Universität zu Köln
Historisch-Kulturwissenschaftliche Informationsverarbeitung
Basisinformationstechnologie II
Dr. Jan G. Wieners - jan.wieners@uni-koeln.de

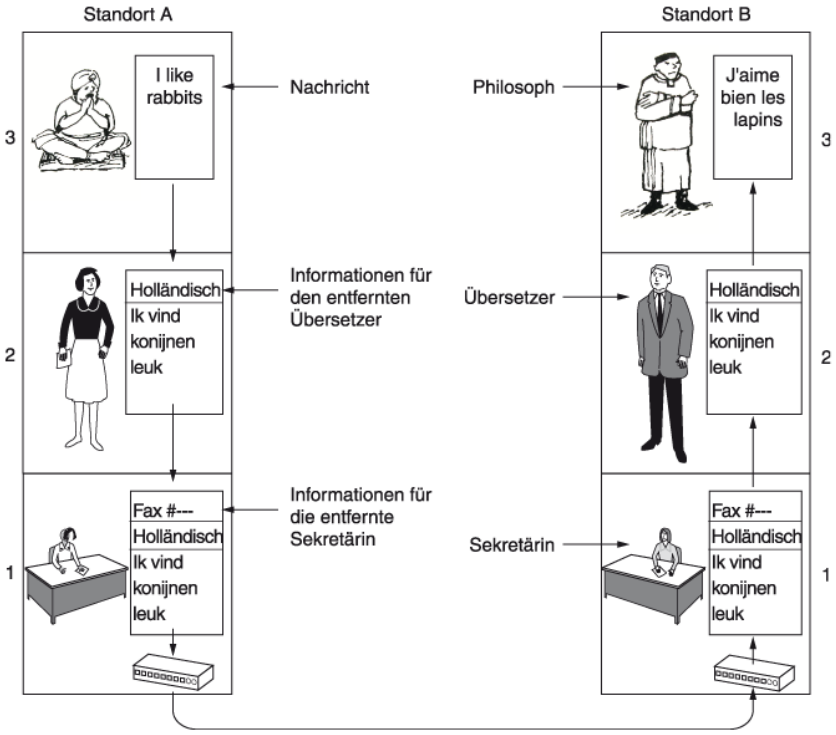
Internet-Infrastruktur: Von Rechnernetzen zum Internet

Schichtenmodelle / -Architektur

Um ihre Komplexität zu verringern, d.h. damit nicht jede Netzwerkarchitektur ihr eigenes Süppchen braut mit jeweils eigenen Standards, Verfahrensweisen, Ansichten, wie Details zu funktionieren haben, sind die meisten Netze als mehrere übereinander liegende **Schichten** oder **Ebenen** aufgebaut. Anzahl, Bezeichnung, Inhalt und Funktion der einzelnen Schichten unterscheiden sich von einem Netz zum anderen. In allen Netzen haben Schichten den Zweck, den jeweils höheren Schichten bestimmte **Dienste** (Services) zur Verfügung zu stellen, diese Schichten aber von Einzelheiten, wie die Dienste angeboten oder implementiert werden, abzuschirmen.

Zwischen je zwei angrenzenden Schichten befindet sich eine **Schnittstelle** (**Interface**). Die Schnittstelle definiert, welche Operationen und Dienste die untere der oberen Schicht anbietet.

Eine Menge von Schichten und Protokollen wird als Netzarchitektur bezeichnet. Die Spezifikation einer Architektur muss genügend Informationen enthalten, damit ein Softwareentwickler / eine Sotwareentwicklerin sein Programm oder seine Hardware so implementieren kann, dass dem vorgesehenen Protokoll Genüge getan wird. Eine Liste der Protokolle, die ein bestimmtes System verwenden kann, wird als **Protokollstapel** (Protocol Stack) bezeichnet.



Beispiel: Philosophen-/Philosophinnenmodell
Aus: Tanenbaum, Andrew S.: Computernetzwerke. 4., überarbeitete Auflage. Pearson Studium, München 2003.

