

Kommunikation und Datenhaltung

6. Beschreibungsmethoden für Dienste und Protokolle



Prof. Dr. Martina Zitterbart
Dipl.-Inform. Martin Röhrich
[zit | roehricht]@tm.uka.de



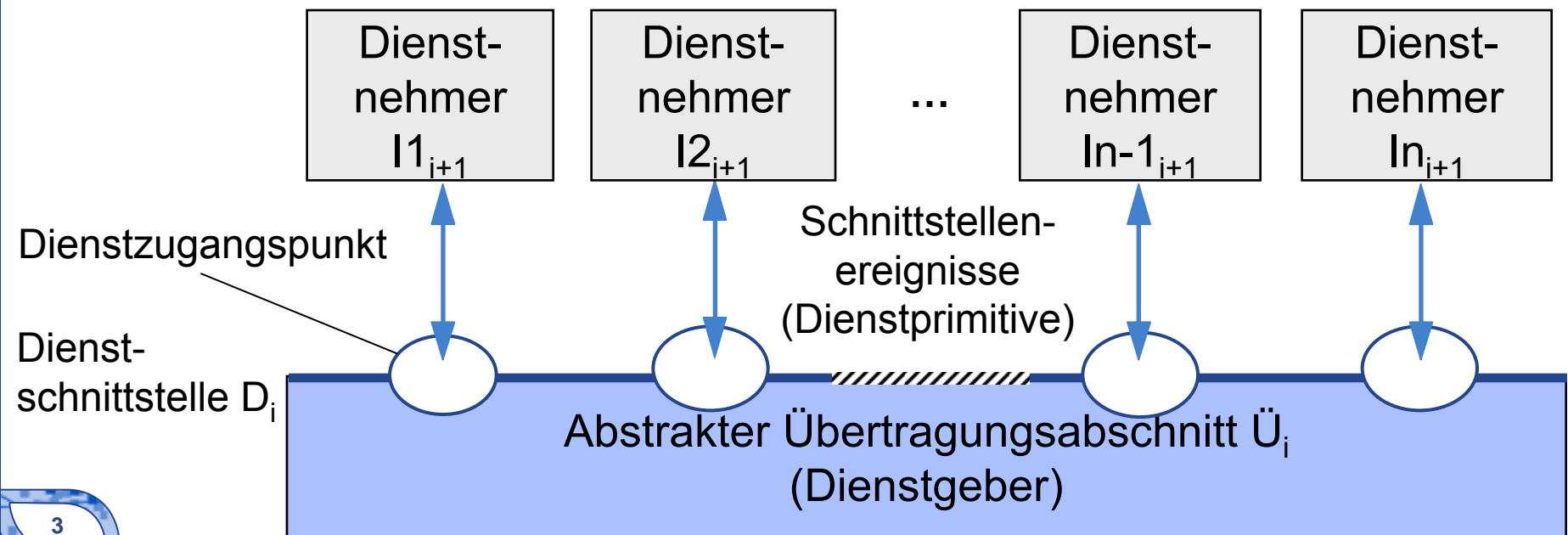
1. Einführung
 2. Physikalische Grundlagen
 3. Protokollmechanismen
 4. Geschichtete Architekturen
 5. Sicherungsschicht: HDLC
 6. Beschreibungsmethoden
 7. Sicherungsschicht:
Lokale Netze
 8. Netzkopplung und Vermittlung
 9. Die Transportschicht
 10. Anwendungssysteme
 11. Middleware
- 6.1 Dienste
 - 6.2 Weg-Zeit-Diagramme
 - 6.3 Zustandsübergangdiagramme
 - 6.4 Protokollfunktionalität

- ... wo stehen wir in der Vorlesung?
 - ... eine Reihe von Bausteinen für die Kommunikation aus Kapitel 3
 - ... grundlegende Architekturbetrachtungen aus Kapitel 4
 - ... ein Beispielprotokoll (HDLC) aus Kapitel 5
 - ▶ Informell beschrieben
 - ▶ Formale Beschreibung eines Protokolls?

- Ziel dieses Kapitels
 - Techniken zur formalen Beschreibung von Diensten und Protokollen
 - ▶ ... bilden Basis für Verifikation

6.1 Dienste: Dienstnehmer / Dienstgeber

- Dienstnehmer und Dienstgeber
 - Dienstprimitive zwischen Dienstnehmer und Dienstgeber
 - ▶ Entspricht vertikaler Kommunikation zwischen Schichten
 - Inanspruchnahme von Diensten über Dienstzugangspunkte
- Beschreibung eines Dienstes
 - Verhalten an der Dienstschnittstelle
 - Abstrakter Übertragungsabschnitt bildet eine elementare Instanz



- Dienstschnittstelle
 - Menge räumlich verteilter Dienstzugangspunkte
 - Schnittstellenergebnisse treten an Dienstzugangspunkten auf
 - ▶ Dargestellt über Dienstprimitive
- Abstrakte Sicht eines Dienstes
 - Entspricht Sicht auf seine Dienstschnittstelle
 - ▶ Menge von Dienstzugangspunkten
 - ▶ Menge von Dienstprimitive
 - ▶ Zeitpunkt
 - ▶ Zugangspunktadresse (d.h. Ort)
 - ▶ Typ
 - ▶ Ablauffestlegungen

- Zusammensetzung der Beschreibung von Dienstprimitive aus drei Bestandteilen
 - Name
 - ▶ Z.B. Bitübertragung, Sicherung, Vermittlung, Transport
 - ▶ Engl.: Physical, Data Link, Network, Transport
 - ▶ Oft in abgekürzter Form (z.B. Ph, DL, N, T)
 - ▶ Kann auch entfallen, wenn nur ein Dienst betrachtet wird
 - Name einer Dienstleistung
 - ▶ Z.B. Connect, Data, Release, Reset, Abort, Disconnect
 - ▶ Oder Abkürzung: Con, Dat, Rel, Res, Abo, Dis
 - Bezeichnung der Grundtypen von Dienstprimitive
 - ▶ Request (Req)
 - ▶ Beauftragung (Dienstnehmer → Dienstgeber)
 - ▶ Indication (Ind)
 - ▶ Benachrichtigung des Partners (Dienstgeber → Dienstnehmer)
 - ▶ Response (Rsp)
 - ▶ Beantwortung durch Partner (Dienstnehmer → Dienstgeber)
 - ▶ Confirmation (Cnf)
 - ▶ Benachrichtigung über Abschluss (Dienstgeber → Dienstnehmer)

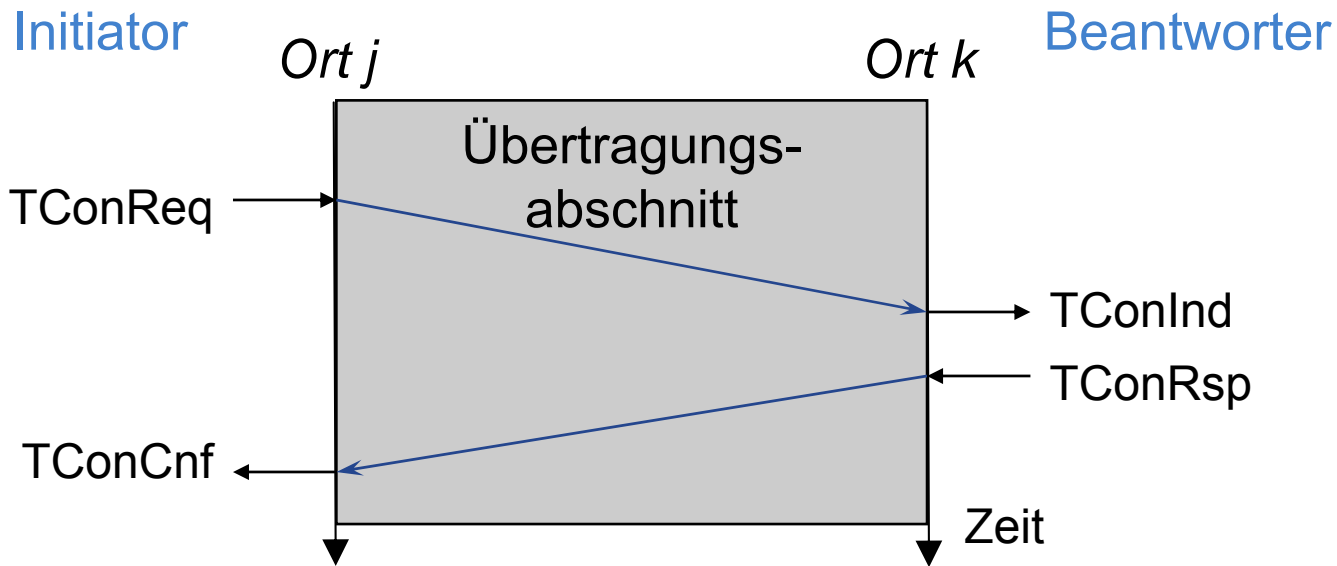
- Zusätzlich: Parameter
- Beispiel
 - TConReq(*Adressen*)
 - ▶ Verbindungsaufbauanforderung an der Schnittstelle zum Transportdienst
 - ▶ Adressen identifizieren Kommunikationspartner
 - HTTPGet[Req](*URL*)
 - ▶ Anforderung der HTML-Seite, die durch *URL* identifiziert wird
- Überblick

Name	Dienstleistung	Grundtypen	Parameter
Physical (Ph)	Connect (Con)	Request (Req)	<i>Abhängig vom Dienst</i>
Data Link (DL)	Data (Dat)	Indication (Ind)	
Network (N)	Release (Rel)	Response (Rsp)	
Transport (T)	Abort (Abo)	Confirmation (Cnf)	
HTTP	Provider Abort (PAbo)		
FTP	Disconnect (Dis)		
...	...		

