Inhalt

[Auto Configuration bei IPv4 1](#_Toc73720486)

[Automatic Private IP Addressing (APIPA) 1](#_Toc73720487)

[Internetverbindung 1](#_Toc73720488)

[Service Discovery 1](#_Toc73720489)

[Trivial File Transfer Protocol (TFTP) 2](#_Toc73720490)

[TFTP Funktionen 2](#_Toc73720491)

[Simple Mail Transfer Protocol (SMTP) 3](#_Toc73720492)

[SMTP 822 Format 3](#_Toc73720493)

[SMTP Kommandos 3](#_Toc73720494)

[SMTP Bestätigungen 4](#_Toc73720495)

[MIME Standard (Multipurpose Internet Mail Extensions) 4](#_Toc73720496)

[Content Transfer Encoding 4](#_Toc73720497)

[Hypertext Transfer Protocol (HTTP) 5](#_Toc73720498)

[Funktionsweise 5](#_Toc73720499)

[HTTP-Request 5](#_Toc73720500)

[HTTP-Reply 5](#_Toc73720501)

# Auto Configuration bei IPv4

Durch APIPA können Clients auch ohne DHCP-Server (im Heimnetz) miteinander über eine IP-Adresse kommunizieren.

## Automatic Private IP Addressing (APIPA)

Wenn ein Client hochfährt sucht er zuerst ein DHCP-Server (Dynamic Host Configuration Protocol). Falls dies nicht funktioniert verwendet der Client APIPA um sich selbst eine IP und Subnetzmaske zuzuweisen.

Die IP-Adresse wird zwischen 169.254.1.0 und 169.254.244.255 gewählt (Klasse B). Die Subnetzmaske wird auf 255.255.0.0 gesetzt

Vor der Kommunikation muss der Client prüfen, ob die gewählte IP-Adresse schon verwendet wird. Falls dies zutrifft wird eine neue ausgewählt. Zudem wird alle 5 Minuten ein DHCP-Request gesendet und die Parameter von dort übernommen, sobald ein DHCP-Server gefunden wurde.

## Internetverbindung

Damit ein APIPA-Client eine Internet-Verbindung herstellen kann muss beim Router Proxy-ARP (Address Resolution Protocol) aktiviert sein. Der Router antwortet dann auf alle Requests, welche nicht aus dem lokalen Subnetz kommen mit der MAC-Adresse.

