

Content Delivery Networks

Levering av innhold? Har ikke det vært hensikten med nettverk siden tidenes morgen? Hva kan være meningen med – og årsaken til – at vi plutselig har begynt å snakke om egne nettverk for levering av innhold? Har smarte hoder eller leverandører klart å etablere et nytt begrep med nye kunstige behov som følge?

På mange måter er CONTENT DELIVERY NETWORKS (CDN) et mislykket navn. Vi har alle et forhold til begrepet 'innhold' – som sjelden sammenfaller med CDN-betydningen. For eksempel tilsier all logikk at hver gang vi sender en epost-melding – med eller uten vedlegg, transporterer nettverket 'innhold' av (mer eller mindre) intelligent karakter. Denne forvirringen må vi leve med. 'Innhold' er atter et begrep som står foran en redefinisjon – eller i alle fall en utvidelse av sin gamle betydning.

Den nye betydningen er knyttet til nettleseren som rammeverk og presentasjonsmekanisme: I utgangspunktet er alt som kan presenteres på en Web-side for 'innhold' å regne. Men kan dette innholdet være så viktig eller vanskelig at vi trenger egne nettverk å levere det med? Eller betyr CDN noe helt annet?

Total begrepsforvirring

Begrepsforvirringen stopper ikke der: Beslektede begreper strømmer på og gir leverandørene nytutklekkede historier å fortelle – med høyst varierende kobling til virkeligheten: *Content Management, Content Routing, Intelligent Routing, Layer 7 Switching, Content Switching, Traffic Provisioning, Content Distribution, Content Caching* – og så videre. Det er høyst betimelig å stille spørsmålet *where's the beef?* Hva skal vi med alt dette og hvordan bidrar produktene og teknologiene til at vi blir mer effektive, tjener mer, hever kvaliteten?

Forvirringen er i realiteten helt unødvendig, og er først og fremst forårsaket av leverandørers og eksperters behov for å virke ekstra kreative. Bak sløret av nye uttrykk og flotte figurer finner vi kun unntaksvis noe annet enn gamle løsninger benyttet på nye måter eller i nye sammenhenger. Det betyr ikke at CDN og tilhørende herligheter – eller andre lignende eksempler – er uviktige eller verdiløse, men at tåken som legges i mange tilfeller overskygger de relevante poengene.

CDN – et virtuelt nettverk

Det første og viktigste poenget i tilknytning til CDN er at nettverket er virtuelt – et *overlay network* på fagspråket: Det er kun unntaksvis snakk om en egen fysisk infrastruktur for innholdslevering, men om å anvende deler av eksisterende infrastruktur til et spesifikt formål. Opprinnelsen til problematikken som har brakt temaet på bane, er den eksplosive veksten i bruk av nettleser som rammeverk og verktøy for alle slags IT-oppgaver.

I bunn og grunn handler det om logistikk: Innholdet er varen, brukerne er kunden, og oppgaven er å levere varen i henhold til kundens forventninger på raskeste og mest kosteffektive måte. Det betyr i sin tur å få kontroll over varetransporten, og her kommer CDN-begrepet naturlig inn.

Fra bleier til voksen effektivitet

Det begynte med enkle Web-tjenere som leverte HTML-sider og prosesserte forespørsler fra brukerne. Etersom innholdet fra én side til den neste ofte besto av de samme elementene, og fordi mange brukere på samme sted hyppig forespurte de samme sidene, dukket det tidlig opp såkalte *Web-cacher* og *proxy*-tjenere. Optimaliseringen var åpenbar både i teori og praksis, og kan sammenlignes med budbil kontra container-transport: Å sende enkeltksemplarer av Se og Hør eller Kapital med hvert sitt bud til abonnentene ville ikke ha latt seg gjennomføre i praksis. Aviskiosken på hjørnet eller postbudets lokale fordelingskontor fungerer som *proxy* for slike leveranser.

Første punkt i effektiviseringsprosessen er med andre ord å distribuere innholdet til 'leveringssteder' så nær kundene som mulig, samtidig med at vi sørger for å opprettholde produktets ferskvarestatus. Enkle og smarte mekanismer tar vare på slike forhold, og kan tilpasses ulike praktiske behov: Det begynte med en enkel datostempling på innholdet, eller en rask kontroll tilbake til kilden hver gang et gitt objekt ble forespurt.

I dag er utviklingen kommet et kvantesprang videre, og mange løsninger legger til rette for avanserte policy-spesifikasjoner for hvordan kontroll og oppdatering skal gjøres. Her er noen eksempler:

- ✓ En spesifikk regel eller et regelsett kan gjelde for en gitt adresse, for grupper av adresser, for enkeltobjekter, objekttyper eller andre kombinasjoner.
- ✓ En policy kan angi spesifikke oppdateringstidspunkter og/eller forhold som skal tas i betraktning før en oppdatering gjøres – ukedag, tid på døgnet, trafikkbelastning bakover mot kilden, inngående trafikkbelastning og så videre.

