

### Inhalt:

- Aufgaben und Funktionen

-

Die **Sicherungsschicht** überträgt ganze **Rahmen** (frames) über eine Teilstrecke (link). Die Rahmen müssen fehlerfrei übertragen werden. Da Übertragungsfehler grundsätzlich nicht vermeidbar sind, müssen Möglichkeiten zur Fehlererkennung und -beseitigung bestehen.

Im Einzelnen erledigt die Sicherungsschicht die folgenden Aufgaben:

- **Rahmensynchronisation** (frame synchronisation): Anfang und Ende eines Rahmens müssen erkannt werden.
- **Flusssteuerung** (flow control): Der Sender darf den Empfänger nicht mit Rahmen überschwemmen, er muss die Empfangsbereitschaft des Empfängers berücksichtigen.
- **Fehlersicherung** (error control): Bei der Übertragung aufgetretene Fehler müssen behoben werden.
- Unterscheidung von **Nutzdaten und Steuerinformation**: Der Empfänger muss zwischen Nutzdaten und Steuerinformation unterscheiden können, da beide auf demselben Übertragungsweg übertragen werden sollen.
- **Übertragungssteuerung** (link management): Eine geordnete, fehlerfreie Übertragung muss durch geeignete Mechanismen für Koordination und Kooperation sichergestellt werden.

**Vielfachzugriffsverfahren** gehören ebenfalls zur Sicherungsschicht.

Wenn man die Funktionen der Bitübertragungsschicht hinzunimmt, lässt sich der gesamte Aufbau einer Teilstrecke in einer Übersicht darstellen. In einzelnen Anwendungen sind nicht alle Funktionsblöcke erforderlich. Dies betrifft z. B. die **Verwürfelung** (scrambling, bezweckt das Aufbrechen von langen Null- und Eins-Folgen zur spektralen Formung des Leitungssignals) und die Modulation (falls keine Breitbandübertragung verlangt wird).

Die Schicht 2 realisiert eine fehlerfreie Punkt-zu-Punkt-Übertragung ganzer Rahmen zwischen benachbarten Stationen. Dabei können entweder zwei Stationen **direkt** miteinander oder mehrere Stationen über ein **Bussystem** verbunden sein. Das Bussystem wirkt als **Broadcastnetz**, d. h. jede Station kann das Signal jeder anderen Station direkt, ohne das Durchlaufen von zwischengeschalteten Systemen empfangen.

### 3) Protokollmechanismen zur Längen Anpassung

**Segmentierung/Reassemblierung**: Die zulässige Paketlänge ist aus verschiedenen Gründen begrenzt. Falls die Nutzdaten länger sind als die auf einer Teilstrecke zulässige Länge, müssen sie segmentiert, also aufgespalten werden. Beim Empfänger muss die Segmentierung rückgängig gemacht werden (Reassemblierung), bevor die Nutzdaten dem Anwendungsprozess übergeben werden.

#### Uni- und bidirektionale Übertragung

Eine unidirektionale Übertragung von A nach B lässt in der Gegenrichtung (B nach A) keine Kommunikation zu. Fast immer ist eine bidirektionale Übertragung gewünscht, bei der beide Übertragungsrichtungen nutzbar sind.

Eine Teilstrecke kann um- oder bidirektional genutzt werden. Der Begriff **Simplex** bezeichnet eine unidirektionale Übertragung, **Duplex** eine bidirektionale Übertragung. Im **Halbduplex-Betrieb** werden die beiden Übertragungsrichtungen abwechselnd genutzt, im **Vollduplex-Betrieb** gleichzeitig.

Bei einer bidirektionalen Übertragung werden in der Gegenrichtung nicht nur Nutzdaten übertragen, sondern auch **Quittungen** (acknowledgment). Eine Quittung kann auch einem Nutzdatenpaket mitgegeben werden, diese wird als Piggyback-Quittung bezeichnet.