## Service Discovery

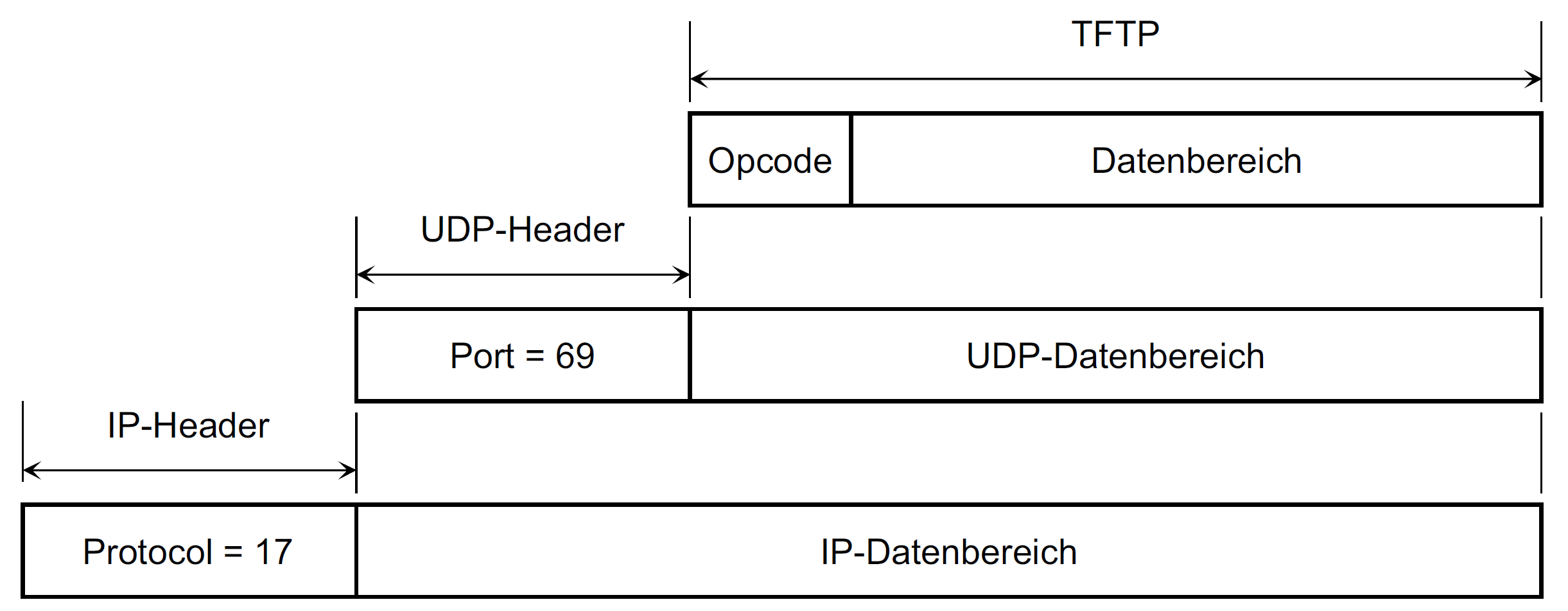
Um im Heimnetz auch andere Geräte (Bsp. Drucker) ohne DHCP-Server zu erreichen kann mDNS verwendet werden. mDNS ist ein lokaler Name Service ohne DNS-Server. Als Top Level Domain wird .local verwendet, Anfragen werden an die Multi-Cast-Adresse 244.0.0.251 gesendet, welche für mDNS reserviert ist, Verzeichniseinträge funktionieren gleich wie bei DNS, der Port ist 5353.

Services (wie Drucker) warden nun über Multicast gesucht und alle Geräte, welche Informationen dazu haben antworten ebenfalls über Multicast. So werden IP und Port des Geräts in Erfahrung gebracht und eine TCP-Verbindung kann aufgebaut werden.

# Trivial File Transfer Protocol (TFTP)

Protokoll zur Datenübertragung (wie FTP), es hat nur wenige Kommandos und einen geringeren Codeumfang. Da TFTP auf dem ungesicherten UDP aufbaut benötigt es eigene Sicherheitsmechanismen, um die fehlerfreie Übertragung zu gewährleisten.

Aufbau des TFTP-Pakets:

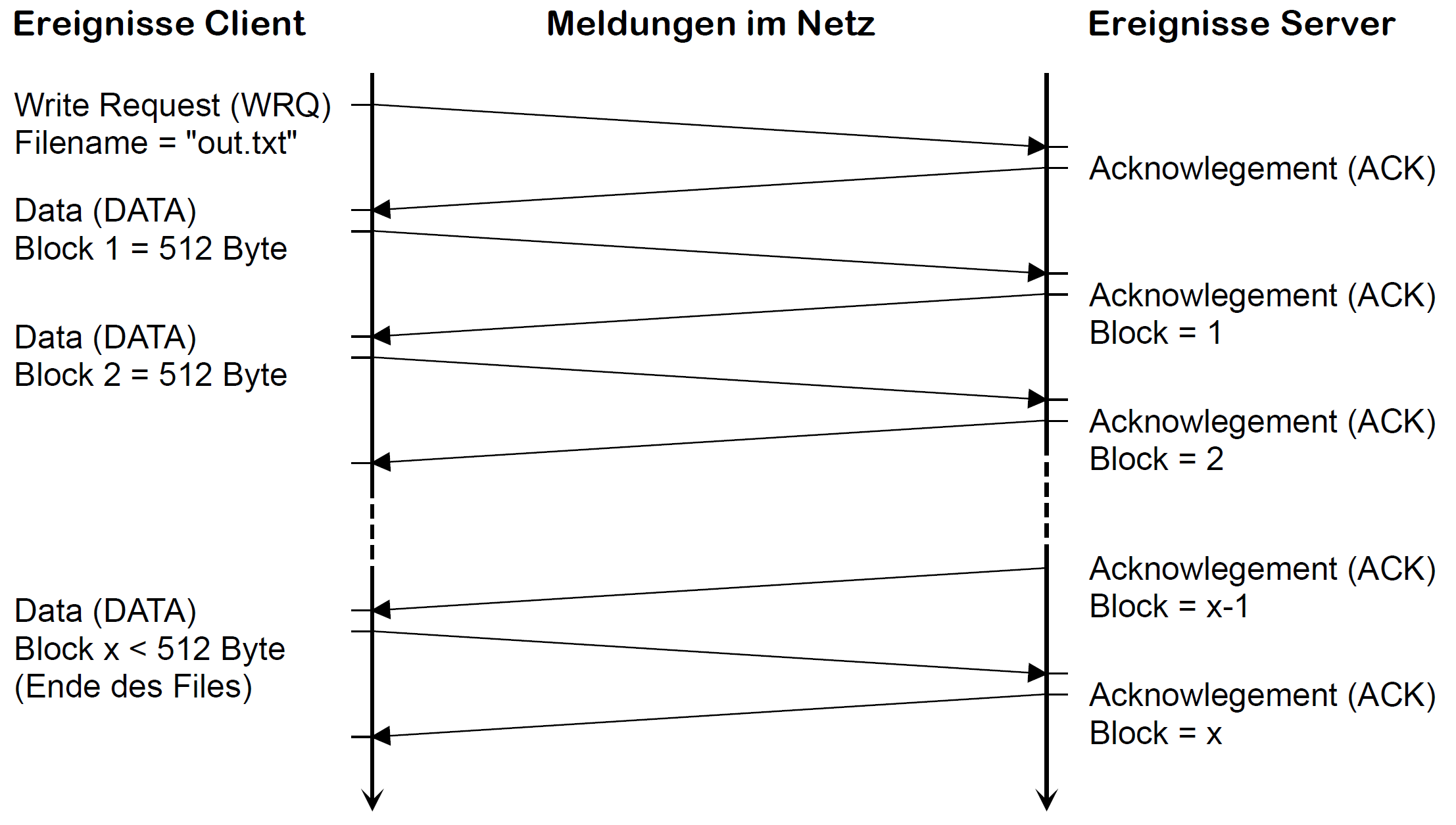


## TFTP Funktionen

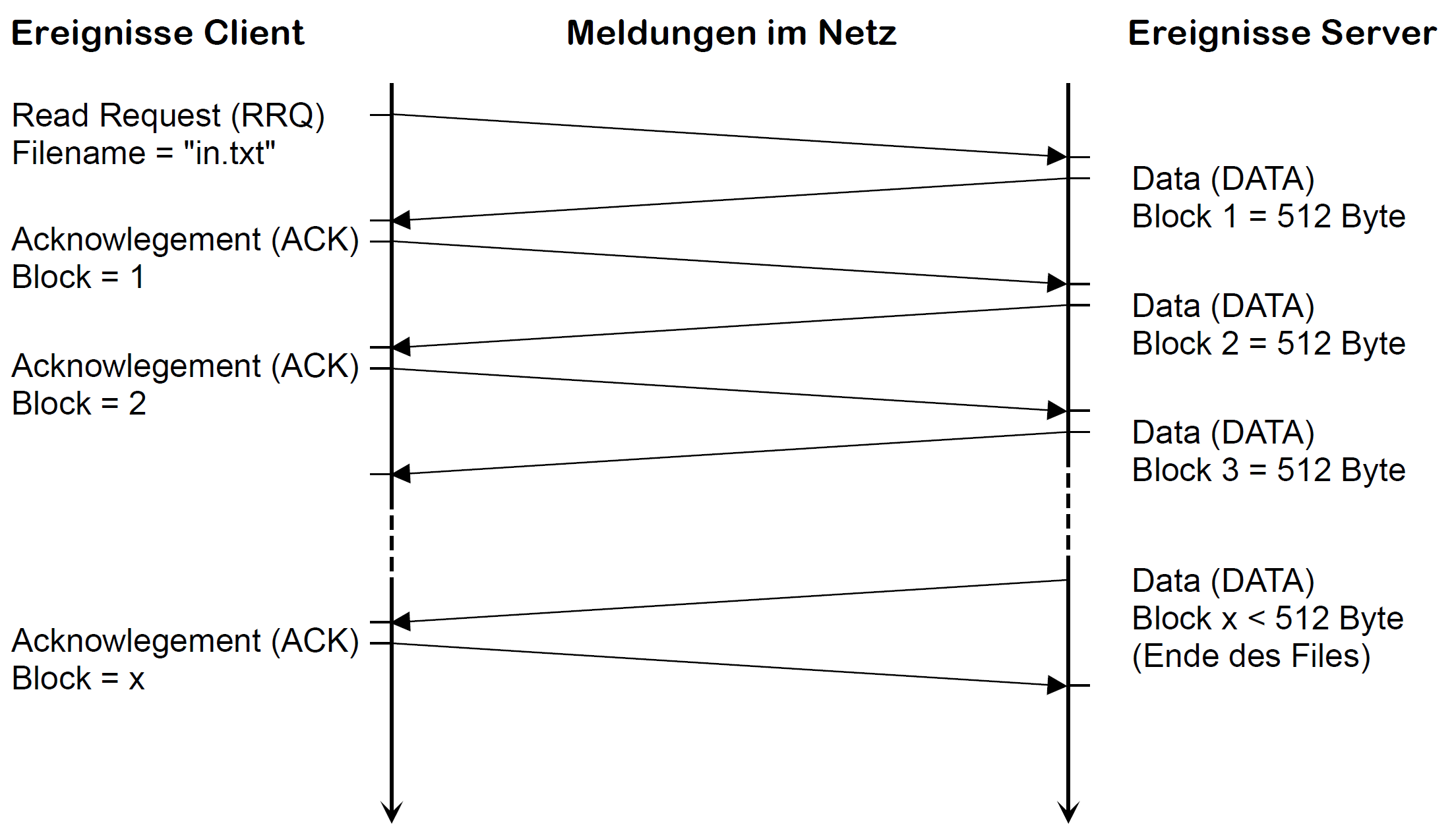
TFTP hat lediglich fünf Funktionen:

* Read-Request (RRQ) fordert eine Datei über einen TFTP-Server an.
* Write-Request (WRQ) schickt eine Datei an einen anderen Host.
* Acknowledgement (ACK) bestätigt ein korrekt empfangenes Paket.
* Data (DATA) übermittelt die Daten. Standard-Blockgrösse ist 512 Byte.
* Error (ERROR) beschreibt einen Fehler und beendet die Verbindung.

Beispiel für File senden:



Beispiel für File empfangen:



# Simple Mail Transfer Protocol (SMTP)

SMTP ist ein Standard-Protokoll zum Versenden und Weiterleiten von E-Mails. SMTP übernimmt lediglich den Transfer, zu anderen Aufgaben, wie der Zwischenspeicherung beim Senden / Weiterleiten und der Übermittlung an das SMTP System macht das Protokoll keine Angaben.

## SMTP 822 Format (RFC 822)

Die E-Mail besteht aus einem Header und einem Text aus ASCII-Zeichen. Diese sind durch eine Leerzeile getrennt und der Text kann beliebig lang sein.

Verbindungsaufbau, Übertragung und Verbindungsabbau erfolgt in folgenden Schritten:

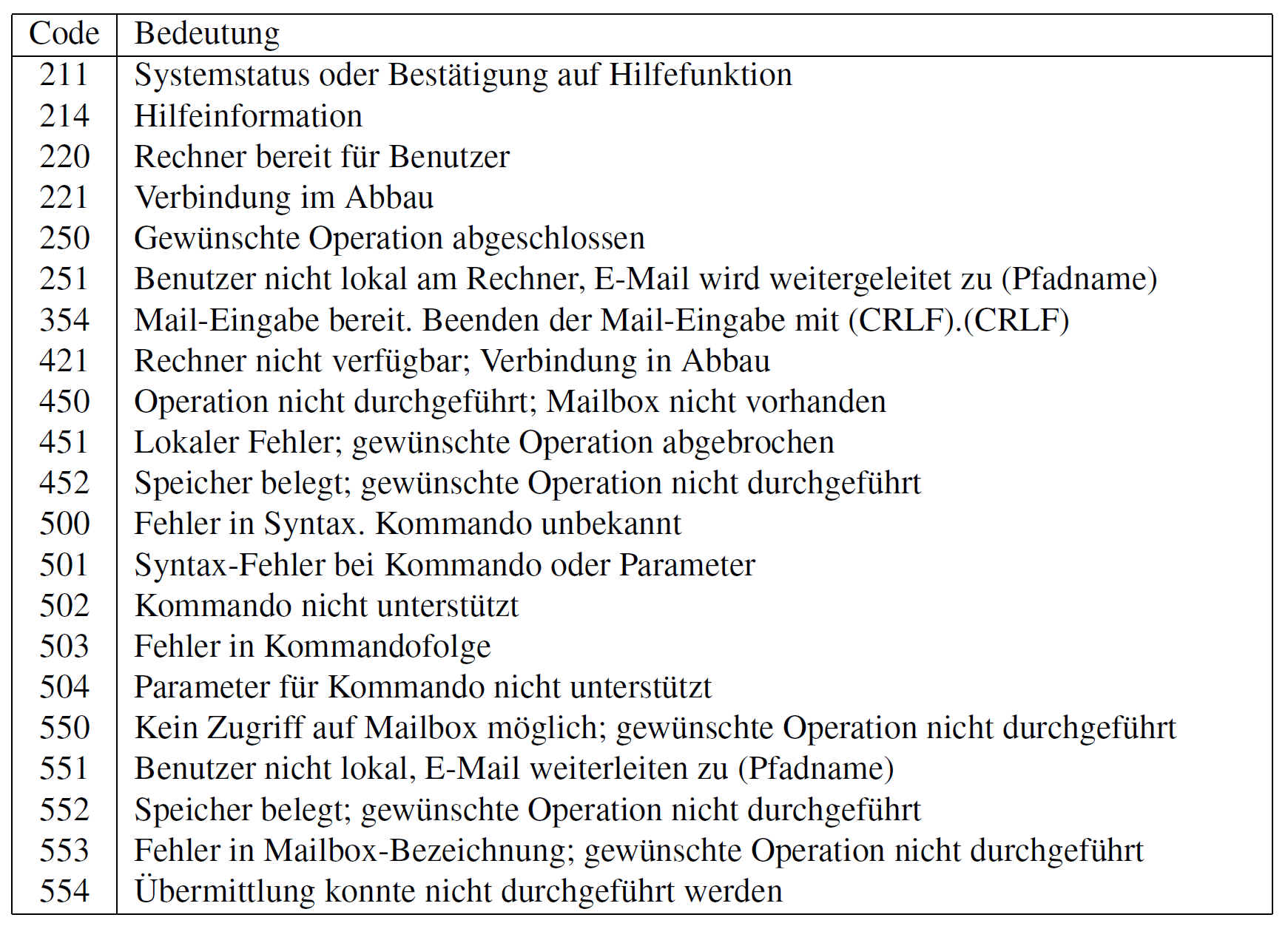
* Sender eröffnet eine TCP Verbindung zum Empfänger
* Empfänger identifiziert sich
* Sender identifiziert sich
* Empfänger akzeptiert die Identifikation des Senders
* Absender weist sich durch ein Mail-Kommando aus
* Durch eines oder mehrere RCPT (Recipient) Kommandos werden die Empfänger definiert
* Durch das Data Kommando wird der Text übermittelt
* Quit Kommando
* Bestätigung des Quit Kommandos
* Abbau der Verbindung

## SMTP Kommandos

* **Hello** meldet den Client beim Server an
* **Mail** identifiziert den Absender
* **Recipient** definiert die Empfängeradresse
* **Data** bestimmt den Nachrichtentext
* **Reset** bricht die Übertragung ab
* **Send** sendet die E-Mail direkt an ein Terminal
* **Verify** überprüft die Benutzernamen
* **Expand** ermöglicht die Abfrage von Mail-Listen (Liste von Benutzernamen und dazugehörige Adressen)
* **Help** gibt Informationen zur Bedienung zurück
* **Quit** beendet die SMTP Anwendung
* **Turn** tausch Sender und Empfänger

## SMTP Bestätigungen

Bestehen aus dreistelligen Zahlenfolgen, auf welche Text folgt. Auf jedes SMTP Kommando muss mindestens eine Bestätigung folgen.



