



openHPI-Kurs: Wie funktioniert das Internet?

TCP Ports

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

TCP Ports (1/4)

- Um TCP-Verbindung aufrechterhalten zu können, müssen Sender und Empfänger jeweils Endpunkte – **Sockets** – für die Kommunikation einrichten
- Jeder Socket verfügt über reservierten Speicherplatz zum Puffern der empfangenen oder zu sendenden Daten
- Zuordnung der Sockets ist auf beide Kommunikationspartner und Zeitdauer der jeweiligen TCP-Verbindung beschränkt
- Jeder Socket besitzt **Socket-Nummer**, die sich aus
 - IP-Adresse des Rechners und
 - lokal zuordenbarer 16-Bit-Nummer – **Port** – ergibt
- Ports sind **Service Access Points** der Transportschicht

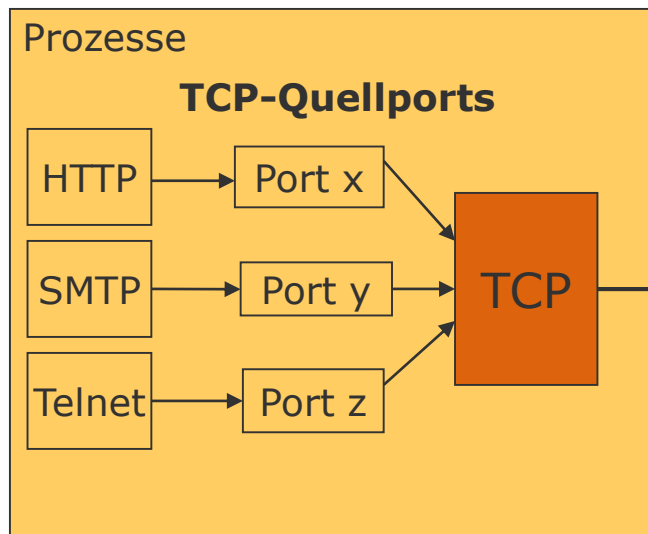
TCP Ports (2/4)

- Ports mit Nummern
 - 0 – 1.023 – **Well Known Ports** – sind weltweit eindeutig für Standarddienste reserviert, z.B.
 - 80 – HTTP, 23 – Telnet, 53 – DNS, ...
 - 1.024 – 65.535 können beliebigen Anwendungsprogrammen zugeordnet werden
- Man unterscheidet:
 - **Reservierte Ports:**
 - 0 – 255 für TCP/IP-Anwendungen
 - 256 – 1.023 für bestimmte UNIX-Anwendungen
 - **Registrierte Ports:**
 - 1.024 – 49.151 von IANA registriert
 - **Private, dynamische Ports:** Rest

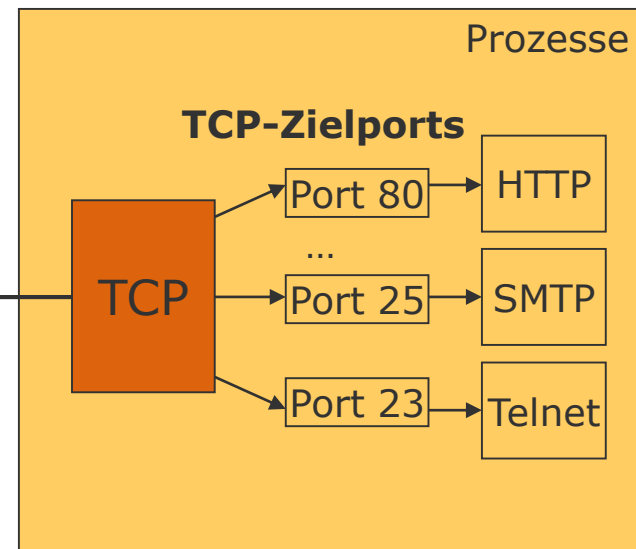
TCP Ports (3/4)

TCP-Verbindungen zwischen verschiedenen Anwendungsprozessen verlaufen über TCP-Ports

Rechner A



Rechner B



TCP - Transport

TCP Ports (4/4)

- Portnummern sind zu wählen beim Verbindungsaufbau zwischen zwei Anwendungsprogrammen:
 - **Zielpport:** Portnummer der gewünschten Anwendung
 - **Quellport:** Nicht reservierte und zur Zeit nicht anders genutzte Portnummer
- TCP-Verbindung kann anhand der beiden Socket-Nummern über das Tupel (Socket Sender, Socket Empfänger) eindeutig identifiziert werden
- Ein Server-Socket kann gleichzeitig mehreren Verbindungen dienen
- Anwendungsprogramme nutzen den TCP-Transportdienst über sogenannte **TCP-Primitive** – Primitives, z.B. Request, Response, Confirm, Read, Write, ..., die über den Socket implementiert sind