

Das soll sich im Kontext von „Industrie 4.0“ ändern. Bestellungen sollen automatisiert übertragen, Liefertermine bei Produktionsstörungen in Echtzeit aktualisiert, Produktionsprozesse bis hin zur Losgröße 1 flexibilisiert werden. Diese Anforderungen können nur auf einer soliden Datenbasis, die aus der Produktion – der eigenen ebenso wie der der Lieferanten – möglichst in Echtzeit geliefert wird, erfüllt werden. Damit gewinnt die Übertragung von Information aus der Ebene der Automatisierung bzw. der Produktion in die Ebene der Informationstechnik (IT) immer mehr an Bedeutung.

---

## 1.2 Aufbau des Buches

In diesem Buch werden im Sinne einer *angewandten* Netzwerktechnik besonders die höheren Schichten der Netzwerktechnik behandelt. Die einzelnen Kapitel stellen jeweils ein Thema vor und vertiefen es – soweit angebracht – am Beispiel eines Java-Programms.

In Kap. 2 werden das Thema Informationsdarstellung/Codierung und die Java-Stream-Klassen dargestellt. Diese Themen bilden den Kontext für die weiteren Kapitel und definieren die wichtigsten Begriffe.

Kap. 3 stellt Dateiformate im Allgemeinen vor und erklärt JSON als ein erstes, bedeutendes Format. HTML als weiteres, wichtiges Format wird in Kap. 4 dargestellt.

Für den Austausch der im Dateiformat beschriebenen Daten sind Protokolle nötig, die in den Kap. 5 bis 8 dargestellt werden. Die einzelnen Protokolle bauen teilweise aufeinander auf und werden von hohem zu niedrigem Abstraktionsgrad dargestellt: Kap. 5, HTTP, und Kap. 6, OPC UA, behandeln komplexe Protokolle, bei denen üblicherweise nur der Client implementiert wird, der sich mit einem Standardserver verbindet. TCP/IP, Kap. 7, und UDP/IP, Kap. 8, sind universelle Protokolle, bei denen im Allgemeinen „beide Seiten“, also Client und Server, implementiert werden. Kap. 9 stellt die zunehmend an Bedeutung gewinnenden Web Services dar. Dabei wird neben den populären

REST und SOAP Services auch ein kompletter Eigenbau vorgestellt.

Kap. 10 geht kurz auf weiterführende Themen ein, die im Buch nicht ausführlich behandelt werden.

Am Ende ausgewählter Kapitel stehen Übungsaufgaben, Lösungsvorschläge dazu finden sich in Anhang A.

Auf der Internetseite <http://angewnwt.hof-university.de/> finden sich die Quellcodes zu den Beispielen aus dem Buch, sowie die Datendateien, Wireshark-Mitschnitte und weiteres Zusatzmaterial.

---

## 1.3 Grundlagen der Kommunikation

Die Netzwerktechnik behandelt die Kommunikation zwischen Maschinen bzw. Rechnern. Diese Kommunikation findet zwischen mindestens zwei Partnern statt. Dazu müssen die Partner eine Verbindung aufbauen und danach Daten übertragen.

Eine populäre Anwendung für die Kommunikation mit mehreren Partnern ist beispielsweise das Live-Streamen eines Videos. Moderne Verfahren nehmen hierbei aber in Kauf, dass nicht alle Empfänger alle Daten erhalten.

### 1.3.1 Zeitlicher Ablauf der Kommunikation

Für Menschen ist der Ablauf der Kommunikation so klar, dass es ihnen oft schwerfällt, die vielen, ohne weiteres Nachdenken durchgeführten Schritte auf die Kommunikation zwischen Maschinen zu übertragen. Wir wollen uns deswegen im Folgenden am Beispiel eines Telefongesprächs die grundlegenden Mechanismen veranschaulichen.

1. Die beiden Teilnehmer, die kommunizieren wollen, arbeiten zunächst nicht synchron. Jeder von ihnen erledigt eine andere Aufgabe.

Einer der beiden Teilnehmer ist aber bereit, Anrufe anzunehmen. Programmtechnisch wird das anders realisiert als im ech-

ten Leben: Der Teilnehmer – für den Programmierer ist das der Server – wartet auf einen Anruf und tut sonst nichts.

Der zweite Teilnehmer ergreift irgendwann die Initiative und ruft den ersten Teilnehmer an. Programmtechnisch wird das genauso realisiert: Der zweite Teilnehmer – für den Programmierer ist das der Client – kommt in seinem Programmablauf zu dem Punkt, an dem er Daten übertragen oder empfangen will, und baut eine Verbindung zum ersten Teilnehmer auf.

2. Jetzt werden beide Teilnehmer synchronisiert: Der zweite wartet darauf, dass der erste abhebt. Der erste unterbricht seine Tätigkeit bzw. sein Warten und nimmt das Gespräch an. Sobald er das Gespräch angenommen hat, sind aus Netzwerksicht beide Teilnehmer verbunden und können mit der eigentlichen Kommunikation beginnen.

Bis hierher war es nötig, dass der zweite Teilnehmer eine „Adresse“ des ersten Teilnehmers hatte und der erste Teilnehmer unter dieser „Adresse“ auf einen Anruf gewartet hat.

