

Noen lærdommer fra Internettets historie (Hanseth & Lyytinens design-prinsipper for II)

INF 3290, mandag 29. oktober 2012
Margunn Aanestad, margunn@ifi.uio.no

Plan for i dag:

- ▶ **Tema: Internettets historie**

- ▶ Eksempel på en vellykket etablering av en global informasjonsinfrastruktur
 - ▶ Hva gjorde det mulig?
 - ▶ Hva kan vi lære?

- ▶ **Designprinsipper for dynamisk kompleksitet:**

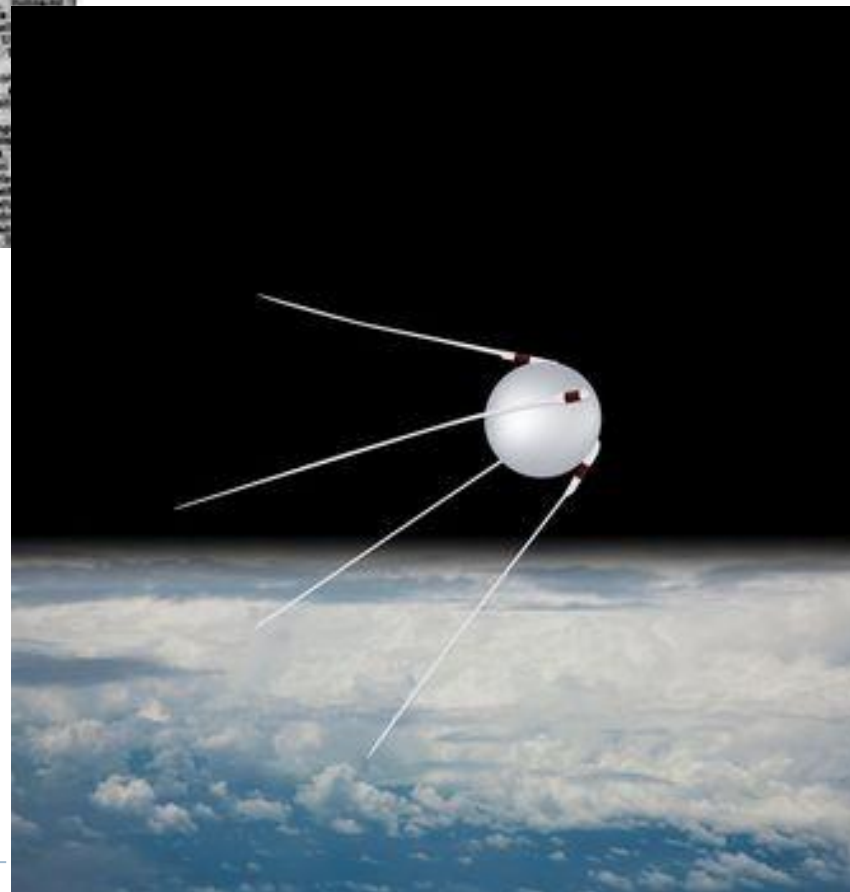
- ▶ (Hanseth og Lyytinen, 2010)
- ▶ Prinsipper for å etablere informasjonsinfrastrukturer, basert på hva man kan lære av Internettets historie



Oktober 1957: Sputnik

Februar 1958:
ARPA etablert
(Advanced Research Projects
Agency)

