

SISU **analys**

Nr. 8

OSI

***Referensmodell
och
specifikationsmetod
av
Thomas Muth***

Redaktör: Anders Eriksson
SISU - Svenska Institutet för Systemutveckling

Box 1250, 164 28 Kista, 08 - 752 1600, Electrum, Isafjordsgatan 22, Kista

OSI - Referensmodell och specifikationsmetod

I en värld där vi kommunicerar allt mer utvecklas regler för hur vi skall kommunicera i olika situationer. En urgammal standard är den standard som vi människor har, när vi kommunicerar. Vi börjar med att säga "Hej". Är vi okända för varandra fortsätter vi med att presentera oss med namn och vad vi gör. Därefter pratar vi om vädret för att därefter ta upp det som vi träffats för. Hela sessionen avslutas med att vi säger "Hej då" och skiljs. Teknisk kommunikation skiljer sig inte mycket från det nyss beskrivna. Det är dock föga troligt att tekniska system utbyter åsikter om vädret.

Allt fler ansluter sig till de standards som ryms inom referensmodellen Open Systems Interconnection - OSI. Många sätter likhetstecken mellan standards och OSI, men det är bara en aspekt på OSI. OSI kan också användas som referensmodell och specifikationsmetod. Att betrakta OSI ur dessa två aspekter underlättar kommunikation mellan dem som utvecklar kommunikationssystem samt ökar förståelsen för hur kommunikation utformas.

I det här numret av SISU analys kommer OSI att framförallt behandlas som referensmodell och som specifikationsmetod. Numret är skrivet av civ.ing. Thomas Muth, som kombinerar sin lärar- och forskargärning på Institutionen för Data- och Systemvetenskap, Stockholms Universitet, med att utveckla metoder på Ellementel AB.

B Anders Eriksson
Redaktör

SAMMANFATTNING

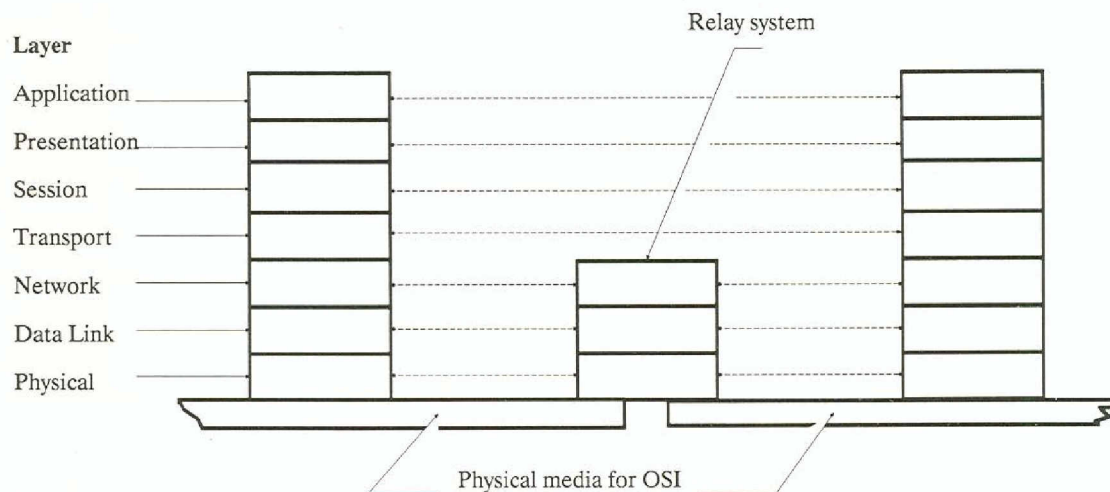
OSI representerar det kanske mest omfattande systemarbete som någonsin bedrivits. Standarder som beskriver OSI's Referens Modell, såväl som specifika protokoll och tjänster, avspeglar en genomtänkt specifikationsmetodik under stark utveckling. På senare tid försöker man inom såväl ISO som CCITT att också i allt högre grad utnyttja formella metoder för specificering av standarder. Språk har tagits fram för att beskriva datatyper (ASN.1) och för att beskriva kommunicerande processer (LOTOS, SDL, ESTELLE).

Rapporten koncentrerar sig huvudsakligen på OSI's Referens Modell. Efter en kortfattad presentation av de viktigaste begreppen i modellen, beskrivs denna ur olika aspekter, valda så att de belyser modellens generalitet snarare än de för OSI specifika arkitektur egenskaperna. OSI tjänster och protokoll beskrivs endast undantagsvist. Istället lämnas rikligt med referenser till aktuella ISO standarder och CCITT rekommendationer. De mest grundläggande av dessa är också upptagna i referenslistan.

OSI's dokumentation är mycket begreppstung. De begrepp som formellt definierats i OSI presenteras med versaler. För att undvika missförstånd utnyttjas företrädesvis engelska begrepp, där fastlagda svenska begrepp saknas (vilket är regel snarare än undantag). Att rapporten därigenom får en "svengelsk" framtoning kan tyvärr inte undvikas.

1. Inledning

Akronymen OSI står för "Open Systems Interconnection", vilket jag tror få kan ha undgått att notera. Bakom OSI står ISO (International Standards Organization). ISO samarbetar på internationell nivå bl.a. med CCITT (Consultative Committee for International Telegraph and Telephones), som är en av de organisationer som ger omfattande impulser till ISO-standarder inom kommunikationsområdet. CCITT är också den största avnämaren av OSI-standarder (genom att CCITT's rekommendationer är bindande för dess primära medlemmar - världens teleförvaltningar).



Figur 1.1.

Figur 1.1 utgör en modell av OSI som många har sett men som säger relativt lite. OSI beskrivs av ISO i för närvarande ca 160 standarddokument (runt 6000 sidor). Till detta kommer en ungefär lika stor mängd CCITT rekommendationer i V-, X-, I-, T- och F-serierna. Dessa kan i varierande utsträckning

betraktas som antingen tillämpningar av OSI-konceptet, som bidragande till konceptets utveckling eller som dubbleringar av OSI-standarder.

Mängden ord och bilder som behövs för att beskriva OSI indikerar att figur 1.1 inte är OSI-modellen utan snarare en symbol för OSI.

OSI definieras formellt av de standarder som utges av ISO. Grovt sett kan de delas in

- OSI Reference Model (OSI RM)
- Protokoll- och tjänstespecifikationer per skikt och för "management"
- Övrigt, som omfattar rubriker som
 - Formella språk.
 - Standardiserade objekt som hanteras av protokoll. Hit hör modeller av DOKUMENT, FILSYSTEM och FILER, TERMINALER, KATALOGER m.m.
 - Standarder för testning av OSI-produkter, s.k. "conformance testing".

OSI kan ses ur olika aspekter och symbolisera olika saker för olika grupper/individer såsom

- teleförvaltningar
- andra och tredje parts leverantörer av kommunikations- och nättjänster
- datorsystemleverantörer
- leverantörer av distribuerade applikationssystem
- systemutvecklare och konstruktörer
- forskare
- politiska och industriella intressentgrupper
- standardiseringsorganisationer.

Några aspekter som representerar olika sätt att se på OSI:

1. En 7-skiktad modell för kommunikationstjänster.
2. Realiseringsoberoende beskrivningar av tjänster på var och en av de 7 nivåerna i modellen.
3. En mängd protokollspecifikationer, där varje protokoll är klassat efter vilken eller vilka tjänster det utnyttjar och vilka tjänster det ger.

4. Exempel på ett framgångsrikt sätt att bedriva internationell standardisering inom ett mycket komplext område.
5. Tillämpning av en specifikationsmetod för skiktade system.

Punkterna 4 och 5 hör ihop. Standardiseringsarbetet inom OSI representerar det kanske svåraste och i vilket fall det mest omfattande systemarbetet som någonsin bedrivits - dessutom (som det verkar) med stor genomslagskraft. En bidragande orsak till framgången med OSI torde vara den specifikationsmetodik som tillämpas och vars komponenter är modellen av en skiktad "kommunikationsserver" (OSI RM) samt en alltmer stringent metodik för att beskriva tjänster och protokoll. Härvid utnyttjar man sig av formella språk och beprövade matematiska modeller (från automatteorin och Milners teori om kommunicerande processer - CCS och LOTOS).

OSI-standarder utgör därför mer än ett internationellt standardiserat alternativ till SNA, Arpanet m.fl. arkitekturer. OSI uppvisar också en specifikationsmetodik, som visserligen utvecklas och förfinas inom ISO för OSI ändamål, men som kan användas för att beskriva kommunikationsarkitekturer i allmänhet och också (i viss utsträckning) skiktade arkitekturer av annan typ.

2. OSI Referens Modell - begrepp

2.1 Grundläggande principer i OSI RM

Ett sätt att se på modeller är att betrakta dem som begreppsapparater (=språk) med vars hjälp man kan beskriva och diskutera företeelser i den konkreta världen. I den bemärkelsen är OSI RM en internationellt standardiserad begreppsapparat, som används för att beskriva verkliga kommunikationssystemens egenskaper. Samtliga begrepp klassificeras också som abstrakta i den meningen att de kan användas för att beskriva alla tänkbara realiseringar - de är realiseringoberoende.

En grundläggand tanke i OSI RM är att betrakta kommunikationssystem som SERVERS (stödsystem) åt andra system. De senare är (i förhållande till kommunikationssystemet) applikationssystem (företrädesvis sådana vars komponenter är fysiskt distribuerade såsom i distribuerade filsystem, distribuerade processorsystem, människor som kommunicerar med hjälp av kommunikationssystem etc.). Kommunikationssystem erbjuder i detta synsätt SERVICE (tjänster) åt USERS.

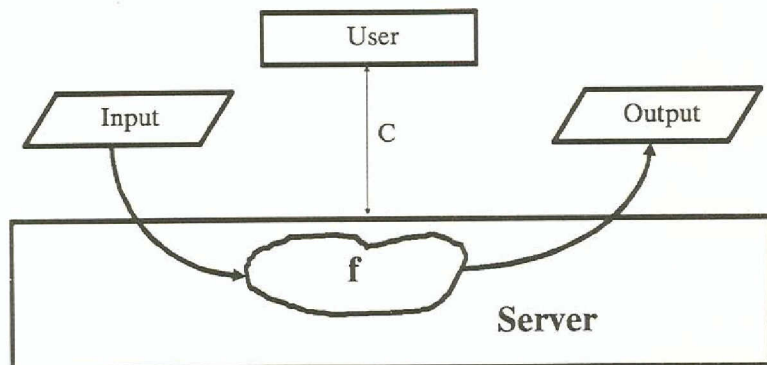
En annan bärande ide' är modellens skiktade struktur. Kommunikationssystem döljer för users den komplexa mediastrukturen och nättopologin, problemet att övervinna transmissionstörningar och mycket annat, genom att i successiva steg transformera bilden av t.ex. seriekopplade, otillförlitliga transmissionslänkar och nätinterna kopplingspunkter till en bild av enkla, logiska förbindelser med hög tillförlitlighet, mellan fysiskt åtskilda users. När users beställer tjänster av ett kommunikationssystem sker dialogen dememellan

vanligen i termer av sådana logiska förbindelser (CONNECTIONS). Dialogens element kallas för PRIMITIVER.

OSI RM beskrivs i sin helhet i /ISO 84a/ (för CCITT's motsvarighet, se /CCITT 84a/). OSI RM omfattar en mängd begrepp, för många för att kunna beskrivas något så när heltäckande här. I det följande avsnitten presenteras därför bara de väsentligaste, som en introduktion till efterföljande kapitel.

2.2 SERVER, SERVICE, SERVICE PROVIDER, SAP

En SERVER är en komponent som erbjuder någon typ av tjänst åt andra komponenter i ett system.

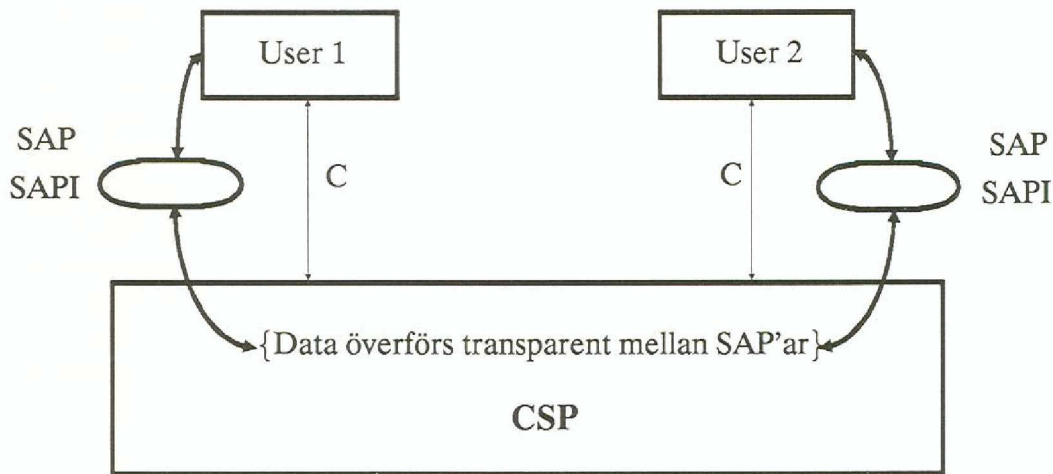


Figur 2.1.

Figur 2.1 exemplifierar en databearbetande server. Man kan även tänka sig andra typer av servers, t.ex. en som kan lagra data (DBHS) eller en som styr en process (t.ex. ett trafikstyrningsprogram i en växel). OSI definierar för skikt 1 t.o.m. skikt 6 en speciell sorts server som har minst två samtidiga användare och speciella adresseringsegenskaper. Detta är en (COMMUNICATION) SERVICE PROVIDER, där ordet "communication" vanligen underförstås. I det följande kommer jag att använda akronymen "CSP" för sådana servers.

Exempel på realiseringar av CSP's är telenätet, som ger tjänster av en kvalitet som motsvarar OSI's skikt 3. Mer kvalificerade CSP's är transporttjänstens (skikt 4) realisering i Teletex och presentations-tjänstens (skikt 6) realisering i ett elektroniskt postsystem enligt X.400-standarden.

Enligt OSI RM är en CSP's primära uppgift att transportera data mellan två eller fler angivna SERVICE ACCESS POINTs (SAP).



Figur 2.2.

En SAP är en logiskt accesspunkt i ett fysiskt system, där user kan lämna och hämta information. En SAPI (SAP-identifierare) är accesspunktens adress. Om en user önskar överföra information till en annan user måste han veta denna users SAPI och ange adressen till CSP'n.

