

ADA512 - Cyber Physical Systems and IoT

Intro

Cyber Physical systems

"Cyber Physical Systems" (CPS) refererer til systemer hvor datamaskinbaserte algoritmer (cyber-delen) tett integreres med fysiske prosesser. Her er en enkel forklaring:

Tenk deg en moderne bil. Den har mange sensorer (for eksempel for avstandsmåling, temperatur, hastighet osv.) og aktuatorer (ting som får noe til å skje, som motorer eller bremsesystemer). Denne bilen kan også ha en datamaskin som kontinuerlig leser informasjon fra sensorene og bestemmer hvordan aktuatorene skal oppføre seg. For eksempel, hvis sensorene oppdager at bilen nærmer seg et objekt for raskt, kan datamaskinen bestemme seg for å automatisk bremse bilen.

I dette scenarioet er de "fysiske" aspektene ting som sensorene, aktuatorene, og selve bilen. Den "cyber" delen er datamaskinen og programvaren som analyserer dataen og tar beslutninger.

Så, et "Cyber Physical System" er et system hvor den digitale (cyber) delen og den fysiske verden er dypt integrert, og de samarbeider for å oppnå spesifikke mål. Disse systemene kan finnes i mange bransjer, fra helsevesen til produksjon og transport.

IOT

"IoT" står for "Internet of Things". Her er en enkel forklaring:

Forestill deg alle de vanlige tingene rundt deg – lamper, kjøleskap, dører, klokker, biler. Nå, tenk deg at hver av disse tingene har evnen til å koble seg til internettet og kommunisere med andre enheter eller systemer.

For eksempel, kjøleskapet ditt kan være koblet til internett og gi deg beskjed når melken går ut på dato. Eller en lampe som du kan slå av eller på fra telefonen din, uansett hvor du er i verden. Eller en dør som låser seg automatisk når den føler at alle har forlatt huset.

IoT handler altså om å gjøre vanlige objekter "smarte" ved å koble dem til internett, slik at de kan samle inn data, motta instruksjoner, eller interagere med andre enheter eller systemer.

Det bringer en ny dimensjon av bekvemmelighet, automatisering og innovasjon til hverdagslivet, men kommer også med utfordringer, spesielt knyttet til sikkerhet og personvern.

Nettverksteknologi

Networks of devices

- Enheter kan være direkte koblet til en eller flere enheter.
- Vanligvis har vi ikke en dedikert kobling mellom enheter, meldingen må passere flere andre enheter mellom to endepunkter – dette betyr at vi har flere koblinger mellom kilde og destinasjon.
- Vanligvis er det mer enn en rute til destinasjonen.
- Kommunikasjonsmedier mellom enheter kan variere fra kobling til kobling.
- Kommunikasjonsteknologier som brukes på forskjellige koblinger kan variere.
- En rute kan brukes for meldingen, en annen rute kan brukes for svar

Kommunikasjon mellom enheter

Wired

- Electromagnetic signals transmitted over a physical medium (physical cables)
- Types: twisted pair, coaxial, optical fiber, ...
- «one to one», «multiple access», ...

Wireless

- Electromagnetic signals (Infrared-, Radio- or Microwave waves)
- Not bounded to physical medium (travel through air, water, vacuum, ...)
- «one to one», «multiple access», ...

Nettverkstopologier

Fysisk eller logisk layout til elementene i et nettverk:

We differentiate between:

- Physical topology: physical connections and elements
- Logical topology: which devices may exchange information

Example (network 192.168.0.0/24, with no traffic filtering):

- It may be “star” with one switch at center and hosts otherwise
- Logical topology (L3) is “fully mesh” (“fully connected”) between hosts

Buss: signaler fra én node detekteres til alle noder

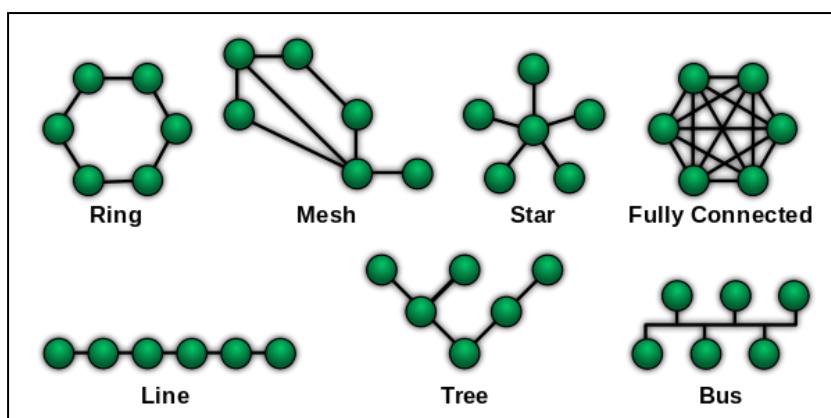
- Alle noder leser pakke destinasjonsadressen (er jeg mottakeren?). Hvis adressen samsvarer med nodenes adresse: les data, hvis adressen ikke stemmer: ignorer data

Stjerne: alle noder koblet til en sentral node, all trafikk går gjennom denne noden (bryter?)

- Koblinger mellom noder er én til én (ingen kollisjon?)
- Enkelt å legge til nye noder

Ring: hver node koblet til to naboyer, alle noder utgjør en lukket sløyfe. Signaler går gjennom nodene i en bestemt retning

- Hver node har et inn-grensesnitt og et ut-grensesnitt i en fysisk ring. Har vanligvis en "Token Passing"-ordning.
- Noden med token kan legge ved en melding til den



Kommunikasjonsprotokoller

En samling av regler og konvensjoner som datamaskiner må følge for å kunne kommunisere med hverandre over et nettverk. De definerer blant annet hvordan data skal pakkes, adresseres, overføres, mottas og feilhåndteres.

TCP/IP (Transmission Control Protocol/Internet Protocol):

- TCP er ansvarlig for å bryte ned data i mindre pakker, overføre dem over nettverket og sette dem sammen igjen i mottakerenden. IP er ansvarlig for å rute pakkene gjennom nettverket til riktig destinasjon.

HTTP (Hypertext Transfer Protocol):

- Dette er protokollen som brukes for å overføre web-sider fra en web-server til en web-klient (som en nettleser).

HTTPS (Hypertext Transfer Protocol Secure):

- Dette er en sikker versjon av HTTP, som bruker SSL/TLS protokollen for å kryptere dataene som overføres.

FTP (File Transfer Protocol):

- Dette er en protokoll som brukes for å overføre filer mellom datamaskiner over et nettverk.

SMTP (Simple Mail Transfer Protocol):

- Dette er protokollen som brukes for å sende e-post over internett.

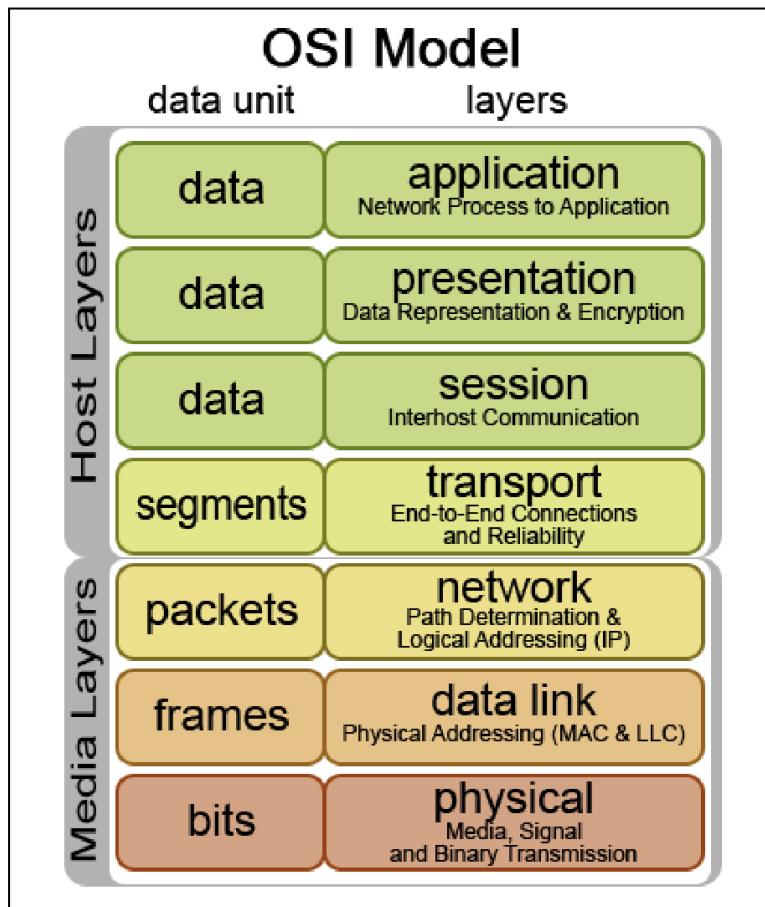
IMAP (Internet Message Access Protocol):

- Dette er en protokoll som brukes for å hente e-post fra en e-postserver.

