

Analyse, arkitektur og design av nettverk

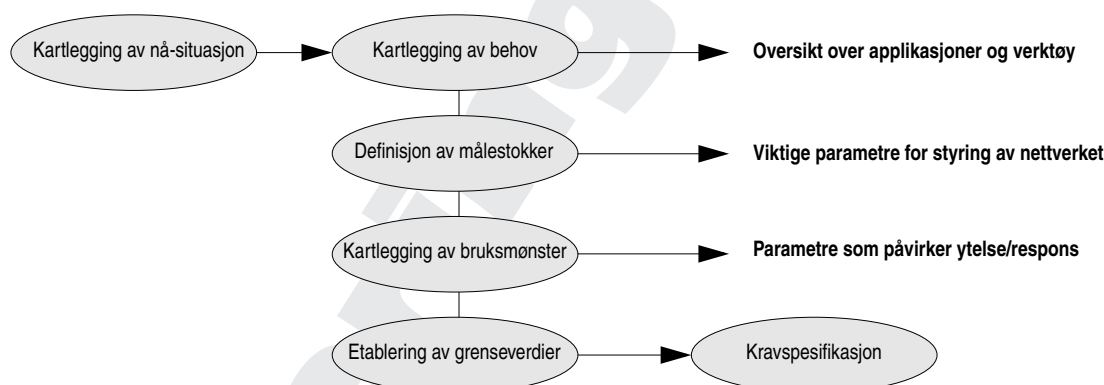
Dette er 4. artikkel i en serie med fokus på bygging av moderne lokalnett: Behovsanalyse, krav, komponenter, innhold, design og styring. Serien er praktisk orientert på den måten at den gir råd og vink med hensyn til hvordan problemstillingene kan angripes for å komme frem til optimale resultater på kortest mulig tid.

Selve nettverksanalysen består av en rekke faser. To av dem er behovsanalyse og kartlegging av funksjonelle krav, som vi gjennomgikk avslutningsvis i forrige utgave. Dermed har vi også svart på et spørsmål som uunnværlig dukker opp i kjølvannet av denne floraen av begreper: Hva er forskjellen på behovsanalyse for nettverket og nettverksanalyse? Den første er kort og godt en del av den andre – et trinn i en større prosess.

Behovsanalysen på nettverksnivå avslører hva vi trenger, mens videre analyse skal fortelle både hvor vi står og hvor vi skal. En viktig del av prosessen er å avdekke hva som finnes – av kapasitet, trafikk, problemer, flaskehalsen og så videre. Kombinasjonen munner ut i en kapasitetsplan som i sin tur blir grunnlag for vår nettverksarkitektur.

Analyseprosessen

Noe forenklet kan nettverksanalyse-prosessen, som vi skal gjennomgå i denne artikkelen, illustreres som vist i figur 2.



Figur 2 Selve analyse-prosessen består av 6 distinkte faser.

Kartlegging av nåsituasjon

Et innlysende utgangspunkt for enhver analyse – uansett sammenheng: Hvor står vi?

- ✓ Definisjon av problemet eller utfordringen – sett fra organisasjonens synspunkt og for oss som skal utføre arbeidet. Begge er like viktige, og det spiller ingen rolle for viktigheten hva vår rolle er – en intern IT-gruppe, -avdeling, konsulent eller kontraktør.
- ✓ Prosjektets omfang: Tørre fakta – om antall brukere, noder, applikasjoner etc.

- ✓ Teknisk situasjonsbeskrivelse: Hvilke teknologier er i bruk, hvor homogen er utstyrssammensetningen og så videre.
- ✓ Geografi: Hvordan fordeler brukere, applikasjoner og utstyr seg over lokasjonene som skal dekkes? Med mindre organisasjonen er liten (5-25 personer) er dette nødvendig, selv om alle medarbeidere befinner seg i samme bygning. At fordelingen blir særdeles enkel og oversiktlig under små forhold, tar vi med som en implisitt fordel.

Med i grunnlaget hører også rammebetingelser – økonomiske, fysiske, kapasitetsmessige, eventuelt andre ressurser, samt hvilken frihetsgrad vi har til å velge bort eksisterende teknologier og tilhørende utstyr. Klare rammer reduserer sjansene for at vi i prosessen gjør valg som viser seg å ikke kunne realiseres, og bidrar på den måten til å gjøre veien til målet kortere.

