

**MASTER**

**SNA, OSI en X.25**

**SNA in een OSI-licht en SNA/X.25 protocol oonversie**

van den Ouweland, C.J.P.M.

*Award date:*  
1988

[Link to publication](#)

**Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Technische universiteit Eindhoven  
Faculteit der Elektrotechniek  
Vakgroep Digitale Systemen (EB)

*Embargo  
tot 1991*

## **SNA, OSI en X.25**

---

**door C. J. P. M.**

**van den Ouweland**

**SNA in een OSI-licht**

**en**

**SNA/X.25 protocolconversie**

Verslag van het afstudeerproject verricht  
onder verantwoordelijkheid van:  
prof. ir. M.P.J. Stevens, Technische Universiteit Eindhoven  
ir. F. Velthuis, Alcatel Nederland B.V.

Den Haag, 26 juli 1988.

De afdeling Elektrotechniek van de Technische Universiteit  
Eindhoven aanvaardt geen enkele verantwoordelijkheid  
aangaande de inhoud van afstudeerverslagen.

## SAMENVATTING

SNA en OSI zijn twee netwerkkarchitecturen die een aantal elementen, zowel wat betreft doel als invulling daarvan, gemeen hebben, maar ook voor een aanzienlijk deel wezenlijk verschillend zijn. Behalve vergelijken, kunnen we proberen samenwerking te bewerkstelligen, door bepaalde denkbeelden van het ene model te integreren in het andere model, maar ook door de verschillen te accepteren en gateway-achtige functies te specificeren.

Het is mogelijk een op SNA gebaseerde infrastructuur te definiëren, die van OSI het idee adopteert een scheidslijn te trekken tussen de netwerk service en de op het netwerk aangesloten DTE's.

Voor de realisering van dit netwerk leggen we op de tweede manier een link tussen SNA en OSI: we integreren X.25 in het SNA netwerk. Hiervoor zijn protocolconverters nodig, die we PCU's noemen. Deze PCU's vervullen tal van functies die benodigd zijn om X.25 op een efficiënte wijze te integreren, onder meer het opzetten, beheren en afbreken van X.25 calls op grond van verkeersaanbod. Bovendien hebben de PCU's een taak m.b.t. error-recovery.

## INHOUD

HOOFDSTUK 1	INLEIDING	
HOOFDSTUK 2	SNA - EEN VERGELIJKING MET DE UITGANGSPUNTEN VAN OSI	
2.1	UITGANGSPUNTEN . . . . .	6
2.2	HISTORIE . . . . .	7
2.3	SNA GETEST AAN DE OSI-UITGANGSPUNTEN . . . . .	10
HOOFDSTUK 3	EEN SNA-INFRASTRUCTUUR	
3.1	DOELSTELLING . . . . .	16
3.2	HET GEBRUIKERSSTANDPUNT . . . . .	16
3.3	HET SNA-STANDPUNT . . . . .	17
3.3.1	Gateways . . . . .	17
3.3.2	Waarom gateways? . . . . .	18
3.3.3	Globale structuur . . . . .	19
3.4	DE ONDERLIGGENDE NETWERKSTRUCTUUR . . . . .	20
3.4.1	X.25 onderbouw met PVC's . . . . .	21
3.4.2	Protocol Conversie met PCU's . . . . .	21
3.4.3	Samenvatting van alle functies van de PCU . . . . .	24
3.4.3.1	Conversie SDLC naar X.25 . . . . .	25
3.4.3.2	Vertalen SNA-adres naar X.25 adres voor data van SNA-node naar netwerk . . . . .	25
3.4.3.3	Inpakken SNA data in X.25 informatie-pakketten	26
3.4.3.4	Uitpakken van X.25 informatie-pakketten en reproduceren SNA-pakket . . . . .	26
3.4.3.5	Concentratie en expansie . . . . .	26
3.4.3.6	Local Switching . . . . .	27
3.4.3.7	Calls opzetten en afbreken . . . . .	27
3.4.3.8	Bijhouden van boekhouding van calls en bekende SNA/X.25 adresparen . . . . .	27
3.4.3.9	Error recovery . . . . .	28
3.4.4	ACD (Address Conversion Database) . . . . .	28
3.4.5	Noodzaak van adresconversie . . . . .	29
HOOFDSTUK 4	SPECIFICATIE PCU	
4.1	SPECIFICATIE VANUIT HET GEZICHTSPUNT VAN DE GEBRUIKER . . . . .	31
4.1.1	Strekking van dit gezichtspunt . . . . .	31
4.1.2	Configuraties . . . . .	32
4.1.2.1	SDLC . . . . .	32
4.1.2.2	QLLC . . . . .	33
4.1.2.3	PCU . . . . .	33
4.1.3	Kostenaspect . . . . .	34
4.1.3.1	Uitgangspunten . . . . .	34
4.1.3.2	Kostenberekening met PVC's . . . . .	35
4.1.3.3	Kostenberekening met PCU's . . . . .	35

## Inhoud

4.1.3.4	Afweging PVC/PCU . . . . .	36
4.1.3.5	Internationale aspecten . . . . .	37
4.1.4	Configuraties met de PCU . . . . .	37
4.1.5	Vertragingstijden in het netwerk . . . . .	39
4.2	INWENDIGE SPECIFICATIE . . . . .	41
4.2.1	Blokschema . . . . .	42
4.2.2	Lagenmodellen . . . . .	42
4.2.3	De lokale functie . . . . .	46
4.2.3.1	SDLC . . . . .	46
4.2.3.2	QLLC . . . . .	49
4.2.4	De netwerk functie . . . . .	50
4.2.4.1	EFTP . . . . .	50
4.2.4.1.1	Headerloos protocol . . . . .	51
4.2.4.1.2	Call management in EFTP . . . . .	51
4.2.4.1.3	Error recovery in EFTP . . . . .	52
4.2.4.2	Compatibiliteit met QLLC . . . . .	52
4.2.5	PCU beheer . . . . .	52
4.2.5.1	Afbeelden Locale kanalen op Netwerkkkanalen . .	52
HOOFDSTUK 5	ERROR FREE TRANSPORT LAYER PROTOCOL	
5.1	UITGANGSPUNTEN . . . . .	54
5.2	FUNCTIONALITEIT VAN DE TRANSPORT LAAG . . . . .	56
5.3	PROCEDURES VOOR HET OPSTARTEN VAN EEN T-CONNECTION	57
5.4	PROCEDURES VOOR HET DOORGEVEN VAN USERDATA . . . . .	58
5.4.1	Acknowledgement . . . . .	58
5.4.2	More Bit . . . . .	59
5.4.3	Qualified TSDU's . . . . .	59
5.4.4	Expedited data . . . . .	59
5.5	PROCEDURES VOOR ERROR RECOVERY . . . . .	60
5.5.1	N-Reset . . . . .	60
5.5.2	N-Clear-Ind . . . . .	60
5.6	PROCEDURES VOOR HET BEËINDIGEN VAN DE T-CONNECTION	
61		
5.6.1	T-Disc-Req-collision . . . . .	61
5.6.2	T-Conn-Abort . . . . .	61
5.7	PROCEDURES VOOR HET BELEEFD TIJDELIJK BEËINDIGEN VAN	
	EEN CALL. . . . .	62
5.7.1	T-PINVC-collision . . . . .	63
5.7.2	Hervatten van de call na een T-PINVC . . . . .	63
5.7.3	Reconnect-Collision . . . . .	63
5.8	OVERZICHT TPDU'S MET HUN OPMAAK . . . . .	64
HOOFDSTUK 6	CONCLUSIE	
HOOFDSTUK 7	LITERATUUR	
HOOFDSTUK 8	AFKORTINGEN	

APPENDIX A      A

## HOOFDSTUK 1

### INLEIDING

In de informatiemaatschappij, die momenteel bezig is te ontstaan, zien we enerzijds steeds groeiende mogelijkheden in informatieverwerking en anderzijds een eveneens hard groeiende behoefte aan informatietransport. Openbare netwerken volgens de CCITT specificatie X.25 (lit. 6) bieden algemeen toepasbare communicatiefaciliteiten en vinden ruime opgang over de gehele wereld. Anderzijds vormt apparatuur, geënt op de IBM-mainframe (en -mini) wereld een groot deel van de informatie-verwerkingscapaciteit.

Het is dan ook zaak deze beide werelden te integreren, of althans te laten samenwerken. Genoemde IBM (-compatibele) apparatuur maakt gebruik van een communicatieconcept, genaamd SNA (Systems Network Architecture), dat sterkt afwijkt van het OSI/CCITT concept, dat de naam OSI draagt (Open Systems Interconnection).

Dit onderzoek slaat op twee niveaus een brug over deze kloof. Op het hogere niveau wordt het OSI-uitgangspunt een scherpe scheidslijn te trekken tussen netwerk en gebruikers (DTE's) ingepast in SNA.

Het lagere niveau is het niveau van de protocolconversie: hoe kan een X.25 netwerk ingepast worden in een SNA-netwerk?

## HOOFDSTUK 2

### SNA - EEN VERGELIJKING MET DE UITGANGSPUNTEN VAN OSI

In een willekeurige inleiding tot de computercommunicatie zal me allicht een beschouwing aantreffen over netwerkmodellen en -architecturen, waarin bijvoorbeeld OSI en SNA worden besproken. Zoals er overeenkomsten aan te wijzen zijn, zijn er verschillen en die verschillen wil ik hier, in het geval van OSI (Open System Interconnection vs. SNA (Systems Network Architecture) nader belichten.

Ik zal op deze plaats niet een volledige beschrijving geven van OSI hiervoor verwijs ik naar de literatuuropgave achterin: 1 en 3. Hetzelfde geldt overigens voor SNA, zie lit. 2, 5. Ik zal vertrekken van een viertal uitgangspunten, die aan de basis staan van OSI en SNA hieraan toetsen.

OSI is een netwerkmodel, dat pretendeert een referentiekader te zijn voor verschillende netwerkarchitecturen. SNA is zo' netwerkarchitectuur, die overigens niet is opgezet met het OSI model in gedachten (dat zou een anachronisme zijn), maar het is wel interessant om te proberen het OSI model toe te passen op SNA. In de volgende paragraaf zal ik daarom SNA toetsen aan een aantal uitgangspunten van OSI.

#### 2.1 UITGANGSPUNTEN

De volgende uitgangspunten zijn door ISO (International Organisation for Standardisation) toegepast bij het ontwerp van het 'Referenc Model of Open Systems Interconnection (OSI) (lit. 1):

- o Ieder niveau van abstractie heeft zijn eigen laag.
- o Iedere laag vervult een gedefinieerde functie
- o De grenzen tussen de lagen worden zo gekozen dat de informatie-uitwisseling tussen de lagen minimaal is.



- o Het aantal lagen moet zo groot zijn, dat verschillende functies niet in dezelfde laag gestopt hoeven worden, en zo klein dat de architectuur niet onhandelbaar wordt.

De uitgangspunten die gehanteerd werden bij het definiëren van SNA daarentegen waren geheel anders. Een aantal overwegingen zijn de volgende:

- o SNA is uitsluitend ontwikkeld om IBM-apparatuur te laten communiceren en niet met de gedachte aan een verscheidenheid aan leveranciers en een nog grotere verscheidenheid aan apparatuur. Dit betekent, dat SNA een veel dwingender karakter heeft, terwijl OSI steeds allerlei zaken heeft moeten open houden.
- o SNA definieert precies welke nodes (knooppunten) in het netwerk voorkomen en wat hun karakteristieken zijn. Dit is het gesloten karakter van een SNA netwerk: het SNA netwerk is allesomvattend. Een OSI netwerk is open, dat betekent (o.m.) dat de nodes vallen buiten het definitiegebied van de architectuur. Een dergelijk netwerk verbindt verschillende open systemen.
- o SNA is eerder ontstaan en was initieel ook relatief eenvoudig. In de loop der jaren is er een grote hoeveelheid functionaliteit bijgekomen, maar altijd met de compatibiliteit met de vorige generaties hoog in het vaandel. Dit heeft tot gevolg dat een hoeveelheid 'historische ballast' steeds meegesleept moet worden. Daardoor is de hele architectuur niet altijd meer even eenvoudig en helder. OSI heeft wat dat betreft het voordeel gehad van een frisse start, waarbij geleerd kon worden van de problemen van reeds bestaande netwerk(architectur)en. Bij de ontwikkeling van OSI heeft men eerst een raamwerk opgezet, waarbinnen de deelspecificaties later werden ingevuld.

## 2.2 HISTORIE

SNA is, veel meer dan OSI, het resultaat van een historische ontwikkeling. De geschiedenis van SNA is dan ook veel ouder. De oorspronkelijke beschrijvingen van SNA (SNA-0, 1 en 2), onderscheidden slechts de volgende lagen:

Application laag
Function Management laag
Transmission Subsystem laag

De Transmission Subsystem laag draagt de zorg, dat data van een bepaald punt in het netwerk naar een bestemmingspunt gebracht worden. Deze punten noemt SNA NAU's: Network Addressable Units. Als zodanig is de Transmission subsystem laag vergelijkbaar met de lagen 1, 2 en 3 in OSI. De Function management laag vervult alle communicatie functies die slechts end-to-end betekenis hebben (Deze functies bevinden zich o.m. in de LU's: Logical Units.) Deze laag valt dus te vergelijken met OSI lagen 4, 5 en 6. De Application laag tenslotte vervult de toepassing, die uitstijgt boven de communicatiefunctie. Dit correspondeert met de Application laag 7 in OSI.

Deze indeling dateert uit 1974. In 1976, met de introductie van SNA-3 (of SNA/ACF: Advanced Communications Functions), en in 1979 (SNA4) werden verfijningen aangebracht, die het aantal lagen opvoerden.

De Function Management laag valt uiteen in de volgende lagen:

NAU services Manager
Function Management Data Services (FMD)
Data Flow Control
Transmission Control

De Transmission Control laag heeft tot taak de transmissie tussen de twee kanten van een sessie orderlijk te laten verlopen, m.b.t. de transmissie. Deze laag kan sessies starten en beëindigen en voortdurend controleren, bijvoorbeeld m.b.v. volgnummers.

De functies van de Data Flow Control laag vallen in twee categorieën uiteen:

- o Functies die te maken hebben met de data-eenheden (RU's Request Units) die door de onderliggende lagen worden getransporteerd, zoals wie wanneer RU's mag sturen, het aangeven van RU's die bij elkaar horen (Chaining, brackets).
- o Functies die de algemene data stroom betreffen.

De Function Management Data Services laag stuurt data door naar de NAU Services Manager die de gevraagde functie kan verrichten. Er zijn namelijk verschillende NAU Services, zoals End-user Services en Network Services voor de verschillende soorten NAU's (PU, LU en SSCP).

De Transmission Subsystem laag valt uiteen in de volgende lagen:

Virtual Route Control	
Explicit Route Control	samen: Path Control
Transmission Group Control	Laag
Data Link Control	

De Data Link Control laag kent verschillende invullingen: SDLC, System/370 Channel, HDLC, token ring LAN en specificeert het link-laag protocol.

De Transmission Group Control laag definieert het protocol dat toegepast wordt tussen twee naburige knooppunten waartussen meerdere links parallel aanwezig zijn.

De Explicit Route Control laag definieert routes tussen iedere twee knooppunten in het netwerk (om precies te zijn, tussen alle paren subarea's) door middel van genummerde Explicit Routes.

De Virtual Route Control laag voegt hieraan toe de functionaliteit van de Class Of Service en definieert over de Explicit Routes heen zogenaamde Virtual Routes.

Er wordt wel schertsend gezegd, dat IBM in 1979, onder druk van het opkomende OSI model, snel het aantal lagen in SNA opvoerde tot 7, om niet achter te blijven bij de 7 OSI lagen. Men moet zich echter terdege realiseren dat, op het moment dat SNA werd aangekondigd in 1974, iets als OSI nog niet bestond en dat SNA in de loop der jaren steeds moest veranderen om bij te blijven bij de groeiende eisen, inzichten en technieken.

In 1982 bijvoorbeeld, introduceerde IBM Advanced Program to Program Communication ('Advanced' is in IBM jargon de kreet om iets aan te duiden dat eigenlijk al lang had moeten kunnen, maar pas juist mogelijk is geworden), waarmee communicatie mogelijk werd tussen twee applicatie programma's (LU 6.2).

In 1983 introduceert IBM SNI: SNA Network Interconnection dat verbindingen tussen overigens onafhankelijke netwerken mogelijk maakt middels gateways.

In 1986 werd Advanced (!) Peer to Peer Networking aangekondigd, waarmee de routing in een netwerk op een dynamischer manier wordt geregeld. Hiermee werd ingespeeld op zaken als Local Area Networks (LAN's) en X.25.

Samenvattend kunnen we nu de volgende zeven SNA-lagen onderscheiden:

Application
NAU services Manager
Function Management Data Services
Data Flow Control
Transmission Control
Path Control
Data Link Control

Overigens ontbreekt aan deze opsomming van lagen een laag met de naam als 'Physical Control', vergelijkbaar met laag 1 in het OSI model. Zoals SNA en OSI het er over eens zijn, dat de 'end user' niet tot de netwerk architectuur behoort, zo stelt SNA dat de physical laag niet tot deze architectuur behoort. Wanneer we de physical laag onder SNA denken, ontstaan er dus niet zeven maar acht lagen. What's in a number.

Het resultaat van de hier kort beschreven historische ontwikkelingen is nu dat de scheiding tussen de zeven SNA lagen echter niet zo strikt als mogelijk was geweest, anders gezegd, niet zo strikt als in OSI. De oorspronkelijke drie lagen (application, function management en transmission subsystem) zijn goed gescheiden. De scheiding tussen de verschillende deellagen echter is veel vager. Dit zal in het navolgende ter sprake komen.

### 2.3 SNA GETEST AAN DE OSI-UITGANGSPUNTEN

Laten we nu eens terugkeren op de genoemde uitgangspunten van OSI en zien in hoeverre SNA die uitgangspunten deelt.

Het eerste uitgangspunt luidt:

'Ieder niveau van abstractie heeft zijn eigen laag'.

Als we kijken naar de drie grote lagen: Applicatie, Function Management en Transmission Subsystem, dan zien we dat de Transmission Subsystem laag verantwoordelijk is voor het transport van 'SNA message units' tussen knooppunten in het netwerk. Anders gezegd: het Transmission Subsystem biedt een datagram service. Het transmission subsystem vormt een connectieloos netwerk. De function management laag heeft een hoger niveau van abstractie. Het definieert 'sessions', die verbindingen tussen de zogenaamde NAU's voorstellen. Deze sessions kunnen opgebouwd en afgebroken kunnen worden, en er is een variëteit aan mechanismes om data te groeperen ('blocking', 'chaining', 'brackets', 'sync points') of juist te splitsen ('segmenting'). De function Management laag definieert ook mechanismes voor data flow control, dat in SNA aangeduidt wordt als 'Pacing' (de term 'Data Flow Control' wordt ook gebruikt in SNA, en

heeft betrekking op o.m. chaining en brackets). De application laag ten slotte mag door de gebruiker van SNA worden ingevuld, zij valt buiten het werkingsbereik van SNA (net zoals de physical laag aan de andere kant van het spectrum).

De onderverdeling van het Transmission Subsystem in de lagen Data Link, Transmission Group, Explicit en Virtual Route Control is eveneens een voorbeeld van verschillende lagen van abstractie: Ieder nieuw niveau begeeft zich op een hoger vlak en bemoeit zich steeds minder met de details.