#### Rahmenbildung und Rahmensynchronisation

Die Bildung von Rahmen soll sicherstellen, dass die übertragenen Signale einer Kommunikationsbeziehung zugeordnet und korrekt interpretiert werden können. Die **Rahmensynchronisation** ermöglicht die eindeutige Erkennung von Rahmenanfang und -ende. Dafür gibt es drei Möglichkeiten:

Rahmenanfang und -ende werden durch eine festgelegte **Markierung** (flag) gekennzeichnet, z. B. bei HDLC.

Der Rahmenanfang wird markiert, die **Rahmenlänge** wird explizit im Header angegeben, z. B. bei TCP. Die Rahmenlänge ist **konstant** und fest vereinbart. Damit muss nur der Rahmenanfang markiert werden, z.B. bei ATM. Die Rahmensynchronisation setzt eine zuverlässige **Bitsynchronisation** (Taktregenerierung) voraus.

### Asynchrone Übertragung

Bei der asynchronen Übertragung werden einzelne Zeichen übertragen. Jedes Zeichen wird als Rahmen dargestellt, der ein **Startbit**,  $n$  Datenbits und ein (oder mehr) **Stopbits** enthält. Die Taktgeber im Sender und Empfänger liefern nominal dieselbe Frequenz, sind aber unabhängig voneinander. Dadurch ist eine (geringe) Frequenzdifferenz unvermeidlich. Für die Übertragungsdauer eines Zeichens wird durch das Startbit bzw. durch dessen Startflanke eine hinreichende Synchronisation hergestellt. Damit kann der Empfänger die zeitliche Lage der empfangenen Bits bestimmen und das empfangene Signal in der Bitmitte abtasten.

Zwischen zwei Zeichen liegt eine Pause, deren Dauer mindestens die Länge des Stopbits beträgt. Bei der asynchronen Übertragung wird ein bestimmter Zeichencode (häufig der ASCII-Code) zugrunde gelegt, der (auf dem Bildschirm und Drucker) darstellbare Zeichen und Steuerzeichen enthält. Damit ist die Übertragung nicht transparent, d.h., es können nicht beliebige Bitkombinationen übertragen werden. Zur Interpretation eines Steuerzeichens als normales Zeichen kann das Escape-Zeichen vorangestellt werden.

Die asynchrone Übertragung ist einfach, die Zahl der pro Zeichen übertragbaren Nutzbits in der Praxis maximal 8 und der Zusatzaufwand zur Übertragung von Start- und Stopbits ist hoch.

### Synchrone Übertragung

Im Gegensatz zur asynchronen Übertragung werden die Taktgeber von Sender und Empfänger aufeinander **synchronisiert**. Dies erfolgt zu Beginn eines Rahmens durch **Synchronisationsbits**. Die Synchronisation wird während der Übertragung aufrechterhalten. Da man den Takt des Senders in der Regel nicht auf einer eigenen Leitung übertragen möchte, muss der Leitungscode genügend Taktinformation beinhalten.

Die synchrone Übertragung kann bitorientiert oder zeichenorientiert durchgeführt:

Bei der **bitorientierten** Variante wird ein festes Bitmuster (häufig 01111110) als **Rahmenbegrenzung** (flag) gewählt. Als Nutzdaten sollen beliebige Bitmuster mit beliebiger Länge übertragen werden. Damit eine Folge von sechs Einsen in den Nutzdaten nicht als Rahmenbegrenzung interpretiert wird, wird **Bitstopfen** (bit stuffing) angewendet. Dabei wird nach jeweils fünf Einsen vom Sender eine Null eingefügt und vom Empfänger wieder entfernt. Somit kann die Bitfolge für die Rahmenbegrenzung in den Nutzdaten nicht auftreten.

Bei der **zeichenorientierten** Variante werden - wie bei der asynchronen Übertragung - Zeichen eines bestimmten Codes übertragen, allerdings ohne Start- und Stopbits. Escape-Zeichen werden wie bei der asynchronen Übertragung verwendet.