Verbindungsorientierte und verbindungslose Dienste

Schichten können den jeweils übergeordneten Schichten zwei verschiedene Dienstarten zur Verfügung stellen: verbindungsorientierte und verbindungslose.

Ein **verbindungsorientierter** Dienst ist analog zum Telefonsystem konzipiert. Der Dienstnutzer baut eine Verbindung auf, benutzt die Verbindung und löst sie wieder. Die Kommunikation mit Hilfe eines verbindungsorientierten Dienstes lässt sich mit der Analogie zu einem Rohr veranschaulichen: Der Sender schiebt Objekte (Bits) an einem Ende hinein, der Empfänger entnimmt sie am anderen Ende in der gleichen Reihenfolge. In den meisten Fällen bleibt die Reihenfolge erhalten, so dass die Bits in der Reihenfolge eintreffen, in der sie gesendet wurden.

Ein **verbindungsloser** Dienst ist analog zum Postsystem konzipiert: Jede Nachricht (jeder Brief) trägt eine vollständige Adresse und wird unabhängig von allen anderen durch das System gesendet. Normalerweise kommt diejenige von zwei Nachrichten, die früher gesendet wurde, auch früher an. Zuweilen kann sich die erste aber auch verzögern, so dass die zweite vor der ersten ankommt.

Dienste können nach ihrer Dienstgüte klassifiziert werden. Dienste sind **zuverlässig**, wenn sie nie Daten verlieren. Ein zuverlässiger Dienst wird implementiert, indem der Empfänger den Erhalt jeder Nachricht bestätigen muss, so dass der Sender Gewissheit hat, dass sie angekommen ist. Dieser Bestätigungsprozess bedeutet zusätzliche Lasten und Verzögerungen, die sich oft lohnen, vielfach aber äußerst unerwünscht sind.

Dienstprimitive

Ein Dienst wird formal als eine Gruppe von Dienstprimitiven (Operationen) angegeben, mit denen ein Benutzerprozess auf den Dienst zugreifen kann. Diese Operationen weisen den Dienst an, eine Aktion auszuführen, oder berichten über eine Aktion, die von einer gleichgestellten Einheit ausgeführt wurde.

Beispiele Dienstprimitive:

LISTEN	Blockade, auf eingehende Nachricht wartend
CONNECT	Verbindung zu gleichgestellter Einheit aufbauen
RECEIVE	Blockade, auf eingehende Nachricht wartend
SEND	Nachricht an eine gleichgestellte Einheit senden
DISCONNET	Verbindung beenden

Beziehungen zwischen Diensten und Protokollen

Ein **Dienst** ist eine Gruppe von Dienstprimitiven (Operationen), die eine Schicht der über ihr liegenden Schicht zur Verfügung stellt. Der Dienst definiert, welche Operationen diese Schicht für ihre Benutzer ausführen kann. Ein Dienst fungiert als eine Schnittstelle zwischen zwei Schichten, wobei die untere Schicht der Dienstanbieter und die obere der Dienstnutzer ist.

Ein **Protokoll** ist eine Menge von Regeln, die das Format und die Bedeutung der von den gleichgestellten Einheiten innerhalb einer Schicht ausgetauschten Pakete oder Nachrichten festlegt. Mit Protokollen führen die Einheiten ihre definierten Dienste aus.

Dienste beziehen sich auf die Schnittstellen zwischen den Schichten; Protokolle beziehen sich auf die Pakete, die zwischen gleichgestellten Einheiten auf verschiedenen Rechnern versendet werden.

Das ISO-OSI-Referenzmodell

Das ISO-OSI (Open Systems Interconnection) behandelt die Verbindung offener Systeme, d.h. Systeme, die für die Kommunikation mit anderen Systemen offen sind, besteht aus sieben Schichten und ist selbst keine Netzarchitektur, d.h. ist sehr abstrakt, weil es die in den Schichten zu benutzenden Dienste und Protokolle nicht im Detail angibt. Das OSI-Modell sagt lediglich etwas darüber aus, welche Aufgaben die einzelnen Schichten ausführen sollen.