De onderverdeling van de Function Management laag heeft echter niets te maken met abstractieniveaus. We zullen dit in het hiernavolgende duidelijk maken, maar we nemen eerst het tweede OSI-uitgangspunt er ter vergelijking bij:

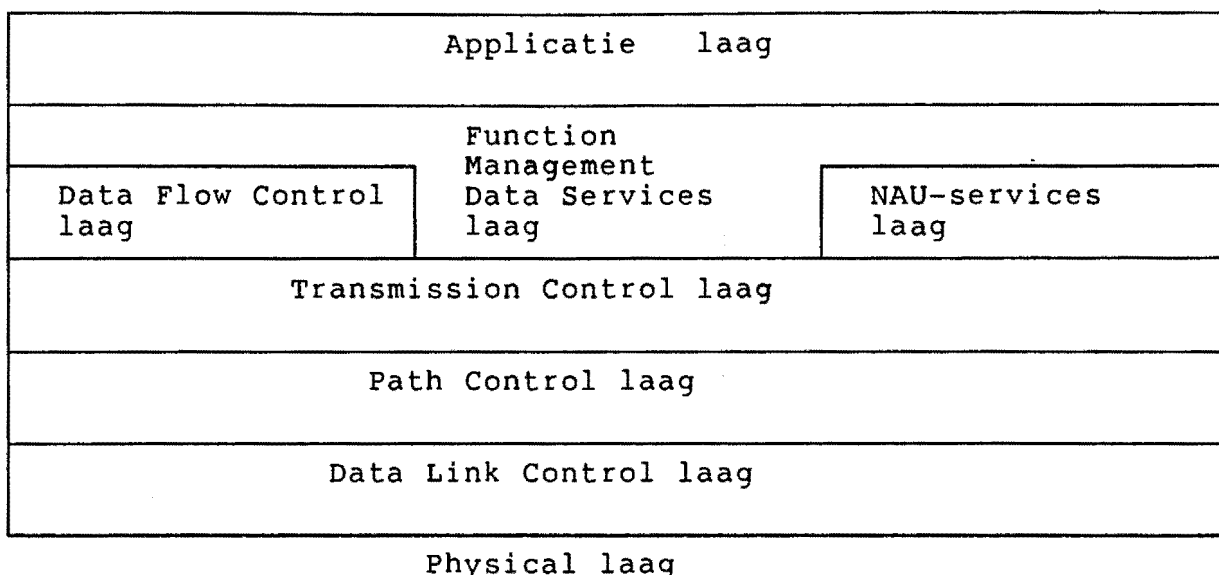
'Iedere laag in OSI vervult een bepaalde, strikt gedefinieerde functie.'

In SNA geldt deze stelling ook, wanneer toegepast op de drie oorspronkelijke lagen. Het betekent, dat er een functionaliteit, of service, geboden wordt naar de bovenliggende laag en een bepaalde functionaliteit geëist wordt van de onderliggende laag. Essentieel is, dat de betreffende laag gedefinieerd wordt door de functionaliteit die het veronderstelt van de onderliggende laag, en de functionaliteit die het biedt naar de bovenliggende laag. Dat zijn ook de enige twee 'entity's' waarmee een laag te maken heeft. Die twee grensvlakken heten 'interfaces'. Een interface moet vooral niet verward worden met een 'protocol', dat is een voorschrift voor communicatie tussen twee entity's in dezelfde laag.

Wanneer men de leden van de zeven-lagenstructuur van SNA één voor één onder de loep neemt, kan men voor een aantal lagen duidelijk de interfaces aangeven, terwijl andere interfaces nauwelijks als zodanig bestaan.

De Data Link Control laag is zeer goed vergelijkbaar met de gelijknamige laag in OSI, en vormt een goed gescheiden laag, zoals bijvoorbeeld gespecificeerd door SDLC, Synchronous Data Link Control.

De Path Control laag als geheel, en de Transmission Control laag zijn eveneens ieder systemen met een duidelijke interface naar boven en naar beneden. De drie bovenliggende lagen Data Flow Control, Function Management Data Services en NAU Services kunnen beter niet beschouwd worden als lagen, in de zin zoals boven beschreven: lagen die een interface hebben van laag op laag, en een bepaalde service leveren en verwachten naar / van respectievelijk hun bovenbuur en benedenbuur. In plaats hiervan, stel ik voor deze drie eenheden naast elkaar te plaatsen op de Transmission Control laag, zoals in het volgende schema:



Dit is niet eens zo gek als het lijkt, en ik zal het het volgend voor aanvoeren.

In het algemeen is het zo, dat iedere laag van zijn bovenbuur een hoeveelheid data aangeleverd krijgt, die getransporteerd moet worden. De betreffende laag beschouwt de aangeleverde data onveranderlijk als betekenisloze data, die ten behoeve van de bovenliggende laag (min of meer) onveranderd getransporteerd moet worden.

Hiervoor wordt vaak het begrip transparantie gebruikt, dat tweezijdig werkt. De bovenliggende laag kan willekeurig welke data aanleveren en interesseert zich niet voor de methoden die de onderliggende laag ter transport toepast; de onderliggende laag interesseert zich niet voor de inhoud van de data. Transparantie betekent voor de bovenliggende laag, dat alles wat onder gebeurt onzichtbaar is, zij kijkt daar dwars door heen (naar de collega laag, de 'peer entity' waarmee gecommuniceerd wordt via het protocol). In het OSI lagensysteem bestaat er een bepaalde correspondentie tussen het begrip laag enerzijds en het begrip header (of omvattend frame) anderzijds. De onderliggende laag transporteert de aangeleverde data (de body) door het in te pakken in een frame volgens het protocol dat geldt voor die laag. Dat frame bestaat in principe uit een header plus een trailer, maar meestal alleen uit een header.

Omdat iedere laag dat zo doet, kan men op het fysieke niveau dat voorbij zien stromen, die bestaat uit een aantal headers gevolgd door de user-data en een aantal trailers. In een zeven-lagen systeem verwacht men dus in principe zeven headers na elkaar.

Laten we nu eens kijken naar een SNA Basic Link Unit, 'een data pakket zoals dat op de link laag wordt getransporteerd. Hier wordt een bepaald type BLU geschetst, namelijk een ten behoeve van Function Management Data Services.

Link header	Transm. header	Request header	Function manag. hdr	End-user data	Link Trailer
-------------	----------------	----------------	---------------------	---------------	--------------

Al tellend komen we tot vier headers, user-data en een link trailer die hoort bij de Link Header. Bij welke lagen horen deze headers?

De Link Header en Trailer horen bij de Data Link Control laag, dat wil zeggen, de Link Header wordt gedefinieerd door het protocol van de Data Link Control laag.

De Transmission Header hoort bij de Path Control laag. Merk op, dat de Path Control laag als geheel slechts met een header correspondeert, terwijl deze laag verondersteld wordt te bestaan uit drie sublagen. Omdat de drie sublagen niet ieder voor zich een header hebben, kunnen de peer entity's ook niet onafhankelijk met elkaar communiceren. Voorts kan er geen sprake zijn van interfaces tussen de sublagen, in benoemen van de verschillende functies die de Path Control laag heeft.

De Request Header vervolgens is een product van de Transmission Control laag.

De Function Management Header tenslotte wordt gebruikt door de Function Management laag en we zien hier weer een merkwaardig verschijnsel: De Data Flow Control laag wordt geacht tussen de Function Management laag en Transmission Control laag in te liggen, maar is nergens terug te vinden in de BLU, evenmin als de NAU services laag. De reden hiervoor is de volgende. De geschetste BLU is een BLU voor het transport van end-user data. Zo'n BLU heeft de opmaak zoals beschreven. Er zijn echter andere BLU's, die informatie uitwisselen voor de NAU services laag en de Data Flow Control laag.

De algemene opmaak voor BLU's is de volgende:

Link header	Transm. header	Request header	Request Unit	Link Trailer
-------------	----------------	----------------	--------------	--------------

De Request Unit (RU) kan gebruikt worden door ieder van de lagen Data Flow Control, Function Management Data Services (FMD) en NAU services. Zoals we reeds gezien hebben, bestaat de RU in geval van de als voorbeeld genoemde FMD services uit Function Management Header plus user-data. Alle drie sublagen communiceren echter direct met RU's, gebruik makend van de Transmission Control laag.

Een ander voorbeeld dat aantoont dat niet alle 'lagen' zich steeds als zodanig gedragen, wordt geleverd door het volgende (lit. 5 pag. 137).

SNA definieert zogenaamde exception requests. Wanneer een request (een data eenheid die door de Path Control laag wordt getransporteerd) via een aantal knooppunten zijn bestemming bereikt is het mogelijk dat de Path Control laag een fout aantreft in de getransporteerde data. Het betreft hier dan niet een fout op het niveau van de Path Control Laag, maar een fout in de te transporteren data. SNA erkent, dat het niet gepast zou zijn voor de Path Control laag om de data te weigeren. De Path Control laag zal het pakket gewoon naar de bestemming moeten transporteren. SNA definieert echter, dat het de Path Control laag toegestaan is de inhoud van de request te modificeren en wel op de volgende wijze. Het bericht (de request) wordt afgekapt tot een lengte van 3 byte (of zoveel korter de originele request was), en voorafgegaan door vier bytes zogenaamde sense data (dit impliceert wellicht dat er ook non-sense data bestaat?). Bovendien wordt in de request header een bit gezet om aan te duiden dat de resulterende request een exception request is. De aldus verminkte request wordt naar de bestemming doorgestuurd.

Deze wijze van het uitvoeren van de transporttaak strookt mijns inziens in het geheel niet met een gelaagde structuur. In een lagenmodel is de door een bepaalde laag te transporteren data voor die laag altijd betekenisloos en een laag zal daaraan dus nooit kunnen en mogen veranderen. Het is mogelijk dat er fouten optreden in het traject van bron naar bestemming, en dat een laag de gevraagde functie daardoor niet uit kan voeren, maar dit kan nooit systematisch gebeuren op grond van de aangeboden data.

Het derde geciteerde uitgangspunt van OSI luidt:

'De grenzen tussen de lagen worden zo gekozen dat de informatie-uitwisseling tussen de lagen minimaal is.'

Ook op dit punt kunnen we niet anders concluderen dan dat de interface tussen de drie oorspronkelijke lagen logisch gekozen is maar dat in de sublagen de scheiding niet is aangevraagd op grond van minimale informatie-uitwisseling. De lagen, die volgens het lagenmodel helemaal niet met elkaar interfaceren, staan in het herziene model zelfs op gelijke hoogte met elkaar, zodat een aanzienlijke verstrengeling onvermijdelijk is.

'Het aantal lagen moet zo groot zijn, dat verschillende functies niet in dezelfde laag gestopt hoeven worden, en zo klein dat de architectuur niet onhandelbaar wordt.'

Wat dit betreft, kan men zeggen dat IBM zich in een later stadium aan dit uitgangspunt heeft geconfirmeerd door het opvoeren van het aantal lagen.



## HOOFDSTUK 3

### EEN SNA-INFRASTRUCTUUR

Het doel van het onderhavige project is, verbindingen te leggen tussen SNA en OSI. Deze verbindingen zullen in dit hoofdstuk en het volgende op twee niveaus worden gelegd.

De niveaus die ik hier bedoel, zijn onderscheidbaar in het in het vorige hoofdstuk gegeven lagenschema. Zoals aldaar betoogd, is het onmogelijk een scheiding te trekken tussen bepaalde niveaus in het SNA-schema. Wanneer we willen gaan spelen met het lagenschema, moeten we dus een plaats kiezen waar dat op een geschikte manier kan. In het volgende hoofdstuk zal het lagenschema opengeknipt worden tussen de Path Control en de Data Link Control laag.

In dit hoofdstuk zullen we echter eerst op een andere manier tegen de vergelijking SNA - OSI aankijken. Ik wil hier de ik het idee ontvouwen van een SNA-netwerk of -infrastructuur. Deze infrastructuur biedt een overkoepelend netwerk waaraan diverse deelnetwerken (of niet in een netwerk opgenomen apparatuur) kunnen worden verbonden.

Deze infrastructuur kan ten dienste staan van een bepaald bedrijf, maar het is eveneens denkbaar, dat een instantie een dergelijke infrastructuur openbaar zou aanbieden, vergelijkbaar met diensten als het telefoonnet, telexnet en datanet, maar ook met zaken als PAD- (Packet Assembly and Disassembly) en VAP-diensten (Videotex Access Point).

Voordelen voor de gebruikers kunnen onder meer zijn:

- o Eindgebruikers (dit begrip wordt later gedefinieerd) kunnen vanaf één locatie inloggen op meerdere E.D.P. diensten.
- o Gatewayaansluitingen (komen eveneens nog ter sprake) kunnen op relatief goedkope wijze aangesloten worden op een netwerk waarbij men geen leased lines hoeft te exploiteren en evenmin zelf PSI (Packet Switching Interface) software hoeft te gebruiken.

We beschouwen het netwerk vanaf een drietal standpunten.

In de eerste plaats kan het van buiten beschouwd worden: welk dienst biedt het netwerk aan? Het netwerk verbindt de aangesloten gebruikers. Deze gebruikersaansluitingen kunnen we in twee groepen verdelen: De eindaansluitingen zijn voor kleine gebruikers. Zij bezitten uitsluitend een aantal beeldschermen met een cluster controller. De gatewayaansluitingen daarentegen zijn 'grote aansluitingen waarop hosts dan wel hele (particuliere) SNA-netwerken aangesloten kunnen worden.

De tweede zienswijze is die vanuit het SNA-standpunt: De SNA-infrastructuur, tesamen met de op de eindaansluitingen aangesloten cluster controllers en beeldschermen vormt een SNA-netwerk. De particuliere netwerken vormen eveneens zelfstandig netwerken. De verbindingen tussen de particuliere netwerken enerzijds en de infrastructuur anderzijds geschiedt door middel van gateways. Een gateway bevindt zich in beide netwerken, en gedraagt zich in beide netwerken als een Communications Controller (ComC). Door middel van netwerkadresconversies kan een gateway (een deel van) het ene netwerk bereikbaar stellen voor het andere netwerk.

De derde zienswijze bevindt zich op een lager niveau: hoe is de infrastructuur intern opgebouwd? We zullen zien, dat het aantrekkelijk is om gebruik te maken van de reeds bestaande infrastructuur in de vorm van datanet, een X.25 netwerk. Transporttaken kunnen uitstekend gerealiseerd worden door dit netwerk, omdat datanet een groot aantal problemen, variërend van routing en flexibele verkeersspreiding tot netwerkbeheer, oplost.

### 3.1 DOELSTELLING

Het doel van de SNA-infrastructuur luidt als volgt: Het aanbieden van een netwerkdienst, waardoor SNA-apparatuur verbonden kan worden. Zowel hosts en netwerken, bestaande uit één of meer hosts als eenvoudige cluster controllers kunnen op het netwerk worden aangesloten. Het netwerk biedt de mogelijkheid sessions op te zetten tussen een willekeurige host en een willekeurige cluster controller, maar ook tussen hosts onderling.

### 3.2 HET GEBRUIKERSSTANDPUNT

Vanuit het gebruikersstandpunt gezien, biedt het netwerk de volgende functies:

- Mogelijkheid tot aansluiten van cluster controllers met beeldschermen. Deze aansluitingen noemen we eindaansluitingen.

- Mogelijkheid tot aansluiting op bestaande SNA-netwerken, lees: hosts, van gebruikers. Deze aansluitingen noemen we gateway-aansluitingen, omdat ze een gateway functie bieden op het netwerk. Het is hier belangrijk op te merken, dat een SNA-netwerk van een gebruiker zijn integriteit volkomen kan bewaren: alle gebruiker-SNA-netwerken vormen onafhankelijke SNA-netwerken, net zoals de SNA-infrastructuur een onafhankelijk netwerk vormt.

De koppeling tussen de diverse SNA-netwerken geschiedt middels gateways, die een half-doorzichtige overgang bieden tussen de netwerken.

- Mogelijkheid tot communicatie van gateway-aansluiting naar iedere andere gateway-aansluiting.
- Mogelijkheid tot communicatie van eindaansluiting naar willekeurige gateway-aansluitingen.

### 3.3 HET SNA-STANDPUNT

Als we de SNA-infrastructuur van een SNA-point-of-view bekijken, zien we een verzameling ComC's die op een bepaalde, nog nader te analyseren wijze aan elkaar zijn gekoppeld. Het belangrijkste aspect van deze koppeling is, op dit moment, het feit dat iedere ComC iedere andere ComC kan bereiken, middels wat heet het backbone netwerk.

Er bestaan twee soorten ComC's, namelijk ComC's die verbindingen hebben met eindaansluitingen en gateways.

De ComC's die eindaansluitingen bedienen, zijn 'gewone' ComC's, met een link naar de rest van het backbone netwerk en links naar de cluster controllers die er op zijn aangesloten.

De gateway-ComC's, kortweg gateways zijn ComC's die aansluitingen hebben in verschillende, overigens onafhankelijke SNA-netwerken. Zij staan als het ware met hun benen in verschillende netwerken. Gateways kunnen een poort openen tussen die netwerken.

#### 3.3.1 Gateways

We zullen nu dieper ingaan op de werkwijze van gateways.

Binnen een SNA netwerk bestaan twee soorten 'adressen'. Er zijn symbolische namen, waarmee op een hoog niveau bestemmingen kunnen worden aangeduid. Er zijn echter ook 'path control' adressen, die door de Path Control Layer van SNA worden begrepen. Bij het aangaan van een sessie, wordt een symbolische naam gespecificeerd, die eerst (via een tabel) moet worden vertaald naar een adres. Deze vertaaltabel is een gegeven voor één netwerk, er is dus binnen het netwerk een een-op-een relatie tussen de adressen en de symbolische

namen.

Wanneer nu twee SNA-netwerken gekoppeld worden door een gateway, dan ontstaan er in ieder geval twee 'adresruimten' voor de path control adressen. De symbolische namen kunnen zo gekozen worden, dat een gedeelte van de namen in beide netwerken bekend is. Het overige namen zouden dan privé adressen genoemd kunnen worden. Het is ook mogelijk om een adres dat in het ene netwerk bekend staat onder bijvoorbeeld de naam JAN, in het andere netwerk een andere naam te geven, bijvoorbeeld NETB\_JAN.

Hoe verloopt nu de vertaling van symbolische namen, die in het ene andere netwerk liggen, naar path control adressen? De gateway wordt vanuit elk der netwerken gezien als een ComC, die toegang geeft tot een deel van de adresruimte. De symbolische namen die refereren aan het andere netwerk, worden vertaald naar adressen die verwijzen naar de gateway-ComC.

De gateway ontvangt dus de sessies die bestemd zijn voor het andere netwerk, maar gedraagt zich alsof hij zelf de bestemming is. Naar het andere netwerk toe gedraagt hij zich eveneens als een ComC die deel uit maakt van het SNA netwerk. Binnenkomend verkeer wordt doorgestuurd naar het bestemmingsnetwerk, terwijl de adressen en zo nodig de logische namen via een tabel door de gateway worden vertaald. Beide SNA-netwerken merken in het geheel niet op dat deze verbinding in feite naar een ander netwerk ligt. Een SNA netwerk kan dat ook niet opmerken, omdat de notie 'ander netwerk' niet gedefinieerd is binnen SNA.

### 3.3.2 Waarom gateways?

Waarom is gekozen voor de toepassing van gateways?

Met het netwerk moeten twee onderling strijdige belangen worden verenigd: enerzijds moet de op het netwerk aangesloten apparatuur kunnen communiceren, anderzijds moet niet te sterk worden ingegrepen op de reeds bestaande structuren van privé-netwerken die op de SNA-infrastructuur worden aangesloten.

Voor aan te sluiten cluster controllers is de laatste eis niet zo belangrijk, maar voor bestaande SNA-netwerken is het heel belangrijk. Privé netwerken hebben altijd een bepaald 'nummerplan' in de zin van SNA-adressen, waarvan het niet wenselijk is dat te combineren met het nummerplan van de SNA-infrastructuur, laat staan met dat van andere particuliere netwerken.

Gateways bieden de mogelijkheid om de verschillende netwerken onafhankelijk te laten blijven.

