7. Sicherungsschicht

1. Aufgaben der Sicherungsschicht
2. Fehlererkennung und Fehlerkorrektur
3. Flußsteuerung
4. High-level Data Link Control (HDLC)
5. PPP – Point-to-Point-Protocol

7.1.1. Kennzeichnen der Rahmengrenzen

a) mittels Anfangs- und Endflags (Begrenzer):
Rahmenformat:

SD	Steuerinformation von Schicht 2	Daten	ED

SD := Startdelimiter, ED := Enddelimiter
Beispiel: SD = ED = 0 11 11 11 0

Diese Zeichenkette darf in den Daten nicht vorkommen.
$\Longrightarrow$ Bitstopfen := nach 5 aufeinander folgenden Einsen fügt der Sender eine 0 ein. Empfänger streicht nach 5 aufeinander folgenden Einsen eine Null heraus.

Beispiel: HDLC

Daten:		0110111111011111001
Übertragen wird:	SD	01101111101011111 0 001 ED

b) Mischverfahren: SD + Feld mit Längenangabe
$\Longrightarrow$ Schicht 2 muß Anzahl Bits zählen
Beispiel: Ethernet

c) Verstöße gegen Kodierungsregeln
Beispiel: Tokenbus

7.1.2. Fehlerkontrolle

Link-to-Link-Überprüfung:
Wurde der Rahmen korrekt übertragen?

Ansatz: Sender berechnet und schickt eine Prüfsumme (Redundanzbits).

Der Empfänger berechnet ebenfalls die Prüfsumme für den empfangenen Rahmen. „Stimmt" die Prüfsumme, dann ist die Übertragung („höchstwahrscheinlich") korrekt und die Daten werden an Schicht 3 weitergegeben. Andernfalls wird der Rahmen verworfen.

Beispiel: Hamming-Code, Cyclic Redundancy Code (CRC)

7.1.3. Flußsteuerung

Problem:
Unterschiedlich schnell arbeitender Sender und Empfänger.
Falls der Empfänger langsamer arbeitet als der Sender, können beim Empfänger Datenpuffer überlaufen und Rahmen verloren gehen.
$\Longrightarrow$ Protokolle zur Synchronisation des Datenflusses zwischen Sender und Empfänger,
z.B. Stop-and-wait-Protokolle, Schiebefenster-Protokolle

Ansatz: Codewort der Länge $m + r$

m Datenbits	r Redundanzbits

Fehlerkorrekturcodes

2^m gültige Wörter

$b_1 \dots b_m \overset{\text{Codierung}}{\mapsto} b_1 \dots b_m \quad b_{m+1} \dots b_{m+r}$

Fehler: $b_1 \dots \overline{b_i} \dots b_m$

Definition 33:

Der Hamming-Abstand $h(b_1 \dots b_n,\ b'_1 \dots b'_n)$ ist die Anzahl der unterschiedlichen Bits, d.h. die Anzahl $i \quad \text{mit} \quad \mathrm{b_i} \neq \mathrm{b'_i}$.

Definition 34:

1. Ein Übertragungsfehler heißt l-Bit-Fehler, falls $h(b_1 \dots b_n,\ r(b_1 \dots b_n)) = l$ gilt, wobei $r(b_1 \dots b_n)$ das empfangene Wort bezeichne.

Theorem 4: Gegeben sei eine Codierung $\mathcal{C}$

$$\mathcal{C}(b_1 \dots b_m) \quad = \quad b_1 \dots b_m \quad b_{m+1} \dots b_{m+r}$$

a) Ist der minimale Hammingabstand aller gültigen Codewörter $d+1$ so lassen sich alle l-Bit-Fehler, $l = 1, \dots d$ erkennen.

b) Der minimale Hammingabstand aller Codewörter ist $2d+1$. $\Longleftrightarrow$ Es lassen sich alle l-Bit-Fehler, $l = 1, \dots d$ erkennen und korrigieren.

Beispiel für 1-Bit-Korrekturcode (Hamming 1950)

Das Datenwort wird aufgefüllt mit Redundanzbits (Checkbits) an den Positionen b_i , $i = 2^j$, $j = 0, 1, 2, \ldots$

Checkbits sind Parity Bits:
Bit an Position k wird von allen Checkbits b_{2^j} geprüft, für die gilt, daß 2^j in der Dualzahldarstellung von k vorkommt.

Beim Empfangen werden alle Checkbits überprüft. Falls Checkbit b_i falsch, setze counter: = counter $+i$
Bit b_k verfälscht $\Rightarrow$ alle Checkbits b_i, die b_k mitprüfen sind verfälscht

$$\begin{aligned} \text{counter} = 0 &\iff \text{Codewort ist korrekt empfangen} \\ \text{counter} = k &\iff \text{Bit } b_k \text{ ist verfälscht} \end{aligned}$$

Bemerkung:

1. Mit dem Hamming-Code können 1-Bit-Fehler erkannt und **korrigiert** werden.
2. Wird im IEEE 802.16 Netzen (drahtlose Breitbandnetze, Wireless Man) benutzt. Bekannt unter dem Namen WiMAX (Worldwide Interoperability for Microwave Access)

WAN: WiMax war im Bereich Drahtlose Netze ca. 2005-2009 eine Pionier-Technologie zur Abdeckung von Distanzen von mehreren Kilometern (also deutlich längere Strecken als WiFi). WiMax benutzt schon MIMO (Multiple-Input and Multiple-Output Antennas), um höhere Bandbreiten zu erzielen.

Statt WiMAX hat sich mittlerweile das später entwickelte LTE (4G) durchgesetzt.

Fehlererkennung mittels Cyclic Redundancy Code (CRC)

zyklische Blocksicherung, Polynomcodes

$$
\begin{array}{lcl}
b_1 \ldots b_m & \longmapsto & b_1\, x^{m-1} + b_2\, x^{m-2} + \ldots + b_{m-1}\, x^1 + b_m \\
1\,1\,0\,0\,0\,1 & \longmapsto & x^5 + x^4 + \ldots + 1 =: M(x)
\end{array}
$$

gegeben: „Generatorpolynom" $G(x)$

Idee: Das zum Codewort zugehörige Polynom soll in $Z_2[x]$ durch $G(x)$ teilbar sein.

Fehlererkennung: Das empfangene Wort wird auf das zugehörige Polynom abgebildet.

Geht Polynomdivision in $Z_2[x]$ mit $G(x)$ auf?