Når vi forestiller oss en typisk Web-side med flere titalls grafiske elementer, dekorasjoner og menyer som sjelden eller aldri forandrer seg, er det lett å innse hvilken optimalisering en *cache*-funksjon umiddelbart representerer. Utover å gi raskere levering av 'varene' til 'kunden', bidrar et slikt lokalt distribusjons-lager til å gi:

- ✓ ... større stabilitet i og med at datakilden kan være ute av drift uten at brukerne merker noe som helst,
- ✓ ... dramatisk lavere belastning på kilden,
- ✓ ... og ikke minst redusert belastning på nettverket mellom kunder og kilden.

Mer komplisert trenger det ikke å være. Vi har etablert et grunnleggende CDN.

Fra Web-sider til Web-SERVICES

Beskrivelsen ovenfor avslører at vi naturlig kan dele innholdet som leveres til en nettleser i flere typer eller kategorier. Det som begynte med ren tekst og enkle grafiske elementer, er i dag et sammensurium av ulike objekttyper med ulike krav til ferskhets, båndbredde og lokale ressurser:

- ✓ **Statisk** – logoer, figurer, bilder og *bitmaps* som ikke endrer seg over tid.
- ✓ **Semi-statisk** – tekst og andre objekter som endrer seg sjelden, for eksempel én gang per dag eller uke.
- ✓ **Dynamisk** – innhold som genereres på forespørsel fra brukeren, for eksempel personlig tilpassede sider – med nyheter, børsinformasjon etc.
- ✓ **Lagret multimedia** – sammenhengende datastrømmer med (typisk) video-opptak, redigert og lagret for levering på forespørsel.
- ✓ **Levende multimedia** – direkte overføring av (typisk) radio eller TV/video-programmer (*web-cast*), for eksempel en konsert, et foredrag/forelesning eller levende nyheter.

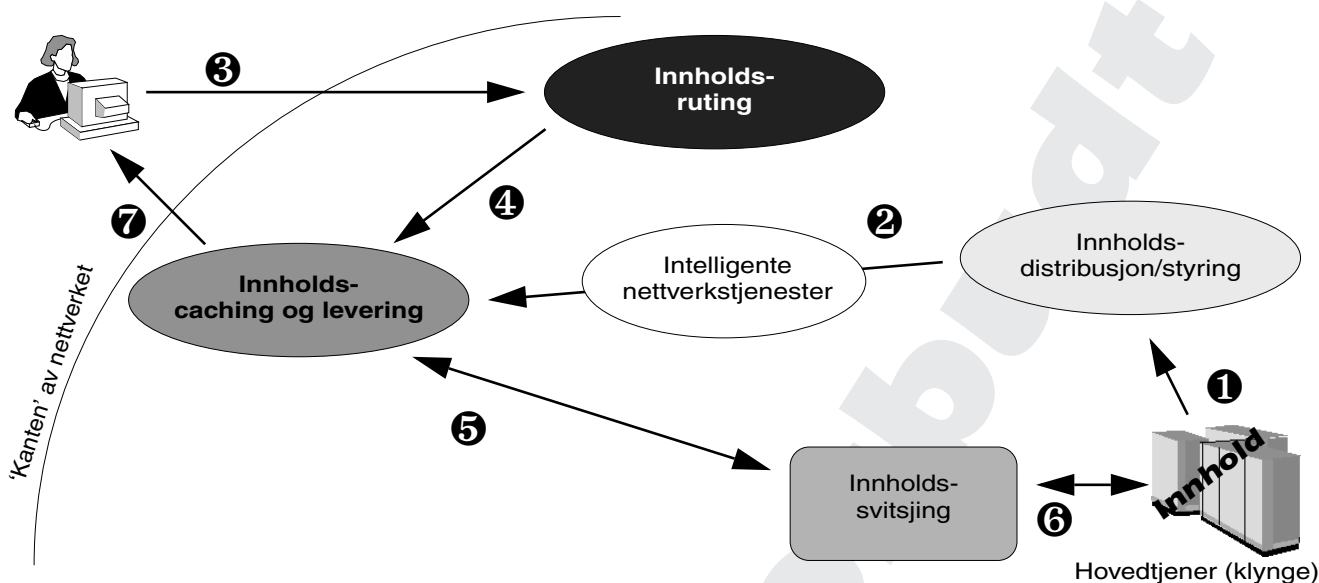
Å tilfredsstille kunden

Å etablere et CDN internt i en stor organisasjon er naturligvis langt enklere enn å gjøre det samme over Internettet, fordi vi har kontroll over vårt eget nettverk. Derfor er det i intranett-sammenheng CDN har satt størst spor etter seg så langt, og det er dette markedet leverandørsiden fokuserer mest på.

