

Analyse, arkitektur og design av nettverk

Dette er 6. 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.

Forrige artikkel gikk i sin helhet med til å presentere flytanalyse – en fersk metode for studie av trafikkmønstre i digitale nettverk, og et allerede anerkjent verktøy i forbindelse med kapasitetsplanlegging og design av nettverk.

Avslutningsvis gikk vi gjennom en metode for identifikasjon av datastrømmer som fokuserte på datakilder og datakonsumenter: Gjennom å finne hvor data oppstår og hvor de blir av, kan vi avdekke viktige datastrømmer i nettverket, deres retning, utbredelse og andre karakteristika.

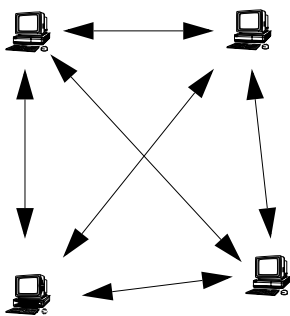
Flytmodeller

Et annet og alternativt hjelpemiddel for identifikasjon og gruppering av datastrømmer, kalles flytmodeller. En flytmodell karakteriserer (beskriver) datastrømmenes hovedretning og gjentakelsesgrad.

Mens tradisjonelle retningslinjer for design av nettverk postulerer at data flyter relativt likt frem og tilbake mellom to parter, er dagens virkelighet i de fleste tilfeller en annen: Strømmen i den ene retningen er som regel langt større enn i den andre. Vi trenger ikke å gå lenger enn til en typisk nettleser-sesjon for å finne et eksempel: Store mengder data strømmer mot brukeren, mens forespørslene som går ut er små, korte og relativt sett sjeldne. Denne asymmetrien er karakteristisk for moderne applikasjoner, et forhold som blant annet avspeiler seg i overføringsteknologier som ADSL, hvor kapasiteten i den ene retningen er flere ganger større enn i den andre.

Med gjentakelsesgrad – som ikke er noen god betegnelse – beskriver vi datastrømmens unikit og hierarkiske egenskaper: En unik datastrøm, for eksempel en epost-utveksling eller en IM-sesjon mellom 2-4 brukere, forekommer én eneste gang. Dersom de samme brukerne i løpet av avgrenset tid aksesserer samme informasjon i en database eller ser på de samme nettsidene, står vi overfor en gjentakende datastrøm. En slik gjentakelse er interessant fordi den på den ene siden representerer bortkastet båndbredde, og på den andre siden lett kan optimaliseres. En *Web-proxy* eller *-cache* er en typisk optimalisering av en slik gjentatt datastrøm – den bidrar til å flytte gjentakelsen nærmere brukerne, og reduserer derigjennom belastningen på nettverket som helhet.

Denne egenskapen er en av de aller viktigste å ta hensyn til i en nettverks-design. Den har stor innflytelse på hvordan vi grupperer og for-deler brukere, tjenere, båndbredde, nettverksutstyr, adresser og så videre, og ikke minst hvordan vi angriper utfordringer knyttet til lokal *caching* og optimalisering.



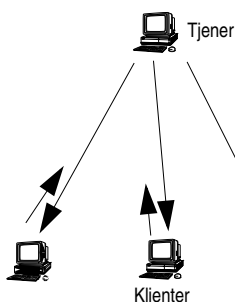
Peer-to-peer – eller alle-til-alle – er en tradisjonsrik modell både i Internettet og andre sammenhenger, men har beskjeden praktisk betydning i typiske nettverk av i dag.

Vi skal kort gjennomgå tre modeller som demonstrerer metoden og tankegangen. Det finnes flere, for eksempel kooperativ databehandling – *cooperative computing*, som først og fremst forekommer i forskningsmiljøer, og derfor ikke er av generell interesse – i dag. For hver modell skal vi identifisere det vi kaller 'kritiske flyt', datastrømmene som har størst betydning for den prosessen vi arbeider med.

Peer-to-peer

Tiden går, og det har fortsatt ikke dukket opp noen god norsk oversettelse for dette uttrykket. 'Alle-til-alle' gir imidlertid et rimelig bilde. Modellen dekker tilfeller der applikasjonene og brukerne har omtrent like kommunikasjonsbehov. Selv om den ikke harmonerer spesielt godt med dagens typiske anvendelser og løsninger, er modellen et naturlig startpunkt for flytanalysen inntil vi eventuelt har skaffet oss mer detaljert kunnskap om hva som egentlig foregår: Dersom vi antar alle-til-alle dataflyt, er sjansen for underdimensjonering små. Videre finnes det moderne applikasjoner som harmonerer med modellen, for eksempel fildelingstjenester à la Napster, og samarbeidsløsninger fra nykommeren InView, som vi nevnte i artikkelen om *Instant Messaging* på side 4.