Falls $\begin{cases} \text{Ja} & \hat{=} \quad \text{höchstwahrscheinlich kein Fehler} \\ \text{Nein} & \hat{=} \quad \text{Fehler!} \end{cases}$

Algorithmus zur Berechnung des Codeworts

Codewort $b_1 \ldots b_m \quad \underbrace{b_{m+1} \ldots b_{m+r}}_{\textit{Checksumme}}$

1. $r := \text{Grad}\,(G\,(x))$

2. Polynomdivision mit Rest:
   Bestimme das Restpolynom $R\,(x)$ mit
   $x^r \cdot M\,(x) = h\,(x) \cdot G\,(x) + R\,(x)$ in $Z_2\,[x]$
   und $\text{Grad}(R\,(x)) \leq r$

3. Das Codewort ist die zum Polynom
   $T\,(x) = x^r \cdot M\,(x) - R\,(x) = x^r \cdot M\,(x) + R\,(x)$
   gehörige Bitkette $\quad b_1 \ldots b_m \quad b_{m+1} \ldots b_{m+r}$

Was prüft der Empfänger:

1. Geht die Polynomdivision in $Z_2[x]$ mit $G(x)$ auf?

   Falls $\begin{cases} \text{Ja} & \hat{=} \quad \text{höchstwahrscheinlich kein Fehler} \\ \text{Nein} & \hat{=} \quad \text{Fehler!} \end{cases}$

2. Falls ja: Die Prüfsummenbits abschneiden und die empfangene Nachricht weiterreichen (an die höheren Schichten)
3. Falls nein: Drop :-(

Übungsaufgabe: Generatorpolynom wie oben. Sind folgende Nachrichten korrekt empfangen?

- 0000 0101 1111
- 11 0101 1011 10

Bemerkung:

1. Addition, Subtraktion in $Z_2 \quad \hat{=} \quad$ Exklusiv Oder
2. Übertragungsfehler:
   Empfangen wird $T(x) + E(x)$
   $(T(x) + E(x)) / G(x) = h(x) + \frac{E(x)}{G(x)}$
   $G(x) \mid E(x) \Longleftrightarrow$ Fehler wird nicht entdeckt
3. $(x+1) \mid G(x) \Longrightarrow$ Beliebige ungerade Anzahl von Bit-Fehlern wird erkannt.
4. Beispiele für Generatorpolynome:
   $$\begin{aligned} CRC-12 &= x^{12}+x^{11}+x^3+x^2+x+1 \\ CRC-16 &= x^{16}+x^{15}+x^2+1 \\ CRC-CCITT &= x^{16}+x^{12}+x^5+1 \end{aligned}$$
   enthalten alle $(x+1)$ als Primfaktor
   6-Bit Character: $CRC-12$
   8-Bit Character: $CRC-16$, $CRC-CCITT$

Definition 35: Burstfehler der Länge $j :\Longleftrightarrow$

$$\exists\, i > 0, a_i \in Z_2 : E(x) = x^i \cdot (x^{j-1} + a_{j-2} \cdot x^{j-2} + \cdots + a_1 \cdot x^1 + 1)$$

Bem.:

5) $CRC - 16, CRC - CCITT$ erkennen alle 1-Bit- und
2-Bit-Fehler,
alle Fehler mit ungerader Bitzahl
Burstfehler bis maximal Länge 16
99,997 % aller Burstfehler bis Länge 17
99,998 % aller Burstfehler bis Länge $j > 18$

6) Berechnung der Checksum in Hardware mit Shift-Register
(Peterson, Brown 1961)

Kommunikationsarten zwischen Stationen A, B:

- simplex := unidirektional $A \to B$
- halbduplex := abwechselnd $A \to B$ und $B \to A$
- (voll-)duplex := bidirektional $A \leftrightarrow B$

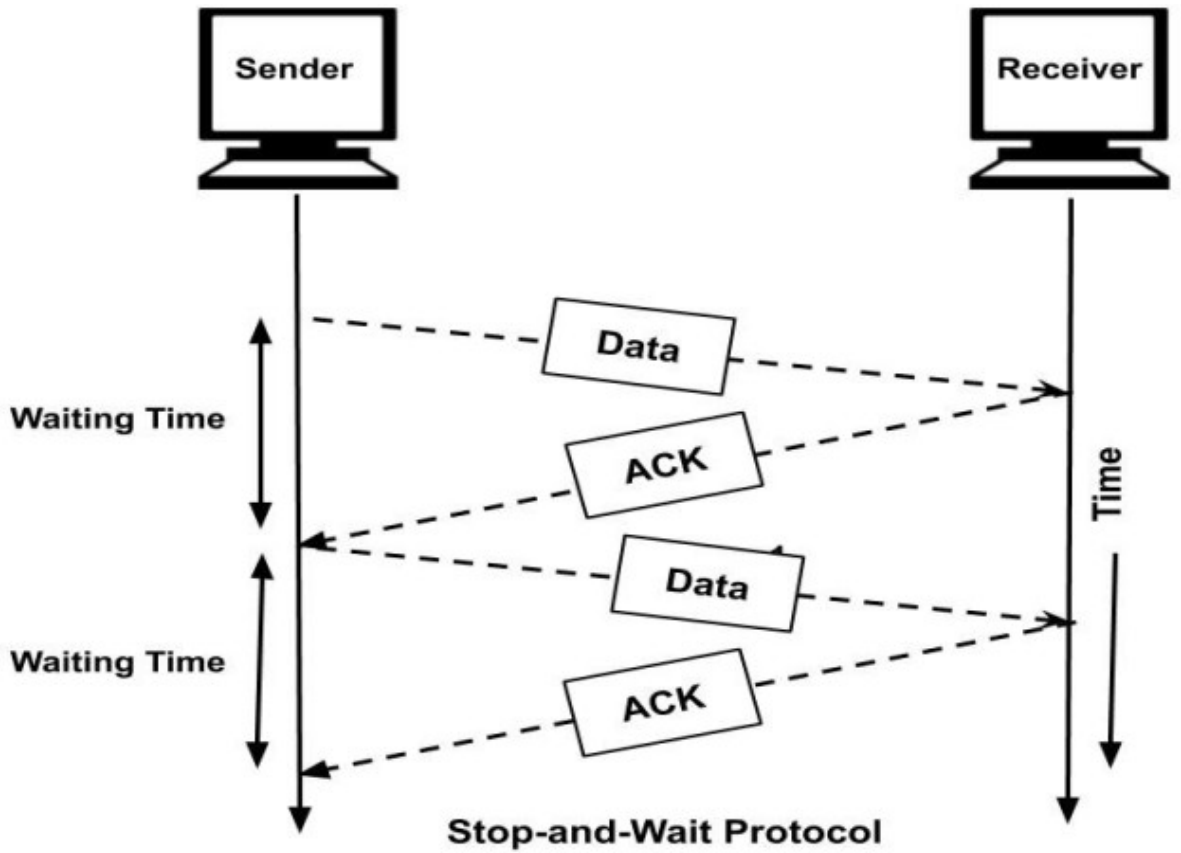
Probleme:

1. unzuverlässige Leitungen
2. Sender und Empfänger arbeiten i. d. R. nicht gleich schnell

Stop-and-Wait-Protokolle

Simplexprotokolle

Idee: dynamisches Anpassen der Übertragungsgeschwindigkeit mittels expliziter Quittungen ($\simeq$ Rahmen mit leerer Payload)



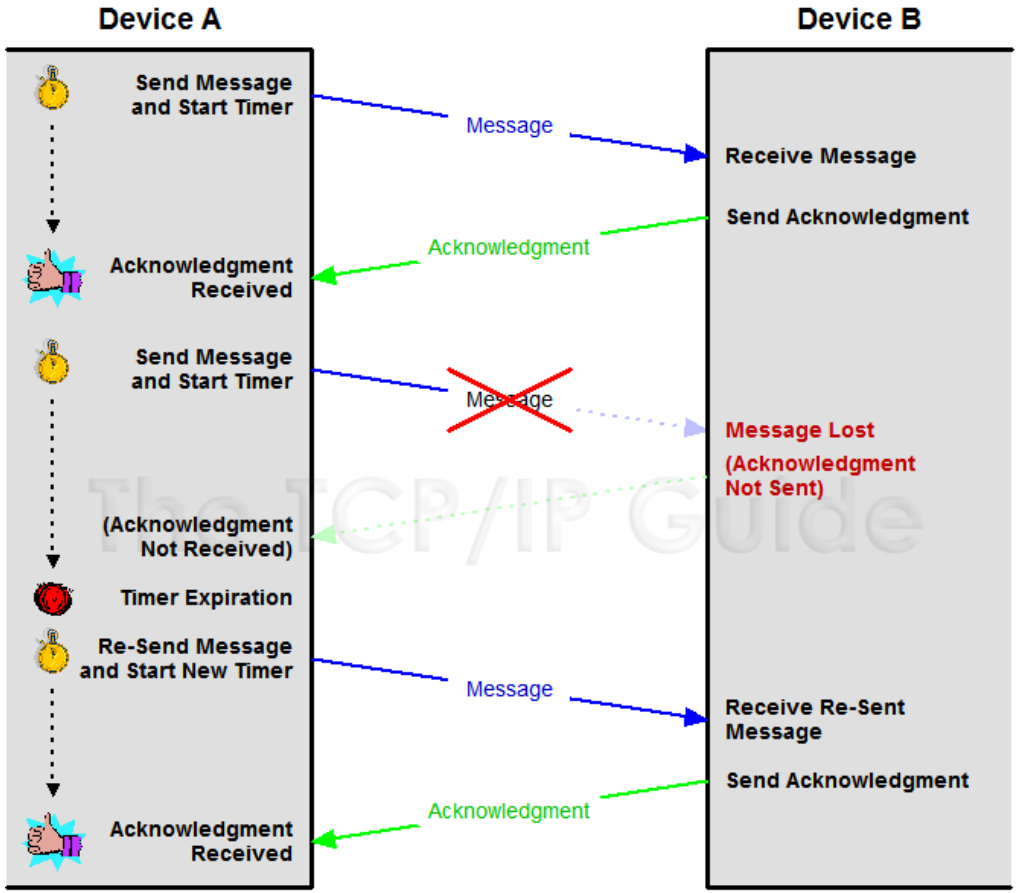
Quelle: `https://afteracademy.com/blog/what-is-stop-and-wait-protocol`

Probleme:

- Rahmen kann verloren gehen.
- Quittung kann verloren gehen.
- bisher nur Flußsteuerung, keine Fehlerkontrolle mit Retransmission für verloren gegangene Pakete.

PAR-Protokolle (Positive Acknowledgement with Retransmission)

Idee: Nach einem Timeout wird die Nachricht wiederholt.



Quelle: `http://www.tcpipguide.com/free/t_TCPSlidingWindowAcknowledgmentSystemForDataTranspo-3.htm`

Probleme:

- Was passiert, wenn die Quittung verloren geht?
  $\Longrightarrow$ Rahmenfolgenummern, um Duplikate erkennen zu können
- Was ist, wenn der Timeout-Wert zu klein eingestellt ist?

Optimierung bei Duplexprotokoll: Die Quittung wird nicht als eigener Rahmen verschickt, sondern im Header der Rückantwort notiert (sogenanntes Piggy Backing).

aber: Symmetrischer Datenverkehr wird vorausgesetzt.

Nachteil: Stop-and-Wait-Protokolle sind ineffizient bei langen Laufzeiten

Beispiel: Satellitenverbindung mit 64 Kbps,
Signalausbreitungsgeschwindigkeit: 270 ms
Rahmen 1280 bit
$\Longrightarrow$ Nach ca. 20 ms ist der Sender mit der Übertragung fertig.
$\Longrightarrow$ Nach fühestens $20 + 270 + 270 = 560$ ms ist eine Quittung eingetroffen.

stattdessen: Der Sender könnte $560 : 20 = 28$ Rahmen verschicken.

Schiebefenster-Protokolle (engl. Sliding-Window)

- n-Bit Folgenummern: $0, 1, ..., 2^n - 1$ werden modulo 2^n vergeben
- Das Sende- bzw. Empfangsfenster gibt die Nummern derjenigen Rahmen an, die aktuell verschickt bzw. empfangen werden dürfen.
- Der Sender startet für jeden abgeschickten Rahmen einen Timer.
- Der Empfänger schickt nach Empfang eines Rahmens eine Quittung.
  Quittung $Q_k = k :\Longleftrightarrow$ Der Empfänger hat alle Rahmen bis einschließlich Nummer k erhalten.

  alternativ:
  Quittung $Q_k = k :\Longleftrightarrow$ Der Empfänger hat alle Rahmen bis einschließlich Nummer $k - 1$ erhalten und erwartet jetzt Rahmennummer k.

Weiterschieben des Sendefensters:

- Erhält der Sender eine Quittung für den Rahmen k und ist k der erste Rahmen im Sendefenster, wird das Sendefenster um eine Nummer weitergeschoben.
  Falls die nun erste Rahmennummer schon bestätigt war, wird das Sendefenster erneut eine Rahmennummer weitergeschoben u.s.w.

