

Tutoriumsblatt 1 mit Musterlösung

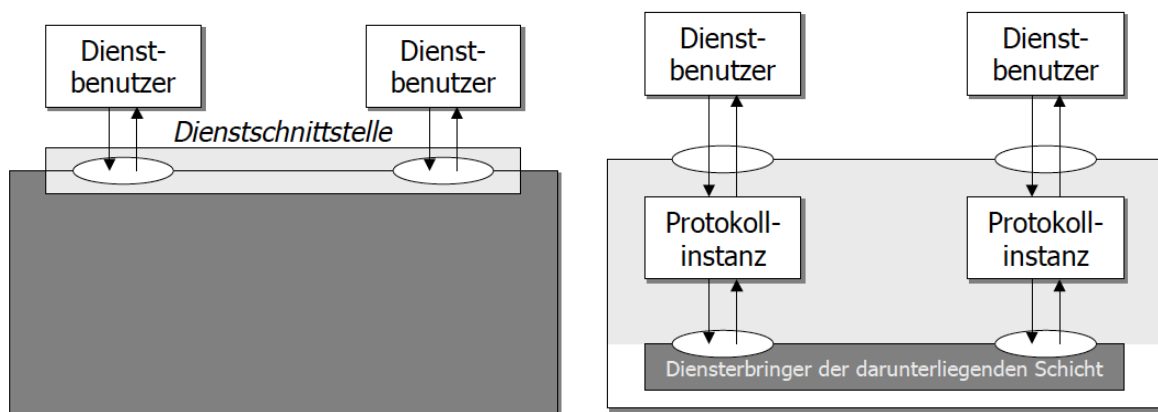
Aufgabe 1.1: Dienste und Protokolle

- Grenzen Sie die Begriffe *Dienst* und *Protokoll* gegeneinander ab.
- Was bedeuten die Abkürzungen *SDU*, *PDU* und *PCI*, und in welchem Verhältnis stehen die Begriffe zueinander? Erläutern Sie die Begriffe am Beispiel des HTTP¹.

Lösung 1.1

1.a) *Dienst*: Funktionalität eines Kommunikationssystems, d.h. das 'was' des Kommunikationssystems. Oder auch: die Gesamtheit der Funktionalität der Dienstprimitive, die eine Schicht der nächsthöheren Schicht (oder einer Anwendung) an einem Dienstzugangspunkt zur Verfügung stellt. Die Details, wie der Dienst erbracht wird, sind verborgen.

Protokoll: Umsetzung der Funktionalität eines Kommunikationssystems, d.h. das 'wie' des Kommunikationssystems. Genauer: die Regeln und Konventionen (z.B. Paketformat, Abfolge, Bedeutung), mit denen zwei Instanzen der gleichen Schicht miteinander kommunizieren. Protokolle ermöglichen Schichten, ihre angebotenen Dienste auszuführen.



- 1.b)** SDU - Service Data Unit
 PCI - Protocol Control Information
 PDU - Protocol Data Unit

Betrachten wir eine Schicht N . Diese Schicht bekommt von der Schicht $N + 1$ Daten zur Übertragung. Eine Dateneinheit der Schicht $N + 1$ stellt für Schicht N die SDU dar – die 'Data Unit', für die der Dienst der Schicht erbracht werden soll.

Schicht N versieht diese SDU mit Kontrollinformation – den PCI. Diese werden der SDU meist in Form eines Headers vorangestellt. Die Kontrollinformationen sind notwendig, um zwischen den Protokollinstanzen einer Schicht Informationen auszutauschen, damit das Protokoll seinen Dienst erbringen kann. Erst die PCI ermöglichen es allen beteiligten Protokollinstanzen, immer im gleichen Zustand zu sein und Fehlersituationen zu erkennen/beheben.

Die SDU und die PCI bilden die PDU der Schicht N – eine Einheit, die durch das Protokoll weitergeleitet bzw. verarbeitet wird.

¹http://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Für Schicht $N - 1$ stellt die PDU der Schicht N wiederum die SDU dar.

All diese Begriffe sind auch z.B. bei HTTP zu finden. Dieses Protokoll dient zur Anforderung und Übertragung von Webseiten (und allem anderen auch; aber das sei hier mal egal) und ist ein Protokoll der Anwendungsschicht. Ruft man eine URL (z.B. `www.i4.de/index.html`) im Browser auf, wird sie mittels HTTP an den Webserver geschickt und eine Webseite wird zurückschickt. Die URL bzw. die Webseite sind die SDU, die zur Übertragung an HTTP übergeben wird. Es wird jeweils ein Header hinzugefügt (PCI), z.B. die Information, um welchen Typ von Anfrage es sich handelt oder ob die angeforderte Webseite existiert.

