

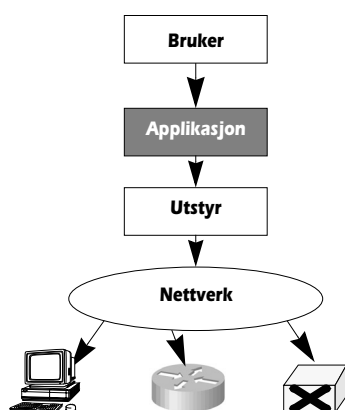
Analyse, arkitektur og design av nettverk

Dette er 3. artikkel i en serie med fokus på bygging av moderne lokalnett: Behovsanalyse, krav, komponenter, innhold, design og styring. Serien er praktisk orientert på den måten at den gir råd og vink med hensyn til hvordan problemstillingene kan angripes for å komme frem til optimale resultater på kortest mulig tid.

I forrige artikkel diskuterte vi nettverksanalyse på et overordnet nivå, og presenterte metoder for å definere grunnleggende krav til infrastrukturen. Vi kom raskt frem til at ulike vinklinger ikke bare leder til forskjellige resultater, men også høyst ulik kompleksitet i prosessen: Det er rimelig å velge tjenester som utgangspunkt for analysene, mens teknologien er et hjelpemiddel, ikke et mål i seg selv.

Tjenester implementeres av én eller flere applikasjoner hvis behov er hva infrastrukturen til slutt skal tilfredsstille. Vårt neste naturlige trinn er derfor å se nærmere på nettopp disse behovene: Foreta en behovsanalyse på applikasjonsnivå – og deretter fortsette nedover gjennom den hierarkiske strukturen vi har valgt.

Behovsanalyse



POS – Point Of Sales, kassasystemer

Illustrasjonen i marginen har fulgt oss siden første artikkel i serien, og visualiserer en sammenheng som er spesielt viktig i forbindelse med behovsanalysen: Riktignok er det brukernes behov vi til slutt skal tilfredsstille, men i tillegg til å være lite spesifikke er de også personavhengige (subjektive). Dette gir få relevante holdepunkter for en konkret kravspesifikasjon. I realiteten er det applikasjons- og tjeneste-utviklerne som skal håndtere brukernes behov, mens vi – i nettverksarkitekturen – skal ta hensyn til neste nivå, applikasjonenes behov.

Applikasjonsnivå

Samtidig med at behovene på applikasjonsnivå er mer spesifikke og konkrete, kan de fortsatt være subjektive. Spesielt kommer dette til uttrykk i forbindelse med relativ prioritering: Hva er virksomhetskritisk og hvorfor? Svaret kommer an på hvem vi spør og kanskje sågar tidspunktet. Dette representerer ingen stor utfordring i praksis, men er viktig å være klar over. Vi arbeider fortsatt ikke med absolutte kvantiteter og fakta. Ett av resultatene fra analyseprosessen er nettopp å få plassert slike 'subjektiviteter' der de hører hjemme.

Som et bidrag til denne prosessen er det hensiktsmessig å kategorisere applikasjonene i behovsklasser, for eksempel:

- ✓ Virksomhetskritisk
- ✓ Interaktiv
- ✓ Transaksjonsorientert
- ✓ Sann tid / ikke sann tid
- ✓ Responskritisk (telefoni, multimedia, POS-systemer⁸)

⁸ Kassasystemer (POS-systemer) er et spesielt godt eksempel i denne sammenhengen. De hører hjemme i de fleste kategorier: Responskritisk, virksomhetskritisk, sann tid og transaksjonsorientert.

Klassene er ikke disjunkte, men representerer på sett og vis ulike dimensjoner. Videre er det innlysende at vi både i analyseprosessen og i andre sammenhenger med fordel kan gruppere applikasjoner som har lignende karakteristika og dermed behov. Er organisasjonen spredt på flere geografiske steder, kan det også være hensiktsmessig å foreta geografisk gruppering – i henhold til hvor brukerne finnes eller hvor utstyret som betjener applikasjonene er lokalisert.

