

## Kommunikation und Datenhaltung Sommersemester 2009

### 2. K-Übungsblatt

#### Aufgabe 1: HDLC-Protokoll und Sicherungsschicht

- (a) Nennen Sie drei Aufgaben die die Sicherungsschicht erfüllen soll und geben Sie beispielhafte Realisierungen dafür an.

**Fehlererkennung:** CRC-Verfahren zur Erkennung von Bitfehlern

**Fehlerbehebung:** Quittungen und Sendewiederholungen; Einsatz eines ARQ-Verfahrens; Flusskontrolle

**Strukturierung des Datenstroms:** Sequenznummern zur Erhaltung der Übertragungsreihenfolge

**Medienzugangskontrolle bei geteilten Medien:** CSMA/CD

- (b) In welche Phasen untergliedert sich der Protokollablauf beim Datentransfer bei HDLC?

Der Protokollablauf unterteilt sich in drei Phasen: Verbindungsaufbau, Datentransfer und Verbindungsabbau.

Folgende Aufgabenteile beziehen sich auf die in der Vorlesung vorgestellten HDLC-Dateneinheiten mit folgendem Aufbau:

Flag 01111110	Adressfeld	Steuerfeld	Daten	Prüfsumme	Flag 01111110
------------------	------------	------------	-------	-----------	------------------

Abbildung 1: Format von HDLC-Dateneinheiten

- (c) Wie wird verhindert, dass innerhalb der Nutzdaten fälschlicherweise die Flag-Bitfolge übertragen wird?

Dies wird durch das sogenannte *Bitstopfen* verhindert. Bei diesem Verfahren wird **innerhalb der Nutzdaten** nach fünf aufeinander folgenden Einsen vom Sender stets eine Null eingefügt. Der Empfänger kann dies entsprechend rückgängig machen, indem er nach fünf Einsen die eingefügte Null wieder entfernt, bevor er die Nutzdaten weiterverarbeitet.

- (d) Der Empfänger erhält über das Übertragungsmedium folgende Bitfolge innerhalb der Nutzdaten:

110101111101011111001011111011011111110110

Welche Nutzdaten wurden vom Sender verschickt und auf welches Problem stößt der Empfänger?

Der Empfänger entfernt wie weiter oben beschrieben die Null nach fünf aufeinander folgenden Einsen und rekonstruiert hierdurch die vom Sender verschickten Nutzdaten wie folgt:

1101011111101111101011111101111111...

Sobald der Empfänger auf die sieben aufeinander folgenden Einsen im hinteren Teil der empfangenen Daten stößt, kann er von einem Übertragungsfehler ausgehen, da sieben aufeinander folgende Einsen nicht vorkommen dürfen. Innerhalb der Nutzdaten wäre nach fünf Einsen eine Null eingefügt worden und das Flag, welches den Rahmen begrenzt, enthält nur sechs Einsen.

- (e) Wie nennt man die Eigenschaft, die durch das Verfahren des vorangegangenen Aufgabenteils erzielt wird?

Die erreichte Eigenschaft heißt *Codeparenz* und bedeutet, dass es für die Anwendungen vollkommen transparent bleibt, welche Mechanismen zur Übertragung noch verwendet werden. Die Anwendung auf Empfängerseite erhält genau die Daten, die von der Anwendung auf Senderseite verschickt wurden, auch wenn die tatsächlich übertragenen Daten aufgrund des Bitstopfens eventuell anders aussehen.

- (f) HDLC kann im *Asynchronous Balanced Mode* das sog. *Piggyback Acknowledgement* Verfahren anwenden. Was versteht man unter diesem Bestätigungsmechanismus?

Das Piggyback-Verfahren wird in verschiedenen Schichten genutzt, um Kontroll-Overhead zu reduzieren. Die wesentliche Idee hinter dem Verfahren besteht darin, Acknowledgement-Informationen in einer Dateneinheit mit zu senden, die selbst Daten enthält, anstelle eine separate ACK-Dateneinheit zu erstellen.

## Aufgabe 2: Zustandsübergangsdiagramme

- (a) Beschreiben Sie das Verhalten einer generischen Verbindungsaufbau und -abbauphase (2-Wege-Handshake) mittels eines Zustandsübergangsdiagramms. Benutzen Sie hierzu die Schnittstellenereignisse *ConReq*, *ConInd*, *ConRsp*, *ConCnf*, *DisReq* sowie *DisInd*.