Beispiel Anfrage: `GET index.html HTTP/1.1`. Hier wird die Anfrage (in gekürzter Version, aber das ist eine Eigenart von HTTP) in die PCI eingebettet.

Beispiel Antwort: `HTTP/1.1 200 OK` gefolgt von der Webseite. Dies sind PCI gefolgt von der SDU.

In beiden Fällen gibt es noch viele weitere PCI, aber das ist hier egal.

In beiden Fällen werden PCI und SDU zur PDU zusammengefasst (wie oben dargestellt) und an TCP auf der nächstunteren Schicht gegeben. Für TCP ist dies die SDU.

Aufgabe 1.2: Dienstprimitive und Weg-Zeit-Diagramme

Ein Dienst wird über Dienstprimitive bereitgestellt; der Zusammenhang zwischen Dienstprimitive kann durch Weg-Zeit-Diagramme visualisiert werden.

- a) Betrachten Sie den Telefondienst auf Folie I-19. Gezeigt ist die Kommunikation zweier Dienstnutzer über den Dienstbringer 'Telefongesellschaft'. Zur Initiierung, Durchführung und Beendigung des Telefongesprächs werden Dienstprimitive verwendet. Dienstprimitive haben generell einen von vier Grundtypen: **Request**, **Indication**, **Response** und **Confirmation**, siehe Folie I-28.

Der auf der Folie dargestellte Ablauf des Telefongesprächs verwendet 'intuitiv' benannte Dienstprimitive. *Geben Sie für jedes Dienstprimativ an, von wem (Aufrufer) zu wem (Aufgerufener) es geht und welchen Typ es hat.* Nutzen Sie dabei als Bezeichnung der Dienstnutzer und Dienstzugangspunkte die folgenden Namen:



Tragen Sie ihre Lösung in die folgende Tabelle ein:

Nr.	Primitiv beim Telefongespräch	Aufrufer	Aufgerufener	Typ
1	Wahlaktivierung			
2	Wählton			
3	Wahl der Nummer			
4	Klingeln			
5	Freizeichen			
6	Gesprächsannahme			
7	Ende Freizeitichen / Gesprächsaufbau			
8	Gespräch	---	---	---
9	Gesprächsbeendigung			
10	Belegttton			

Erstellen Sie *je ein Weg-Zeit-Diagramm* der Dienste zur Übertragung

- b) einer *WhatsApp-Nachricht* und
c) einer *E-Mail* vom Absender bis hin zum Mail-Server des Empfängers.

Was sind jeweils Dienstnutzer, was Dienstbringer? Welche Dienstprimitive werden verwendet? Sind die Dienste bestätigt oder unbestätigt?

Lösung 1.2

2.a)

Nr.	Primitiv beim Telefongespräch	Aufrufer	Aufgerufener	Typ
1	Wahlaktivierung	Alice	Telefon 1	Request (?)
2	Wählton	Telefon 1	Alice	Indication (?)
3	Wahl der Nummer	Alice	Telefon 1	Request
4	Klingeln	Telefon 2	Bob	Indication
5	Freizeichen	Telefon 1	Alice	Indication (?)
6	Gesprächsannahme	Bob	Telefon 2	Response
7	Ende Freizeichen / Gesprächsaufbau	Telefon 1	Alice	Confirmation
8	Gespräch	---	---	---
9	Gesprächsbeendigung	Alice	Telefon 1	Request
10	Belegtton	Telefon 2	Bob	Indication

Man sieht hier, dass sich die Realität oft nicht ganz an das Modell hält. Zeile 2 zum Beispiel: ist es hier eine Confirmation? Oder doch eine Indication? Eigentlich hätte doch was zur anderen Seite geschickt werden müssen. Hier ist es also eigentlich weder das eine noch das andere; Zeile 1 und 2 stellen eine lokale Interaktion dar, die gar nicht zum Modell passt. Eher passt hier noch die Bezeichnung Indication, da eine Confirmation nur durch einen Response ausgelöst werden kann.