Et annet forhold som også har med rammebetingelser å gjøre, er valg av hvilket ytelsesnivå vi vil eller kan legge oss på: Høyt eller normalt. Behovsanalysen for applikasjoner (se forrige artikkel) vil normalt avsløre om vi trenger høy ytelse, eller om 'normal' er tilstrekkelig. Behov for høy ytelse forårsakes som regel av én eller noen få applikasjoner, brukere og/eller systemer. Grafikk-intensive anvendelser og multimedia er typiske drivkrefter i denne forbindelse. Dersom analysen ikke avslører noen klar ytelsesmessig behovsprofil, kan det bli nødvendig å vurdere begge, evaluere forskjellene og til slutt gjøre et valg ut fra kostnader, fordeler og ulemper.

Vi har allerede på dette punkt tilstrekkelig informasjon til å kunne gi 'kunden'⁴ en pekepinn om hva som blir å finne i den andre enden av tunnelen – skape riktige forventninger:

- ✓ En kortfattet og foreløpig evaluering og karakteristikkk av utfordringen.
- ✓ Et foreløpig estimat for prosjektets ressursbehov og tidsrammer.

Det er for tidlig å gå i dybden, og forsøk på å gjøre det vil virke mot sin hensikt. Målsettingen er å bringe til torgs opplagte utfordringer og observasjoner, slik at eventuelle urealistiske forventninger fra kundens side kan avsløres. Altfor mange prosjekter havner på galt spor fordi forventninger aldri eller altfor sent synkroniseres med hva som er mulig og realistisk.

Kartlegging av behov

I tillegg til de behovsanalysene vi allerede har foretatt, er det som regel ønskelig med direkte kontakt med brukere og ledere for å avdekke andre reelle og potensielle behov. Organisasjonens størrelse og IT-verktøyenes rolle er styrende for hvordan dette bør gjøres – alternativene er blant annet spørreskjemaer, telefonintervjuer, én-til-én samtaler, gruppemøter, brainstorming-sesjoner og så videre. Jo mer tid som

⁴ Det er hensiktsmessig å bruke betegnelsen 'kunde' også når egen organisasjon betjenes.

brukes, desto mer relevant og verdifull blir informasjonen. Samtidig er det kritisk å løpende vurdere verdien av det som kommer frem, og å stoppe i tide.

Siden synspunktene som fremkommer i slike sammenhenger, alltid er subjektive, er filtreringen og 'veiingen' viktig. Se etter 'røde flagg' som indikerer misforståelser, feilfokuseringer eller overlagt villedning i forhold til virkeligheten. Det er naturlig at den enkeltes oppgaver og omgivelser overprioriteres i egne øyne, og veiing av informasjonen er like naturlig. Likeledes vil vi ofte møte overlagte uklarheter som er ment å påvirke veivalg i en retning eller grad som ikke er påkrevet.

Eksempler som bør påkalle ekstra kritisk oppmerksomhet i en slik behovsanalyse, er:

- ✓ Misbruk av begrepet 'sann tid' (se behovsanalyse for applikasjoner i forrige artikkel).
- ✓ Tilgjengelighets-krav i prosenter – uten tilleggs-kvalifikasjoner: 99,99% høres fint ut, men vet personen som fremmer kravet hva det betyr og hvorfor 2 desimaler er riktigere enn 1 eller 3?
- ✓ 'Høy ytelse' – uten ytterligere kvalifikasjoner.
- ✓ Krav som spriker eller er inkonsistente fra person til person som utfører samme type arbeid med samme verktøy.
- ✓ Åpenbart urealistiske krav eller forventninger.

Å systematisere behovene slik at de kommer over på en nyttig form, er en oppgave i seg selv. Dersom datamengden er stor, kan det være hensiktsmessig å finne kommersielle verktøy som er laget for nettopp slike situasjoner.

Definisjon av målestokker

Sjelden får vi flere hevede øyenbryn enn når begrepet 'målestokk' kommer på banen. Vi har lett for å glemme at de fleste målestokker i hverdagen er subjektive – og høyst variable selv for én og samme person. Det som er viktig i det ene øyeblikket, blir uviktig i det neste – fordi noe annet dukker opp som er enda viktigere.

For å kunne starte, drive og hale i land et prosjekt som involverer mange personer og betydelig ressurser, er det nødvendig å definere parametre og å etablere målestokker som for det første er relevante og for det andre objektive (målbare). For eksempel er 'høy tilgjengelighet' subjektivt, mens 99,9% er objektivt – gitt at prosenten lar seg måle eller beregne.

