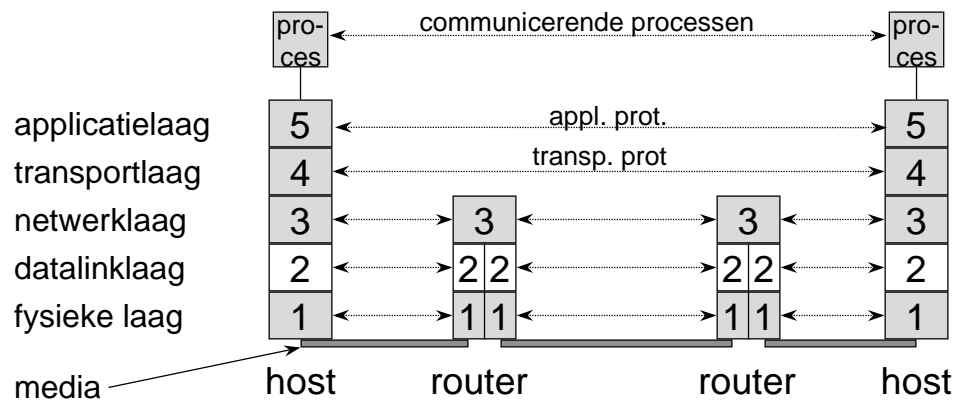
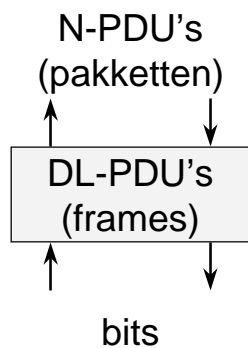


3. De datalink-laag

Datalinklaag



Datalinklaag (laag 2)



Doel:

Betrouwbare en efficiënte communicatie tussen twee aangrenzende machines.

- foutbeheersing;
- stroombeheersing.

(Bits worden daartoe in blokken gegroepeerd (frames) met speciale bitpatronen aan begin en eind om voor de herkenning)

De zender zendt o.a. dataframes, de ontvanger bevestigingsframes.

Noodzaak van een Datalink laag

Datalink laag maakt van fysieke "lijn" een kanaal waarover bits op de juiste wijze worden overgedragen.

Rekening te houden met:

- Transmissiefouten (foutkans)
- transmissiesnelheid (bit/s)
- transmissielooptijd (μ sec/km)
- verwerkingstijd machines

3.1.1 Aan Netwerklaag geboden diensten

Verbindingsloos

- Onbevestigd verbindingsloos
 - onafhankelijke frames
 - geen poging tot herstel
 - bij lage foutkans
 - bij tijdgebonden data ("real time"), bijv. spraak.
- Bevestigd verbindingsloos
 - onafhankelijke frames
 - elk frame wordt bevestigd
 - nogmaals verzenden indien fout

Geboden diensten (2)

Verbindingsgericht

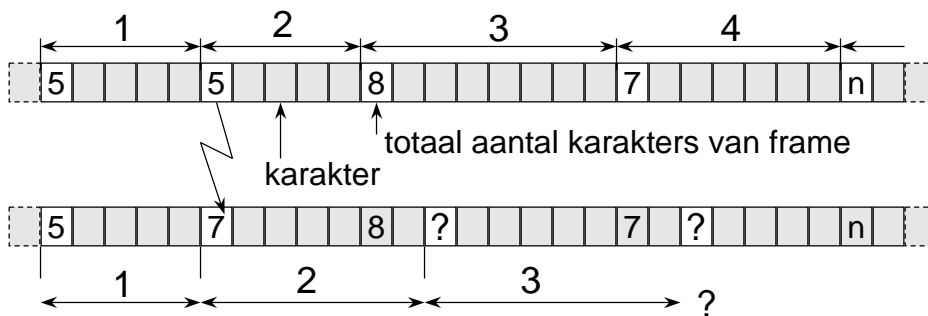
- Tot stand brengen verbinding
(faciliteiten, variabelen, tellerstanden initialiseren)
- verzenden
- Opheffen verbinding
(vrijgeven faciliteiten, variabelen, buffers)

3.1.2 Frame-indeling

Datalinklaag deelt berichten op in blokken (*frames*):

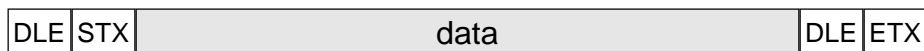
- Fysieke laag maakt fouten. DL-laag tracht die te herstellen: per frame een vast aantal bits reserveren voor detectie of correctie
- Datalinklaag moet de bitstroom (snelheid) kunnen regelen, doet dat op basis van frames.
- Verschillende markeringsmethoden voor frames

a. Tellen van karakters



Foutgevoelig, weinig meer gebruikt.