3. Für die eigentliche Kommunikation ist es nun nötig, dass beide Teilnehmer dieselbe Sprache sprechen und sicherstellen, dass der jeweils hörende Teilnehmer auch in der Lage ist, die Botschaft des Sprechenden aufzunehmen. Damit könnte sich ein einfacher Dialog ergeben:

- a. Teilnehmer 1: „Hier spricht die Auskunft. Was kann ich für Sie tun?“

- b. Teilnehmer 2: „Hier ist der Franz.“

- c. Teilnehmer 2: „Geben Sie mir bitte die Telefonnummer von Rosi!“

- d. Teilnehmer 1: „32 16 8“

- e. Teilnehmer 2: „32 16 und wie bitte?“

- f. Teilnehmer 1: „32 16 8“

- g. Teilnehmer 2: „Danke! Auf Wiederhören.“

Technisch werden diese Aspekte in einem Kommunikationsprotokoll festgelegt. Mehr dazu in Abschn. [1.3.2](#).

4. Danach wird die Verbindung abgebaut, indem einer der beiden Teilnehmer auflegt. Der andere Teilnehmer erkennt, dass die Verbindung abgebaut wurde.

### 1.3.2 Das Prinzip eines Kommunikationsprotokolls

Im Beispiel aus Abschn. 1.3.1 verstecken sich viele Aspekte, die in einem Kommunikationsprotokoll geregelt werden.

Punkt 3a signalisiert den erfolgreichen Verbindungsaufbau und identifiziert Teilnehmer 1. Bei der Programmierung ist klar, dass die Verbindung steht, wenn das Abheben erkannt wurde. Dieser Schritt muss also nicht extra spezifiziert werden.

Punkt 3b identifiziert den Anrufer. Dies stellt eine einfache Authentifizierung dar. Auf dieser Basis könnten nicht berechtigte Teilnehmer abgewiesen werden.

Punkt 3c stellt eine Anfrage dar. In einem Kommunikationsprotokoll muss festgelegt werden, welche Anfragen zulässig sind und wie signalisiert wird, welche Anfrage gerade gestellt wird. Die Anfrage kann beispielsweise durch eine Zahl codiert werden: 0 – Nummer, 1 – Adresse, ... „Rosi“ ist ein Attribut der Anfrage und stellt klar, worauf sich diese bezieht. Auch hierfür muss das Protokoll festlegen, wie das Attribut übertragen wird (Zeichenkette, maximale Länge, ...).

Punkt 3d stellt die Antwort dar. Hierfür muss im Protokoll wieder festgelegt werden, welche Antworten bei welcher Anfrage möglich sind und wie sie codiert werden.

Die Punkte 3e und 3f stellen eine Fehlerbehandlung dar. Teilnehmer 2 hat offenbar erkannt, dass die Nummer nicht vollständig angekommen ist und fragt nochmal nach. Für diese Funktionalität ist es nötig, dass die Teilnehmer fehlerhafte Übertragungen erkennen können, indem beispielsweise Prüfdaten mit übertragen werden.

Teilnehmer 1 überträgt daraufhin in 3f die Antwort ganz oder teilweise nochmal.

Punkt 3g signalisiert das Ende der Kommunikation. Da das Auflegen ebenfalls das Ende der Verbindung signalisiert, ist es programmtechnisch nicht zwingend nötig, eine (höfliche) Beendigungsnachricht zu senden.

Neben den bisher beschriebenen logischen Aspekten muss ein Kommunikationsprotokoll noch viele weitere technische Aspekte behandeln.

Sehr wichtig ist die Festlegung, welche Daten in welcher Reihenfolge übertragen werden. In unserem Beispiel gibt es den Namen eines Teilnehmers und die Telefonnummer eines Teilnehmers. Wir sprechen hier von Attributen einer Teilnehmerinstanz.

Auch die Codierung der Nachrichten muss definiert werden, z. B. welches Symbol für die Ziffer ‚3‘ übertragen wird.

Weiterhin muss festgelegt werden, wie viel Information für ein Datum übertragen wird. Weder beim Attribut Name, im Beispiel mit dem Wert „Rosi“, noch beim Attribut Telefonnummer, im Beispiel „32 16 8“, kann von einer festen Länge ausgegangen werden. In solchen Fällen muss entweder vor dem Attribut die Länge des Attributes übertragen werden oder einzelne Attribute durch Trennzeichen wie Kommata, Semikola oder Zeilenvorschübe getrennt werden, damit die Gegenstelle die einzelnen Attribute im Datenstrom trennen kann.

---

## 1.4 Schichtmodelle

Die in Abschn. 1.3.2 angesprochenen Aspekte finden sich in jeder Anwendung, sind aber nur teilweise anwendungsspezifisch. Der reine Datentransport ist beispielsweise unabhängig von der Art der transportierten Daten, so dass ein und derselbe Transportmechanismus in verschiedenen Anwendungen verwendet werden kann.

Dieser Gedanke führt zu den in der Netzwerktechnik üblichen Schichtmodellen. Die einzelnen Aspekte werden auf verschiedene abstrakte Schichten abgebildet. Dabei kümmert sich jede Schicht nur um ihre Aufgabe, beispielsweise um die Übertragung eines Datenpakets und das Erkennen von Übertragungsfehlern.

Dazu ein Beispiel: Ein Firmenchef will einem anderen zum Geburtstag gratulieren. Er schreibt also „Alles Gute“ auf eine Karte und gibt sie mit dem Hinweis „für meinen guten Freund Franz“ seiner Sekretärin. Die steckt die Karte in einen Umschlag und ergänzt die korrekte Adresse. Der Umschlag kommt dann in die Poststelle und von dort zusammen mit allen gesammelten Sendungen zur Post. Diese befördert (unter anderem) den Umschlag ans Ziel. Die Poststelle dort bringt sie zur Chefsekretärin. Diese