Ein generisches Zustandsübergangsdiagramm ohne Datentransfer ist in [Abbildung 2](#) angegeben.

- (b) Erweitern Sie dieses Diagramm in zwei neue Diagramme dahingehend, dass zum einen ein unbestätigter Datentransfer und zum anderen ein bestätigter Datentransfer möglich sind. Die Verbindung soll hierbei beim bestätigten Datentransfer auch beim Ausbleiben der Bestätigung abgebaut werden können. Verwenden Sie die zusätzlichen Schnittstellenereignisse *DatReq*, *DatInd*, *DatRsp* sowie *DatCnf*.

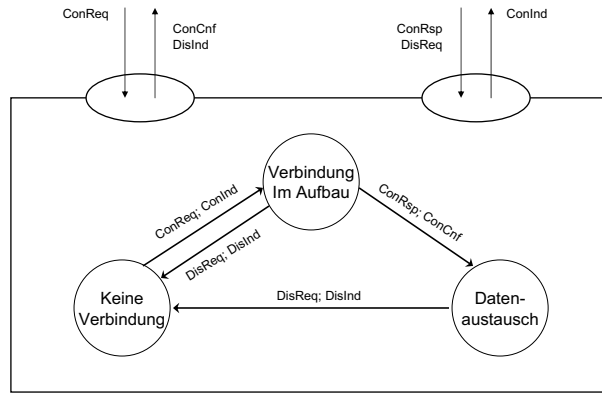


Abbildung 2: Zustandsübergangsdiagramm ohne Datentransfer

Die beiden Zustandsübergangsdiagramme für einen unbestätigten und einen bestätigten Datentransfer sind in den Abbildungen 3 und 4 angegeben.

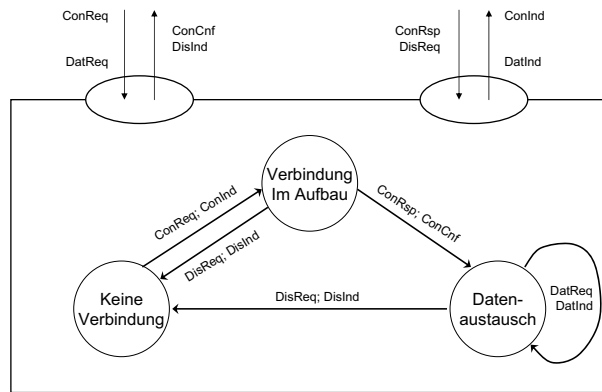


Abbildung 3: Zustandsübergangsdiagramm mit unbestätigtem Datentransfer

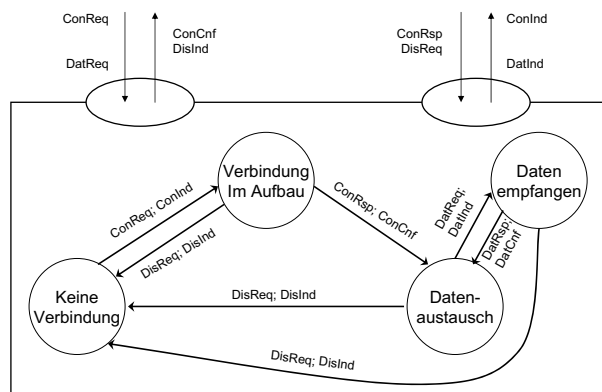


Abbildung 4: Zustandsübergangsdiagramm mit bestätigtem Datentransfer

### Aufgabe 3: Ablauffestlegungen

Instanzen erzeugen nach dem Eintreffen eines Dienstprimitivs eine PDU, die sie als Parameter eines Dienstprimitivs der darunter liegenden Schicht weiterreichen. Analog nimmt eine Instanz eine ankommende PDU von der unteren Schicht entgegen und erzeugt ein entsprechendes Dienstprimitiv.

- (a) Skizzieren Sie kurz (anhand eines Modells) die Unterscheidung zwischen Dienstspezifikation und Protokollspezifikation.

Die *Dienstspezifikation* (Abbildung 5, links) beschreibt das an der Dienstschnittstelle von außen betrachtete Verhalten eines Dienstes.

Die *Protokollspezifikation* (Abbildung 5, rechts) legt durch Regeln und Formate das Kommunikationsverhalten zweier (Protokoll-)Instanzen fest, um einen speziellen Dienst zu erbringen.