b. Begin- en eind-karakters



- DLE, STX of DLE, ETX geven framegrenzen aan.
- Deze combinaties zouden in de data kunnen voorkomen.
- Vóór het toevoegen van de framegrenzen in de data
DLE => DLE,DLE (*character stuffing*)
- In de bij de ontvanger binnenkomende bitstroom geven slechts DLE, STX of DLE, ETX framegrenzen aan.
- DLE, DLE, STX en DLE, DLE, ETX worden als data herkend en weer omgezet in DLE, STX of DLE, ETX
(DLE = DataLink Escape, STX=Start of Text, ETX=End of Text)

Voorbeeld *character stuffing*

Data A | B | DLE|STX | C | D | DLE|ETX | -----

Voor verzenden DLE => DLE | DLE | :

 A | B | DLE|DLE|STX | C | D | DLE|DLE|ETX | -----

Begin en eindkarakters toevoegen:

DLE|STX | A | B | DLE|DLE|STX | C | D | DLE|DLE|ETX | ----- | DLE|ETX



Overzenden

De paren DLE|STX en DLE|ETX opsporen en verwijderen:

 A | B | DLE|DLE|STX | C | D | DLE|DLE|ETX | -----

 DLE | DLE | => DLE | :

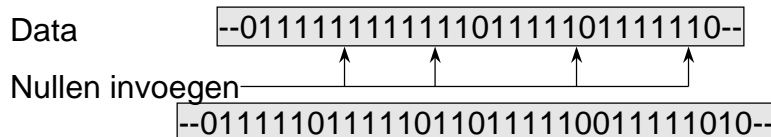
 A | B | DLE|STX | C | D | DLE|ETX | -----

c. Begin en eindvlaggen

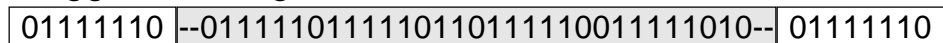


- Vlagpatroon kan ook in de data voorkomen
- Voor verzenden dit patroon verwijderen door na elke 5 énen in de data daar een "0" in te voegen
(*bit stuffing*).
- De ontvanger detecteert de vlaggen, en verwijdert ze
- In de data wordt na elke 5 énen een "0" weggehaald.

Voorbeeld *bit stuffing*

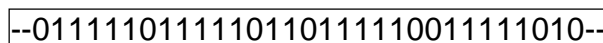


Vlaggen toevoegen:

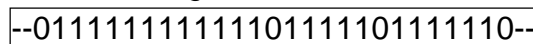


Overzenden

De paren vlaggen opsporen en verwijderen:



Na elke 5 énen een "0" weghalen:



d. Bijzondere signalen in de fysieke laag

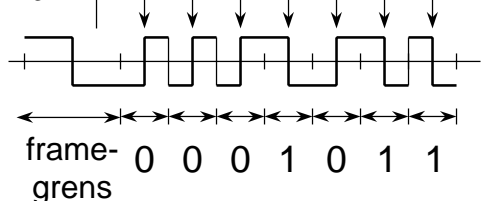
Voorbeeld: Manchester-codering.

Normale "0":  normale "1": 

Elk bit heeft een signaalwisseling in het midden

Framegrens: 

Vb.:



Een normaal bit heeft een signaalwisseling in het midden, bij een framegrens ontbreekt dat op die plaats.

3.2 Foutdetectie en -correctie

- Fouten komen met vlagen (*bursts*)
- Indien fouten onafhankelijk => veel foute frames
- Indien fouten in vlagen => weinig foute frames, echter wel ernstig beschadigd
- Door redundante codering *foutdetectie* of eventueel *foutcorrectie (Forward Error Correction)* mogelijk.

Hamming-afstand

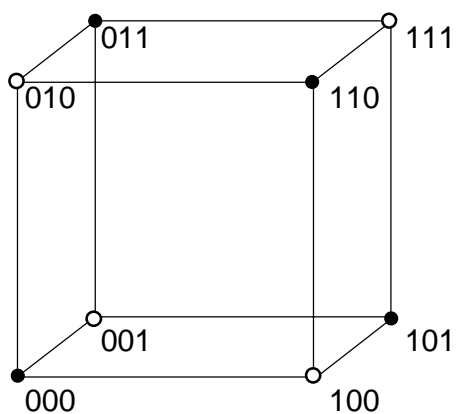
- Een n-bits code bestaat uit een verzameling codewoorden van n bits. (niet iedere combinatie van n bits hoeft een geldig codewoord te zijn)
- Hamming-afstand van twee codewoorden, is het aantal bits dat ze verschillen.
- Hamming-afstand van een code is het minimum van de Hamming-afstanden van alle mogelijke paren van codewoorden.