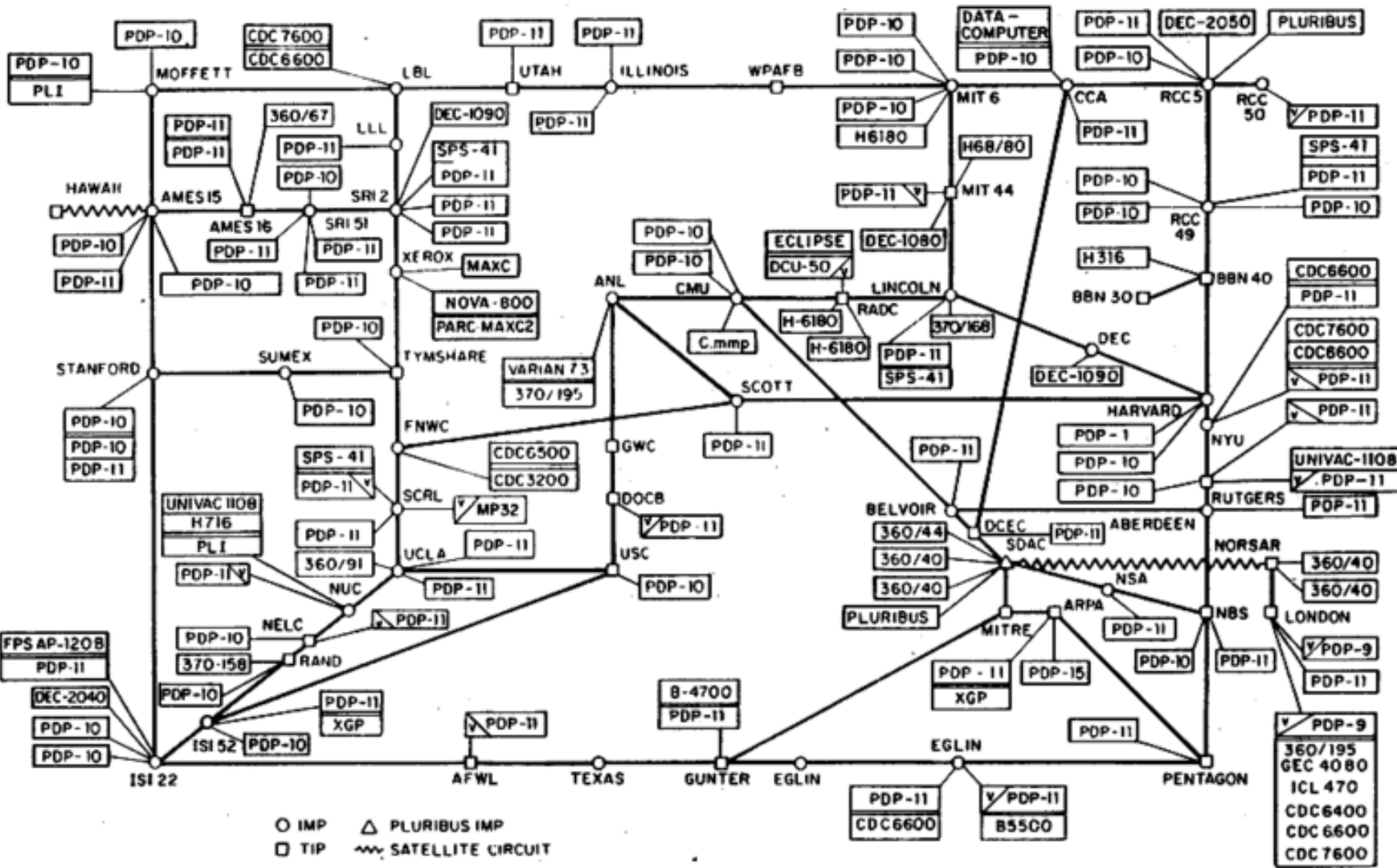
Senere DARPA



▶ Kort historisk oversikt

- ▶ 1969: De 4 første ARPANET-nodene ble koblet sammen
- ▶ 1972: e-post tatt i bruk for at utviklerne skulle kommunisere
- ▶ 1973: internasjonale noder (London og Kjeller, Norge)
- ▶ Fra Arpanet til Internet
 - ▶ TCP/IP-protokollen tatt i bruk ca 1982/3
 - ▶ Det ble et større nettverk (men fortsatt for forskere) rundt 1985
 - ▶ Oslonett (IFI, 1991) første åpne webserver utenfor universitets-miljø (privatpersoner, småbedrifter)
 - ▶ Domain Name System (1983)
 - ▶ World Wide Web (1991, CERN)
 - ▶ ... osv.

ARPANET LOGICAL MAP, MARCH 1977



(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

NAMES SHOWN ARE IMP NAMES, NOT NECESSARILY) HOST NAMES

Internetts historie

<http://prezi.com/jxskp5e3-b4s/internet-growth/>

En annen måte...