POP3 (Post Office Protocol version 3):

- Dette er en annen protokoll som brukes for å hente e-post fra en e-postserver.

OSI model



"Formålet med denne referansemodellen for åpen systemkobling er å tilby et felles grunnlag for koordinering av standarder utvikling med formål om systemkobling, samtidig som det tillater eksisterende standarder å bli satt i perspektiv innenfor den overordnede Referansemodellen."

Application layer (L7):

- protokollene innenfor dette laget sikrer at programmer kan overføre filer, sende e-poster og ellers utføre tjenester over nettverket.
 - HTTP, HTTPS for nettlesere, FTP for filoverføring, SMTP for e-post, Telnet, SSH for terminaler

Presentation layer (L6):

- hvis to noder kommuniserer i et annet format, sikrer protokollene innenfor dette laget riktig oversettelse til et felles format. Komprimering/dekomprimering av data.
Kryptering/dekryptering av data.

Session layer(L5):

- protokollene innenfor dette laget muliggjør at brukere kan opprette forbindelser (sessions).

Transport layer (L4):

- protokollene i dette laget sikrer at data kommer frem til destinasjonen på riktig måte (segmentering, flytkontroll, ...). Store mengder data deles inn i passende segmenter og nummereres.
- Eksempler: TCP, UDP

Network layer (L3):

- protokollene letter pakkeadressering for overføring mellom flere nettverk. Protokoller for å finne passende rute hvis det er flere veier for dataene er også på dette nivået.
 - Eksempler: IPV4, IPV6, OSPF, BGP, ...
 - (Layer 2-protokoller muliggjør adressering innenfor et nettverk)

Data Link Layer (L2):

- Dette laget produserer små rammer av rå bitstrøm fra det fysiske laget. Disse rammene er merket med mottakerens (og avsenderens) adresse. Laget definerer også feilretting, prosedyrer for ny sending, prosedyrer for justering av datahastighet

Physical layer (L1):

- Det fysiske laget er den fysiske forbindelsen mellom enhetene på nettverket. Fokuserer på å sende og motta rå databiter over den fysiske forbindelsen. Spesifiserer de mekaniske og elektriske grensesnittene for å opprette og opprettholde den fysiske forbindelsen. Standardene på dette laget definerer for det meste ledninger, plugger og kontakter.

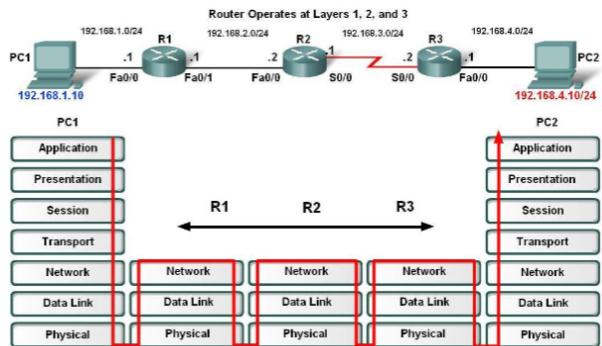
Et gitt lag i OSI-modellen kommuniserer vanligvis med tre andre OSI-lag:

- Laget rett over det
- Laget rett under det
- Peer-laget i andre nettverkstilkoblede datasystemer

Applikasjonslaget i System A kommuniserer for eksempel med nettverkslaget i System A, det fysiske laget i System A, og datalinklaget i System B.

An instance of a program on PC1 is sending a message to an instance of (another?) program on PC2. Data is passing several network devices (three routers in this case).

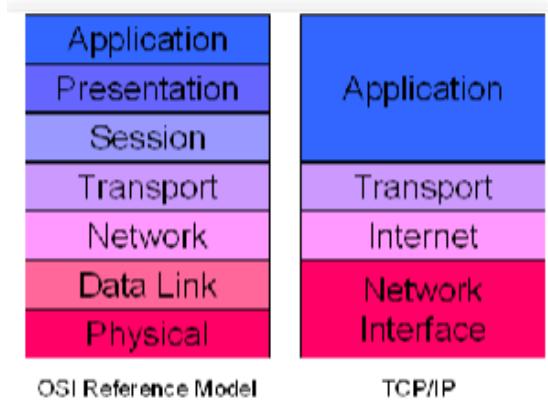
Devices are using (implementations of) different protocols that are ensuring that receiver gets the message



TCP-IP arkitektur

TCP-IP-arkitektur: et alternativ til OSI-modellen, delvis samsvarer med OSI

- TCP og IP er de mest betydningsfulle protokollene – derav navnet



4 layers:

- Application: application protocols used by process-to-process communication
- Transport:
 - Usually TCP – Transaction Control Protocol
 - Splitting messages into a number of packets
 - Labelling the packets with packet number
 - Request missed packages
 - Adjust transmission rate to a rate that gives an acceptable package loss
- Internet:
 - This layer holds IPv4 and IPv6
 - Addressing and network related services (security, errors during message-transit, ...)
- Network interface

- Correspond to L2 and L1 in OSI

Typer nettverk

Personal Area Network (PAN):

- Dette er det minste nettverket og er vanligvis begrenset til en person og enhetene de bruker, som for eksempel en smarttelefon, et nettbrett og en bærbar datamaskin. Disse enhetene kan være koblet sammen via Bluetooth eller Wi-Fi.

Local Area Network (LAN):

- Dette er et nettverk som dekker et lite geografisk område, som for eksempel et kontor, et hus eller en campus. LAN brukes ofte for å dele ressurser som filer, skrivere og internettforbindelser.

Metropolitan Area Network (MAN):

- Dette er et nettverk som dekker et større geografisk område, som for eksempel en hel by. MAN brukes ofte for å koble sammen flere LAN.

Wide Area Network (WAN):

- Dette er et nettverk som dekker et veldig stort geografisk område, som for eksempel et helt land eller kontinent. Internett er det mest kjente eksemplet på et WAN.

Identifikatorer i et nettverk

En unik identifikator (adresse) er nødvendig for at enhetene i et nettverk skal kunne identifisere og kommunisere med hverandre.

Networks: L2 network

A (L2) network will we define as group of devices (that are connected/reachable - directly or indirectly) - that may exchange information by using L2 protocol

- Your home devices (computers, TV, streamer, ...) may be L2 network
- A group of devices in a (part of a?) company/organisation/institution
 - Companies usually use many L2-networks that are connected
 - HVL networks: students, staff, management?, ...'

LAN (local area networks) are L2 networks? – often but it depends on who you talk to ☺

MAC-adresser for Ethernet-nettverk:

- Er en unik identifikator tildelt nettverkskortet i en enhet for å identifisere den på et nettverk. En MAC-adresse består av 48 bits, eller 6 bytes, som ofte er skrevet i heksadesimal format, for eksempel: 00:1A:2B:3C:4D:5E. De brukes på data link-laget i OSI-modellen for å identifisere enheter på et lokalt nettverk. Ethernet, som er den mest vanlige teknologien brukt for LAN, bruker MAC-adresser for å sørge for at data blir levert til riktig enhet.

IP-adresser for IP-nettverk (IPv4, IPv6):

- Dette er adresser som er tildelt enheter på et nettverk av nettverksadministratoren eller automatisk via DHCP (Dynamic Host Configuration Protocol). IP-adresser brukes på nettverkslaget i OSI-modellen for å rute data mellom forskjellige subnett og nettverk. IPv4-adresser består av 32 bits, mens IPv6-adresser består av 128 bits, som gir flere mulige adresser.

Ethernet

Ethernet:

- A family of network technologies for LANs (protocols within layer 1 and 2 in the OSI reference model)
- Standardized in IEEE 802.3 for wired networks and in IEEE 802.11 for wireless networks

Unique identifiers: 6x8bit identifiers

- First three bytes: producer (organizationally unique identifier)
- Last three bytes: unique identifier
- (Sidemark: see also universaly vs. locally adminstred)

MAC addresses are used for identifying sender/receiver in L2 (ethernet) networks

Det er en standard for hvordan datamaskiner og andre enheter i et lokalt nettverk (LAN) kan sende data til hverandre. Ethernet er den mest brukte LAN-teknologien i verden på grunn av dens pålitelighet, hastighet og kostnadseffektivitet.

