

Analyse, arkitektur og design av nettverk

Dette er 8. 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.

Vi er i full gang med nettverksarkitekturen, og introduserte i forrige artikkel komponenter, metodikk og modeller: Elementer og verktøy som må beherskes for å komme frem til en optimal arkitektur med rimelig innsats. I denne artikkelen ser vi nærmere på noen av komponentene som allerede er introdusert, og diskuterer deres rolle i den store sammenhengen.

IP-tjenester

Det er ikke nødvendigvis innlysende hva uttrykket IP-tjenester betyr. Hvilket nivå snakker vi om? Hva slags tjenester? Det er nærliggende å assosiere på høynivå-tjenester som epost, web eller telefoni, men betegnelsen skal oppfattes mer bokstavelig: Tjenester på IP-nivå, dvs. på transport-nivå i nettverket.

Den viktigste årsaken til at betegnelsen mangler korrekt assosiasjon hos de fleste, er at vi ikke er vant med å tenke på slike tjenester. Våre IP-baserte lokalnett har hittil stort sett vært fri for 'komplikasjoner' på transportnivå. Denne enkelheten har vært en viktig suksessfaktor for både IP og Ethernet, men virkeligheten forandrer seg. Gårsdagens suksessfaktor er i ferd med å bli dagens problem eller hindring.

IP-tjenester er funksjoner og egenskaper som setter oss i stand til å kontrollere og styre nettverksressurser, slik at vi kan levere den ytelse og pålitelighet som kreves av brukere, applikasjoner og tilknyttet utstyr.

IP-tjenester handler med andre ord om styring og kontroll av nettverksressurser – for å ...

- ✓ ... forbedre den generelle ytelsen i nettverket (målt i ytelse og responstider sett fra brukernes side).
- ✓ ... tilgodese spesifikke behov fra spesielle brukere og anvendelser, eksisterende eller planlagte.
- ✓ ... kontrollere bruken av ressurser – for avregning/internfakturering, statistikk, planlegging eller andre formål.

I praksis betyr det én eller flere av følgende mekanismer:

- ✓ Aksesskontroll – på to nivåer: Tradisjonell brukeraksess som tidligere har vært kontrollert av individuelle vertsmaskiner, må over på nettverksnivå både for å kunne styre ressurser og for å kunne realisere *single sign-on*. Det andre nivået er knyttet til bruk av ressurser, spesielt båndbredde. Når kvalitetsgarantier skal gis, er det nødvendig å kunne stoppe etterspørselen når ressursene er konsumert – tilsvarende opptattsignal når alle linjer er i bruk.
- ✓ Trafikkstyring: Mulighet for å kunne justere grunnleggende nettverksparametre løpende – i henhold til etterspørsel og be-

lastning. De fleste organisasjoner har belastningstopper som hittil har vært håndtert gjennom overallokering av ressurser (takhøyde), en metode som ikke lenger er pålitelig fordi belastningsbildet og økonomiske krav har forandret seg.

- ✓ Kontroll over utvalgte deler av nettverket som skal ivareta spesielt krevende eller på andre måter uvanlige behov (for eksempel trådløse segmenter eller distribusjonsnettverk for video).
- ✓ Introduksjon av 'tilbakekoblingssløyfer' fra brukere, applikasjoner, utstyr og ledere som muliggjør automatisk regulering av kritiske parametre i takt med behovene.

Mens både hensikt, målsettinger og mekanismer ved første øyekast kan virke både innlysende og nødvendige, er det viktig med en 'reality check' – en behovsprøve – før vi setter i gang. For det første er det et faktum at innføring av IP-tjenester ofte blir en kostbar affære. Derfor kontrollerer vi både to og tre ganger at de virkelig trengs. For det andre må vi ha et klart forhold til hva vi ønsker å oppnå: Finnes problemet vi forsøker å løse, er det tilstrekkelig stort/viktig, står forventede resultater i forhold til hva som må investeres? Og sist, men ikke minst: Er IP-tjenester hele eller deler av løsningen på problemet? I siste tilfelle, hvilke andre elementer må på plass? Finnes de, når kommer de, hva koster de og så videre?

Med disse spørsmålene avklart, har vi grunnlag for å lage det vi kan kalle en 'delarkitektur' for IP-tjenester.

Sikkerhet

Sikkerhet er en tilstrekkelig åpenbar utfordring til å påkalle oppmerksomhet på egen hånd. En sikkerhetsarkitektur kan utarbeides og implementeres for seg selv – hvilket har vært regelen snarere enn unntaket – fordi sikkerheten ble glemt da den eksisterende nettverksarkitekturen ble laget. Siden en sikkerhetsarkitektur nødvendigvis må få konsekvenser for nettverksarkitekturen og vise versa, er det imidlertid optimalt å utvikle dem sammen – gjøre dem til naturlige deler av den samme referansearkitekturen.