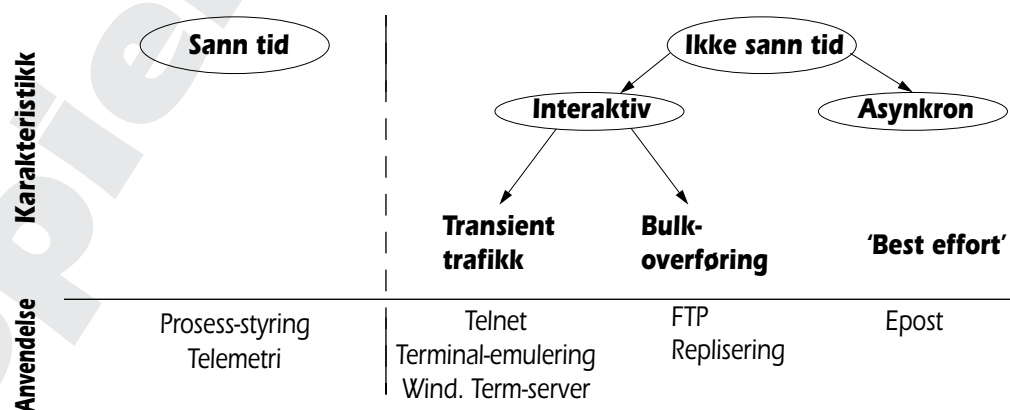
Følgende oversikt utdyper hva vi legger i de ulike klassene, og kan med fordel tilpasses den aktuelle situasjonen.

Virksomhetskritisk signaliserer behov for spesielt høy grad av pålitelighet – i betydningen 'evne til nøyaktig og konsistent levering av informasjon uten avbrudd og forsinkelse'. Typiske evalueringer går på faktorer som:

- ✓ Tapt omsetning (POS-systemer, nettsalg, telefonsalg etc.)
- ✓ Tap av informasjon (logging, trafikkontroll, registrering, nettsalg etc.)
- ✓ Tap av følsomme data (personinformasjon, medisinsk informasjon etc.)
- ✓ Tap av liv (styring av biltrafikk, lufttrafikk etc.)

Sanntids-applikasjoner har stringente krav med hensyn til respons, og er typisk å finne i forbindelse med styringssystemer og datainnsamling. Historisk har slike styringssystemer primært vært å finne i industrielle sammenhenger, men finnes nå rundt oss på alle kanter – i heisen, i bilen – og ikke minst i styringssystemer for datamaskiner og nettverk.

Interaktive (samhandlende) applikasjoner betyr at brukere er involvert. At to applikasjoner begge er interaktive betyr imidlertid ikke at deres belastningskarakteristika eller ressursbehov har noe felles, et forhold som illustreres i figuren nedenfor.



Ikke-interaktive applikasjoner har tradisjonelt hatt betegnelser som *batch*-programmer eller 'satsvise applikasjoner' – uttrykk med opprin-

nelse helt tilbake til hullkortenes dager. Windows-generasjonen har kun unntaksvis noe forhold til disse begrepene, og dagens tenåringer har laget sitt eget uttrykk: 'Boter', en forkortelse for 'roboter' eller 'automater'. Hva barnet kalles spiller ingen rolle så lenge vi er enige om betydningen, og klassen dekker logisk nok alt som ikke er interaktivt: Tjenerprosesser, bakgrunnsprosesser, programmer som kjøres automatisk på spesielle tidspunkter eller i spesielle situasjoner – og så videre. Gode eksempler er automatisk sikkerhetskopiering, replisering, epost-agenter, Web-tjenere, datainnsamling fra overvåkingsystemer og så videre i det uendelige.

At slike programmer normalt er usynlige, gjør at de gjerne får beskjeden oppmerksomhet, og i en del tilfeller blir fullstendig glemt – helt til de stopper eller feiler. De blir ikke mindre viktige av den grunn, og spiller en kritisk rolle i vår hverdag – og dermed i behovsanalysen.

Tabell 1 Eksempel på gruppering av applikasjoner etter karakteristika (behov).