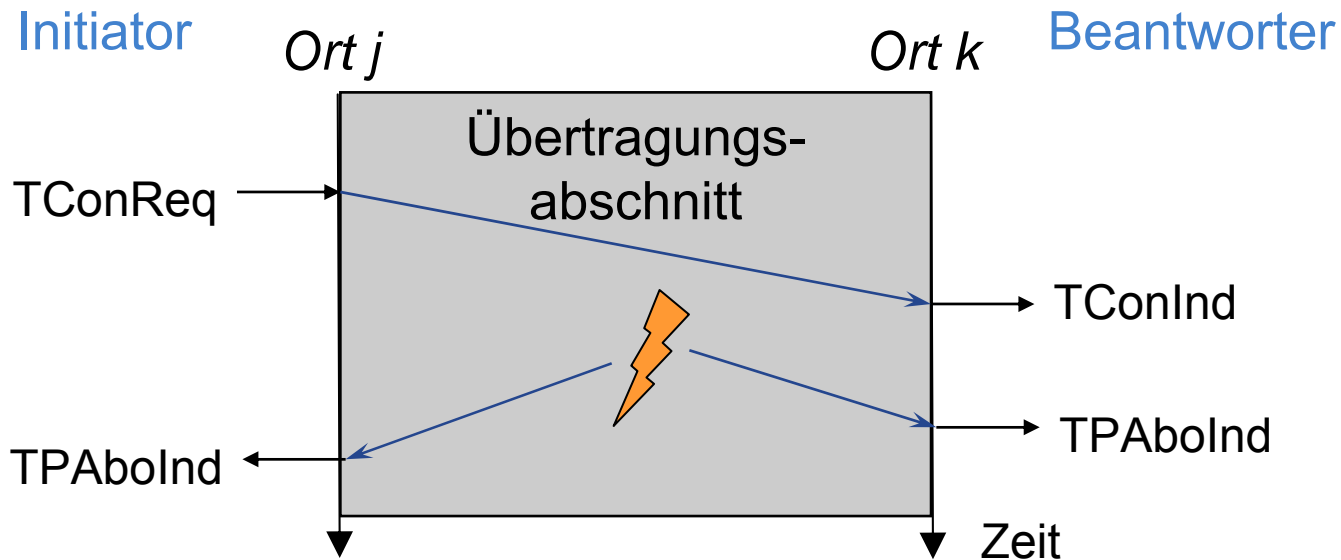
- **Weg-Zeit-Diagramme**
 - Zur groben Beschreibung von möglichen Abläufen an Dienstschnittstellen
 - ▶ Vgl. Nutzung zur Beschreibung von Abläufen des HDLC-Protokolls in Kapitel 5

- **Zustandsübergangsdigramme**
 - Nutzung von Zustandsautomaten
 - Auch genutzt zur Beschreibung von Protokollinstanzen

- Aufgabe
 - Darstellung räumlich verteilter Abläufe
 - Grafische Darstellung innerhalb eines 2-Achsen-Koordinatensystems
 - ▶ Vertikale Achse: Zeitachse
 - ▶ Abstrahiert vom tatsächlichen Zeitpunkt
 - ▶ Horizontale Achse: Räumliche Distanz
 - ▶ Abstrahiert von der tatsächlichen räumlichen Distanz
- Dargestellter Ablauf
 - Menge von Ereignissen (Dienstprimitiven)
 - ▶ Auftragen nach Ort und relativem Zeitpunkt



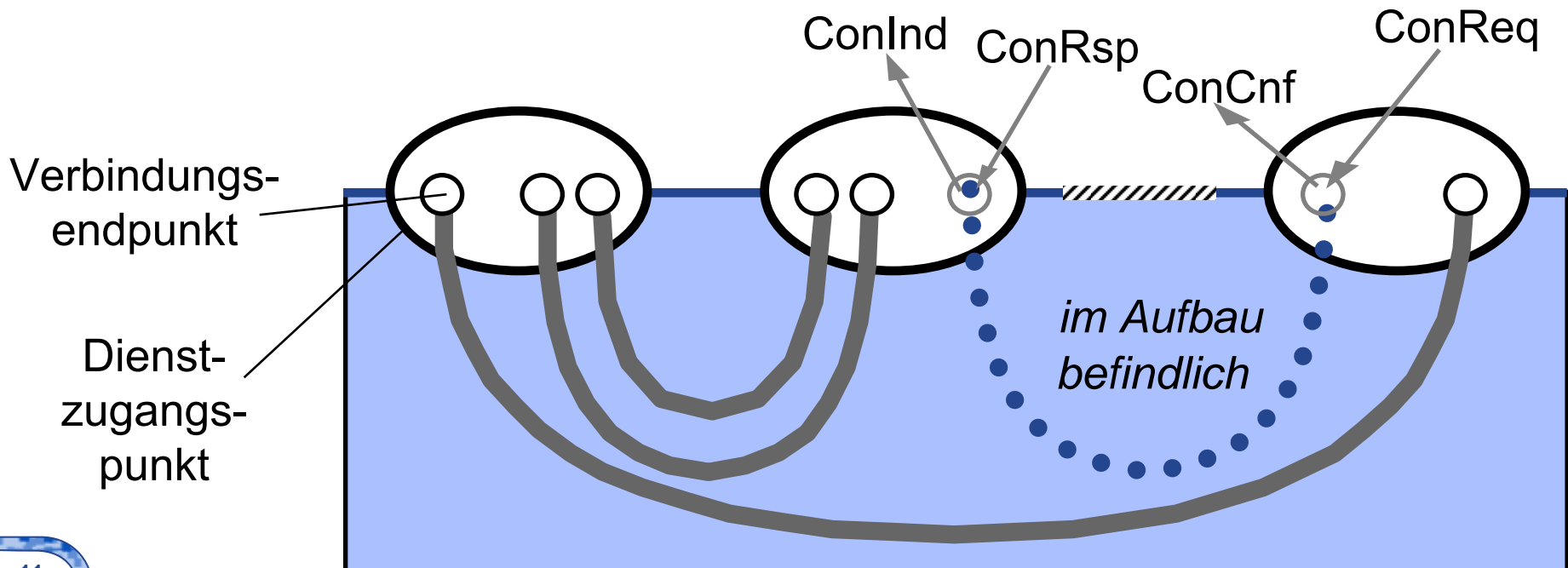
- Regulärer Ablauf der bestätigten Dienstleistung „Verbindungsaufbau“
 - Links: Zugangspunkt des Initiators am Ort *j*
 - Rechts: Zugangspunkt des Beantworters am Ort *k*
- Darstellung von Alternativen
 - In Weg-Zeit-Diagrammen unübersichtlich
 - Je Sachverhalt separates Diagramm



• Alternativer Ablauf

- Verbindungsaufbau wird an Partner kommuniziert
- Bearbeitung des Verbindungsaufbaus durch den Dienstgeber (Übertragungsabschnitt) abgebrochen, bevor Partner antworten kann

- Unterstützung mehrerer Dienstzugangspunkte
 - Dienstzugangspunktadressen identifizieren Benutzer
 - Adressen als Parameter in einigen Dienstprimitiven
- Pro Dienstzugangspunkt mehrere Verbindungen möglich
 - Kennzeichnung durch Verbindungsendpunkte



Beispiel: Verbindungsaufbau

Initiator

Beantworter

* und † bedeuten, dass die so markierten Daten möglicherweise geändert wurden.

ConReq(
*Initiatoradresse,
 Beantworteradresse,
 Qualitätsparameter,
 weitere Eigenschaften,
 kurze Nutzdaten₁*)

ConInd(
*Initiatoradresse,
 Beantworteradresse,
 Qualitätsparameter*,
 weitere Eigenschaften*,
 kurze Nutzdaten₁*)

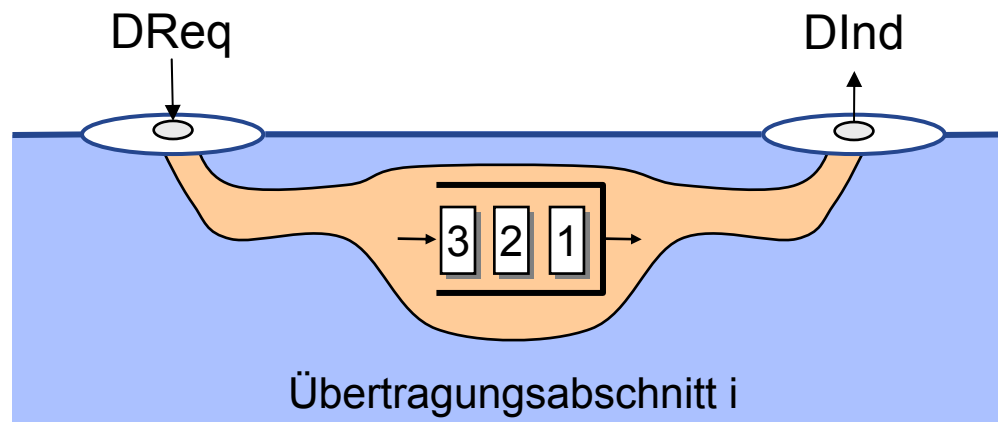


ConRsp(
*Initiatoradresse,
 Beantworteradresse,
 Qualitätsparameter†,
 weitere Eigenschaften†,
 kurze Nutzdaten₂*)

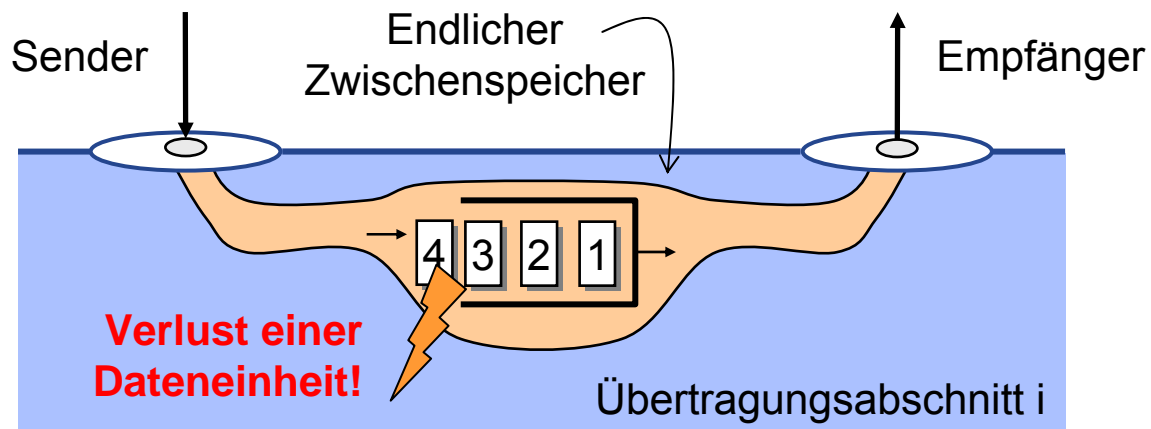
ConCnf (
*Initiatoradresse,
 Beantworteradresse,
 Qualitätsparameter†,
 weitere Eigenschaften†,
 kurze Nutzdaten₂*)