En alminnelig definisjon av nettverkssikkerhet kan i kortversjon formuleres slik: *Beskyttelse av nettverk, tjenester og datastrømmer mot uautorisert endring, ødeleggelse og innsyn. Forsikring om at nettverket utfører sine kritiske oppgaver på en korrekt måte, uten skadelige bivirkninger.*

Mens den første delen av definisjonen er innlysende, er den andre mindre opplagt. Hva har dette med sikkerhet å gjøre? Poenget er for det første at sikkerhet har to sider: Datasikkerhet og driftssikkerhet. Dessuten er det like viktig å vite – med tilstrekkelig stor pålitelighet – at det vi gjør ikke har kjente eller potensielle følgeskader. Dette forholdet har i større grad med beskyttelse av personopplysninger og andre kritiske data enn med fysiske skader (for eksempel stråling, belastningsskader) å gjøre.

En sikkerhetsarkitektur omfatter eller tar hensyn til følgende elementer:

- ✓ En trussel-analyse
- ✓ Regler og rutiner (*policy*)
- ✓ Fysisk sikkerhet
- ✓ Bevissthet hos den enkelte medarbeider
- ✓ Applikasjons- og protokoll-sikkerhet
- ✓ Autentisering, katalogtjenester
- ✓ Kryptering (sikring av data under transport og lagring)
- ✓ Sikring av fjernaksess
- ✓ Ytre forsvarsverk (kringvern, *perimeter security*)

Mens alle disse elementene er velkjente for Mellvik-Rapportens lesere gjennom tallrike serier og enkeltartikler, er det også et faktum at målet beveger seg. Statisk sikkerhet finnes ikke, og tilpasninger til virkeligheten må skje løpende. Denne dynamikken inkorporeres i en sikkerhetsarkitektur, som per definisjon skal være rettleddende og overordnet slik at den stimulerer, ikke hindrer kontinuerlige tilpasninger til hverdagen.

Et godt eksempel på denne dynamikken er VPN-løsninger, der mer eller mindre proprietære, IPSec-baserte produkter dominerte markedet for et år siden, mens SSL nå er i ferd med å overta. Et annet eksempel er fokuseringen på virus og SPAM, som vi diskuterer i en annen artikkel i denne utgaven.

Tema sikkerhetsarkitektur er tilstrekkelig viktig og omfattende til at vi har avsatt en hel artikkel til en grundigere gjennomgang. Artikkelen kommer til høsten.

Styring og drift

Til tross for at automatisering av driftsfunksjoner har stått øverst på agendaen i utallige sammenhenger i et tosifret antall år, er drift fortsatt en stor utfordring – som må håndteres av mennesker. Det betyr ikke at utviklingen har stått stille, men at den kun såvidt har holdt tritt med tilveksten av nye oppgaver.

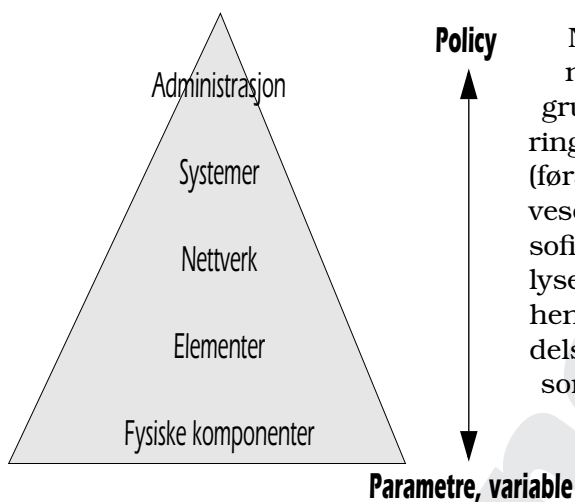
En driftsarkitektur omfatter typisk noen eller alle av følgende punkter (se figur 1):

- ✓ Drift av nettverksutstyr (noder, fysisk utstyr)
- ✓ Drift av 'nettverks-elementer' – samlinger av fysiske enheter som behandles under ett, for eksempel brannmurer eller nettverkssegmenter
- ✓ (Overordnet) drift av tjenester
- ✓ (Overordnet) drift av nettverket
- ✓ Administrasjon (ikke-teknisk drift – planlegging, budsjettering, avtaler, internfakturerer etc.)