Modellen gir ingen kritiske flyt og ingen gjentakelsesgrad, og er et minste felles multiplum når annen informasjon mangler.



Klient-tjener er en modell med karakteristisk asymmetrisk dataflyt og er typisk for det vi i dag kaller tradisjonelle applikasjoner.

Klient-tjener

Dette er den mest alminnelige situasjonen og dermed den mest anvendelige modellen. Her finner vi både gjentakelsesgrad og hovedretninger. Dataflyten er utpreget asymmetrisk, fra tjeneren mot klientene. I noen tilfeller er den nærmest for enveis å regne: Trafikken fra klientene er forsvinnende. Dermed er det også innlysende at trafikken fra tjenerne er en kritisk flyt.

Videre forekommer det i en del sammenhenger at flere klienter mottar samme data eller datastrøm samtidig, hvilket trigger det forholdet vi har kalt gjentakelsesgrad, og kvalifiserer evaluering av mekanismer som *multicasting* for å redusere nettverksbelastningen. Video-baserte opplæringsprogrammer er et godt eksempel i den forbindelse.



Copyright © 1997 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited

Kommunikasjon er en kunst, ikke minst i teknologi-sammenheng.

De fleste anvendelser med nettleseren som brukergrensesnitt hører også hjemme i denne modellen – med utpreget gjentakelsesgrad, slik vi nevnte innledningsvis. Her kommer *caching* på flere nivåer til anvendelse, og transaksjonsprotokollen (HTTP) mellom klienter og tjenere er tilpasset dette, med mekanismer for å kontrollere 'ferskhetsgraden' på dataene.

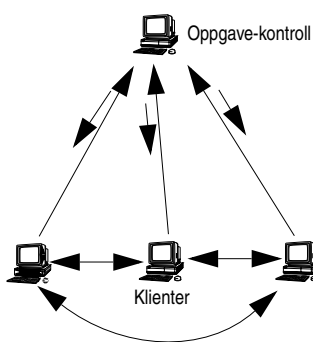
Når slik bruk utvikles videre til omfattende *Content Delivery Networks* (CDNs, se Mellvik-Rapporten nr. 100), der lokale *cache-* og *proxy-*tjenere koordinerer aktivitetene seg imellom i flere nivåer, passer klient-tjener modellen kun i kanten av nettverket (klienter til nærmeste *cache*). De sentrale delene hører hjemme i en flytmodell som kalles kooperativ databehandling, som er langt mer komplisert fordi datastrømmene på tvers blir store og uforutsigbare. Slike scenarier forekommer for sjelden på våre kanter til at det er relevant å diskutere den kooperative modellen her.

Distribuert databehandling

Det hørtes på mange måter ut som et pust fra fortiden, fra slutten av 80-tallet – da PCene skulle løse verdensproblemene sammen ved hjelp av lokalnett. I et slikt historisk perspektiv, og ikke minst i lys av klare sentraliseringstrender som har preget IT-utviklingen de siste årene, er det interessant at distribuert databehandling igjen kommer på mote. Formen og sammenhengen er imidlertid annerledes enn tilfellet var i det vi kan kalle 1. generasjons distribuerte systemer.

Denne modellen er i utgangspunktet mindre forutsigbar enn de to foregående, fordi rollen til den sentrale, koordinerende enheten varierer sterkt mellom ulike anvendelser. Avhengig av oppgavene kan datastrømmene være små eller store og de kan flyte den ene, den andre eller begge veier. Dersom flyten viser seg å passe med klient/tjenermodellen, bruker vi naturligvis denne. I mange tilfeller er den imidlertid helt motsatt av klient/tjener: Liten trafikk ut fra 'senteret', mye trafikk tilbake fra nodene. Videre er det svært ofte slik at nodene, som utfører det egentlige arbeidet, har betydelige datautvekslinger seg imellom. Så mange variable betyr at det skal betydelig innsats til for å for det første kvantifisere datastrømmene, og for det andre å avgjøre hva som er kritisk.

Modellen kommer til anvendelse både i forbindelse med store applikasjoner innen økonomi og finans, og ikke minst parallell-prosessering av tunge regneoppgaver. Den siste tids voksende fokus på utnyttelse av regnekraften som stort sett ligger 'død' i den enkelte PC, aktualiserer slike anvendelser – og dermed denne flytmodellen, der kritiske flyt typisk er mellom nodene. Gjentakelsesgraden kan være betydelig i tilfeller der en node skal formidle sine resultater til mange omkringliggende noder samtidig, hvilket påkaller spesiell oppmerksomhet i forbindelse med kapasitetsplanleggingen.