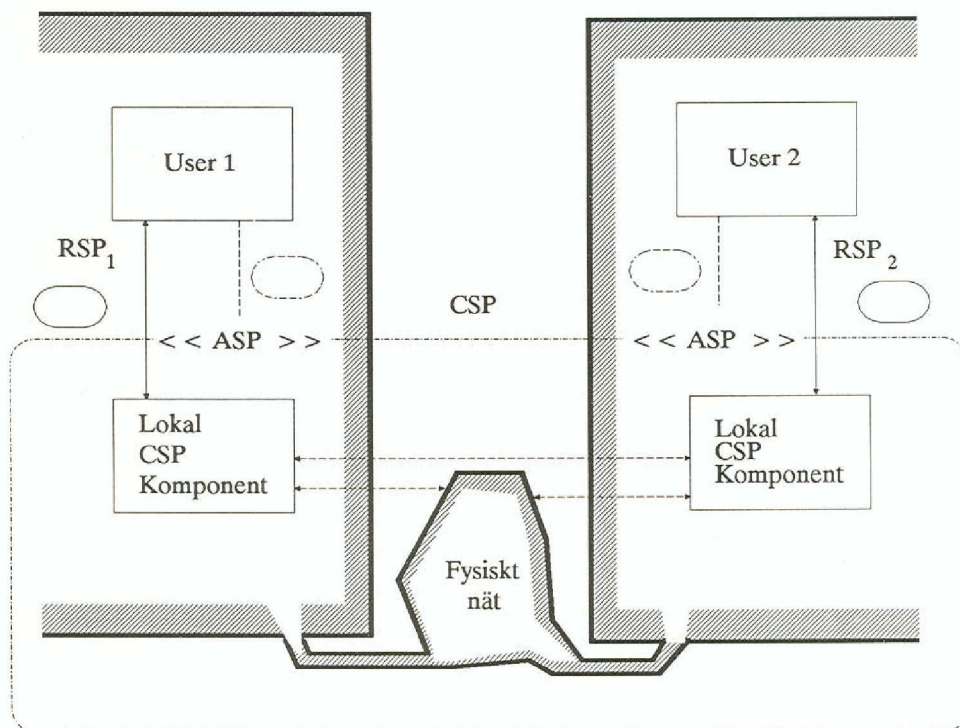
Internt i CSP'n förmedlas SAPI-värdet i protokoll-meddelanden mellan sändande, mottagande och eventuella reläande, fysiska system. Hur en SAP realiseras internt i ett fysiskt system är däremot en lokal fråga inom varje system, och specificeras inte av OSI RM. En SAP's realisering beror på vald teknologi och i praktiken främst på vilket skikt man betraktar. Exempel på realiseringar är en terminalanslutningspunkt på ett lokalt nät, en abonnentledning i ett telefonnät, en minnesarea i ett OS eller en variabel i ett proceduranrop.

Figur 2.2 visar också att den kontrollinformation (ungefär: operationskod och parametrar) som utväxlas mellan user och CSP skall hållas isär från de data som transporteras mellan users. I en realisering kan i och för sig operationen (t.ex. "Sänd till SAPI=X") och datat (det som skall sändas) läggas i samma instruktion till CSP'n, men operationskoden och dess parametrar är avsedda för CSP'n, datat för adresserad user.

En CSP representerar en kommunikationstjänst som kan utnyttjas av vilket annat distribuerat system som helst. Till skillnad från de kommunikationstjänster som impliceras av moderna programmeringsspråk, se bl.a. /SK 87/, håller CSP'n varken reda på vad komponenter i applikationssystem heter eller var de befinner sig. En CSP transporterar enbart data mellan angivna SAP's. I distribuerade datorsystem tänkes därför CSP-funktioner (OSI skikt 1-6) utgöra en för den vanlige programmeraren osynlig funktion i operativsystemet. Endast vid applikationssystemets konfiguration kommer man i kontakt med CSP-funktionen.

Om det är väsentligt att sändaren av data blir upplyst om huruvida mottagaren verkligen accesserat data, kan en CSP erbjuda en så kallad CONFIRMED SERVICE. Detta innebär att CSP'n begär en explicit kvittens från mottagande user när denne har accesserat sist överförda data. CSP'n levererar denna kvittens till sändande user.

En OSI CSP förutsätts vanligen understödja datatransport mellan olika fysiska system (t.ex. noder i ett nät), även om specifikationen för en CSP-tjänst inte gör några sådana förutsättningar.



<< ASP >> : Abstract Service Interface

○ : SAP

○ : Lokal realisering av SAP'en

RSP_{1,2} : Lokal realisering av ASP'ar.
(meddelanden, subrutinanrop, flaggor el dyl.)

←-----→ : Protokoll i CSP'n.

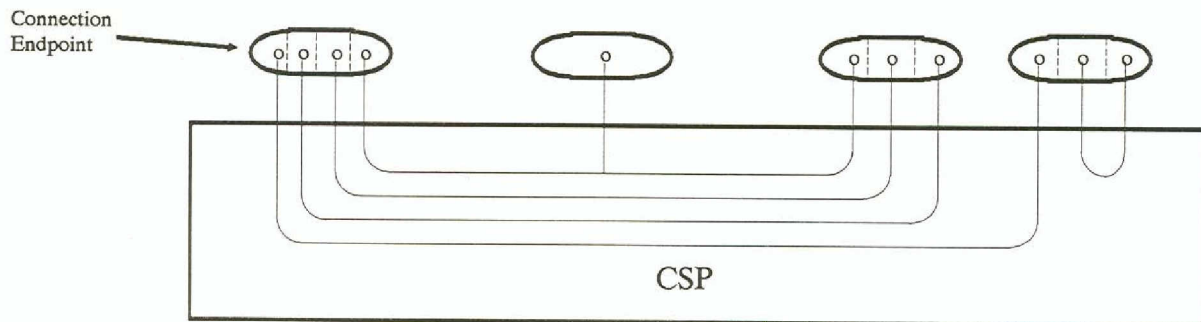
FIGUR 2.3

En CSP kan enligt figur 2.3 uppenbarligen inte byggas av 1 komponent i ett system. I figuren utgör CSP'n därför en abstrakt komponent, som kräver en intern struktur och en realisering. I den öppna miljö som OSI avser att understödja, får en CSP inte bindas till ett visst sätt att realisera kontrollinterface och SAP'ar. De kommandon/responser som utväxlas mellan lokal user och CSP uttrycks därför bara funktionellt och kallas därför ABSTRACT SERVICE PRIMITIV (ASP). Den uppsättning

primitiver som definieras för en viss CSP eller tjänst, samt regler för vilka primitivsekvenser som är tillåtna, utgör ett ABSTRACT SERVICE INTERFACE, som ur users synvinkel representerar CSP'n.

2.3 CONNECTION

En CSP kan erbjuda CONNECTIONORIENTED eller CONNECTIONLESS tjänster (eller både och).



Figur 2.4.

Figur 2.4 visar en CSP som erbjuder CONNECTIONS. Figuren visar exempel på både 2-parts connections och MULTIENDPOINT CONNECTION. Man kan tänka sig en connection som en sorts virtuell kanal mellan SAP's och att data som ligger på en SAP är öronmärkt med den connection som det tillhör. Den abstrakta variabel som identifierar ENDPOINTS lokalt kallas för CONNECTION ENDPOINT IDENTIFIER.

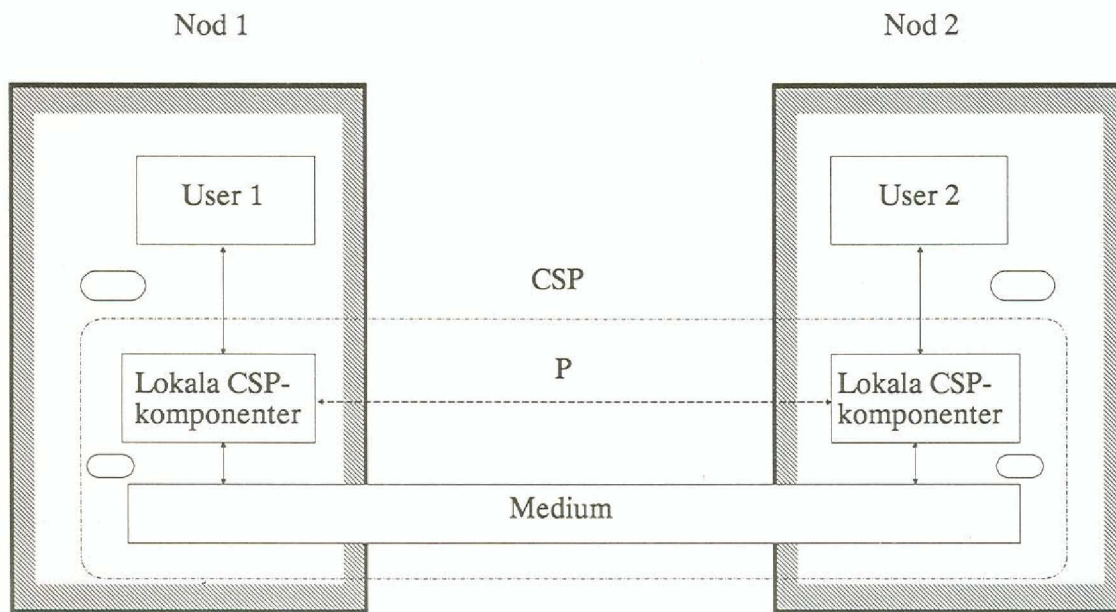
Telefonnätet erbjuder connectionoriented tjänster (en telefonkanal mellan A- och B-abonnent). Samma gäller den datakanal som kan erhållas mellan två modemanvändare via telefonnätet eller mellan två DATEX-användare.

En connectionoriented CSP kräver att förbindelsen först beställs och kopplas upp innan den kan användas för informationsöverföring. Servern belastas proportionellt mot den tid som förbindelsen står uppställd, men inte av hur mycket data som sänds över den. Vid connectionless tjänster förväntas user istället skicka fullständig adressinformation i varje datapaket som han vill överföra. Istället kan adressen växla från ett paket till ett annat och servern belastas i stort sett endast proportionellt mot antalet överförda paket.

Realisering av en connectionsless CSP kräver vanligen ett paktnät eller ett lokalt, bredbandigt nät i botten. Exempel på sådana är Arpanätets skikt 3 tjänst, det svenska DATAPAK, Ethernet m.fl.

2.4 LAYER

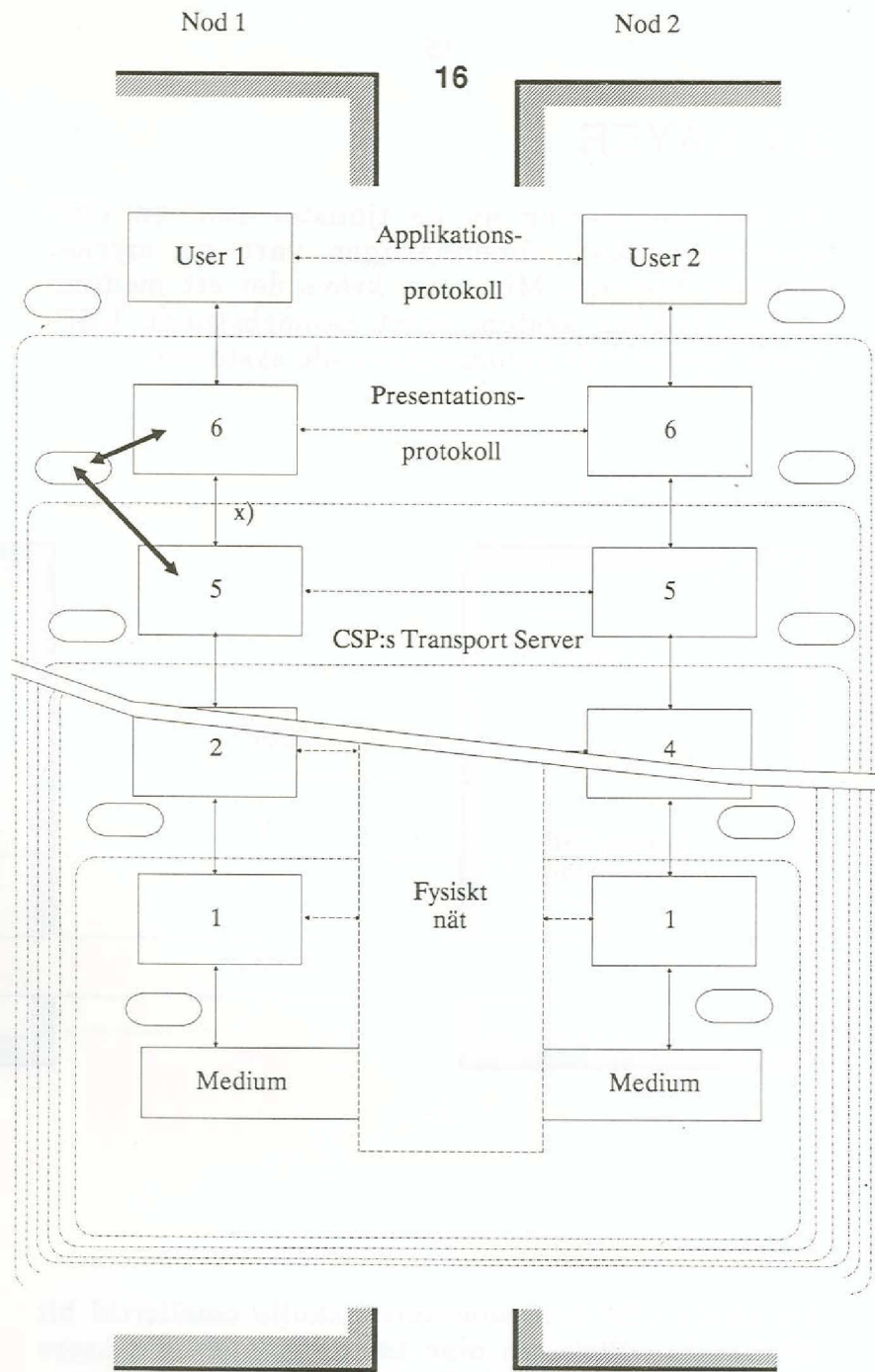
En implementering av de tjänster som en CSP specificerar kan uppenbarligen vara en mycket komplex historia. Minimum krävs det ett medium mellan fysiska system samt samarbetande CSP-komponenter i de kommunicerande systemen.



Figur 2.5.

De lokala CSP-komponenterna skulle emellertid bli mycket komplexa om man tar hänsyn till att users kan ha mycket höga krav på kommunikations-tjänsten, t.ex. att felfritt kunna överföra godtyckliga datatyper över ett fysiskt nät av vilken topologi och teknologi som helst. OSI utnyttjar därför den från Arpanätet och OS-teknologin välkända skiktning-principen för att dölja en servers komplexitet och teknologi, samt för att höja abstraktionsnivån i successiva steg.

Principerna för OSI's skiktning är väl beskrivna, främst i ISO-standarder, se /ISO 84a/, men också i /TA 81/ och /SK 87/.



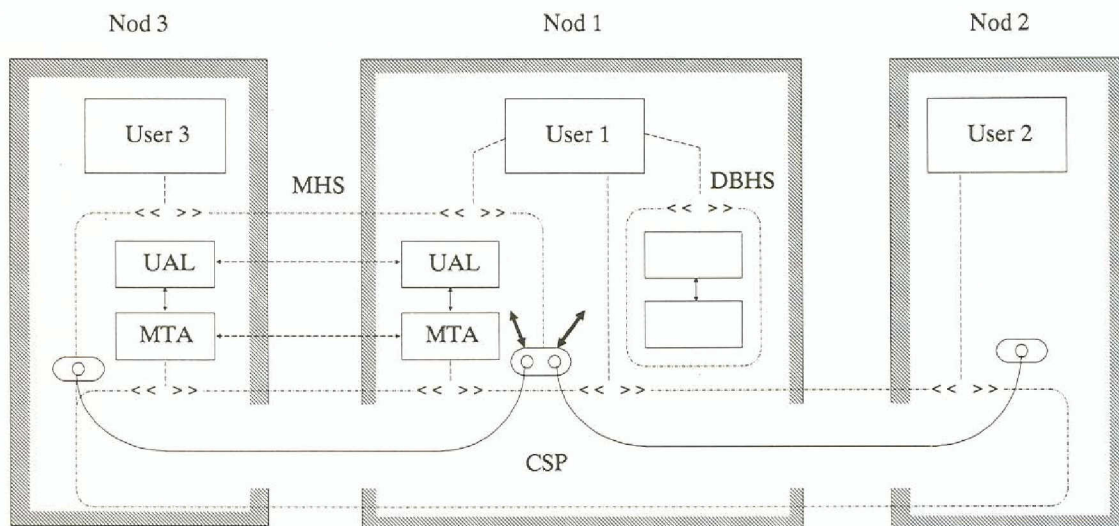
x) Realisering i Nod 1 av Sessionsskiktets ASP'ar

Figur 2.6.