Hamming-afstand

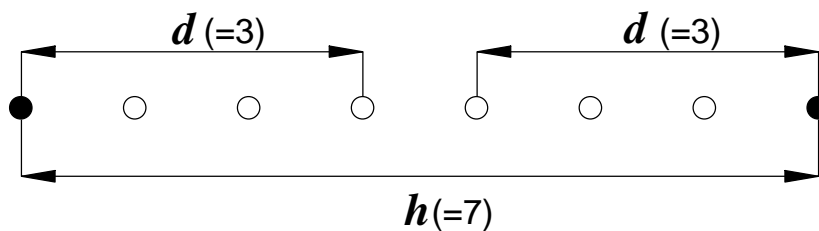
Voorbeeld:

Driebitscode met Hammingafstand 2

- Gebruikte code
- Niet gebruikte code



Hammingafstand nodig voor correctie



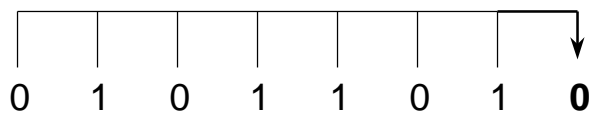
Herstel mogelijk indien: $h \geq 2d + 1$

h = Hammingafstand [bits]
 d = aantal foute bits

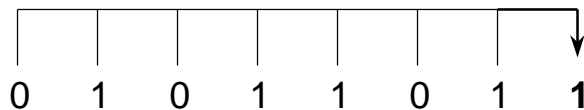
of $d \leq \left\lfloor \frac{h-1}{2} \right\rfloor$

Eenvoudige pariteit

Even pariteit

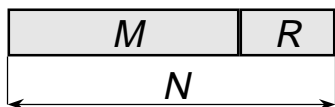


Oneven pariteit



3.2.1 Eén bit kunnen herstellen

Hoeveel bits nodig?



$M =$ databits
 $R =$ bits voor foutcontrole

- De databits hebben 2^M verschillende bitpatronen
- Per bitpatroon N patronen door een fout van één bit

$$2^M + 2^M \cdot N \leq 2^N$$

Oorspronkelijk aantal bitpatronen \uparrow 2^M
 Aantal foute bitpatronen \uparrow $2^M \cdot N$
 Totaal aantal bitpatronen \uparrow 2^N

Met $N = M + R$ volgt $M + R + 1 \leq 2^R$
 (R bepalen door proberen)

Hoeveel bits toevoegen voor herstel?

$M + R + 1 \leq 2^R$ (Herstel van één fout bit in bericht)

$M = 1$	$R + 2 \leq 2^R$
$R = 1$	$3 \leq 2?$
$R = 2$	$4 \leq 4$

$M = 2$	$R + 3 \leq 2^R$
$R = 2$	$5 \leq 4?$
$R = 3$	$6 \leq 8$

$M = 3$	$R + 4 \leq 2^R$
$R = 3$	$7 \leq 8$

$M = 4$	$R + 5 \leq 2^R$
$R = 3$	$8 \leq 8$

$M = 5$	$R + 4 \leq 2^R$
$R = 3$	$9 \leq 8?$
$R = 4$	$10 \leq 16$

$M = 6$	$R = 4$ $11 \leq 16$
	\vdots \vdots
$M = 11$	$R = 4$ $16 \leq 16$

Aantal extra bits nodig

(Herstel van één fout bit)

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+															
=															
R	2	3	3	3	4	4	4	4	4	4	4	5	5	5	5
=															
N	3	5	6	7	9	10	11	12	13	14	15	17	18	19	20

M = netto aantal bits van het bericht (databits)
 R = aantal benodigde controlebits
 N = totaal aantal bits (machten van 2 ontbreken!)

in3001 Datacommunicatie en Netwerken: Datalink laag 21

Hammingcodering

- Nummer de plaatsen, begin links met 1
- Plaats pariteitsbits op plaatsnummers die machten van 2 zijn (1, 2, 4, 8, 16...)

P	P	3	P	5	6	7	P	9	10	11	12
----------	----------	---	----------	---	---	---	----------	---	----	----	----

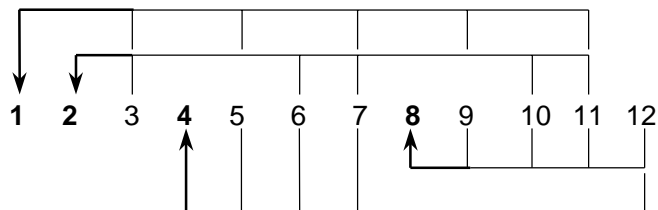
- Plaatsnrs. van de databits ontbinden in som van machten van 2 (Vb. 11=1 + 2+ 8)
- Een pariteitsbit telt die databits waarbij zijn plaatsnr. in de "ontbinding" voorkomt

