

CAM Programmeren voor dummies

*“Licht in de duisternis van programmatuurontwikkeling
voor CAMS en smartcards”*

Versie 1.0

Inhoudsopgave

1. Inleiding	3
2. Een globaal overzicht.....	5
3. UCAS CAM (nog leeg).....	8
4. Matrix CAM (nog leeg)	9
5. Programmeerkennis (nog leeg).....	10
6. Coderingen (nog leeg).....	11
7. Emulatiesoftware (nog leeg).....	12
8. Smartcards (nog leeg)	13
Appendix 1: DVB Algemeen	14
Appendix 2: Releasenotes	18

1. Inleiding

Ontvangst via een schotel en receiver was vier maanden geleden nog geheel nieuw voor mij. De eerste maanden ben ik bezig geweest met mijn decoder, een Manhattan Mx met geïntegreerde Matrix Reloaded CAM. Geweldig op gang geholpen door mijn leverancier kon ik al snel wat spelen met emulaties en titanium en fun kaarten.

Ik liep natuurlijk net als iedereen regelmatig tegen het merkwaardige fenomeen aan dat schermen zomaar ineens volledig zwart werden. De sleuteltjes tot de oplossing van dit probleem bleken vaak wel beschikbaar. Of deze echter tezamen met eventuele andere vereiste aanpassingen ook in emulaties of kaarten beschikbaar kwamen, bleek kennelijk een overwegend commerciële afweging te zijn. Voor de nodige euri kon de opvolger van een CAM of kaart worden aangeschaft omdat updates voor de voorganger niet meer verschenen. Terwijl het mij nooit echt duidelijk is geworden wat er in technologisch opzicht nou daadwerkelijk was verbeterd aan de opvolger.

De voor het ontwikkelen van een emulatie benodigde kennis lijkt schaars en voor zover ik kan zien wordt deze bewust schaars gehouden. Dit feit draagt natuurlijk ook niet bij aan het ontworstelen aan de wurggreep van de leveranciers. Terwijl zij zonder deze gedwongen verkoop veel meer gestimuleerd zouden worden tot het produceren van daadwerkelijk snellere en betere hardware met meer mogelijkheden.

Het lijkt mij een goede zaak dat meer mensen zich bezighouden met de ontwikkeling van software voor CAMS en kaarten. Natuurlijk schuilt hier het risico in van ondeugdelijke emulaties en smakeloze grappen, maar het voordeel van de onafhankelijkheid van producenten lijkt mij hier tegenop te wegen. Open source heeft zich uiteindelijk ook op andere vlakken bewezen. Bovendien zouden maatregelen mogelijk zijn zoals een “certificering” van een emulatie door een partij als sat4all.

Ik heb me daarom voorgenomen deze materie te onderzoeken en te kijken of ik enerzijds zelf tot een emulatie zou kunnen komen en anderzijds gaandeweg mijn bevindingen in een soort van tutorial zou kunnen vastleggen. Dit laatste in de hoop ook anderen te stimuleren zich meer in deze materie te verdiepen. Door de bevindingen gaandeweg te publiceren, hoop ik op terugkoppeling van anderen die verder gevorderd zijn dan ik, zodat ik dit weer in dit document kan verwerken. Hiermee kan het een document van ons allemaal worden.

Mijn persoonlijke achtergrond ligt weliswaar in software-ontwikkeling, maar dan voor financieel administratieve systemen. Navraag leerde mij dat de software voor CAM's in de programmeertaal C en/of assembly kon worden geschreven. Nu heb ik in een grijs verleden wel eens een jaartje C geprogrammeerd en mijn kennis van assembly is puur theoretisch, maar ik heb wel het gevoel dat ik dit met de nodige studie zou moeten kunnen leren. Ik heb voor mijzelf een werkplan gemaakt dat mij de volgende kennis zou moeten opleveren:

- Algemene kennis van (gecodeerde) satelliet uitzendingen
- Algemene kennis van de werking van CAMS en kaarten
- Programmeerkennis van C en/of assembly
- Gedetailleerde kennis van een specifieke CAM
- Gedetailleerde kennis van minimaal één van de bekende coderingen (seca, conax, etc.)