### 3.3.3 Globale structuur

Het geheel van SNA-infrastructuur plus de erop aangesloten gebruikers wordt geïllustreerd in fig. 1. We kunnen zeggen, dat het bestaat uit de volgende zelfstandige SNA netwerken:

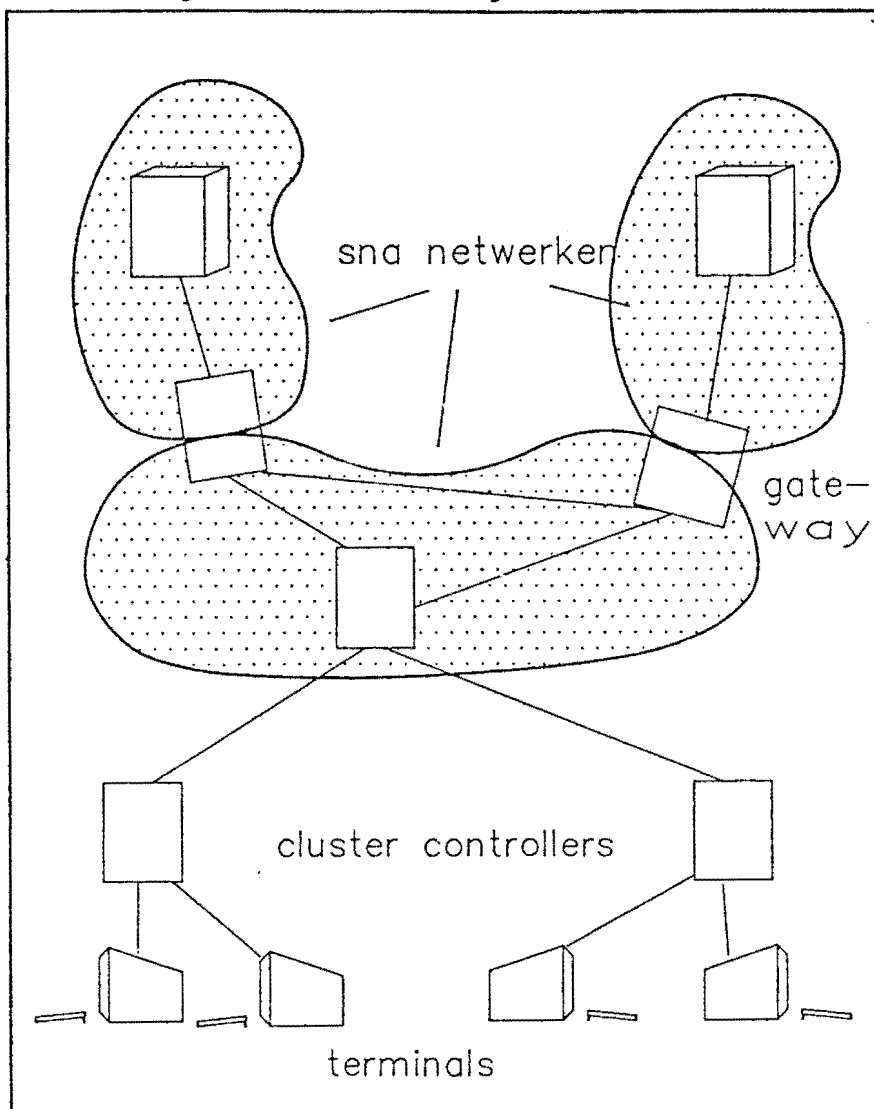


Fig. 1. SNA-infrastructuur en gebruikers.

- De SNA-infrastructuur: de ComC's, mogelijk een host computer, de verbindingen tussen die verschillende nodes, en de op de eindaansluitingen aangesloten CC's.
- Verschillende particuliere SNA-netwerken, die als het ware gegroepeerd zijn rond het netwerk. Gateways verbinden ieder netwerk met het de SNA-infrastructuur. De verschillende particuliere netwerken kunnen met elkaar communiceren door het openbare netwerk heen, gebruik makend van twee gateways in serie.

### 3.4 DE ONDERLIGGENDE NETWERKSTRUCTUUR

Op een niveau, weer een stapje lager, zien we dat de beschrijving op SNA-niveau nog een aantal belangrijke aspecten onbelicht laat. Zoals gezegd, vormen de ComC's binnen het netwerk een backbone netwerk. In een traditioneel netwerk worden de ComC's onderling verbonden door leased lines.

Deze aanpak heeft echter een aantal nadelen, waarvan de belangrijkste zijn:

- Leased lines zijn niet goedkoop.
- Zij zijn niet flexibel qua capaciteit: een leased line ligt er of hij ligt er niet. Plotselinge wisselingen van de hoeveelheid verkeer kunnen niet flexibel worden opgevangen.

Omdat een volledig vermaasd netwerk onmogelijk wordt, wanneer er meer dan enkele knooppunten worden verbonden (er zouden  $\frac{1}{2}n(n-1)$  links nodig zijn), wordt altijd een gedeeltelijk vermaasd netwerk gebouwd. Dit fenomeen resulteert voorts in de volgende nadelen:

- Vrijwel alle sessies moeten gebruik maken van tussenliggende ComC's. Op de capaciteit van deze ComC's wordt derhalve een aanslag gedaan.
- Ze moeten gebruik maken van meer dan een leased line, we spreken van meerdere 'hops', waardoor in veel gevallen een omweg moet worden genomen. Een omweg betekent meer en langere leidingen en dus verminderde efficiëntie.
- Voorts kan het tijdelijk uitvallen van ComC's vervelend gevolgen hebben, als het netwerk onvoldoende sterk vermaasd is.

Vanwege deze nadelen verdient het sterke aanbeveling gebruik te maken van X.25 als onderliggende communicatiestructuur. Hiermee is het mogelijk, vanuit SNA-standpunt gezien, een volledig vermaasd netwerk te creëren dat lang niet de prijs heeft van  $\frac{1}{2}n(n-1)$  ( $n$  is het aantal ComC's) leased lines van een echt vermaasd netwerk.

Ook van knooppunten in een X.25 netwerk is het denkbaar dat zij uitvallen, en in principe zijn de bovengenoemde nadelen evenzeer van toepassing op elk ander communicatienetwerk als op het onderhavige net. Doordat het X.25 netwerk echter niet alleen als basis dient voor het hier beschreven SNA netwerk, maar als datacommunicatienetwerk een veel ruimer toepassingsbereik heeft, en dus groter is, is het beter vermaasd, en dus uitermate betrouwbaar bij een redelijke kostprijs.

### 3.4.1 X.25 onderbouw met PVC's

X.25 wordt als onderliggende netwerkstructuur gebruikt, dat wil zeggen, X.25 bevindt zich in een lagenmodel onder de Path Control Layer van SNA. Om dit te kunnen realiseren is een protocol conversie noodzakelijk, die de in SNA formaat aangeboden messages kan versturen over X.25 VC's.

Er zijn twee fundamenteel verschillende manieren om protocol conversie te doen. De eerste manier wordt hier besproken. Het alternatief wordt in de volgende paragraaf besproken.

We kunnen op de volgende manier een volledig vermaasd SNA-netwerk opzetten. Gebruik ComC's met PSI (lit. 4). PSI is een optie die het de ComC mogelijk maakt rechtstreeks op een X.25 netwerk aangesloten te worden. De ComC gebruikt dan een aantal PVC's als leased lines.

Bij een beperkt aantal ComC's is het mogelijk om iedere ComC PVC's te laten hebben met iedere andere ComC. Het aantal ComC's moet dan niet veel groter zijn dan een stuk of twintig, daar anders de hoeveelheid PVC's de spuigaten uitloopt, volgens de formule  $PVC's = \frac{1}{2} n (n-1)$ , waarin n het aantal ComC's is.

### 3.4.2 Protocol Conversie met PCU's

Indien het aantal ComC's niet langer klein is, wordt het onderhouden van zo'n groot aantal PVC's een onhaalbare zaak. Er zijn in principe twee mogelijkheden: Stap af van het concept van een volledig vermaasd netwerk, en voer slechts een gedeelte van de PVC's uit. Als nadelige consequentie heeft dit, dat voor een gedeelte van de mogelijke verbindingen gebruik gemaakt moet worden van meerdere X.25 hops. Dit betekent slechtere efficiëntie, daar twee hops ook twee maal de kosten betekent. In de paragraaf 'Noodzaak van adresconversie' worden systematisch enkele factoren naast elkaar gezet.

Om toch een volledig vermaasd netwerk te simuleren, moet een intelligente conversie toegepast worden. De converter die de conversie uitvoert, althans als dat een zelfstandig apparaat is, noemen we een PCU, een Protocol Conversion Unit, geïllustreerd in fig. 2.

De PCU's staan aan de rand van het X.25 netwerk. Daardoor kan het X.25 netwerk samen met de PCU's weer beschouwd worden als een netwerk. Dit netwerk heeft als taak, elke aangeboden SNA message te transporteren naar de bestemming (opgegeven in het SNA-adres). Zelf produceert of consumeert het netwerk natuurlijk geen verkeer.

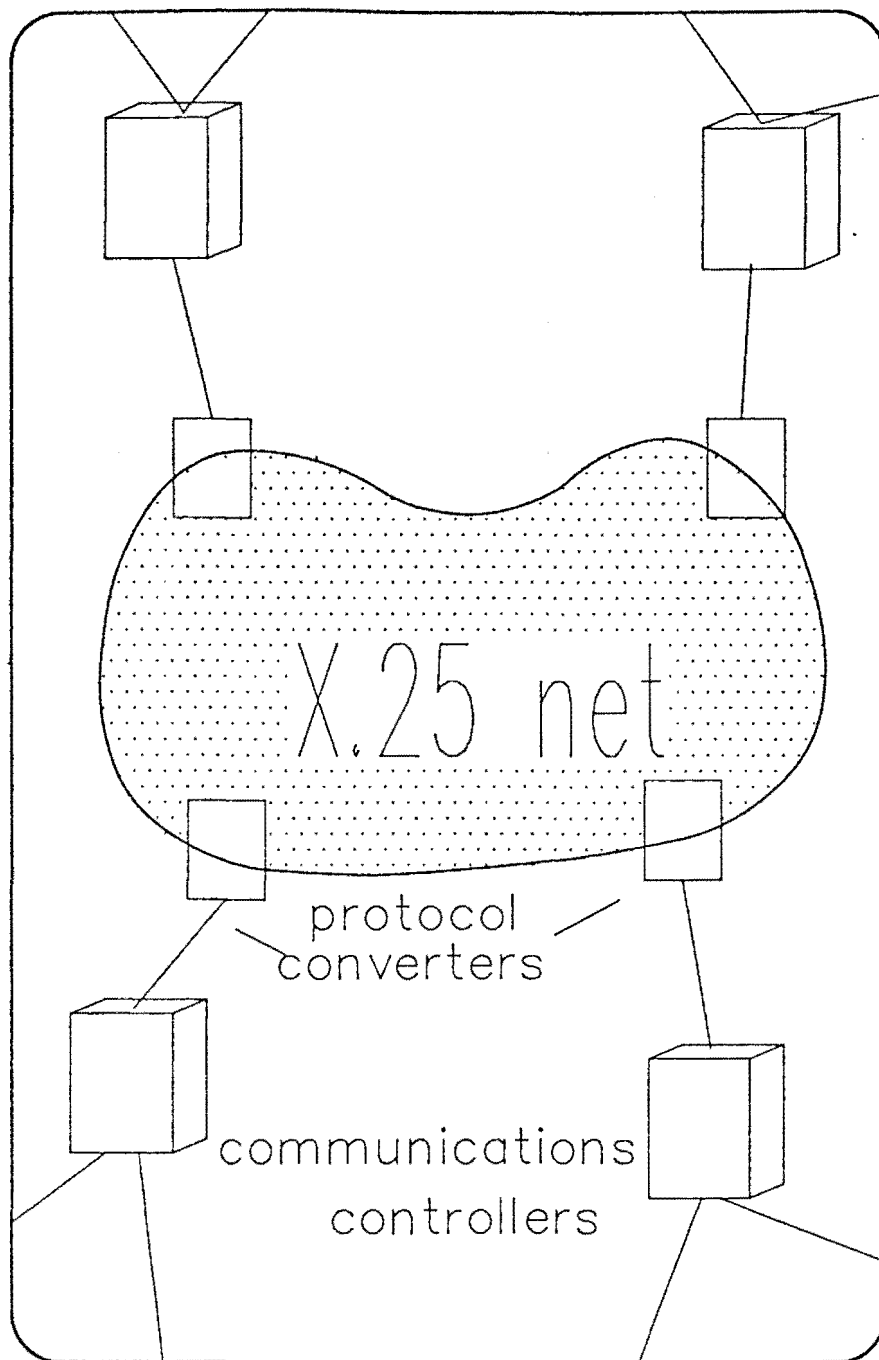


Fig. 2. Protocolconverters tussen  
X.25 en SNA.

We kunnen nu Cluster Controllers en ComC's nemen die niet behoeven te worden uitgerust met PSI software (er zijn omstandigheden denkbaar waarin dit wel nuttig zou kunnen zijn), zij hebben slechts een SDLC verbinding met de PCU.



Op deze SDLC verbinding kan multidrop worden toegepast waarbij de ClC (ComC) het primary station is. De PCU vertegenwoordigt alle secondary stations en transporteert de per multidrop aangeboden data via een X.25 VC.

De PCU kan in een tabel opzoeken, naar welk X.25 netwerkadres iedere message moet worden verstuurd. De PCU zet een SVC op naar de bestemming waarvoor verkeer wordt aangeboden. Indien er reeds een SVC bestaat naar het gevonden X.25-adres, dan kan de message over die bestaande sessie verzonden worden, anders moet onmiddellijk zo'n SVC worden opgezet. Het verzenden van de message wil zeggen: de complete SNA-message, inclusief de Transmission Header (die de SNA-adresinformatie bevat) maar exclusief de Link Control Header, wordt in een X.25 pakket ingepakt, en verstuurd.

De ontvangende PCU haalt de SNA-message er uit door de X.25 pakket header weg te gooien. De PCU stuurt het door naar de ComC die op de PCU is aangesloten.

Zolang er volgens een of ander criterium verkeer over een bepaalde SVC verwacht kan worden, kan deze verbinding blijven bestaan. Er zijn een aantal manieren om te besluiten dat een SVC gecleard kan worden.

- De PCU kan het SNA verkeer observeren, en bijhouden welke sessies er op een bepaald moment bestaan. Als een sessie wordt beëindigd, wordt dat geadministreerd, en als op een bepaald moment alle sessies beëindigd zijn op een SVC, kan de SVC worden gecleard.
- Anderzijds is de volgende, veel eenvoudiger te implementeren strategie mogelijk. Bouw een SVC op wanneer de eerste message in een bepaalde richting wordt aangeboden. Indien regelmatig verkeer wordt aangeboden voor een SVC, laat dan de verbinding bestaan. Pas indien er gedurende een bepaalde tijd (bijvoorbeeld enkele minuten) geen verkeer meer is aangeboden, mag de SVC worden gecleard. Overigens gebeurt dit volledig transparant voor de gebruiker: naar buiten toe laat de PCU niet merken dat de SVC gecleard is, en als er toch opnieuw data worden aangeboden, dan wordt de SVC weer snel hersteld.

De laatstgenoemde methode heeft echter wel een aantal nadelen. Het is de bedoeling dat het opzetten en verbreken van SVC's geheel transparant voor de gebruiker geschiedt. Er kunnen echter problemen ontstaan als het (her)opzetten van de call niet meer lukt (bijvoorbeeld te wijten aan onvoldoende resources)? Een ander nadeel is, dat op het moment dat de gebruiker uitlogt, de SVC gewoon blijft staan, in afwachting van het moment waarop de timeout afloopt.

Het nadeel van de eerste methode is echter, dat er sessies zijn (me name die met SSCP's) die de ganse dag op staan, maar vrijwel geen verkeer hebben. De bijbehorende slecht benutte SVC's zouden een onaanvaardbaar beslag kunnen leggen op de beschikbare faciliteiten voor SVC's. Toch valt dat in de praktijk wel mee. We moeten onderscheid maken tussen de verschillende soorten sessies: de LU-LU, de SSCP-LU, de SSCP-PU en de SSCP-SSCP sessies.

- De LU-LU sessies bestaan alleen zolang er de behoefte bestaat om userdata te uit te wisselen. Uit de aard der zaak worden ze dus relatief efficiënt gebruikt.

De SSCP-LU en SSCP-PU sessies lopen alle van één punt naar de bijbehorende host, en benutten dus met elkaar slechts één SVC. Bovendien kan deze sessie vaak gedeeld worden met LU-LU sessie die dezelfde route nemen. Alleen als er geen sprake is van LU-LU sessies op de SVC waarop de SSCP-LU/PU sessies verlopen is er sprake van slecht benutte SVC's.

Tenslotte de SSCP-SSCP sessie. In het geval van terminal - host communicatie wordt er een SSCP-SSCP sessie opgezet tussen de desbetreffende (applicatie) host en de host die de terminal beheert. Deze sessie is voornamelijk van belang bij het opzetten en afbreken van de terminal - host verbinding. Gedurende de levensduur van de LU-LU sessie wordt er met de SSCP-SSCP sessie, die wel blijft bestaan, niet veel meer gedaan. Bovendien zal het niet bepaald uitzonderlijk zijn, dat er een SVC speciaal moet worden opgezet voor één SSCP-SSCP sessie. In veel gevallen is er dus sprake van een slecht benutte SVC.

Het zou overwogen kunnen worden, de SSCP-SSCP sessie niet een eigen SVC te geven, maar deze te laten verlopen over de reeds bestaande of op te zetten SVC's voor de SSCP-LU sessie en de LU-LU sessie.

### 3.4.3 Samenvatting van alle functies van de PCU

In de vorige paragraaf hebben we de belangrijkste functie van de PCU uiteengezet: het verzorgen van de routing van SNA messages door een X.25 netwerk. In deze paragraaf zullen we proberen naast deze hoofdfunctie de overige functies van de PCU uiteen te zetten.

PCU's vervullen de volgende functies:

1. Conversie SDLC naar X.25.
2. Vertalen SNA-adres naar X.25 adres voor data van SNA-node naar netwerk.
3. Inpakken SNA data in X.25 informatie-pakketten.
4. Uitpakken van X.25 informatie-pakketten en reproduceren SNA-pakket.

5. Concentratie en expansie.
6. Local Switching.
7. Calls opzetten en afbreken.
8. Bijhouden van boekhouding van calls en bekende SNA/X.25 adresparen.
9. Error recovery.

#### 3.4.3.1 Conversie SDLC naar X.25

In een groot aantal gevallen zal het zo zijn, dat een cluster controller uitsluitend beschikt over een SDLC aansluiting naar het netwerk. In dat geval zal de eerste taak van de PCU zijn, het converteren van SDLC naar X.25. Dit gebeurt dan op de al vaak in verkrijgbare apparatuur toegepaste manier van inpakken van SNA PIU's (pakketten op transmission level) in X.25 informatie pakketten. Deze inpakwijze komt er overigens op neer, dat de uiteindelijk door het X.25 te transporteren pakketten twee netwerk adressen kennen. In de eerste plaats het SNA adres, dat zich bevindt in de Transmission Header (de transmission header is het eerste deel van de PIU). In de tweede plaats het X.25 adres, dat zich bevindt in de packet header van het call request packet.

#### 3.4.3.2 Vertalen SNA-adres naar X.25 adres voor data van SNA-node naar netwerk

De pakketten komen vanaf de aangesloten SNA-nodes binnen over hetzij een X.25 aansluiting dan wel een SDLC aansluiting. Voor SNA-nodes die via een SDLC verbinding zijn aangesloten op de PCU, geldt het volgende. In de meeste gevallen (met uitzondering van local switching) moeten de pakketten de PCU via de X.25-net aansluiting verlaten, maar niet dan nadat zij zijn voorzien van een X.25 netwerk adres.

De PCU accepteert het binnenkomende SNA-pakket, kleedt het uit totdat hij slechts een PIU over heeft. Op dit niveau is het onderscheid tussen X.25 aansluitingen en SDLC aansluitingen dan ook verdwenen. De Path Information Unit bestaat uit een Transmission Header gevolgd door een BIU (Basic Information Unit).

De BIU kan voor de PCU interessant zijn, in het geval dat de PCU het verloop (opzetten en afbreken) van sessies gadeslaat. De transmission header is in ieder geval belangrijk: deze bevat het bestemmingsadres (DAF: Destination Address Field), zowel als het zendende adres (OAF: Originating Address Field). Het DAF is hiervan het meest belangrijk: het moet door de PCU geconverteerd worden naar een X.25 adres.

De PCU houdt een tabel bij, waarin genoteerd staat, welke call actief zijn, en welke SNA adressen op bepaalde calls gemapt kunnen worden. Als het DAF daarin voorkomt, kan met behulp van deze tabel de conversie naar X.25 adres gemaakt worden.

Zoniet, dan moet het SNA adres opgezocht worden in de algemene omzettingstabel. In principe kan deze omzettingstabel SNA-adres/X.25 adres centraal dan wel perifeer opgeslagen worden. Bij centrale opslag van de omzettingstabel is er een Address Conversion Database (ACD), die geraadpleegd kan worden door de PCU's. De PCU moet daartoe contact zoeken (call opzetten) met de ACD om het benodigde SNA adres te kunnen opvragen. Met het aldus gevonden adres moet eerst een call worden opgezet. Tevens moet de boekhouding van de PCU worden bijgewerkt.

#### 3.4.3.3 Inpakken SNA data in X.25 informatie-pakketten

Wanneer het corresponderende X.25 adres eenmaal bekend is, wordt een X.25 informatiepakket samengesteld, dat een informatie veld bevat dat gevuld wordt met de PIU. Dit betekent dat het SNA netwerk adres meegezonden wordt, ondanks het feit dat het SNA-adres al geconverteerd is naar een X.25 adres. Dit heeft het voordeel, dat straks het terugconverteren heel eenvoudig wordt (gooi de verpakking gewoon weg) en als tweede voordeel, dat het mogelijk wordt meerdere SNA-adressen te mappen op een en hetzelfde X.25 adres.