For store og sammensatte nettverk er en optimal driftsarkitektur en grunnleggende forutsetning for at det resulterende nettverket skal bli tilfredstillende. Prosessen som leder frem til denne subarkitekturen, er tilsvarende komplisert og krevende. Mindre nettverk – med opp til noen få tusen noder, er en langt enklere utfordring der mange av valgene er opplagte.

På våre kanter er denne størrelsesorden regelen snarere enn unntaket, hvilket er positivt i den forstand at det er enklere å komme til målet. Medaljens bakside er at amatørerne ikke like lett lar seg avsløre. Tusenvis av nettverk mangler ikke bare arkitektur, men også sikkerhet og styring. Problemene blir imidlertid kun unntaksvis oppdaget – typisk i forbindelse med at det skjer en katastrofe. At enorme kostnader kunne ha vært spart gjennom en profesjonalisering av slike nettverk, er et faktum – og en fjern drøm.

SNMP – Simple Network Management Protocol
MIB – Management Information Base



Figur 1 Nettverksdrift har 5 nivåer som må behandles i en driftsarkitektur.

Nettverksdrift kan deles i to grunnleggende funksjoner: Drift av fysiske komponenter, enkeltvis eller i grupper, og innsamling/overføring/behandling av styringsinformasjon. Utviklingen av standardprotokoller (først og fremst SNMP) og dataformater (MIB) har bidratt vesentlig til å forenkle utfordringene, sammen med sofistikerte verktøy for innsamling, rapportering og analyse av styringsdata. Ikke desto mindre er det mange hensyn å ta i forbindelse med planlegging og utarbeidelse av en arkitektur – inklusive etablering av hvem som har ansvaret for hva (for eksempel konfigurasjonsstyring, utstyrsregister og kapasitetsplanlegging). Videre skal instrumenteringen planlegges og dens ytelsesmessige konsekvenser analyseres (for eksempel 'hvor stor del av tilgjengelig båndbredde konsumeres av styringsinformasjon?').

Driftsarkitekturen har tette koblinger til andre hovedelementer i vår referansearkitektur, for eksempel:

- ✓ IP-tjenester: Styringsarkitekturen må ta hensyn til og omfatte prioriterings-mekanismer (QoS), tjenesteavtaler (SLA) og tilknyttede *policies*.
- ✓ Dynamisk ruting og adressering (se nedenfor) kan kun realiseres gjennom et effektivt styringssystem.
- ✓ Styringssystemet må for det første kunne ta hånd om grunnleggende sikringsfunksjoner, og må dessuten kunne beskyttes i seg selv. Altfor ofte glemmes styringssystemet og -mekanismene når sikkerheten evalueres, og blir risikomomenter i seg selv.

Vår avhengighet av nettverket tilsier at disse problemstillingene tas alvorlig, selv i relativt små nettverk. Det har aldri vært mer kostbart å være fattig, og å spare penger på de gale stedene blir lett katastrofalt. En velfundert driftsarkitektur gir et solid fundament å vokse på.

Adressering og ruting

Som vi konstaterte i introduksjonen av arkitektur-komponentene i forrige utgave, er adressering og ruting en triviell sak i små og enkle nettverk. Vi introduserte samtidig de ulike adresseringsstrukturene – subnetting, supernetting og NAT – som har sørget for å forlenge levetiden til dagens IP-protokoll med ti år allerede. De er kompliserende og nødvendige, og må tas hensyn til i enhver nettverksarkitektur.

Videre har valgene vi gjør i forbindelse med adressering og ruting store konsekvenser for sikkerhetsarkitekturen. Et nettverk over minimal størrelse deles naturlig inn i segmenter både for å bedre mulighetene for trafikkstyring og av sikkerhetsmessige årsaker. Likeledes har valg av NAT (private adresser) kontra offisielle adresser store konsekvenser for både sikkerhet og trafikkflyt. Alle valg har positive og negative sider som må evalueres i forhold til hverandre, og som vil gi ulike konklusjoner avhengig av omstendigheter og prioriteringer.