Met deze kennis zou men uiteindelijk toch in staat moeten zijn om software voor een CAM te ontwikkelen.

Om deze kennis te verkrijgen bestudeer ik de nodige documenten en websites en val ik medemensen lastig. Wat ik daarvan begrijp en waarvan ik denk dat het zinvol is voor de doelstelling, neem ik in dit document op. Daarbij neem ik zoveel mogelijk verwijzingen op naar de sites met de achtergrondinformatie. Informatie die mij ook interessant lijkt, maar die niet direct vereist lijkt voor de doelstelling, neem ik op in appendices.

Het document komt dus in stappen tot stand en bij iedere afronding van een fase, of na het verkrijgen van input voor correcties zal een nieuwe versie worden gemaakt. De x.0 (punt nul) versie zal altijd het document zijn met de eerste poging tot het verstrekken van nieuwe informatie. In documenten met een hoger subnummer zullen fouten zijn gecorrigeerd of aangedragen verbeteringen zijn aangebracht.

Of het document ooit klaar en volledig zal zijn? Geen idee. Zal het gewenste effect worden bereikt? Geen idee. Maar het is in ieder geval wel leuk om eraan te werken!

Hermanator
3 oktober 2004

Met dank aan: Bommeltje, Duwgati, Pic-o-matic, MiLo, Ozzo, Wildcard, EnEmA en anderen.

2. Een globaal overzicht

MPEG-2

Een uitzending waarnaar je via de schotel kijkt is eigenlijk een **MPEG-2** bestand zoals ook de talloze filmpjes waarmee we elkaar bestoken in de mail. Het bestand wordt in dat geval uit je mail gehaald en opgeslagen op je harddisk waarna het met bijvoorbeeld de Mediaplayer van Windows wordt afgespeeld. Een gesimplificeerde voorstelling is dat nu dit bestand als “**datastream**” wordt verzonden vanaf de satelliet. De satellietontvanger speelt deze datastream direct tijdens het ontvangen af op de televisie. Deze datastream kun je zien als een lange reeks bytes van het oorspronkelijke MPEG-2 bestand, netjes gegroepeerd in **packets**, met hierin toegevoegde controle-informatie om een correcte ontvangst te garanderen.

De werkelijkheid is vanzelfsprekend iets complexer. Een satelliet bestaat bijvoorbeeld uit verschillende transponders die meerdere transportstreams kunnen uitzenden met daarbinnen meerdere substreams (de programma's, teletekst, EPG, etc.). De programma's kunnen weer over meerdere satellieten zijn gegroepeerd in een boeket, waarop men zich kan abonneren. Binnen een substream worden tussen de packets met de daadwerkelijke informatie (beeld, geluid, etc.) ook packets met besturingsgegevens tussengevoegd. Hierin bevinden zich gegevens over het uitgezonden programma, de codering, abonnementen, etc. De ontvanger filtert deze berichten uit de datastream voor verdere verwerking. Wat meer achtergrondinformatie hierover is opgenomen in Appendix 1.

Scrambling

Onder andere om ervoor te zorgen dat alleen “bevoegden” de uitzending kunnen zien, wordt de datastream **scrambled** (gecodeerd) verzonden. Dit uitzenden van MPEG-2 via de satelliet en de scrambling hiervan is in Europa gestandaardiseerd in de **DVB-s** specificaties (**Digital Video Broadcast via Satelliet**). De scrambling zelf vindt plaats met het gestandaardiseerde DVB Common Scrambling Algorithm (**CSA**).

Deze CSA scrambling voorziet in codering met het zogenaamde **controlword**. Aan de hand van dit controlword wordt een algoritmische bewerking op de datastream uitgevoerd, waardoor dit een onherkenbare reeks tekens wordt. In de ontvanger kan, mits dit controlword bekend is, via de omgekeerde bewerking de datastream weer worden omgezet naar een herkenbare MPEG-2 datastream. De ontvanger kan deze dan, feitelijk net als de Windows Mediaplayer, op de televisie laten zien.