Auch Zeile 5 sieht seltsam aus: wieso bekomme ich denn hier eine Indication zurück? Müsste es nicht eine Confirmation sein? Die Antwort darauf ist definitiv nein, denn wir bekommen ja noch keine Erfolgsmeldung – die gibt es erst in Zeile 7. Zeile 5 ist im Modell nicht vorgesehen, tritt aber in der Praxis z.B. auch bei SIP (Session Initiation Protocol) auf, welches diesen Ablauf nachbildet. Bei SIP wird es als 'vorläufige Quittung' bezeichnet, etwas, was man mit dem Modell nicht machen kann. Daher hier als Krücke die Indication.

UND zu Zeile 9 und 10 könnte man noch erwähnen: was für ein Request/Indication isses denn? Verbindungsabbau (Dis) oder Abbruch (Abort)? Das ist wohl auch Interpretationssache.

2.b) Die Dienstenutzer sind die Apps auf beiden Seiten. Der Dienstbringer ist der WhatsApp-Dienst, der die Nachrichten überträgt (und dabei alles an Information absaugt, was möglich ist), mittels des Extensible Messaging and Presence Protocol (XMPP).

Als Dienstprimitive gibt es

- Absenden der Nachricht (request)
- Auslieferung der Nachricht (indication)

- Rückgabe der Empfangsquittung (response)
- Auslieferung der Empfangsquittung (confirmation)

Der Absender der Nachricht bekommt eine Empfangsquittung. Diese wird ihm durch einen Haken an der Nachricht dargestellt. Der Dienst ist also bestätigt.

Wie man eventuell aus eigener Erfahrung weiß, können sich die Haken blau färben: der Empfänger wird noch einmal eine Response geben, sobald der menschliche Nutzer die Nachricht gelesen hat (naja, eigentlich eher, sobald die Nachricht auf dem Display des Geräts dargestellt wurde, ob sie gelesen wurde, kann das Gerät ja nicht überprüfen). Dies ist wieder eine Inkonsistenz zum schönen Modell: es sind keine zwei Bestätigungen vorgesehen. Man könnte nun die Lesebestätigung als neuen Request auffassen, der auf Seiten des ursprünglichen Absenders eine Indication auslöst (blaue Haken). Aber ganz sauber gesehen ist es ja keine eigene Datenübertragung, sondern nur eine Zustandsmeldung, die sich auf die vorherige Datenübertragung bezieht. Also: die Rückgabe der Empfangsquittung als neuen Request zu bezeichnen, passt auch wieder nicht ganz. Außer wenn man vielleicht das 'Lesen' als eine Art eigenen Dienst auffasst, der von der vorherigen Auslieferung unabhängig ist. Oder wie auch immer. Hier versagt das Modell also auch wieder dabei, die Realität ganz genau abzubilden.

Auch noch erwähnenswert: während des Schreibens werden auch schon Statusberichte an den Empfänger gesendet. Das wären auch noch weitere Request/Indication-Aktionen, die vorab zur Übermittlung des Status dienen.

2.c) Der Dienstanutzer sind der E-Mail-Client des menschlichen Benutzers und der E-Mail-Server des Empfängers. Das ist nicht ganz korrekt: der E-Mail-Client würde die Daten zunächst seinem eigenen Server übermitteln und dieser stellt sie an den anderen Server zu. Auch könnte man sagen, dass die E-Mail dem empfangenden Benutzer zugestellt wird, aber das ist in der Praxis ein separater Schritt unter Verwendung eines anderen Dienstes.

Der Dienstbringer ist das E-Mail-System, mit SMTP als Protokoll.

Hier gibt es nur zwei Dienstprimitive:

- Absenden der E-Mail (request) und
- Auslieferung der E-Mail auf dem Ziel-Mail-Server (indication).

Der Dienst ist also unbestätigt.

Würde man den ganzen Weg bis hin zum Empfänger berücksichtigen, gäbe es wieder die Möglichkeit der Anforderung einer Lesebestätigung, man hat bei dieser Betrachtung also einen optional bestätigten Dienst.

So, jetzt haben wir mal wieder die ganze Aufgabe lang auf das Modell geschimpft. Ja und warum bitte schön ist es dann trotzdem in der Vorlesung drin? An sich sind diese Dienstprimitive und Diagramme ein gutes Hilfsmittel, sich die Funktionsweise von Kommunikationssystemen klar zu machen. Nur ist die Realität eben komplexer als das Modell, so dass sie auch Fälle beinhaltet, die das Modell nicht abbilden kann. Man sollte immer auch über die Modelle hinausdenken.