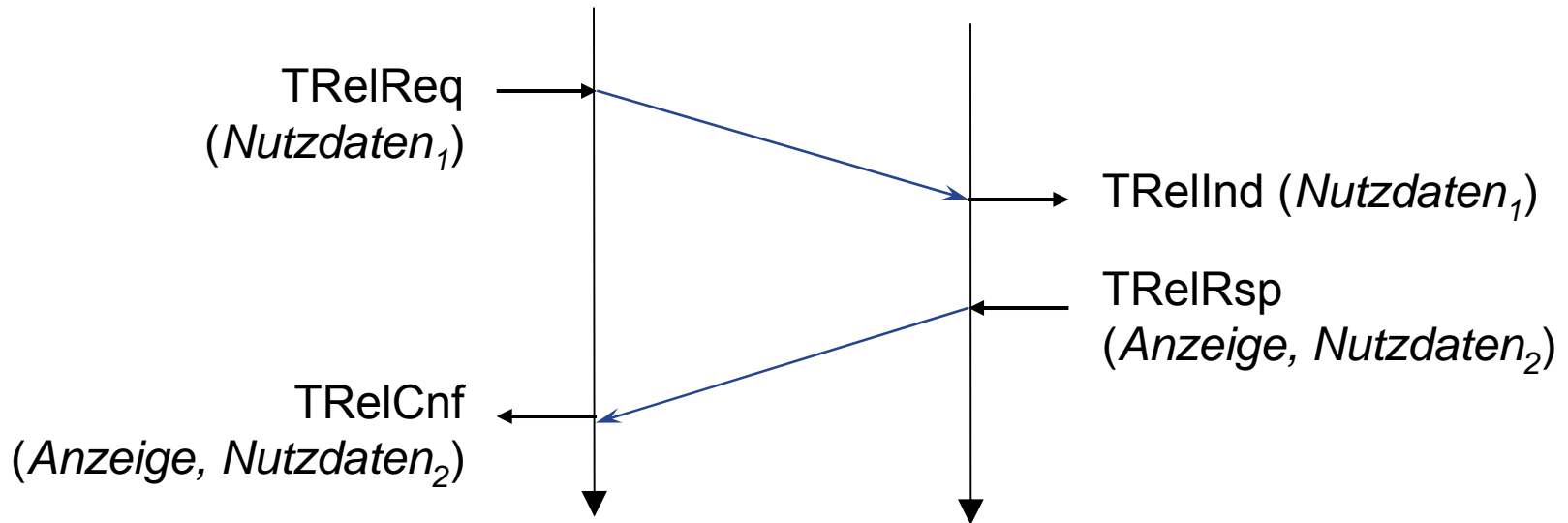
1 = Nutzdaten des Initiators
 2 = Nutzdaten des Beantworters

- Auslieferungsdisziplin
 - Zeitliche Reihenfolge der anstoßenden und daraus resultierenden Dienstprimitive
 - ▶ DReq und DInd
 - Varianten
 - ▶ treu zur Einlieferungsreihenfolge (FIFO)
 - ▶ zufällig
 - ▶ FIFO und priorisiert



- Rückstau
 - Zwischenpufferung der Daten in zwischenliegenden Systemen des Übertragungsabschnitts
 - Problem: Endlicher Zwischenspeicher
 - ▶ Verluste von Dateneinheiten
 - → **Staukontrolle** löst das Problem





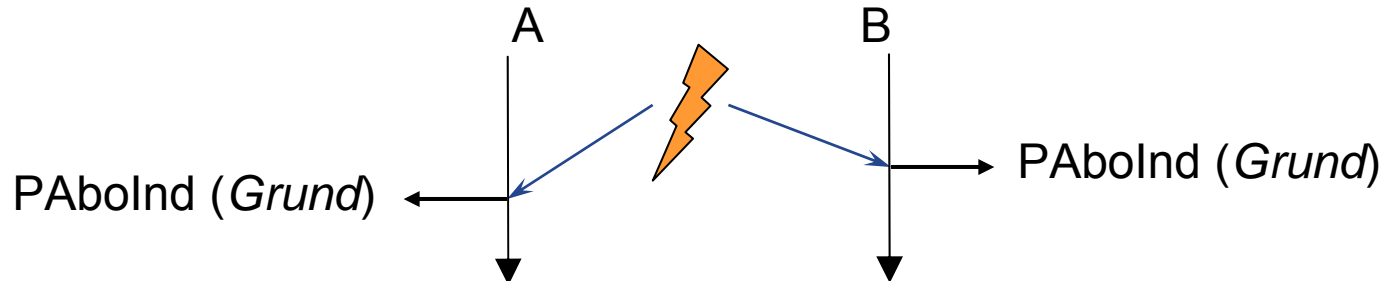
- Geregelter Verbindungsabbau
 - Partner kann Verbindungsabbau zustimmen oder ihn ablehnen

1 = Nutzdaten des Initiators
2 = Nutzdaten des Beantworters

- Außerplanmäßiger Abbruch einer Verbindung

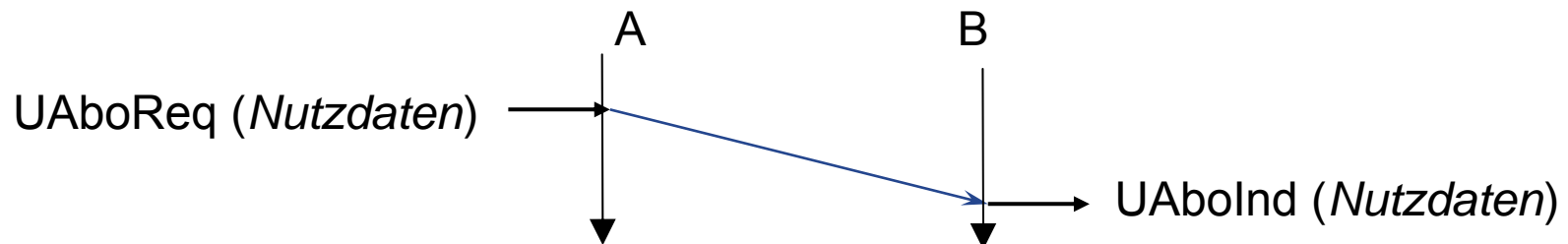
- Erbringerabbruch (Provider Abort, PAbo)

- ▶ Kann in jedem Zustand einer Verbindung passieren
- ▶ Keine Gleichzeitigkeit / bestimmte Reihenfolge garantiert

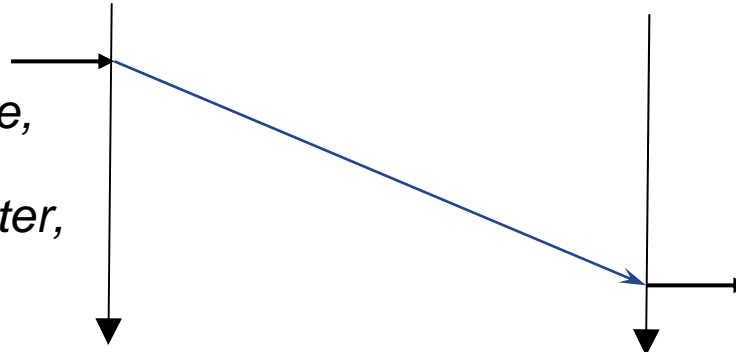


- Nutzerabbruch (User Abort, UAbo)

- ▶ Verbindung gilt für Initiator sofort als abgebrochen



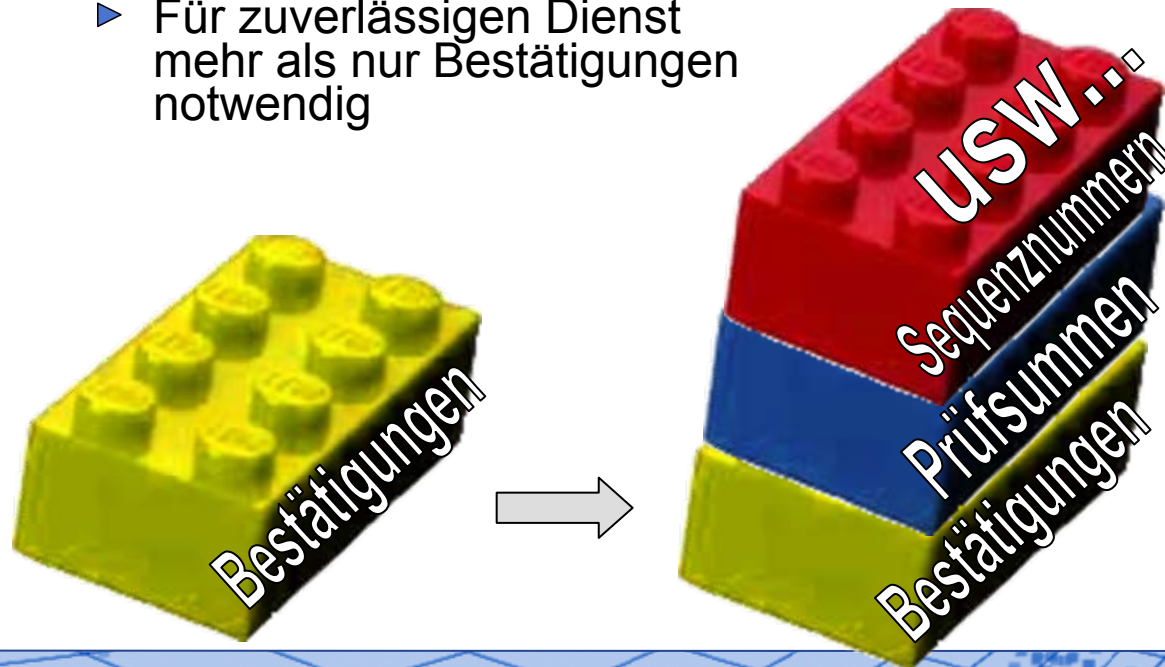
UnitDataReq(
Absenderadresse,
Zieladresse,
Qualitätsparameter,
Nutzdaten)