Het aldus geconstrueerde X.25-pakket wordt aangeboden aan het X.25 netwerk, dat ervoor zorg draagt dat het uiteindelijk bij de PCU terecht komt waarop de bestemmings-node is aangesloten.

#### 3.4.3.4 Uitpakken van X.25 informatie-pakketten en reproducere SNA-pakket

Bij de bestemmings-PCU aangekomen, wordt het X.25 pakket ontdaan van alle X.25 informatie, zodat uitsluitend het information field overblijft. Dit information field bevat de oorspronkelijke SNA PIU, die de complete SNA-adresinformatie bevat, zodat de PIU meteen aan de ComC kan worden aangeboden.

#### 3.4.3.5 Concentratie en expansie

Men kan PCU's maken waarop meerdere SNA nodes kunnen worden aangesloten. De PCU fungeert dan als concentrator. Dit heeft als consequentie dat meerdere ComC-adressen worden gemapt op het zelfde X.25 netwerk adres. De PCU moet dan voor binnenkomende pakketten gaan bepalen voor welke ComC zij bestemd zijn (expansie). Dit kan op twee manieren:

- Maak een unieke mapping van SNA-adressen op X.25 subnummers. De PCU wordt nu bereikt via het X.25 adres zonder subnummer, terwijl de PCU zelf het subnummer gebruikt om de expansie. De mappingsinformatie is nu opgeslagen in de ACD.
- De PCU bezit een tabel waarin SNA-adressen gemapt worden op de ComC die ze moet afhandelen. In de PCU komen pakketten aan met verschillende bestemmingen, met hetzelfde X.25 nummer, die door de PCU worden geschicht op grond van hun SNA-adres.

Beide oplossingen hebben hun voor- en nadelen. Een voordeel van de eerste oplossing is, dat op slechts één plaats SNA-adressen worden gemapt op X.25 adressen: in de ACD. Indien aan een bepaalde ComC een redelijk complex netwerk is verbonden, met veel verschillende bestemmingen die via de ComC te bereiken zijn, is het onhandig al die informatie op twee plaatsen opgeslagen te hebben. Een voordeel van de tweede oplossing is, dat de X.25 subnummers niet gebruikt worden, en dus vrij blijven voor andere toepassingen.

#### 3.4.3.6 Local Switching

Indien op een PCU meerdere SNA-nodes zijn aangesloten, biedt de PCU de mogelijkheid eventuele communicatie tussen deze locale nodes af te handelen via local switching.

#### 3.4.3.7 Calls opzetten en afbreken

Zie paragraaf 'Routing door PCU's'.

#### 3.4.3.8 Bijhouden van boekhouding van calls en bekende SNA/X.25 adresparen

De PCU moet een boekhouding bijhouden van de volgende gegevens.

Permanente gegevens:

1. Tabel van X.25 subnummers vs. ComC's. Indien een PCU meerdere ComC-aansluitingen heeft, kan door middel van een X.25 subnummer worden aangegeven welke ComC de bestemming zal moeten zijn.
2. Of: De SNA-adressen van de aangesloten nodes. Indien meerdere ComC's aangesloten zijn op één PCU, terwijl geen gebruik wordt gemaakt van bovengenoemde subnummers, moet de PCU toch kunnen bepalen welke ComC de bestemming is van een pakket. Derhalve moet de PCU kennis bezitten over de adressen van de verschillende ComC's.

3. Indien een centrale database (ACD) gebruikt wordt: X:25 adres van de ACD.
4. Anders: Adresconversietabel SNA/X.25 adressen.

Dynamische gegevens:

1. Tabel van actieve calls, met aan iedere call toegevoegd de SNA-adressen die ermee corresponderen en de tijd die verstrekt is sinds het laatste pakket dat gebruik maakte van de call. Ook de richting waarin de call was opgezet moet worden bijgehouden.
2. Tabel van niet-actieve calls, dat zijn calls die inmiddels zijn gecleared, maar waarvan de gegevens nog worden gehandhaafd.

#### 3.4.3.9 Error recovery

Tenslotte moet de PCU in staat zijn bepaalde error recovery procedures af te handelen, in het bijzonder de X.25 pakket laa reset procedure.

Het is een bekend fenomeen dat SNA apparatuur, die is uitgerust met PSI, zich niet kan herstellen van het optreden van resets in het X.25 netwerk. De PCU kan een nette reset-recovery als toegevoegde waarde bieden.

#### 3.4.4 ACD (Address Conversion Database)

De adresconversietabel SNA-adres/X.25-adres kan hetzij gecopiëerd worden naar alle PCU's, zodat iedere PCU zijn eigen copie van de tabel heeft, dan wel centraal worden opgeslagen, en onderhouden, in een ACD.

Indien een ACD wordt toegepast, is de volgende beschrijving van toepassing:

De belangrijkste taak van de ACD is het beschikbaar stellen van paren X.25-adres/SNA-adres, kortweg paren, aan de PCU's. Het is mogelijk, dat X.25 adressen met subnummering worden toegepast, maar het is ook mogelijk dat verschillende SNA adressen worden gemapt op hetzelfde X.25 adres.

De ACD heeft de mogelijkheid, wijzigingen aan te brengen in de database (toevoegen, wijzigen, verhuizen, etc.) terwijl het netwerk operationeel is. Om dit mogelijk te maken, houdt de ACD precies bij, welke PCU's een copie hebben van een paar uit de database. Moet een bepaald gegeven gewijzigd worden waarvan copien liggen bij een of meer PCU's, dan worden deze eerst geïnstrueerd, hun eventuele daarop gebaseerde calls af te breken en hun copie van het paar te vernietigen. Wanneer PCU's paren vernietigen, hetzij vrijwillig hetzij onder dwang van de ACD, dan stellen zij de ACD hiervan onverwijld in kennis. Aldus kunnen de copien van de database

steeds in overeenstemming met de centrale database worden gehouden.

### 3.4.5 Noodzaak van adresconversie

De hierboven beschreven netwerkopzet maakt gebruik van PCU's om SNA-adressen te converteren naar X.25 adressen, zodat de communicatie rechtstreeks, dat wil zeggen via één 'hop' kan gebeuren. Verder worden calls naar believen opgezet en afgebroken. Deze strategie heeft een economische grond: Er wordt op deze wijze voordelig omgesprongen met het gebruik van resources in de vorm van transportcapaciteit, en een aantal andere factoren.

In onderstaande tabel zullen we een aantal van deze factoren nader berekenen. We zullen naast elkaar zetten het resource-gebruik zonder toepassing van PCU's en het gebruik met toepassing van PCU's. Ter vergelijking gaan we uit van een netwerk bestaande uit drie locaties:

- o een locatie waar zich de applicatie-host bevindt;
- o een locatie waarin zich de terminal bevindt alsmede een ComC;
- o een locatie waarin zich de SSCP-host van de terminal bevindt. [listelement]...

In de traditionele situatie zijn de terminal-ComC en de SSCP-host via een X.25 call met elkaar verbonden, evenals de applicatie-host met de SSCP-host.

	zonder pcu	met pcu
aantal calls in rust	2	0
aantal calls bij opzetten	2	tot 4
aantal calls in werking	2	1
te transporteren verkeer	200 %	100 %

Wanneer zonder PCU gewerkt wordt, zijn steeds 2 calls nodig ongeacht in welke fase het systeem verkeert. De PCU maakt call overbodig als er geen verkeer is, en weet de hoeveelheid calls te reduceren tot één als de session eenmaal is opgezet. Daar staa tegenover dat in de opbouwfase het aantal calls kan oplopen tot vier, maar dat is een kortdurende situatie.

Het meest dramatische verschil is gelegen in de hoeveelheid verkeer die door het X.25 netwerk gestuurd wordt. Zonder PCU moet alle user-data tweemaal door het netwerk: van terminal naar zijn eigen ComC, en van daar naar de applicatiehost. De hoeveelheid verkeer kan men dus omschrijven als 200 % van de aangeboden verkeersdichtheid. De PCU maakt deze omweg overbodig, zodat de hoeveelheid verkeer, die het netwerk moet verwerken, effectief 100 % wordt.



## HOOFDSTUK 4

### SPECIFICATIE PCU

Dit hoofdstuk geeft een specificatie van de functies van de PCU, Protocol Conversion Unit. De PCU is een apparaat, dan wel stuk software, waarin de functies van een X.25-netwerk worden uitgebreid met laag 4 faciliteiten. Bovendien wordt SDLC naar X.25 protocolconversie aangeboden. De PCU kan gezien worden als onderdeel van het in het vorige hoofdstuk besproken SNA-netwerk, maar de applicaties van de hier gepresenteerde PCU zijn zeker niet tot dit netwerk beperkt.

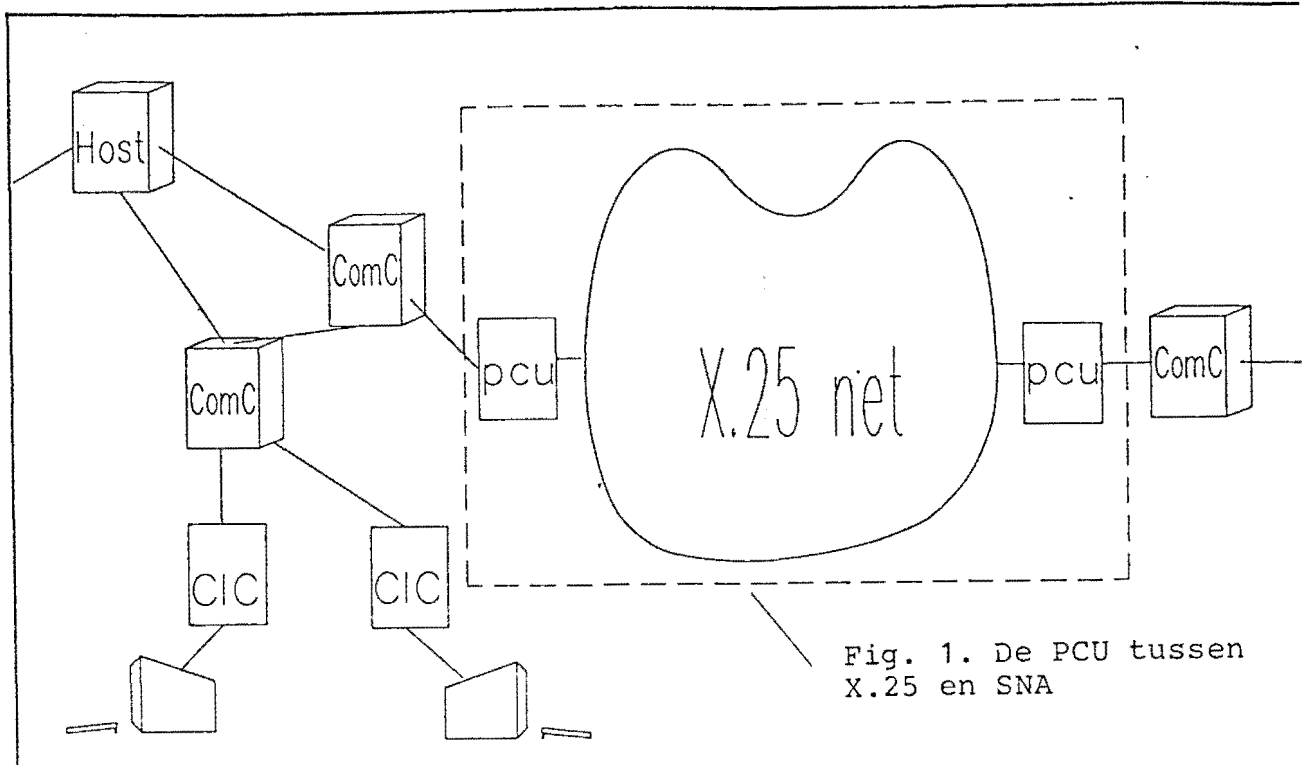
De PCU is een eenheid die kan worden ingezet op X.25 verbindingen tussen Communication Controllers (ComC's) en/of Cluster Controllers (ClC's) in een SNA-netwerk. De PCU voegt bepaalde functies toe aan de functies die het X.25-net biedt. De belangrijkste hiervan zijn: SVC-management, error recovery, SDLC en QLLC ondersteuning.

#### 4.1 SPECIFICATIE VANUIT HET GEZICHTSPUNT VAN DE GEBRUIKER

##### 4.1.1 Strekking van dit gezichtspunt

Laten we beginnen een toepassing van de PCU uiteen te zetten. Men stelle zich een situatie voor, waarbij twee ComC's zich in twee verschillende plaatsen bevinden, bijvoorbeeld den Haag en Eindhoven. Deze twee ComC's moeten verbonden worden, en er wordt op grond van kostenoverwegingen (zie paragraaf kostenaspect) en correcte werking voor gekozen de verbinding aan te leggen met X.25 en daaraan toegevoegd een paar PCU's. Zowel in den Haag als in Eindhoven is nu een PCU nodig. Deze PCU bevindt zich logisch gezien tussen het netwerk en de ComC, zodat de situatie ontstaat die in figuur 1 is uitgebeeld.

Het eerste deel van dit hoofdstuk behandelt de gedragingen van de verbinding PCU - X.25-net - PCU, zonder in te gaan op de daadwerkelijke functie die beide PCU's hierin vervullen, zonder in te gaan op de protocollen die gebruikt worden tussen de PCU's, en zeker zonder in te gaan op de inwendige details van de PCU.



Wel zal worden beschreven welke functies de gebruiker mag verwachten van twee PCU's plus X.25-net, in welke omstandigheden deze configuratie is toe te passen en wat de voordelen zijn van deze oplossing boven andere oplossingen.

De strekking van dit deel is het gezichtspunt dat uitgebeeld wordt door de streeplijn in figuur 1. Wat binnen de streeplijn zit beschouwen we voorlopig als 'black box'.

#### 4.1.2 Configuraties

Voor de in vorige paragraaf gestelde omstandigheden zijn traditioneel een aantal configuraties denkbaar.

##### 4.1.2.1 SDLC

De verbinding tussen twee delen van een SNA-netwerk, in dit geval ComC's, kan in de eerste plaats worden gerealiseerd met een lease line. Over die leased line wordt dan het SDLC protocol gebruikt. Deze oplossing is alleen de goedkoopste indien er sprake is van een regelmatige, grote hoeveelheid data die getransporteerd moet worden over de verbinding.

In de tweede plaats is het mogelijk een SDLC verbinding te maken over een geschakelde telefoonlijn. Nadeel hiervan is, dat de betrouwbaarheid relatief gering is en dat de snelheid een beperkende factor kan zijn. Voorts is ook deze oplossing (afhankelijk van de omstandigheden) vaak niet de goedkoopste.

#### 4.1.2.2 QLLC

In de derde plaats is het mogelijk X.25 te gebruiken als netwerk. Men kan hiervoor QLLC software huren, die op de ComC wordt geïnstalleerd, dan wel voor een hardware oplossing kiezen. Voor een verbinding die continu aanwezig moet zijn (vergelijkbaar met een leased line) kan men een PVC (Permanent Virtual Circuit) opzetten terwijl een verbinding die slechts enkele uren per dag wenselijk is het beste gerealiseerd kan worden middels een SVC (Virtual Call).

In dit verband zijn er twee belangrijke nadelen van de bestaande hard- en software. Er wordt gebruik gemaakt van QLLC als protocol over X.25. Het eerste nadeel van QLLC betreft error recovery. De netwerk laag van X.25 levert foutvrij transport in zoverre, dat er nooit fouten optreden zonder dat daarvan een melding wordt gegeven naar de bovenliggende laag. Die melding komt in de vorm van een reset pakket. De bedoeling van dit reset pakket is, dat de bovenliggende laag een error recovery procedure start. QLLC implementeert echter geen error recovery, maar reageert (enigszins prematuur) op een reset met het melden van een fatale link error naar zijn superieuren, waardoor alle bestaande SNA sessies over die link automatisch worden beëindigd.

Het andere nadeel betreft de afweging die men verplicht wordt te maken voor het kiezen tussen PVC en SVC. Een SVC heeft het voordeel dat het goedkoper is dan een PVC indien er maar een paar uur per dag gebruik wordt gemaakt van de verbinding. Anderzijds moet een SVC speciaal (door de operator) worden opgezet op het moment dat iemand er gebruik van wil maken: minder flexibel dus.

Verder zijn internationale PVC's momenteel veelal nog niet mogelijk, zodat er gekozen moet worden voor de duurdere of onhandigere opzet met SVC's.

#### 4.1.2.3 PCU

Op deze twee punten is er behoefte aan apparatuur die deze bezwaren ondervangt, de PCU. In de eerste plaats wordt een error recovery faciliteit toegevoegd aan QLLC, zodat door X.25 gerapporteerde fouten worden opgevangen, zonder dat dit fatale gevolgen heeft voor SNA-sessies. In de tweede plaats zorgt de PCU ervoor dat gebruik kan worden gemaakt van SVC's, zonder dat het ten koste gaat van de flexibiliteit. De PCU zet automatisch een call op als er gebruik wordt gemaakt van de verbinding, terwijl die call ook weer automatisch wordt beëindigd op het moment dat de verbinding niet meer nodig is.

#### 4.1.3 Kostenaspect

In deze paragraaf willen we aantonen, dat het uit kostenoverweging in vele gevallen goedkoper is om een PCU in te zetten. De PCU maakt het mogelijk gebruik te maken van SVC's, zonder dat de gebruiker daar iets van merkt.

Ten grondslag aan deze vergelijking ligt de tariefstructuur van het X.25-netwerk. Als voorbeeld gaan we uit van het Nederlands datanet. De kostenvergelijking draait voornamelijk uit op het vergelijken van het vaste tarief dat voor een PVC moet worden neergelegd, vergeleken met het 'tijdtarief' dat voor een SVC betaald moet worden (zie appendix A).

Het tarief voor een PVC bedraagt f. 100,- per maand. Het tarief voor een VC is 2,5 cent per minuut. Hierin zijn toegangskosten tot het net en de kosten voor transport (bedrag per segment) niet begrepen.

Nu volgt een gedetailleerde uitwerking van e.e.a.

##### 4.1.3.1 Uitgangspunten

- o Er is sprake van een netwerk met  $n$  ComC's, die volledig vermaast met elkaar verbonden moeten worden, in het ene geval door middel van PVC's, in het andere geval door middel van PCU's.
- o Iedere ComC maakt dagelijks gebruik van de verbinding met gemiddeld  $q$  andere ComC's ( $q < n$ ). Ieder van die  $q$  'communicatie-bursts' heeft gemiddeld een duur van  $h$  uren. Let wel: iedere periode waarin geen verkeer is en die langer duurt dan pakweg een minuut telt \*niet\* mee voor  $h$ !
- o De kosten voor het opzetten van een SVC worden verwaarloosd ten opzichte van de kosten per minuut van de SVC.
- o Er zijn 20 werkdagen per maand waarop deze bursts plaatsvinden; de overige dagen is er geen verkeer.
- o In het alternatief waarin geen PCU's worden toegepast, is de goedkoopste oplossing het opzetten van PVC's tussen alle mazen.
- o We nemen voorts aan dat de huurprijs van een PCU gelijk is aan de huurprijs van X.25 software in een ComC, zijnde f.600,- per maand (excl. BTW), zodat deze kostenfactor in de vergelijking niet hoeft worden meegenomen.
- o We nemen voorts aan, dat de hoeveelheid verkeer, gemeten in segmenten, niet verschilt in beide alternatieven, zodat de verkeerskosten voor de afweging geen rol spelen. Voor het totale kostenplaatje gaan we echter uit van een verkeersdichtheid per verbinding per uur ter grootte  $A$  (segmenten/uur).

Opmerking. Voor een terminal-host verbinding zou men kunnen rekenen met de volgende getallen: Uitgaande van een blockmode-terminal waarop 6 schermen per minuut worden vertoond, waarop steeds een user-respons komt. De schermen vergen 1200 byte per stuk, dat is dus 20 segmenten van 64 byte. De respons komt op 200 byte; dat is 4 segmenten. Verkeersdichtheid A wordt nu:

$$A = 6 \times 60 \times (4 + 20) = 8640 \text{ segmenten/uur}$$

- o We gaan uit van de nederlandse tarieven voor datanet verkeer (zie appendix A).