1 telt	3	5	7	9	11	...
2 telt	3	6	7	10	11	...
4 telt	5	6	7		12	...
8 telt			9	10	11	12

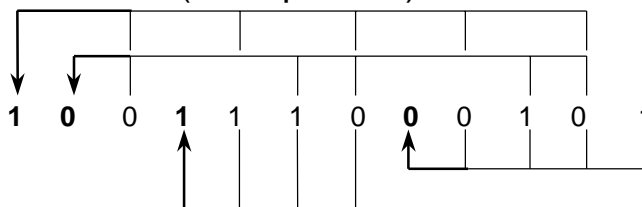
in3001 Datacommunicatie en Netwerken: Datalink laag 22

Hammingcodering (voorbeeld)

$$M = 8 \Rightarrow R \geq 4$$



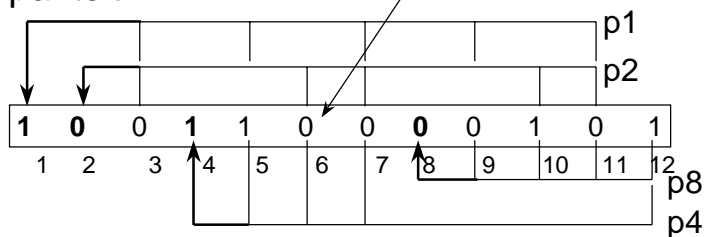
Databericht 0 1 1 0 0 1 0 1 (even pariteit.):



Voorbeeld: correctie van fout

Even pariteit

Fout bit op plaats 6



p1 3 5 7 9 11

p2 3 6 7 10 11

p4 5 6 7 12

p8 9 10 11 12

plaats 6 is de enige die alleen in p2 en p4 voorkomt

Welk bit is er dan fout?

fout in bit	indien fout in pariteitsbits
1	p1
2	p2
3	p1 & p2
4	p4
5	p1 & p4
6	p2 & p4
7	p1 & p2 & p4
8	p8
9	p1 & p8
10	p2 & p8
11	p1 & p2 & p8
12	p4 & p8

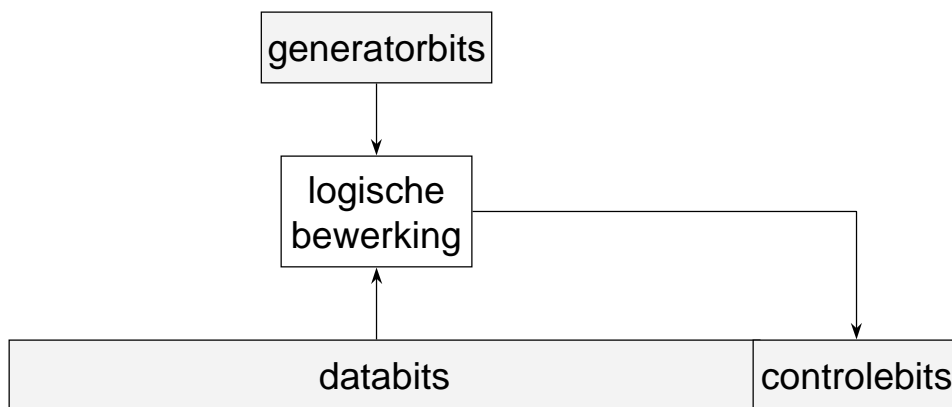
Detectie van fouten door bursts

codewoord	d	a	t	a	c	o	m
1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0
4	0	0	0	0	0	1	1
5	1	0	1	0	0	1	1
6	0	0	0	0	1	1	0
7	0	1	0	1	1	1	1
P	0	0	1	0	1	1	0

volgorde van verzenden

3.2.2 Cyclic Redundancy Code (CRC)

Uit een afgesproken bitrij, de generator, en de databits worden de controlebits berekend



Bitreeks voorgesteld door een veelterm

$$a_{k-1}x^{k-1} + a_{k-2}x^{k-2} \dots a_2x^2 + a_1x + a_0$$

Machten van x geven het plaatsnummer aan; de bijbehoren coëfficiënten a_i zijn 1 of 0.

Voorbeeld:

1	1	0	0	0	1
x^5	$+ x^4$				$+ x^0$

Voorbeeld:

1	1	0	1	0	0	1	0
x^7	$+ x^6$		$+ x^4$			$+ x$	

Modulo-2 rekenen

$$0 \text{ +/- } 0 = 0$$

$$1 \text{ +/- } 1 = 0$$

$$1 \text{ +/- } 0 = 1$$

$$0 \text{ +/- } 1 = 1$$

$$\begin{array}{r} \text{Vb.: } x^3 + x \\ \underline{x^3 + 1} \text{ +/-} \\ x + 1 \end{array} \quad \begin{array}{r} 1010 \\ \underline{1001} \text{ +/-} \\ 0011 \end{array}$$

$$\begin{array}{r} \text{Vb.: } x^3 + x \\ \underline{x^3 + x} \text{ +/-} \\ 0 \end{array} \quad \begin{array}{r} 1010 \\ \underline{1010} \text{ +/-} \\ 0000 \end{array}$$

Nullen toevoegen aan bitrij

Een bitrij voorgesteld door polynoom $M(x)$.