UnitDataInd(
Absenderadresse,
Zieladresse,
Qualitätsparameter,
Nutzdaten)

- Keine Verbindungen zwischen Kommunikationspartnern
 - Kein Zusammenhang zwischen verschiedenen Übertragungsleistungen
 - Unterstützt keine Auslieferungsdisziplin
 - ▶ z.B. keine Garantie für Reihenfolgegetreue
- Datagramm-Dienst realisiert unzuverlässige Dienstleistung
 - keine Aushandlung zwischen Kommunikationspartnern

- Bestätigt \neq Zuverlässig
 - Zuverlässiger Dienst stellt sicher, dass alle Daten korrekt übertragen wurden (Wiederholung Kapitel 3)
 - ▶ Alle Daten korrekt
 - ▶ In der richtigen Reihenfolge
 - ▶ Ohne Duplikate
 - ▶ Ohne Phantom-Dateneinheiten
 - Bestätigungen nur ein Baustein für zuverlässige Dienste
 - ▶ Für zuverlässigen Dienst mehr als nur Bestätigungen notwendig



Auszug aus
möglichem
Aufbau eines
zuverlässigen
Dienstes

Anwendungsbeispiel: Fluganfrage im Internet

WWW-Benutzer:
Flugbucher



Eingabe

www.weg-hier.info/search.pl
von: Frankfurt
nach: Auckland
am: 02.06.06

HTTP-Instanz

Baue Transport-
schichtverbin-
dung auf

Schicke URL &
Parameter
der gewünsch-
ten Suche
(inkl. Quittung)

Gebe Seite an
Web-Browser

Dienst-
nehmer

Protokoll-
instanz

Basisdienst

Dienst

HTTP-Instanz

Nehme
Transport-
schicht-
verbindung an

Gebe URL &
Parameter an
Web-Server

Schicke Inhalt
der gewünsch-
ten Seite



Reisebüro
WWW-Server

Generierung der
Ergebnisseite
www.weg-hier.info/search.pl

Übergebe Inhalt der Seite
www.weg-hier.info/search.pl

Dienst-
nehmer

Protokoll-
instanz

TConReq

TConInd

TConRsp

TConCnf

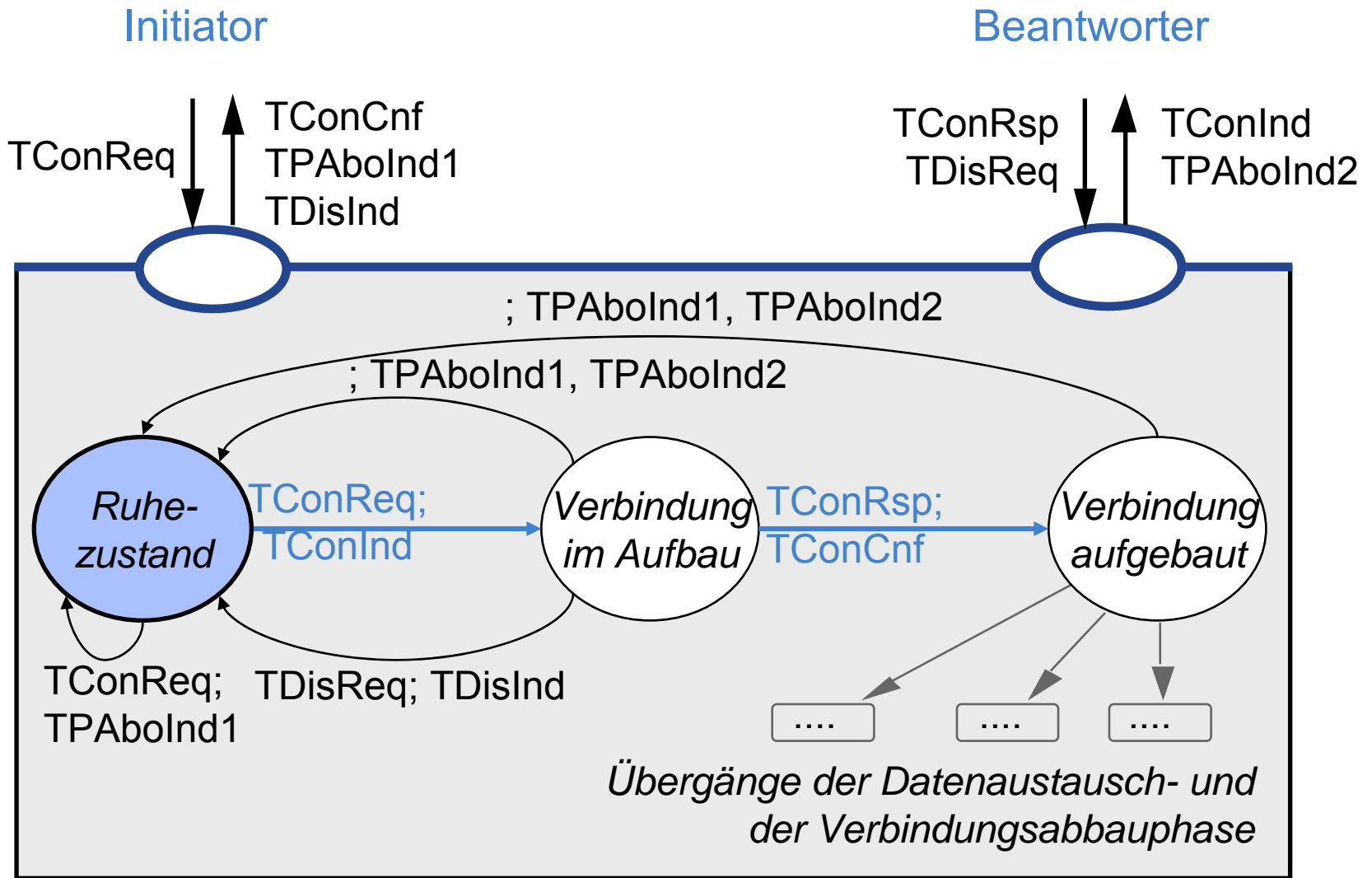
TDatReq

TDatInd

TDatReq

TDatInd

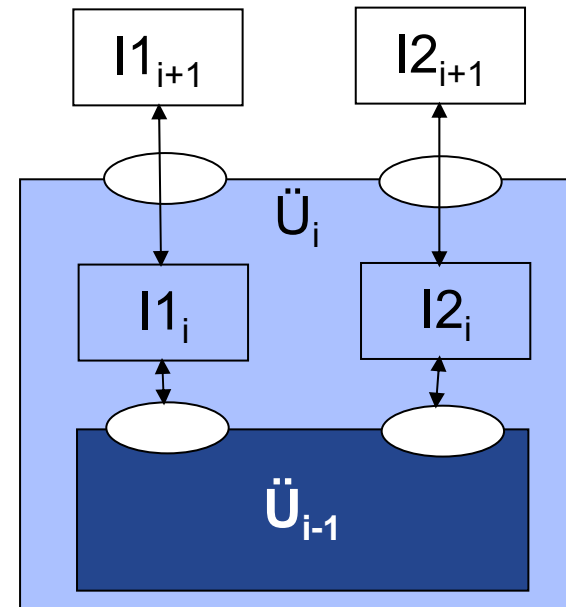
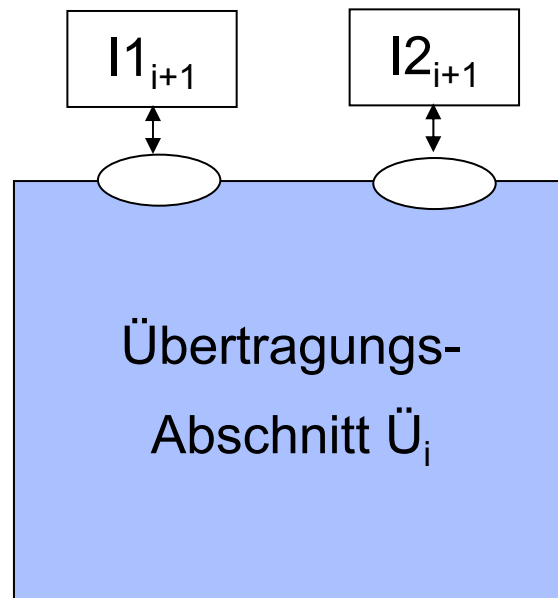
- Zustandsautomaten
 - Können Verhalten des Übertragungsabschnittes vollständig beschreiben
- Graphisches Hilfsmittel
 - Zustandsübergangsdigramme
 - ▶ Zustand
 - ▶ als Kreise notiert
 - ▶ Zustandsübergang
 - ▶ Pfeil vom aktuellen Zustand in den Nachfolgezustand
 - ▶ Beschriftung: Diensttypen Stimulus und Reaktion
 - ▷ Z.B. Stimulus; Reaktion1, Reaktion2
 - ▶ Spontanübergänge: Keine Angabe eines Stimulus



- Dienstprimitive beim Initiator
 - TConReq: Anforderung eines Verbindungsaufbaus
 - TConConf: Verbindungsaufbau erfolgreich
 - TDisInd: Beantworter lehnt Verbindungsaufbau ab
 - TPAboInd1: Abbruch durch Übertragungsabschnitt
- Dienstprimitive beim Beantworter
 - TConRsp: Bestätigung des Verbindungsaufbauwunschs
 - TDisReq: Ablehnung des Verbindungsaufbauwunschs
 - TConInd: Mitteilung des Verbindungsaufbauwunschs
 - TPAboInd2: Abbruch durch Übertragungsabschnitt