Distribuert databehandling utmerker seg i flyt-sammenheng ved at trafikken mellom 'sjefen' som koordinerer og 'dronene' som utfører arbeidet er sporadisk, mens mengde og hyppighet i datautvekslingen mellom klientene gjerne er stor.

Flyt karakteristikken blir dermed ofte den motsatte av hva tilfellet er for klient/tjener-modellen.

Metoder og modeller

Hensikten med flytanalysen er å avdekke viktige datastrømmer i nettverket, kunnskap som er nødvendig for kapasitetsplanlegging og design av nettverket. Hvilke metoder som benyttes for å få frem denne kunnskapen, er av mindre betydning. Det er resultatet som teller og at veien frem til pålitelige resultater er så kort som mulig.

De to mekanismene vi har gjennomgått (identifikasjon av kilder/mottakere og flytmodeller), har ulike karakteristika og angrepsvinkler, men leder frem til samme resultat – og kan gjerne blandes. Faktum er at slik blanding er regelen snarere enn unntaket.

Grenser og hierarkier

Flytinformasjonen gir ikke bare grunnlag for kapasitetsplanlegging og nettverksdesign, men også indikasjoner om hvordan hele infrastrukturen – inklusive systemer og brukere – bør organiseres. Vi har avdekket individuelle og sammensatte flyt, lokale flyt og stamnett-flyt, kunnskap som sammen med oversikten over tjenere, brukere og andre ressurser gir grunnlag for optimaliseringer.

Gjentagelsesgrad – eller 'hierarkiske egenskaper', som vi var inne på i forbindelse med flytmodeller ovenfor, er én viktig faktor: Alle nettverk over minimal størrelse inneholder naturlige grenser som rammer inn lokal-områder. Lokal trafikk skal holdes lokal, både av effektivitets-hensyn og av sikkerhetsmessige årsaker. Flytmodellene – og spesielt gjentagelsesgraden – hjelper oss å finne ut hvor det er naturlig å trekke disse grensene og hvilke mekanismer som er optimale for å utnytte gjentagelsesgraden. Ofte, men ikke alltid, havner slike grenser mellom etasjer, mellom bygninger, mellom avdelinger eller mellom geografiske områder. Mens geografi er viktig fordi avstand typisk betyr forbindelser med begrenset båndbredde, er det først og fremst trafikk og praktisk organisering som skal være styrende for våre valg.

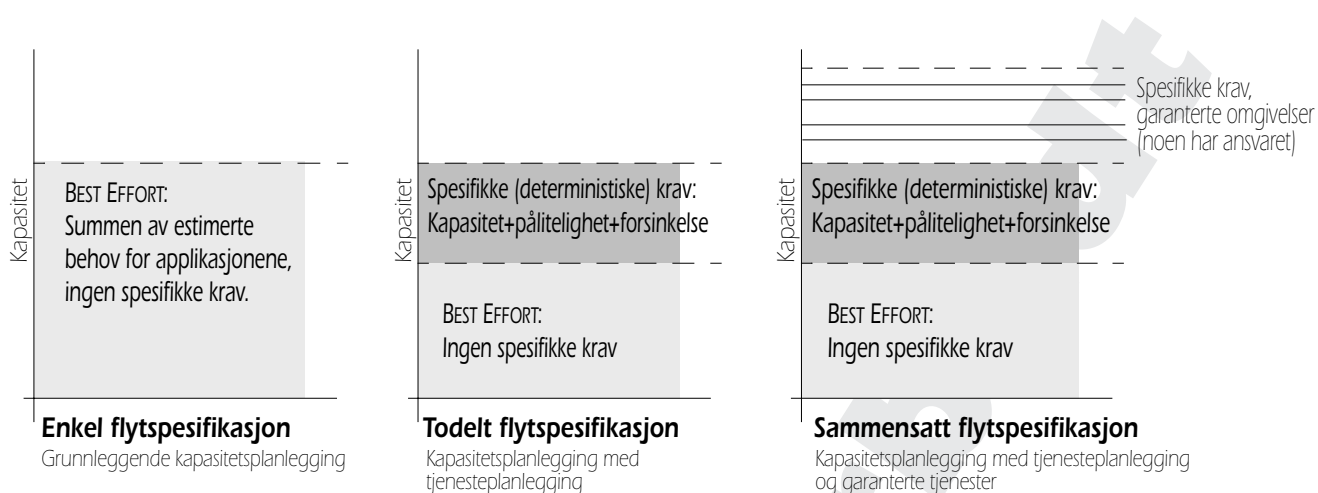
En tradisjonell 80/20-fordeling av lokal kontra fjerntrafikk er ikke like relevant i dag som for 5 år siden, fordi tilgang til og pris på båndbredde over lange avstander har forandret seg vesentlig i positiv retning. Det betyr at vi kan prioritere annerledes og at regnestykkene blir forskjellige, men evalueringen og prioriteringen må fortsatt gjøres.