ECM Entitlement Control Message

Het controlword wordt om de 2 tot 10 seconden vervangen (door een random controlword generator bij het uitzenden) om de beveiliging te verhogen. Alleen met het juiste controlword, op het juiste moment heb je dus (gedurende 2-10 seconden) beeld. Je ontvanger moet dus ook met diezelfde frequentie worden voorzien van het nieuwe controlword, om continu beeld te kunnen tonen. De controlwords worden hiertoe in de datastream in een speciaal soort berichten (in aparte packets) tussengevoegd: de **ECM (Entitlement Control Message)**. Deze ECM's worden uit de datastream gefilterd en apart behandeld. Het controlword wordt uit een ECM bericht gehaald en hiermee kan weer een stukje van de datastream worden ontcijferd en op de televisie worden getoond. Tot het volgende ECM bericht wordt ontvangen met een nieuw controlword. Ook andere, op dit moment voor ons begrip nog niet relevante informatie over de uitzending wordt met ECM berichten verzonden.

Encryptie / Keys

Nu gaat het er om wie de controlwords uit de ECM's kunnen en mogen halen. Hier komen de coderingen **Seca**, **Conax**, **Irdeto**, etc. om de hoek kijken. Op het ECM bericht wordt bij een "beschermd" uitzending namelijk een **encryptie** losgelaten van de soort codering waarvoor de provider heeft gekozen. Alleen als een ontvanger in staat is het ECM bericht te decrypten, kan het controlword worden verkregen en kan de datastream correct worden descrambled. Er is dus een duidelijk verschil tussen scrambling en encryption:

- **Scrambling** is de standaard DVB-s CSA codering, waarbij alle uitzendingen worden versleuteld met controlwords.
- **Encryption** is een aparte codering om de meegezonden controlwords te beveiligen, zodat de uitzending alleen descrambled kan worden door hiertoe gerechtigde kijkers.

Als een provider uitzendt in meerdere coderingen (zoals bijvoorbeeld bij Nederlandse zenders Seca/Irdeto) dan worden gewoon voor beide coderingen ECM berichten verzonden met de juiste controlwords.

Omdat alle uitzendingen plaatsvinden volgens de DVB-s specificaties plaatsvinden, wordt een **FTA (Free to Air)** uitzending wél op dezelfde manier gescrambled, maar worden de ECM berichten met de controlwords niet encrypted. Hierdoor kan dus met iedere ontvanger de uitzending worden bekeken.

Voor "beschermd" uitzendingen moet een ontvanger dus in staat zijn om de ECM berichten hierin te kunnen decrypten. Het encryptie-algoritme op zich is niet voldoende, want dan zou iedere ontvanger alle uitzendingen in de hem bekende encryptie op de televisie kunnen toveren. De ECM berichten worden hierom encrypted met periodiek (bijvoorbeeld wekelijks of maandelijks) wijzigende **keys**. Deze keys zijn dan ook de gewilde sleuteltjes waar we altijd zo druk naar zoeken. Helaas wijzigt een provider ook nog wel eens andere onderdelen van het encryptieproces, en dan redden alleen de sleuteltjes ons niet.

EMM (Entitlement Management Message)

Hoe komen deze periodiek wijzigende keys dan weer in de ontvangers van de abonnees? Hiervoor is een nieuwe soort berichten bedacht: de **EMM (Entitlement Management Message)** berichten. Ook deze berichten worden met dezelfde encryptie (seca, irdeto, etc.) versleuteld en in de datastream tussengevoegd. Maar aangezien de EMM bedoeld is om alleen diegenen met een geldig abonnement te voorzien van o.a. nieuwe keys, moeten deze met een per abonnement unieke key worden versleuteld.

Een abonnee heeft een **smartcard** met abonnementsgegevens, waaronder een unieke **masterkey**. Een EMM bericht met onder andere de nieuwe keys wordt nu encrypted met deze masterkey. Een EMM bericht kan dus uitsluitend met behulp van deze ene specifieke smartcard worden ontsleuteld. De nieuwe keys in het EMM bericht worden bijgewerkt op de smartcard. Met de EMM's worden nog andere kenmerken van het abonnement en het algoritme verzonden en bijgewerkt op de smartcard. De codering kent daarom naast het algoritme ook commando's om de kaart te bewerken. Samenvattend:

- Iedere **datastream** wordt **gescrambled met controlwords**;
- Die controlwords worden in de datastream meegezonden in **ECM berichten**;
- Deze ECM berichten kunnen worden **encrypted met keys**;
- Deze keys worden periodiek verzonden in een **EMM bericht** per abonnement;
- Een EMM bericht wordt encrypted met een per **smartcard** unieke **masterkey**;
- Op de smartcard staat deze masterkey en worden de keys bijgewerkt.