- Verhalten des Übertragungsabschnitts durch Zustandsautomaten repräsentiert
- Drei Zustände
 - Ruhezustand
 - ▶ Startzustand
 - ▶ Nur TConReq als Stimulus möglich
 - ▶ ConInd als Reaktion, falls Medium Wunsch annimmt
 - ▶ PAbolnd1 als Reaktion, falls Medium Wunsch ablehnt
 - Verbindung im Aufbau
 - ▶ ConReq erhalten und mit ConInd reagiert
 - ▶ ConRsp erwartet, dann ConConf als Reaktion
 - Verbindung aufgebaut
 - ▶ Zustand kann im Beispiel nur durch Spontanübergang verlassen werden

- Abstrakter Übertragungsabschnitt
 - Interne Betrachtung
 - ▶ Instanzen, die über untergeordneten Übertragungsabschnitt kommunizieren
- Protokoll
 - Formate und Regeln zur Kooperation der Instanzen einer Schicht
 - ▶ Beschreibung der Regeln als Zustandsübergangdiagramme
 - ▶ Formate hier nicht weiter betrachtet



- Informelle Beschreibung
 - Nutzung einer unbestätigten Datenübertragung als Basisdienst
 - Bereitstellung einer bestätigten Datenübertragung für den Dienstonutzer
 - ▶ Verwendung des ARQ-Verfahrens Stop-and-Wait
 - ▶ Senden einer Dateneinheit – Warten auf Quittung
 - ▶ Halbduplex-Datenübertragung
 - ▷ Mit Quittung können Nutzdaten übertragen werden
 - ▶ Sequenznummern alternieren zwischen 0 und 1
 - ▷ Unterscheidung zwischen dem Verlust von Dateneinheiten und demjenigen von Quittungen möglich
 - Nachdem eine bestätigte Datenübertragung vollständig beendet ist (Quittung erhalten) geht Instanz in einen Ruhezustand über, bis die nächste Übertragungsanforderung eintrifft
 - ▶ Zwei Zustände: Ruhezustand und Warten auf Response
 - Dienstprimitive
 - ▶ DtReq, DtInd, DtRsp, DtCnf

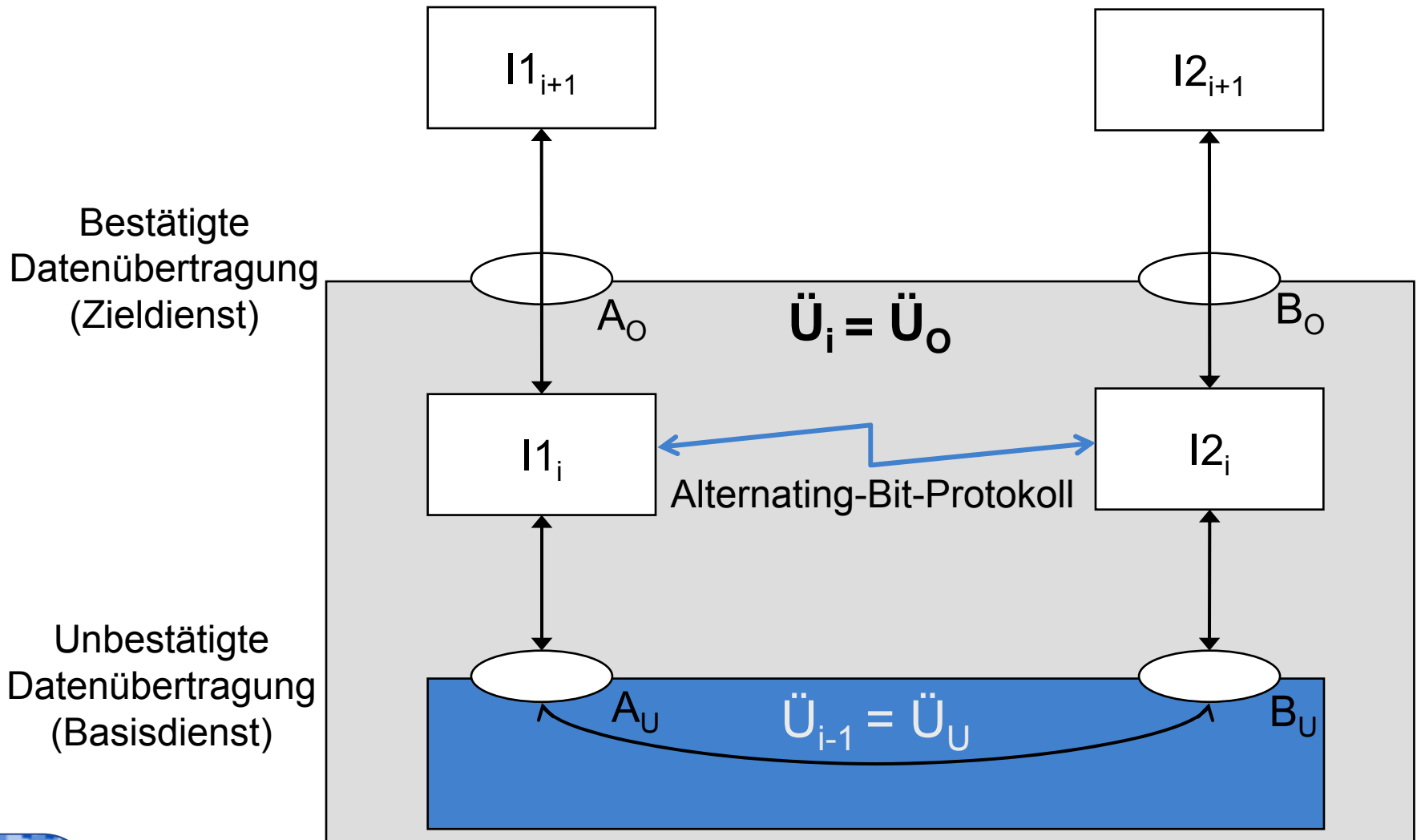
- Vereinfachungen

- Zieldienst O

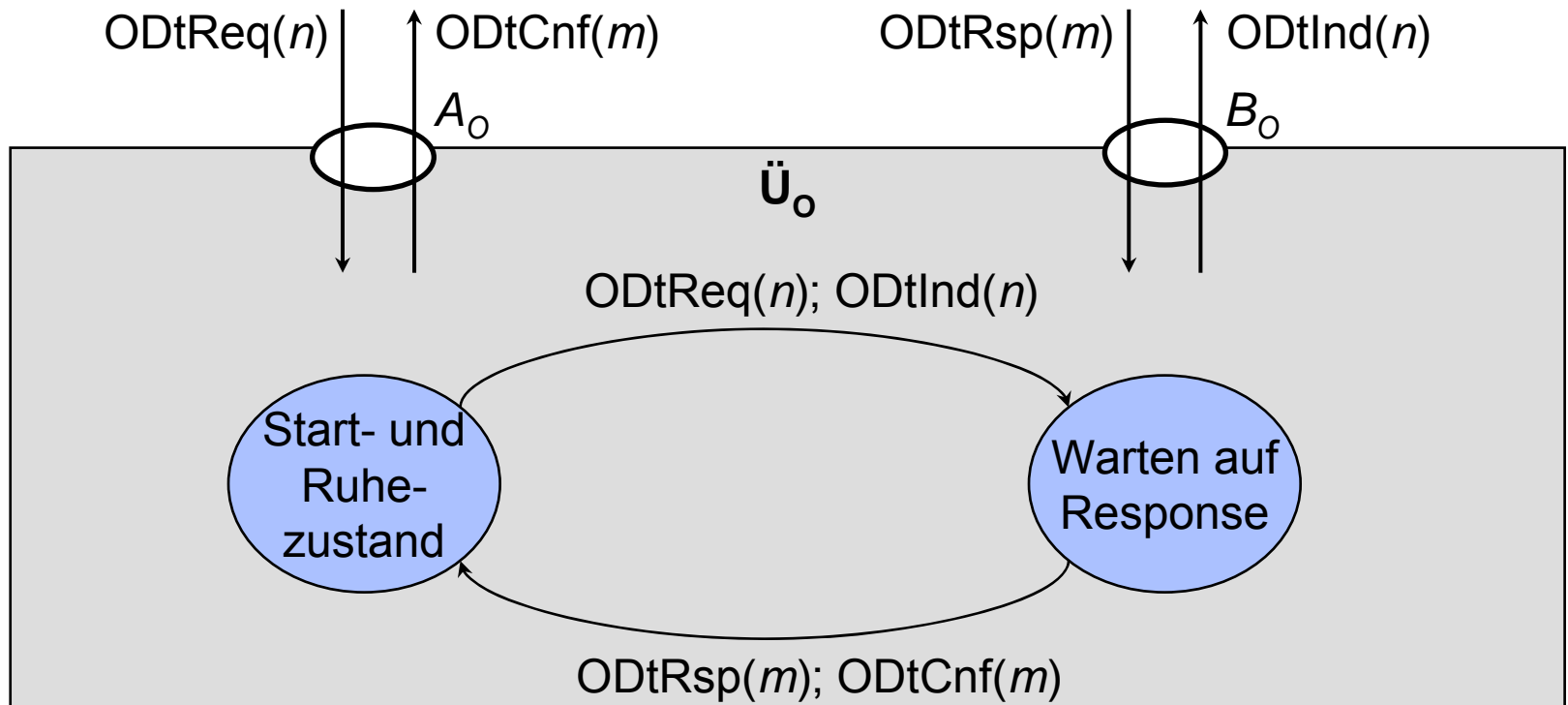
- ▶ Es darf sich immer nur eine Dienstleistung in Bearbeitung befinden
 - ▶ Eine Dienstleistung entspricht dem Senden einer Dateneinheit
 - ▶ Für praktischen Einsatz in vielen Fällen ineffizient (vgl. Kapitel 3)
 - ▶ Fehlerfälle (Verfälschungen etc.) nicht betrachtet
 - ▶ Für praktischen Einsatz ist dieses Protokoll daher nicht tauglich

- Basisdienst U

- ▶ Verfälschungen werden dem Dienstnehmer angezeigt
 - ▶ In der Praxis besteht immer eine gewisse Wahrscheinlichkeit für unentdeckte Verfälschungen
 - ▶ Dienst ist immer verfügbar