Die zelfde bitrij maar dan met r nullen erachter

is dan $x^r M(x)$

Voorbeeld: $x^5 + x^4 + x + 1 \triangleq 110011$

$$x^4(x^5 + x^4 + x + 1) =$$

$$x^9 + x^8 + x^5 + x^4 \triangleq 1100110000$$

Algoritme CRC

- Afgesproken generatorveelterm $G(x)$ met graad r
- Bericht $M(x)$
- Maak $x^r M(x)$
- Deel dit door $G(x)$
- Tel de rest van de deling, $R(x)$, op bij $x^r M(x) \Rightarrow$
- Verstuur $x^r M(x) + R(x)$ (Dit is deelbaar door $G(x)$)
- Bij de ontvanger wordt $x^r M(x) + R(x)$ gedeeld door $G(x)$
- Is de rest 0 dan is het bericht goed
- Is de rest *niet* 0 dan is het bericht beschadigd

Algoritme CRC (2)

$$\frac{x^r M(x)}{G(x)} = D(x) + \frac{R(x)}{G(x)}$$

$$x^r M(x) = D(x) \cdot G(x) + R(x)$$

$$\underbrace{x^r M(x) + R(x)}_{\substack{\text{versturen;} \\ \text{is deelbaar door } G(x)}} = D(x) \cdot G(x) + \underbrace{\cancel{R(x)} + \cancel{R(x)}}_{\substack{\text{is 0 bij mod 2} \\ \text{optelling}}$$

Voorbeeld CRC berekening

Generatorveelterm: $x^3 + x + 1$ (1 0 1 1)
 Data: x^3+1 (1 0 0 1)
 Data met nullen: x^6+x^3 (1 0 0 1 0 0 0)

$$\begin{array}{r}
 x^3 + x + 1 \ / \ x^6 + \quad x^3 \ \backslash \\
 \underline{x^6 + x^4 + x^3} \\
 \quad x^4 \\
 \quad \underline{x^4 + \quad x^2 + x} \\
 \text{rest} \Rightarrow \quad \quad \quad x^2 + x
 \end{array}$$

Verzonden bericht:
 $x^6+x^3 + x^2 + x$

Of direct met enen en nullen:

$$\begin{array}{r}
 1011 \ / \ 1001000 \ \backslash \\
 \underline{1011} \\
 \quad 1000 \\
 \quad \underline{1011} \\
 \quad \quad 110 \\
 \text{rest} \Rightarrow \quad 110 \\
 \text{bericht met nullen} \Rightarrow \underline{1001000} \\
 \text{verzonden bericht} \Rightarrow \underline{1001110}
 \end{array}$$

Generatorveeltermen

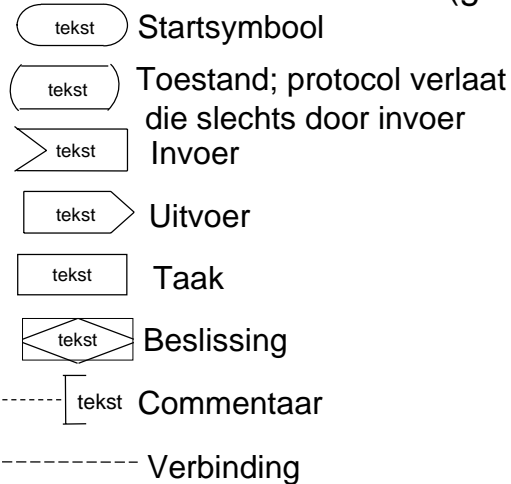
ITU-T $G(x) = x^{16} + x^{12} + x^5 + 1$ <==

CRC-16 (IBM) $G(x) = x^{16} + x^{12} + x^2 + 1$

CRC-12 $G(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1$

3.3 Elementaire datalinkprotocollen

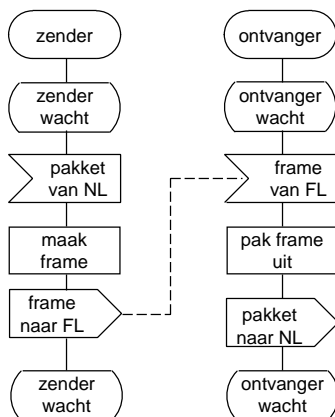
SDL- Specification and Description Language
(genormaliseerd door ITU)