Aufgabe 1.3: Alternating Bit Protocol

Machen Sie sich noch einmal die Darstellung von Protokollinstanzen anhand von Zustandsübergangsdiagrammen (Automaten) klar. Betrachten Sie dazu eine vereinfachte Darstellung des Alternating-Bit-Protocol, die im Folgenden beschrieben ist:

Zwei Prozesse (Sender und Empfänger) tauschen Nachrichten über ein unzuverlässiges Medium aus. Bei der Übertragung über das Medium können Daten verloren gehen, eine Verfälschung von Daten findet allerdings nicht statt. Daher nutzen Sender und Empfänger zur Übertragung einen Dienst O , der auf das Medium aufgesetzt wird und eine zuverlässige Übertragung ermöglicht:

Die empfangende Instanz von O (bezeichnet mit E) bestätigt jede Nachricht der sendenden Instanz S , damit diese weiß, ob eine Nachricht erfolgreich zugestellt wurde oder ob sie verlorengegangen ist und neu übertragen werden muss. Um unterscheiden zu können, welche Bestätigung zu welcher Nachricht gehört, werden die Nachrichten abwechselnd mit den Bits 0 und 1 markiert, und diese Markierung wird auch in die Bestätigung aufgenommen. S kann erst die nächste Nachricht senden, wenn sie eine Bestätigung erhalten hat, dass die vorherige Nachricht erfolgreich übertragen wurde. Da Nachrichten verloren gehen können, wird das Senden einer Nachricht so lange wiederholt, bis klar ist, dass diese Nachricht auch angekommen ist.

Die beiden Prozesse verhalten sich wie folgt:

Sender: S erhält vom Sender die zu versendenden Daten (Aktion `accept`), versieht diese mit dem nächsten Markierungs-Bit (0 oder 1, beginnend mit 0) und schickt sie an den Empfänger (Aktion `send0`). Sobald eine Bestätigung der Nachricht eintrifft (`ack0`), kann er die nächste Nachricht des Senders entgegennehmen (`accept`). Diese wird mit Bit 1 markiert und durch `send1` versendet, usw. Wird eine bestimmte Zeit lang keine Bestätigung empfangen, wird die Nachricht mit der gleichen Nummer noch einmal wiederholt.

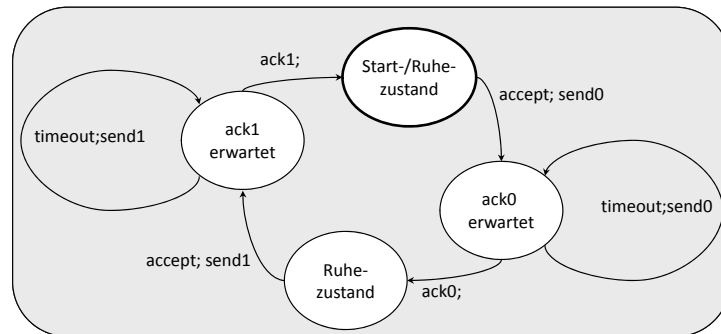
Empfänger: E empfängt eine Nachricht von S (mit Bit 0, `send0`), gibt diese Nachricht nach oben an den Empfänger weiter (`deliver`) und schickt die Bestätigung (`ack0`) zurück. Sobald eine neue Nachricht eintrifft (mit Bit 1, `send1`), kann diese wieder nach oben gegeben werden (`deliver`) und es wird ein `ack1` zurückgesendet, usw. Trifft eine Nachricht mit nicht erwarteter Nummer ein, wird die vorherige Bestätigung wiederholt.

Modellieren Sie S und E jeweils durch ein Zustandsübergangsdiagramm.

Lösung 1.3

Vereinfachte Darstellung des Alternating Bit Protocols, falls es mit dem Verständnis des Protokolls Probleme geben sollte – wir nutzen kein weiteres Medium, sondern betrachten nur den in der Vorlesung übersprungenen Schritt: wie kann ich denn das Zustandsübergangsdiagramm eines Dienstes aufsplitten in zwei Instanzen des Dienstes. Den Schritt zur Berücksichtigung einer weiteren Schicht behandeln wir hier nicht, das Ergebnis ist ja in der Vorlesung zu finden. Da es keine unterliegende Schicht gibt, können wir auch keine ICI zur Anzeige eines Übertragungsfehlers verwenden, drum werden in dieser Aufgabe Übertragungsfehler ausgeschlossen.

Sender als endlicher Automat



Empfänger als endlicher Automat