Met behulp van deze uitgangspunten zullen we de kosten per ComC, anders gezegd de totale kosten gedeeld door n, berekenen voor beide alternatieven. Eerst voor de situatie met PVC's, vervolgens wanneer PCU's worden ingezet, die SVC's mogelijk maken.

#### 4.1.3.2 Kostenberekening met PVC's

Post	Bedrag per maand per ComC in f.
X.25 software in ComC	600
Toegang tot het net (2400 b/s)	400
Kosten (n-2) extra logische kanelen (f. 5,= per kanaal)	5(n-2)
PVC kosten maandelijks f100,= per PVC	$\frac{1}{2} * 100 (n-1) = 50(n-1)$
Volumekosten f0,0025 per segment	$\frac{1}{2} * Ahq * 0,0025 * 20 = 0,025 Ahq$
<b>Totaal</b>	<b>940 + 55 n + 0,025 Ahq</b>

#### 4.1.3.3 Kostenberekening met PCU's

Post	Bedrag per maand per ComC in f.
Huurkosten PCU	600
Toegang tot het net (2400 b/s)	400
Kosten (n-2) extra logische kanelen (f. 5,= per kanaal)	5 (n-2)
Tijdtarief SVC's f0,025 per minuut	$\frac{1}{2} * 0,025 * 60 * 20hq = 15 hq$
Volumekosten f0,0025 per segment	$\frac{1}{2} * Ahq * 0,0025 * 20 = 0,025 Ahq$
<b>Totaal</b>	<b>990 + 5 n + 15 hq + 0,025 Ahq</b>

4.1.3.4 Afweging PVC/PCU

Zoals in de twee bovenstaande optelsommen te zien is, is de ene post die verschilt de post PVC tarief per maand resp. S tijdtarief.

We kunnen dan ook concluderen, dat gebruik van PCU's voordelig is indien:

$$10 * (n-1) > 3 * hq,$$

waarin (nogmaals), n het aantal ComC's in het netwerk, q h gemiddelde aantal ComC's waarmee iedere ComC dagelijks contact heeft, h de tijdsduur gedurende welke dit plaatsvindt.

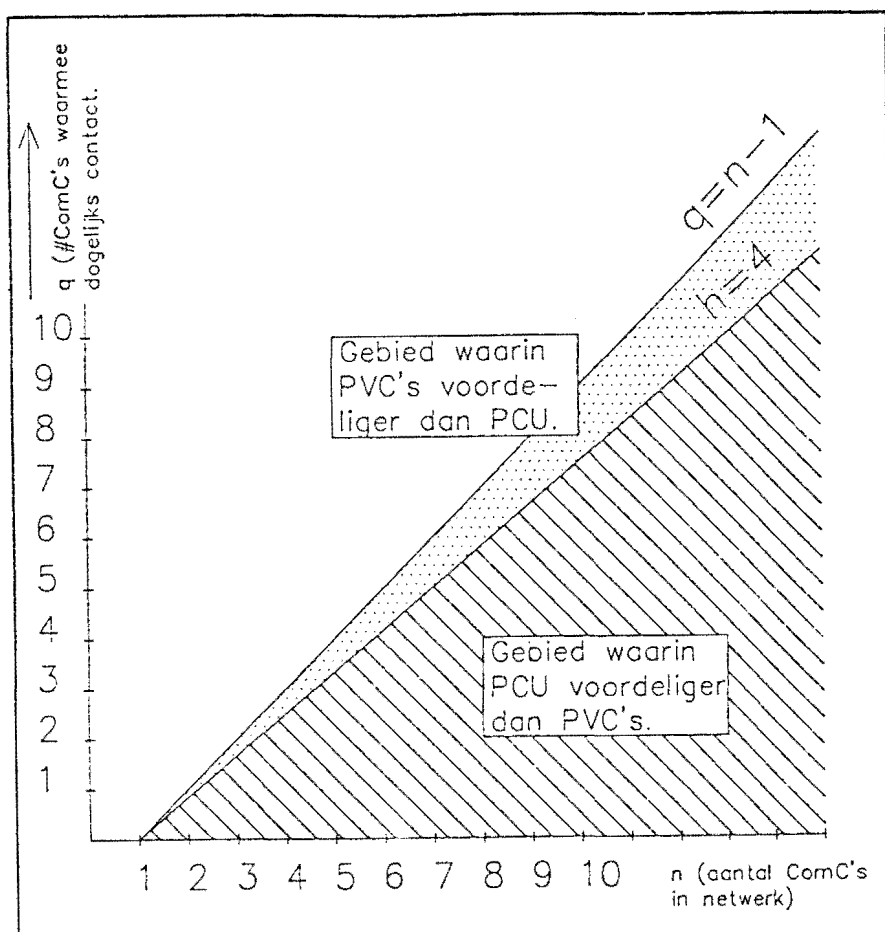


Fig. 2. Afweging PVC tegen PCU.  
 Gegeven het aantal ComC's, de connectiviteit q en uren per dag h geeft de grafiek de beste optie

Voor  $h = 4$  uur, is in figuur 2 uitgezet welke combinaties van  $n$  en  $q$  vragen om toepassing van een PCU, respectievelijk beter m.b.v. PVC's kunnen worden opgelost. Voor kleinere  $h$  verdraait de scheidslijn linksom. Bij  $h < 3,3$  uur is toepassing van PCU's in alle gevallen voordeliger.

#### 4.1.3.5 Internationale aspecten

In het internationale verkeer zijn vaak (nog) geen PVC's mogelijk, zodat men verplicht is met SVC's te werken. Indien men (binnen Europa) een SVC 7 uur per dag, 20 dagen per maand wil gebruiken, bedragen de kosten hiervan plm f. 800 per maand. De post 50 ( $n-2$ ) hierboven verandert hierdoor in 400 ( $n-1$ ). Dit voegt een extra argument toe voor het gebruik van PCU's.

#### 4.1.4 Configuraties met de PCU

De PCU is in een aantal verschillende omstandigheden toe te passen. De belangrijkste verschillen zijn:

- o ComC/Cluster Controller (ClC) dan wel ComC/ComC verbinding
- o Een van de volgende vier mogelijkheden voor de toepassing van multidropping op SDLC verbindingen:
  - o Point to point verbinding (geen multidropping).
  - o Multidrop end-to-end: Twee PCU's die aan beide locale zijden een multidrop aansluiting hebben.
  - o Multidrop aan de ene kant, de PCU aldaar mapt echter de verschillende secondary stations op verschillende X.25 adressen, zodat aan de andere zijde van de verbinding de PCU's zonder multidrop verbonden zijn met hun SNA-knooppunten.
  - o Een combinatie van de laatste twee mogelijkheden.
- o Verbinding tussen ComC en PCU middels SDLC link dan wel QLLC verbinding.

In figuur 3 zijn zes verschillende (zes waarschijnlijke) configuraties uitgebeeld.

1. Point-to-point verbinding tussen ComC en ClC.

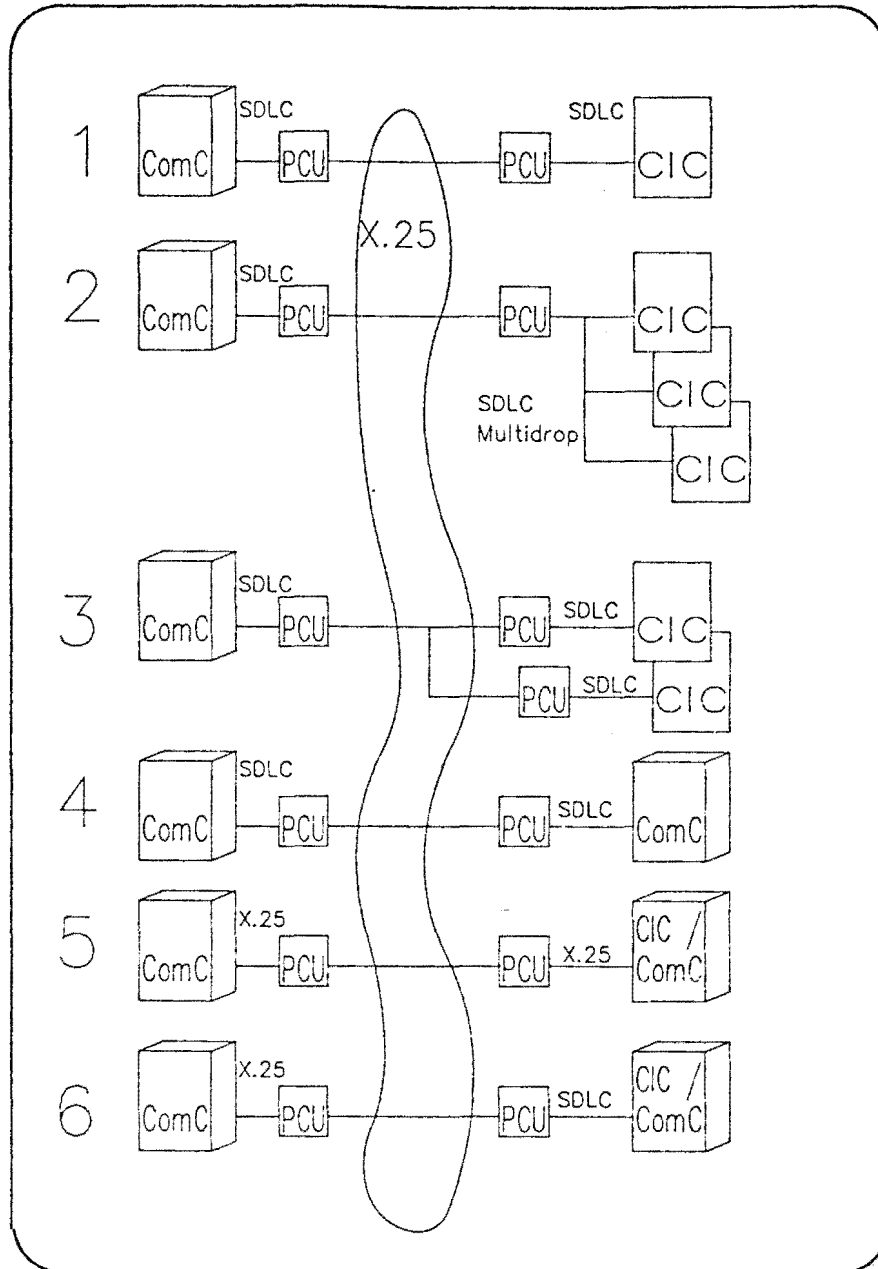


Fig. 3. Zes configuraties met de PCU.

2. Multidrop verbinding tussen ComC en een aantal CIC's, welke zich alle aan dezelfde PCU bevinden.



3. Multidrop verbinding tussen ComC en een aantal ClC's, welke zich aan verschillende X.25-net adressen en verschillende PCU's bevinden.
4. Point-to-point verbinding tussen twee ComC's
5. Verbinding tussen twee ComC's of tussen ComC en ClC, waarbij aan beide zijden QLLC wordt toegepast.
6. Verbinding tussen twee ComC's of tussen ComC en ClC, waarbij aan een van beide zijden QLLC wordt toegepast.

Er zijn nog meer configuraties denkbaar, maar deze komen steeds op dezelfde principes neer.

#### 4.1.5 Vertragingstijden in het netwerk

Hoeveel delay ontstaat er in een verbinding via ComC's, x.25, etc.? In de knooppunten, zoals ComC's, etc. op zich ontstaat niet veel vertraging. Echter wel in de links tussen de knooppunten. De grootste vertraging ontstaat in verbindingen met lage bitrate.

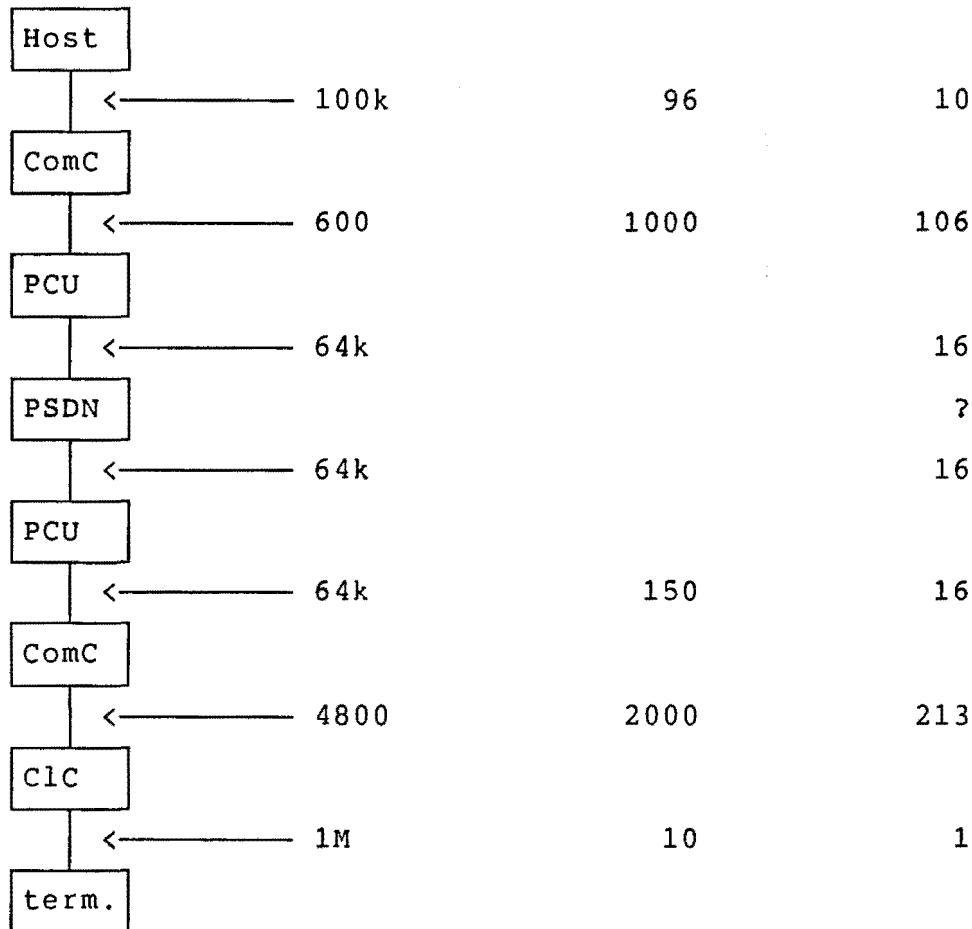
We zullen hier ingaan op de delaytijd in het interactieve verkeer terminal - host. Een typische grootte voor een 'plaatje', d.w.z. een beeldscherm vol tekst is 800 à 1200 bytes. Het oversturen van zo'n plaatje als geheel over bijvoorbeeld een 4800 b/s verbinding kost 2 seconde.

Laten we nauwkeuriger beschouwen hoeveel tijd het versturen van zo'n plaatje van 1200 bytes kost. Hiervoor gaan we uit van de volgende veronderstellingen.

- o Neem aan, dat de tijd, die in de nodes wordt doorgebracht verwaarloosbaar is ten opzichte van de tijd die nodig is voor transmissie over de links.
- o Voor het gemak van de berekening, verwaarlozen we verder de tijd die nodig is voor het versturen van headers. Hoe kleiner de pakketten, hoe groter deze tijd wordt.
- o Er vindt geen vertraging plaats door flow-control mechanismes
- o Indien het plaatje wordt verstuurd als één pakket, is de transmissietijd gelijk aan de som van de delaytijden van de links.

De transmissietijden van de verschillende links in een typische terminal - host verbinding staan vermeld in de volgende tabel, zowel voor een pakketgrootte van 1200 byte als voor een pakketgrootte van 128 byte. In het X.25 netwerk heeft het geen zin om pakketten van 1200 byte te beschouwen omdat er een maximum is van 128 byte.

Node	bitrate (b/s)	delay (ms) per 1200 byte	per pakket van... 128 byte
------	---------------	-----------------------------	-------------------------------



De transmissietijd van een pakket van 1200 byte bedraagt nu:  $t = 96 + 1000 + 16 + 16 + 150 + 2000 + 10 = 3288$  ms

Deze tijd is behoorlijk lang, zeker als men bedenkt, dat in het interactieve verkeer er altijd sprake is van een hoeveelheid data heen, en als reactie daarop, een hoeveelheid data terug, zodat de persoon achter de terminal tweemaal een dergelijke tijd moet wachten op antwoord van de host.

Het is mogelijk om een kleinere transmissietijd te bewerkstelligen, door het plaatje niet als één pakket te versturen, maar het te segmenteren. Dit kan in SNA gespecificeerd, worden, door RU-size (request unit: eenheid van user-data) op te geven. Hierdoor wordt de tijd voor het versturen van een segment kleiner, zodat een tussenliggende node het ontvangen pakket al weer door kan sturen voor het hele plaatje compleet is.

Men kan eenvoudig nagaan, dat de totale transmissietijd van een gesegmenteerde hoeveelheid data gelijk is aan de som van:

- o De totale transmissietijd van alle messages over de langzaamste link.
- o De som van de delaytijden van één message op de overige links.

In het geval dat de RU-size is beperkt tot 128 bytes, wordt ons plaatje van 1200 byte gesegmenteerd in 10 pakketten. De traagste link is de verbinding tussen ComC en ClC, waarover de transmissietijd 213 ms bedraagt. De totale transmissietijd bedraagt dus:

$$t = 10 + 106 + 16 + 16 + 16 + 10 * 213 + 1 = 2295 \text{ ms.}$$

Dit betekent een daling van de transmissietijd met 31%.

Hierbij moet worden aangemerkt, dat de overhead in de zin van message headers veel steeds belangrijker wordt naarmate de pakketten kleiner worden.

Conclusie mag zijn dat het aan te bevelen is, voor terminal - host verkeer een kleine RU-size in te stellen.

#### 4.2 INWENDIGE SPECIFICATIE

Terwijl het eerste deel van dit hoofdstuk zich bezig hield met de verschijningsvorm voor de gebruiker van een verbinding, opgezet met PCU's, willen we hier beschrijven op welke manier e.e.a. in de PCU geïmplementeerd wordt. Het gezichtspunt van dit hoofdstuk is uitgebeeld in figuur 4. De lezer vergelijk deze figuur met figuur 1.

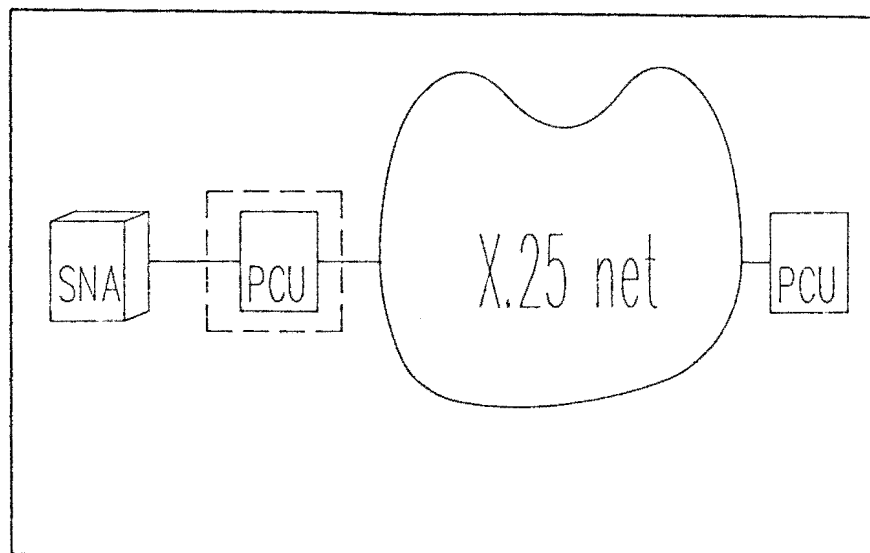


Fig. 4. Gezichtspunt van 4.2.

#### 4.2.1 Blokschema

De PCU kan worden beschreven in de vorm van het blokschema van figuur 5. Hierin kunnen twee helften worden onderscheiden: lokale functie en de netwerk functie.