3.3.1 Onbeperkt simplex protocol (prot. 1)

“Utopia”

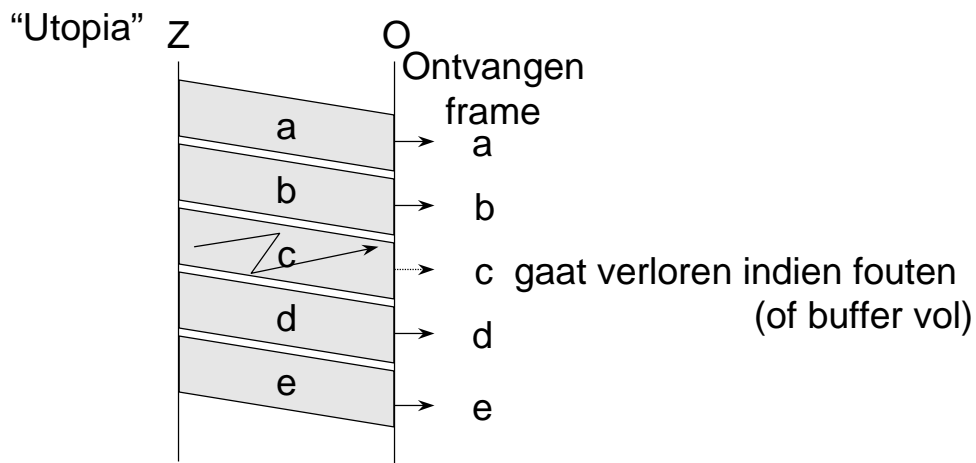
- geen foutcorrectie
- geen stroombeheersing



Dus niet geschikt voor kanaal met fouten of voor trage ontvanger.