Från figur 2.6 kan noteras att OSI specificerar en struktur av nästlade CSP, där alla CSP's arbetar på liknande sätt, bl.a. via SAP's. En överordnad CSP får sina egenskaper genom att på närmast lägre CSP placera en PROTOCOL MACHINE, bestående av 2 eller flera ENTITIES, vilka i samarbete (enligt de regler som föreskrivs i ett PROTOCOL) utför en LAYER FUNCTION.

2.5 SERVICE ELEMENT

Skiktningens principen omfattar en viss bestämd server. En komponent i ett system behöver emellertid normalt tillgång till många servers, både distribuerade och enbart lokalt tillgängliga. Varje server har då sin egen arkitektur, skiktad eller oskiktad.



User 1 utnyttjar en i Nod 1 lokaldatabashanterare, skickar/tar emot "mail" till/från User 3 och kommunicerar över CSP'n med User 2.

Figur 2.7.

Detta sagt som en varning för den inte alltför sällan förekommande sammanblandningen mellan ett systems komponentarkitektur och den skiktade arkitekturen för en server.

OSI utnyttjar detta synsätt bl.a. för att modellera entiteter i applikationsskiktet (skikt 7). Logiska element som skapar en server i en entitet kallas då för APPLICATION SERVICE ELEMENT och förtydligas som "A part of an application-entity which provides an OSI environment capability, using underlying services where appropriate".

2.6 FUNKTIONELLA ELEMENT I ENTITETER OCH I CSP

Många delproblem som protokoll i OSI-hierarkin skall lösa går igen i flera skikt, och har därför fått standardiserade definitioner i OSI RM. Hit hör begrepp som MULTIPLEXING, ADDRESS MAPPING, SEGMENTATION, BLOCKING, FLOW CONTROL m.fl. Den del av protokollens funktion och logik som hanterar ett sådant delproblem kallas vanligen ELEMENT OF LAYER OPERATION eller ELEMENT OF SERVICE, beroende på om man betraktar funktionens roll i protokollet eller dess effekt på den tjänst som user får. En entitet i ett skikt kan därför modelleras som en funktion sammansatt av ett antal av dessa funktionella element, samt ett eller flera för skiktets funktion unika element.

Många OSI-tjänster har dessutom för user valbara tjänsteparametrar. Tjänsten modelleras då ofta som om CSP'n hade en intern logisk struktur av FUNCTIONAL UNITs vilka är synliga för user. Varje gång user accesserar tjänsten börjar han då med att tala om för CSP'n vilka funktioner han kommer att behöva ha tillgång till under sessionen. Ett exempel på detta finner man i OSI's nättjänst (skikt 3), där user i den första primitiven som utväxlas med CSP'n, väljer om datatransporten skall vara "confirmed" eller ej och om han behöver tillgång till "expressdatatjänst" (EXPEDITED DATA). I standarderna för sessions- och presentationstjänsterna (skikt 5 och 6) beskrivs hela tjänsten i termer av sådana functional units.

2.7 PROTOCOL DATA UNIT, SERVICE DATA UNIT mm

Entiteter samarbetar genom att skicka meddelanden till varandra. Dessa innehåller för entiteterna begriplig information, uttryckt i en för dem begriplig kodning. Ett sådant meddelande kallas för PROTOCOL DATA UNIT (PDU).

Beroende på hur stora datamängder den CSP som utnyttjas kan åta sig att transportera per interaktion, kan en PDU behöva segmenteras eller alternativt flera PDU'er blockas. Den resulterande datamängden, som sedan placeras på SAP'en för transport, kallas då för SERVICE DATA UNIT (SDU). En sådan SDU kan sedan i sin tur segmenteras eller blockas innan den placeras i CSP'ns PDU'er som USER DATA.

Ibland vill man även kunna tala om HUR interaktionen mellan user och CSP går till. T.ex. kan en SDU överföras till en SAP i delar p.g.a. den valda lokala kommunikationsmekanism. En sådan del utgör då en INTERFACE DATA UNIT (IDU).

Det måste då förutsättas att en IDU innehåller sådan kontrollinformation, INTERFACE CONTROL INFORMATION (ICI), att den lokala CSP-komponenten bl.a. kan utläsa när hela SDU'n är överförd till SAP'en.

3. OSI RM - en specifikationsmetod?

Den här aspekten på OSI RM har hitills varit oförtjänt undanskymd i litteraturen. Vissa av de specifikationstekniker som används i OSI RM är tillämpliga allmänt för modellering av "servers", d.v.s. tjänster i system. Hit hör ASP- och protokollrelaterade begrepp (det senare för modellering av distribuerade tjänster). Andra begrepp och beskrivningstekniker i OSI RM, t.ex. SAP-begreppet, är specifikt för kommunikationssystem, men inte bundet till OSI's 7-skiktmodell.

Dock börjar dessa aspekter på OSI RM tona fram, t.ex. i /RC87/, där man bl.a. finner formuleringen att "DDN (Arpanet) arkitekturen kan betraktas som en tolkning av OSI".

I OSI RM definieras ett antal abstrakta termer med vars hjälp godtyckliga kommunikationsarkitekturer och protokollfunktioner kan beskrivas. Förutom de tidigare nämnda definierar OSI RM också ett antal generella begrepp med vars hjälp kommunikationstjänster kan klassificeras och beskrivas. Hit hör NORMAL- och EXPEDITED DATA TRANSFER, en tjänstestruktur bestående av PHASES och SERVICES, CONFIRMED- och UNCONFIRMED SERVICE, en kategoriindelning av tjänster i MANDATORY-, PROVIDER OPTIONAL- och USER OPTIONAL.

En annan grupp av tjänsteorienterade begrepp är till för att beskriva interaktionen mellan USER och SERVICE PROVIDER. Hit hör primitivbegreppet ABSTRACT SERVICE PRIMITIVE (ASP) samt dess klassificering i REQUEST, INDICATION, RESPONSE och CONFIRM, begreppet SERVICE DATA UNIT (SDU) och PROTOCOL DATA UNIT (PDU) samt relationen mellan ASP, SDU och PDU.

Sammantaget utgör denna begreppsapparat termer i ett abstrakt språk för att specificera kommunikationsarkitekturer och tjänster i dessa. Det här språket bygger enbart på principen med skiktade kommunikationssystem, en princip som härstammar från Arpanätsutvecklingen i slutet av 60-talet, och som utnyttjas i alla de leverantörsspecifika arkitekturer som kommit fram under 70-talet.

OSI standarder för protokoll och tjänster utnyttjar givetvis det här språket - med tillägg av de för OSI's skiktindelning speciella, skiktrelaterade begreppen. Samma teknik som inom OSI-världen kan också användas för att t.ex. beskriva DDN(Arpanet), SNA(IBM), DNA(DEC) eller varför inte arkitekturen för slutna system såsom AXE (Ericssons och Televerkets publika växelsystem). Så är naturligtvis inte fallet av historiska och andra skäl. Skiktningens principen möjliggör emellertid ändå jämförelser mellan OSI och andra arkitekturer, även om skillnaden i terminologi naturligtvis försvårar.

4. OSI RM

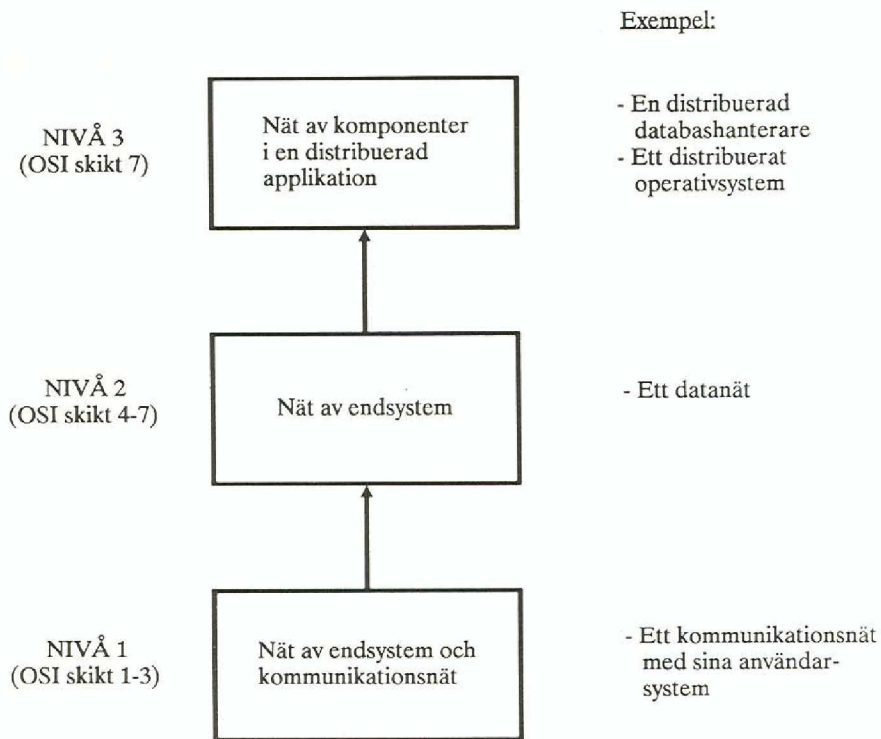
- En modell med 7 skikt

En aspekt på OSI RM är de 7 namngivna skikten. Här har vart och ett av skikten ett specifikt syfte och funktion.

OSI RM beskriver de 7 skiktens funktioner och tjänster i allmänna termer. Det medför att varje enskild OSI-standard för en tjänst (en SERVICE DEFINITION) eller ett protokoll (en PROTOCOL SPECIFICATION) är en "instansiering" (=exempel) av OSI RM's skiktdefinition. Det innebär också att det av ISO valda sättet att realisera kraven från referensmodellen i tjänster och protokoll (se följande kapitel) utgör ett sätt av många tänkbara.

Beskrivningsmässigt framgår modellens allmänna karaktär av att man i OSI RM använder naturligt språk för att beskriva såväl vilken typ av funktion som krävs i ett skikt, som vilken typ av tjänst som erhålles. Däremot ger man sig inte in på att beskriva gränssnittet mellan user och service provider, eller på att beskriva hur skiktfunktioner är distribuerad och vilka meddelanden som protokollentiteter utväxlar.

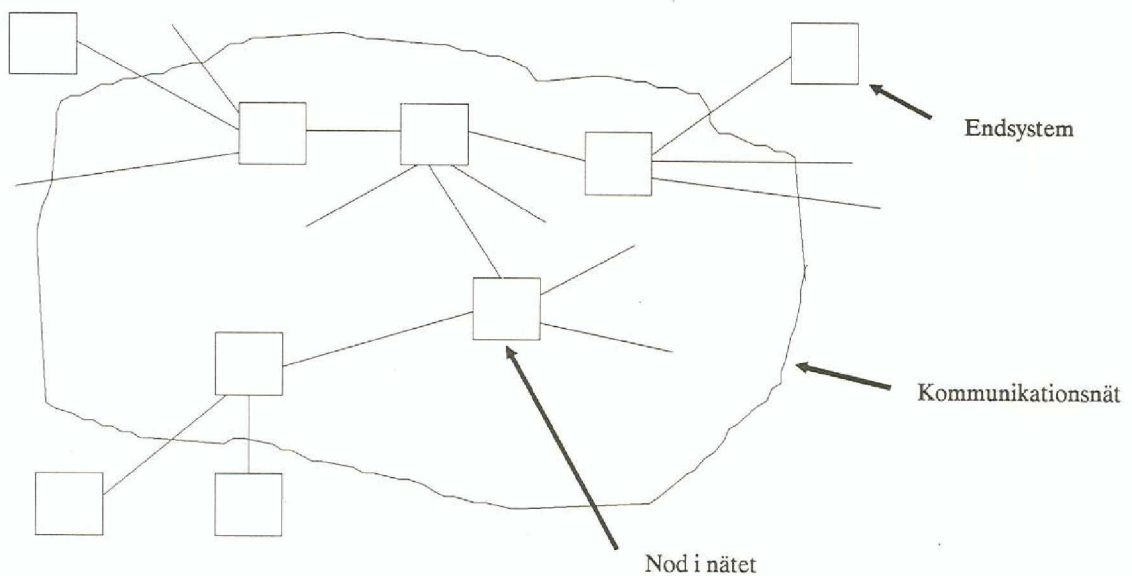
OSI's 7-skiktsmodell kan delvis härledas genom att analysera ett distribuerat systems nätstruktur på 3 olika abstraktionsnivåer:



Figur 4.1.

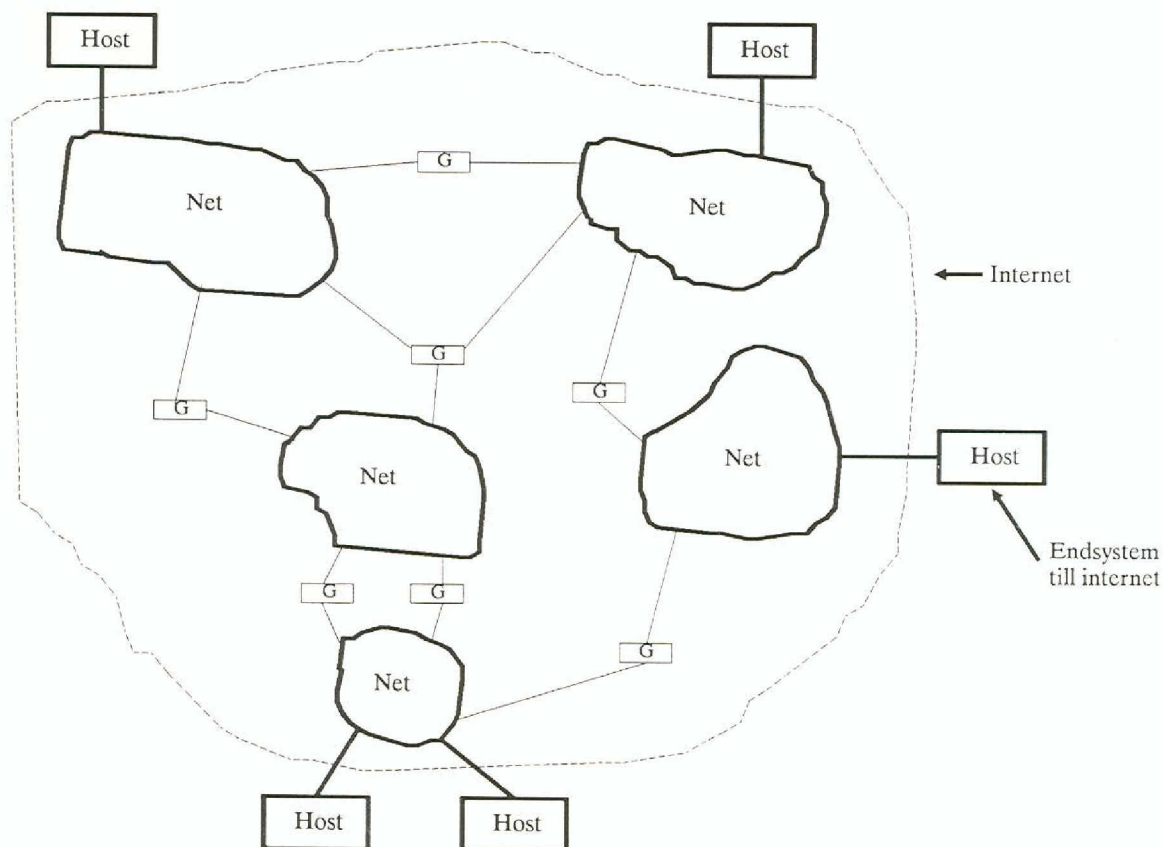
NIVÅ 1:

Nivå 1 beskriver ett logiskt nät, vars komponenter består av ett kommunikationsnät och ett antal ENDSYSTEMS. Endsystem kan utgöras av terminaler, av datorer, telefoner eller grupper av fysiska komponenter (t.ex. modem, front-end dator, värd-dator), medan kommunikationsnätet kan vara ett komplext system av transmissionslänkar och kopplingsystem.

