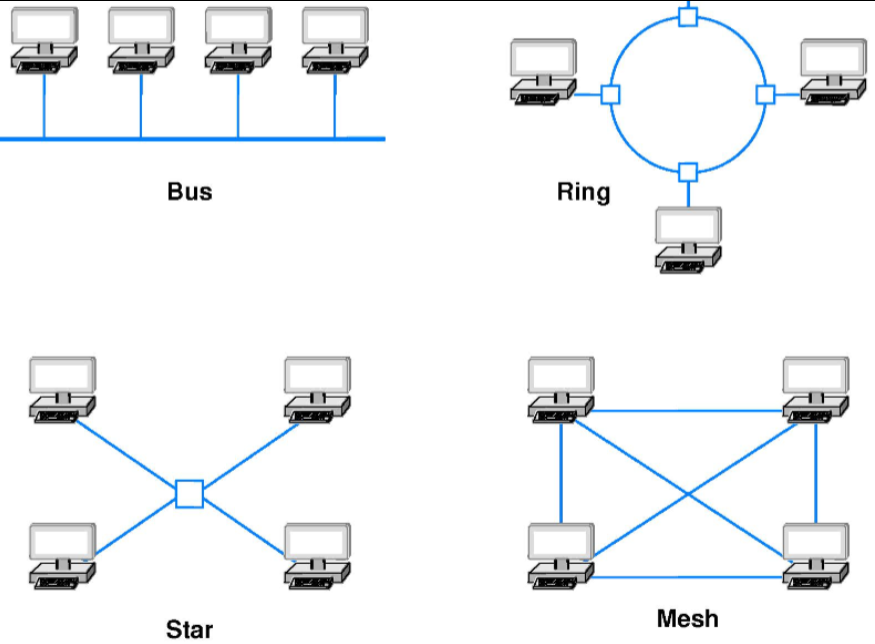
RAC: Sjekker parity både vertikalt og horisontalt som en tabell 

**Convolutional Error Codes**

Blokker med data blir påfestet en kort sjekk verdi, basert på polynomverdi og restledd. Bra til å detektere vanlige feil fra svekking/attentuation.

## Nettverk del 2

Nettverk karakteriseres gjerne etter området de dekker: LAN (LocalArea Network) dekker typisk et rom eller en bygning. MAN (MetropolitianArea Network) dekker typisk en stor by. WAN (WideArea Network) dekker typisk flere byer.



**Adressering i LAN:**

Hver datamaskin i et pakkesvitsjet LAN har en unikadresse: Media Access Control (MAC) address. Kalles også fysisk adresse, hardware adress, m.m.

Hver pakke inneholder MAC-adressen til avsender-node og mottaker-node. Lag 2 i TCP/IP-stacken(Network interfacelayer) legger til MAC-adresseinformasjon på pakken.

Unicast, broadcastofmulticaster tre typer adressering som brukes i forbindelse med pakkelevering.

**Universal service:**

Tillate kommunikasjon mellom datamaskiner uavhengig av type nettverk de sitter på. Internett er en sammenkopling av ulike fysiske nettverk. Universal service er et grunnleggende premiss for at Internett fungerer.

**Rutere:**

Maskinvare med prosessor, minne og I/O-grensesnitt som brukes for å nettverk. Kan kople sammen ulike typer nettverk (f.eks. to LANs, et LAN og et WAN, eller to WANs) og ulike nettverksteknologier. Bruker protokoller for å støtte kommunikasjon mellom ulike typer nettverk.

**Virituelt nettverk:**

internett-programvare skaper en illusjon av et enkeltstående, sømløst kommunikasjonssystem bestående av mange datamaskiner.Internetter et eksempel på et (verdensomspennende) virtuelt nettverk. Internett trenger i mindre grad å forholde seg til dem. Illusjonen muliggjøres av TCP/IP.

For at abstraksjonen skal fungere må alle datamaskiner på internett ha: Felles adresseringsformat og Unik adresse (MAC-adressen er generelt kun tilgjengelige innenfor samme lokalnett).