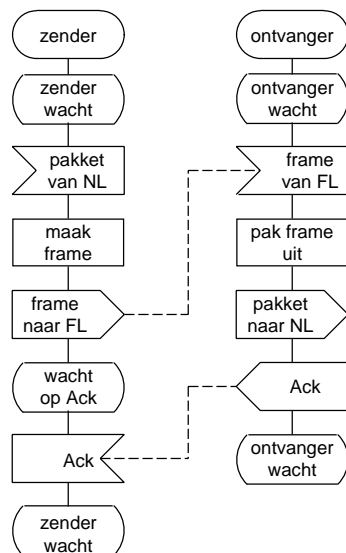
NL = netwerklaag
FL = fysieke laag

Onbeperkt simplex protocol



Utopia = Idealistisch maar niet realistisch

3.3.2 Simplex “stop and wait” (prot. 2)

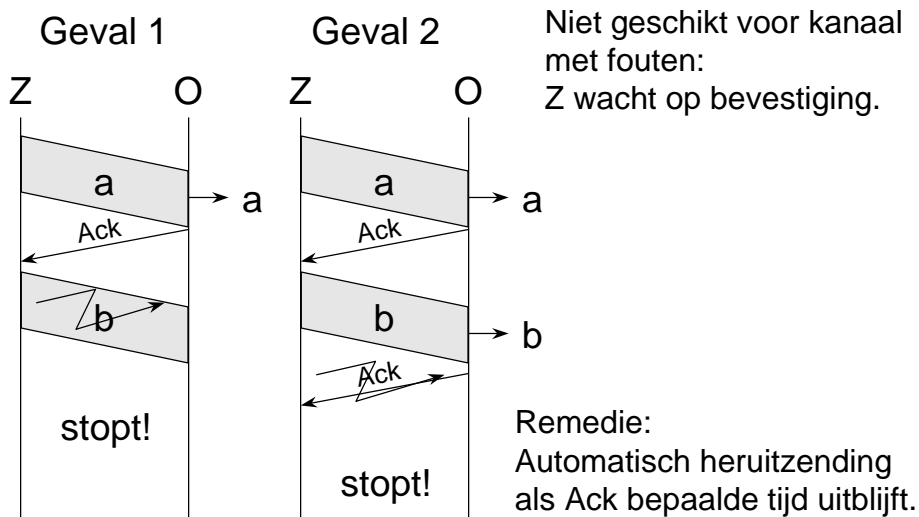


- geen foutcorrectie
- wel stroombeheersing

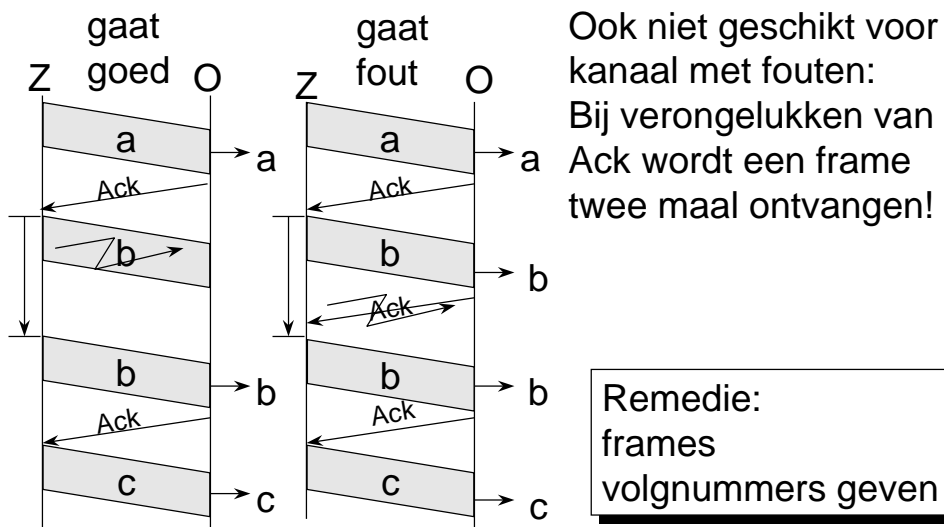
Dus niet geschikt voor kanaal met fouten

Ack = acknowledgement
(bevestigingsbericht)

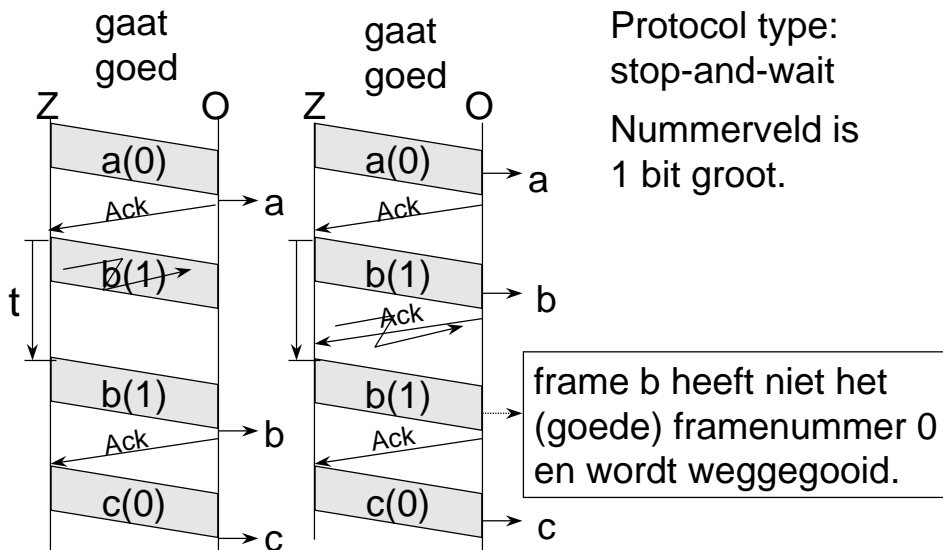
Simplex "stop and wait" (prot. 2)



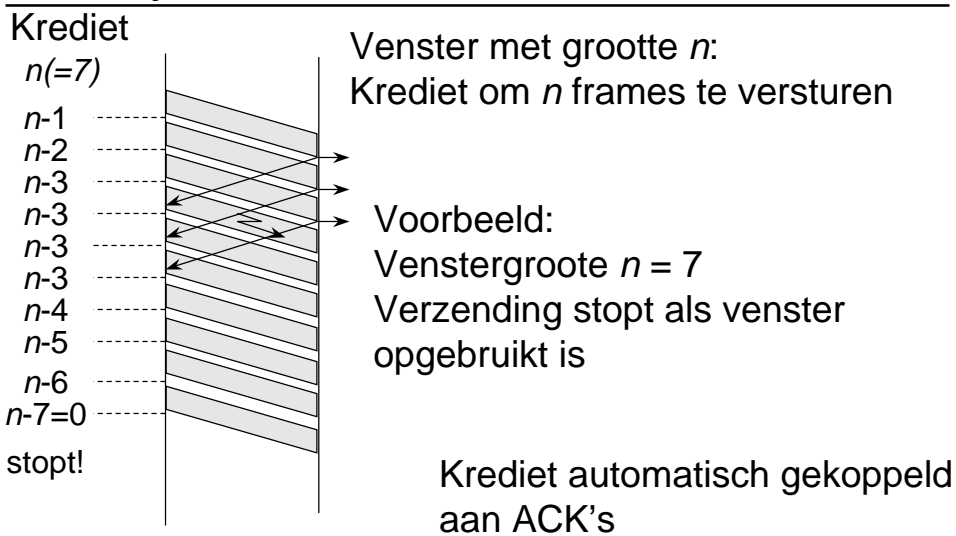
Simplex "stop and wait" met heruitzending