Applikasjon	Virksomhets-kritisk	Interaktiv	Sann-tid	Trans.-orientert	Respons-kritisk
Grafikk-intensive applikasjoner (CAD/CAM, bildebehandling etc.)		✓			✓
Telemetri, kontroll, styring	✓	✓	✓		
Distribuert databehandling (typisk Office-verktøy på PCer)	✓	✓			
CRM, ERP, regnskap	✓	✓		✓	✓
Web-aksess, programvare-utvikling		✓			
Telefoni, multimedia	✓	✓	✓		✓
Dataoverføring (backup, replisering o.l.)					
Driftsverktøy	✓	✓	(✓)		
POS (kassasystemer)	✓	✓	✓	✓	✓

Tabellen ovenfor illustrerer hvordan applikasjoner kan grupperes etter behov (ytelseskarakteristika) – en oppstilling som er nyttig både på grunn av oversikten den gir og som referanse/kontroll på ulike stadier i analyse- og design-prosessene: Er alle med? Er de hensyntatt i samsvarende med behovene?

Igjen er vår oppstilling for et eksempel å regne. I en aktuell situasjon kan det være hensiktsmessig å gruppere annerledes og/eller ha flere grupper og klasser. Ta for eksempel de 20 mest kjente eller viktigste applikasjonene i din virksomhet, kategoriser dem, og bruk disse gruppene når resten av samlingen skal klassifiseres. Bli ikke overrasket om gjennomgangen avdekker 5 ganger flere applikasjoner enn forventet. Dette er regelen, ikke unntaket. I en organisasjon med noen hundre eller flere ansatte og fartstid over 10 år, vil tallet normalt være tresifret.

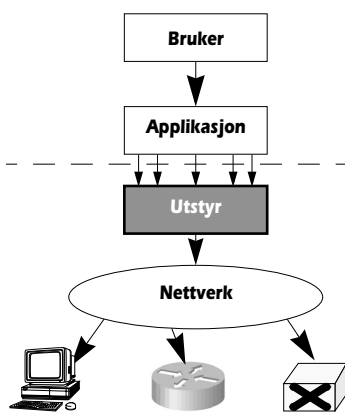
Oppsummering

Applikasjonsanalysen gir oversikt over karakteristika, krav og behov for applikasjonene som skal betjenes av den underliggende infrastrukturen. Kunnskapen kan med fordel tabell-oppstilles omtrent på samme måte som vi har gjort med gruppene og klassene ovenfor, der hver applikasjon karakteriseres i fire (eller flere) 'dimensjoner':

- ✓ Plassering (geografisk)
- ✓ Ytelseskrav/karakteristika
- ✓ Type
- ✓ Klasse

Denne kunnskapen blir inngangsdata til neste nivå i kjeden (se figur i margin).

Behovsanalyse – utstyr



Jo lenger ned i kjeden vi kommer, desto mer spesifikke blir både krav og behov. For analyse- og designprosessen er dette en lettelse i den forstand at vi i økende grad får fakta i stedet for estimater eller subjektive betraktninger å holde oss til. At disse 'fakta' i mange tilfeller har sin opprinnelse i tidligere estimater, spiller ingen rolle for prosessen og arbeidet, men understreker viktigheten av at det gjøres en grundig jobb hele veien.

Utstyret som skal betjene applikasjonene, havner naturlig i én av tre velkjente kategorier:

- ✓ **Brukerutstyr:** PCer/klientmaskiner/klienter inkl. bærbare, arbeidsstasjoner, Apple Macintosh og PDAer som er aktive på nettverket.
- ✓ **Tjenere:** Regneressurser, lagringsressurser, applikasjonstjenere m.m.
- ✓ **Spesialutstyr:** Projektorer (videokanoner), kamera-utstyr, skannere, utskriftsenheter, medisinsk utstyr, superregnere, oppgavespesifikt håndholdt utstyr (f.eks. strekkode-skannere), produksjonsutstyr m.m.

Mens klientutstyr generelt og i noen grad tjenere er fleksible med hensyn til geografisk plassering, er det motsatte typisk tilfelle med spesialutstyr, spesielt tungt produksjonsutstyr. Dette påvirker vår fleksibilitet når arkitekturen skal utvikles.