Weiterschieben des Empfangsfensters:

- Wird ein Rahmen empfangen und ist der Rahmen der erste Rahmen im Empfangsfenster, wird das Empfangsfenster um eine Nummer weitergeschoben.
  Falls der nun erste Rahmen im Empfangsfenster schon empfangen wurde, wird das Empfangsfenster erneut eine Rahmennummer weitergeschoben. u.s.w.

Was passiert, wenn der empfangene Rahmen nicht der erste Rahmen im Empfangsfenster ist?

1. Möglichkeit: Go-Back-N

- Der Empfänger nimmt Pakete nur in Reihenfolge an:
  $|\text{Empfangsfenster}| = 1$
- Wird ein Rahmen mit falscher Rahmennummer empfangen, wiederholt der Empfänger die letzte Quittung.

2. Möglichkeit: Selective Repeat

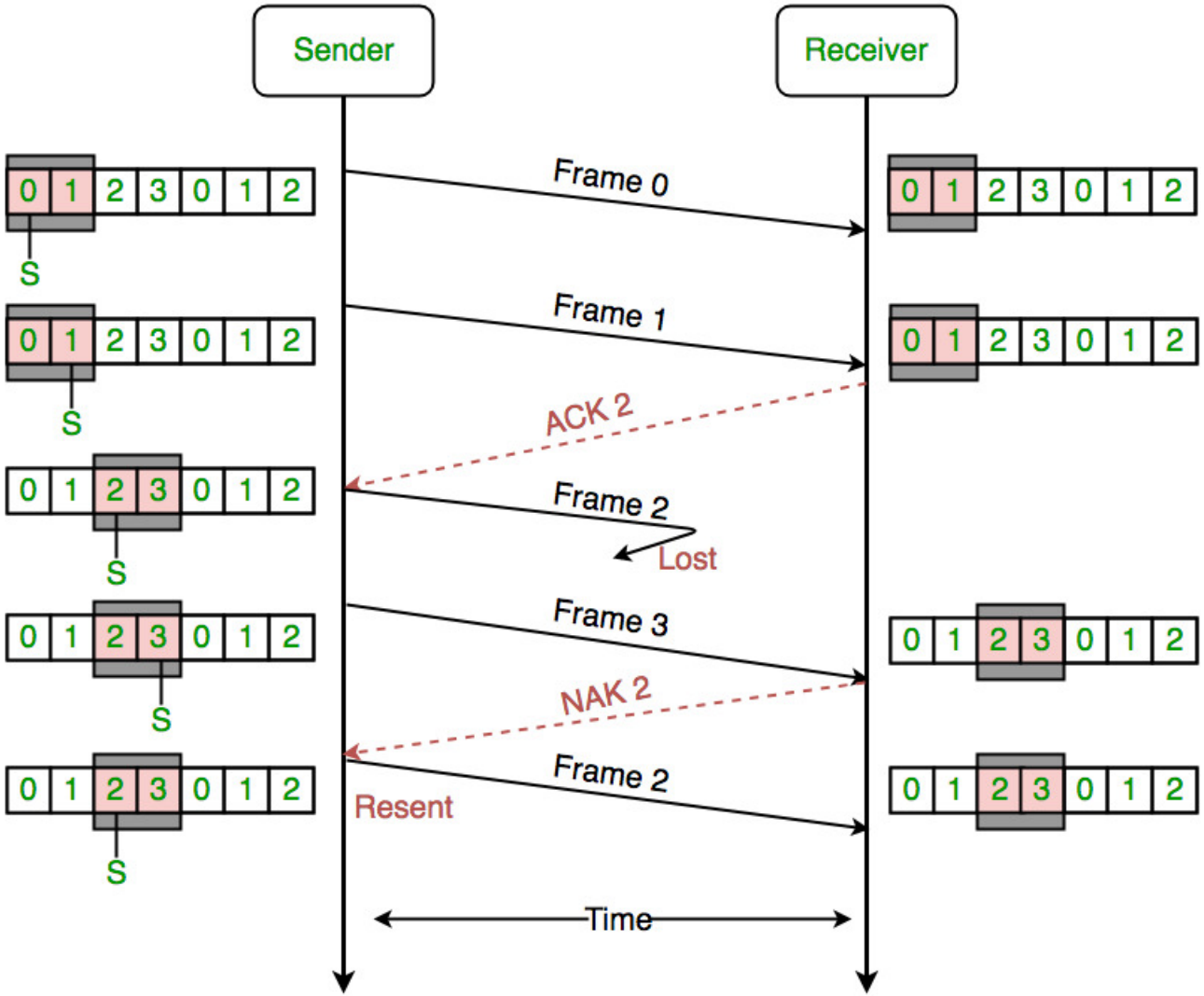
- Empfangsfenster > 1
- Im Fehlerfall: selektives Wiederholen

Bem.: Das Empfangsfenster muß klein genug sein, um Duplikate von neuen Rahmen unterscheiden zu können.

Beispiel: 3-Bit Rahmennummern: 0, 1, ..., 7
Größe vom Sendefenster = Größe vom Empfangsfenster = 5

1. Sender schickt nacheinander die Rahmen mit den Nummern 0, 1, 2, 3, 4
2. Empfänger quittiert jeden Rahmen und schiebt sein Empfangsfenster weiter auf: 5, 6, 7, 0, 1
3. gestörter Rückkanal: Die Quittungen für alle Rahmen gehen verloren.
4. Senderseite: Nacheinander gibt es einen Timeout für die Rahmen 0 bis 4 $\Longrightarrow$ Sender wiederholt die Rahmen 0 bis 4
5. Der Empfänger verwirft/ignoriert die Rahmen 2 bis 4, weil sie nicht im aktuellen Empfangsfenster liegen, nimmt aber die Rahmen 0 und 1 als „*neu*" an. :-(

$\Rightarrow$ Größe vom Sendefenster + Größe vom Empfangsfenster $<= 2^n$, damit das alte und neue Empfangsfenster nach dem Weiterschieben noch disjunkt sind. $\Longrightarrow$ Sende- und Empfangsfenster maximal 2^{n-1}



Beispiel: 2-Bit Folgenummern: 0, 1, 2 und 3. Maximale Sende- und Empfangsfenster haben Größe 2.

Quelle: `https://www.geeksforgeeks.org/sliding-window-protocol-set-3-selective-repeat/`

Übungsaufgabe:

Probieren Sie die folgende Simulation von Selective Repeat und Go-Back-N aus: https://www2.tkn.tu-berlin.de/teaching/rn/animations/gbn_sr/

- Wird der Effekt eines *zu kleinen* Timeout-Werts korrekt simuliert?
- Wann werden *data delivered to upper network layer*?
- Wann können die gelb gefärbten Pufferplätze (*transmission confirmed*) beim Sender wiederverwendet werden?
- Wird der Unterschied zwischen Selective Repeat und Go-Back-N in der Simulation sichtbar?