Cam (Conditional Access Module)

Wat is nu de plaats van de CAM in dit hele proces? Bij oudere ontvangers zit alle genoemde functionaliteit in de ontvanger. Deze kent dan ook meestal maar één encryptie zoals bijvoorbeeld de door Canal Digitaal “goedgekeurde” ontvangers. Het enige dat dit type ontvanger nog nodig heeft is een geldige smartcard. Met dit type ontvanger kunnen we dan ook uitsluitend FTA uitzendingen bekijken en de uitzendingen die zijn gecodeerd in de encryptie die de ontvanger kent.

Een modernere ontvanger heeft één of meerdere zogenaamde **CI slots (Common Interface)**. Een CI slot is een sleuf in de ontvanger waarin een zogenaamde **CAM (Conditional Access Module)** kan worden gestoken. Een CAM heeft de vorm van een PCM/CIA module met een sleuf waarin een smartcard kan worden geschoven. De benodigde functionaliteit om de encryptie aan te kunnen zit nu in de CAM. Zo zou je bijvoorbeeld een Conax Cam of een Seca Cam kunnen aanschaffen. Hierdoor hoef je voor iedere encryptie in ieder geval geen aparte ontvanger meer te hebben. Je verwisselt gewoon de CAM. Het enige dat je dan nog nodig hebt is een smartcard met een abonnement voor de programma's die je wilt kunnen zien.

Om er onder andere voor te zorgen dat fabrikanten van ontvangers en fabrikanten van CAM's onafhankelijk van elkaar producten kunnen ontwikkelen die toch met elkaar samenwerken is het uitwisselingsprotocol (de interface) tussen de ontvanger en de CAM gestandaardiseerd. Dit zijn de specificaties voor het CI slot en deze zijn vastgelegd in “*EN 50221 Common Interface Specification for Conditional Access and other Digital Video Broadcasting Decoder Applications*”. Omdat dit de specificaties zijn van hoe de CAM communiceert met de ontvanger, zijn dit voor onze studie ongelooflijk belangrijke specificaties.

UCAS CAM

Nog fraaier is de **UCAS (Universal Common Access System) CAM** die meerdere coderingen aankan, zoals bijvoorbeeld de Magic Module, de Matrix, de Xcam en de Dragon. Nu hebben we in principe nog maar één CAM nodig in onze ontvanger. En om het “ongemak” van geldige abonnementskaarten te voorkomen, hebben deze CAM's software aan boord, die naar de ontvanger toe net doen alsof ze een smartcard in de sleuf hebben. Deze software noemt men een **emulatie**. Het is deze emulatie die we downloaden van internet en in de CAM laden met onze CAS, Multipro, etc. De sleuteltjes die normaal van de kaart worden gehaald, zijn nu in het geheugen van de CAM geladen in de emulatie. Deze sleuteltjes kunnen in de meeste emulaties ook met de afstandsbediening van de ontvanger worden bewerkt. En als de software in de CAM er zelf niet uitkomt, kan hij altijd nog te rade gaan bij een titaniumkaart, funcard of andere all purpose card met gegevens van diverse providers.

En nu verder...

In dit hoofdstuk hebben we de algemene werking van (gecodeerde) uitzendingen via de satelliet in kaart gebracht en de plaats van de CAM hierin bepaald. In het volgende hoofdstuk gaan we de interne werking van de CAM nader onderzoeken en de wijze waarop de CAM met de ontvanger communiceert in beeld brengen.