IP-adressering sørger for at en kan sende data til alle maskiner på internett. Sender merker pakken med IP-adressen til mottaker (i tillegg til egen IP-adresse). IP-protokoll-programvare sørger for at pakken når frem.

**IPv4:**

En IPv4-adresse består av 32 bits og har et prefiks og et suffiks som tilsammen gir en unik adresse.

IP-prefikset identifiserer det unike fysiske nettverket (nettverksnummeret) en datamaskin er tilknyttet.

IP-suffikset identifiserer en spesifikk datamaskinen på nettverket.

Fordeling av nettverksnummer må håndteres globalt (av ICANN). Fordeling av suffikser kan håndteres lokalt.

Dotted Decimal Notationer en uttrykksform som gjør det lettere å lese, skrive og snakke om IP-adresser.

**IPv6:**

En IPv6-adresse består av 128 bits, mens en IPv4-adresse består av kun 32 2 IPv6 har mye mer brukbare adresser i forhold til IPv4. IPv6 gjør ruterens oppgave enklere i forhold til IPv4, IPv6 er bedre egnet til mobilnett enn IPv4. IPv6-adresser er representert i en heksadesimal, kolon-separert notasjon, mens IPv4-adresse bruke dot-titallssystemet.

## Nettverk del 3

**Håndtering av duplikater og pakker i feil rekkefølge (Sequencing):**

Avsender merker hver pakke med et sekvensummer. Mottaker lagrer sekvensnummer på siste mottatt pakke i tillegg til en liste med andre pakker som kom i feil rekkefølge.

Hvis forventet pakke -> lever til neste lag

Hvis uforventet pakke -> legg pakken i vente-listen

Mottaker forkaster nye pakker som er mottatt før, eller som ligger i vente-listen.

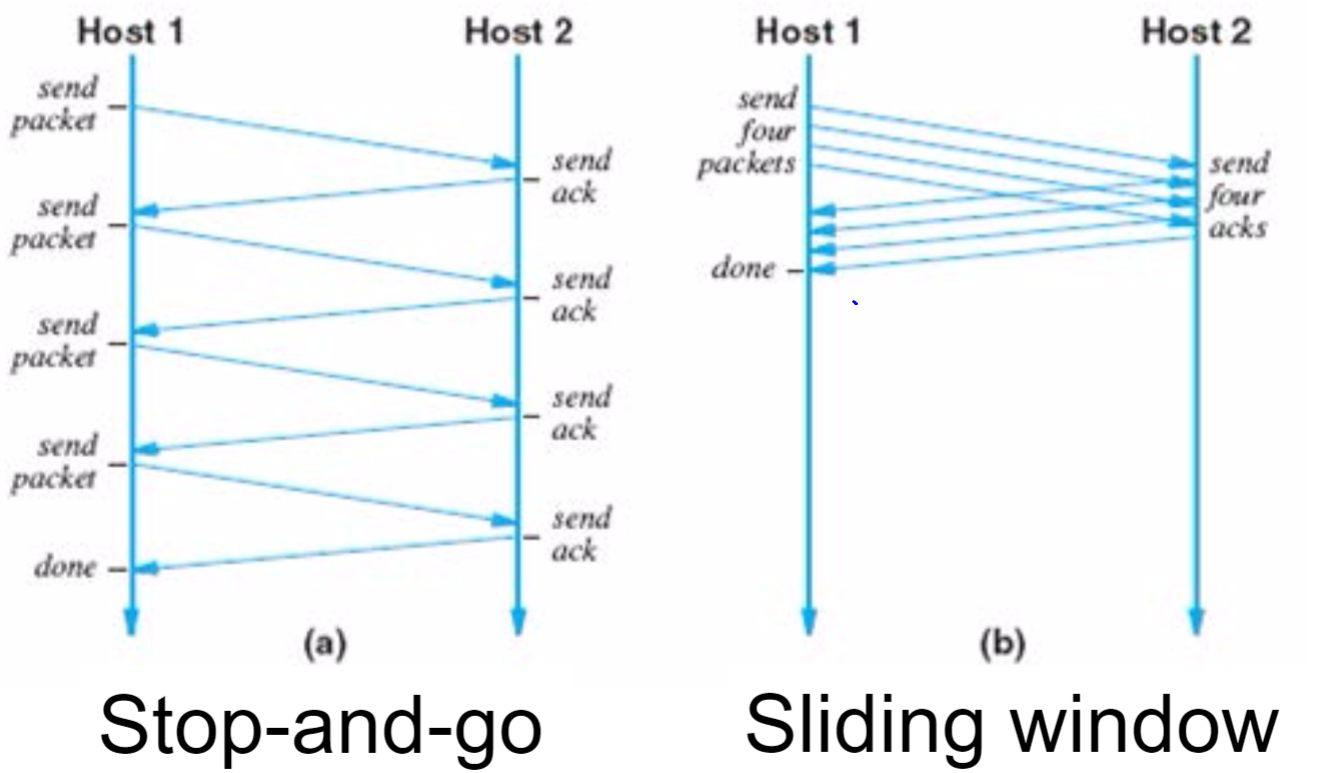
**Håndtering av pakketap (retransmisjon):**