**Figur 4.2.**

De kommunikationsfunktioner som behövs för att denna nivå skall kunna understödja nästa nivå representeras i OSI RM av skikt 1, 2 och 3. Det fysiska mediet betraktas idag som en integrerad del av skikt 1 (ett synsätt som kan diskuteras).

Ett kommunikationsnät måste ofta byggas av ett antal redan befintliga nät, s.k. subnät (SUBNET), som kopplas ihop via gateways eller relänoder, för att på så sätt skapa ett överordnat nät, ett INTERNET eller GLOBAL NETWORK.

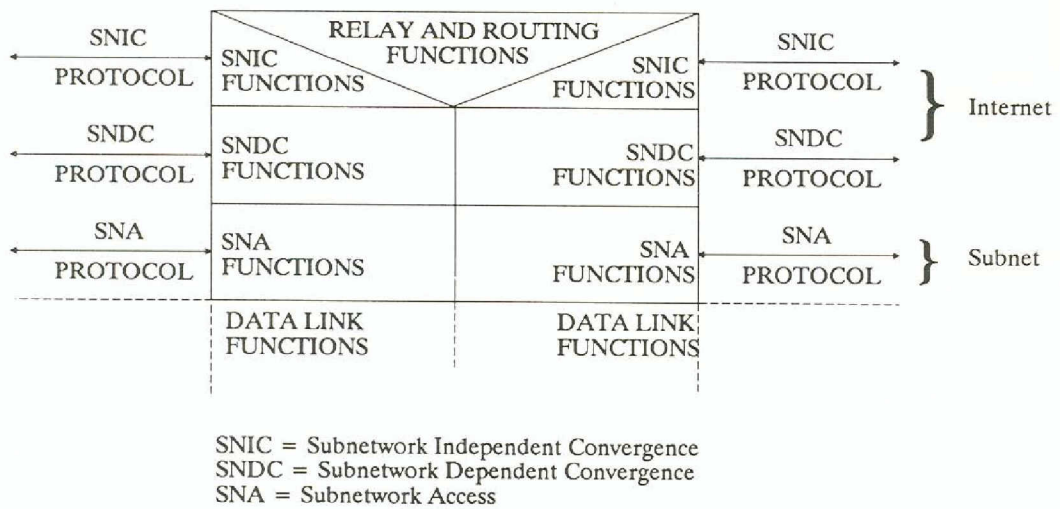


Figur 4.3.

Detta innebär att såväl endsystem som gateways måste ha ett extra skikt 3, ett internet skikt ovanpå den skikt 3 funktion som krävs för att accessera subnät. Med hänsyn till att de subnät som utnyttjas kan vara mer eller mindre ofullständiga och olika med avseende på den tjänst man vill erbjuda skikt 4, kan det vara motiverat att dela in internet skiktet i 2 subskikt. Det undre, SNDC i figur 4.4, lyfter upp subnätstjänsten till önskad kvalitetsnivå, det övre,

SNIC, styr vägval och förmedlar data via subnäten. Den skikt 3 interna struktur som diskuteras inom OSI idag åskådliggörs av figur 4.4, som visar en tänkt bestyckning av en gateway i ett internet, uttryckt i OSI's något klumpiga och övertydliga benämningssätt.

Organization of the Network Layer



Figur 4.4.

NIVÅ 2:

På nästa abstraktionsnivå underförstår man funktionerna på nivå 1, d.v.s. skikt 3 tjänster anses tillgängliga för transport av data mellan de system som beskrivs. Nu beskrivs ett distribuerat system som ett nät av kommunicerande endsystem, t.ex. en distribuerad processorarkitektur, ett nät av samarbetande datorsystem, ett informationsnät bestående av ett antal databassystem och användare eller det logiska nät som världens telefonnätanvändare utgör.

Den här nivån innehåller OSI's skikt 4, 5 och 6 samt ett antal grundläggande funktioner i skikt 7 för stöd till distribuerade applikationssystem. Dessa funktioner är de s.k. CASE- eller ACSE-funktionerna, se kap. 6.2.

Man kan lägga märke till att alla protokoll på den här nivån är END-TO-END, relativt nivå 1 (d.v.s. skikt 3) Det betyder att systemkomponenter som utför den här typen av funktioner bara kan finnas i nivå 1's endsystem.

NIVÅ 3:

Här beskrivs distribuerade applikationssystem, vilka förutsätts ha tillgång till nivå 2 tjänster (skikt 6 och eventuella ACSE-funktioner). Ett sådant system utgörs då av ett nät av kommunicerande komponenter eller processer, t.ex. en användarunik distribuerad applikation, ett meddelandehanteringssystem, ett distribuerat operativsystem eller en distribuerad databashanterare.

Den här nivån representerar övriga funktioner i OSI's skikt 7. En del funktioner är av allmänt intresse (t.ex. de ovan nämnda) och karakteriseras då som standardiserade service element i skikt 7, så kallade SASE- eller ASE-funktioner, se kap. 6.

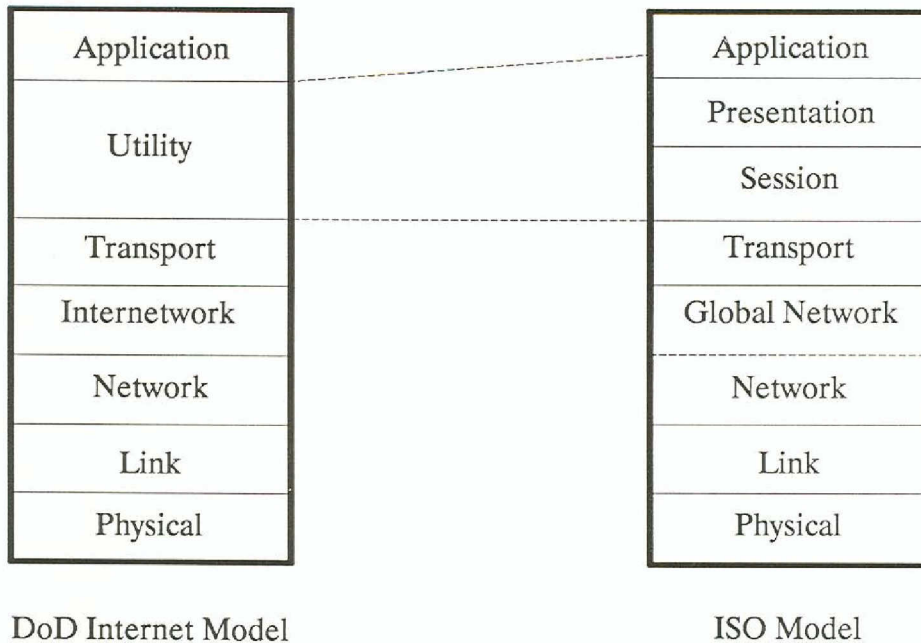
I förhållande till nivå 1 finns komponenter som utför nivå 3 funktioner också enbart i endsystem. Däremot kan en ASE-tjänst, t.ex. ett meddelandehanteringssystem (MHS), innebära att endsystem i sin tur indelas i endsystem respektive system som ingår i ett kommunikationsnät för meddelandehantering.

De senare utgör då nätnoder i detta kommunikationsnät.

Behovet av såväl ACSE- som ASE-funktioner varierar mellan olika applikationer och mellan olika OSI-baserade nät. Dessa funktioner betraktas därför som från varandra oberoende tjänster i skikt 7 (d.v.s. de utgör INTE subskikt i skikt 7). ACSE- och ASE-element är tillgängliga för applikationsunika komponenter i skikt 7, antingen via lokalt implementerade gränssnitt (t.ex. som OS-funktioner) eller kan de integreras i komponenterna.

Genom att OSI RM till väsentlig del baseras på en syn på begreppet distribuerat system som torde vara gemensam för alla arkitekturer, finns det stora likheter mellan OSI RM och andra arkitekturer såsom DDN, SNA DNA m.fl. Speciellt gäller det skikt 1 till 3 samt den mest grundläggande end-to-end funktionen:

TRANSPORT-funktionen (skikt 4). Syftet med denna är att skapa en felfri, transparent bitorienterad kanal mellan endsystem, som för användare av denna kanal (komponenter i skikt 5 till 7) döljer - så gott det går - de varierande egenskaper och topologier som underliggande nät har. Däremot råder det större olikheter mellan OSI och andra arkitekturer, relativt OSI's skikt 5 till 7:



Figur 4.5.

Som framgår av figur 4.5 saknar DDN (Arpanet) explicita SESSIONS- och PRESENTATION skikt (skikt 5 och 6), såväl som ACSE och ASE. Detta innebär naturligtvis inte att motsvarande funktioner saknas i DDN's "utilities", utan enbart att de är integrerade i och unika för varje utility.

Delvis som en effekt av utvecklingen av de publika telematiska tjänsterna (Teletex, Videotex, Telefax och Mixed Mode) fann man inom OSI tidigt ett behov av att göra synligt såväl dialog- och sessionsstyrningsmekanismer som kodning av de data som komponenter i applikationsskiktet skickar till varandra. Ett datatypsbeskrivande språk, ASN.1 = Abstract Syntax Notation nr 1, samt kodningsalgoritmer för presentationsskiktet finns därför definierade som "draft standard", /ISO 85a/ och /ISO 85b/. Beträffande sessions- och presentations-skiktets tjänster och protokoll, se /ISO 84b/, /ISO 84c/, /ISO 86a/ samt /ISO 86b/.

Andra påtagliga skäl till OSI's skikt 5 och 6 ligger i OSI's primära målsättning, d.v.s. att stödja kommunikation mellan heterogena system och att underlätta konstruktion av applikationskomponenter. Uppenbarligen medför OSI skikt 5, 6 samt ACSE att konstruktörer av såväl utilities som andra komponenter i högre grad kan koncentrera sig på de för applikationen unika funktionerna.

Å andra sidan medger DDN-arkitekturen betydligt effektivare realiseringar än OSI synsättet. Det hävdas också av en del författare, t.ex. i /SK 87/, att sessionsstyrning i de flesta fall bör vara applikationsunik och därför inte motiverar något sessionsskikt.

Jämfört med DDN's utilities (funktioner för filhantering, electronic mail, kataloger, terminalhantering m.m.) så utgör motsvarande OSI-utilities ASE-funktioner. Dessa utnyttjar både de tjänster som presentationskiktet ger och ACSE-tjänster. ACSE understödjer bl.a. namnanrop mellan komponenter samt transaktionshantering i distribuerade applikationer med höga krav på säkerhet. Exempel på applikationer med sådana behov är distribuerade databaser och banksystem. ACSE beskrivs i /ISO 86c/ och /ISO 86d/ samt i /ISO 85c/ och /ISO 85d/.

Det bör i detta sammanhang påpekas att även andra arkitekturer, speciellt SNA, DNA och XNS (Xerox) starkt påverkat utvecklingen av OSI's skikt 5 till 7. Dessa arkitekturers skiktindelning påminner därför mer om OSI's än vad DDN gör.

Som en summering kan man konstatera att OSI RM's skiktade modell har en generalitet långt utöver OSI. Orsaken ligger dels i att modellen begränsar sig till att beskriva (i naturligt språk) VAD ett skikt utför, inte HUR (varken protokollspecifikationer eller tjänstedefinitioner ingår i OSI RM). Detta medför att många tjänster är i stort sett identiska mellan olika arkitekturer. Det är därför inte speciellt förvånande att t.ex. den av IP- och TCP-protokollen skapade transporttjänsten i DDN, uppvisar mycket små skillnader relativt OSI's transporttjänst, vilket bl.a. konstateras i /RC 87/.

5. OSI's tjänster

5.1 Inledning

I det föregående beskrevs OSI's "server" begrepp, COMMUNICATION SERVICE PROVIDER (CSP). Där beskrevs också hur en sådan server kan ges en nästlad struktur av protokollmaskiner och interna CSP.

En specifikation av en OSI-tjänst syftar till att så exakt som möjligt specificera de tjänster som en CSP levererar/skall kunna leverera, utan att

- a ta ställning till i vilken nod i ett distribuerat system (nivå 1, 2 eller 3) användaren befinner sig (och därmed där tjänsten måste vara tillgänglig).
- b ange vilken mekanism som utnyttjas i noden för att den "primitiva händelse" som en "abstract service primitive" (ASP) uttrycker, skall hända
- c ta ställning till effektivitets och tillförlitlighetsaspekter på CSP.

En OSI SERVICE DEFINITION kan då sägas utgöra en "realiseringsoberoende specifikation av kommunikationstjänsten".

I det följande diskuteras en del viktiga aspekter på OSI's tjänstedefinitioner:

För att förstå OSI's sätt att specificera tjänster behöver man förstå begreppet ASP, se avsnitt 5.2, speciellt hur ASP'ar relateras till meddelanden. Det är också väsentligt att man kan skilja på det abstrakta interfacet som en ASP-baserad tjänste-

definition utgör och motsvarande konkreta interface i verkliga system, se avsnitt 5.3.

OSI's tjänstedefinitioner börjar också få ett alltmer enhetligt utförande, se avsnitt 5.4.

I avsnitt 5.5 ges en kort översikt över standardiseringsläget på det här området.

Med tanke på att OSI relativt andra arkitekturer ännu inte resulterat i kommersiellt hållbara produkter på alla områden, diskuteras i avsnitt 5.6 en del tänkbara strategier för att blanda OSI-produkter med andra och förbereda övergången till rena OSI-miljöer.

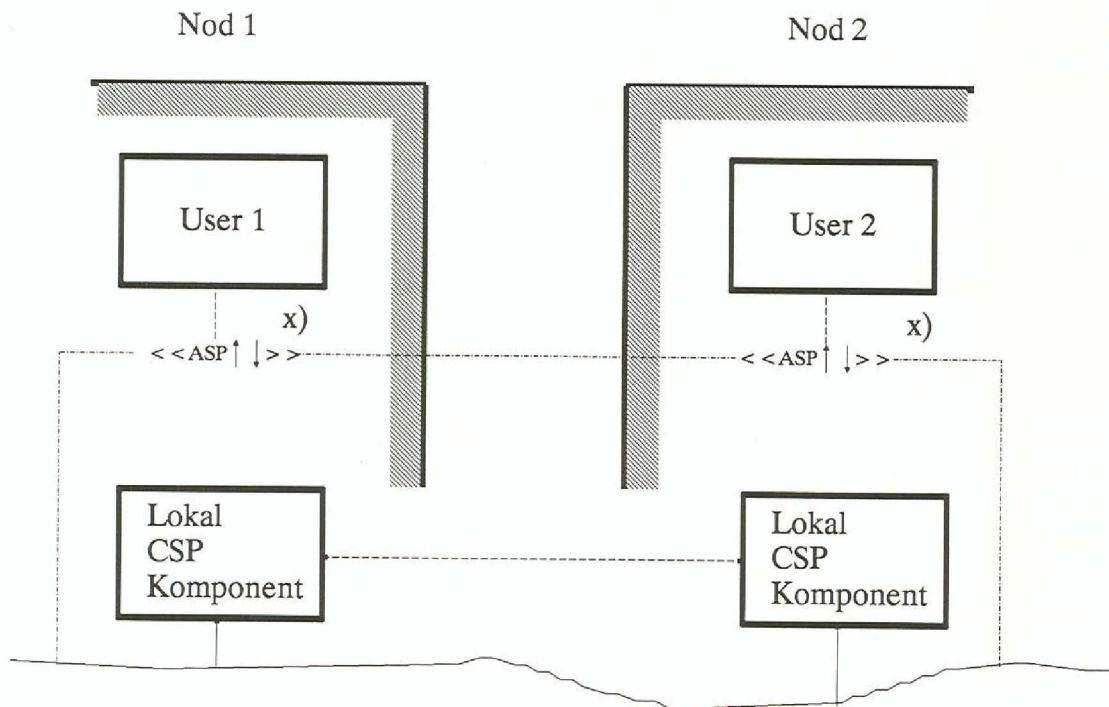
5.2 Abstrakta service primitiver

Begreppet ASP är fundamentalt för OSI's sätt att beskriva tjänster. En ASP beskriver en primitiv händelse - ett EVENT - dvs inte någon viss kommunikationsmekanism såsom message passing, proceduranrop eller delat minne. En händelse kallas ibland också för "en atomär form av synkron interaktion". Denna definition användes bl.a. i språket LOTOS, (BRI 85). Den kan också kräva sin förklaring:

Beskrivningen av en händelse anger dess innebörd eller syfte, inte hur systemet, där de interagerande komponenterna befinner sig, bär sig åt för att överföra information om händelsen mellan komponenterna. Följaktligen, om den kommunikationsmekanism som utnyttjas inte lyckas överföra all information om önskad händelse mellan komponenterna, så har händelsen inte heller hänt. Därmed har inte heller någon synkronisering skett mellan dem.