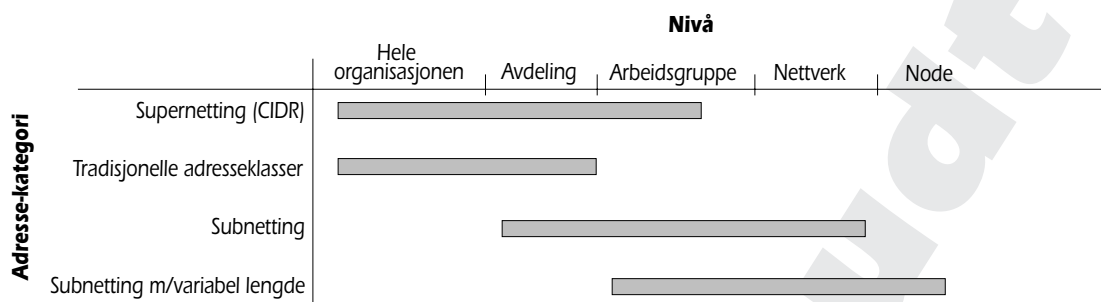
Her er kompleksitets-betraktninger viktige: Valg som i teknisk forstand er nærmest opplagte, kan vise seg å være bortimot umulige fordi de introduserer for mye kompleksitet i systemet. Slik kompleksitet er en større trussel mot driftssikkerheten enn noen enkeltstående boks, forbindelse eller løsning. For eksempel kan en stor organisasjon som eier ('har fått tildelt') 3 eller flere disjunkte offisielle IP-adresse-segmenter, i mange tilfeller oppnå vesentlig forenklingsgevinst ved å gå over til private adresser.

Igjen står vi overfor et glimrende eksempel på at gårsdagens sannheter er dagens løgner. Av en rekke tekniske årsaker var det så sent som for et års tid siden uaktuelt å vurdere en slik konvertering. NAT ble betraktet som hemmende, og offisielle adresser et *must*, blant annet på grunn av komplikasjoner i tilknytning til IPSec og VPN. Disse komplikasjonene er nå for historie å regne.

Et annet forhold som påvirker bildet er den økende bruken av Internettet som WAN. Tradisjonelle leide linjer erstattes i voksende grad av logiske tunneler via Internettet – med eller uten kryptering. Resultatet er vesentlig lavere kostnader på bekostning av større kompleksitet på lavt nivå i nettverket – spesielt i forbindelse med ruting og adressering. Denne kompleksitetsøkningen må holdes under kontroll, blant annet gjennom optimale produkt- og teknologivalg for de logiske tunnelene.

Grafen nedenfor gir en pekepinn med hensyn til hvilke adresseringsmekanismer som egner seg på ulike nivåer eller størrelsesordener. Med hensyn til rutingstrategi skal vi ta følgende forhold i betraktning:

- ✓ Kompleksitet: Hold behovet for hyppige oppdateringer av rutingtabeller på et minimum. At disse som regel er automatiske, er av underordnet betydning i denne sammenheng. Dette er en balansegang: Rask konvergens (se nedenfor) krever hyppige oppdateringer, som i sin tur krever båndbredde og ruter-ressurser, spesielt dersom tabellene er store.
- ✓ Konvergeringstid: Tiden det tar for endringer å forplante seg til alle rutere i nettverket er en kritisk parameter, og skal



Figur 2 Hvor er det hensiktsmessig med de ulike adresseringstypene vi har til disposisjon? Grafen gir en indikasjon.

være så lav som mulig, samtidig med at den må balanseres mot oppdateringshyppigheten (administrativ trafikk mellom ruterne).

OSPF – Open Shortest Path First
RIP – Routing Information Protocol

- ✓ Belastningsbilde: Store rutingtabeller, supernetting og subnetting med variabel lengde er krevende for ruterne. Har de nok ressurser?
- ✓ Stabilitet: En samling rutere er en 'prosess' i regulerings teknisk forstand, med tilbakekoblingsløyfer og dempeledd. Et slikt system kan oscillere – komme i selvsving – en tilstand som for enhver pris må unngås (den setter hele nettverket ut av funksjon).

Som vi var inne på ovenfor, kommer de fleste av disse problemstillingene til syne først når nettverket blir av betydelig størrelse – med et to eller tre-sifret antall rutere, tilsvarende mange subnett og en rekke forbindelser til omverden. Riktige valg styres med andre ord i stor grad av nettverkets omfang og kompleksitet. For eksempel: At OSPF er ansett for å være den beste ruting-protokollen betyr ikke at den er riktig for enhver sammenheng. Har vi et forholdsvis enkelt nettverk med 5 eller 10 rutere og oversiktlige forhold, kan RIP2 være et riktig valg fordi den er enklere – og tilstrekkelig til å gjøre jobben. Er forholdene enda enklere, kan statiske ruter være optimalt. Flere rutingprotokoller kan være nødvendig, men er aldri ønskelig.

Referansearkitektur

Gjennomgangen av disse 'underarkitekturene' demonstrerer først og fremst metodikken vi bruker for å komme frem til en referansearkitektur. Andre elementer som skal inngå, behandles etter samme mal (se figur 13 i forrige utgave).

Neste utgave

Vi er til veis ende med arkitekturen, og tar i august-utgaven fatt på innspurten – med nettverksdesign. ■