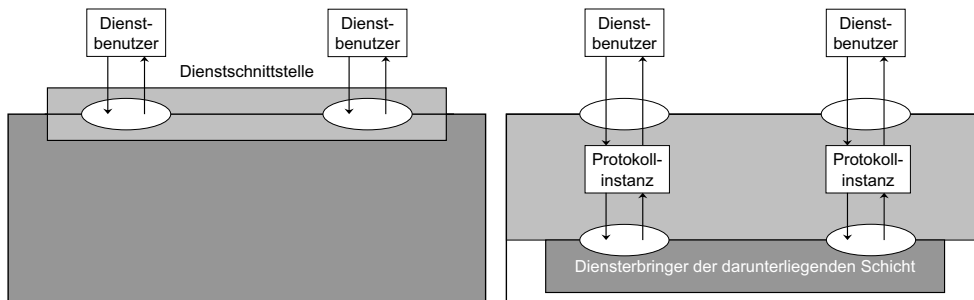


Abbildung 5: Dienstspezifikation und Protokollspezifikation

- (b) Entwickeln Sie beide Spezifikationen am Beispiel des bestätigten Transportverbindungsaufbaudienstes in Form von Zustandsübergangdiagrammen, wobei der Transportinstanz noch keine Netzwerkverbindung zur Verfügung steht. Der zugrundeliegende N-Dienst soll zuverlässig sein. In Tabelle 1 sind die Abbildungsregeln zwischen Transportdienstprimitiv und TPDU gezeigt. Die Vermittlungsschicht bietet der Transportdienstinstanz einen bestätigten Verbindungsaufbaudienst (*NCon*) und einen unbestätigten Datentransferdienst (*NDat*) an.

Transportdienstprimitiv	TPDU
TConReq/Ind	RQ
TConRsp/Cnf	RE
TDisReq/Ind	DC

Tabelle 1: Abbildung zwischen Dienstprimitiv und PDU

Die Dienstspezifikation ist in Abbildung 6 dargestellt. Man beachte, dass die Verfügbarkeit einer Netzwerkverbindung an der Transportdienstschnittstelle nicht erkennbar ist.

Die Protokollspezifikation wird in Abbildung 7 wiedergegeben. Man beachte, dass der Automat die bestehende Schicht-3-Verbindung im Falle eines fehlgeschlagenen Aufbaus der Schicht-4-Verbindung weiter bestehen lässt.