Devices in L2 network

Computers (network adapter)

Switches (L2)

- Specialized computers
- Computers are usually connected to these
- Forwards messages (packets)
- Learning (use MAC-tables) – need for broadcast is greatly reduced
 - How does switch learn

Router (for access to Internet or other L2 networks)

- router separate L2 networks (by its operation)
- discussed later in this presentation

Hubs, Bridges and Repeaters are today largely replaced by switches/better cables

L3 networks

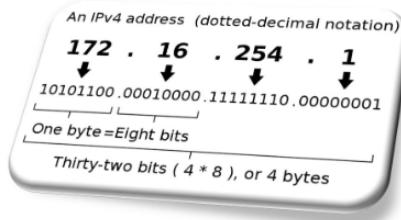
- Et (L3) nettverk vil vi definere som en gruppe enheter (som er koblet / nåbare - direkte eller indirekte) - som kan utveksle informasjon ved å bruke L3-protokoll (som igjen bruker L2-protokoller).
- L3-nettverk kan spenne over flere L2-nettverk

IP og nettverk som bruker IP

Provides the logical connection between network devices by providing identification for each device

IPv4:

- 32 bits identifiers, usually expressed as 4 numbers [0,255] separated by dot (.)
- $2^{32} = 4\ 294\ 967\ 296$ addresses
 - Lack of available addresses is a problem (IPv6 is discussed later in this course)
- 0.0.0.0 – 255.255.255.255 (but some addresses are used for special purposes)



Devices in L3 network

Routers

- Responsible for forwarding packet, based in IP-address of destination (sending on appropriate link in order to reach destination)
- May be configured to be able to learn/update route-information
- May be configured to do additional services (packet filtering/firewalls, DHCP services, ...)

L3 Switches

- Some (advanced) switches may perform (in addition to switching operation) number of responsibilities that routers do – these are called L3-switches

One may also use specialized devices that perform special operations such as security

IP - subnets

In order to make routing process (forwarding of IP-packets) manageable, IP addresses are grouped into groups that we refer to as networks (in some discussions) and subnets (in other discussions).

This is done by adding subnet mask: 32 bits of “1-s followed by 0-s”

- Subnet mask is also written as A.B.C.D where A, B, C, D are numbers in the interval [0,255] but we will see that only some numbers in this intervals may be used: 255, 252, 224, ...
- Subnet mask is also written as /N where N is number of ‘1’-s in subnet mask

Example (subnet mask):

- 11111111 11111111 11111110 00000000
- 255.255.254.0
- /23

Subnet mask logically divide network-address in two parts:

- Network identifier part (first X bit of an address that correspond to 1-s in subnet mask)
- Host identifier part (last Y bit of an address that correspond to 0-s in subnet mask)

Nettverksadresser

To hosts/verter anses som i samme nettverk hvis de har samme nettverksidentifikator.

Nettverksadressen (adressen til et nettverk) er et tall vi får når vi tar nettverksidentifikatordelen av en adresse og fyller resten (av 32 bits) med 0-er.

Nettverksadressen identifiserer alle adresser som tilhører samme nettverk (samme subnett).

- Bruken av subnettmasker deler adresseplassen i subnett.
- Ulike organisasjoner bruker forskjellige subnett/nettverksadresser.

Med hensyn til vortsdelen av en adresse:

- To enheter i samme nettverk (subnett) kan ikke ha samme vortsid (vi trenger unike identifikatorer).
- Ingen host i et nettverk kan ha bare 0-er som sin vortsid (det ville vært nettverksadressen).
- Ingen host i et nettverk kan ha bare 1-er som sin vortsid (brukt til kringkasting).

Rutere bruker nettverksadressene til destinasjonen (og ikke nøyaktige vortsadresser) når beslutningen tas om hvor en pakke skal videresendes - rutingstabeller er fylt med nettverksadresser.

Spesielle IPv4-addresser

Loopback-adresser:

- Dette er adresser som brukes for å teste nettverkskortet på en datamaskin. IP-adresser fra 127.0.0.1 til 127.255.255.255 er reservert for loopback, men 127.0.0.1 er den mest brukte. Når en datamaskin sender en melding til en loopback-adresse, sendes meldingen direkte tilbake til maskinen uten å faktisk gå ut på nettverket. Dette er nyttig for feilsøking og testing.

Private adresser:

- Dette er IP-adresser som er reservert for privat bruk, det vil si innenfor et lokalt nettverk (LAN). Disse adressene kan ikke brukes på internett. Det er tre blokker med private IP-adresser:
 - 10.0.0.0 til 10.255.255.255
 - 172.16.0.0 til 172.31.255.255
 - 192.168.0.0 til 192.168.255.255

DNS (Domain Name System)

Domain name: (human-friendly) string of text that may be translated into IP address

- Domain name registrar

DNS is a service that translates domain names into IP addresses:

- wikipedia.org → 185.15.59.224
- wikipedia.org → 2a02:ec80:300:ed1a::1: (IPv6 address)
- vg.no → 195.88.55.16

Dynamic Host Configuration Protocol (DHCP)

Manuell konfigurering av IP-adresser kan være upraktisk. DHCP er en klient-server-protokoll som brukes (av IP-verd som klient) for å automatisk oppnå tilgjengelig IP-adresse (fra DHCP-server) og relatert informasjon (maske, standard gateway, etc.) i nettverket.

Supernets

Supernet er en prosess med (logisk) kombinasjon av små nettverk i et enkelt supernetverk. Vi kan kombinere flere (sub)nettverk i et supernet som inneholder undernettverk. Eksempel:

- 192.168.0.0/24 og 192.168.1.0/24 kan kombineres til 192.168.0.0/23

Bruken av supernet kan redusere lengden på rutingtabeller. To eller flere relaterte nettverk kan bli annonsert som supernet for rutere som er "tilstrekkelig langt unna".

Merk at nettverket 192.168.0.0/16 også er et supernett for 192.168.0.0/24 og 192.168.1.0/24, men det inneholder også mange andre nettverk - noen eksempler: 192.168.2.0/24, 192.168.16.0/20, ... og mange flere.

- Å annonse "for store" supernett til rutere kan resultere i unødvendig trafikk (pakker blir droppet senere enn de kunne ha vært), eller pakken kan bli sendt "feil vei".
- Vi bør forsøke å introdusere bare supernett som ikke inneholder flere adresser som ikke er en del av undernettene som er kombinert til supernett.

Transport layer: port numbers

En IP-adresse identifiserer en host, men er ikke nok til å identifisere prosessen (eller aktiviteten innen en prosess - en tråd) (blant mange som kjører) som vil motta (eller sender) data.

Et portnummer er et tall [1 - 65535] som brukes i TCP og UDP - to transportlag (L4) protokoller.

Portnumre (sammen med IP-adresser) brukes til å entydig identifisere en kommunikasjon mellom to endepunkter.

Rutere og rutingtabeller

Hovedfunksjonen til en ruter er å videresende pakker til destinasjonen (ved hjelp av den mest passende banen - beste rute).

Rutingtabellen inneholder ruter "kjent" for ruteren.

Ruter kan konfigureres statisk (av administratoren) og/eller et ruteringsprotokoll(er) kan brukes for å dynamisk lære/oppdatere ruter.

Hver ruter grensesnitt er vanligvis i et annet nettverk (dens IP-adresse).

Rutingprotokoller

Rutingprotokoller: programmer som kjøres på rutere som (dynamisk) utveksler ruteringsinformasjon mellom rutere og foreslår de beste rutene for rutingtabellen.

- Eksempler: Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Routing Information Protocol (RIP), Border Gateway Protocol (BGP)
- Metrisk: hvor "god" en rute er - den beste ruten foreslås til rutingtabellen
 - OSPF: metrisk relatert til båndbredde
 - EIGRP: relatert til båndbredde, belastning, pålitelighet, forsinkelse og MTU
 - RIP: hop-telling
 - BGP: kompleks beslutningsprosess ☺ ("beste sti" brukes i stedet for "metrisk")
 - Administrativ avstand - foretrukket rekkefølge med hensyn til informasjonskilden
 - RIP – 120, OSPF 110, ..., Statisk 1, Tilkoblet 0

VLAN

- segmentering av sammenkoblet nettverk

Gitt et nettverk av enheter - er et VLAN en (logisk definert) undergruppe som isolerer undergruppentrafikk (L2) fra resten av nettverket. Vi kan ha forskjellige L2-nettverk koblet til samme switch (eller gruppe av sammenkoblede switcher).