Når en pakke ankommer intakt sender mottaker en bekreftelse (ACK). Hver gang avsender sender en pakke starter den en timer. Dersom bekreftelse fra mottaker ankommer før timeren utløper kanselleres timeren. Dersom timeren utløper før bekreftelse blir mottatt sendes en kopi av den (tapte?) pakken (retransmisjon) og en ny timer startes.

**Håndtering av forsinkelser (replay errors):**

Replay error: Forsinket pakke fra tidligere sesjon blir akseptert i senere sesjon. Riktig pakke blir avvist som duplikat. Kan oppstå i forbindelse med svært lange forsinkelser.

Håndteres gjennom å merke pakker i en sesjon med en unik ID (f.eks. sesjonenes starttidspunkt)–Protokoll-software avviser pakker med feil ID.



**Nettverksytelse**:

Latency (delay)–Tiden det tar å overføre data over et nettverk fra et endepunkt til et annet.

Throughput (capacity, a. k. a. bandwith)–Mengden data som kan overføres innenfor en gitt tidsenhet.

Jitter (variability)–Endringer i forsinkelser og lengden på forsinkelsene

**Qualityof Service (QoS):** Noen vil betale sin ISP mer for å få garantier om hastighet, båndbredde etc

## Nettverk del 4

**Sikkerhetsutfordringer:**

Phishing: Opptre som en kjent nettside (f.eks. nettbank) for å få tak i personlig informasjon som f.eks. aksesskoder, kontonummer, etc.

Misrepresentation: Oppgi feilaktige opplysninger om et produkt eller tjeneste eller levere produkter tjenester som er falsk eller av dårlig kvalitet.

Scams: Lure brukere av f.ekset nettsted eller en tjeneste til å investere penger eller gjøre noe ulovlig.

Denial of service: Bevist blokkering av tilgang til en nettsted eller tjeneste.

Loss of control: Uvedkommende tar kontroll over en brukers datamaskin.

Loss of data: Tap av åndsverk eller annen verdifull informasjon.

**Sikkerhetsangrep:**

Wiretapping (avlytting): Kopiere datapakker som traverserer nettet for å få tak i informasjon.

Replay: Sende pakker fanget opp fra tidligere sesjoner (f.eks. passord-pakker fra tidligere pålogginger).

Buffer overflow: Oversende mer data enn hva mottaker forventer. Overskrider et databuffers grenser og skriver til nabolokasjoner i minnet. Kan føre til minneaksessproblemer, feil resultater og krasj.

Spoofing: Bruke falsk IP-kildeadresse for å lure mottaker til å prosessere pakken. Bruke falske e-postavsendere, misbruke domenenavn. Avlede datatrafikk rettet mot en server.

DenialofService (DoS/DDoS): Overøse en vert med datapakker (fra én eller flere kilder) for å ta bruke opp alle ressurser. Skaper lange forsinkelser.

SYN flood: Form for DoS-angrep. Angriperen sender en serie med SYN-forespørsler (synkroniseringsforespørsler om å opprette forbindelse) mot et datasystem for å forhindre annen trafikk.

Passwordbreaking: Automatiserte systemer laget for å ”knekke” passord og dekrypteringsnøklerfor å få uautorisert aksess til en nettressurs.