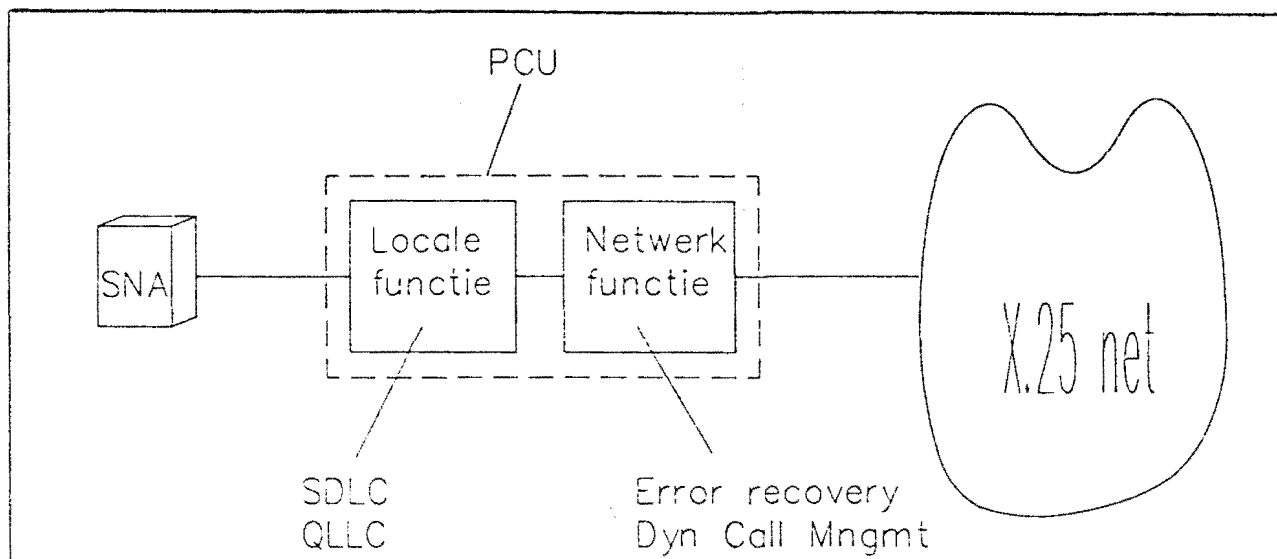


Fig. 5. Blokschema PCU.

De lokale functie verzorgt de verbinding met de ComC of CLO. Derhalve vinden we hierin terug het SDLC protocol, zowel als het QLLC protocol. Beide bestaan uit een aantal lagen, die hieronder verder aan de orde komen.

De netwerk functie verzorgt de verbinding met het X.25 netwerk. Voorts vinden we hier:

- o Error recovery protocol
- o protocol voor het opzetten en afbreken van X.25 calls: Dynamische Call Management (DCM)
- o Optioneel: Multiplexing protocollen

#### 4.2.2 Lagenmodellen

Een gedetailleerder inzicht bieden de volgende 'lagenplaatjes'. De volgende tekeningen moeten worden gezien als een schema in de geest van OSI. Ze voldoen echter in zoverre niet aan OSI, dat een (behoorlijk) aantal lagen in feite dubbel voorkomen. Zowel SNA als X.25 kent bijvoorbeeld een netwerk laag (al heet dat in SNA dan de Path Control Layer). Vanuit OSI gezien zijn alle SNA-lagen een invulling van de applicatie laag. Vanuit SNA gezien bieden de OSI-lagen slechts een link laag service. Vanuit dit 'wederzijdse onbegrip' is het verklaarbaar dat diverse lagen dubbel voorkomen in

de hieronder geschetste modellen.

In het eerstvolgende schema wordt de volgende situatie gemodelleerd.

Er is sprake van twee ComC's, die een verbinding opzetten middels hun PCU's, die op hun beurt onderling zijn gekoppeld door het PSDN (Public Switched Data Network), een X.25-netwerk. Voorts wordt er van uitgegaan dat de PCU en de ComC in beide gevallen met elkaar verbonden zijn door een SDLC verbinding.

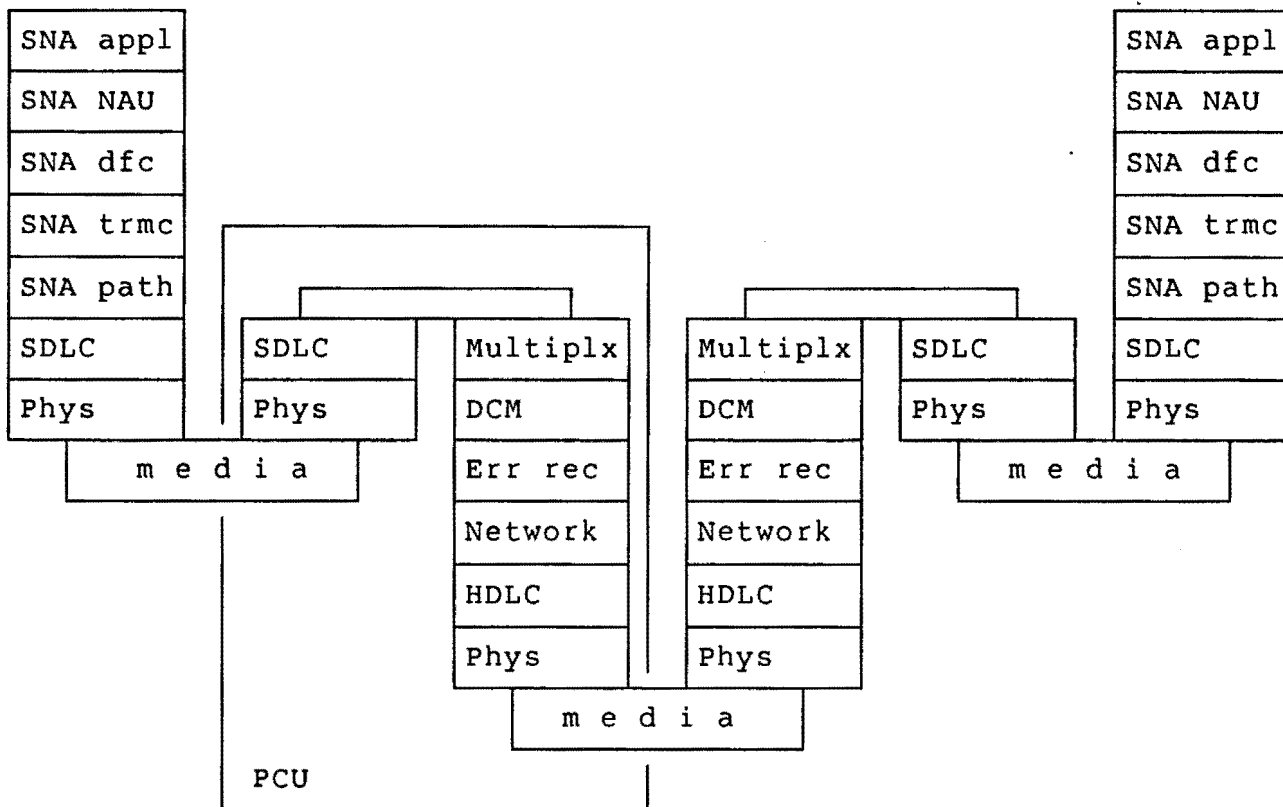
In het schema zijn 6 kolommen onderscheidbaar. De meest linkse (kolom 1) evenals de meest rechtse (kolom 6) representeert een ComC. De linker ComC heeft een verbinding (via de physical media, aangeduid als m e d i a) met kolom 2. Kolommen 2 en 3 modelleren de bij de ComC behorende PCU, aangegeven door een kader.

Kolommen 4 en 5 modelleren de PCU aan het andere eind van de verbinding. De PCU's onderling zijn verbonden door het X.25-netwerk, dat wegens plaatsgebrek is weggelaten. Slechts de m e d i a zijn aangegeven, maar men denke zich daartussen een compleet X.25-netwerk, dat nog lagen 1, 2 en 3 zal herbergen.

Kolom 6 tenslotte staat voor de ComC aan het verre eind van de verbinding, verbonden met kolom 5 door de m e d i a aldaar.

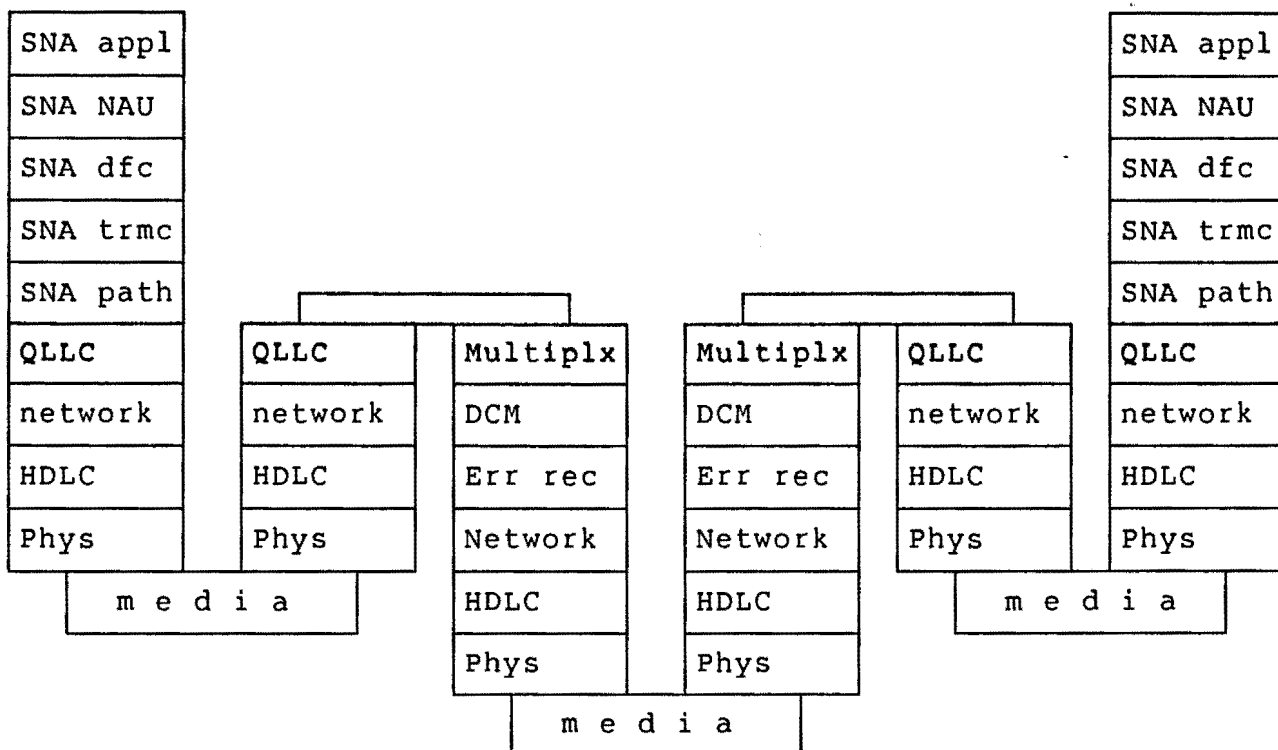
In de kolommen 3 en 4 vinden we de typische PCU-lagen terug, die moeten zorgen voor error recovery (err rec), het beheer van SVC's (DCM) en het multiplexen van meerdere verbindingen over een call (multiplx).

Men mag deze drie lagen beschouwen als een invulling van OSI laag 4: de transportlaag. Drie lagen betekent dus: drie sublagen.

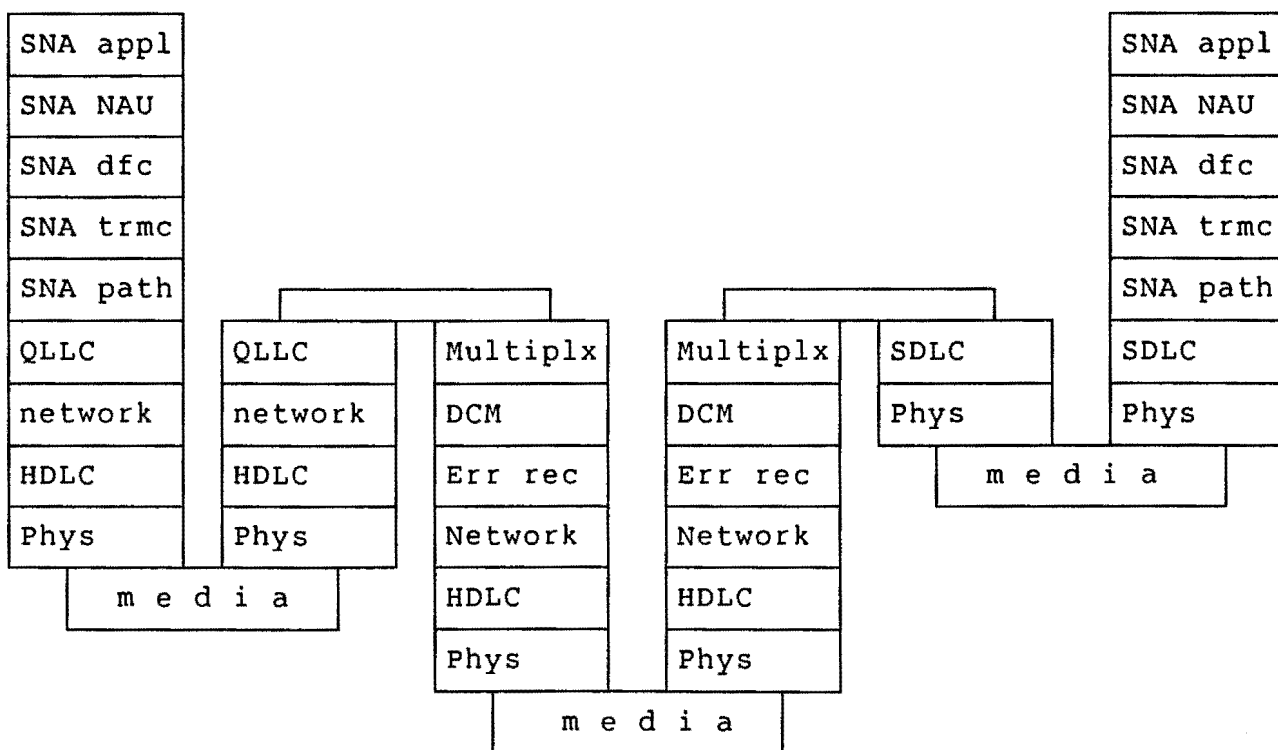


Zoals te zien in het schema, kolom 2, ondersteunt de PCU naar de ComC toe het SDLC protocol. Het SDLC protocol levert uiteindelijk de PIU (Path Information Unit). De PCU stuurt de PIU door naar kolom 3, die hem als TSDU (Transport Service Data Unit) beschouwt. Kolom drie voegt de Multiplexing, DCM- Error Recovery, netwerk-, link- en fysieke lagen toevoegt aan de PIU. Samen met de corresponderende lagen in de andere PCU garandeert dit een foutloze en effectieve overdracht van TSDU's. Aldaar wordt de PIU weer middels het SDLC protocol naar de andere ComC verstuurd.

In het volgende schema wordt uitgebeeld, hoe de situatie wordt, indien beide ComC's niet SDLC maar QLLC als protocol gebruiken. De PCU's aan beide einden ondersteunen hiertoe het QLLC protocol, en bieden eenzelfde interface naar de session laag in kolommen 3 en 4 van het schema.



Tenslotte is het ook mogelijk, aan de ene kant SDLC te gebruiken terwijl aan de andere kant QLLC wordt ingezet, er ontstaat dan het volgende, asymmetrische schema:



#### 4.2.3 De locale functie

De locale functie van de PCU ondersteunt zowel SDLC als X.25/QLLC. Beide protocollen zijn in staat meerdere links over een fysieke verbinding te realiseren. Bij SDLC heet dit fenomeen multidropping, hetgeen zich afspeelt op de link laag (dit is niet verwonderlijk, daar SDLC een link laag protocol is). Bij X.25/QLLC maken we gebruik van de netwerk-adresseringsmogelijkheden van X.25 (netwerk-laag), om meerdere calls over een verbinding te hebben (of door middel van PVC's).

De verschillende link laag verbindingen die ontstaan door multidropping en de verschillende calls die in X.25 verbindingen aanwezig zijn, geven we aan met het begrip 'locale kanalen'. Ieder van deze locale kanalen wordt door de PCU afgebeeld op een 'netwerk kanaal', dat is een verbinding met het X.25-netwerk. Hierover later meer.

##### 4.2.3.1 SDLC

SDLC kan in een aantal configuraties worden gebruikt:

- o Point to point

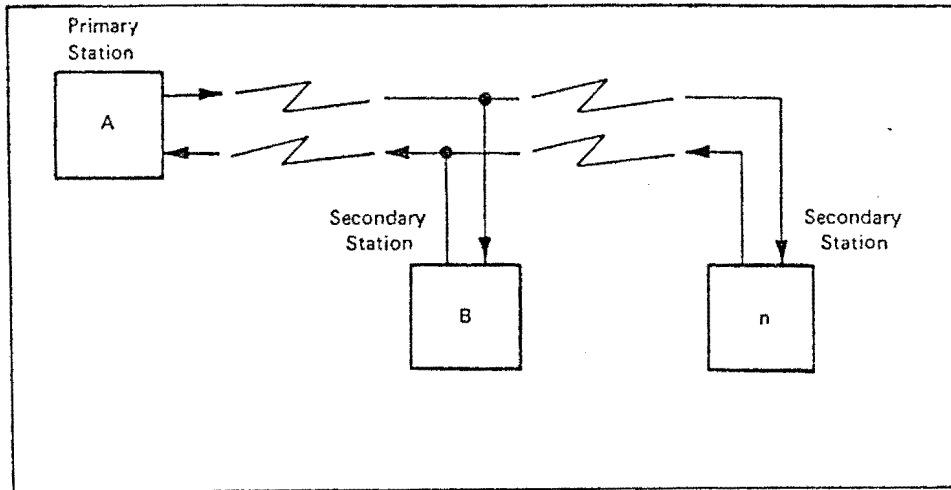
Hiermee is het mogelijk een verbinding te maken van ComC/ClC naar PCU die precies overeenkomt met één lokaal kanaal. Zoals bekend, kent een point-to-point SDLC verbinding altijd een primary en een secondary link station. Het primary link station is verantwoordelijk voor de link. In principe is er de keuze ofwel de ComC ofwel de de PCU aan te wijzen als primary link station. In verband met de compatibiliteit met de volgende opties verdient echter de configuratie ComC primary en PCU secondary de voorkeur.

- o Normale multidrop

Door middel van multidropping is het mogelijk om verschillende locale kanalen af te beelden over één fysieke verbinding. Dit is vaak wenselijk omdat een ComC meestal verbindingen onderhoudt met verschillende andere ComC's of ClC's. In dit geval moet de ComC het primary link station zijn en de PCU het secondary station. De andere ComC's/ClC's waarmee gecommuniceerd wordt, moeten als secondary link station fungeren, en kunnen dus niet multidroppen.

De PCU gedraagt zich dan als een verzameling secondary link stations. De verbindingen tussen primary link station en secondary link stations is uitgebeeld in figuur 6. De verschillende secondary link stations worden in SDLC geselecteerd door middel van een secondary link station adres.





Figuur 6. Multidropping

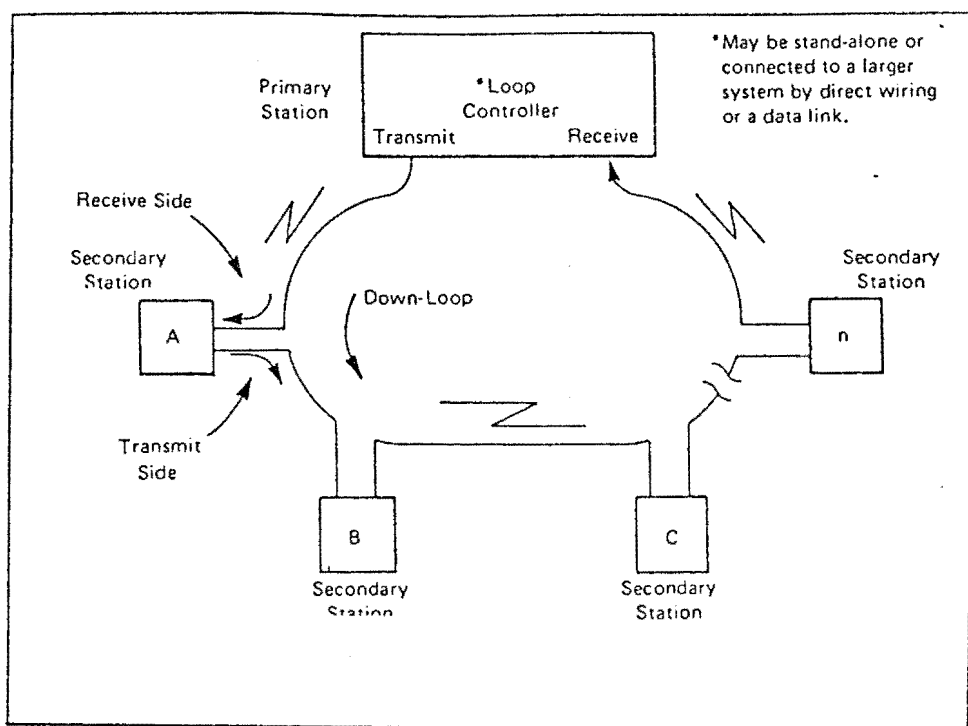
In een multidrop SDLC verbinding is, evenals in een point to point link het primary link station verantwoordelijk voor het regelmatig pollen van de secondary link stations. Dit gebeurt door regelmatig de secondary stations een RR (receiver ready) frame te sturen met het P/F-bit (Poll) op 1. Indien het secondary station data te versturen heeft, verstuurt het een aantal frames, waarvan alleen het laatste frame het P/F-bit (Final) op 1 heeft. Als het primary station dit 'final' pakket heeft ontvangen, kan het een ander station pollen.