Transport layer: Transmission Control Protocol (TCP)

Transport Layer Protocol (TCP):

- Adressering (porter) - for å identifisere høyere lag aktiviteter
- Mange forskjellige applikasjoner kan bruke TCP - vi må holde dataene adskilt
- Etablering, forvaltning og avslutning av forbindelse
- Datahåndtering og emballasje (segmentering, pakking av segmenter i TCP-meldinger med TCP-header)
- Overføring av data til L3-protokoll (IP)

- Gir pålitelighet og overføring (ACK av overførte data)
- Flytkontroll og overbelastningsforebygging - hvor mye data kan sendes uten å motta ACK?

Transport layer: User Datagram Protocol (UDP)

- Adressering (porter) - for å identifisere høyere lag aktiviteter
- Mange forskjellige applikasjoner kan bruke TCP - vi må holde dataene adskilt
- Datahåndtering og emballasje (segmentering, pakking av segmenter i UDP-meldinger med UDP-header)
- Overføring av data til L3-protokoll (IP)

Merk:

Begge UDP og TCP bruker portnumre, men de er forskjellige protokoller. Samme portnummer KAN brukes samtidig av TCP og UDP.

IPv6

Noen fordeler:

- Større adresseplass (2^{128} adresser)
- Eliminerer behovet for NAT - og eliminerer problemer knyttet til bruk av NAT
- Enklere headerformat
- Bruker utvidelsesheader ved behov
- Eksempler: Godkjenningsheader (AH) og innkapslingssikkerhetslast (ESP)
- Forbedret støtte for QoS og mobile enheter

QOS - quality of services

- Er beskrivelsen eller måling av den samlede ytelsen til en tjeneste

Det er:

- forme og begrense kapasitet
- sikre rettferdig tilgang til ressurser
- tildele passende prioriteringer til individuelle pakker som reiser gjennom Nettverk
- håndtere forsinkelser i dataoverføring
- administrere bufring av redundante pakker
- bestemme egenskapene til pakketap
- unngå overbelastning

Eksempler:

- Ulike protokoller – tid
- Ruter hjemme, VoIP
- Overflødighet
- Skalerbarhet

Throughput (gjennomstrøming)

- system throughput eller aggregate throughput (samlet) er summen av dataprisene som leveres til alle terminaler i et nettverk
- Throughput kan bestemmes numerisk ved å bruke kōteori, hvor belastningen i pakker per tidsenhet er betegnet som ankomstraten (λ), og reduksjonen i pakker per tidsenhet er betegnet som avgangsraten (μ).

M/M/1:

- **M:** Ankomster er en Poisson-prosess (dvs. tilfeldige ankomster med en konstant middelrate).
- **M:** Betjeningstider er eksponentielt fordelt (dvs. en konstant middelservicehastighet).
- **1:** Det er én server i systemet.

Delay

- Nettverksforsinkelse er en design- og ytelseskarakteristikk ved et telekommunikasjonsnettverk. Den angir forsinkelsen for en databit å reise over nettverket fra ett kommunikasjonspunkt til et annet.

Processing delay (Prosesseringsforsinkelse):

- tiden det tar en ruter å behandle pakkeheaderen.

Queuing delay (Køforsinkelse):

- tiden pakken tilbringer i ruteringskøer.

Transmission delay (Overføringsforsinkelse):

- tiden det tar å dytte pakkens biter på lenken.

Propagation delay (Propageringsforsinkelse):

- tiden det tar for et signal å forplante seg gjennom mediet.

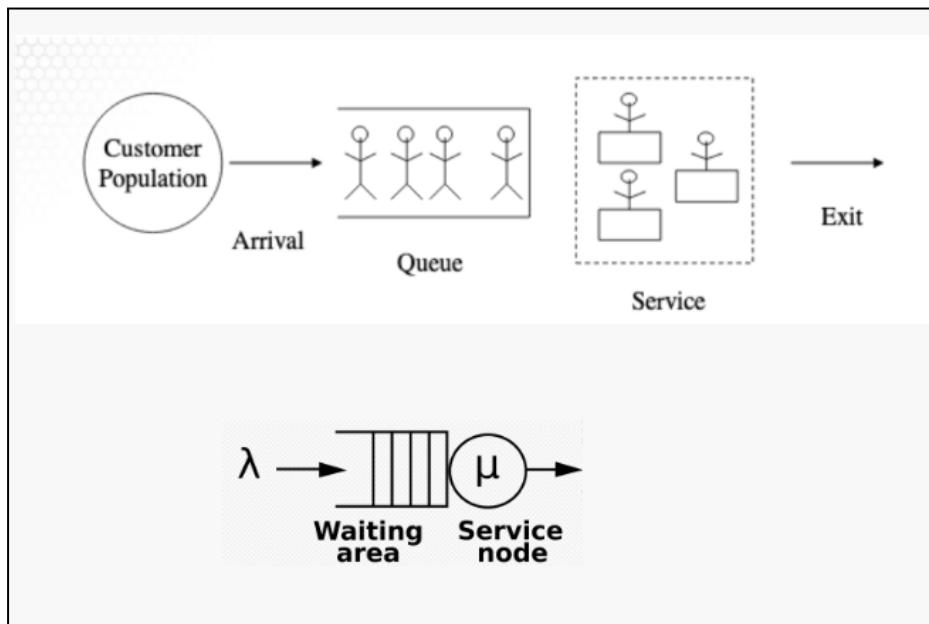
Jitter

- I elektronikk og telekommunikasjon er jitter avviket fra den sanne periodisiteten til et antatt periodisk signal, ofte i forhold til et referanse klokkesignal. I applikasjoner for gjenoppretting av klokke kalles det tidsjitter.
- Jitter er en betydelig, og vanligvis uønsket, faktor i designet av nesten alle kommunikasjonslenker.

Packet loss

- Pakketap oppstår når en eller flere datapakker som reiser over et nettverk, ikke når sin destinasjon.
- Pakketap skyldes enten feil i dataoverføring, typisk over trådløse nettverk, eller nettverksoverbelastning.
- Pakketap måles som en prosentandel av tapte pakker i forhold til sendte pakker.

Køteori (Queueing theory)



- I køteori representerer en M/M/1-kø kølengden i et system med en enkelt server, der ankomster bestemmes av en Poisson-prosess og tjenestetidene for jobbene har en eksponentiell fordeling. Modellnavnet er skrevet i Kendalls notasjon.

λ = gjennomsnittlig ankomstrate,

μ = gjennomsnittlig tjenesterate,

$\mu > \lambda$ kun i enkeltsvermodellen

Ankomstrate (λ): Dette er raten av ankomster til systemet per tidsenhet

Service Rate (μ): Dette er raten hvor tjenesten blir fullført av serveren per tidsenhet.

Trafikkintensitet (ρ): Dette er forholdet mellom ankomst- og servicehastighetene

Formel 1: Utilization parameter / Opptatt periode

$$\rho = \lambda / \mu$$

Formel 2: Sannsynligheten for null enheter i køen

$$P_0 = 1 - \lambda/\mu$$

Formel 3: Sannsynligheten for at nøyaktig ‘n’ enheter er i systemet

$$P_n = (1 - \lambda/\mu) * (\lambda/\mu)^n$$

Formel 4: Forventet antall enheter i køen / kølengde

$$L_q = \lambda^2 / (\mu * (\mu - \lambda))$$

Formel 5: Forventet ventetid i køen

$$W_q = L_q / \lambda$$

Formel 6: Sannsynligheten for at en ankomst vil måtte vente mer enn v ($v > 0$) ventetid i systemet

$$x = e^{-(\lambda - \mu)v}$$

Overview of dynamical models of CPS components

Cyber-physical systems (CPS):

- › Composed of:
 - › Digital and analog devices
 - › Interfaces
 - › Networks
 - › and the like with the natural and man-made physical world.
- › CPS Analysis and Design is challenging due to the inherent interconnected and heterogeneous combination of behaviors.
- › Suitable CPS analysis and design tools must allow a combination of:
 - › physical (or continuous dynamics)
 - › cyber (or computational components)
 - › as well as handle a variety of types of perturbations, as external disturbances, time delays, and system failures.

- › Safety & reliability specifications imposed in CPS applications embodied as stringent robustness standards, worsen the matter.
- › Hybrid system models can capture both continuous and discrete dynamics, they represent a very natural framework for the study of CPS¹.
- › This article proposes a hybrid control systems approach to analysis and design of cyber-physical systems.
- › This article provides from the literature an overview of hybrid control theory methods that are suitable for modeling, analysis, and design of CPS.
- › The ideas are illustrated in several examples throughout the article.