7.4 High-level Data Link Control (HDLC)

ISO-Protokoll
öffentliche Netze / WAN
stammt von IBM's SDLC (Synchronous Data Link Control) ab

verwandte Protokolle:

ANSI:	Advanced Data Communication Control Procedure (ADCCP)
CCITT:	Link Access Procedure-Balanced (LAP-B)

Grundlage für die Sicherungsschicht in

- Mobilfunktelefonie über GSM (2G)
- Einwahl in das Internet mittels Point-to-Point-Protocol (PPP)
- CISCO Router: benutzen HDLC Serial Interfaces, HDLC over Layer 2 Tunneling Protocol (RFC 4349, 2006),
- ...

HDLC unterscheidet Primär- und Sekundärstationen:

Primärstation := Leitstation, die den Datentransfer steuert, schickt sogenannte `command`-Steuerrahmen.
Sekundärstation := Station arbeitet unter der Leitung der Primärstation und reagiert mit sogenannten `response`-Steuerrahmen.

Mögliche Topologien:
a) Punkt-zu-Punkt-Verbindung: dedizierte Leitungen zwischen den Stationen
b) Mehrpunktverbindung: Leitung ist ein Broadcast-Kanal

HDLC kennt unterschiedliche Betriebsarten:

1) nichtbalancierte Konfiguration: In der nichtbalancierten Konfiguration werden zwei Betriebsarten unterschieden:
Normal Response Mode (NRM) := Die Primärstation initiiert den Datentransfer zur Sekundärstation. Die Sekundärstation kann Daten nur auf Anfrage der Primärstation hin senden. (Halbduplexbetrieb)
Asynchronous Response Mode (ARM) := Auch die Sekundärstation kann den Datentransfer initiieren, aber: Die Kontrolle bleibt bei der Primärstation. (Duplexbetrieb, Punkt-zu-Punkt-Verbindung)

2) balancierte Konfiguration: Kombinierte Stationen, die sowohl die Funktion einer Primärstation als auch einer Sekundärstation ausüben.
Betriebsart:
Asynchronous Balanced Mode (ABM) := Jede der beiden Stationen kann den Datentransfer initiieren. (Duplexbetrieb)

Rahmentypen:

- information frame: Datenrahmen
- unnumbered frame: Steuerrahmen für Auf- und Abbau der Verbindung

SNRM	Aufrufbetrieb aufnehmen (set normal response mode)
SARM	Konkurrenzbetrieb aufnehmen (set asynchronous response mode)
SABM	Konkurrenzbetrieb aufnehmen (set asynchronous balanced mode)
DISC	Abbruch der Verbindung (disconnect)
UA	nicht nummerierte Quittung für alle unnummerierten Steuerrahmen (unnumbered acknowledge)
CMDR	negative Bestätigung der Kommandos (command reject)

- supervisery frame: Steuerrahmen für Fluß- und Fehlerkontrolle

RR	Station empfangsbereit (receive ready)
RNR	Station nicht empfangsbereit (receive not ready)
REJ	Aufforderung an Empfänger R, die Nachricht ab N(R) einschließlich nochmals an S zu übertragen (reject)
SREJ	Aufforderung an den Empfänger R, die Nachricht mit der Nummer N(R) nochmals an S zu übertragen (selective reject)

Wieviele Headerfelder brauchen wir für die Steuerrahmen?

1 Control Byte

Bedeutung des Control-Felds

Rahmentyp	Befehle	Antwort	1 2 3 4	5	6 7 8
information	I	I	0 N(S)	P	N(R)
supervisery		RR	1 0 0 0	P/F	N(R)
supervisery		RNR	1 0 1 0	P/F	N(R)
supervisery		REJ	1 0 0 1	P/F	N(R)
supervisery		SREJ	1 0 1 1	P/F	N(R)
unnumbered	SNRM		1 1 0 0	P	0 0 1
unnumbered	SARM		1 1 1 1	P/F	0 0 0
unnumbered	SABM		1 1 1 1	P	1 0 0
unnumbered	DISC		1 1 0 0	P	0 1 0
unnumbered		UA	1 1 0 0	F	1 1 0
unnumbered		CMDR	1 1 1 0	F	0 0 1

Das fünfte Bit hat unterschiedliche Bedeutung: Poll-/Final-Bit.
Poll-Bit gesetzt $\Longrightarrow$ Sofort antworten, nicht “piggyback”.
Final-Bit gesetzt $\Longrightarrow$ kennzeichnet Quittungsrahmen.

Rahmenformat:

SD	Adresse 1 Byte	Control 1 Byte	**Daten** $\geq$ 0 Byte	Prüfsumme 2 Byte	ED

SD = ED = 0 11 11 11 0 mit Bitstopfen
Generatorpolynom für Prüfsumme: CRC-CCITT

Das Control-Feld gibt u.a. den Rahmentyp an (Datenrahmen oder Steuerrahmen (Control Frame)) und enthält Sende- und Quittungsfolgenummer.

$\Longrightarrow$ Größe des HDLC-Headers: 6 Byte

Schiebefensterprotokoll: Sendefolgenummer (modulo 8), d.h. der Sender kann maximal 7 Datenrahmen schicken ohne eine Quittung erhalten zu haben. Quittungen werden default-mäßig piggyback verschickt.
N(R): = Sendefolgenummer des zuletzt erhaltenen Rahmens +1 („erwartete Rahmen") quittiert alle Rahmen bis N(R)-1.