MTBF – Mean Time Between Failure
 MTTR – Mean Time To Repair/
 Recover

Andre eksempler på slike parametre med tilhørende målestokk er:

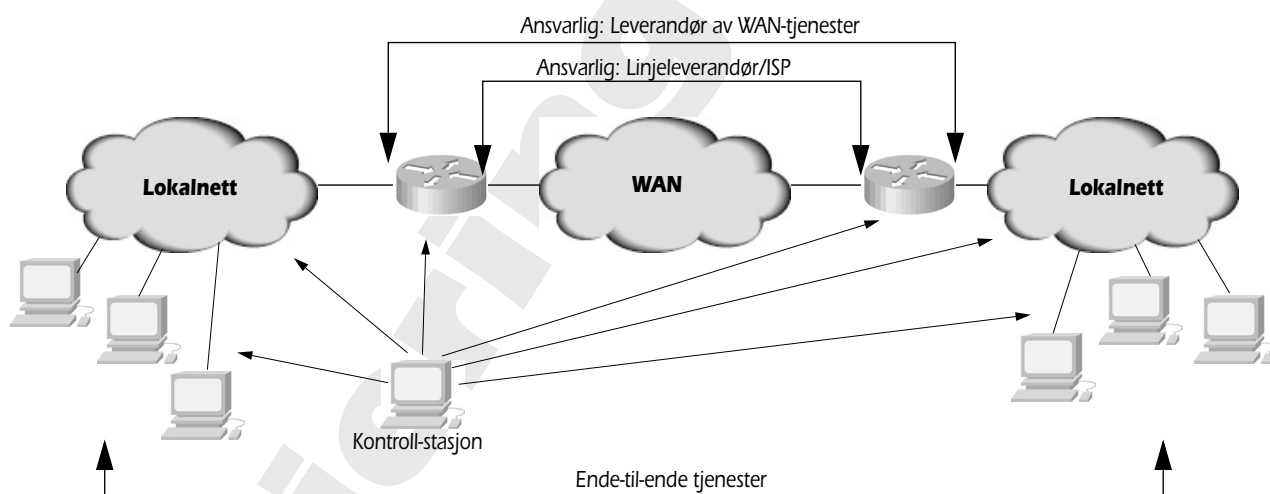
- ✓ Pålitelighet: Feilfrekvenser (MTBF), gjenopprettingstider (MTTR), pakketap
- ✓ Kapasitet: Transaksjoner per tidsenhet (TPS), pakker per sekund (PPS), bits per sekund, størrelse (MB, GB) og så videre

- ✓ Forsinkelser: Ende-til-ende, rundetider, behandlingstid, variasjon (effekten av 'andre' forhold).

Her er det en alminnelig fallgruve å overfokusere på målbarhet på bekostning av hva som er relevant: Langt fra alt som er målbart, har noen innflytelse på saken eller prosjektet. Topphastigheten på en bil er grei å vite, men sjelden relevant for hvorvidt den egner seg for et gitt formål.

En lang rekke verktøy for måling av ulike nettverks-relaterte parametre finnes – frie og kommersielle. Velkjente eksempler av det enkle og effektive slaget er *ping*, *traceroute*, *TCPdump* og *pathchar*. Mens avanserte kommersielle verktøy med glimrende rapportgeneratorer både ser attraktive ut og gir mye relevant informasjon, er det ofte de enkle verktøyene som gir best resultat. For mye informasjon kan forstyrre både fokusering og evne til å se det opplagte.

I tillegg til å finne og etablere målestokker og parametre, er det naturligvis kritisk hvor vi måler. Temperaturen i Bergen er til minimal nytte dersom vi ønsker å vite noe om kjøreforholdene i Bodø. Likeledes er det sjelden nyttig å foreta målinger utenfor eget ansvarsområde. Har vi for eksempel en nettverksoperatør som er ansvarlig for fjernnettverket, er målinger på hans domene uinteressante, mens målinger på egen side forteller i hvilken grad leverandøren oppfyller sine forpliktelser. Figur 3 identifiserer noen typiske målepunkter med tilhørende karakteristika.



Figur 3 Valg av målepunkter er kritisk både for hva vi måler og relevansen av dataene. Å måle en parameter vi ikke har ansvaret for er som regel bortkastet tid.