Dynamical models

- › Temporal evolution of the CPS variables are captured using dynamical models;
- › The state of the physical components is typically determined by variables that change according to physical time and take values from real numbers;
- › The state of the cyber components is usually defined by variables that change within the code, which is executed at discrete-time events, and that take values from discrete sets;
- › The evolution of the continuous variables is captured by *differential inclusions* while the evolution of the discrete variables is captured by *difference inclusions*;
- › These inclusions (equations) are typically nonlinear due to the complex dynamics of those variables.
- › Furthermore, conditions determining the change of the continuous and discrete variables according to the said equations (inclusions) can be captured by functions of the variables, inputs, and outputs:
- › Since the dynamical model of a system will naturally assume a solution concept attached to it, a notion of time would also be imposed on the cyber component (advantage provided by hybrid models of CPS);
- › To determine the change of continuous variables of the physical components, as well as discrete variables of the cyber components, a solution to such models will be parameterized by a notion of time;
- › Further, these models are presented briefly and illustrated in examples.

Models of physical components

- > The physical components of a CPS include
 - > the analog elements,
 - > physical systems,
 - > the environment;
 - > The dynamics of the physical component is captured by differential equations (or inclusions). Semantic proper to:
 - > environment modeling in embedded systems;
 - > the system to study in dynamical systems theory;
 - > the plant to control in control theory;
 - > The proposed model of the physical components consists of a continuous-time system, in which typically the time variable parameterizes the variables of the system, which are called states.
- > The mathematical description of the physical component is given by:
- $$\dot{z} \in F_P(z, u), \quad y = h(z, u) \quad (1)$$
- >, where:
- > z is the state variable from \mathbb{R}^{n_P}
 - > u is the input variable from \mathbb{R}^{m_P}
 - > y is the output variable from \mathbb{R}^{r_P}
 - > F_p and h are functions from $\mathbb{R}^{n_P} \times \mathbb{R}^{m_P}$
 - > $t \in \mathbb{R}_{\geq 0}$ is the time variable that parameterizes the variables of the system, which are called states.
 - > \dot{z} is the derivative of $z(t)$ with respect to $t \in [0, \infty)$.
- > In certain cases, it would be needed to impose restrictions on the state and inputs to the physical component:
- $$(z, u) \in C_P \subset \mathbb{R}^{n_P} \times \mathbb{R}^{m_P} \quad (2)$$

- > Example 2.1: A linear time-invariant model of the physical components is defined by
 - > $\dot{z} = F_P(z, u) = A_P z + B_P u,$
 - > $y = h(z, u) = M_P z + N_P u$
 - >, with A_P , B_P , M_P , and N_P as the system, input, output and feedforward matrices of appropriate dimensions.
 - > Thus, the evolution of the temperature of a room with a heater can be modeled by
$$\dot{z} = -z + [z_\Delta \quad 1] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$
 - >, where:
 - > z is the temperature of the room
 - > $u = (u_1, u_2)$ is the input: u_1 represents the heater as turned on ($u_1 = 1$) or off ($u_1 = 0$), u_2 is the temperature outside the room
 - > z_Δ is the heater capacity (constant).
- > Example 2.2: Dubins vehicle, a kinematic model of a vehicle (particle) moving on a plane with a constraint (minimum turning radius). The particle dynamics is :
- $$\dot{z}_1 = \nu \sin z_3, \quad \dot{z}_2 = \nu \cos z_3, \quad \dot{z}_3 = u$$
- >, where:
- > $(z_1, z_2) \in \mathbb{R}^2$ denotes position
 - > $z_3 \in \mathbb{R}$ denotes orientation with respect to the vertical axis
 - > ν is the velocity of the vehicle
 - > $u \in [-\bar{u}, \bar{u}]$ is the angular velocity input, $\bar{u} = \frac{\nu}{\rho}$,
 - > ρ is the minimum turning radius
- > The mathematical model can be as in (1)-(2), of a hybrid system with inputs:
- $$F_P(z, u) = \begin{bmatrix} \nu \sin z_3 \\ \nu \cos z_3 \\ u \end{bmatrix}$$

Models of cyber components

- › The **cyber components of a CPS** include those in charge of performing:
 - › computations
 - › implementing algorithms
 - › transmitting digital data over networks
- › These tasks involve **variables that change at discrete events only** and performed by:
 - › code (at the software level)
 - › logic-based mechanisms (circuit level)
- › Further the following notations are used:
 - › $\eta \in \Upsilon \subset \mathbb{R}^{nc}$ is the state variable of the cyber component
 - › $v \in \mathcal{V} \subset \mathbb{R}^{mc}$ is the input variable affecting the cyber component
 - › $\zeta \in \mathbb{R}^{rc}$ is the output variable
 - › κ is the output function (defines ζ)
 - › G_C is the set-valued map (the right-hand side of a **difference inclusion**)
- › **Deterministic case FSM:** A finite state machine consists of the following objects:
 - An input alphabet Σ
 - A finite set of states Q
 - A set of output symbols Δ
 - An output function $\kappa : Q \rightarrow \Delta$; and
 - A transition function $\delta : Q \times \Sigma \rightarrow Q$.
- › Characteristics:
 - › q_0 is the **initial state**.
 - › $Q_\infty \subset Q$ is the **set of final states** that, at times, is imposed.
 - › $\delta(q, v)$ is the **transition function** defined for each state q and each input v .
 - › By convention $\delta(q, \emptyset) = q$, known as the **basis condition**.
 - › $\delta(q, ab) = \delta((q, a), b)$ defines δ as an **extended transition function** that can be evaluated for input strings.
- › The **mathematical description of the cyber component** is given by:

$$\eta^+ \in G_C(\eta, v), \quad \zeta = \kappa(\eta, v) \quad (4)$$
- › In certain cases, restrictions on the state and inputs to the cyber component are needed that could be modeled by

$$(\eta, v) \in D_C \subset \Upsilon \times \mathcal{V} \quad (5)$$
- › **1) Pure Finite State Machines:** A finite state machine (FSM) or deterministic finite automaton (DFA) is a system with inputs, states, and outputs taking values from discrete sets that are updated at discrete transitions (or jumps) triggered by its inputs.
- › Using the following notations of the FSM:
 - › q denotes the states (or mode)
 - › v denotes the inputs
 - › r denotes the outputs
- › The δ and κ functions are defined as
 - › **Total functions**, when defined for all points of their domains
 - › **Partial functions**, when defined not for all points of their domains
- › Thus, given a FSM and the initial state q_0 , a transition to a state $q_1 = \delta(q_0, v)$ is performed when an input v is applied to it. After the transition, the output of the machine is updated to $\kappa(q_1)$. This mechanism can be captured by the **difference inclusion/equation**:

$$q^+ = \delta(q, v) \quad \zeta = \kappa(q) \quad (6)$$
- › ,with $(q, v) \in Q \times \Sigma$
- › This model corresponds to the model of the cyber components in (4)-(5) with

$$\eta = q, \quad \Upsilon = Q, \quad \mathcal{V} = \Sigma,$$

$$G_C = \delta, \quad D_C = \Upsilon \times \mathcal{V}$$

- Example 2.3: The finite state machine in Figure 1 has two modes and one input. Its output is equal to the current mode. It is given by the difference equation in (6) with:

$$Q = \{A, B\}, \quad \Sigma = \{0, 1\}$$

$$\delta(q, v) = \begin{cases} A & \text{if } v = 1 \\ B & \text{if } v = 0 \end{cases}, \quad \kappa(q) = q$$

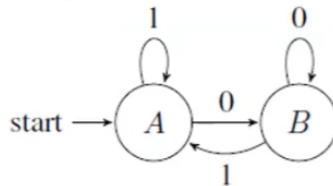


Fig. 1. A finite state machine with two modes and one input.

- Nondeterministic case FSM: can be treated similarly by using a set-valued transition function $\delta : Q \times \Sigma \rightrightarrows Q \times \Delta$ with

$$\delta(q, ab) = \bigcup_{q' \in \delta(q, a)} \delta(q', b)$$

- 2) Finite State Machines with Conditional Structures as Guards: In applications, it is often desired that the FSM jumps are triggered on conditional structures, e.g., perform a transition when $v < 0$.

- Conditional structures can be added to a pure FSM by allowing for an infinite input alphabet Σ and including the conditional structure as a *guard*.

- Guard is a boolean-valued expression that evaluates to:
 - true when the transition is enabled
 - false otherwise.