Literatuurverwijzingen:

<http://www.duwgati.com>

<http://www.satinfo.org/archive/Documentation/CAS-model.pdf>

<http://www.nhk.or.jp/str/publica/bt/en/le0012.pdf>

<http://users.pandora.be/satelliet/mpegtrans.pdf>

<http://www.videoaudioreport.nl/index.php?action=1&catno=57&artno=1079&category=2004-april>

3. UCAS CAM (nog leeg)

4. Matrix CAM (nog leeg)

5. Programmeerkennis (nog leeg)

6. Coderingen (nog leeg)

7. Emulatiesoftware (nog leeg)

8. Smartcards (nog leeg)

Appendix 1: DVB Algemeen

(Overgenomen van : <http://mccb.be.eu.org/leden/krbonne/sat-tv.belgie.html#D4>)

4. DVB-S: Digital Video Broadcasting - Satellite

DVB-S is een systeem voor digitale uitzendingen van audio, video en data via satelliet. Het vormt eveneens de basis voor DVB-MS, de variant van DVB over LMDS/MVDS. Hier volgt een beschrijving over enkele belangrijkste elementen van digitale uitzendingen en DVB in het algemeen en DVB-S in het bijzonder.

4.1 Streams, kanalen en boeketten

Het eerste belangrijke verschil tussen analoge en digitale uitzendingen is dat het verband tussen een transponder en een TV-programma volledig verdwijnt. Bij analoge uitzendingen bestaat er een duidelijk één-op-één verband tussen een 'kanaal' (een transponder op een satelliet) en een TV-programma. Eén kanaal komt overeen met één programma en visa-versa. Bij digitale uitzendingen is dit helemaal niet meer van toepassing.

- Een satelliet-transponder kan één of meerdere 'transport-streams' bevatten.
- Een transport-stream kan één of meerdere TV-, radio- of data-kanalen bevatten.
- Een TV- of radio-programma bestaat uit een combinatie van een aantal audio- en/of video- en/of data-kanalen.
- Anderzijds worden verschillende programma's (mogelijk verspreid over verschillende transponders op één of meerdere satellieten) gebundeld tot één enkel boeket.

4.1.1. De transport-stream

Een 'transport-stream' is, zoals de naam het zegt, de 'transportlaag' van digitale TV. Het is een stroom van bits, uitgezonden door de satelliet, met een bepaalde snelheid, op een bepaalde frequentie en polariteit van een satelliet-transponder. Ze komt overeen met 'zoveel miljoen bits per seconde'.

Een ander verschil met analoge uitzendingen is dat een transport-stream NIET altijd de gehele transponder van de satelliet in beslag neemt. Soms kan een transport-stream slechts een deel van het spectrum van een transponder in beslag nemen, waardoor één transponder op een satelliet meerdere transport-streams kan bevatten. Soms bevindt zich op een transponder zowel een analog TV-kanaal als een digitale transport-stream.

4.1.2 De 'PES': Packetised Elementary Stream

- Binnenin de transport-stream bevindt zich één of meerdere 'substreams'; de zogenaamde 'PES' (Packetised Elementary Stream).
- Elke PES bevat één enkele uitzending: een TV-videosignaal, een geluidskanaal, teletekstinformatie, ondertitelinginformatie, of 'pure data'.
- Elke PES kan een verschillend bitrate hebben naar gelang het type informatie dat in de sub-stream opgeslagen zit. Het spreekt voor zich dat een videosignaal meer informatie moet bevatten dan een stream die enkel ondertitelings-informatie bevat.
- Elke PES wordt geïdentificeerd aan de hand van een 'PID' (PES Id).

4.1.3 Het boeket

Een boeket is een groepering van programma's van een bepaalde aanbieder (bv. een betaal TV-aanbieder). Een boeket kan zelfs verspreid zijn over verschillende transponders van een satelliet of zelfs over meerdere satellieten op dezelfde positie.

4.2: SRs, FECs: de beschrijving van een transport-stream

Een transport-stream wordt bepaald door vier waarden:

1. Satelliet-positie (bv. 'Astra 1' op 19,2 graden oost).
2. Frequentie en polarisatie: (bv. 12,574 GHz, horizontale polarisatie)
3. SR (Symbol Rate): zie hieronder
4. FEC (Forward Error correction), zie hieronder.