Port Scanning: Forsøk på å kople til åpne protokollporter på en vert for å finne en svakhet (porter brukes for å identifisere prosesser, eller type nettverkstjeneste).

Packetinterception: Modifisering av pakker på vei fra avsender til mottaker.

**Sikkerhetsteknologier:**

Hashing: hashing-mekanismer kan lage meldings-autentiseringskoder som ikke kan knekkes eller forfalskes. I noen hashing-mekanismer benyttes en hemmelig nøkkel som bare avsender og mottaker har. Nøkkelen brukes til å generer en hash-kode (kort bitsekvens) som avsender merker meldingen med.–Mottaker bruker nøkkel for å undersøke om han/hun får generert samme hash-kode. En angriper kan ikke modifisere pakken uten å introdusere feil. En gyldig hash-kode betyr mao. at pakken er autentisk.

**Kryptering:** Å kryptere en melding vil si å endre på dataene i en melding, slik at kun riktig mottaker kan rekonstruere den opprinnelige meldingen.

Plaintext: Orginalmeldingen før kryptering.

Cyphertext: Den krypterete meldingen.

Encryptionkey: Bitsekvens som brukes til kryptering.

Decryptionkey: Bitsekvens som brukes til dekryptering

Krypteringsteknikker kategoriseres etter hvordan de bruker krypteringsnøkler: Private key systems (systemer med privat nøkkel) eller Public key systems (systemer med offentlig nøkkel)

Kryptering med privat nøkkel: Partene deler en hemmelig nøkkel som brukes både for kryptering og dekryptering.–Partene kan både sende og motta krypterte meldinger

Kryptering med offentlig nøkkel: Hver part får en (hemmelig) privat og en offentlig (distribuerbar) nøkkel. En melding kryptert med en offentlig nøkkel kan kun dekrypteres med den korresponderende private nøkkelen.

Digitale signaturer (autentisering): En melding kan krypteres to ganger for å garanterer både autentisitet (riktig avsender) og konfidensialitet (riktig mottaker). Først dekrypteres(signeres) meldingen ved at avsender bruker sin private nøkkel. Deretter krypterer avsender meldingen en gang til ved å bruke mottakers offentlige nøkkel.

**Brannmurer (firewalls):** Overvåker og kontrollerer trafikk inn og ut av et internt nettverk for å beskytte mot problemer som kan komme utenfra (f.eks. fra internett). Typisk innebygd i switcher og rutere. Sentraliserer kontroll over nettverkstrafikken som går inn og ut. Implementerer sikkerhetsretningslinjene til en organisasjon og stopper trafikk (pakker) som ikke er i tråd med disse.

**Inntrengings-deteksjonssystem (IDS):** Mekanisme som monitorer pakker og gir administrator beskjed når sikkerhetsbrudd avdekkes (en brannmur kan kun stoppe enkeltpakker som utgjør en potensiell trussel).

**Innholdsskanning og dyp pakkeinspeksjon:** En brannmur sjekker kun pakkehodet –ikke pakkeinnholdet. Innholdsanalyse kan sjekke pakkeinnhold for skadelig programvare, som f.eks. virus.

**Virtuelle private nettverk (VPN):**

Bruker kryptering for å tilby sikker aksess (via internett) til et lokalt nettverk for klienter som befinner seg utenfor (f.eks. tilgang NTNUs nettverk hjemmefra). Opprinnelig laget for å kople sammen lokale nettverk til en organisasjon som er fysisk distribuert i et sikkert virtuelt nettverk.

Stand-alonedevice: Bruker får egen ruter VPN ruter som automatisk etablerer sikker kommunikasjonskanal med VPN server.

VPN software: VPN programvare kjører på brukerens datamaskin, Krypterer alle pakker som så sendes til VPN serveren koblingen er knyttet mot. Dekrypterer alle innkommende pakker.

**Pakkekryptering:**

Det finnes hovedsakelig tre valgmuligheter mht. hvordan pakker som skal sendes over Internett kan krypteres: Payloadencryption, IP-in-IP tunneling og IP-in-TCP tunneling.