- To define FSM with transitions according to conditional structures, let the function $l : Q \times \Sigma \times \Delta \rightarrow \mathbb{R}$ be a testing function for the transition condition for each mode or state q .

Models of systems at interface between physical and cyber components

- The models describing the behavior of the physical and the cyber components have significantly different dynamics.
- Due to this, their interconnection requires interfaces that condition and convert the signals appropriately.
- Further, we propose mathematical models for some of the most widely used interfaces.
- In the section to follow, these models will be used to define a complete model of a cyber-physical system.
- 1) Analog-to-Digital Converters (ADCs): are sampling devices commonly used to provide measurements of the physical systems to the cyber components.
- Thus, ADCs sample their input, which is usually provided by a sensor measuring the system output y at a given periodic rate T_s^* .
- When the timer reaches the value of the sampling time T_s^* , the timer is reset to zero and the sample state is updated with the inputs to the sampling device.
- The proposed sampling device model has both continuous and discrete dynamics.
- If the timer state has not reached T_s^* , then the dynamics are such that the timer state increases continuously with a constant, unitary rate. When T_s^* is reached, the timer state is reset to zero and the sample state is mapped to the inputs of the sampling device.
- To implement this mechanism, we employ a timer state $\tau_s \in \mathbb{R}_{\geq 0}$, an input $v_s \in \mathbb{R}^{r_p}$ and a sample state $m_s \in \mathbb{R}^{r_p}$. Then, the model of the sampling devices is

$$\dot{\tau}_s = 1, \quad \dot{m}_s = 0 \quad \text{when } \tau_s \in [0, T_s^*] \quad (12)$$

$$\tau_s^+ = 0, \quad m_s^+ = v_s \quad \text{when } \tau_s \geq T_s^* \quad (13)$$

- › 2) Digital-to-Analog Converters (DACs): perform the task of converging digital signals into analog equivalents.
- › The digital signals in the cyber components need to be converted to analog signals for their use in the physical world.
- › One of the most common models for a DAC is the zero-order hold model (ZOH).
- › A ZOH converts a digital signal at its input into an analog signal at its output. Its output is updated at discrete time instants, typically periodically, and held constant in between updates, until new information is available at the next sampling time.
- › We will model DACs as ZOH devices with dynamics similar to (12)-(13). Notations:
 - › the timer state $\tau_h \in \mathbb{R}_{\geq 0}$
 - › the sample state $m_h \in \mathbb{R}^{rc}$
 - › the inputs $v_h \in \mathbb{R}^{rc}$
- › The following notations are used:
 - › information occurs at instants $\{t_i\}_{i=1}^{i^*}$,
 - › satisfying
$$T_N^{*\min}, T_N^{*\max} \in [0, \infty]$$

$$T_N^{*\min} \leq T_N^{*\max}$$
 - › i^* is the number of transmission events, which might be finite or infinite.
- › A mathematical model capturing the DAC mechanism is given by

$$\dot{\tau}_N = -1, \quad \dot{m}_N = 0 \quad (16)$$

when $\tau_N \in [0, T_N^{*\max}]$

$$\tau_N^+ \in [T_N^{*\min}, T_N^{*\max}], \quad m_N^+ = v_N \quad (17)$$

when $\tau_N \leq 0$
- › This proposed DAC model is given by

$$\dot{\tau}_h = 1, \quad \dot{m}_h = 0 \quad \text{when } \tau_h \in [0, T_h^*] \quad (14)$$

$$\tau_h^+ = 0, \quad m_h^+ = v_h \quad \text{when } \tau_h \geq T_h^* \quad (15)$$
- › 3) Digital Networks: The information transfer between the physical and cyber components, or between subsystems within the cyber components, might occur over a digital communication network.
- › The communication links are not capable of continuously transmitting information, but rather they can only transmit sampled information at discrete time instants.
- › Combining the ideas in the models of the converters in the previous sections, we propose a model of a digital network link that has a variable that triggers the transfer of information provided at its input, and that stores that information until new information arrives.
- › Note: The models in this section are given in terms of differential and difference equations / inclusions. The reason for this is that they are at the interface between physical and cyber components.
- › These type of models are referred to as hybrid inclusions and will be the class of models we will employ to study cyber-physical systems in the remainder of this article (not developed further here).
- › In general, the interface will be modeled as a hybrid inclusion with state λ , input w , output ψ , where F_I defines the continuous dynamics on C_I and G_I the discrete dynamics on D_I of the interface:

$$\dot{\lambda} \in F_I(\lambda, w) \quad \text{when } (\lambda, w) \in C_I \quad (18)$$

$$\lambda^+ \in G_I(\lambda, w) \quad \text{when } (\lambda, w) \in D_I \quad (19)$$

$$\psi = \varphi(\lambda) \quad (20)$$

Combining models of physical and cyber components

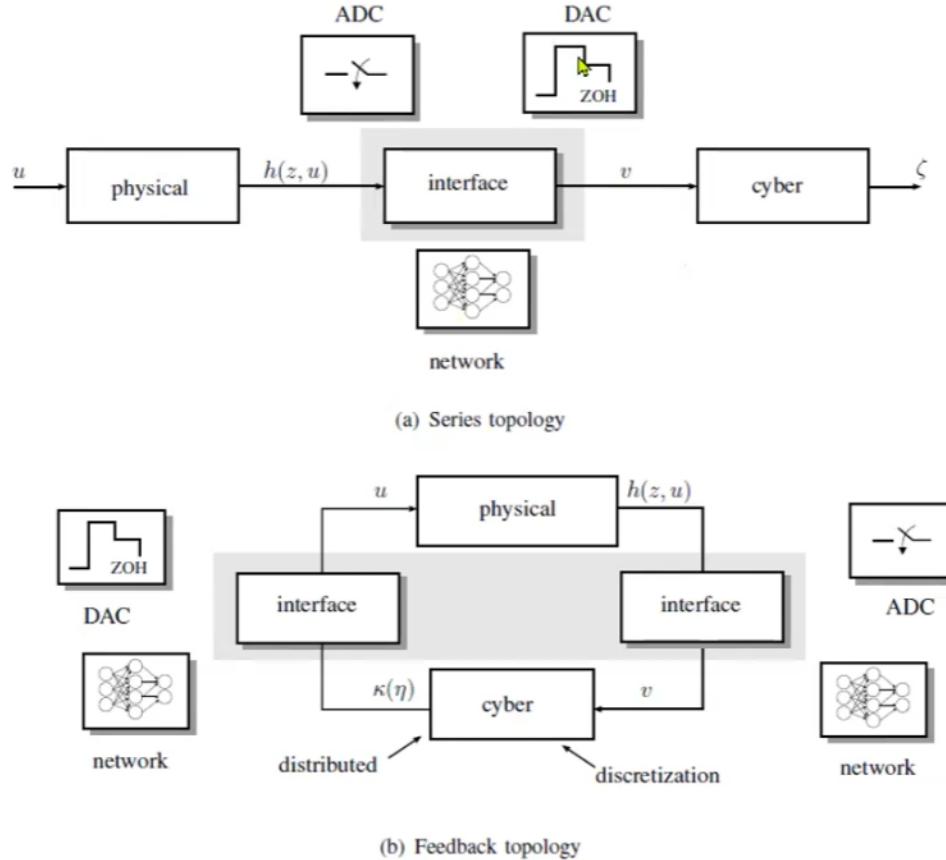


Fig. 2. **Cyber-physical systems:** series and parallel interconnections between a plant (part of the physical component), controller (part of the cyber component), and interfaces/converters/signal conditioners (part of both the physical and cyber components).