Das ISO-OSI-Modell besteht aus:

- **Bitübertragungsschicht** (Physical Layer): kümmert sich um die Übertragung von reinen Bits über einen Kommunikationskanal. Grundfragen:
  - Wie viel Volt sollen einer logischen 1 entsprechen?
  - Wie viele Nanosekunden soll ein Bit dauern?
  - Soll die Übertragung in beide Richtungen gleichzeitig erfolgen?
  - Wie viele Pins hat der Netzwerkstecker, wofür werden die Pins verwendet?
- **Sicherungsschicht** (Data Link Layer): Aufgabe, Übertragungsfehler zu mindern bzw. auszuschließen. Teilt den Bitstrom der physikalischen Schicht auf in Datenrahmen (Frames), innerhalb derer Fehlererkennung und Fehlerkorrektur möglich ist, z.B. anhand von Bitmustern, Prüfsummen, etc. Weitere Aufgabe der Sicherungsschicht ist die Sicherung gegen Datenüberschwemmung.
- **Vermittlungsschicht** (Network Layer): Steuert den Betrieb des Verbindungsnetzes. Wichtigste Designaufgabe ist hierbei u.a. die Auswahl der Paketrouten (Routing) vom Ursprungs- zum Bestimmungsort. Voraussetzung dabei: Die Endsysteme müssen eindeutig adressierbar sein. Beispiel: IP (Internet Protocol)
- **Transportschicht** (Transport Layer): Grundlegende Aufgabe, Daten von der Sitzungsschicht zu übernehmen, diese ggf. in kleinere Einheiten zu zerlegen, sie an die Vermittlungsschicht zu übergeben und sicherzustellen, dass alle Teile richtig am anderen Ende ankommen - freilich muss das alles hochperformant vonstattengehen.
- **Sitzungsschicht** (Session Layer): Ermöglicht es Benutzern an verschiedenen Rechnern, Sitzungen untereinander aufzubauen. Enthält darüber hinaus die Dialogsteuerung (Dialogue Control), die den Ablauf der Kommunikation organisiert. Grundfrage: Wer darf zum Zeitpunkt X Daten übertragen?
- **Darstellungsschicht** (Presentation Layer): Ermöglicht die Kommunikation zwischen Systemen mit unterschiedlicher lokaler Darstellung der Daten (z.B.: Speicherung eines int-Wertes in 16 o. 32 Bit). Hat die Syntax und Semantik der übertragenen Informationen zum Inhalt.
- **Anwendungsschicht** (Application Layer): Enthält eine Vielzahl häufig von Benutzern benötigter Protokolle (z.B. HTTP), stellt anwendungsspezifische Dienste bereit, z.B. für den Datentransfer (FTP) oder Email. Hier finden sich Applikationen und Protokolle, die sich nicht eindeutig den beiden darunterliegenden Schichten zuordnen lassen, z.B. WWW-Browser und Web-Server (z.B. Apache), das Protokoll HTTP, etc.

Das TCP/IP Referenzmodell

Intendiert für den Einsatz im ARPANET und Internet, wurde das TCP / IP Referenzmodell entworfen mit dem Gedanken im Hinterkopf, das komplette Netz nicht abhängig von einzelnen Elementen (Router, Gateways, etc.) zu halten, sondern autarkes Agieren (auch im Kriegsfalle) sicherzustellen.

Das TPC-IP Modell besteht aus:

- **Internetschicht** (Internet Layer): Sicherheitsnadel, die die gesamte Architektur zusammenhält. Aufgabe: Den Hosts ermöglichen, Pakete in jedes beliebige Netz einzuspeisen und unabhängig an das Ziel zu befördern. Die

Internetschicht definiert ein offizielles Paketformat und Protokoll: IP (Internet Protocol). Internetschicht hat die Aufgabe, IP-Pakete richtig zuzustellen. Ähnlich der OSI-Vermittlungsschicht.