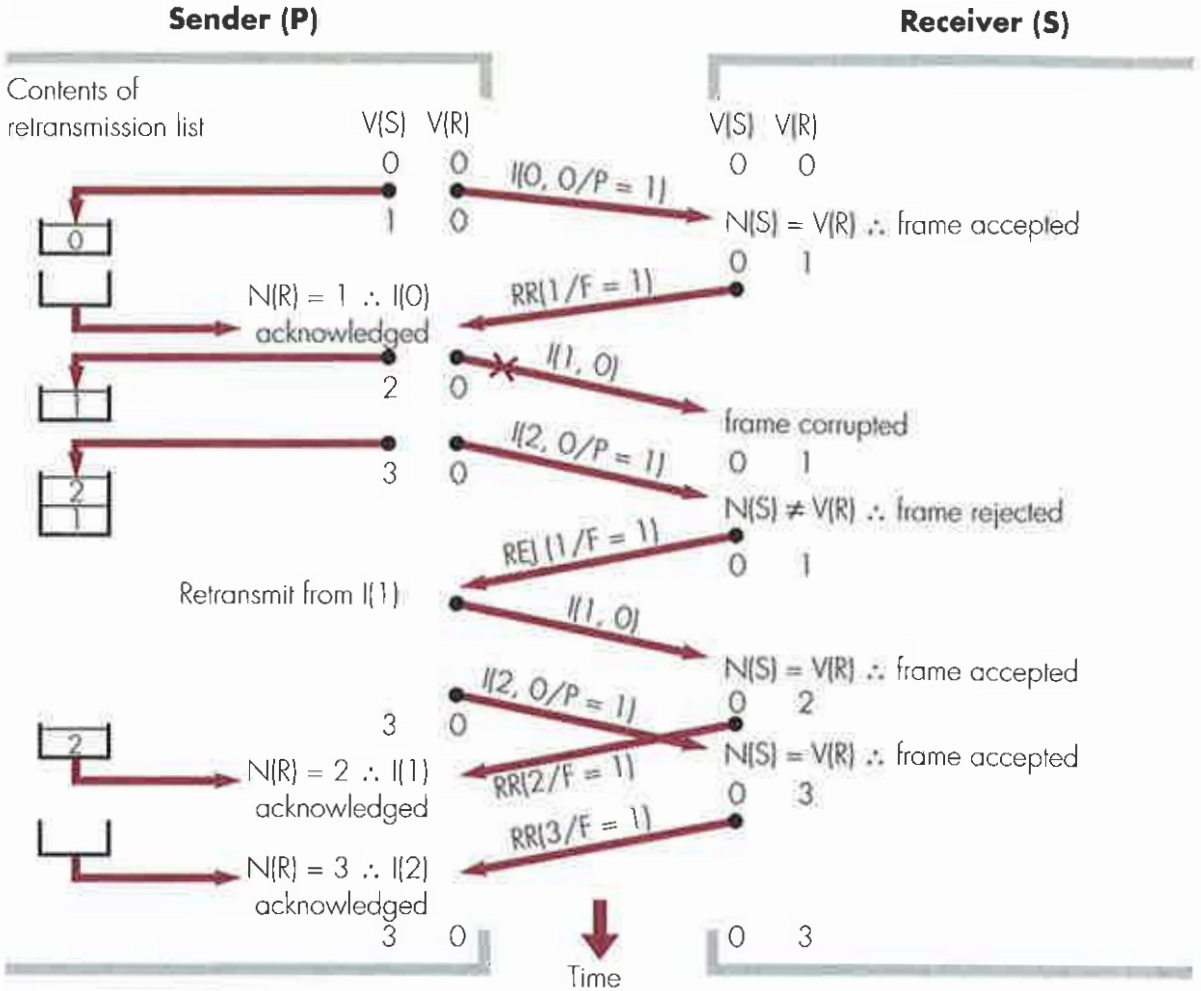


Figure 6.37 HDLC normal response mode: example frame sequence diagram with single primary and secondary (i.e. no piggyback acknowledgments).

Quelle: Fred Halsall: Multimedia Communications - Applications, Networks, Protocols and Standards