Med hänsyn till definitionen av en tjänst är det inte heller intressant att specificera den mängd underliggande händelser som kan krävas lokalt för att skapa händelsen, eftersom dessa troligen är olika i olika noder och system (speciellt i den heterogena miljö av olika system som är den OSI avser att stödja).

Ett exempel:



x) Pilarna markerar vem som initierar en ASP

Figur 5.1.

Både user 1 och 2 kan t.ex. begära att få en "connection" i en "request-ASP" och få ett svar av CSP i en "confirm-ASP". I OSI-språk är detta händelser som beskrivs av ASP'ar av typ

```
<CSP-id> - CONNECT request (parametrar);
<CSP-id> - CONNECT confirm (parametrar);
```

I det här fallet kan man tänka sig att U2 och CSP-komponenten i N2 är programenheter i ett och samma system, och att båda händelserna specificeras i ett proceduranrop i U2, t.ex. som

```
CSPCALL OPEN (IN: requestparametrar :OUT:
confirmparametrar);
```


U1 däremot kan vara en operatör som kommunicerar med en lokal kommunikationsdator (som innehåller den lokala CSP-komponenten) via terminal. Var och en av händelserna kräver då troligen en dialog med en serie av transaktioner (kommandon och responser), där varje enskild transaktion innehåller en del av ASP-informationen. En transaktion utgör i sig dessutom en serie av teckenöverföringar mellan terminal och dator.

I båda fallen handlar det emellertid om exakt samma typ av händelse. Det händelserna har gemensamt i båda noderna är

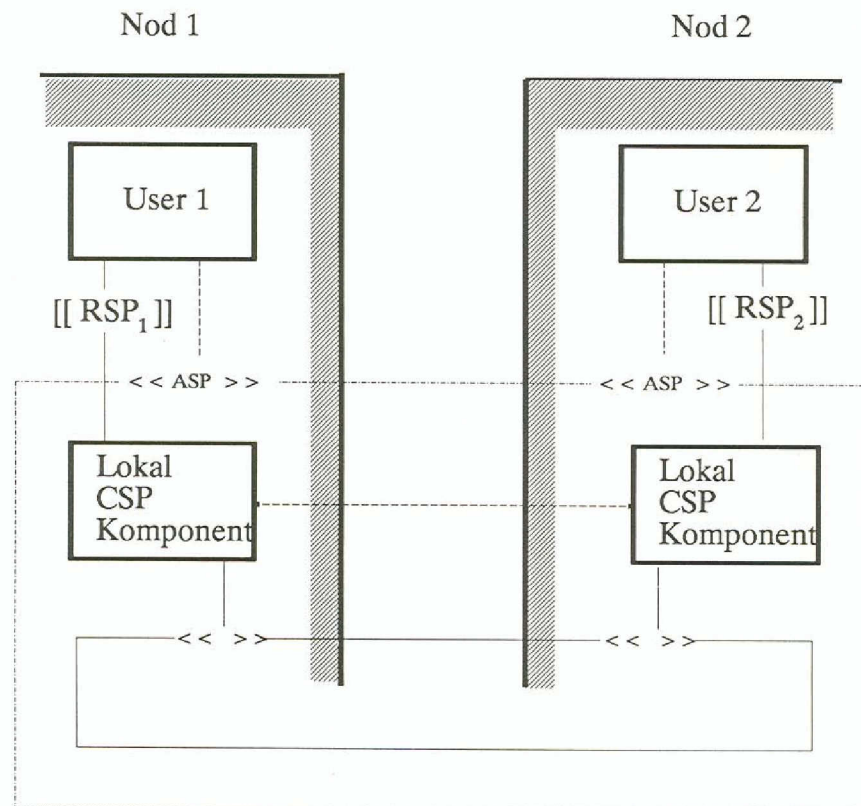
- semantiskt sett samma parametrar överförs mellan user och lokal CSP-komponent i båda noderna, även om det sker med helt olika transaktionsmönster, syntax och kodning.
- resulterar i en (gemensam) protokollaktivitet mellan noderna.

Specifikationen av en ASP beskriver alltså en atomär händelse ur CSP-synpunkt, inte ur realiserings-synpunkt. Den är dessutom formell enbart på så sätt att den entydigt

- a definierar vilken CSP den tillhör
- b anger riktning (initierad av user - request, response, initierad av CSP - indication, confirm)
- c anger typ av tjänst (i request och indication)
- d anger om begäran av en tjänst bekräftas (förekomster av respons och confirm)
- e anger den exakta betydelsen av varje ingående parameter, uttryckt i termer som baserar sig på den tjänst som primitiven avser.

5.3 Abstrakta och konkreta interface

ASP'ar utgör grundelement för att beskriva tjänster (eller snarare hur man kommer åt tjänster). De är synkrona, atomära händelser mellan users och providers (CSP) och ingår i legala händelsesekvenser i snittet mellan user och provider. Detta snitt kallas därför ASBTRACT SERVICE INTERFACE, till skillnad från det konkreta interfacet i en nod eller ett system som är dess realisering.



`<< ASP >>` Abstract Service Interface
(Samma definition i båda noderna)

`[[RSP1]]` Konkret Service Interface i nod N1
(En realisering av `<< ASP >>`)

`[[RSP2]]` Konkret Service Interface i nod N2
(En realisering av `<< ASP >>`)

Figur 5.2.

För att förtydliga skillnaden mellan abstrakta och konkreta interface - några ord om det senare: Ett konkret interface representeras t.ex. av de primitiver i ett konstruktionsspråk som utnyttjas vid en implementering av komponenter. Dessa primitiver har en semantisk innebörd som understöds av det system där komponenten arbetar, samtidigt som de överför data motsvarande ASP-information. Vilka mekanismer som behövs beror av den lokala systemkonfigurationen. User och CSP-komponent kan t.ex. vara processer under samma OS, vilket möjliggör användningen av en OS-understödd IPC-mekanism (IPC = Interprocess Communication). En alternativ konfiguration av komponenter i S1 kan placera U1 och den lokal CSP-komponenten i olika fysiska system, t.ex. U1 i en värddator och CSP-komponenten i en front-end kommunikationsdator. Det konkreta interfacet utgörs då av meddelanden som utväxlas mellan dessa datorer, understödda av en lokal kommunikationsmekanism.

De språkliga primitiver som används kan antingen göra den lokala kommunikationsmekanism osynlig för user (t.ex. genom ett proceduranrop per händelse eller tjänst), eller synligt (i form av OS-understödda IPC-primitiver av typ "SEND" och "RECEIVE"). Exempel på det förra, se /CN 84/, på det senare i bl.a. /SK 87/.

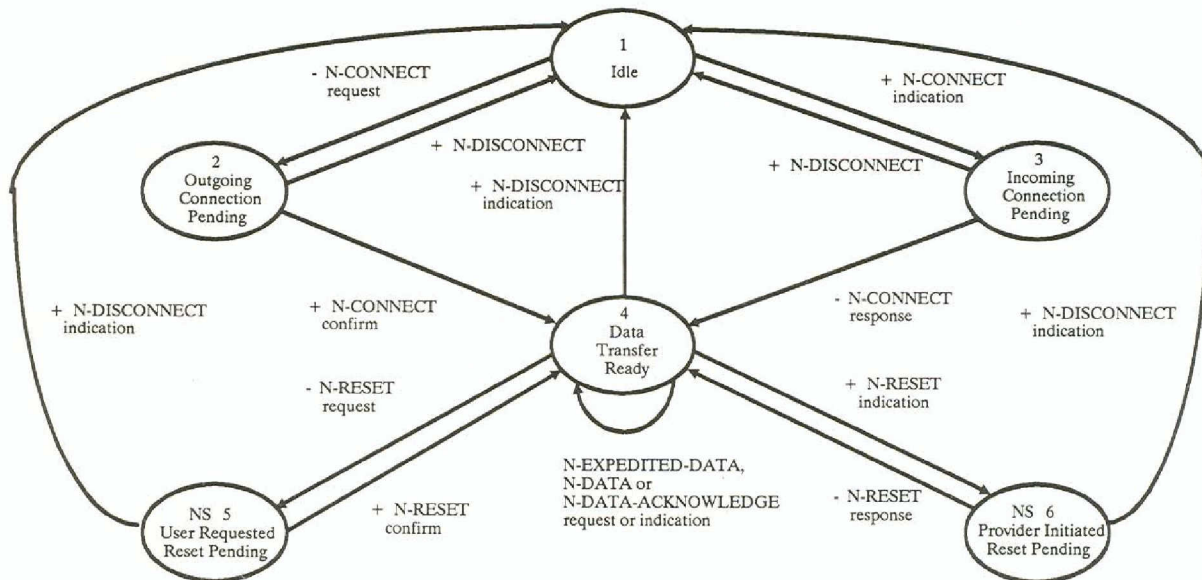
5.4 Metoder för att beskriva tjänster

Beskrivning av en OSI CSP kallas för en SERVICE DEFINITION. Den/de tjänster som servern levererar beskrivs vanligen ur 3 aspekter:

- a Den tänkta CSP-strukturen, bestående av functional units och deras funktioner beskrivs, främst i syfte att skapa förståelse för de parametrar som ingår i ASP'ar. Man anger också vilka enheter (om några) som skall vara valbara för user.
- b ASP'arna grupperas i de olika tjänster som CSP'n kan erbjuda och beskrivs syntaktiskt.
- c Därefter beskrivs de legala händelsesekvenserna av ASP'ar i det abstrakta interfacet mellan CSP och users.

ISO (och även CCITT) utnyttjar två olika matematiska modeller för händelsebeskrivningen, dels en tillståndorienterad, dels en händelsesekventiell modell. Det förekommer att båda modellerna utnyttjas i samma standard, sannolikt på grund av deras olika uttrycksförmåga.

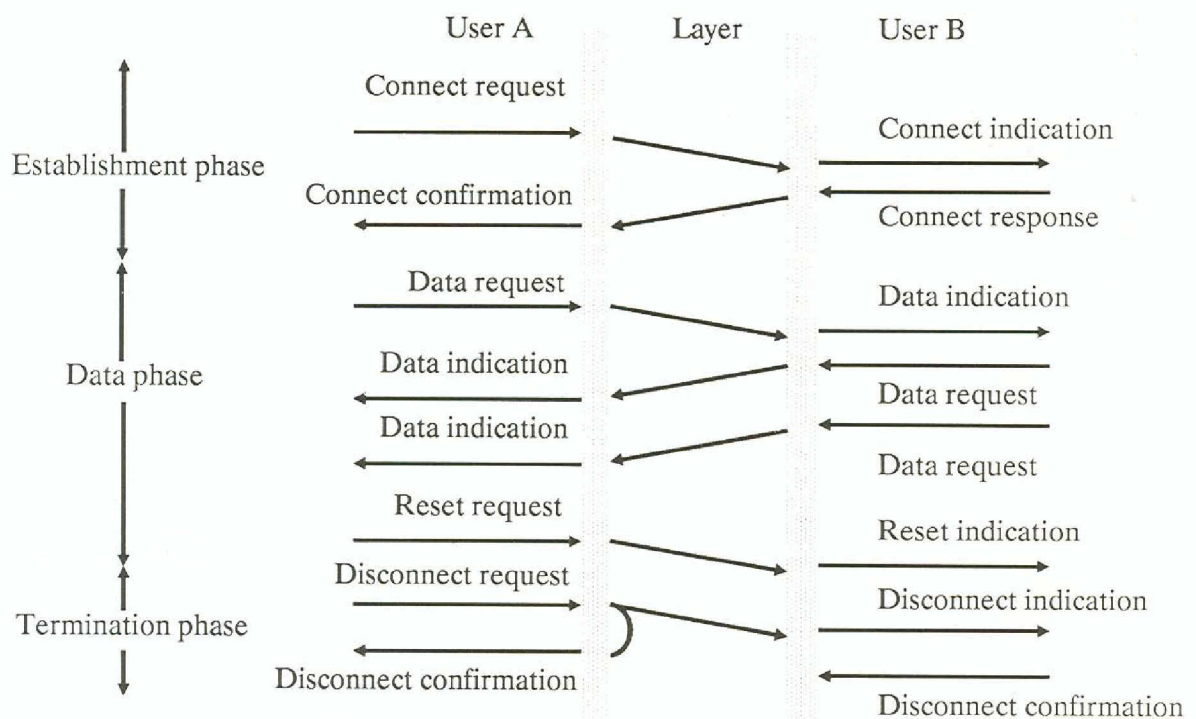
Den tillståndorienterade metoden beskriver samtliga legala händelsesekvenser i det abstrakta interfacet. Service providern betraktas som en tillståndsmaskin som via primitiver indikerar sitt tillstånd med avseende på den av user begärda tjänsten:



Figur 5.3.

Figur 5.3 visar CSP's tjänstetillstånd sedda ur users synvinkel. "+" markerar primitiver som initierats av servern, "-" primitiver som initierats av user.