- ▶ **Brukere = utviklere**
 - ▶ Dvs. insentiver og innsats var tett koblet
- ▶ **Gradvis teknologisk utvikling**
 - ▶ Først basal teknologi for fildeling og meldings-utveksling
 - ▶ Med internetts vekst kom nye utfordringer (for eksempel behov for DNS, IPv6, nye organisatoriske former osv.)
- ▶ **Åpen arkitektur**
 - ▶ Åpent for heterogene nettverk (Inter-networking), → kunne "spise" sine konkurrenter)
 - ▶ Ende-til-ende arkitektur (utvikling delegert til brukere/'markedet')
- ▶ **Krav om "running code" før løsningen ble standard**



Internett-standarder

Slagord: (the IETF Credo)

”We reject: kings, presidents and voting. We believe in: rough consensus and running code”

Internett-standarder kalles ”RFC’er” (RFC+nummer)

”Request For Comments”

<http://www.ietf.org/rfc/rfc2031.txt>



Internett-standarder

Hva skiller Internett-standardene fra vanlige standarder?

Læringsorientert, erfaringsbaser, iterativ prosess

Flere nivåer av "modenhet" på standardene:

Archival standard – informational standard - experimental standard –
proposed standard – draft standard – standard

Ulike grader av "tvang":

Required – recommended – elective – limited use – not
recommended.



Hanseth og Lyytinen (2010)

- ▶ Hanseth og Lyytinen sier:
 - ▶ Før har IT-forskere gjort deskriptive (beskrivende) case-studier, ofte av problematiske IT-prosesser
 - ▶ Nå trenges en **design-teori** for informasjonsinfrastrukturer – hvordan skal man bygge dem?

- ▶ Kjerneutfordring: å håndtere dynamisk kompleksitet
 - ▶ De formulerer 5 designprinsipper (og 19 'regler')
 - ▶ Disse er basert på Complex Adaptive Systems-teori
 - ▶ Internettets historie: et case å lære av/illustrere prinsippene

Dynamisk kompleksitet:

Problembeskrivelse (sitat fra abstractet):

”It’s are controlled by emergent, distributed and episodic forms of control.”

“It’s evolutionary dynamics are nonlinear, path dependent and influenced by network effects and unbounded user and designer learning.”



To sentrale design-utfordringer

▶ ”Bootstrappings”-utfordringer

- ▶ Skal man lykkes i å etablere noe som helst, må det gi en viss verdi til de første som skal ta det i bruk
- ▶ Hvordan gjør man det når verdien er avhengig av mange brukere? (nettverks/kommunikasjonsteknologier)

▶ Vekst/utviklings/tilpasnings-utfordringer

- ▶ Satser man bare på ad hoc design og lokale løsninger, vil man støte på problemer (stagnasjon). Design må forholde seg til fremtidig endring i både skala og funksjonalitet
- ▶ Hvordan gjør man det når man ikke vet hva som vil skje?

Design-prinsippier:

For bootstrappings-problemet:

1. Design initially for usefulness
2. Draw upon existing installed base
3. Expand installed base by persuasive tactics

For adaptiv vekst-problemet:

4. Make each IT capability simple
5. Modularize the II by building separately its principal functions and sub-infrastructures using layering and gateways

▶ Design principles for the bootstrap problem:

- ▶ 1. Design initially for usefulness
- ▶ 2. Draw upon existing installed base
- ▶ 3. Expand installed base by persuasive tactics

Prinsipp 1:

Man hadde store visjoner/scenarier, men laget først enkle løsninger (fjern-innlogging, filoverføring, epost) som hadde direkte bruksverdi for utviklerne selv

Prinsipp 2:

- TCP/IP kunne kjøre på ulike underliggende nettverksløsninger (radio, satellitt, datanettverk, modem over telefonlinjer..)

TCP/IP bundlet med UNIX BSD

Web: ikke bare html-dokumenter, men kunne embedde andre data (feks fra databaser) i html ved hjelp av CGI – økte bruksverdien dramatisk

▶ Design principles for the adaptability problem:

- ▶ 4. Make each IT capability simple
- ▶ 5. Modularize the II by building separately its principal functions and sub-infrastructures using layering and gateways

Til prinsipp 4:

'Simplicity' var et uttalt krav til løsningene (i den første RFC'en)

Minimale protokoller – liten risiko for feil/tvetydighet i implementasjon

Til prinsipp 5:

Enkle arkitektur-prinsipper (for eksempel ende-til-ende-tenkning, modularisering)

Lagdeling: transport-, tjeneste- og applikasjons-infrastruktur

-Åpnet for innovasjon 'oppå' TCP/IP

-Muliggjorde distribuert håndtering for eksempel W3C for webteknologi

Koble parallelle II ved hjelp av gateways (transisjon IPv4 til IPv6)