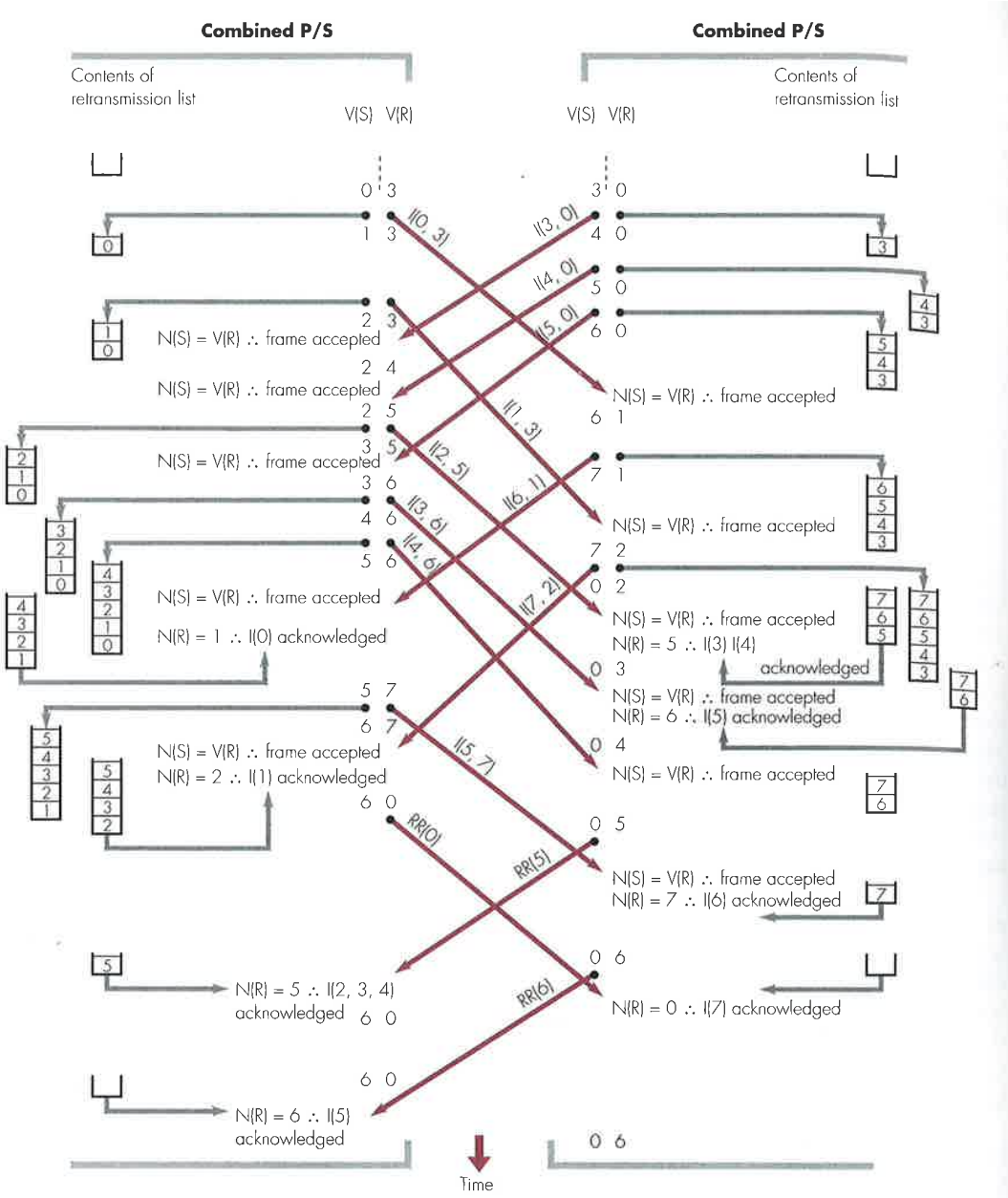
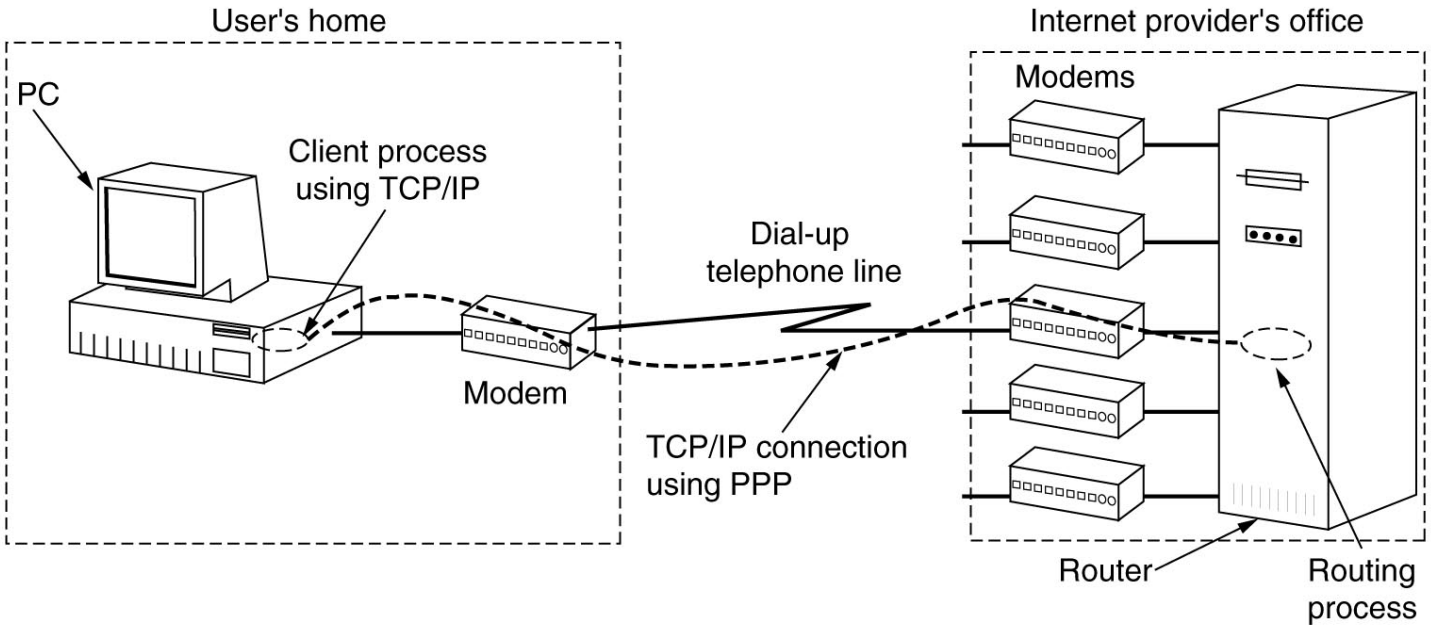


Figure 6.38 HDLC asynchronous balanced mode: piggyback acknowledgment procedure.

Quelle: Fred Halsall: Multimedia Communications - Applications, Networks, Protocols and Standards

De facto Standardprotokoll für die Einwahl in das Internet:

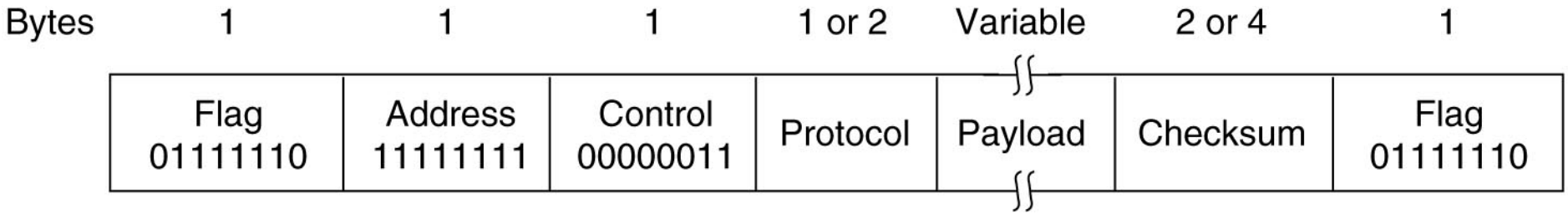


Einwahl über Modems RFC 1661, 1662, 1663: PPP über HDLC
heutzutage üblich: PPP over Ethernet (PPPoE)

HDLC-Rahmenformat:

SD	Adresse 1 Byte	Control 1 Byte	Daten $\geq$ 0 Byte	Prüfsumme 2 Byte	ED

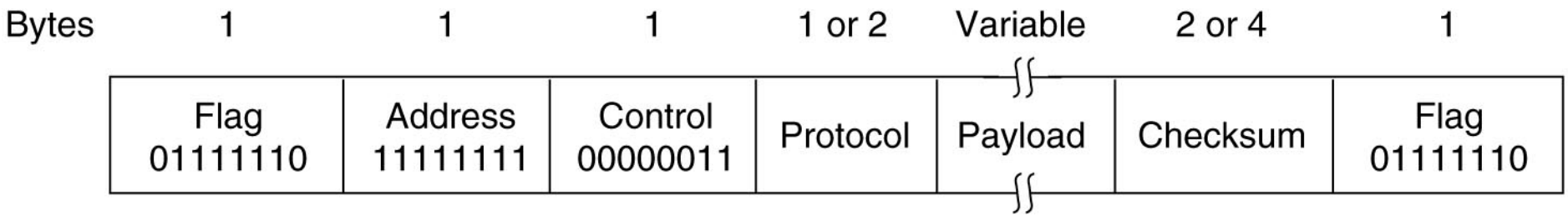
Das PPP Rahmenformat



- Es werden keine Adressen auf der Sicherungsschicht benutzt: Address = 11111111 bedeutet eine Broadcast-Adresse.
- Control = 00000011 bedeutet, daß nur *unnumbered frames* geschickt werden, d.h. es wird keine Flußkontrolle durchgeführt.
  (Erweiterungen für zuverlässigen Dienst in RFC 1663, aber selten benutzt).