Control systems engineering

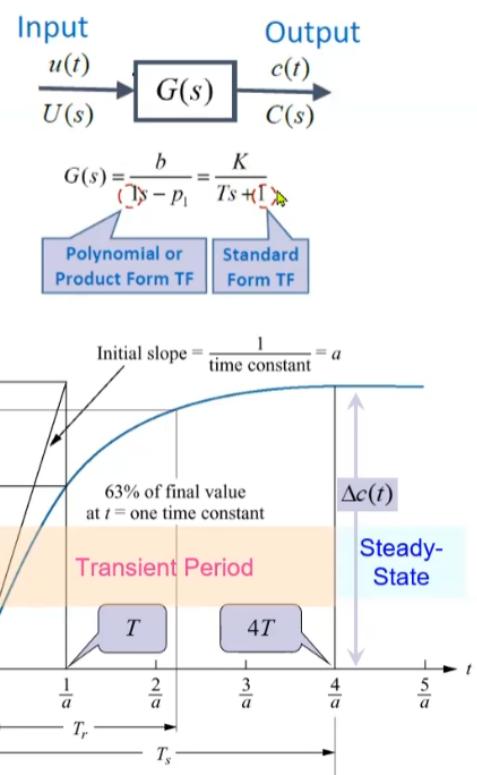
- The aim in control engineering: to control the system.
- Process description (mathematical modeling) through:
 - › Differential equations
 - › Transfer functions (in Laplace-domain/plane)
 - › Block diagram
- NB: These modeling methods are equivalent.
- Design of a control system is based on mathematical modeling of the process and control systems engineering techniques:
 - › In time domain, frequency domain (Laplace-domain), etc.
 - › According to pole placement, the shape of the system response, stability margins
 - › Desired system behavior

First-order systems

4.3. First-Order Systems: Fundamental Properties

- First Order System Characteristics:
 - Gain (Forsterkning) $K = \Delta c(t)/\Delta u(t)$: the ratio of the steady-state variation of the output $\Delta c(t)$ to the steady-state variation of the input $\Delta u(t)$. Here, the input is a unit step (enhetssprang);
 - Time Constant (Tidskonstant) $T = 1/a$: the time in which the step response $c(t)$ reaches 63% of the steady-state value $c(\infty)$;
 - Rise Time (Stigningstid) $T_r \approx 2.2/a = 2.2T$: the time in which the output increases from 10% to 90% of the steady-state value $c(\infty)$;
 - Settling Time (Innstillingstid) $T_s \approx 4/a = 4T$ (for 2% steady-state error interval): The time length of the transient period.
- NB: The settling time is defined with respect to the desired Steady-State Error interval (as a rule of thumb 2% of the steady-state value).

*) It is treated only as a numerical example in the previous section 5.2 Test Input Signals.



Exercise First-Order System

Consider the dynamical models of the following first order processes:

- **Charging of a capacitor** having the transfer function of the system given by

$$G(s) = \frac{V_c(s)}{V_s(s)} = \frac{1}{RCs + 1}$$

- **Grindstone driven by a motor without gear** where the transfer function is

$$G(s) = \frac{\Omega(s)}{T_M(s)} = \frac{1}{Js + D}$$

Do the following:

- a) What are the poles, zeroes, the gain, and the time constant of those systems? Use the literary form.
- b) Choose the following values for the characteristics (constants) of both systems: $R = 16 \text{ k}\Omega$, $C = 10 \mu\text{F}$, $J = 0.16 \text{ kg m}^2$, and $D = 1 \text{ N m/(rad/sec)}$. What are now the poles, zeroes, the gain, and the time constant of both systems? Compare the transfer function of the one system with the other. Comment;
- c) Choose one system and the unit step as the input of the system. Calculate the rise time, the settling time for 2% steady-state error and the final value (steady-state). What is the step response of the system?
- d) Plot the unit step response in Matlab. Use the LTI viewer and check the values calculated at 1c) and 1d) from the plot of the unit step response.

IoT

- > IoT describes physical objects (or a group of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices.
- > Over Internet or other communication networks
- > It has evolved due to the convergence of multiple technologies
- > Increasing concerns about privacy and security

IoT provides a vision of the all-communicating world with:

- > Increased data volume
- > Increased data-rate
- > Extended battery life
- > Reduced end-to-end latency
- > Possibilities of:
 - Integration of sensors and embedded system with cyber-physical system (CPS)
 - Device-to-device communications (D2D)
 - 5G wireless systems with IoT as a center

Early in 2010, there were three characterized IoT visions discussed:

- › Internet Oriented vision -> connectivity between the objects
- › Things Oriented vision -> generic objects
- › Knowledge Oriented vision -> how to represent, store and organize information

ITU defines IoT as: "from anytime, anyplace connectivity for anyone; we will now have the connectivity for anything" (anytime, anywhere, anything)

In a nutshell, the ultimate objective -> **to plug and play smart objects**

IoT Layers and Protocol Stacks

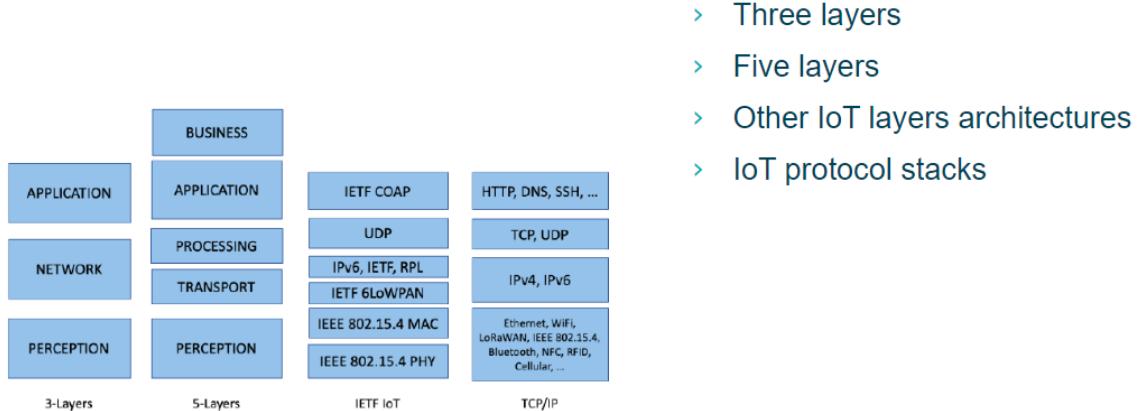
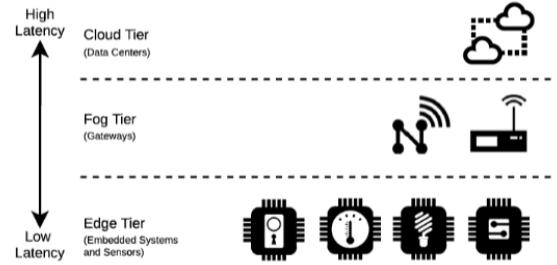


Figure 1. IoT architectures and protocol stacks

IoT Architecture Model

Traditional IT computing model: a direct connection between end devices and the cloud has limitations for IoT networks. Thus, fog computing has been utilized to mitigate challenges.

- › Lower level are the IoT devices.
- › Fog tier provides contextual location awareness, low latency, geographical distribution, heterogeneity support, interoperability, real-time interactions support, scalability and agility of federated, fog-node clusters.
- › On the top are the data centers and the Cloud.



Some 5G-enabled IoT Examples

The higher data rates possible in 5G-IoT make it possible for the implementation of data-hungry and computation-intensive Artificial Intelligence (AI) algorithms for various user applications, for example:

- › Virtual speech recognition
- › Video classification
- › Making intelligent decisions in real-time

Big Data Processing Enhancement:

- › AI algorithms can extrapolate and make more accurate decisions after finding patterns from a data set based on the currently stored and processed data.
- › Ever-present connectivity provided by 5G leads to the creation of an enormous amount of data.
- › The objective of the 5G Intelligent IoT is to process huge amounts of data intelligently, optimize the communication channels, making intelligent decisions uninterrupted.
- › Examples:
 - Natural language processing
 - Face recognition

Smart Transportation Systems

- > The integration of 5G with IoT lets vehicles access the internet in a more efficient way
- > Possible for:
 - > Self-driven vehicles
 - > Smart transportation system:
 - Provides new features for more control
 - Makes efficient traffic routes based on sensors in traffic lights
 - Helps with reducing manual labor -> reduces costs

Utilizing Abundant Data of Inter-Connected IoT devices

- > Used for prediction of accidents and crimes
- > Helps innovate new ideas
- > Real-time data allowed devices to be controlled without much human interaction
- > Possible to provide surveillance
- > Bring value to society

Diverse spørsmål

Smarthus

Prøv å skissere plan for smarthus. Skriv om funksjoner, sensorer, nettverk og sikkerhet:

1. Funksjoner i Smarthuset

a. Automatisert Belysning:

- Tilpasser lysstyrken basert på tid på dagen og tilstedeværelse i rommet.
- Mulighet for stemmestyring og fjernkontroll.

b. Temperaturkontroll:

- Smarte termostater som justerer temperatur basert på vaner og værforhold.
- Integrering med vinduer for automatisk åpning/lukking for temperaturregulering.

c. Sikkerhet og Overvåking:

- Kameraer og bevegelsessensorer for overvåking.
- Smarte låser og alarmsystemer.

d. Energiadministrasjon:

- Overvåking og styring av energiforbruk.
- Integrering med fornybare energikilder som solpaneler.

e. Underholdning og Media:

- Sentralisert kontroll av TV, musikksystemer og andre medieenheter.
- Tilpasset brukeropplevelse basert på preferanser.

f. Helse og Velvære:

- Sensorer for luftkvalitet og vannkvalitet.
- Automatiserte trenings- og velværeprogrammer.