God oversikt over datastrømmene i nettverket er en forutsetning for å kunne gjøre riktige valg, og gir for eksempel anledning til å vurdere flere alternativer: Fokus på kostnader, fokus på lokalitet, fokus på stabilitet (i organisasjonsmessig forstand) og så videre.

Flyt-spesifikasjoner

Armert med kunnskap om datastrømmene i nettverket og hver enkelt applikasjons krav og behov, har vi grunnlag for å utarbeide flytspesifikasjoner for de datastrømmene vi i prosessen har kategorisert som viktige (kritiske flyt). Spesifikasjonen kan innta én av tre ulike former:

- ✓ En enkel flytspesifikasjon, som brukes for kapasitetsplanlegging når detaljert informasjon om enkelt-flyt og deres karakteristika er tilgjengelig.



Figur 4 Flytspesifikasjoner kan sammenstilles grafisk på denne måten. Det sammensatte tilfellet er spesielt interessant: Garanterte omgivelser kan ikke blandes, de må alltid behandles enkeltvis, hvilket stiller spesielle krav til nettverket. Det kan ikke finnes mange slike flyt. Da blir bildet for komplisert og oversikten deretter. Mange flyter med garantikrav er et signal om at grupperingen kan bli bedre, at gjentakelsesgraden ikke er utnyttet eller at vi bør se etter muligheter for å begrense utbredelsen av en flyt (holde den innenfor et segment/partisjon/område).

teristika mangler. Dette er den tradisjonelle 'best effort' varianten, som har gitt akseptable resultater fordi kravene til forutsigbar respons og ytelse har vært beskjedne.

- ✓ En todelte flytspesifikasjon som inneholder én *best effort* (tradisjonell) del og én spesifisert del som ivaretar krav til forutsigbar respons fra én eller flere applikasjoner.
- ✓ En sammensatt flytspesifikasjon som i tillegg til de to delene ovenfor detaljerer komponenter og krav for hver enkelt flyt.

Det er åpenbart at vanskelighetsgraden vokser dramatisk fra den enkle til den sammensatte varianten. Likeledes er det knapt noen overraskelse at den todelte varianten representerer et alminnelig og akseptabelt kompromiss for de fleste omgivelser. Vi kommer fra en verden der *best effort* – 'det går som regel bra' – har vært regelen, og fortsatt er akseptabel for mange anvendelser.

En rekke nye tjenester og anvendelser som introduseres i disse dager, må imidlertid håndteres annerledes. Telefoni er det mest synlige eksemplet, men det finnes mange av dem: Alarmtjenester, videokonferanser, IM (se side 4), undervisning og virksomhetskritiske funksjoner for å nevne noen. Her er *best effort* åpenbart ikke tilstrekkelig. Dataflyt knyttet til slike anvendelser fordrer en todelte eller sammensatt flytspesifikasjon. Applikasjonenes kravspesifikasjon, som vi utarbeidet tidligere i prosessen, avdekker umiddelbart slike behov, og inneholder de data vi trenger til flytspesifikasjonen.

Følgende punkter er viktige å ta hensyn til når tallene skal settes sammen:

- ✓ Datastrømmer i *best effort* kategorien har ingen spesifikke krav til pålitelighet eller forsinkelse, og bidrar dermed utelukkende til rene kapasitetsberegninger.