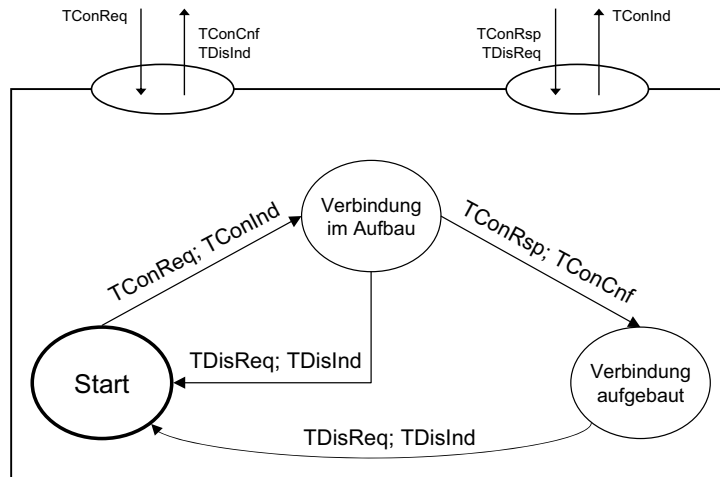


Abbildung 6: Dienstspezifikation

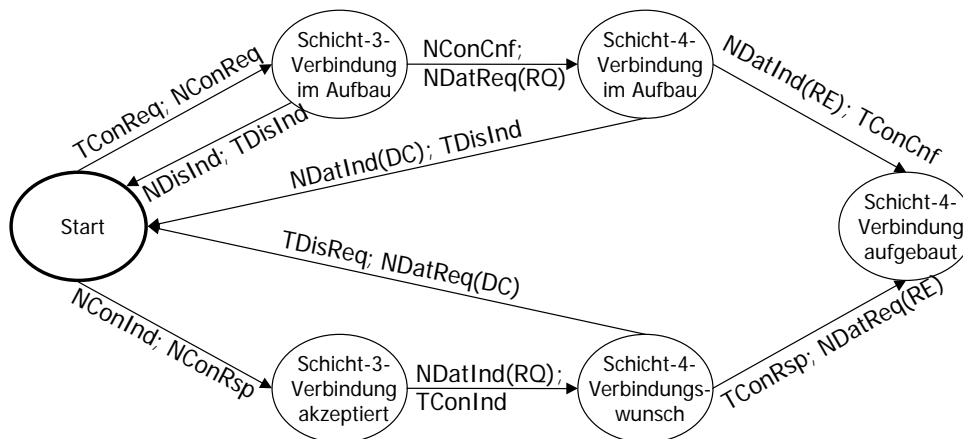


Abbildung 7: Protokollspezifikation

- (c) Betrachten Sie nun einen unzuverlässigen N-Dienst, bei dem jedoch keine Phantomnachrichten, Nachrichtenverfälschungen und Reihenfolgevertauschungen auftreten können. Welches Problem kann bei diesem N-Dienst auftreten und mit welchen Mechanismen kann man diesem Problem begegnen?

Ist der N-Dienst unzuverlässig, können Dateneinheiten verlorengehen. Dies kann beispielsweise den Verlust der RE-TPDU bewirken, so dass der gerufene Dienstbenutzer die Verbindung als aufgebaut ansieht, wohingegen der rufende Dienstbenutzer immer noch auf die Akzeptanz bzw. Ablehnung der Verbindung wartet. Dieses Problem wird umgangen, indem Timer eingesetzt werden, nach deren Ablauf der letzte Datentransfer wiederholt wird.

Dabei kann es natürlich zur Verdopplung von TPDU kommen, die durch spezielle Kennungen vom Empfänger erkannt werden. Um sicherzustellen, dass ein Verbindungsaufbauversuch nur dann als akzeptiert angesehen wird, wenn beide Dienstbenutzer davon unterrichtet sind, kann zudem ein Dreizeige-Handshake eingesetzt werden.

## Aufgabe 4: Lokale Netze

- (a) Die Sicherungsschicht wird üblicherweise weiter in zwei Unterbereiche unterteilt. Nennen Sie die Bezeichnungen dieser beiden Teilschichten und deren Aufgaben.

Die Sicherungsschicht kann man logisch in zwei weitere Teilschichten separieren, namentlich in *Medium Access Control* (MAC, Schicht 2a) und *Logical Link Control* (LLC, Schicht 2b). Während sich die MAC-Schicht um die Zugangskontrolle für ein geteiltes Medium kümmert, erfüllt die LLC-Schicht Protokollfunktionen, die der Sicherung der Punkt-zu-Punkt-Verbindung dienen, beispielsweise gegen Verfälschung, Verlust oder Reihenfolgevertauschung, sowie Flusskontrollmechanismen und eine Strukturierung der Übertragung.

- (b) Welche Medienzuteilungsverfahren kommen jeweils bei Ethernet und Token-Ring zum Einsatz?

Bei beiden kommt Zeitmultiplex mit asynchronem Zugriff zum Einsatz. Bei Ethernet handelt es sich um einen konkurrierenden Zugriff mittels CSMA/CD, während Token-Ring einen kontrollierten Zugriff anhand eines zirkulierenden Senderechts nutzt.

- (c) Welche Aufgabe hat das Padding-Feld (PAD) einer Ethernet-Dateneinheit in Bezug auf den Medienzugriff bei Ethernet?

Das Padding-Feld dient dazu, die minimale Länge einer Dateneinheit für die Funktionsfähigkeit von CSMA/CD bei gegebener maximaler Segmentlänge sicherzustellen. Es muss bei CSMA/CD gewährleistet sein, dass das Senden der Dateneinheit nach der Signallaufzeit durch das Medium und zurück noch nicht beendet ist um etwaige Kollisionen zu identifizieren. Hierfür ist bei gegebener maximaler Länge des Segments eine minimale Länge der Dateneinheit notwendig. Falls zu wenig Nutzdaten für das Erreichen dieser Untergrenze versandt wurden, wird die Dateneinheit über das Padding-Feld auf die minimale Länge aufgefüllt.

- (d) Wie werden bei Token-Ring Quittungen für Nachrichten erzeugt?