Kartlegging av bruksmønster

Et nettverk uten brukere representerer i beste fall kunstige omgivelser, og er i verste fall verdiløst for en nettverksanalyse. Samtidig er brukerne og den belastning de forårsaker i høyeste grad variable. Hensikten med kartlegging av bruksmønster er å slå fast om forståelse av brukerne og deres verktøy kan gi oss grunnlag for pålitelige estimater av nettverksbelastning og tilhørende ytelse.

Analysen kan gi oss direkte anvendbare resultater eller grunnlag for en simuleringsmodell som i sin tur gir anvendbare resultater. Elementene som skal inn i modellen, er brukere, applikasjoner og nettverk. Mulige modeller dekker hele spekteret fra enkle tilnærmelser til komplekse og tidkrevende simuleringer med avanserte verktøy.

I tillegg til å gi et bilde av hvordan situasjonen er i dag og hvilken effekt planlagte forandringer vil få, gir avanserte simuleringer også mulighet for teoretiske eksperimenter – omtrent på samme måte som vi eksperimenterer med tall i et regneark: Vi kan studere hvilke endringer som gir hvilke konsekvenser. Mens dette krever betydelig innsats både i tid, datainnsamling og verktøyanskaffelser, kan virtuell eksperimentering gi formidable gevinster – spesielt for store og komplekse nettverk: Plassering av utstyr, test av ulike utstyrskonfigurasjoner, pålitelighets- og robusthets-vurderinger – og ikke minst 'stress-testing' for å se hvordan infrastrukturen oppfører seg under ekstreme belastninger.

Uansett hvilket nivå vi legger kartleggingen på, vil det ikke være mulig å dekke verken alle brukere eller alle applikasjoner. Vi må foreta et utvalg, og prioriterer i den forbindelse alltid brukere og applikasjoner som er mest ressurskrevende. Det er disse som til slutt styrer dimensjonering og plassering av kritiske elementer i infrastrukturen.

Uansett modell og verktøy trenger vi de samme grunnlagsdata for kartleggingen:

- ✓ For brukere: Når de arbeider, varighet, hyppighet, hvilke applikasjoner som brukes og hvordan de brukes.
- ✓ For tradisjonelle applikasjoner: Karakteristika som er relevante for infrastrukturen – datamengder, hyppighet og varighet av dataoverføringer, toleransegrenser for forsinkelser og svartider.
- ✓ For moderne applikasjoner – som er utviklet for nettverksbaserte miljøer, og som derfor gjerne på egen hånd samler statistikker om bruk: Datamengder, overføringstider, tidspunkter, hyppigheter etc.

Utskriftstjenere passer som et enkelt eksempel: Gjennom kontinuerlig overvåking av utskriftskøen på alle tjenerne i en gitt periode, registreres jobbstørrelser, tidspunkter, hvilken tjener og hvilken printer. Resultatene gir blant annet informasjon om maksimal jobbstørrelse, typisk jobbstørrelse, samt når og hvor belastningstoppe kommer. Dette er nok informasjon til å kunne trekke konklusjoner med hensyn til optimal plassering, printer-typer,⁵ kapasitetsbehov og belastning – informasjon som overføres direkte til arkitektur og deretter design.

5 Ulike typer skrivere kan gi dramatisk forskjellig nettverks- og tjener-belastning for samme jobb. Likeledes kan det spille avgjørende rolle hvilke innstillinger brukeren har valgt når jobben genereres og hvilken driver som er installert. Analysen gir indikasjoner om hvor det er nyttig å foreta nærmere undersøkelser med tanke på optimaliseringer.

Etablering av grenseverdier

Hvor går grensene – for nødvendig kapasitet, båndbredde, ytelse, tilgjengelighet og så videre? Analysen ovenfor og behovsanalysen i forrige artikkel gir oss grunnlagsmaterialet, men må konkretiseres og kvantifiseres. Hvilke nettverksforbindelser må ha gigabit-kapasitet? Hvilke systemer og nettverkssegmenter må ha tilgjengelighet på over 99,9% (maks. 10 minutter nedetid per uke)?

Uansett datagrunnlag blir dette ingen eksakt øvelse. Analysen gir en mer eller mindre romslig pekepinn om hvor vi må havne. Deretter må en strategisk beslutning foretas med hensyn til hvorvidt vi skal satse på romslighet eller nødvendighet. En annen vinkling på spørsmålet er hvorvidt tid eller penger er vår knappeste ressurs: Er det tid, velger vi romslighet (for eksempel god ytelsesmessig takhøyde i systemer og nettverk), i motsatt fall er nøysomhet nødvendig – så tar vi de ytelsesmessige utfordringene når de måtte dukke opp.