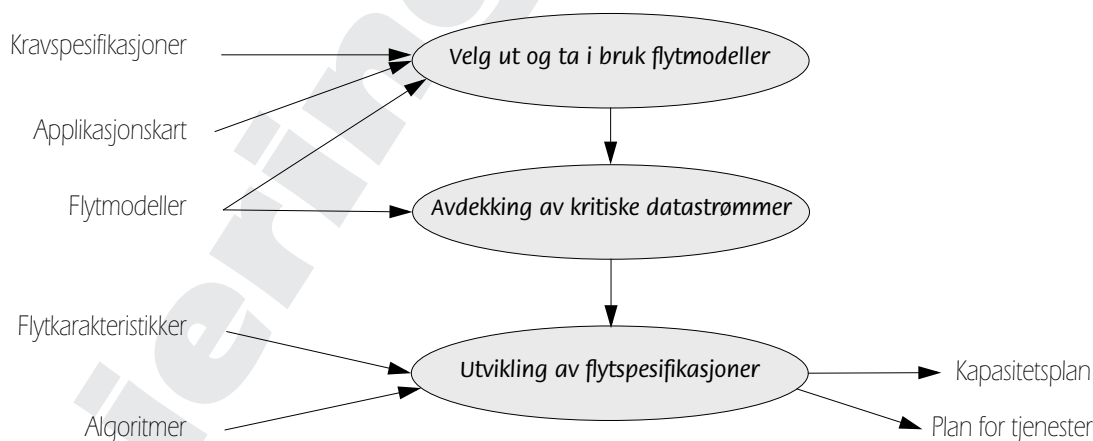
- ✓ For spesifiserte datastrømmer (flyt med spesifiserte krav til båndbredde og pålitelighet – se forrige artikkel) utnytter vi alle elementer når flytspesifikasjonen skal utarbeides. Kravene kombineres for alle datastrømmer som inngår i en sammensatt flyt.
- ✓ Absolutte krav – for eksempel til maksimal forsinkelse eller pålitelighet, behandles individuelt når flytspesifikasjonen skal kalkuleres.
- ✓ Kapasitetskravene som fremkommer fra enkle, delte og sammensatte flytspesifikasjoner, kan være estimerte, og må i så fall justeres for spesifikke krav, eller de kan være målte/simulerte.

De tre typene av flytspesifikasjoner kan fremstilles grafisk som vist på figur 4.

Flytanalyse i praksis

Det er vel og bra med modeller og mekanismer, men hvordan angriper vi flytanalysen i praksis? Veien fra teori til konkrete resultater kan være både lang og krevende inntil erfaring har rukket å fjerne alle eller de fleste spørsmålene som dukker opp underveis. Vi kan ikke eliminere behovet for praktisk erfaring, men gjøre veien lettere med enkelte observasjoner rundt selve prosessen.

Figur 5 gjennomgår prosessen og presenterer fasene på veien mot identifikasjon av datastrømmene i nettverket. Begynn med å finne



Figur 5 Flytanalyse-prosessen består av tre hovedelementer eller faser, og munner ut i en kapasitetsplan, og i de fleste tilfeller en plan for tjenester med spesielle krav eller behov.

hvilken eller hvilke flytmodeller som passer, og å identifisere de viktigste datakildene og datakonsumentene. Når flytene er funnet, skal de prioriteres, i de fleste omgivelser en relativt enkel oppgave.

Forenkling

Prosesen vi har skissert, er generalisert for å dekke flest mulig tilfeller og situasjoner. Det betyr at den noen ganger kan være for knapp eller

enkel, i andre tilfeller for omfattende. I de fleste tilfeller er forholdene tilstrekkelig oversiktlige til at vi kan slippe billigere unna. Enkle forhold kvalifiserer snarveier uten at det i nevneverdig grad går på bekostning av resultatet. Noen slike forenklinger kan være:

- ✓ Flytmodeller er viktige og forenklende i seg selv. Vi kan velge å se bort fra datastrømmer som ikke lar seg plassere i en flytmodell.
- ✓ Videre kan vi velge å se bort fra individuelle flyt. Det er de sammensatte flytene (se forrige artikkel) som er viktige og som teller i en større sammenheng.
- ✓ Se stort på små variasjoner mellom ulike datastrømmer, standardiser dem der det overhodet er mulig.

Nettverksanalyse generelt og flytanalyse spesielt er krevende disipliner hvis viktighet og nytteverdig vokser med nettverkets størrelse og kompleksitet. Som vi har vært inne på tidligere i serien, finnes det fagmateriale som behandler temaene med større grundighet en vi har mulighet for i en slik artikkelserie. James McCabes *Practical Computer Network Analysis and Design* [Morgan Kaufmann Publishers 1998] er et godt eksempel – grundig, om ikke spesielt lett tilgjengelig i formen.

Neste utgave

Dermed avslutter vi den første og største fasen på veien frem mot et moderne nettverk. I neste utgave starter vi arbeidet med å sette sammen elementene og informasjonen vi har skaffet til veie, til en nettverksarkitektur. Etter de flestes oppfatning er det nå den interessante delen av arbeidet begynner. Det meste av rutinearbeidet – datainnsamling, korrelering og systematisering – er gjort, og den mer kreative fasen kan begynne. ■