Leverandørsiden har virkelig lagt seg i selen de siste to årene: Etter at Web-teknologien passerte fra bleie- og barnestadiet – der tekst, enkel grafikk, *proxy*-tjenere og *caché* dominerte, til puberteten, ble det åpenbart at vi står overfor et enormt marked. Portaler, sanntids datastrømmer, Web-tjenester og så videre, forutsetter at 'vareleveringen' er pålitelig og av forutsigbar, god kvalitet.

En strøm av nye produkter dukket opp, tett fulgt av nye begreper og betegnelser – som vi påpekte i innledningen. Kun unntaksvis var det snakk om ny teknologi per se, kun nye anvendelser av eksisterende løsninger. Det ene er like godt som det andre, så lenge resultatet står i forhold til forventninger og investeringer, og for en gangs skyld står vi overfor et flertall av eksempler som overgår forventningene. Dermed har utviklingen fått ytterligere vann på mølla, og det finnes i dag knapt en eneste seriøs aktør i nettverksmarkedet som ikke også har en løsning for optimalisering av innholdslevering. F5 Networks var en av pionerene på området, mens Cisco – helt i tråd med etablerte tradisjoner – kjøpte opp en liten aktør med solid teknologi, og har på rekordtid posisjonert seg som ledende på enda et område.

Spesielt organisasjoner med utstrakt bruk av nettbaserte opplæringsprogrammer har høstet store gevinster via sofistikerte CDN-systemer. Her ligger også en viktig årsak til Ciscos suksess på området: Selskapet er en av verdens største brukere av nettopp nett-basert opplæring.



Figur 1 Et avansert CONTENT DELIVERY NETWORK er komplisert i teknologisk forstand, men enkelt og pålitelig sett fra brukerens side. Når systemet først er på plass, og gitt at det har fått en fornuftig arkitektur, er det også svært overkommelig i driftsmessig forstand.

og en av de største brukerne av egne løsninger. Et bedre test- og referanse-miljø kan ingen produktutvikler ønske seg.

CDN i praksis

Figur 1 viser en prinsippskisse for hvordan et avansert CDN kan realiseres. Ved å gjennomgå prosessen – fra innholdet blir gjort tilgjengelig til det befinner seg på brukerens skjerm – får vi en klarere forståelse for hvordan det hele fungerer på et overordnet nivå. Samtidig får vi plassert noen av begrepene som ble introdusert i innledningen. Nummereringen refererer til tallene i figuren.

- 1** Innholdet blir plassert på hovedtjeneren i systemet. Denne tjeneren – ofte en klynge av tjenere som arbeider mot felles masselager (SAN) – er aldri synlig utenfra, men betjener innholdsnettverket på eget initiativ eller på forespørsel. Når nytt innhold blir lagt ut, som tilfellet er her, informerer hovedtjeneren styringssystemet om dette.
- 2** Styringssystemet sørger for at innholdet blir distribuert til kanten av nettverket så snart som mulig, i henhold til gjeldende policy og i forhold til innholdets prioritet. I store og/eller høyt belastede nettverk, kan såkalte intelligente nettverkstjenester introduseres for å ta vare på prioritering og ruting av trafikken.
- 3** En bruker spør etter innholdet. Forespørselen går ikke til kilden, men til en 'innholds-ruter' som kan dirigere forespørselen dit det er optimalt.
- 4** Innholds-ruteren er et nøkkelement i CDN-sammenheng: Den holder styr på belastning i ulike deler av nettverket, noder som er ute av drift, hvilke noder som er nær (topologisk) hvilke brukergrupper (adresser) og andre forhold, og velger ut

fra denne kunnskapen et optimalt kontaktpunkt for brukeren.

- 5 Hit kommer vi sjelden: Kant-noden som ble valgt av innholds-ruteren, kan i de fleste tilfeller levere innholdet direkte til brukeren, hvilket avslutter transaksjonen. I noen tilfeller må den foreta en kontroll for å forsikre seg om at innholdet fortsatt er gyldig/oppdatert. Dette styres av policy og/eller av innholdets 'varighets-stempel'.
- 6 I verste fall må innholdet oppdateres. Innholds-svitsjens oppgave er å finne den tjeneren i klyngen som er optimal for oppgaven.
- 7 Innholdet leveres til brukeren.

Mens dette er en omfattende prosess, er det samtidig innlysende hvilke optimaliseringer og andre fordeler den gir. Vi ser at styringssystemet og det regelverk som legges til grunn, er selve nøkkelen i løsningen. I et stort nettverk med mange trafikktyper og høy belastning, ser vi at det kreves både innsikt og eksperimentering for å komme frem til optimale regler for et CDN. Videre er det interessant at det ofte er de enkle forholdene som skaper størst problemer: Vi har lett for å miste skogen av syne på grunn av trærne. For eksempel er det opplagt – og lett å glemme – at regelverket må blokkere for video-strømmer over linjer som ikke har kapasitet til å håndtere slikt innhold.