## MIME Standard (Multipurpose Internet Mail Extensions)

Ermöglicht dem Sender und Empfänger der Mail eine Kodierung zu wählen, damit auch nicht-ASCII-Zeichen versendet werden können (weitere Sonderzeichen, Bilder, …). MIME fügt zwei Zeilen in den Header ein. Die Erste, um zu deklarieren, dass MIME verwendet wurde und die Zweite, um festzulegen, wie die MIME Daten in den Rumpf eingebunden werden.

Die Vorteile von MIME sind die Flexibilität selbst einen Standard bestimmen zu können und dass es mit älteren Systemen kompatibel ist, welche MIME nicht kennen, da Sie den Text weiterhin interpretieren können.

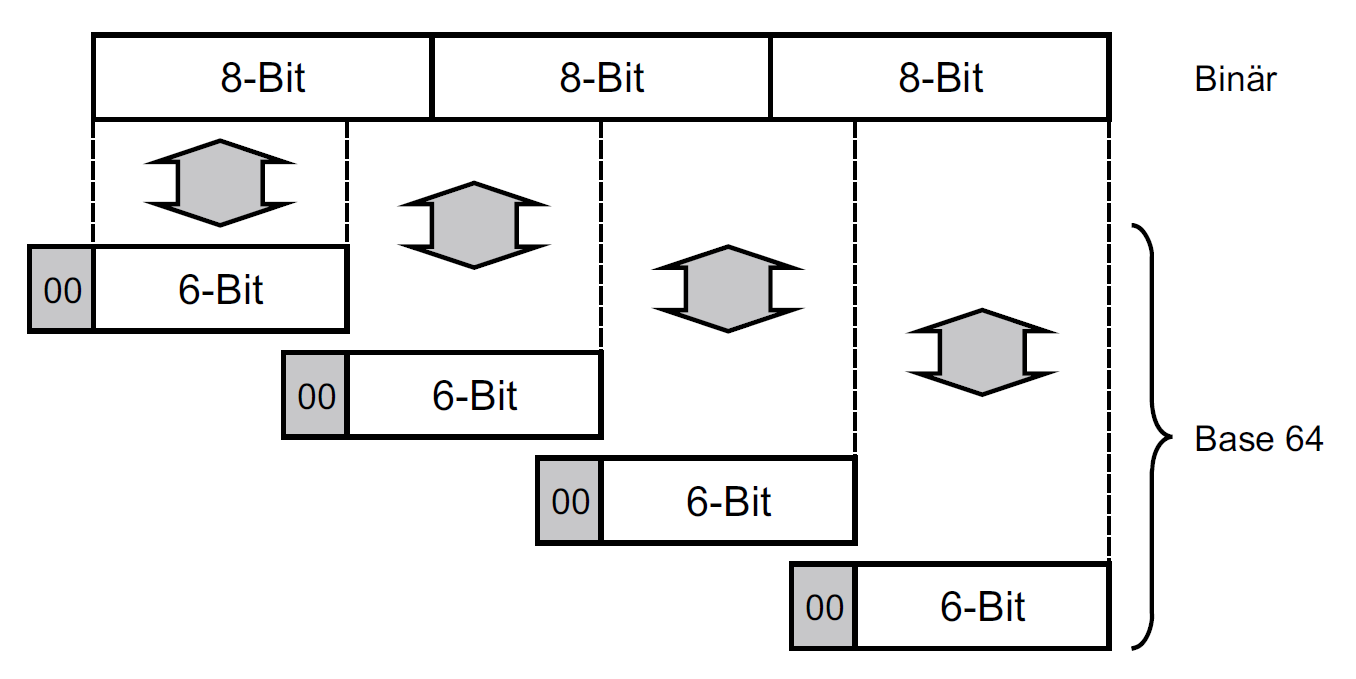
## Content Transfer Encoding

Die MIME-Typen können zusätzlich um folgenden Codierungsarten erweitert werden:

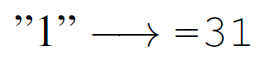
* B-Encoding (Base 64)
* Q-Encoding (Quoted Printable)
* Keine Codierung (binäre Übertragung)

**Base 64**

Jeweils 3 binäre Bytes werden in 4 Bitgruppen mit jeweils 6 Bit aufgeteilt. Für rein binäre Dateien wie Bilder oder Programme besser als Quoted Printable geeignet.



**Quoted Printable**

Jeder Wert wird durch 3 ASCII Zeichen dargestellt, welche jeweils mit einem = beginnen und haben danach 2 Hex-Ziffern. Beispiele:   

Eignet sich gut für Texte, welche gelegentlich weitere Sonderzeichen wir Umlaute enthalten. Eignet sich weniger gut für grosse Datenmengen und rein binäre Daten, da es eine schlechtere Effizienz hat (30%).

# Hypertext Transfer Protocol (HTTP)

Ist für die Übertragung von Ressourcen (HTML-Dokumente, Bilder, …), welche durch eine URL (Unique Resource Locater) identifiziert werden. Die erste Version erschien 1989.

## Funktionsweise

Ist wie SMTP ein ASCII-Protokoll, es werden ebenfalls MIME-Typen und Status Codes verwendet. HTTP ist dabei verbindungsorientiert (durch gesicherte TCP-Verbindung) und zustandlos (keine Informationen über die Verbindung werden zwischen zwei Transaktionen gespeichert).

Der wartende Server (meistens Port 80) nimmt den Request entgegen und sendet bei erfolgreicher Bearbeitung eine Antwort, ansonsten eine Fehlermeldung. Für eine HTML-Seite werden üblicherweise mehrere (separate) Transaktionen benötigt.

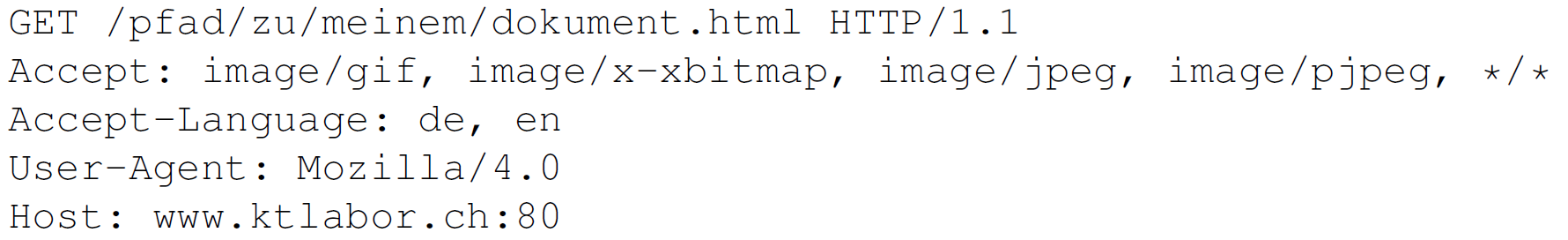
Einige Client-Request Methoden sind:

* GET, fordert eine bestimmte URL an.
* POST, schickt Daten zum Abspeichern an den Server.
* HEAD, wie GET aber nur die Header-Informationen kommen zurück.
* DELETE, löscht die Daten einer bestimmten URL.
* OPTIONS, HTTP-Server liefert Informationen über die verfügbaren Optionen.

## HTTP-Request

Die häufigsten Requests sind GET und PUT

Typischer GET-Request:



## HTTP-Reply

Der Sever muss sobald er den Request erhalten hat die gewünschten Ressourcen finden und diese zurückschicken.

Das Folgende ist ein Beispiel einer Antwort auf einen GET-Request, welcher ein HTML-Dokument anfordert: (vgl. oben)