HDLC-Rahmenformat:

SD	Adresse 1 Byte	Control 1 Byte	Daten $\geq$ 0 Byte	Prüfsumme 2 Byte	ED



- PPP umfaßt zwei Teilprotokolle: Link Control Protocol (LCP) und Network Control Protocol (NCP).
- Neu ist das Protocol-Feld, das angibt, welches Protokoll die *Payload* erzeugt hat. Es gibt definierte Codes für LCP, NCP, IP, IPv6, ...

Link Control Protocol (LCP)

Aufgaben:

- Aktivieren und Testen von Leitungen,
- Verhandeln von Optionen, z.B.
  - Weglassen des *Address*- und *Control*-Felds,
  - Maximallänge der *Payload* (Default ist 1500 Bytes),
  - Größe des *Protocol*- und *Checksum*-Felds
- Abbau der PPP-Verbindung

Die LCP-Rahmentypen

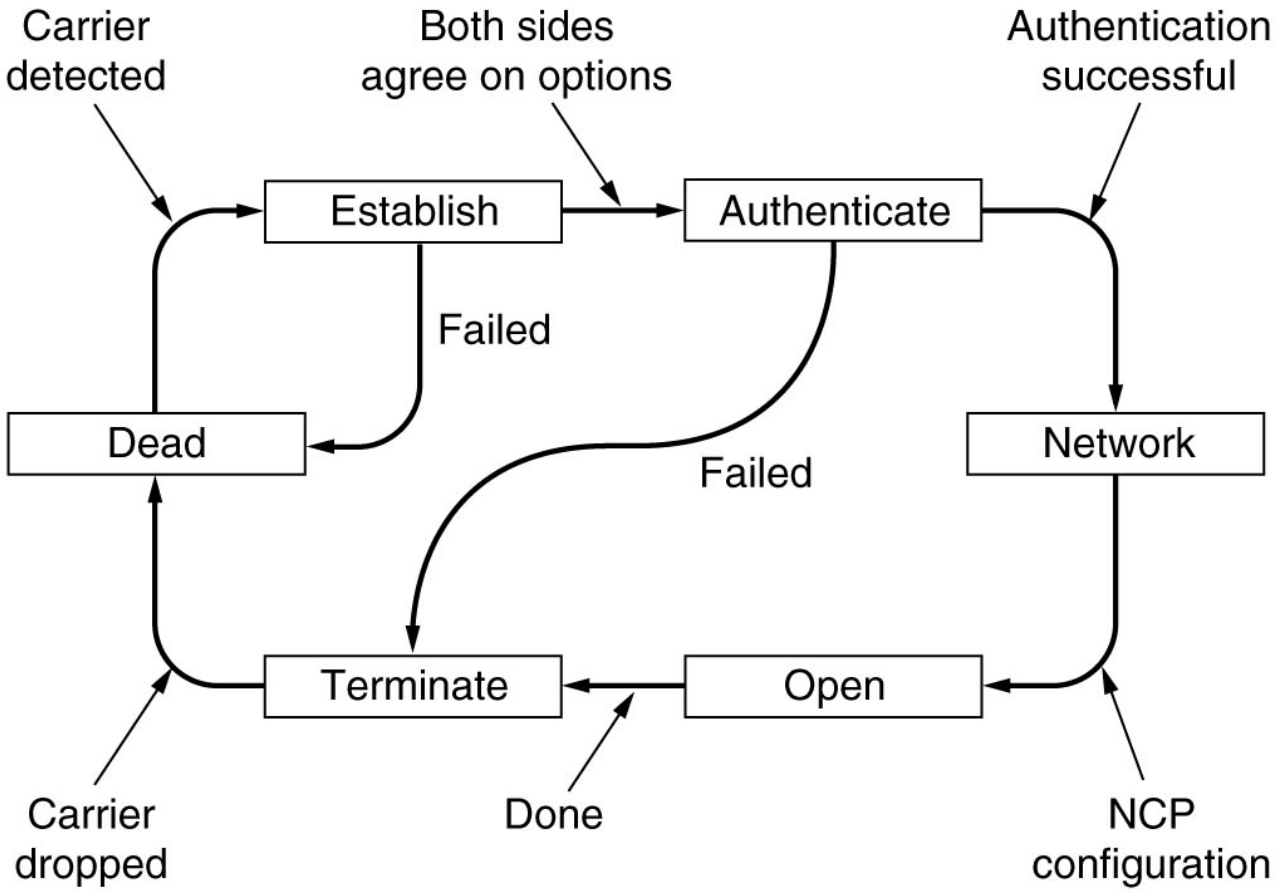
I steht für Initiator. R steht für Responder.

Name	Direction	Description
Configure-request	I → R	List of proposed options and values
Configure-ack	I ← R	All options are accepted
Configure-nak	I ← R	Some options are not accepted
Configure-reject	I ← R	Some options are not negotiable
Terminate-request	I → R	Request to shut the line down
Terminate-ack	I ← R	OK, line shut down
Code-reject	I ← R	Unknown request received
Protocol-reject	I ← R	Unknown protocol requested
Echo-request	I → R	Please send this frame back
Echo-reply	I ← R	Here is the frame back
Discard-request	I → R	Just discard this frame (for testing)

Network Control Protocol (NCP)

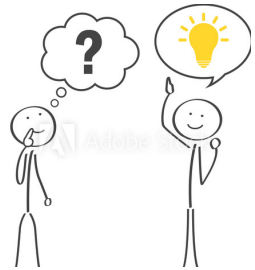
Für jedes unterstützte Protokoll der Vermittlungsschicht wird eine andere NCP-Version benötigt.
Das zu IP gehörige NCP-Protokoll ist z.B. für das dynamische Aushandeln von IP-Adressen zuständig.

Vereinfachte Darstellung der Phasen beim Auf- und Abbau von PPP-Verbindungen



Quelle: Tanenbaum

Was Sie jetzt wissen sollten ...



- Was sind die Aufgaben der Sicherungsschicht?
- Was macht das CRC-Verfahren? Wie funktioniert es?
- Was sind die Vor- und Nachteile von Stop-and-Wait-Protokollen, PAR-Protokollen und Schiebefensterprotokollen?
- Was sind die Haupzutaten von HDLC? Macht HDLC Flußkontrolle? Macht HDLC Fehlerkontrolle? Und wenn ja, wie?
- Wozu braucht man das PPP-Protokoll?