Grenseverdiene – eller kravene – havner naturlig i to kategorier:

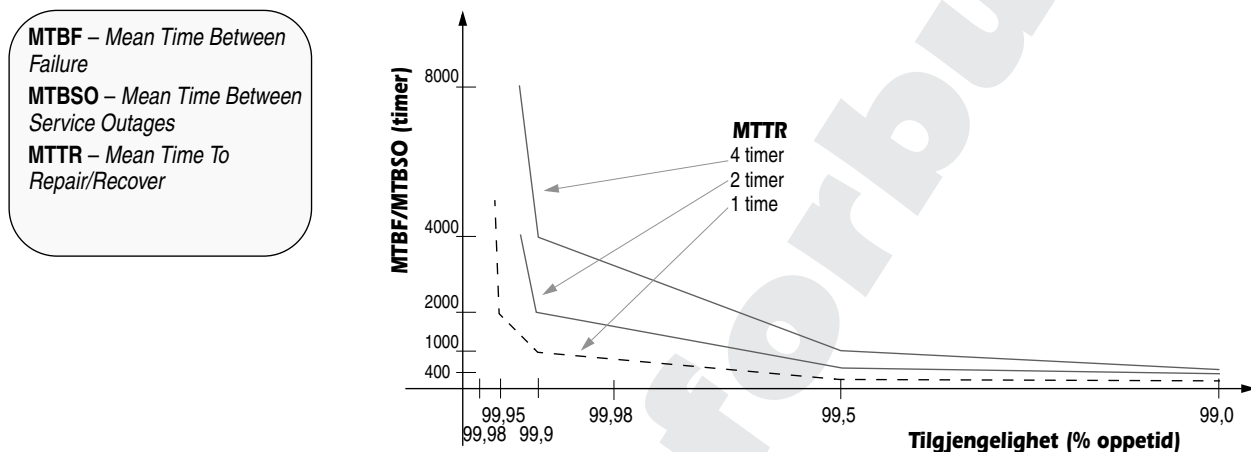
- ✓ Spesifikke krav – som er spesielle for det aktuelle miljøet og styrt av applikasjoner og verktøy, bruksmønster, utstyrvalg og andre lokale forhold.
- ✓ Generelle krav – typisk tommelfinger-regler som benyttes når miljøet ikke gir noen pekepinn om hvor det er riktig eller nødvendig å plassere seg.

Som regel vil det være hensiktsmessig med hjelpemidler for å komme frem til pålitelige resultater med rimelig innsats. Slike hjelpemidler kan spenne fra enkle formler og grafer til omfattende database-baserte verktøy. Igjen kommer vi langt med sunn fornuft og enkle hjelpemidler, og vi skal ta frem noen eksempler som demonstrerer tankegangen.

Tilgjengelighet: 'Oppetid' er et annet uttrykk som brukes i denne sammenhengen. Vi må avgjøre både hva som er nødvendig og hvor/hvordan den skal måles. Mens prosent er den normale målestokken, er det først etter en konvertering til minutter og sekunder at vi får et forhold til prosentenes innhold – for eksempel: 99,5% tillater opp til 50 minutter nedetid per uke, 99,9% = 10 minutter, 99,95% = 5 minutter, 99,98% = 2 minutter. Det er minuttene, ikke prosentene som teller i praksis, og referansen skal alltid være 24 timer per døgn, 7 dager i uken. En annen referanse kan være akseptabel for spesielle anvendelser, tiden er forlenget ute for 9-til-16 datasystemer.

Den andre utfordringen er å velge målepunkter: Hva er relevant og hva er rimelig? For brukeren spiller det liten rolle hvor problemet er: At hans eller hennes tastatur ikke fungerer er like ille som at tjeneren eller nettverket er nede. På den andre siden vil en tjenesteleverandør neppe akseptere ansvaret for en kaffekopp som havnet i et tastatur. Videre er det stor forskjell på utilgjengelighet som rammer én bruker og generell utilgjengelighet som rammer alle eller mange. Slike betraktninger må legges til grunn når målepunkter/referansepunkter velges.

Pålitelighet har mye med tilgjengelighet å gjøre, men er ikke det samme. Pålitelighet sier noe om feilhyppighet på den ene siden og hvor lang tid det tar å rette feil når de først oppstår. Sammenhengen mellom tilgjengelighet og pålitelighet kan fremstilles grafisk, og blir et hjelpemiddel for å komme frem til riktige/rimelige grenseverdier på begge områder. Figur 4 viser et eksempel.