2. Sensorer og Enheter

Bevegelsessensorer: For lysstyring og sikkerhetssystemer.

Temperatursensorer: For HVAC (Heating, Ventilation, and Air Conditioning) systemer.

Fuktighetssensorer: For å overvåke og kontrollere inneklima.

Røyk- og Gassensorer: For brann- og gasslekkasjesikkerhet.

Vannsensorer: For å oppdage lekkasjer og oversvømmelser.

Dør- og Vindussensorer: For sikkerhet og energieffektivitet.

3. Nettverk og Kommunikasjon

Wi-Fi og Zigbee: For trådløs kommunikasjon mellom enheter.

Ethernet: For pålitelig og sikker nettverkstilkobling.

5G/4G: For ekstern tilgang og backup kommunikasjon.

Mesh-nettverk: For å sikre stabil og utvidet dekning i hele huset.

4. Sikkerhet

a. Datakryptering:

- Sikre at all kommunikasjon mellom enheter er kryptert.

b. Sikkerhetsprotokoller:

- Implementere avanserte sikkerhetsprotokoller for å forhindre uautorisert tilgang.

c. Regelmessige Oppdateringer:

- Sørge for at all programvare og firmware er oppdatert for å motstå sikkerhetstrusler.

d. Fysisk Sikkerhet:

- Sikre at alle enheter er beskyttet mot fysisk manipulering.

e. Nettverkssikkerhet:

- Bruke brannmurer og nettverksovervåking for å beskytte mot cyberangrep.

f. Personvern:

- Implementere retningslinjer for personvern og sikre at brukerdata håndteres forsvarlig.

5. Brukergrensesnitt og Kontroll

Smarttelefon-App: For fjernstyring og overvåking.

Stemmekontroll: Integrering med assistenter som Alexa, Google Assistant.

Automatiseringsregler: For å sette opp scenarier og automatiseringer basert på brukerpreferanser.

6. Integrering og Skalerbarhet

Åpen Plattform: For å tillate integrering med fremtidige enheter og teknologier.

Modulær Utforming: For enkel oppgradering og tilpasning til nye behov.

7. Vedlikehold og Support

Fjernovervåking og Diagnostikk: For å raskt identifisere og løse problemer.

Brukerstøtte: Tilgjengelig support for tekniske spørsmål og veiledning.

MAC-addresses as global identifiers

Could we use MAC-addresses as global identifiers (between ALL devices)?

(Do we need IP addresses?)

- Nei, vi kan ikke bruke MAC-adresser som globale identifikatorer mellom alle enheter på internett. MAC-adresser (Media Access Control adresser) er unike identifikatorer tildelt nettverkskort av produsenten, og de er ment for å fungere innenfor det lokale nettverket (LAN). De brukes for å identifisere enheter på et fysisk nettverksnivå og er ikke designet for å bære informasjon om nettverkspllassering utover det lokale nettverket.
- IP-adresser (Internet Protocol adresser), derimot, er designet for å fungere på internett. De gir en måte å identifisere enheter på et globalt nivå og inneholder informasjon som er nødvendig for å rute trafikk over internett. IP-adresser sikrer at data sendes til riktig destinasjon over komplekse nettverk som strekker seg utover lokale nettverk.
- Kort sagt, mens MAC-adresser er viktige for lokal nettverkskommunikasjon, er IP-adresser avgjørende for global kommunikasjon over internett. Begge adressetypene spiller viktige, men forskjellige roller i nettverkskommunikasjon.

Er våre datamaskiner i samme nettverk?

For å avgjøre om to datamaskiner er i samme nettverk, sammenlign nettverksadressene deres. Hvis de er identiske, er datamaskinene i samme subnett. Dette forutsetter at begge maskinene bruker samme subnettmasker.

Eksempel:

Din IP-adresse: 192.168.1.10, Subnettmaske: 255.255.255.0

Min IP-adresse: 192.168.1.15, Subnettmaske: 255.255.255.0

Begge har nettverksadresse: 192.168.1.0

Begge er i samme nettverk.

Gitt adresser, kalkuler...

For gitte adresser (med masker):

192.168.0.10 / 24

192.168.1.10 / 23

192.168.2.10 / 23

192.168.10.10 / 22

192.168.12.254 / 22

Kalkuler:

Nettverks- og vertsidentifikatordel

Adresse av et nettverk

Adresserom av det nettverket - mulige adresser (brukt eller ikke)

Spørsmål: Er noen av adressene i samme nettverk?

For å løse denne oppgaven, vil jeg først beregne nettverksadressen for hver IP-adresse ved å bruke subnettmasken. Deretter vil jeg sjekke om noen av IP-adressene faller innenfor samme adresserom. La oss begynne med beregningene.

Her er resultatene av beregningene:

For 192.168.0.10 / 24:

Nettverksadresse: 192.168.0.0

Broadcastadresse: 192.168.0.255

Antall mulige verter: 256

Brukbar vert IP-område: 192.168.0.1 - 192.168.0.254

For 192.168.1.10 / 23:

Nettverksadresse: 192.168.0.0

Broadcastadresse: 192.168.1.255

Antall mulige verter: 512

Brukbar vert IP-område: 192.168.0.1 - 192.168.1.254

For 192.168.2.10 / 23:

Nettverksadresse: 192.168.2.0

Broadcastadresse: 192.168.3.255

Antall mulige verter: 512

Brukbar vert IP-område: 192.168.2.1 - 192.168.3.254

For 192.168.10.10 / 22:

Nettverksadresse: 192.168.8.0

Broadcastadresse: 192.168.11.255

Antall mulige verter: 1024

Brukbar vert IP-område: 192.168.8.1 - 192.168.11.254

For 192.168.12.254 / 22:

Nettverksadresse: 192.168.12.0

Broadcastadresse: 192.168.15.255

Antall mulige verter: 1024

Brukbar vert IP-område: 192.168.12.1 - 192.168.15.254

Når det gjelder spørsmålet om noen av adressene er i samme nettverk, finner vi at adressene 192.168.0.10/24 og 192.168.1.10/23 er i samme nettverk. Dette er fordi de deler samme nettverksadresse (192.168.0.0) og broadcastadresse (192.168.1.255), noe som indikerer at de er deler av et større /23 nettverk.

Hvordan gjøres beregningene?

La meg forklare hvordan nettverksinformasjonen for IP-adresser med subnettmasker beregnes:

Nettverksadresse: Dette er den første adressen i subnettet og brukes til å identifisere nettverket selv. Det er ikke en brukbar adresse for en vertsenhet. Nettverksadressen blir beregnet ved å utføre en logisk "AND"-operasjon mellom IP-adressen og subnettmasken.

Broadcastadresse: Dette er den siste adressen i subnettet og brukes til å sende data til alle enheter på nettverket samtidig. Broadcastadressen beregnes ved å sette alle bits i vartsdel av adressen til 1, basert på subnettmasken.

Antall mulige verter: Dette er antall brukbare adresser for enheter i nettverket. Det beregnes som $2^{(32-\text{subnettmask})} - 2$. Tallet 2 trekkes fra fordi nettverksadressen og broadcastadressen ikke kan tildeles verter.

Brukbar vert IP-område: Dette er rekkevidden av IP-adresser som kan tildeles til enheter på nettverket. Det starter med nettverksadressen pluss 1, og slutter med broadcastadressen minus 1.

Overlapping av nettverk: To IP-adresser er i samme nettverk hvis nettverksadressene deres er like, og de har en overlappende rekkevidde av IP-adresser. Dette kontrolleres ved å se på nettverks- og broadcastadressene for å se om de faller innenfor samme område.

Her er en steg-for-steg forklaring på hvordan du kan beregne disse manuelt:

1. Konverter både IP-adressen og subnettmasken til binær format.
2. Utfør en binær "AND"-operasjon mellom IP-adressen og subnettmasken for å finne nettverksadressen.
3. For å finne broadcastadressen, inverter subnettmasken (endre alle 0-er til 1-er og vice versa) og utfør en binær "OR"-operasjon mellom dette og IP-adressen.
4. For å finne antall mulige verter, tell antall 0-er i subnettmasken (i binær), hev 2 til makten av dette antallet, og trekk fra 2.

- Den brukbare vert IP-området er enkelt å finne ved å legge til 1 til nettverksadressen og trekke fra 1 fra broadcastadressen.