- Bereitstellung einer bestätigten Datenübertragung
 - Initiator immer am Dienstzugangspunkt A_O
 - Beantworter immer am Dienstzugangspunkt B_O
 - Parameter n, m : Nutzdaten



Initiator

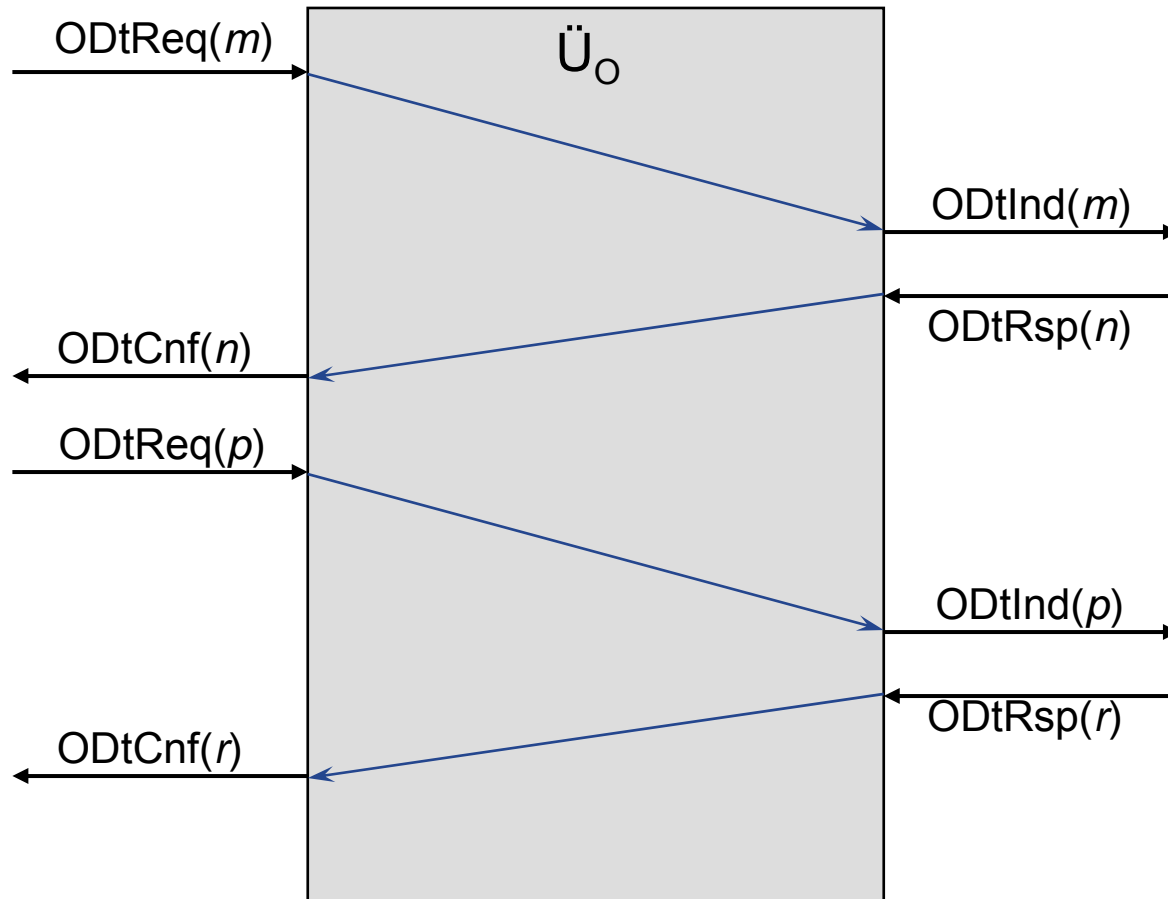
Beantworter

Dienstzugangspunkt

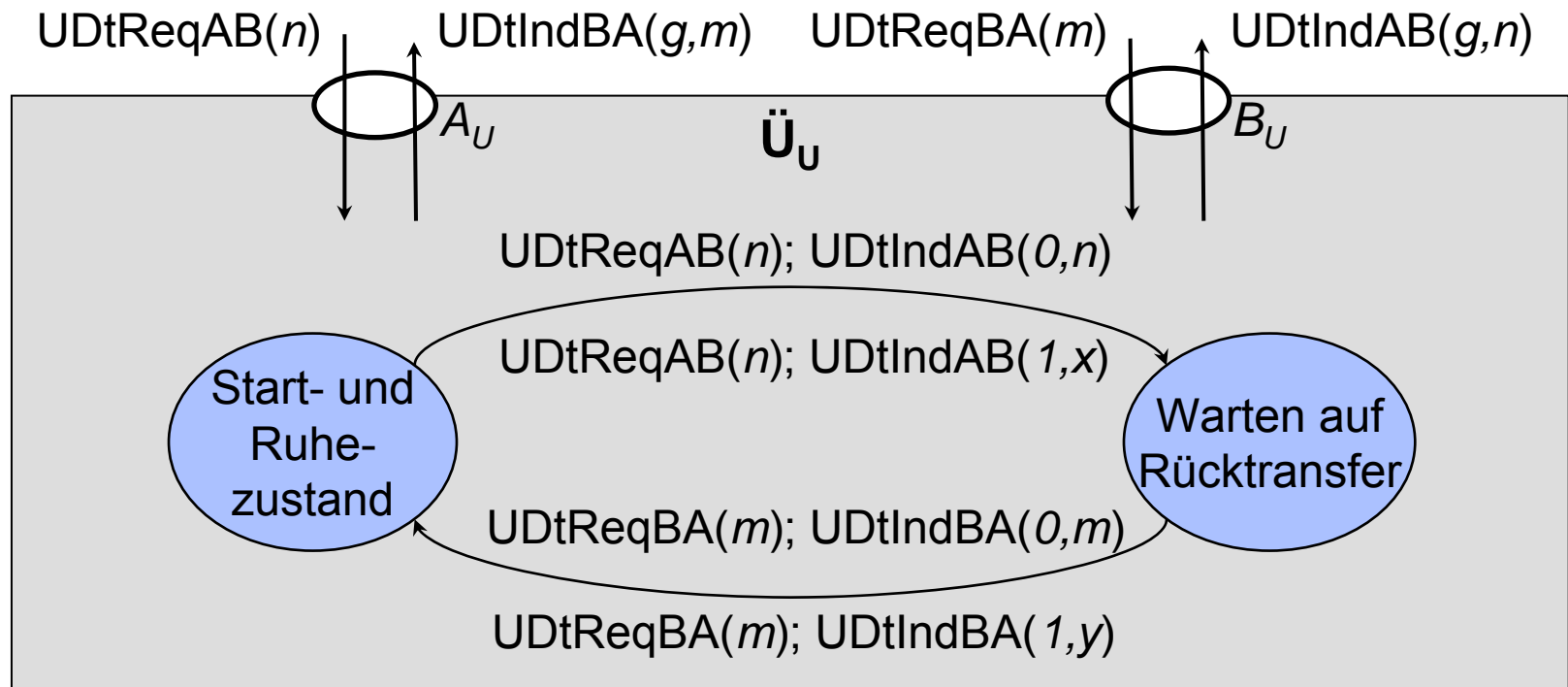
Dienstzugangspunkt

A_o

B_o



- Bereitstellung einer unbestätigten Datenübertragung
 - Parameter n, m : Nutzdaten; x, y : verfälschte Nutzdaten
 - Parameter g : „0“ – keine Verfälschung, „1“ Verfälschung erkannt
 - Start immer am Dienstzugangspunkt A_U
 - Verfälschungen möglich, Verdopplungen und Verlust nicht



Initiator

Beantworter

Dienstzugangspunkt

Dienstzugangspunkt

A_U

B_U

UDtReqAB(n)

\ddot{U}_U

UDtIndAB($0, n$)

UDtIndBA($0, m$)

UDtReqBA(m)

UDtReqAB(n)

Verfälschung

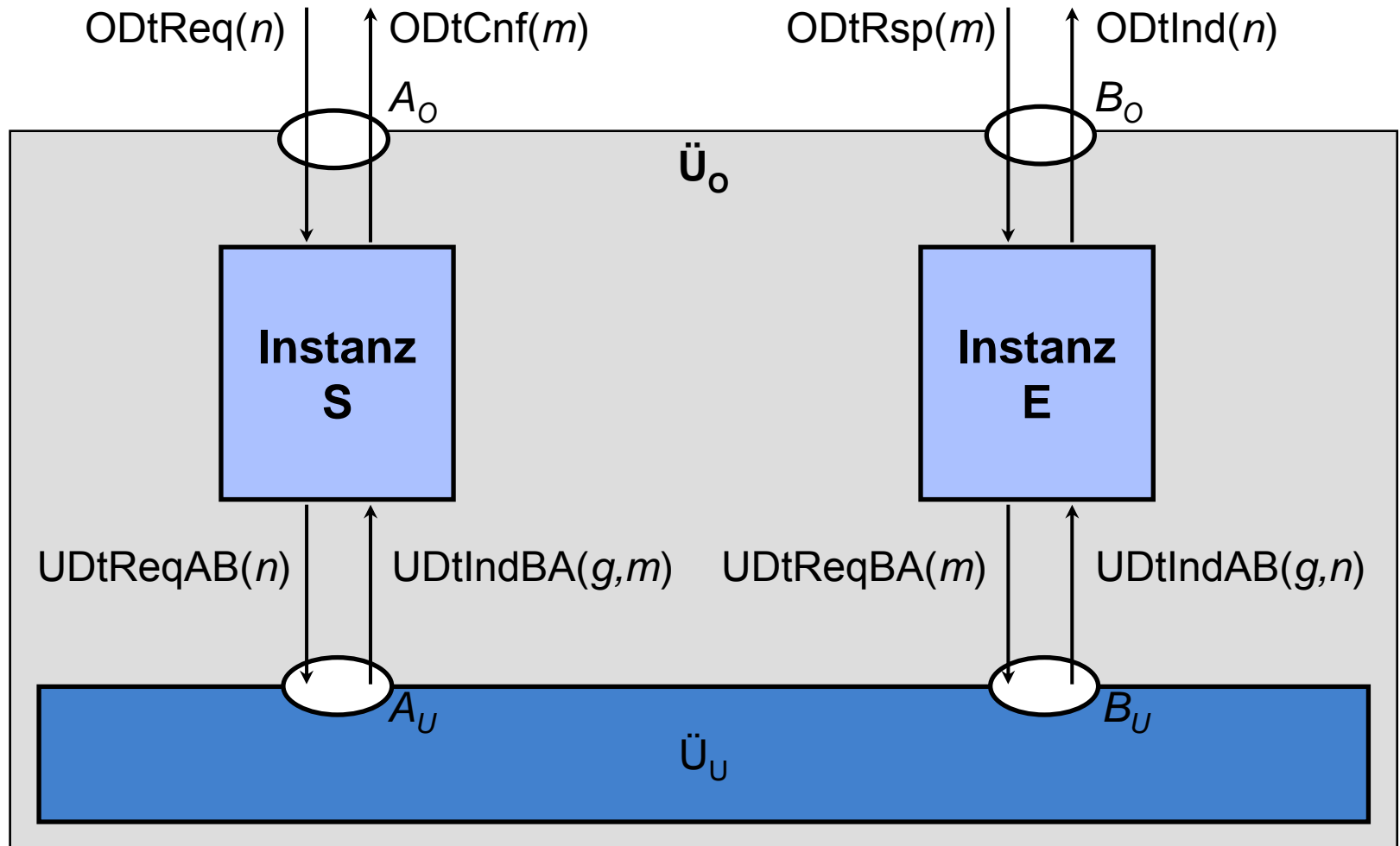
UDtIndAB(**1**, x)