3.3.3 PAR-protocol (prot. 3)



3.4 Glijdend venster



Glijdend venster

Krediet

$n(=7)$

$n-1$

$n-2$

$n-3$

$n-4$

$n-5$

$n-6$

$n-7$

$n-6$

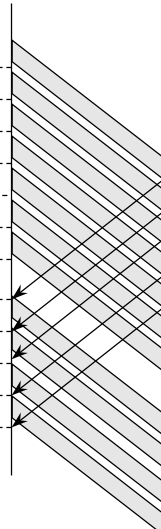
$n-6$

$n-6$

$n-6$

$n-6$

$n-6$



Is venster groot genoeg dan kunnen frames aansluitend worden verzonden: "pipelining"

In dit voorbeeld is de looptijd op de verbinding te groot om aansluitende verzending van frames mogelijk te maken.

Glijdend venster

Zendvenster:

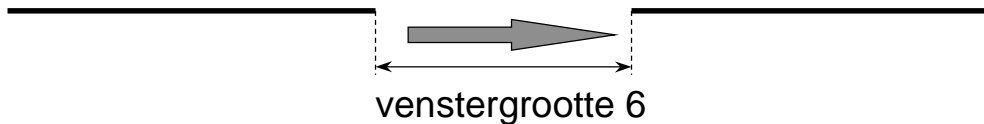
Volnummers die verzonden mogen worden + die verzonden zijn maar nog niet bevestigd

Ontvangstvenster:

Volnummers die geaccepteerd mogen worden.

Volnummers vaak 3 bits (0 ..7) of 7 bits (0 .. 127)

6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4



venstergrootte 6

Piggy Backing

- *Piggy Backing* = het laten meereizen van een bevestiging in de header van een retourbericht. Dit heeft tot gevolg dat een bevestiging soms niet meteen wordt gezonden, maar pas als er een bericht de andere kant opgaat

Error afhandeling bij pipelining

Een fout frame wordt pas laat door zender ontdekt, twee mogelijke wijzen van aanpak

- ***Go back n***

Bij een fout:

- ontvanger gooit alle volgende frames weg, zendt geen bevestigingen
- zender zendt na een time-out alle niet bevestigde frames opnieuw

- ***Selective Repeat***

Bij een fout

- ontvanger buffert alle volgende goed ontvangen frames
- zender zendt alleen foute frame opnieuw

3.6 Voorbeelden datalink-laag protocollen

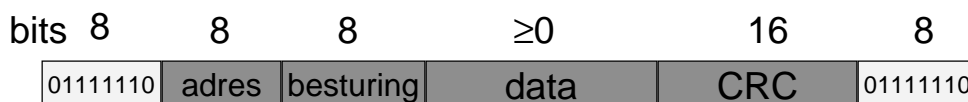
HDLC familie (bit oriented, bit stuffing)

- SDLC - Synchronous Data Link Control (IBM 1969)
- ADCCP - Advanced Data Communication Control Protocol (ANSI 1971)
- HDLC - High-level Data Link Control (ISO)
 - subset van HDLC: LAP - Link Access Procedure (onderdeel van X.25)

X.25 is een toegangsprotocol voor datanetwerken, genormaliseerd door de ITU-T (Internat. Telecomm. Union, een overlegorgaan van PTT's.)

High level Data Link Control

HDLC-frame



- Information frame - Verbindingsgericht datatransport. Besturingsveld bevat o.a. volgnr.
- Supervisory frame - Begeleiding van verbindingsgericht datatransport. Besturingsveld bevat o.a. volgnr. (dataveld ontbreekt)
- Unnumbered frame - Beheer van verbinding en voor verbindingsloos datatransport

PPP (Point-to-Point Protocol)

Ondersteunt:

- Foutdetectie
- Verscheidene netwerkprotocollen
- Dynamisch toekennen van IP-adressen
- Authentiseren
- LCP (Link Control Protocol)
Onderhandelen over datalinkverbinding, maken en verbreken van verbinding
- NCP (Network Control Protocol)
Onderhandelen over netwerkopties

PPP

PPP-frame lijkt veel op HDLC frame maar gebruikt byte stuffing (0x7E) ipv bitstuffing (dataveld geheel aantal bytes)

bytes 1 1 1 1 of 2 variabel 2 of 4 1

01111110	adres	besturing	protocol	data	CRC	01111110
----------	-------	-----------	----------	------	-----	----------

adres 1 1 1 1 1 1 1 1 (iedereen)

besturing 0 0 0 0 0 1 1 (ongenummerd frame)

dit is de verstekwaarde, framenummering is wel mogelijk