Fastlegging av utstyrsbehov betyr å etablere ytelsesparametre for utstyrskomponenter som tilfredsstillende de krav applikasjonene stiller. Ytelse har mange ulike ansikter i den forbindelse: Det kan være regnekraft, hukommelse, båndbredde, lagringskapasitet, antall samtidige brukere – eller, som vanligvis er tilfelle – en kombinasjon. Selv om størrelsene er kvantifiserbare og målbare, er det langt fra alltid mulig å skaffe dem til veie med rimelig innsats. En såkalt 'standard Intel-server' er – når vi kommer ned på dette nivå – ikke så veldig standard likevel. Mellom 20 og 50 ulike parametre, hvorav de fleste er utenfor kontroll eller fatteevne for de fleste av oss, påvirker ytelsen i tilstrekke-

lig grad til å kunne ødelegge et detaljert regnestykke. Vi kan lett forholde oss til diskens type, grensesnitt og rotasjonshastighet, til prosessorklokke og -type, og til hukommelsesmengde, men hva med L2- og L3-cache, *chipset*, FSB-hastighet, hukommelsestype og -bredde eller BIOS? Som vi skal se senere, dukker tilsvarende problemstillinger opp i forbindelse med sammenkoblingsutstyret i nettverket: Buss-type, båndbredde i bakplan, svitsje-metodikk og så videre.

Parametre og ytelse som ikke på rimelig vis lar seg måle, indikerer at detaljeringsnivået er for stort. Vi må gå et hakk tilbake, finne mer generelle skalaer og målemetoder og akseptere deres omtrentlighet – som typisk kan forbedres kraftig ved bruk av erfaring. Den enkle – og ofte riktige – veien til målet er å overdimensjonere utstyret tilstrekkelig til at de ukjente faktorene havner godt innenfor den takhøyden som etableres. For 5 eller 10 år siden kunne dette være problematisk fordi utstyrs-kostnadene på denne måten ble drevet i været. I dag er forholdet et annet. Ytelse i ulike inkarnasjoner er stort sett billig og kan i betydelig grad ødsles for å komme raskere, sikrere eller rimeligere til målet.

Behovsanalyse – nettverk

Kombinasjonen utstyr/applikasjoner/geografisk plassering sørger for grunnleggende inngangsdata til behovsanalysen for nettverket, som er det siste og laveste nivå i vårt hierarki. I tillegg – og like viktige – er følgende nettverksspesifikke forhold:

- ✓ Det finnes i de fleste tilfeller et eksisterende nettverk som legger føringer på neste generasjon på en rekke ulike måter – for eksempel: Utnyttelse av investeringer, leverandørvalg, båndbredde-begrensninger (over avstander), teknologivalg (ATM, Ethernet, fiber/kabel etc.).
- ✓ En eksisterende infrastruktur har et tilhørende drifts- og styringsapparat som legger føringer på hvilke valg som er optimale eller mulige for fremtiden.
- ✓ Skalerings- og ytelses-problematikk: Forventet utvikling i etterspørsel etter båndbredde og andre nettverks-spesifikke ytelser (QoS, sikring m.m.).
- ✓ Samspill mellom nettverk: TCP/IP og Ethernet er dominante teknologier, men ikke enerådende – og selv der andre teknologier er borte, finner vi utfordringer knyttet til samspill mellom teknologi-generasjoner.

Å starte med blanke ark er naturligvis en ideell situasjon, men forekommer sjelden. Faktorene ovenfor er derfor viktige inngangsdata som setter rammene for arkitektur og design. Målsettingen er å ende opp med den mest effektive løsningen, ikke den mest elegante, avanserte eller 'tøffe'.

Funksjonelle krav

Vi har tidligere understreket viktigheten av å bevare fokus på helheten gjennom en slik prosess: Det er lett å bli oppslukt av detaljene, utfordringene, tjenestene og teknologien – i en slik grad at vi til slutt ikke ser

skogen for trær. Tallrike moderniserings-prosjekter har møtt veggen nettopp av denne årsak, til tross for at utgangspunktet i mange tilfeller var det aller beste.