verfälschte Daten m
zeigt Verfälschung an

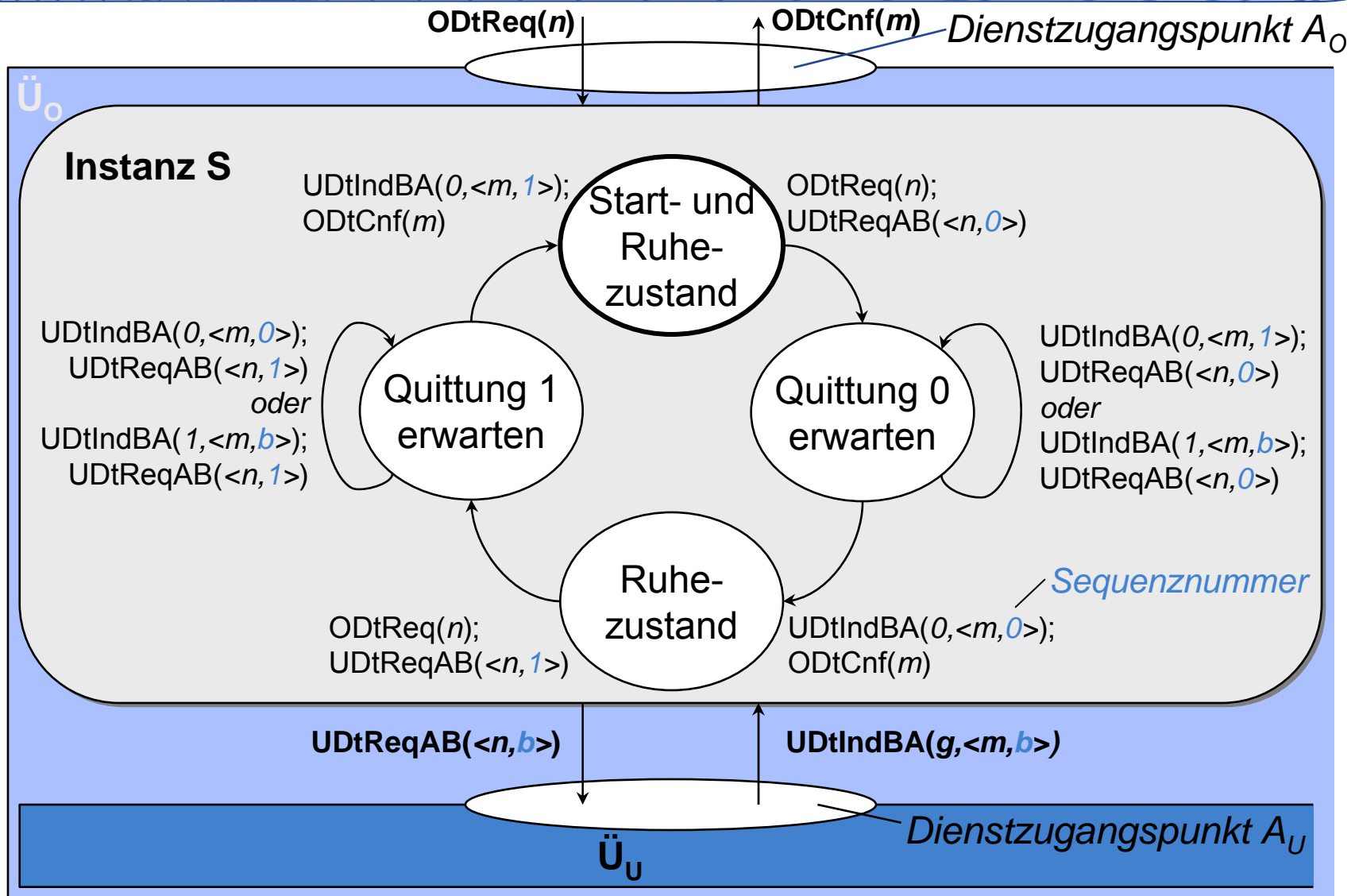
UDtIndBA(**1**, y)

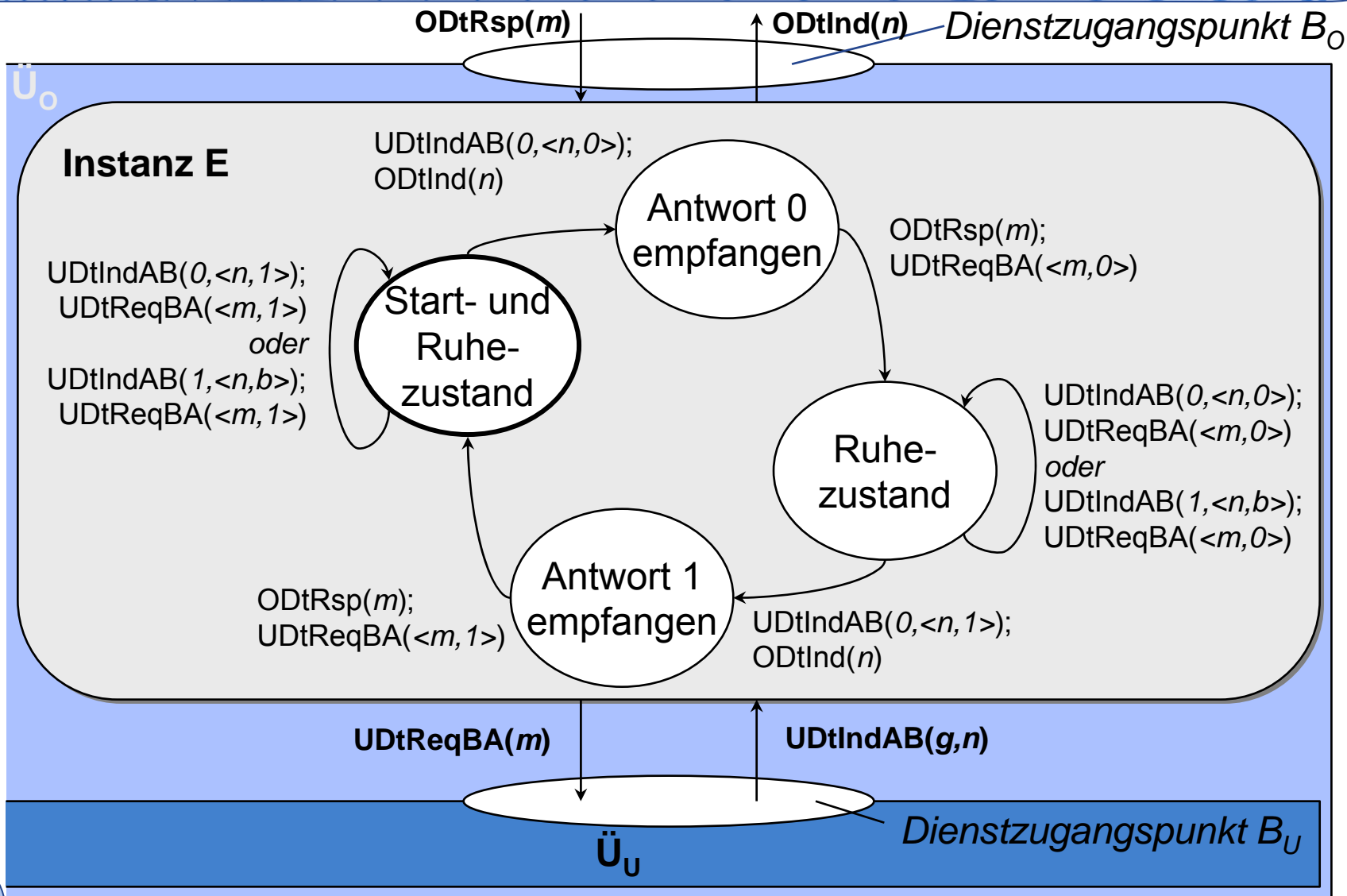
UDtReqBA(m)