- **Transportschicht** (Transport Layer): Soll Kommunikation ermöglichen. Hier wurden zwei Endpunkt-zu-Endpunkt-Übertragungsprotokolle definiert:
  - TCP: zuverlässiges, verbindungsorientiertes Protokoll, über das ein Bytestrom von einem Rechner im Internet fehlerfrei einem anderen Rechner zugestellt wird. Zerlegt den eingehenden Bytestrom in einzelne Nachrichten und leitet diese an die Internetschicht weiter. Am Ziel werden die Nachrichten vom empfangenden TCP-Prozess wieder zum Ausgabestrom zusammengesetzt. TCP verwaltet auch die Flusskontrolle, stellt somit sicher, dass ein langsamer Empfänger nicht von einem schnellen Sender mit Nachrichten überschwemmt wird.
  - UDP: User Datagram Protocol, Benutzerdatagramm-Protokoll. Unzuverlässiges, verbindungsloses Protokoll für Anwendungen, die anstelle der Sicherstellung der Reihenfolge oder der Flusskontrolle von TCP diese Aufgaben lieber selbst bereitstellen. Eingesetzt, wenn Schnelligkeit wichtiger ist als ihre Genauigkeit (z.B. bei der Übertragung von Sprache oder Video).
- **Anwendungsschicht**: Umfasst alle Protokolle der höheren Schichten. Telnet, FTP, SMTP.

Vergleich: OSI- und TCP/IP-Referenzmodell

Zahlreiche Gemeinsamkeiten: Beide Referenzmodelle basieren auf dem Konzept eines Stapels unabhängiger Protokolle. Das ISO-OSI-Modell basiert auf der expliziten Unterscheidung von drei Konzepten:

1. Dienste
2. Schnittstellen
3. Protokolle

Jede Schicht erbringt bestimmte Dienste für die jeweils darüber liegende Schicht. Die Definition eines Dienstes gibt an, was die Schicht macht, nicht wie die darüber liegenden Einheiten darauf zugreifen oder wie die Schicht funktioniert, sie definiert die Semantik der Schicht. Die Schnittstelle einer Schicht teilt den darüber liegenden Prozessen mit, wie sie darauf zugreifen können. Sie definiert die Parameter und welche Ergebnisse erwartet werden können, sagt jedoch nichts darüber aus, wie die Schicht intern funktioniert. Die Protokolle einer Schicht sind die Privatangelegenheit der betreffenden Schicht. Sie kann Protokolle nach eigenem Belieben benutzen, solange die Aufgabe ausgeführt werden kann (d.h. die angebotenen Dienste bereitgestellt werden können). Sie kann diese auch willkürlich ändern, ohne die Software in den höheren Schichten zu beeinflussen.

Das TCP/IP Modell unterschied ursprünglich nicht deutlich zwischen Dienst, Schnittstelle und Protokoll.

Das OSI-Referenzmodell wurde vor der Erfindung der entsprechenden Protokolle entwickelt. Das Modell ist folglich nicht auf eine bestimmte Menge von Protokollen ausgelegt, sondern sehr allgemein formuliert. Die Schattenseite: Auf Entwicklerseite unklar, welche Schicht mit welcher Funktionalität ausgestattet werden sollte.

Bei TCP/IP geschah das Gegenteil: Zuerst kamen die Protokolle, während das Modell eigentlich nur eine Beschreibung der vorhandenen Protokolle war. Mit der Einpassung der Protokolle in das Modell gab es keine Schwierigkeiten. Der einzige Ärger: Das Modell passte zu keinem anderen Protokollstapel. Folglich war es nicht besonders sinnvoll, damit TCP/IP-fremde Netze zu beschreiben.

Literatur

Teile der obigen Ausführungen sind dem (durchaus lesenswerten) Standardwerk über Computernetzwerke entnommen:
Tanenbaum, Andrew S.: Computernetzwerke. 4., überarbeitete Auflage. Pearson Studium, München 2003.