I den händelsesekvensiella modellen betraktas CSP'n som en black box och man beskriver endast legala primitivsekvenser:

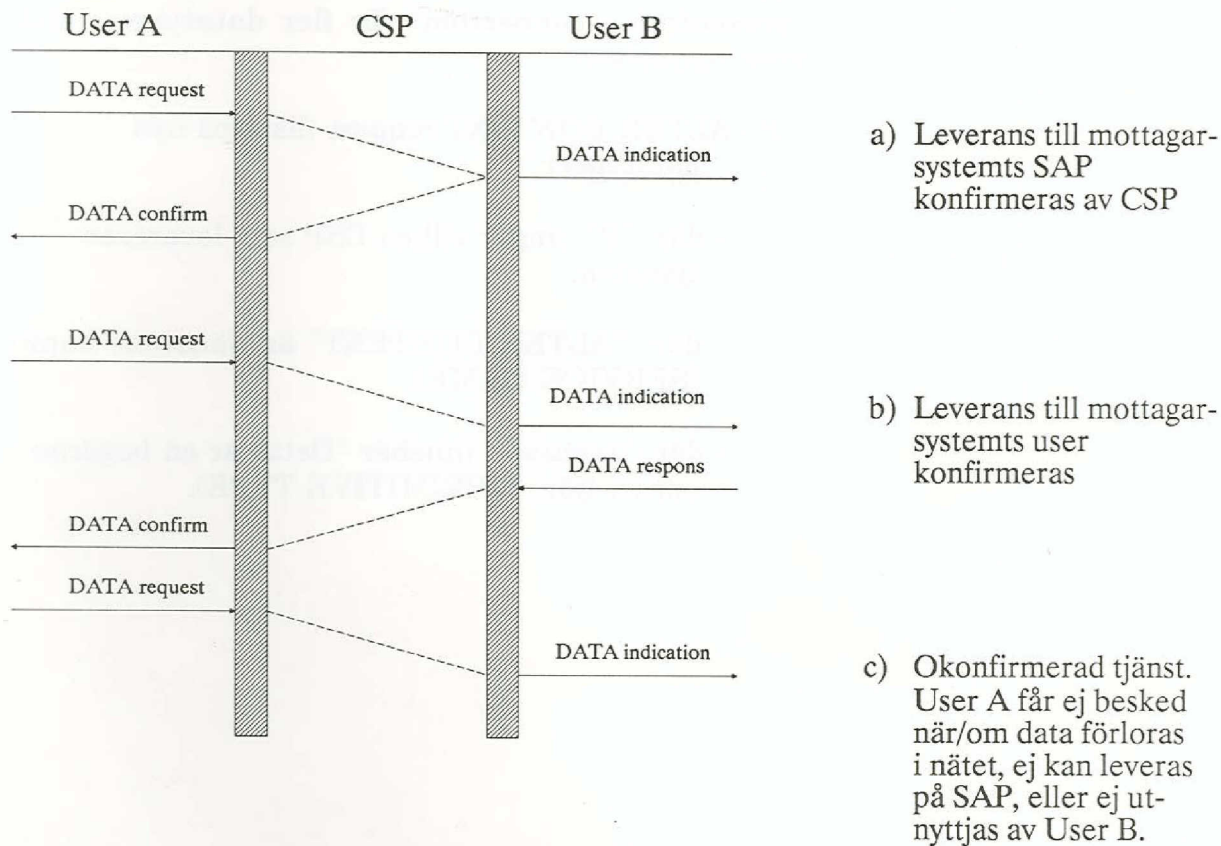


Figur 5.4.

Den teknik som figur 5.4 visar används allt oftare i service definitions. De vertikala linjerna representerar det abstrakta interfacet hos respektive user. Man utnyttjar en notation som är ortogonal i förhållande till det vanliga sättet att rita OSI-arkitekturer. Fördelen gentemot det tillståndsorienterade synsättet är att man bättre kan åskådliggöra förloppens sekventiella egenskaper, med hänsyn tagen till alla users och service provider. Ibland utnyttjas också mellanrummet mellan de vertikala linjerna för att indikera vilka PDU-er i providerns översta skikt som en viss primitiv ger upphov till. Tekniken lämpar sig däremot inte för att åskådliggöra alla tillåtna händelsekvenser - speciellt inte för mer komplicerade CSP eller när man vill bygga en modell för simuleringsändamål.

Figur 5.4 visar att händelsesekvenser för en OSI service provider (som i det här fallet erbjuder en connectionorienterad tjänst) kan indelas i faser. En fas kan erbjuda ett antal olika typer av tjänster (t.ex. "DATA transfer over connection" och "RESET of connection state"). Varje sådan tjänst kan dessutom erhållas ett godtyckligt antal gånger så länge denna connection finns tillgänglig.

Tjänsten definieras indirekt av ett antal ASP'ar (t.ex. DATA request, -indication, -respons och -confirm). Tjänster klassas också efter om de är CONFIRMED eller UNCONFIRMED:



Figur 5.5.

Som en del av OSI RM ingår också en formalism för att beteckna ASP'ar på olika nivåer i OSI-hierarkin, se /ISO 85e/ och /CCITT 84b/. Här användes beteckningarna:

- Ph = Physical Layer
- DL = Data Link Layer
- N = Network Layer
- T = Transport Layer
- S = Session Layer
- P = Presentation Layer
- ... = Beroende på aktuell tjänst i skikt 7. "A" och "C" utnyttjas för ACSE.

Eftersom det är CSP som "erbjuder" tjänsten betecknas ASP'ar efter denna. T.ex., en komponent i applikationskiktet, som i datafas vill utöka en "presentation connection" för fler datatyper, initierar händelsen:

P - ALTER_CONTEXT request (lista på nya datatyper);

- där "P" anger vilken CSP som levererar tjänsten.
- där "ALTER_CONTEXT" är tjänstens namn (SERVICE NAME).
- där "request" innebär "Detta är en begäran om en tjänst (PRIMITIVE TYPE).

5.5 Standarder

Transportskiktets betydelse som "utjämnare" av de tjänster som olika typer av nät (skikt 1-3) ger, medförde att denna service definition var en av de första som standardiserades, och då i connection-oriented utförande, se /ISO 86e/ och CCITT's X.214. Denna standard utformades så att den även skulle täcka in den transporttjänst som det av CCITT skapade transportprotokollet för telematiska tjänster, CCITT T.70, gav.

Så småningom fick även nätskiktet sin service definition i /ISO 85f/ och CCITT's X.213, främst påskyndat av behovet att standardisera tjänstenivån för internet behov.

Idag finns service definitions för de flesta skikt i OSI hierarkin, antingen i preliminär eller i slutlig form. De senast tillkomna är bl.a. länkskiktets service definition, DIS 8886 (som motsvarar CCITT's X.21d - blivande X.212). Denna tjänst är tänkt för de krav som användningen av publika och storskaliga nät ställer. För LAN ingår en enklare service definition i protokollspecifikationen för LLC = Logical Link Control, ISO's DIS 8802/2. Detta är inte en riktig renlärig beskrivningsmetod, vilket visar att även beskrivningsmetodiken inom OSI är under utveckling.

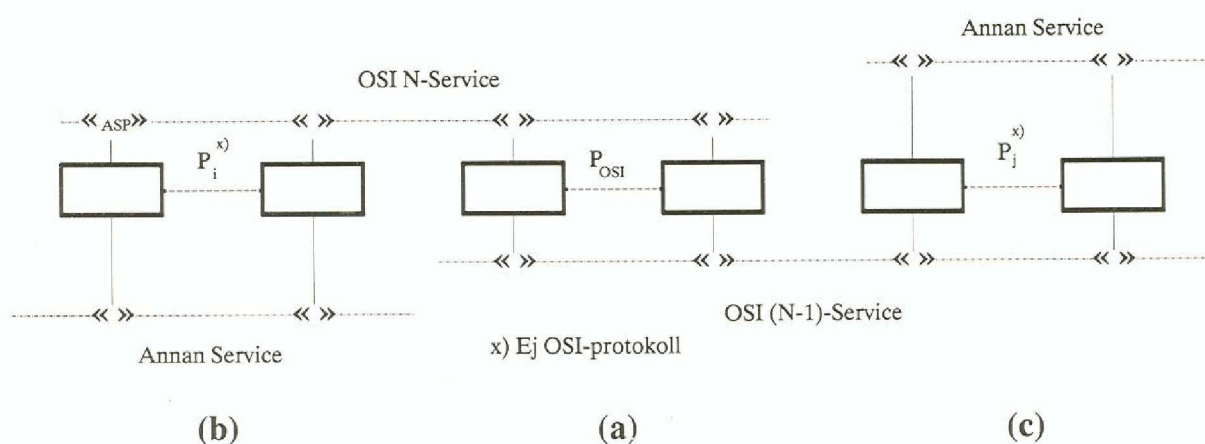
Service definition för sessionsskiktet /ISO 84b/, och CCITT's X.215, har funnits tillgänglig sedan 1984. Presentationskiktets definition, /ISO 86a/, utgavs 1986 preliminärt, och är också på väg inom CCITT som X.226.

Som framgått av det föregående finns det principiellt två sätt att överföra information via en CSP, connectionoriented eller connectionless. Båda har för- och nackdelar som inte skall diskuteras här. Man kan däremot konstatera att inom OSI har den connectionorienterade inriktningen hittills varit den dominerande (inriktad mot publika nättjänster och applikationer såsom filöverföring etc.). Connectionless börjar emellertid bli allt intressantare på alla nivåer i OSI-hierarkin. Detta har resulterat i ett antal nya standardförslag för connectionless services: ISO's DIS 8348 för nätskiktet, DIS 8072 för transportskiktet och motsvarande för presentations-

skiktet diskuteras. På länkskiktsnivå är connection-oriented och connectionless integrerade i samma standard, ISO's DIS 8802/2.

5.6 Strategier

En service definition beskriver de tjänster som en CSP erbjuder till sina användare, utan att förutsätta något om de underliggande tjänster som krävs för detta. Detta ger vissa möjligheter att tillämpa olika strategier för att blanda /anpassa olika arkitekturer till varandra, se figur 5.6:



Figur 5.6.

Figur 5.6 visar 3 typfall på hur tjänster kan utnyttjas och skapas av protokollfunktioner i en arkitektur:

- a** I figur 5.6 (a) baserar sig protokollet på en av OSI definerad tjänst och skapar en av OSI definerad tjänst på närmast högre nivå. Protokollet kan förmodas vara ett av OSI definierat protokoll.

Emellertid skulle protokollet kunna var ett icke standardiserat protokoll. Detta omöjliggör naturligtvis kommunikation i en öppen miljö. Däremot är det möjligt att tillgodogöra sig existerande protokollprodukter både under och över skikt N i denna arkitektur. Tillämpning av en sådan strategi skulle kunna baseras på att OSI vore stabilt över och under skikt N, men att existerande OSI-produkter i skikt N bedömdes som opålitliga eller ineffektiva.

- b** I figur 5.6 (b) baserar sig protokollet på en annan tjänst än OSI's. Protokollet antas emellertid transformera denna tjänst till en OSI-tjänst. Protokollet kan inte vara något OSI-protokoll. Däremot möjliggör konfigurationen att distribuerade systemfunktioner ovanför skikt N kan utvecklas i OSI-miljö.

Den här strategin användes inom den tidigare nämnda internet arkitekturen. Den har också förslagits som en "graceful transition strategy" för att föra över applikationsutveckling från en icke OSI-miljö till OSI med bibehållande av en stabil arkitektur under skikt N. I /RC 87/ diskuteras en sådan strategi för att anpassa DDN's (Arpanet) TCP-nivå till OSI's transportsikt nivå.

- c** I figur 5.6 (c) ges en tredje tänkbar strategi: Den kan vara tänkbar speciellt när man önskar utnyttja stabila OSI-tjänster, t.ex. transport tjänsten, och, i avvaktande på stabila OSI-produkter på högre nivåer, anpassa transporttjänsten till en befintlig kommunikationsarkitektur för applikationssystem. Protokollet i figur 5.6 (c) är givetvis inte någon OSI-standard.

5.7 Summering

Avslutningsvis kan man konstatera att tekniken med realiseringsoberoende service definitions har hög allmängiltighet, d.v.s. långt utanför OSI's skikt-specifika standarder. ASP-begreppet är fundamentalt och har en solid grund. ISO standardiserar också ett händelseorienterat språk (LOTOS), som ägnar sig väl för icke grafisk formell specifikation av tjänster. De grafiska teknikerna och den ASP-notation som beskrivits här är inte bundna till OSI's skiktindelning och lämpar sig lika väl i andra sammanhang. Dock tenderar OSI's notationsteknik ibland att resultera i väl omfattande, övertydliga och därmed svårlästa dokument.

6 OSI's protokoll

6.1 Inledning

Avsikten med detta kapitel är inte att beskriva specifika OSI-protokoll utan att allmänt belysa protokollbegreppet i OSI:

I avsnitt 6.2 diskuteras protokollbegreppet speciellt i relation till interfacebegreppet.

I avsnitt 6.3 och 6.4 avhandlas skiktens och entiteters struktur med tyngdpunkt på applikations-skiktet.

I avsnitt 6.5 diskuteras ett för alla protokoll gemensamt problem, kodning av PDU'er. Speciellt beskrivs den kodningsteknik som presentationsskiktet står för.

I avsnitt 6.6 behandlas OSI's protokollhierarki i sin helhet. Särskilt belyses relationerna mellan protokoll i hierarkin.

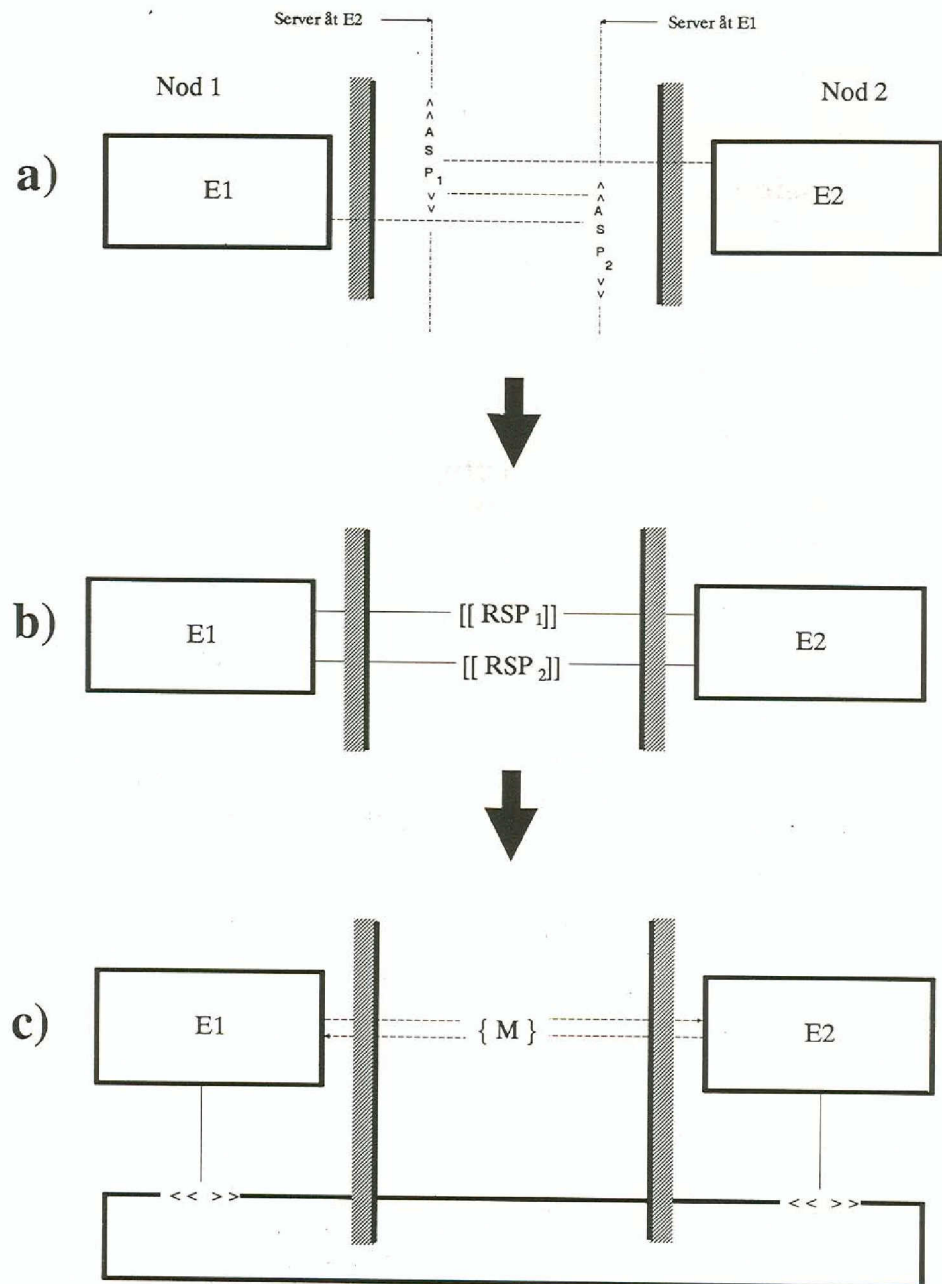
I avsnitt 6.7 slutligen diskuteras hur protokoll beskrivs. I avsnittet diskuteras också problemet att, givet ett visst applikationsområde, välja protokoll ur protokollhierarkin.