Verfälschung



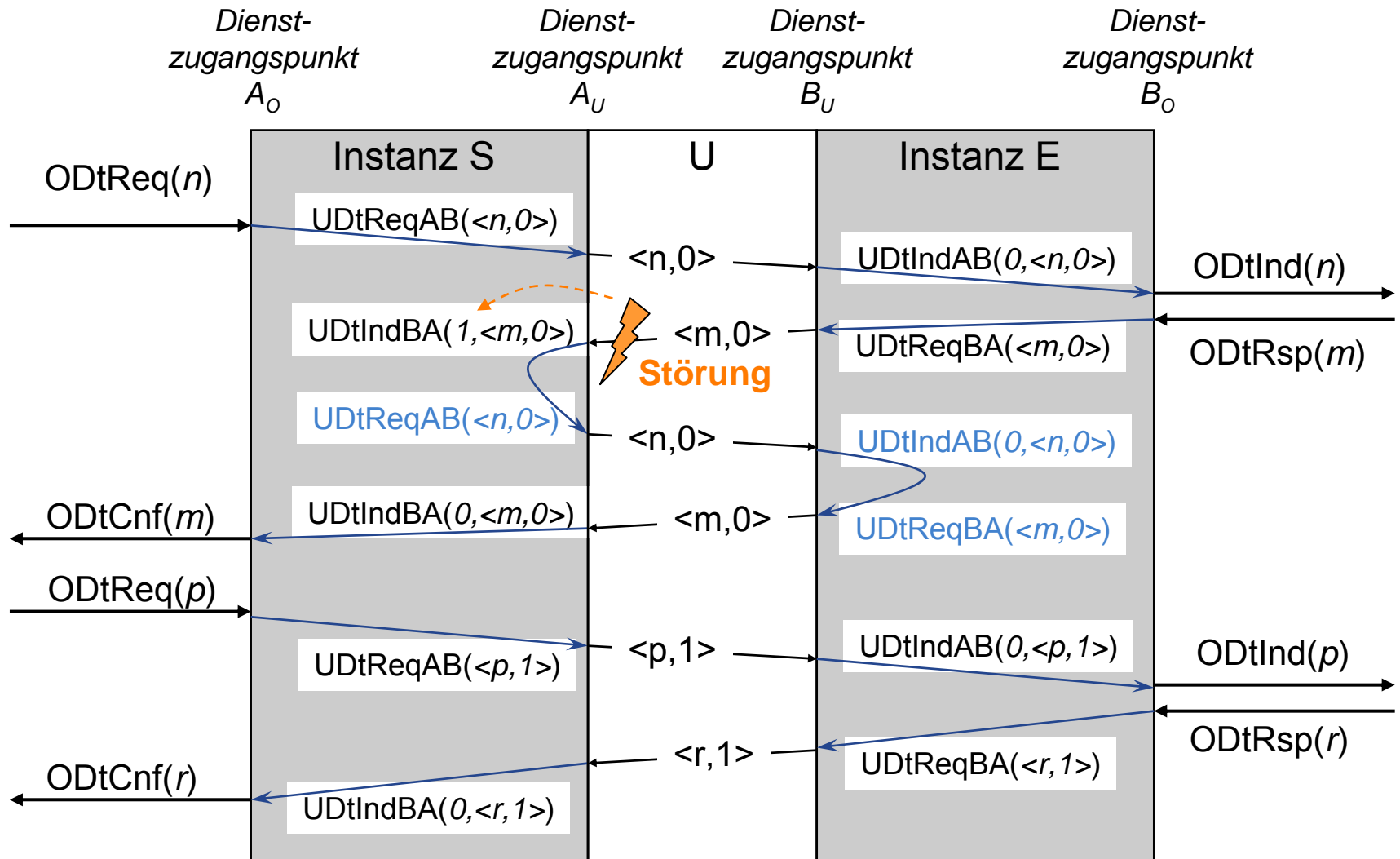
- Endlicher Automat mit 4 Zuständen
 - Start- und Ruhezustand
 - ▶ Durch ODtReq aktiviert
 - ▶ Erzeugt UDtReqAB
 - ▶ Übergang in Folgezustand „Quittung 0 erwarten“
 - Quittung 0 erwarten
 - ▶ Beim Eintreffen von UDtReqBA verlassen
 - ▶ Rücksetzen, d.h. UDtIndAB wiederholen
 - ▶ Falls $g=1$
 - ▶ Falls falsche Sequenznummer ($s=1$)
 - ▶ Ansonsten: Übergang in Ruhezustand, Ausgabe ODtConf(m)
 - Ruhezustand
 - ▶ Mit Start- und Ruhezustand vergleichbar
 - Quittung 1 erwarten
 - ▶ Wie „Quittung 0 erwarten“, nur vertauschte Sequenznummern



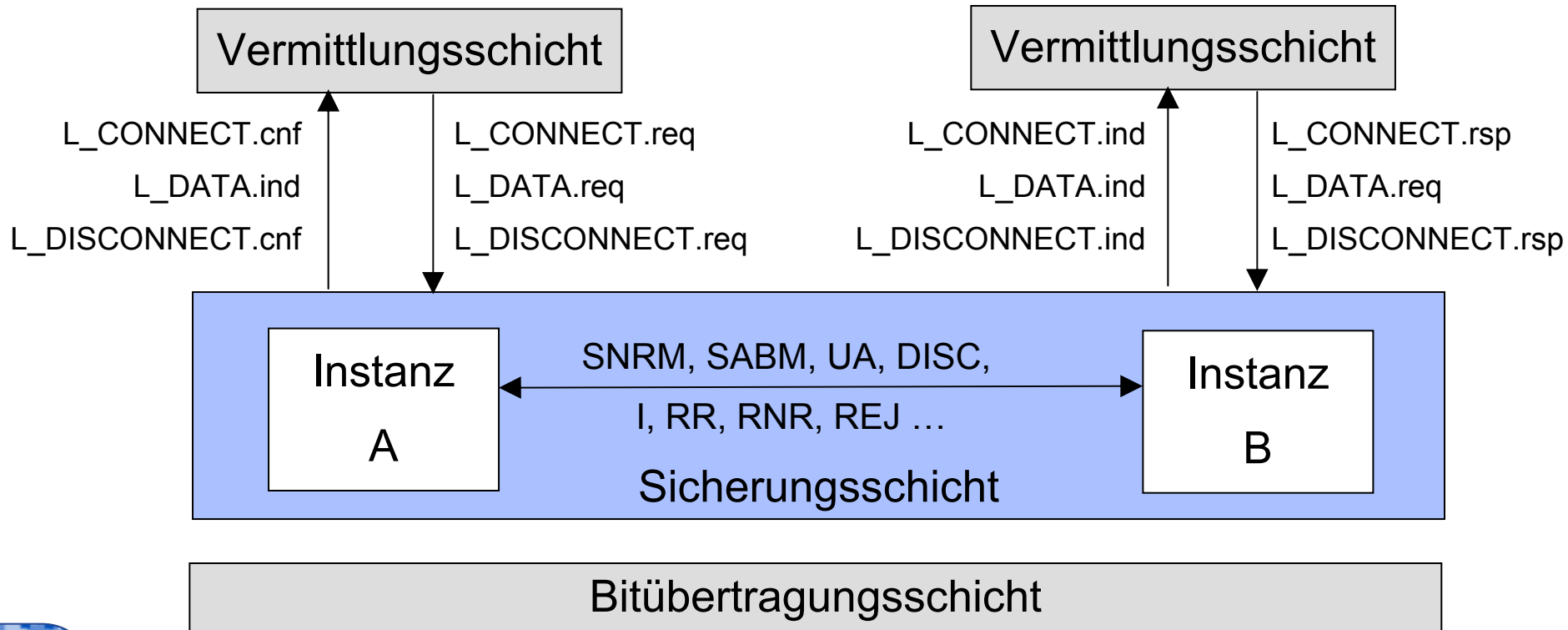


- Sequenznummer mit falschem Wert
 - Wirkt als negative Quittung
 - Aufforderung zur Sendewiederholung
- Sequenznummer mit richtigem Wert
 - Ermöglicht Erkennen von Duplikaten beim Empfänger
- Beispiel auf folgender Folie
 - Dateneinheit mit Sequenznummer 0 wird korrekt übertragen
 - Empfänger antwortet entsprechend
 - Antwort wird verfälscht
 - Sender wiederholt Dateneinheit mit Sequenznummer 0
 - Empfänger erkennt Duplikat
 - ▶ Ignoriert Dateneinheit
 - ▶ Liefert Daten nicht an den Dienstnehmer (höhere Schicht) ab
 - ▶ Wiederholt Antwort

Vollständiger Beispielablauf



- Vereinfachter Überblick
 - Dienstprimitive und Dateneinheiten



- Einfache Beschreibungsmethoden von Diensten und Protokollen
 - Weg-Zeit-Diagramme
 - Zustandsübergangsdigramme auf der Basis endlicher Automaten
- Beispiele
 - Alternating-Bit-Protokoll
 - HDLC
- ... so what?
 - Überprüfung der Korrektheit eines Protokolls
 - Basis für interoperable Implementierungen
 - ▶ Automatische Erzeugung
 - ▶ Händische Programmierung
 - Nicht formale Beschreibung von Protokollen lässt oftmals „Interpretationsspielraum“
 - ▶ Ergebnis: nicht interoperable Implementierungen

- 6.1 Was sind Dienstprimitive und wo treten sie in Erscheinung?
- 6.2 Wofür steht das Dienstprimitiv TDisConf? Was bedeutet NAbolnd?
- 6.3 Geben Sie das Weg-Zeit-Diagramm eines bestätigten Verbindungsaufbaus auf der Transportschicht an.
- 6.4 Erläutern Sie den Zusammenhang von Dienstzugangspunkt und Verbindungsendpunkt.
- 6.5 Wieviele Dateneinheiten können gleichzeitig über das Alternating-Bit-Protokoll zwischen Sender und Empfänger unterwegs sein? Werden hierfür überhaupt Sequenznummern benötigt? Begründen Sie Ihre Antwort.
- 6.6 Welche Dienstprimitive treten im Alternating-Bit-Protokoll am Dienstzugangspunkt zum „tieferliegenden“ Basisdienst auf? Welche am „oberen“ Dienstzugangspunkt zum Alternating-Bit-Protokoll und wo dort genau?
- 6.7 Erläutern Sie den Unterschied zwischen dem Zustandsübergangdiagramm eines Übertragungsabschnittes und dem Automat einer Protokollinstanz.

[AbLo03] Abeck, Lockemann, Schiller, Seitz; Verteilte Informationssysteme, dpunkt-Verlag, 2003

- Kapitel 5

[Hals05] F. Halsall; Computer Networking and the Internet; Addison-Wesley, 2005

- Kapitel 1.4.7 – Protocol Specification

[LoKr93] Lockemann, Krüger, Krumm; Telekommunikation und Datenhaltung; Hanser-Verlag, 1993

- Kapitel 6