Da der Empfänger beim Token-Ring aktiv an das Medium angeschlossen ist, kann er einzelne Bits ändern, bevor er die Dateneinheit wieder auf den Ring gibt. Zudem empfängt der Sender einer Dateneinheit diese als letzte Station wieder, da er sie vom Ring nehmen muss. Aufgrund dieser Tatsachen kann die Quittung »on the fly« erfolgen, d. h. der Empfänger setzt zur Quittierung ein bestimmtes Bit in der Dateneinheit, welches vom Sender erkannt wird, während er die Dateneinheit vom Ring nimmt.

- (e) Welcher der beiden Netz-Typen ist prinzipiell für den Realzeitbetrieb geeignet?

Realzeitanwendungen erfordern Zeitgarantien, d. h. Garantien, dass eine bestimmte Aktion nach einer gewissen Zeitspanne nach der Anforderung abgeschlossen ist. Ethernet ist nicht für Realzeitanwendungen geeignet, da noch nicht einmal garantiert werden kann, ob eine Dateneinheit überhaupt erfolgreich abgeschickt werden kann. Theoretisch ist es möglich, dass es bei jedem Sendeversuch zu einer Kollision kommt, eine Dateneinheit also nie erfolgreich gesendet wird. Je mehr Stationen

an einem Ethernet angeschlossen sind, desto schwieriger wird es, Aussagen über die Leistungsfähigkeit zu treffen. Beim Token-Ring hingegen wird das Senderecht zyklisch weitergereicht. Geht man zudem davon aus, dass jede Station nur eine gewisse Datenmenge senden darf (vorgegeben durch die maximale *Token Holding Time*, THT), bevor sie das Token weitergeben muss und vernachlässigt man einen eventuellen Prioritätsmechanismus, so kann man die maximale Wartezeit angeben, die zwischen Sendewunsch und Anfang des Sendens vergeht:

$$\text{Wartezeit}_{max} = (n - 1) \cdot \text{Sendezeit}(\text{Datenmenge}_{max}) + \text{Tokenumlaufzeit}$$

(Alternativ natürlich auch Berechnung mittels THT.)

Hierbei bezeichne  $n$  die Anzahl der Stationen am Ring.

- (f) Worin liegt der grundlegende Unterschied der beiden Typen beim Anschluss an das Medium?

Beim Token-Ring werden die einzelnen Stationen aktiv an den Ring angeschlossen, d. h. jede Station empfängt eine sich auf dem Ring befindliche Dateneinheit und schickt sie regeneriert an den Nachbarn weiter. Dadurch kann eine größere Ringlänge erreicht werden als bei Ethernet, wo die Stationen nur passiv am Medium angeschlossen sind. Eine Regeneration und somit Verstärkung des Signals erfolgt dort also nicht. Ein Nachteil des aktiven Anschlusses ist allerdings, dass es durch die Regenerierung der Daten zu einer Verzögerung kommt. Beim Token-Ring spricht man üblicherweise von einem »1-Bit-Delay«. Außerdem ist der aktive Anschluss einer Station an ein Medium aufwendiger, da dafür das Medium unterbrochen werden muss, so dass es kurzfristig nicht funktionsfähig ist.

- (g) Vergleichen Sie Ethernet und Token-Ring bezüglich ihres Verhaltens (Vor- und Nachteile) bei niedriger, sowie bei hoher Auslastung.

Unter geringer Auslastung hat Ethernet Vorteile gegenüber Token-Ring. Da wenig Verkehr auf dem gemeinsamen Medium ist, liegt die Chance sehr hoch, dass ein Sender seine Dateneinheit schnell und ohne Kollision versenden kann → geringe Latenz (Wartezeit) in diesem Fall, während der Sender bei Token-Ring erst warten muss, bis er das Token belegen kann.

Bei hoher Auslastung, besteht bei Ethernet vermehrt das Problem, dass ein Sender warten muss bis das Medium frei ist und mit größerer Wahrscheinlichkeit Kollisionen auftreten (da ja viele Sender gleichzeitig sendewillig sind), mit der Folge dass Dateneinheiten wiederholt werden müssen. Beides senkt die Effizienz deutlich. Demgegenüber kann Token-Ring aufgrund des zirkulierenden Senderechts (vermeidet Kollisionen) und der Vorschrift, dass jeder Sender das Token nur für eine gewisse Zeit (Token Holding Time) belegen darf, auch bei hoher Auslastung eine hohe Effizienz vorweisen.