6.2 Vad är ett protokoll?

Begreppet protokoll är inte OSI specifikt. OSI's definition är ungefär:

"A set of rules and formats (syntactic and semantic) which determines the communication behaviour of peer entities in the performance of layer functions".

I det föregående diskuterades sambandet mellan det konkreta, lokala interfacet som utgör realisering av det abstrakta interfacet. Det konstaterades också att det ur nätets eller ett distribuerat systems synvinkel var ointressant hur det lokala interfacet realiserades, så länge som users var överens om vilka händelser som det konkreta interfacet skulle avspegla. I en distribuerad miljö förhåller det sig naturligtvis precis tvärtom:



{M}: Mängden av PDU'er i protokollet
 PDU: Protocol Data Unit ("meddelanden")

Figur 6.1.

Figur 6.1 antyder ett alternativt sätt att definiera protokollbegreppet (som är författarens synsätt, snarare än det av OSI vedertagna):

"Ett protokoll och en kommunikationstjänst utgör en realisering av ett abstrakt interface". De meddelanden (PDU=PROTOCOL DATA UNIT) som definieras i ett protokoll beskrivs i en PROTOCOL SPECIFICATION. Denna kan sägas beskriva hur händelseinformation skapas och överförs mellan parterna i form av PDU'er. På samma sätt som för ASP'ars realisering, kan man förvänta att det ofta krävs mer än en PDU för att överföra information om en händelse mellan parter i en protokollmaskin.

Om det inte vore så att

- a den CSP som entiteterna utnyttjar var behäftad med brister och begränsningar, såsom ändlig överföringstid, tappar meddelanden, levererar meddelanden i annan ordning än de tagits emot, endast klarar en begränsad datamängd per PDU, m.m.
- b och noderna där entiteterna opererar vore behäftade med brister och begränsningar i form av ändlig processeringstid, begränsad buffertkapacitet, felaktig kodning och tolkning av data samt dessutom kraschar då och då så skulle i princip varje PDU kunna innehålla information motsvarande en händelse mellan parterna, och inga andra PDU'er skulle behövas.

Verkligheten är emellertid bistrare än så. En protokollspecifikation innehåller därför oftast en mängd PDU'er, PDU-sekvenser, kontrollparametrar samt tidövervaknings-, flödeskontroll- och sekventeringsmekanismer, vars huvudsakliga syfte är att överbrygga nämnda brister. Speciellt gäller detta protokoll i skikt 2 (länkskiktet) vars händelsebeskrivning skulle bli nästintill trivial om det inte vore för länkskiktets huvudsyfte: att transformera en otillförlitlig fysisk länk till en (nästintill) felfri bitlänk.

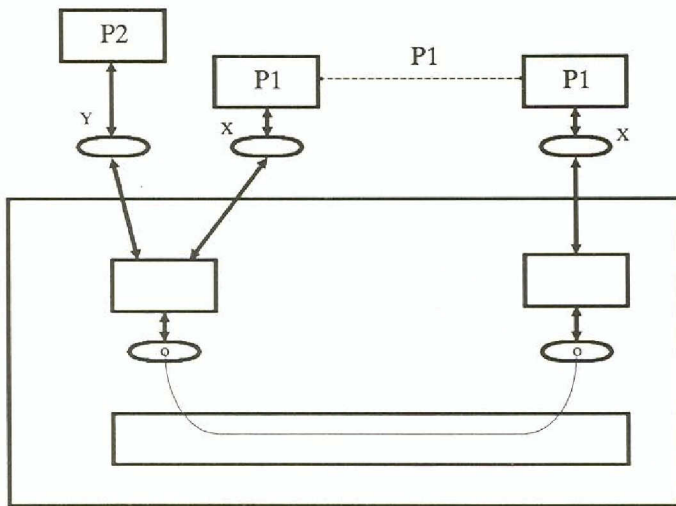
Även protokoll på skikt 3 och 4 är i varierande utsträckning drabbade av brister och begränsningar i underliggande tjänster. Däremot betraktar man

transporttjänsten som en nära idealisk "bitpipe" mellan endsystem. Dock måste protokoll på högre nivåer räkna med totalt avbrott i förbindelsen, orsakat av fel i nät och endsystem. Sådana problem rapporteras då av transporttjänsten medelst

- T - DISCONNECT indication (disconnect reason param.);
där orsakerna vid onormala disconnect (enligt gällande standard /ISO 86e/) redovisas som
- lack of local or remote resources of the T-service provider
 - QOS below minimum (QOS är den utlovade kvaliteten på en T-connection).
 - Misbehaviour of T-service provider (t.ex. ett avbrott på nätnivå eller lägre).

6.3 Strukturering av protokoll

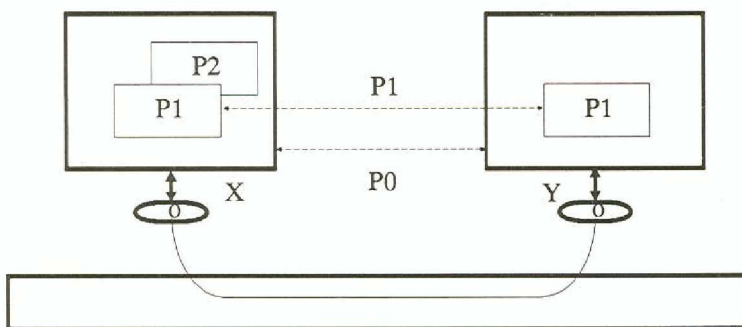
I ett skikt kan det finnas alternativa protokoll. Därav följer att i en öppen miljö kan noder innehålla olika protokollkomponenter, vilket kan försvåra möjligheterna att kommunicera. OSI exemplifierar 3 olika sätt att lösa det här problemet, se figur 6.2:



Olika protokoll associeras till olika SAP

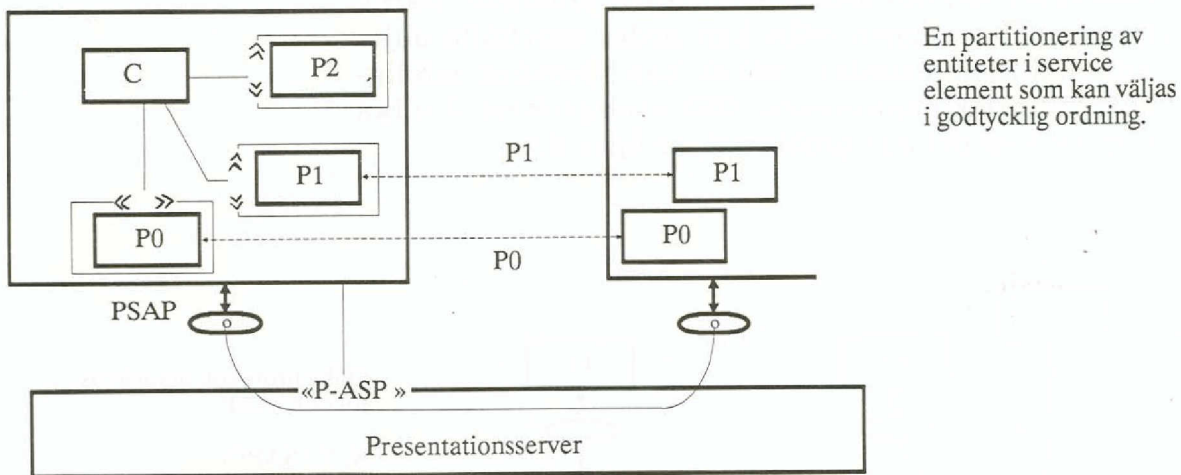
X, Y : SAPI's

Figur 6.2.a.



En hierarkisk partitionering av entiteters protokoll i en valdel (P0) och antal protokollgrenar.

Figur 6.2.b.



Figur 6.2.c.

Modellen enligt figur 6.2 (a) är inflexibel och låser accesspunkter till funktioner. Den utnyttjas när adressering inte behövs för att peka ut alternativa system /noder och när allokeringstransparens inte eftersträvas. Ett exempel är ISDN's skikt 2 protokoll, där alternativa accessprotokoll på skikt 3 pekas ut med hjälp av skikt 2's SAP'ar.

Metoden enligt figur 6.2 (b) kan utnyttjas för att peka ut alternativa protokoll när systemet/noderna ingår i mer än en arkitektur (jämför med figur 5.6). Detta är attraktivt speciellt i flertjänstnät (ISDN), där man kan tänka sig att understödja såväl internationellt standardiserade paketprotokoll (X.25, X.75) som nationella varianter.

Den gemensamma delen av entitetens protokoll i figur 6.2 (b), "P0", utgörs vanligen av en parameter i den första PDU'n som utväxlas. Denna parameter kallas i OSI RM för PROTOCOL IDENTIFIER (CCITT använder i ISDN-rekommendationerna begreppet "protocol discriminator").

Metod (c) i figur 6.2 representerar den syn på applikationsentitetens interna struktur som OSI har. I lägre skikt är det inte nödvändigt att studera entitetens interna struktur. En protokoll specifi-

kation specificerar hela entiteten. En applikations-entitet kan emellertid vara betydligt mer komplext uppbyggd, t.ex. av en mängd standardiserade service element, tillsammans med applikations-unika delar. Det kan inte heller förutsättas att service elementen används i någon specifik ordning, såsom förutsättningen var i fall (b).

Följaktligen uppstår det ett synkroniseringsbehov mellan entiteterna i figur 6.2 (c) med avseende på vilket service element (och därmed del av entitetsprotokoll) som skall gälla i varje tidsögonblick under en session. OSI tänker sig att det finns ett koordinerande element. "c" i figuren, som avgör detta. Detta element utnyttjar i sin tur ett standardiserat element (enligt "P0" i figuren) för val och förhandling av APPLICATION CONTEXT. Den här tjänsten, som kan kallas "application context switching", och motsvarande protokoll förväntas bli en integrerad del i CASE Kernel (ACSE) enligt /ISO 86c/ och /ISO 86d/.

Övriga serviceelement i figuren standardiseras i den utsträckning de har allmänt intresse. De klassas då som ASE. Exempel på sådana är

FTAM - File Transfer, Access and Management,
enligt DIS 8571/1 från 1986.

JTM - Job Transfer and Manipulation, enligt DP
8831 och DP 8832 från 1985.

VT - Virtual Terminal, enligt DIS 9040 och DIS
9041 från 1986.

Även MHS = Message Handling Services hör hit. Denna tjänst och dess protokoll utvecklades emellertid före tillkomsten av ACSE/ASE synsättet och inkluderar därför en egen CASE Kernel funktion (RTS = Reliable Transfer Service) enligt DP 9066 och X.410 från 1985 respektive 1984. MHS-standarderna är mest kända från CCITT's rekommendationer i X.400 serien från 1984. ISO's motsvarigheter är från 1986: DIS 8505, DIS 8883 och DIS 9065.

6.4 Service Element i applikationsskiktet

PDU'er i sessions-, presentations- och applikationsskiktet kan i högre grad än i lägre skikt representera de för respektive skikts funktioner relevanta händelsesekvenserna. På applikationsnivå är naturligtvis målsättningen att kommunicerande entiteter skall vara helt befriade från alla "triviala" kommunikationsproblem och helt kunna koncentrera sig på det "egentliga jobbet". Emellertid kvarstår på denna nivå två problem, även om presentationstjänsten är aldrig så perfekt:

- a Att hitta den PSAP (SAP'ar i presentationskiktet) där den partner som man vill kommunicera med finns. Ur en applikationskomponents synvinkel är det bl.a. önskvärt att en partner kan anropas med namn (APPLICATION TITLE).

Det behövs därför en mekanism i skikt 7 som kan skapa ASSOCIATIONER mellan namngivna entiteter i applikationsskiktet, så att distribuerade applikationssystem i övrigt kan betraktas som nivå 3 system, enligt diskussionen i kap. 4. OSI anvisar två metoder för detta. Den ena ingår i ACSE tjänsterna och dess protokoll från 1986, /ISO 86c/ respektive /ISO 86d/ (kallas där CASE Kernel). OSI anvisar också en fristående katalogtjänst, som primärt inriktar sig på den expanderande sektorn för "electronic mail". ISO's standarder för en sådan DIRECTORY SERVICE utgavs 1986 som förslag, DP 9594/1-7. Motsvarande CCITT förslag kallas fn X.ds 0 - X.ds 6.

- b Att nöjaktigt kunna hantera distribuerade transaktioner. En transaktion som inbegriper åtgärder vid mer än en nod i ett distribuerat system, t.ex. en kreditering vid en banks dator och motsvarande debitering vid en annan banks, kräver att speciella "recovery" åtgärder vidtas om inte båda subtransaktionerna kan utföras. Transaktionen i sin helhet skall vara "atomär". Orsaken till att den distribuerade transaktionen inte kan genomföras kan vara systemkrascher eller partitionering av underliggande nät. Ett protokoll som kan hantera det här problemet kallas för ett "commitment protokoll".

Ett annat krav är att transaktionen skall vara skyddad från interfererande transaktioner tills dess att (i det här fallet) debiteringen har utförts. Protokoll som kan hantera det här problemet i en miljö av distribuerade data kallas för "locking protokoll".

OSI anvisar en lösning till problemet i form av den andra delen av ACSE-standarden: CCR = Commitment, Concurrency, Recovery. CCR-protokollet är emellertid än så länge enbart ett commitment protokoll enligt den så kallade "2-phase commitment" tekniken. ACSE CCR tänkes liksom CASE Kernel utgör ett SERVICE ELEMENT i skikt 7, tillgängligt för applikationsunika komponenter. ISO's standarder från 1985 för CCR är /ISO 85c/ (service definition) och /ISO 85d/ (protocol specification).

6.5 Presentationsproblemet i OSI

Utformning av kommunikationsprimitiver för det abstrakta interfacet i en nod är uppenbart en lokal fråga för noden. En lokal primitiv skulle kunna vara en IPC-primitiv av typ

```
SEND <CONNECT request(parametrar)> TO CSP;
```

eller det tidigare nämnda proceduranropet

```
CSPCALL OPEN (IN: request par.OUT: confirm par.);
```

Sådana primitiver måste givetvis vara kända av de lokalt kommunicerande komponenterna och deras OS, både med avseende på deras syntaktiska och semantiska innebörd och med avseende på de datatyper som parametervärdena representerar. I ett protokollinterface gäller liknande krav:

Händelseinformationen, d.v.s. information om typen av händelse samt dess parametrar, överförs här som data i meddelanden (PDU'er). Det krävs då en exakt överenskommelse mellan kommunicerande entiteter med avseende på PDU'ers parameterstruktur och deras representation. I ett modernt programmeringsspråk för homogena datamiljöer räcker det för konstruktören att definiera variabelers typ. Representation av datatyper (kodning och avkodning på bitnivå) är operativsystemets sak. I en fysiskt distribuerad miljö finns emellertid (normalt) inte något globalt operativsystem, varför kodning av PDU'er vanligen utgör en integrerad del av en protokollspecifikation.

På lägre nivåer i OSI-hierarkin, där antalet protokoll per skikt är få och effektivitet ett primärt krav, optimeras kodningen oftast efter det aktuella protokollets behov. En parameter som bara kan ha 4 olika värden behöver endast 2 bitar, d.v.s. 1/4 octett i en PDU. Det är den sändande entitetens sak att packa dessa bitar på överenskommet sätt i PDU'er och den mottagande entitetens sak att packa upp dem.

I applikationsskiktet är emellertid antalet möjliga tillämpningar och deras (unika) protokoll oändligt.