Payloadencryption: Krypterer kun meldingsinnholdet i pakken (ikke pakkehodet). Kilde-og destinasjonsadresse, pakkestørrelse m.m. kan fanges av uvedkommende.

IP-in-IP tunnelling: Krypterer både meldingsinnholdet og hodet og plasserer resultatet i en ny pakke, som så kan sendes over Internett.

IP-in-TCP tunneling: To parter etablerer en TCP forbindelse. Forbindelsen brukes til å sende en strøm av krypterte datapakker. Små hoder legges til for å markere skillet mellom datapakkene i strømmen (spesifiserer lengden per datagrammet). VPN programvare på mottakersiden leser hodet og antall spesifiserte bytes for å finne datapakken. Når en komplett kryptert melding er mottatt dekrypteresdenne. Pakkene må leveres i riktig rekkefølge (kan skape forsinkelser)

## Internett of things (IoT)

IoT brukes for å beskrive sammenkoplete integrerte systemer hvor datamaskiner kommuniserer med andre datamaskiner.

Valg av nettverksteknologi i M2M styres gjerne av krav til trafikkvolum, mobilitet, og energiforbruk.

Lavt energiforbruk er en viktig faktor i mange IoT-applikasjoner.

## Algoritmer

Definisjon på algoritme: En algoritme er et ordnet sett av entydige, utførbare skritt som definerer en terminerende prosess.

Polyas problemløsningsfaser i programering

1. Forstå problemet
2. Få en ide om hvordan en algoritme kan løse problemet
3. Formuler algoritmen og representer den som et program
4. Evaluer programmet mhp nøyaktighet og potensial for å løse andre problemer

Algoritmekompleksitet

Θ (log2 n): Binærsøk i sorterte data, f.eks. vektor, (cell)array

Θ (n): Sekvensielt søk i f.eks. vektor, (cell)array

Θ (n log n): Sortering med mergesort, quicksort»Nb: worst case hvis spesielt uheldig kan være n2 for quicksort

Θ (n2): Sortering med insertion sort, bubblesort

Θ (n3): Naiv multiplikasjon av to n x n matriser

Θ (2n): Prøve alle mulige kombinasjoner av bits for et passord på binær form

Θ (n!): ”Brute force” løsning av det såkalte ”travelling salesman” problemet

**Finne kompleksitet for egen kode:**

Rent sekvensielt program uten noen repeterte handlinger: Θ(1)

Enkel løkke som går gjennom alle elementer i f.eks. vektor: Θ(n). Der n er lengden til vektoren –NB: om løkka f.eks. bare ser på halvparten av elementene, også Θ(n) –Men om bare et konstant antall, f.eks. første 10 elementer: Θ(1)

Dobbel løkke, ytre løkke går ~n ganger og indre ~n ganger: Θ(n2)

NB:mye ser ut som enkle operasjoner, men skjuler løkker:

F.eks. max(V), sum(V): Θ(n)

Sortering, sort(V): Θ(n log n)

Matrisemultiplikasjon: Θ(n2.373)

**Flere måter å spare tid på**

Implementere ”samme” algoritme på en bedre måte: Bl.a. flytte ut av løkker det som kan gjøres utenfor–For eksempel redusere tidsbruk fra 4 \* n til n. Men har i begge tilfeller en algoritme som er Θ(n)

Parallellisering: gjøre flere ting på en gang. Forutsetter maskin med flere prosessorer, de fleste har dette. F.eks. vektorisering og preallokering i Matlab. Reduserer tidsbruk, endrer vanligvis ikke kompleksitet

Finne en lurere algoritme for å gjøre samme ting: F.eks. oppnå Θ(1) eller Θ(log n) i stedet for Θ(n)–Hvis n blir skikkelig stor, sparer dette mer enn de over. Dele opp programmet lurt, skille fra hverandre. Tidkrevende operasjoner som kan gjøres sjelden. Hyppige operasjoner som må gjøres raskt

**Iterasjon vs. Rekursjon**

Iterasjon: algoritmen benytter løkke(r) for stegvis løsning av problemet

Rekursjon: stegvis repetisjon oppnås ved at en funksjon kaller seg selv