De PCU zal nu, zoals gezegd, de verzameling secondary link stations van de SDLC verbinding emuleren. Op grond van het station adres kan de PCU aan binnenkomende frames zien, bij welk lokaal kanaal zij horen, terwijl de PCU aan uitgaande frames het station adres meegeeft om de ComC te laten weten welk secondary station respons geeft.

Het nadeel van deze configuratie is, dat bij een groot aantal secondary link stations (ofwel een groot aantal lokale kanalen over één SDLC verbinding), het primary link station al een grote hoeveelheid tijd kwijt is met het pollen van de secondary stations. Een en ander is afhankelijk van de transmissiesnelheid, de reactiesnelheid van de primary en secondary stations, en het aantal stations.

o Loop configuratie multidrop

Een andere vorm van multidropping is de loop configuratie. De secondary stations worden hierbij geschakeld zoals figuur 7 aangeeft. Er is een ringvormige verbinding tussen het primary en de secondary links stations, die in slechts een richting kan worden doorlopen.



Figuur 7. Multidrop loop configuration.

Het pollen gebeurt hier als volgt. Laten we er van uit gaan dat het primary station aan het zenden is. Zolang de primary aan het zenden is, sturen de secondary's alle ontvangen bits onmiddellijk door naar het volgende station in de loop. De Primary stuurt nu een UP (Unnumbered Poll) naar een, een aantal of alle secondary link stations. Een secondary mag echter slechts beginnen te zenden na ontvangst van een serie van achtereenvolgende bits gelijk aan 1. Daartoe begint het primary station op een bepaald moment met het uitsturen van uitsluitend 1'en (0111111111...)

Het eerste secondary station in de loop dat iets te versturen heeft stuurt het zevende bit echter niet door, maar stuurt in plaats hiervan een 0, zodat de volgende stations in de loop slechts een vlag (01111110) zien. Vervolgens stuurt het station zijn eigen data de loop in. Als het station uitgezonden is, laat het de serie 1'en passeren, zodat het mechanisme zich bij het volgende station kan herhalen. Een station dat niets te zenden heeft, laat alles onbekommerd passeren. Wanneer tenslotte de serie 1'en het primary station bereikt, heeft het alle dataframes van de secondary's ontvangen, en kan het zelf weer beginnen te zenden.

Ook in dit geval implementeert de PCU de verzameling secondary link stations. Het voordeel van de loop configuratie boven de recht-toe-recht-aan multidrop is, dat het pollen in de loop configuratie veel minder tijd kost: met één UP kunnen alle secondary stations bereikt worden.

#### 4.2.3.2 QLLC

Tussen ComC en PCU kan ook een X.25 verbinding worden gemaakt. X.25 wordt uitgebreid met een QLLC (Qualified Logical Link Control) laag, die equivalenten van SDLC commando's verstuurt.

Hier kan opgemerkt worden, dat de codering van QLLC van SDLC commando's zich waarschijnlijk prima leent als voorbeeld hoe de PCU SDLC frames kan afbeelden op transport laag frames.

SDLC commando's worden gedeeltelijk lokaal afgehandeld, terwijl die SDLC commando's die een end-to-end betekenis hebben, worden verstuurd als X.25 pakketten met het qualifier (Q) bit op 1. QLLC zorgt ervoor dat alleen frames die daadwerkelijk informatie bevatten (dus geen RR frames met P/F-bit=1, die uitsluitend dienen om te pollen) over de X.25 verbinding worden verstuurd. Het pollen wordt lokaal afgehandeld. Door het pollen lokaal af te handelen bespaart men een aanzienlijk deel op de X.25 kosten.

QLLC maakt een onderscheid tussen X.25 pakketten die userdata bevatten, en X.25 pakketten die SDLC commando's vertegenwoordigen. Dit onderscheid wordt aangegeven door gebruik van het X.25 Q-bit (Qualified). User data pakketten hebben Q=0, terwijl de SDLC commando's worden gecodeerd met Q=1.

De volgende functies worden ondersteund door QLLC:

- o SNRM (set normal response mode) wordt verstuurd als QSM (Q set mode)
- o DISC (disconnect) wordt QDISC
- o XID (exchange identification) wordt QXID
- o TEST (test logical link) wordt QTEST
- o UA (unnumbered acknowledgement) wordt QUA
- o RD (request disconnect) wordt QRD
- o RR (receive ready) wordt QRR. Wat overigens het nut is van dit pakket, en of het regelmatig gebruikt wordt, is niet duidelijk.
- o DM (disconnect mode) wordt QDM
- o FRMR (frame reject) wordt QFRMR

#### 4.2.4 De netwerk functie

In dit hoofdstuk worden de functies van de netwerzijde van de PCU beschreven. Deze functies spelen zich in het OSI model af op de netwerklaag. Refererend aan figuur 6, betreft het hier het rechterdeel van de PCU. De volgende deelfuncties kunnen worden onderscheiden:

- o Multiplexing
- o Dynamic Call Management
- o Error recovery

De genoemde drie functies kunnen worden beschouwd als deellagen van de transportlaag, zoals is aangegeven in de paragraaf 'lagenmodellen'.

De functie multiplexing zal in eerste instantie niet worden geïmplementeerd. De reden hiervoor is, dat de overige te implementeren functies een hogere prioriteit hebben en dat de PCU ook zonder multiplexing een volwaardige functie behoudt.

De functie error recovery mag men in dit verband interpreteren als het correct afhandelen van resets die de netwerklaag aanbiedt aan de transportlaag. Er is overwogen, het protocol ECMA-72, of X.224, klasse 2 te adopteren, met als voordelen, dat hiermee zowel error recovery als multiplexing op een gestandaardiseerde manier worden gerealiseerd (lit. 7). We zullen echter zodadelijk zien wat hiervan het nadeel is en een alternatief presenteren.

##### 4.2.4.1 EFTP

De functie Dynamic Call Management is een functie die niet is terug te vinden in aanbevelingen van organisaties als CCITT. Daarom is hiervoor een eigen protocol opgesteld, dat de naam EFTP heeft gekregen. Dit protocol implementeert behalve DCM bovendien een error recovery procedure, vandaar de naam Error Free Transport Protocol.

Voor een exacte beschrijving van EFTP verwijs ik naar het volgende hoofdstuk.

Omdat EFTP reeds een error recovery procedure specificceert, die naadloos aansluit bij X.25, is het gebruik van ECMA-72/X.224 voor error recovery overbodig.

#### 4.2.4.1.1 Headerloos protocol -

EFTP is evenals QLLC een 'header-loos' protocol, wat het volgende inhoudt. Er wordt een onderscheid gemaakt tussen data-units die uitsluitend userdata bevatten en data-units die (mede) control informatie bevatten. De laatste soort informatie is een uitwisseling tussen de betreffende entities, niet direct bedoeld voor de bovenliggende lagen. Headerloos betekent nu, dat de userdata-units verstuurd worden zonder een header. Onderscheid tussen userdata-units en controldata-units wordt zowel in het geval van QLLC als in het geval van EFTP gemaakt met behulp van het Q-bit van X.25.

Wat is nu het voordeel van dit principe? Laten we er, om dit te illustreren, van uit gaan, dat er sprake is van een configuratie als in figuur 4, nummer 5 of 6. De verbinding tussen ComC/ClC en PCU is hierin gerealiseerd met een X.25 verbinding. Het grootste deel van de tijd, zullen er uitsluitend userdatapakketten worden doorgegeven, die, indien volledig gevuld, 128 bytes bevatten. Alle 128 bytes zijn userdata. Het enige wat de PCU hoeft te doen, is ieder binnengekomen pakket door te geven naar het X.25-netwerk.

Wanneer EFTP geen headerloos protocol was geweest, had de PCU de pakketten moeten herarrangeren, omdat volle pakketten te groot zijn om te worden voorzien van een header en naar het netwerk gestuurd te worden. Vol betekent immers dat er niets meer bij kan. Dit is het belangrijkste bezwaar tegen ECMA-72.

Het grootste nadeel hiervan is niet eens, wat men zou kunnen denken, dat de taak van de PCU hierdoor bemoeilijkt wordt. Dat is een marginaal probleem, een kwestie van een stuk software dat iets anders gestructureerd moet worden. Nee, het grootste bezwaar ligt in de vertraging die de userdatapakketten op zouden lopen in de PCU. Herarrangeren van pakketten betekent, dat userdata regelmatig zullen moeten wachten tot er een pakket geheel gevuld kan worden. De te verwachten extra vertraging belooft, afhankelijk van de situatie, gemiddeld in de orde van een halve link-transmissietijd van een pakket, laten we zeggen 100 ms.

#### 4.2.4.1.2 Call management in EFTP -

EFTP maakt een strict onderscheid tussen de begrippen Transport-connectie en Netwerk-call. De gebruiker van EFTP, de TS-user (transport-service-user), heeft de beschikking over de primitives connection-request en disconnection-request voor het starten/beëindigen van een connectie. Op het moment dat de T-entity de connection-request ontvangt, wordt een N-call opgezet om aan dit verzoek te voldoen. Over deze call vindt de informatiebehoefte plaats ten behoeve van de TS-user.

Er zijn drie redenen denkbaar waarom de T-laag de call kan beëindigen:

- o Disconnection request
- o Er is al geruime tijd geen transport van data geweest
- o Er is een gebrek aan resources.

In de twee laatste gevallen zal de N-call gecleard worden, terwijl de T-connectie blijft bestaan. Geeft de TS-user weer te transporteren data door, dan zal de T-entity weer een call opzetten ter ondersteuning van de T-connectie.

EFTP specificeert drie specifieke procedures voor de drie genoemde gevallen.

Indien EFTP faalt in het (her)opzetten van de call, zal dit gemeld worden aan de T-user, en wordt de T-connectie geacht te zijn beëindigd.

4.2.4.1.3 Error recovery in EFTP - EFTP specificeert een error recovery procedure, die alleen in actie komt bij een reset van de netwerklaag. Er vindt een uitwisseling plaats van het nummer van de laatste goed ontvangen userdata-unit, waarna hertransmissie wordt gedaan.

#### 4.2.4.2 Compatibiliteit met QLLC

Naast EFTP zal de PCU ook QLLC ondersteunen als netwerkfunctie, zodat hij gebruikt kan worden als NIA (Network Interface Adapter). De PCU kan zelf detecteren welk protocol gebruikt moet worden, doordat het call request pakket voorzien wordt van een protocol identification, zoals gespecificeerd in QLLC.

#### 4.2.5 PCU beheer

##### 4.2.5.1 Afbeelden Locale kanalen op Netwerkkkanalen

Zoals gezegd, worden de verschillende link laag verbindingen die ontstaan door multidropping en de verschillende calls die in X.25 verbindingen aanwezig zijn, door de locale functie aangeduid als 'locale kanalen'. De netwerkfunctie onderscheidt 'netwerk kanalen'. De PCU doet een 'mapping' van de locale kanalen op de netwerk kanalen.

Het beheer richt zich op deze drie aspecten: beheer van de lokale kanalen, beheer van de netwerk kanalen en beheer van de mapping. Verder zullen de fysieke poorten waarop de kanalen gemapt worden, beheerd moeten worden.

Van de fysieke poorten moeten de volgende eigenschappen gespecificeerd worden:

- o locale- of netwerkpoort
- o Voor locale poort: X.25/QLLC of SDLC
- o Bij SDLC: half duplex/full duplex
- o Bij SDLC: Point-to-point/Multipoint/Loop configuratie
- o Bij SDLC: Primary of Secondary link station
- o Transmissiesnelheid, etc.

Van de lokale kanalen moeten onder meer de volgende karakteristieken worden gespecificeerd.

- o Welke fysieke poort?
- o In geval van X.25/QLLC: DTE-adres
- o In geval van SDLC: secondary station adres.

Van de netwerkkkanalen moeten onder meer de volgende eigenschappen worden gespecificeerd:

- o Welke fysieke poort?
- o DTE-adres.
- o QLLC of EFTP?

De mapping beperkt zich tot het aspect:

- o Welke lokale kanalen worden gemapt op welke netwerkkkanalen?

## HOOFDSTUK 5

### ERROR FREE TRANSPORT LAYER PROTOCOL

Dit is de beschrijving van het Error Free Transport Protocol (EFTP), overeenkomend met de transport laag (laag 4) van OSI volgens CCITT aanbeveling X.200.

De volgende tekst wordt geacht integraal deel uit te maken van deze definitie:

Standard ECMA-72 Transport Protocol 2nd ed. september 1982 § 1.1 t/m § 5.5.2. m.u.v. § 5.4.2.

#### 5.1 UITGANGSPUNTEN

Beschikbaar is een netwerk laag à la X.25 met de volgende features:

- o Er treden geen fouten op in het transport van NSDU's tenzij de netwerk laag een reset geeft.
- o Pakketten worden in de juiste volgorde afgeleverd.
- o Geen duplicaties, geen vermiste pakketten.
- o Indien de netwerk laag een reset geeft, is de communicatie tot aan de reset foutloos geweest, en zal ook daarna foutloos zijn, volgens de vorige twee punten. Echter, in pakketten die voor de reset zijn verstuurd en na de reset aankomen, kunnen verwisselingen, vermissingen en/of duplicaties voorkomen.
- o Reset wordt altijd aan beide einden gemeld.
- o Een Clear Indication die door het netwerk geïnitieerd wordt, wordt altijd naar beide einden gestuurd. Bovendien is zo'n Clear Indication onderscheidbaar van een DTE-geïnitieerde Clear Indication door het cause field.



- o Pakketten verstuurd ná het optreden van de reset, komen aan de andere kant ook pas aan na het aankomen van de reset aldaar.
- o N-laag ondersteunt M en Q bit.
- o De volgende primitives zijn beschikbaar voor gebruik door de Transport laag:
  - Call Request, Indication, Accepted en Connected
  - o Data (met M en Q bit)
  - o Reset, Restart
  - o Clear Request, Indication, Confirmation

## 5.2 FUNCTIONALITEIT VAN DE TRANSPORT LAAG

EFTP ondersteunt de volgende functies:

- o Error recovery: EFTP garandeert foutloze transmissie van userdata. Alleen exceptionele fouten worden gemeld aan de bovenliggende laag, en resulteren in een beëindiging van de connection.
- o Dynamic Call management: Een connection op transportlaagniveau wordt door EFTP vertaald naar een call op netwerklaagniveau. Echter, in geval er gedurende een te bepalen tijd geen verkeer heeft plaatsgevonden over de T-connection, hebben beide T-entity's het recht de N-call te clearen, met behoud van de T-connection. Op het moment dat er weer verkeersaanbod is, zal de transportlaag de onderliggende N-call weer herstellen, en aldus de transparantie ten behoeve van de T-service-user kunnen garanderen.
- o Op deze wijze is het niet langer nodig handmatig calls op te zetten en te clearen, maar geschiedt dit geheel automatisch, terwijl de duur van de calls niet langer is dan strikt noodzakelijk.
- o EFTP ondersteunt het gebruik van Qualified userdata.
- o EFTP ondersteunt het M bit. Flow control wordt uitgevoerd door de onderliggende netwerklaag. Hiertegen is geen bezwaar, aangezien EFTP geen multiplexing definiëert. Een transportlaag met multiplexing maakt een eigen flow control protocol noodzakelijk, daar anders de verschillende gemultiplexte connections elkaar kunnen beïnvloeden.

### 5.3 PROCEDURES VOOR HET OPSTARTEN VAN EEN T-CONNECTION

De volgende procedure wordt gebruikt voor het initieel opzetten van een T-connection op verzoek van de Session laag. De session laag kan userdata bij dit verzoek voegen.

Aan een T-connection kan worden gerefereerd door middel van de zogenaamde reference, dat is een identifier die gebruikt wordt om verschillende connections tussen hetzelfde paar DTE's te onderscheiden. De reference wordt bij het sturen van de T-Conn-Req door de initiërende DTE toegekend aan de op te zetten connection.

Eén T-connection wordt gemapt op (maximaal) één N-call. Voorafgaand aan het sturen van de T-Conn-Req TPDU wordt derhalve een call opgezet middels de N-Call-Request NSDU. In het call request pakket wordt een protocol identification meegestuurd in het userdata field, aangevende dat het het protocol EFTP betreft. Voorts bevat het userdatafield de reference.

Bij het opzetten van een call door DTE A naar DTE B zullen zowel A als B hun variabelen T(S) en T(R) op nul (0) zetten.

Na het totstandkomen van de N-Call, stuurt de initiërende DTE een T-Conn-Req, die altijd genummerd is (met nummer 0) en de userdata van het connection request van de session laag mag bevatten.

#### 5.4 PROCEDURES VOOR HET DOORGEVEN VAN USERDATA

TSDU's (userdata) worden door de T-laag niet voorzien van een header, maar direct als NSDU met Q-bit=0 naar de peer T-entity gestuurd. Deze userdata TPDU's worden onderscheiden van andere TPDU's door dat alle andere TPDU's hun Q-bit hebben aanstaan.

Er vindt zogenaamde impliciete nummering plaats. Dit houdt het volgende in.

De netwerklaag houdt een genummerde copie aan in een buffer van (een subset van de) verzonden TPDU's die nog niet geacknowledget zijn. De copie van een verzonden TPDU wordt genummerd met het nummer T(S). Na verzending wordt  $T(S) := T(S) + 1$ .

De volgende TPDU's worden genummerd:

- o T-Conn-Req
- o T-Data
- o T-Qdata
- o T-Disc-Req
- o T-Conn-Abort
- o T-RR en T-RSM worden alleen genummerd indien er een userdata veld in zit.

De ontvangende T-entity weet de T(S), die bij hem T(R) heet, doordat hij na ontvangst van iedere TPDU van bovengenoemde categorieën zijn T(R) met één ophoogt, en doordat de netwerklaag zo betrouwbaar is.

Er is een bepaalde maximum windowgrootte gedefiniëerd (typisch 128), die impliceert, dat er niet meer dan zo veel genummerde TPDU's onderweg mogen zijn, d.w.z. waarvan de acknowledgement nog niet is ontvangen.

TSDU's kunnen ook worden verstuurd in de TPDU's RR en RSM.

##### 5.4.1 Acknowledgement

De ontvangende T-entity acknowledgedget d.m.v. een T-RR(T(R)) TPDU de ontvangst van genummerde TPDU's tot T(R). (D.w.z. t/m T(R)-1).

Hij doet dit als het window een vastgestelde grootte heeft overschreden. Het T-RR wordt niet impliciet genummerd. Bovendien kan de RR een userdataveld bevatten, waarin een TSDU (die niet de maximum lengte kan hebben, dus geen M bit kan hebben en voorts geen Q bit mag hebben) verstuurd kan worden. Deze TSDU wordt gebufferd en genummerd.

#### 5.4.2 More Bit

De T-Service-user kan TSDU's sturen, groter dan de N-laag packet size. Deze worden op de gebruikelijke manier gesplitst in NSDU's waarvan alleen de laatste M-bit=0 heeft.

#### 5.4.3 Qualified TSDU's

De TS-user kan TSDU's sturen die zijn gemerkt als 'qualified'. Deze worden voorzien van een header, en als qualified NSDU verstuurd. Het aanbrengen van een header is noodzakelijk om deze pakketten te onderscheiden van TPDU's die door de T-laag zelf als Qualified TSDU's worden verstuurd. De aldus geconstrueerde TPDU bevat voorts een T(R) en fungeert aldus als impliciete RR. Een qualified TSDU wordt dus verstuurd in T-Qdata(T(R),QTSDU).

Een qualified TSDU die te groot is voor een enkele TPDU wordt gesplitst in pakketten met zowel M=1 als Q=1. De laatste TPDU heeft M=0 en Q=1.