4.2.1. SR: Symbol rate

Indien men een transportstream op 'technisch' niveau bekijkt, dan is dat eigenlijk een radio-draaggolf die een vast aantal keer per seconde van fase verandert. Elke faseverandering noemt men een 'symbool', en omdat men bij DVB-S gebruik maakt van QPSK-modulatie, vertegenwoordigt elk symbool 2 bits.

De symbol-rate is het aantal keer per seconde dat de transport-stream van fase verandert. Het bepaalt dus effectief de hoeveelheid informatie per seconde wordt verstuurd door de totale transport-stream.

Eén van de eigenschappen van radio-communicatie bepaalt dat de totale hoeveelheid van het radio-spectrum (de zg. bandbreedte) dat een radio-signaal heeft rechtstreeks verband houdt met het aantal maal dat het radiosignaal per second verandert.

Voor digitale uitzendingen op satelliet betekent dit dat de bandbreedte die een transport-stream nodig heeft recht evenredig is met de symbol-rate van de transportstream (want de SR is net de maat voor het aantal veranderingen per second van de draaggolf).

Veel gebruikte symbol-rates zijn 27500 of 22000 Ksymbols/s omdat de benodigde bandbreedte voor zo'n signaal net overeenkomt met wat beschikbaar is op één volledige satelliet-transponder van bepaalde satellieten. Deze SR komt tegen 2 bits per symbol overeen met 55,5 of 44 MBps aan 'pure' bitrate.

4.2.2 Fout-correctie: Solomon-Reed en FEC

Een radioverbinding is nooit perfect, en ook bij satelliet-verbindingen is het altijd mogelijk dat er fouten optreden tijdens het oversturen van een signaal.

Omdat een omroepsysteem slechts in één richting werkt (van satelliet/zender naar de ontvanger) wordt hier gebruik gemaakt van "FEC" (ofwel "Forward Error Correction"). Bij dit soort systeem wordt reeds bij het uitzenden van het radiosignaal extra informatie meegestuurd zodat -indien er fouten optreden bij het oversturen van de bits van de zender/satelliet naar de ontvanger- de ontvanger dit kan detecteren en indien mogelijk ook de fouten kan corrigeren. De error-correctie bits worden dus vooraf doorgestuurd, vandaar de naam "forward error correction".

Een transport-stream van een satelliet-verbinding wordt 'beschermd' door twee verschillende error correctie-technieken: de 'outer-coding' (meestal aangeduid met de naam 'Reed Salomon') en de 'inner-coding' (gewoon aangeduid met 'FEC').

De eerste (Reed-Salomon) neemt een vast percentage van de transport-stream in beslag (8%) en kan niet worden gewijzigd. De tweede (FEC) is wel instelbaar en wordt aangeduid met een breuk. Een FEC van bv. '3/4' betekent dat er per 3 bits 'echte' gegevens er een 4de bit meegestuurd wordt voor error-correctie. Veel gebruikte FEC-waarden zijn 1/2, 3/4, 5/6 of 7/8.

4.2.3 Een voorbeeld: BVN

Bovendien, voorbeelden zijn altijd een stuk duidelijker dan saaie getallen.

Dit zijn de gegevens voor de transport-stream waarin de TV-zender 'BVN' wordt uitgezonden. (BVN = "Beste van Vlaanderen en Nederland", een zender uitgebaat door de VRT en de openbare omroepen uit Nederland)

Positie: Astra 1 (op 19,2 graden oost)

Transponder: 12,574 GHz, horizontale polarisatie

transport-stream: Symbol rate 22000 (Ksymbols/s), FEC 5/6

4.3 De PES: binnenin de transport-stream

Een transport-stream is echter maar de pure 'drager' van de binaire informatie. De eigenlijk informatie die men wenst te bekijken of te beluisteren bevindt zich 'binnenin' de transport-stream in de verschillende 'substreams' of (in het juiste vakjargon) de 'PES' (Packetised Elementary Stream) in de transport-stream.

Even herhalen wat reeds kort werd besproken in 4.1.2.

- Elke PES bevat één stroom met 'informatie'. Deze informatie kan verschillend zijn van aard. Bijvoorbeeld:
 - beeld (video)
 - geluid (mono, stereo, surround sound, ...),
 - teletext-informatie,
 - ondertitel-informatie,
 - 'pure data' (bv. internet-data).
- Elke PES wordt aangeduid via een nummer: de 'PID'.