Overordnede funksjonelle krav gjelder for hele den resulterende løsningen, ikke bare for de laveste (teknologi-orienterte) nivåene. De er like innlysende som de er lette å glemme eller skyve på i kampens hete, og hører typisk hjemme i tre hovedkategorier:

- ✓ **Budsjetter og økonomi:** Selvfølgelig restriksjoner som må påvirke prosessen fra begynnelse til slutt. Vi har allerede nevnt behovet for å utnytte eksisterende investeringer. Videre har vi rammer for hvilke investeringer som kan gjøres, når de kan gjøres, og ikke minst: Driftskostnader – som faretruede ofte blir underestimert av teknologi-kåte prosjektledere. Argumentet om raskt fallende teknologikostnader slår alltid godt an, men gjelder i første rekke utstyr, i mindre grad verktøy, programvare – og personale.
- ✓ **Styring og sikkerhet:** Historisk har styring og sikkerhet kommet i etterkant – ikke bare av design, men også implementasjon av nettverket. Naivitet eller manglende fremsyntet har kun en del av skylden for dette forholdet. Veien er blitt til mens vi har gått, og dimensjoner, utbredelse og avhengighet har jevnt og trutt aksentuert behovene – først for kontroll, siden for sikkerhet. I dag er det uaktuelt å utvikle en nettverksarkitektur som ikke også omfatter en styringsarkitektur og en sikkerhetsarkitektur. Vi diskuterte førstnevnte i detalj i artikkelserien om SLM, *Service Level Management*, i Mellvik-Rapporten nr. 66 - 73.⁹
- ✓ **Nye tjenester:** At hele organisasjonen er bruker av og avhengig av infrastrukturen, er en selvfølge. Mindre innlysende er det at nettverket i det nye årtusen også skal betjene universelle, virksomhetskritiske tjenester som telefoni, telefaks og video-distribusjon. Dette må få følger på alle nivåer – tjenester, teknologi, styring og ikke minst budsjett-rammer. Dessuten introduserer forandringen et delvis ukjent element i ligningen, fordi de fleste organisasjoner og spesialister har beskjeden erfaring med overføring av de nye tjenestene til datanettverket, og konsekvensene for infrastruktur og drift.

Oppsummering

Behovsanalysen handler om å finne, samle og evaluere behovene som skal tilfredsstilles av infrastrukturen – som består av nettverk, systemer, sammenkoblingskomponenter og programvare. Systematisering av behovene på ulike nivåer er nødvendig for å få oversikt og se sammenhenger. Vårt valg av inndeling i nivåer – bruker, applikasjon, utstyr og nettverk – er basert på erfaring, men er langt fra den eneste måten. Hvilken inndeling som velges, er ikke det viktige her, men at metoden passer til målsettingen og organisasjonen.

⁹ Se spesielt nr. 67 side 20. De tre første artiklene i serien er tilgjengelige i ON-LINE utgaver av Mellvik-Rapporten se side 35.

Når vi i de neste artiklene tar fatt i selve nettverksanalysen, kommer de systematiserte behov og krav vi nå har identifisert, stadig til nytte – som styrepinner og rammeverk.

Neste utgave

For 15 år siden besto en nettverksanalyse i hovedsak av konkrete målinger av hva som foregikk på nettverket. Oppgaven er fortsatt vesentlig mer teknisk av natur enn trinnene vi har vært igjennom hitil, men er samtidig langt mer komplisert og sammensatt enn hine dagers målinger.

Vi begynner med å samle inn og systematisere data om dagens situasjon, eller – dersom vi er så heldige å skulle starte med blanke ark – med å konkretisere og kvantifisere behovene. Datainnsamlingen vil i en del tilfeller overlape behovsanalysene vi allerede har foretatt, hvilket ikke er overflødig, men fungerer som en kontroll.

Videre skal vi utvikle målestokker for kravene, som i sin tur overføres til ytelsene som skal leveres. Slike målestokker er spennende og krevende fordi de må være konkrete: Det som ikke kan måles, kan heller ikke kontrolleres, testes eller faktureres. ■