Det måste därför betraktas som den mest elementära och kanske viktigaste stödfunktionen åt applikationssystem att OSI kan erbjuda en standardiserad kodning av de datatyper som kan ingå i PDU'er på denna nivå. Detta är bakgrunden till presentationsskiktet, skikt 6 i OSI-hierarkin.

Presentationstjänsten är så utformad att en applikationsentitet med primitiven

P - CONNECT request (list of ABSTRACT SYNTAX NAMES);

talar om för den lokala P-skikts entiteten vilka datatyper (d.v.s. PDU-data) som skall kunna överföras över önskad P-connection, när den väl är etablerad. Applikationsentiteten definierar sina datatyper i form av "abstract syntax names", enligt ett datatypande språk, som speciellt utvecklas för detta ändamål. Språket kallas ASN.1 = Abstract Syntax Notation, och beskrivs i /ISO 85a/. ASN.1 utnyttjas numera allmänt både av ISO och CCITT vid utformning av protokoll specifikationer. Intressant att notera är att ASN.1 lika väl kan användas för att specificera lägre skikts protokoll, även om kodningen är unik för protokollet. T.ex. presentationskiktets egna PDU'er är beskrivna i ASN.1.

OSI's presentationsprotokoll (i korthet) är uppbyggt kring ett standardiserat sätt att koda varje enskild ASN.1 typ. ASN.1 omfattar såväl i språket inbyggda typer som ur dessa, av användaren konstruerade typer. En presentations-PDU överför såväl kodade data som typinformation för varje set av kodade applikationsdata. Mottagande presentationsentitet kan då identifiera varje typ och transformera datats representation till den lokalt gällande.

Kodningsalgoritmerna kallas för BER = Basic Encoding Rules, och beskrivs av /ISO 85b/. Motsvarande CCITT rekommendation är X.409, som emellertid beskriver både ASN.1 och BER invända i varandra. Ett eventuellt framtida önskemål om att kunna kombinera ASN.1 med andra kodningsalgoritmer än den som BER ger (bl.a. på grund av att BER ofta skapar en ineffektiv kod), har fått ISO att skilja standarder för att beskriva datatyper från kodningsregler.

OSI's presentationsprotokoll och tjänst är avsevärt mer komplicerad än vad utrymmet här medger att beskriva. Intresserade läsare rekommenderas att ta del av presentationsproblemet såsom det beskrivs i /SK 87/, kap 11.

6.6 Protokoll och arkitekturer

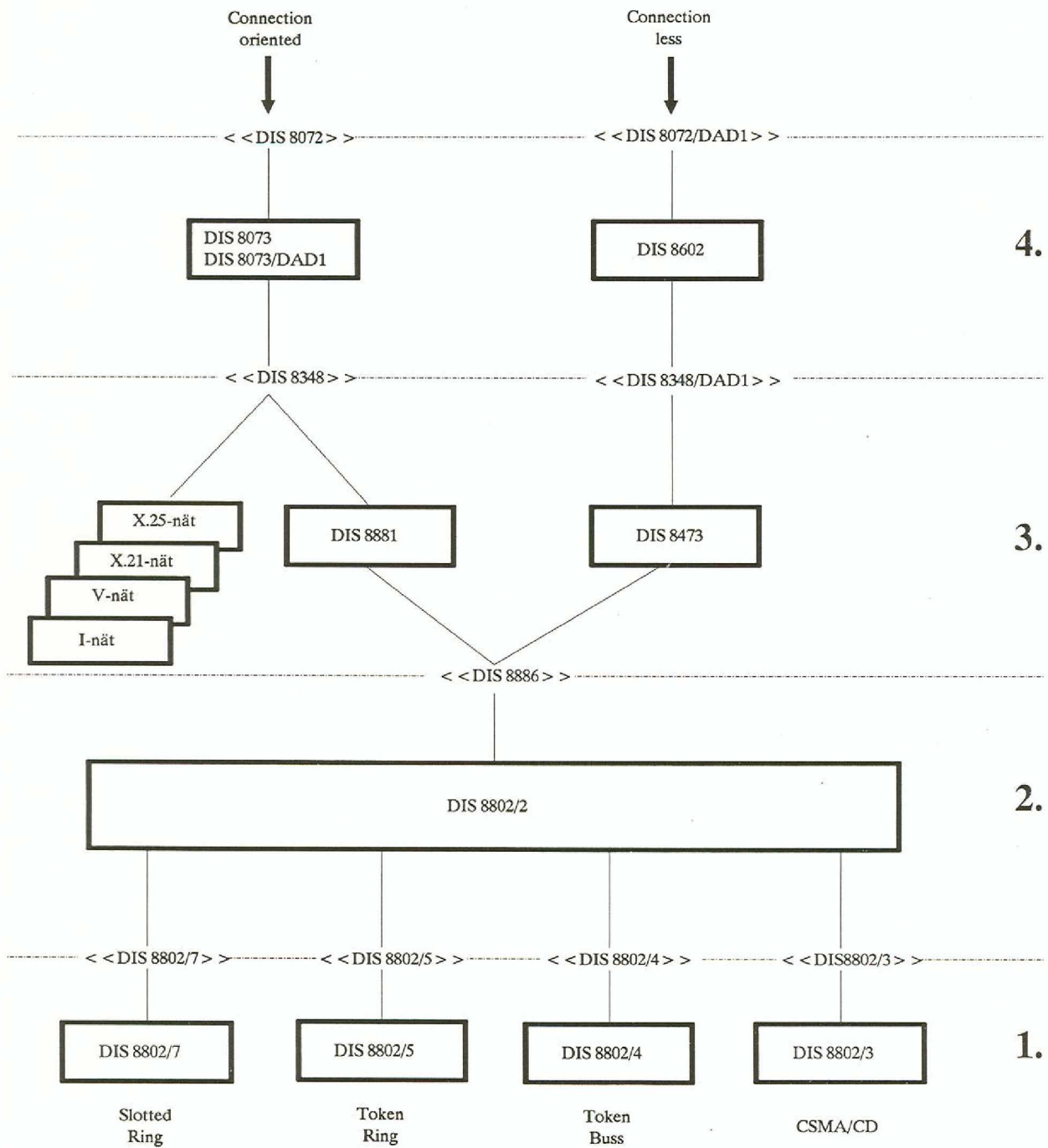
OSI's uppsättning av protokoll omfattar idag samtliga 7 skikt även om många ISO-standarder är av DIS- och DP-karaktär (med DIS avses "Draft International Standard", med DP "Draft proposal"). Inom skikt 1-3 såväl som skikt 7 finns dessutom ett flertal alternativa protokoll. Om man prickar in OSI's protokoll bredvid varandra i OSI's 7-nivåers "bokhylla", erhålles bilden av "the OSI wineglass", smalt på mitten (skikt 4-6) och fullt med protokoll i skikt 7 (ACSE och ASE) och i skikt 1-3 (ett resultat av förkomsten av olika medier och nätteknologier).

En sådan bild kan sägas utgöra "the OSI Protocol Suite", vars enda syfte är att vara underlag för urval av mer begränsade protocol suites, t.ex. för ett visst applikationsområde, en viss region eller marknad eller en viss typ av tjänst.

Tidigare i rapporten har det fastslagits att ett visst protokolls utformning beror av två faktorer:

- a Den kommunikationstjänst som protokollet förutsätts använda sig av
- b Den kommunikationstjänst som protokollet avser att leverera.

Detta faktum innebär att OSI protokoll i närliggande skikt blir relaterade till varandra via de abstrakta tjänstebeskrivningarna. Bilden av OSI protokoll är alltså inte bara en "bokhyllemodell" av protokoll utan snarare en hierarki eller ett nätverk av protokoll där relationerna utgörs av service definitions, se figur 6.3.



Figur 6.4.

(Figur 6.4 visar att media access interface mellan skikt 1 och 2 har beskrivits i samma standarder som skikt 1 protokollet, ett arv från IEEE snarare än en OSI metodik).

Figur 6.4 antyder också en mängd olika protokoll på nätskiktet. T.ex DIS 8881 skapar en connection-oriented tjänst åt transportskiktet för LAN's, DIS 8473 en connectionless. Kan då inte båda varianterna integreras i ett protokoll? I princip ja. T. ex. Länkprotokollet LLC erbjuder båda typerna av tjänster. OSI är emellertid en öppen modell som hela tiden utvecklas. Skälen till uppdelning på olika protokoll är därför oftast historisk. Framförallt har det faktum att OSI startade connectionoriented men numera kompletteras med connectionless på alla nivåer, bidragit till det nätverk av beroenderelationer som idag finns i arkitekturen.

Detta understryker betydelsen av dels fristående service definitions (de utgör länkarna i detta nätverk) dels att hålla åtskilt referensmodellens skiktdefinitioner, se kap 4, från de exakta service definitionerna, se kap 5.

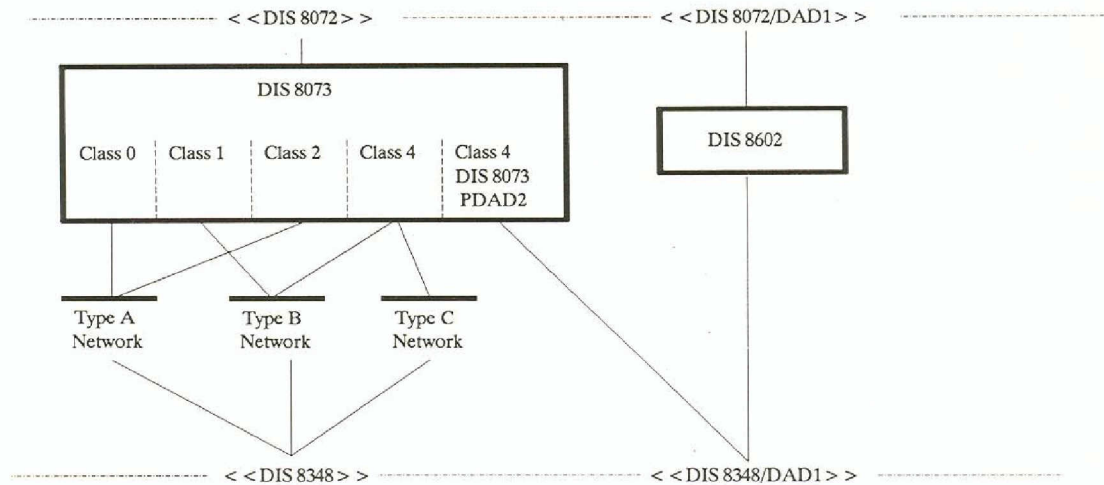
6.7 Summering

Protokollspecifikationer innehåller protokollets mekanismer, PDU'er och deras kodning (för andra protokoll än applikationskiktet) samt PDU-sekvenser, oftast i form av ett tillståndsdigram för de entiteter som ingår i "protokollmaskinen". Dessutom anges den tjänst som protokollet bygger på och man beskriver hur denna tjänst utnyttjas. Speciellt gäller det tjänster och parametrar i ASP'ar som inte är obligatoriska att realisera. Om den tjänst som protokollet leverar på något sätt avviker från den service definition som protokollspecifikationen refererar till, måste givetvis även sådana avvikelser anges.

En service definition är abstrakt även i den meningen att den inte faställer en viss kvalitet på levererade tjänster (t.ex. viss maximal uppkopplingstid). I praktiken innebär det att valet av en specifik protokollhierarki (t.ex. för MAP) inte enbart kan göras efter service definitioner. T.ex. en "broadcasttjänst" över mutiple access nät (LAN's och satellitnät) blir alltid billigare och effektivare än motsvarande tjänst över ett kretskopplat datanät, hur avancerat protokoll man än har för det kretskopplade fallet. Att definiera en "protocol suite" för en viss typ av tjänst eller applikationsområde innebär därför också en djupare analys och ett urval från samtliga OSI-skikt (inklusive medier), för att effektivitets- och kostnadskrav skall kunna uppfyllas. Ett sådant urval kallas ofta för "functional set" eller "stack".

Vissa kvalitetskrav, företrädesvis felsannolikhet, kan däremot tillgodoses av ett enda protokoll. Det är bl.a. detta man försöker åstadkomma i OSI's transportprotokoll, som beskrivs i /ISO 86f/.

Skikt 5



Figur 6.5.

I transportskiktet tar man hänsyn till förekomsten av nät av 3 kvalitetsklasser, A, B och C. C är den mest otillförlitliga typen, jämförbart med en network connection över ett analogt publikt nät. Ett klass 4 protokoll blir uppenbarligen mycket mer komplext än ett klass 0 eller 2 protokoll. Bl.a. måste protokollet kunna upptäcka och rätta till bitfel som orsakats av nätets låga kvalitetsgrad.

En implementering av OSI's transportprotokoll behöver inte omspanna alla klasser. T.ex. CCITT's transportprotokoll T.70 för de telematiska tjänsterna, innehåller bara klass 0. Däremot krävs att kommunicerande transport entiteter kan upplysa varandra om vilka klasser de klarar.

Referenser

- /BRI 85/** Ed Brinksman: A Tutorial on LOTOS. Protocol Specification, testing and Verification, V. Proceedings of IFIP WG 6.1, Fifth International Workshop. 1985.
- /CCITT 84a/** Reference Model of Open Systems Interconnection for CCITT Applications. X.200, 1984.
- /CCITT 84b/** Open Systems Interconnection (OSI) Layer Service Definition Conventions. X.210.
- /CN 84/** Computer Networks: Special Issue on Programming Languages and Open Systems Interconnection. Vol 18, nr 1, febr 1984. (Se speciellt "FORTRAN Interface to X.25 Transport Service", av G. Andreoni).
- /ISO 84a/** Information Processing- Open Systems Interconnection - Basic Reference Model. ISO 7498, 1984.
- /ISO 84b/** Basic Connection-Oriented Session Service Definition. DIS 8326, 1984.
- /ISO 84c/** Basic Connection-Oriented Session Protocol Specification. DIS 8327, 1984.
- /ISO 85a/** Specification of Abstract Syntax Notation One (ASN.1). DIS 8824/2, 1985.
- /ISO 85b/** Specification of Basic Encoding Rules for ASN.1. DIS 8825/2, 1985.
- /ISO 85c/** Definition of Common Application Service Elements (CASE). Part 3: Commitment, Concurrency and Recovery. DIS 8649/3, 1985.
- /ISO 85d/** Specification of Protocol for Common Application Service Elements (CASE), Part 3: Commitment, Concurrency and Recovery. DIS 8650/3, 1985.
- /ISO 85e/** Information Processing- Open Systems Interconnection-Service Conventions. DTR 8509.
- /ISO 85f/** Information Processing-Data Communications. Network Service Definition. DIS 8348, 1985.
- /ISO 86a/** Connection Oriented Presentation Service Definition. DIS 8822, 1986.
- /ISO 86b/** Connection Oriented Presentation Protocol Specification. DIS 8823, 1986.

- /ISO 86c/* Service Definition for Common Application Service Element, Part 2: Association Control. DIS 8649/2, 1986.
- /ISO 86d/* Protocol Specification for Protocols for Common Application Service Elements, Part 2: Association Control. DIS 8650/2, 1986.
- /ISO 86e/* Information Processing Systems- Open Systems Interconnection. Transport Service Definition. ISO 8072, 1986.
- /ISO 86f/* Information Processing Systems- Open Systems Interconnection. Connection Oriented Transport Protocol Specification. ISO 8073, 1986.
- /RC 87/* Marshall T. Rose, Dwight E. Cass. OSI Transport Services on Top of the TCP. Computer Networks and ISDN Systems, 12: 159 - 173, 1987.
- /SK 87/* Morris Sloman and Jeff Kramer. Distributed Systems and Computer Networks. Prentice - Hall, 1987.
- /TA 81/* Andrew Tanenbaum. Computer Networks. Prentice - Hall, 1981.