Daarnaast bevat een transport-stream echter ook nog een hoeveelheid administratieve informatie:

- Benaming en parameters van de individuele substreams.
- Het beeldkanaal en het geluidskanaal die moeten worden gekoppeld voor het verkrijgen van een 'programma'.
- Informatie over andere transport-streams.
- 'Klok'-informatie, nodig om het geluid van een film synchroon te houden met het beeld: de PCR-stream.
- Informatie of een programma al dan niet geëncrypteerd is.
- De 'EPG' ('electronic program guide', de elektronische programmagids)

4.4 En uiteindelijk: het programma

4.4.1 Een TV-programma zoals we het nu kennen

De laatste stap die nog moet gebeuren is het opbouwen van een 'programma': datgene waar wij als kijker naar kijken. Dat verkrijgt men gewoon door verschillende substreams te combineren. Voor een TV-kanaal is dat minimaal één video-sigitaal samen met één geluidskanaal.

Verder moet ook worden vermeld dat er -onzichtbaar voor de gebruiker- altijd nog een 'PCR' (klok) signaal wordt meegestuurd. Dit is nodig om de verschillende datastromen synchroon met elkaar te laten lopen. Indien dit niet klopt, bv. waarbij het geluid voor of achter loopt t.o.v. het beeld, dan spreekt men over 'lip sync' problemen.

4.4.2 Een TV-programma zoals het kan zijn

Echter, één van de voordelen van digitale TV, is dat men in principe alles met alles kan combineren. Hoewel het samenvoegen van één videosignaal met één geluidssignaal de meest logische combinatie is, zijn er veel meer mogelijkheden. Enkele voorbeelden:

Een 'uitgebreid' TV-kanaal. Dit bevat naast de video-stream (het beeld) en de audio (het geluid) eveneens 'teletekst', ondertitels en de elektronische programma gids (EPG).

Soms bevat een TV-kanaal meerdere geluidskanalen: bv. verschillende talen (zoals EbS, Eurosport of Arte) of een combinatie van een 'gewoon' audio-kanaal met een 'Surround'-geluidskanaal.

Het is ook denkbaar om meerdere videokanalen samen te voegen met één enkel audiokanaal. (Bv. een sportwedstrijd bekeken vanuit verschillende camerastandpunten).

Een andere mogelijkheid: een 'onderwijs' TV-kanaal, bestaande uit beeld, geluid en een 'data'-kanaal waarlangs de slides van de cursus worden doorgestuurd.

4.4.3 Het programma: de gegevens

Al deze extra gegevens (Audio-PID, Video-PID, enz.) bepalen samen met de gegevens van de transport-stream het TV-programma. Nemen we opnieuw het programma 'BVN' (zie 4.2.3), dan krijgen we de volgende gegevens:

De transport-stream:

- Astra 1 (op 19,2 graden oost),
- 12,574 GHz, horizontale polarisatie,
- SR 22000, FEC 5/6

De gegevens van het programma 'BVN' binnenin deze transport-stream:

- Video-PID 516
- Audio-PID 690

Appendix 2: Releasenotes

Versie 1.0 (3 oktober 2004)

Document hoofdstukindeling gemaakt en hoofdstukken 1.”Inleiding” en 2.”Een globaal overzicht” geschreven. Appendix 1 gekopieerd van internet en appendix 2 met de releasenotes gemaakt. In deze versie zitten nog een aantal aannames van mij, waarvan ik hoop dat deze door anderen bevestigd of tegengesproken worden. Kleun ik er compleet naast, hoor ik het ook graag. Verder is de smartcard is momenteel nog een wat onderbelicht onderwerp. Misschien moet ik als gevolg hiervan de hoofdstukindeling in de toekomst herzien. Ook kan gedurende de studie blijken dat een andere volgorde wellicht handiger is. Ik ga vooralsnog verder met hoofdstuk 3. “UCAS CAM” door, in afwachting van commentaar op deze versie. Dit commentaar zal ik verwerken in versie 1.1 en hoger. In versie 2.0 zal nieuwe informatie zijn opgenomen.