#### 5.4.4 Expedited data

Expedited TSDU's worden Niet ondersteund.

## 5.5 PROCEDURES VOOR ERROR RECOVERY

### 5.5.1 N-Reset

In geval een T-entity een N-Reset ontvangt van de netwerk laag, wordt de volgende error recovery procedure opgestart. Merk op, dat de procedure volkomen symmetrisch is ten opzichte van de beide einden van de T-connection, en dat de procedure zich dus aan beide zijden op de hier beschreven manier afspeelt.

- o T-entity stuurt een T-REJ(T(R)) naar peer entity. Hierbij wordt T(S) niet opgehoogd.
- o Het uitzenden van andere pakketten wordt tijdelijk gestaakt.
- o Alle binnekomende TPDU's worden genegeerd, m.u.v. T-RSM en T-REJ.
- o Op ontvangst van de T-REJ(n) wordt  $T(S)=n$ . Vervolgens wordt een T-RSM (resume) uitgezonden, en wordt het herzenden van TPDU's, te beginnen bij T(S), vanuit de gebufferde copieën hervat.
- o Op ontvangst van een T-RSM wordt de ontvangst van TPDU's weer hervat.

### 5.5.2 N-Clear-Ind

Bij ontvangst van een N-Clear-Ind, waarvan de cause aangeeft dat de call niet gecleard is op verzoek van de andere DTE, moet DTE die de oorspronkelijk call opzette, proberen opnieuw een call te realiseren. Dit geschiedt door het herhaald uitzenden van Call-Req NSDU's. Indien en wanneer het gelukt is de call opnieuw op te zetten, zal dezelfde procedure als voor de reset afhandeling, zie boven, gestart worden. Beide DTE's beginnen dus met REJ(T(R)) te sturen, etcetera.

Indien de N-Call-Request niet gelukt binnen een zekere tijd  $t$ , wordt de T-connection beëindigd. De DTE die niet zelf de call op hoofde zetten, neemt aan dat de connectie beëindigd is, op het moment dat dezelfde tijd  $t$  is verstreken.

In het onverhoopte geval dat op zeker moment er een call wordt opgezet door een van beide DTE's, terwijl de andere DTE aanneemt, dat de connection is beëindigd, stuurt deze laatste DTE een T-Conn-Abort(T(R)). De ander antwoordt met een N-Clear-Req.

## 5.6 PROCEDURES VOOR HET BEËINDIGEN VAN DE T-CONNECTION

Voor het beëindigen van de T-connection is de volgende procedure voorgeschreven. De T-entity die de connection wil beëindigen, stuurt een T-Disc-Req(T(R)) (Disconnect Request) naar de peer entity en geeft hiermee te kennen:

- o Ik heb tot T(R) ontvangen
- o Ik beloof verder niets te sturen
- o Ik wens de connectie te beëindigen.

De DTE die een T-Disc-Req ontvangt, mag de TSDU's die nog klaar staan met M-bit=1 en bovendien een TSDU zonder M-bit, nog versturen.

De inviterende DTE beantwoordt ieder ontvangen TPDU zonder M-bit met een hernieuwde T-Disc-Req(T(R)), daarmee de ontvangst tot T(R) bevestigend.

De peer DTE cleart de call op het moment dat de T-Disc-Req(n) binnenkomt met n=T(S).

### 5.6.1 T-Disc-Req-collision

Een DTE die een T-Disc-Req heeft verstuurd en bovendien een T-Disc-Req ontvangt, mag een N-CLR-REQ sturen, d.w.z. de call clearen.

Een DTE die een T-Disc-Req heeft verstuurd en bovendien een T-PINVC ontvangt, negeert die T-PINVC.

Een DTE die een T-PINVC heeft verstuurd en een T-Disc-Req ontvangt, reageert op de gebruikelijke wijze op de T-Disc-Req met N-CLR-REQ

### 5.6.2 T-Conn-Abort

In geval van protocol errors is het de DTE toegestaan een T-Conn-Abort te sturen, die beantwoord moet worden met een N-Clear-Req. Dit impliceert het einde van de T-connection.

### 5.7 PROCEDURES VOOR HET BELEEFD TIJDELIJK BEËINDIGEN VAN EEN CALL.

De T-entity zal een call die behoort bij een connection waarover al geruime tijd geen verkeer meer is geweest, beëindigen middels de PINVC(T(R)) TPDU, d.i. Polite INVitation to Clear. Het beëindigen van de call betekent echter niet, dat de connection wordt beëindigd. De connection blijft bestaan, en de session laag wordt derhalve niet ingelicht over het clearen van de call. Er is een systeemp parameter die specificceert hoe groot de tijd is, die verstreken moet zijn na het versturen/ontvangen van de laatste TSDU, alvorens de call om deze reden beëindigd mag worden. De meting geschiedt in beide TSDU-datastromen.

De DTE die een PINVC(n) verstuurt, geeft hiermee te kennen:

- o Ik heb tot n ontvangen.
- o Ik beloof verder niet te zenden, tenzij ik weer data van de peer entity ontvang.
- o Ik wens de call te beëindigen, tenzij ik weer data van de peer entity ontvang.

De DTE komt hierna in de Polite-clearing toestand.

De DTE die de T-PINVC ontvangt, heeft twee opties:

- o De uitnodiging afwijzen door het zenden van een TPDU met  $T(S) \geq n$ .
- o De uitnodiging aanvaarden, door de call te clearen (N-CLR-REQ).

Indien de ontvanger van een T-PINVC constateert dat zijn  $T(S) > n$  (dus dat reeds een TPDU is verstuurd met  $T(S) \geq n$ ), vervalt optie 2. De T-PINVC wordt dan genegeerd.

In het geval de T-PINVC genegeerd wordt, door de peer T-entity, ontvangt de inviterende DTE een TPDU na de versturing van de T-PINVC. De inviterende DTE verlaat hierop de Polite-clearing toestand en mag weer gewoon zenden en ontvangen.

De hier beschreven procedure mag niet aangevangen worden, indien een reset is ontvangen waarvan de error recovery procedure nog niet is voltooid, of indien een T-PINVC of T-Disc-Req is ontvangen van de andere T-entity.



### 5.7.1 T-PINVC-collision

Een DTE die een T-PINVC heeft verstuurd en bovendien een T-PINVC ontvangt, mag een N-CLR-REQ sturen, d.w.z. de call clearen. Zie ook: T-Disc-Req-collision

### 5.7.2 Hervatten van de call na een T-PINVC

Een T-connection die niet gemapt is op een N-call, doordat de call is beëindigd met een T-PINVC, kan door beide DTE's opnieuw op een N-Call worden gemapt. Hiertoe wordt een N-Call-Req verstuurd zoals beschreven in het hoofdstuk "Procedures voor het opstarten van een T-connection." Echter, er wordt geen T-Conn-Req meer verstuurd. Ook worden T(R) en T(S) niet op nul gezet: zij behouden hun waarde. Dit proces heet: Reconnect.

### 5.7.3 Reconnect-Collision

Neem aan dat de volgende situatie zich voordoet. Er bestaat een T-connection die tijdelijk niet is gemapt op een N-call, omdat er een PINVC is uitgewisseld, en beide T-entity's zijn overeengekomen dat de call beëindigd kan worden. Het is nu denkbaar, dat beide T-entity's op hetzelfde moment userdata te versturen hebben, en dus gaan reconnecten. Deze situatie heet een Reconnect collision. Merk op, dat dit een geheel ander verschijnsel is dan de N-call collision, zoals gedefinieerd in X.25.

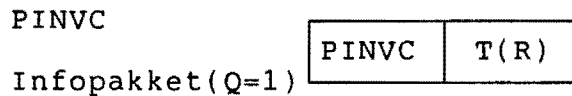
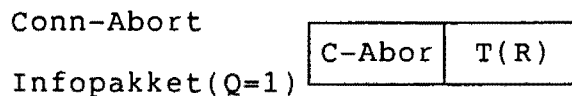
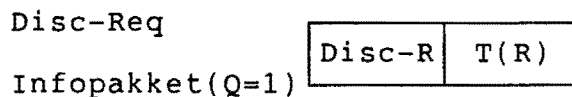
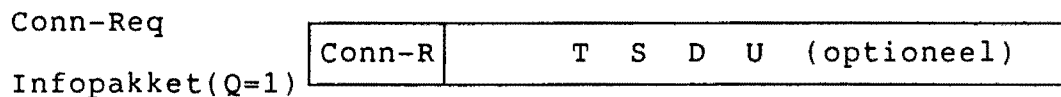
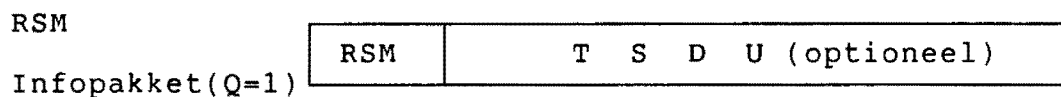
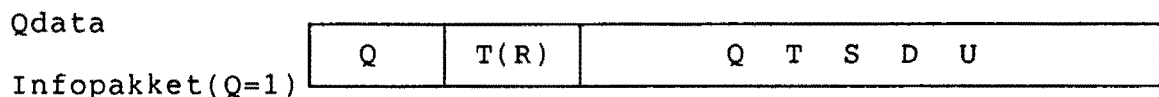
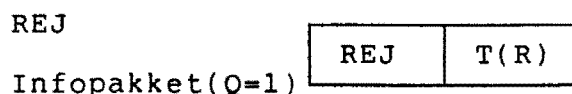
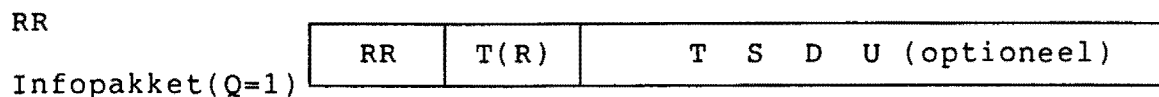
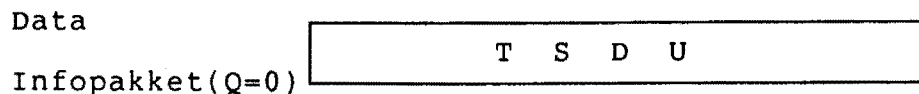
Dit is een ongewenst verschijnsel, aangezien een connection maximaal op één call gemapt kan worden. Dit probleem wordt op de volgende wijze opgelost.

De T-entity die weet, dat hij als laatste en als enige een T-PINVC heeft verstuurd, mag zijn call laten staan, en de binnenkomende N-call-Request beantwoorden met een call clear. Aan dit criterium kunnen nooit beide T-entity's voldoen. Indien een van beide T-entity's aan dit criterium voldoet, is call collision opgelost.

Indien beide call requests met een call clear worden beantwoord, zullen beide T-entity's het na een random tijd (de kans is gelijkmatig gedistribueerd tussen 0 en een zekere tijd) opnieuw proberen, zolang totdat de call tot stand is gekomen.

5.8 OVERZICHT TPDU'S MET HUN OPMAAK

De volgende typen TPDU's zijn in de voorafgaande hoofdstukken beschreven.



## HOOFDSTUK 6

### CONCLUSIE

SNA is een netwerkkarchitectuur met een groot belang, vanwege de positie als de-facto standaard, maar met een eveneens grote historische ballast, die de architectuur niet op alle punten even logisch en doorzichtig maakt. Willen we SNA laten samenwerken met OSI-gebaseerde architecturen, dan is een geschikt punt in de SNA architectuur om protocolconversies te doen, de scheiding tussen Data Link Control en Path Control laag.

Gebruik makend van de in dit verslag beschreven PCU kan men een SNA-netwerk opbouwen dat bestaat uit deelnetwerken op verschillende locaties. Dit netwerk maakt op een zeer kostenbewuste manier gebruik van het X.25 netwerk.

Het vervolgonderzoek spitst zich momenteel toe op de verdere definitie van de taken van de PCU en op het gebruikte protocol EFTP. Het onderzoek op dit laatste punt heeft inmiddels geleid tot een aantal veranderingen in de protocoldefinities, die niet meer in dit verslag zijn verwerkt.

## HOOFDSTUK 7

### LITERATUUR

1. Tanenbaum, Andrew S., Computer Networks, Englewood Cliffs, 1981<sup>1</sup>.
2. Armonk (IBM), Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic, SC30-3112-2, Nov 1980.
3. CCITT, red book, Fascicle VIII.5, Data Communication Networks: Open Systems Interconnection (OSI), system description techniques, Recommendation X.200.
4. IBM, The X.25 Interface For Attaching SNA nodes To Packet-Switched Data Networks; General Information Manual. GA27-3345-1. Section II-8.
5. Gurugé, Anura, SNA Theory and Practice, A comprehensive guide to IBM's Systems Network Architecture. Maidenhead, 1984<sup>1</sup>, herdruk 1987.
6. CCITT, red book, Fascicle VIII.3, Data Communication Networks Interfaces, Geneva 1985, pp. 108-242: Recommendation X.25 .
7. ECMA (European Computer Manufacturers Association), Standard ECMA-72, Transport Protocol, 2nd edition, september 1982.

## HOOFDSTUK 8

### AFKORTINGEN

ACD	Address Conversion Database
BIU	Basic Information Unit
CC	Cluster Controller
CCITT	The International Telegraph and Telephone Consultative Committee
ComC	Communications Controller, ook wel: Front End Processor
DAF	Destination Address Field
ECMA	European Computer Manufacturers Association
EDP	Electronic Data Processing
EFTP	Error Free Transport Protocol
GW	Gateway
ISO	International Organisation for Standardization
NIA	Network Interface Adapter
NSDU	Network Service Data Unit
OAF	Originating Address Field
OSI	Open Systems Interconnection
PCU	Protocol Conversion Unit
PIU	Path Information Unit
PSI	Packet Switching Interface
PVC	Permanent Virtual Circuit
QLLC	Qualified Logical Link Control
SDLC	Synchronous Data Link Control
SNA	Systems Network Architecture
SVC	Switched Virtual Call
TH	Transmission Header
TPDU	Transport Protocol Data Unit
TSDU	Transport Service Data Unit
VC	PVC of SVC

--o--

**KORTINGEN OP NATIONAAL VERBRUIKSBEDRAG DATANET 1**

Onderstaand model geeft aan welk percentage korting wordt verleend.

Deze korting geldt alleen voor het verbruiksbedrag van nationaal verkeer.

Het betreft hier een korting per maand, per aansluiting, waarbij volgens een schijvenmodel de totaalkorting wordt berekend op basis van de kosten voor oproepen, tijd en segmenten\*. Bij PVC-gebruik wordt de korting berekend over de kosten voor het aantal getransporteerde segmenten.

**KORTINGSMODEL**

Kortingspercentage	Opvolgende maandbedragen nationaal verbruik	Totaal opvolgende schijven	Te betalen bij totaal opvolgende schijven	Korting bij totaal opvolgende schijven
10% over de eerste	fl. 250,00		fl. 225,00	fl. 25,00
15% over de volgende	fl. 250,00	fl. 500,00	fl. 437,50	fl. 62,50
24% over de volgende	fl. 750,00	fl. 1.250,00	fl. 1.007,50	fl. 242,50
32% over de volgende	fl. 1.250,00	fl. 2.500,00	fl. 1.857,50	fl. 642,50
40% over de volgende	fl. 2.500,00	fl. 5.000,00	fl. 3.357,50	fl. 1.642,50
47% over de volgende	fl. 7.500,00	fl. 12.500,00	fl. 7.332,50	fl. 5.167,50
52% over de volgende	fl. 12.500,00	fl. 25.000,00	fl. 13.332,50	fl. 11.667,50
55% over elke volgende gulden				

**Kosten voor optionele faciliteiten van aansluitingen<sup>1)</sup>**

Faciliteiten voor vaste X.25-aansluitingen		Maandabonnement
Acceptatie opgeroepene betaalt	per aansluiting	f 25,- <sup>2)</sup>
Opname in besloten gebruikersgroep	per groep per aansluiting	f 5,- <sup>2)</sup>

Extra logische kanalen		Maandabonnement
• voor inkomende en uitgaande oproepen	per kanaal	f 5,-
• voor inkomende oproepen	per kanaal	f 5,-
• voor uitgaande oproepen	per kanaal	f 5,-

1) Voor het gelijktijdig wijzigen van één of meer faciliteiten geldt een mutatie tarief van fl. 25,00.

2) Alleen van toepassing voor nationaal verkeer.

\* 1 segment = maximaal 64 tekens

Voorwaarden voor het gebruik van Datanet 1 en nadere informatie zijn verkrijgbaar bij:

PTT Telecommunicatie, DIDS-DACOM, Postbus 30000, 2500 GA 's-Gravenhage, Telefoon 070-438611

# TARIEVEN DATANET 1

## TOEGANG TOT HET NET

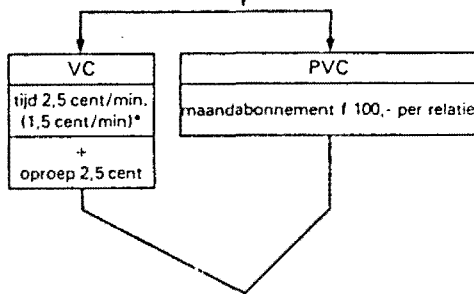
Vaste X.25-aansluitingen		
Enkelvoudige aansluitingen		Aanlegkosten
snelheids-klasse (bit/s)	maand-abonnement 1)	(eenmalig)
2400 (deel)*	fl. 235,-	fl. 500,-
2400	fl. 400,-	fl. 500,-
4800	fl. 650,-	fl. 500,-
9600	fl. 850,-	fl. 500,-
48000	fl. 2200,-	fl. 2000,-
64000	fl. 2200,-	fl. 2000,-

\* Beperking: één logisch kanaal, alleen VC.

Netwerk aansluitingen		
aanlegkosten		
snelheids-klasse (bit/s)	maand-abonnement 1)	(eenmalig)
2400	fl. 1100,-	fl. 1500,-
4800	fl. 1350,-	fl. 1500,-
9600	fl. 1550,-	fl. 1500,-
(per bijkomend modempaar)		fl. 175,-
(excl. eventuele huurlijnen!)		

Gekozen verbindingen	
Pad verkeer (excl. modem) 300/300, 1200/1200, 1200/75 bit/s	
Verkeers-type	maand-abonnement
Ongeïdentificeerd	n.v.t.
Geïdentificeerd	fl. 30,-

### RELATIE TUSSEN AANSLUITINGEN NATIONAAL



### TRANSPORT VAN INFORMATIE NATIONAAL

Volumekosten
0,25 cent per segment (0,15 cent per segment)*

\* in het weekend, op nationale feestdagen en op werkdagen van 19.00 - 08.00 uur.

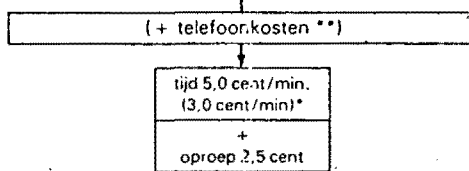
### RELATIE TUSSEN AANSLUITINGEN INTERNAT.

VC EUROPA	VC VS/CANADA	VC overige intercontinentale bestemmingen
tijd: 9,5 ct/min	tijd: 27,0 ct/min	tijd: 36,0 ct/min
+	+	+
oproep: 10 cent	oproep: 25 cent	oproep: 30 cent

### TRANSPORT VAN INFORMATIE INTERNAT.

EUROPA	VS/CANADA	overige intercontinentale bestemmingen
0,57 ct/segment	1,6 ct/segment	1,8 ct/segment

### RELATIE TUSSEN AANSLUITINGEN NATIONAAL

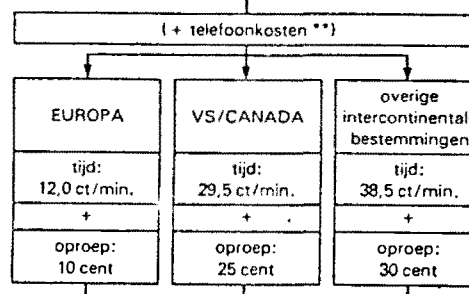


### TRANSPORT VAN INFORMATIE NATIONAAL

Volumekosten
0,50 cent per segment (0,30 cent per segment)*

\* in het weekend, op nationale feestdagen en op werkdagen van 19.00 - 08.00 uur.

### RELATIE TUSSEN AANSLUITINGEN INTERNAT.



### TRANSPORT VAN INFORMATIE INTERNAT.

EUROPA	VS/CANADA	overige intercontinentale bestemmingen
0,82 ct/segment	1,85 ct/segment	2,05 ct/segment