Figur 4 Sammenhengen mellom tilgjengelighet, feilrate og restitusjonstid kan fremstilles slik, og blir på den måten et nyttig verktøy i stedet for en samling tall.

Pakketap/feilrater kan høres ut som et avdanket begrep, men er slett ikke det. Både på fjernforbindelser (leide linjer), Internett-forbindelser og i lokalnettet forekommer fortsatt pakketap – som følge av overbelastning (overforbruk), tekniske svakheter eller programfeil. Pakketap fører til økt responstid – og til nedetid når tapet kommer over en viss grense. Sammenhengene må måles i realistiske omgivelser fordi de blant annet avhenger av teknologien som benyttes og den aktuelle båndbredden.

La oss for eksempel anta at 2% pakketap er en kritisk grenseverdi for hvorvidt kommunikasjonen ansees å være oppe eller nede. Dersom kravet til tilgjengelighet er 99,95%, betyr det at pakketapet kan overstige 2% i maksimalt 22 minutter i løpet av en måned. Er tilgjengelighetskravet 99,98% blir tallet 8,75 minutter – eller 2 minutter per uke.

Å velge riktig nivå: Spørsmålet som stadig dukker opp, er hvilket nivå som er riktig i vår spesielle situasjon – eller for spesifikke anvendelser og brukergrupper. Vi diskuterte forskjellen mellom 'høy ytelse' og 'nødvendig ytelse' ovenfor. Den valgte strategi styrer veivalgene når vi kommer til konkrete situasjoner:

- ✓ Finn retningsgivende verdier for tilgjengelighet, restitusjonstid og feilrater for de viktigste applikasjonene.
- ✓ Finn grenseverdier – akseptable nivåer – for et moderat scenario (lav ytelse) og et høykvalitets-scenario (høy ytelse).

- ✓ Bruk disse data som utgangspunkt for estimater av pålitelighet via ulike veier eller alternativer (utstyr, linjer, leverandører, kapasiteter) i infrastrukturen.

Hensikten med å definere flere kvalitetsnivåer er, som vi også har vært inne på tidligere, å etablere valgmuligheter: Ikke alle applikasjoner – eller organisasjoner – har behov for eller råd til det nivået 'høy ytelse' representerer. På den andre siden kan enkelte anvendelser forutsette et høyt ytelsesnivå for å fungere. For tilgjengelighet kan nivåene for eksempel være 99,0 - 99,5% for testing, 99,5 - 99,9% for et lavt ytelsesnivå og 99,99 - 99,999% for et høyt ytelsesnivå.

Andre områder: Forsinkelser i nettverket er en problemstilling med tallrike sider og ditto konsekvenser. Her spiller brukerne en aktiv rolle i den ene enden, mens fysikkens lover stopper oss i den andre – et forhold som typisk først dukker opp ved kommunikasjon over lange distanser: Det tar endelig tid å flytte data fra Lindesnes til Nordkapp og tilbake – også via fiberkabel. Mellom ytterpunktene finner vi at praktisk talt alle tenkelige elementer har innflytelse på forsinkelsen: Nettverksutstyr, protokoller, applikasjoner, tjenere og fysisk nettverk – for å nevne de viktigste. Hvilke av disse som er betydelige nok til å påvirke arkitekturen, kommer an på forholdene og må evalueres.

Generelle ytelsesmessige forhold er langt enklere å finne grenseverdier for. Som regel er det hensiktsmessig å ta for seg én etter én av de viktigste applikasjonene – eventuelt gruppere dem: Web-tjenester, regnskap/lønn, database, telekonferanser/video, distribuerte applikasjoner. De kan deretter plasseres i en graf, med ytelseskrav langs den ene akse og relativ prioritering langs den andre. En slik fremstilling gir god oversikt og enklere valg/prioriteringer.

Neste utgave

I neste artikkel avslutter vi analyse-prosessen med en diskusjon omkring kravspesifikasjonen (se figur 2). Deretter tar vi fatt på et forholdsvis nytt område innen nettverksanalyse: Flyt-analyse, som i løpet av noen korte år er blitt en disiplin i seg selv. Vi skal presentere hva flyt-analyse er, hvilken nytteverdi slike analyser har og hvordan de kan gjennomføres. ■