En annen observasjon vi kan gjøre avslutningsvis, er at *Content Delivery Networks* i stor grad handler om å 'lure' tradisjonelle mekanismer i nettverket: Innholdet blir sjelden levert fra den kilden vi forespør. Innholds-rutere, -svitsjer og intelligente nettverkstjenester 'trikser' med både datapakker, Internett-adresser og DNS-innslag, og sørger for at en forespørsel havner der den hører hjemme.

Kompleksiteten som følger i kjølvannet av slike mekanismer, er vel og bra så lenge den er ute av syne og alt fungerer som det skal. Når feil oppstår og feilsøking er nødvendig, blir situasjonen en annen: Tradisjonelle metoder fungerer dårlig fordi vi ikke lenger kan stole på det vi ser når datastrømmen studeres. Derfor er implementasjon og drift av CDN en oppgave for eksperter. Først når IPv6 overtar som dominerende transportprotokoll, kan vi regne med reell forenkling på dette nivå (se egen artikkel om IPv6 på side 10).

Konklusjon

Et CDN er et virtuelt nettverk, logiske forbindelser som knytter sammen enheter med spesifikke oppgaver. Nettverket kan være privat (internt) eller offentlig (Internett), og hensikten er alltid å optimalisere brukeropplevelse i forhold til tilgjengelige ressurser. Mens første generasjons CDN besto av *caching* og *proxy*-tjenere, er dagens produkter og løsninger avanserte trafikkmanipulatorer som ikke bare bedrer brukeropplevelsen, men som i mange tilfeller er en forutsetning for at innholdet skal kunne leveres. Det er utenkelig at tusenvis eller hundretusenvis av identiske, parallelle datastrømmer skulle kunne passere gjennom de fysiske nettverkene på samme tid. Multicast-pro-

tokollene som er utviklet for å håndtere slike situasjoner, har kun ført oss et stykke på vei.

CDN-teknologi og -produkter har dermed etablert en nisje på egen hånd, som allerede gir gode resultater for avanserte innholds-leverandører, og som uten tvil vil få en voksende praktisk rolle i tiden fremover – godt ute av syne for brukerne, men desto viktigere for de som har ansvaret for leveringen av datastrømmene.

Navnet og sluttproduktet – innhold – har lett for å føre til en overfokusering på nettopp innholdet i stedet for mekanismene: Et CDN er en distribusjonsmekanisme hvis hovedoppgave er å sørge for logistikken – effektiv, optimal utnyttelse av de tilgjengelige ressursene. Det betyr – blant annet – å kunne kanalisere datastrømmer eller forespørsler dit det finnes ledig kapasitet, og aller helst allokere ledig tjenerkapasitet når det trengs. Så sent som for et par år siden, var en slik problemstilling for rene drømmerier å regne: Vi kan ikke skape ressurser som ikke finnes ved å knipse med fingrene. Dersom vi imidlertid har full kontroll over en homogen samling tjener-ressurser, typisk av typen *blades* – som vi diskuterte i forrige utgave av Mellvik-Rapporten, kan også denne delen av utfordringen håndteres, sågar automatisk.

Det er nettopp i slike scenarier de nye *blade*-baserte tjenerarkitekturerne kommer til sin rett. Det finnes en rekke praktiske eksempler på hvordan kontroll og fleksibilitet på et slikt nivå kan gjøre underverker i forbindelse med håndtering av toppbelastninger i ulike deler av systemet. At de samme tjenerressursene kjører månedsavslutning på regnskapet den ene natten, utfører kompliserte datavarehus-operasjoner den neste, mens de om dagen håndterer *call-center*, nett-salg eller opplæring etter behov, har vært IT-avdelingens drøm siden tidenes morgen. I dag er det mulig, sammen med tilsvarende optimalisering av båndbredde og levering av sofistikert innhold – som for den enkelte bruker ser skreddersydd ut for nettopp henne eller han. Enkelte store miljøer i Europa og USA mener å ha redusert tjener-kostnadene med hele 80% etter å ha innført slike løsninger. Selv om besparelsen var bare halvparten så stor, ville den i høyeste grad være interessant for de fleste.

Betjening av brukere via nettleseren har nådd et nytt nivå, og vi skrapper fortsatt i overflaten med hensyn til hva som er mulig. For hvert teknologinivå som forsvinner ut av syne for brukere og utviklere, dukker det opp et nytt spekter av muligheter og tjenester. CDN er ett av trinnene på denne utviklingsstigen. ■