

Verteilte Systeme

Oktober - November 2023

3. Vorlesung – 25.10.2023

Kurs: TINF21AI1

Dozent: Tobias Schmitt, M.Eng.

Kontakt: d228143@
student.dhbw-mannheim.de

Wiederholungsfragen

- Bei einer Client-Server-Server-Struktur, welche Aufgaben kann der mittlere Server übernehmen?
- Was ist der Unterschied zwischen einem zustandslosen und einem zustandsorientiertem Server?
- Wo können Threads verwaltet werden?
- Was ist Virtualisierung?
- Welche Vorteile bringt Virtualisierung in verteilten Systemen?
- Welche Vorteile bringt Codemigration in verteilten Systemen?
- Was bedeutet schwache und starke Codemigration?
- Was müssen Sie bei der Umsetzung von Prozessmigration bedenken?

Themenüberblick

.Kommunikation

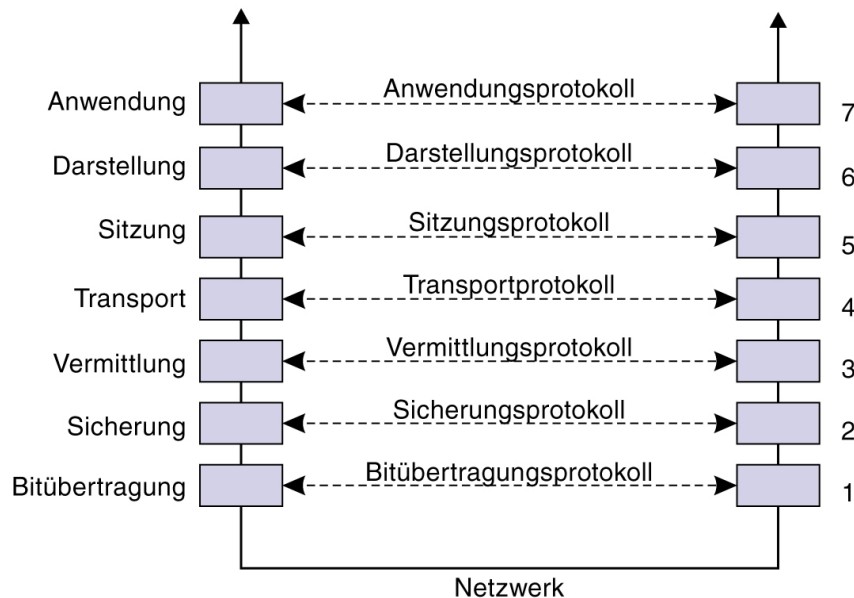
Kommunikation - Fragen

- Ist das IP-Protokoll verbindungslos oder verbindungsorientiert?
- Ist das TCP-Protokoll verbindungslos oder verbindungsorientiert?
- Welches sind die beiden am wenigsten verwendeten Schichten im OSI-Referenzmodell?
- In welche Schicht gehören z.B. allgemeine Authentifizierungsprotokolle?

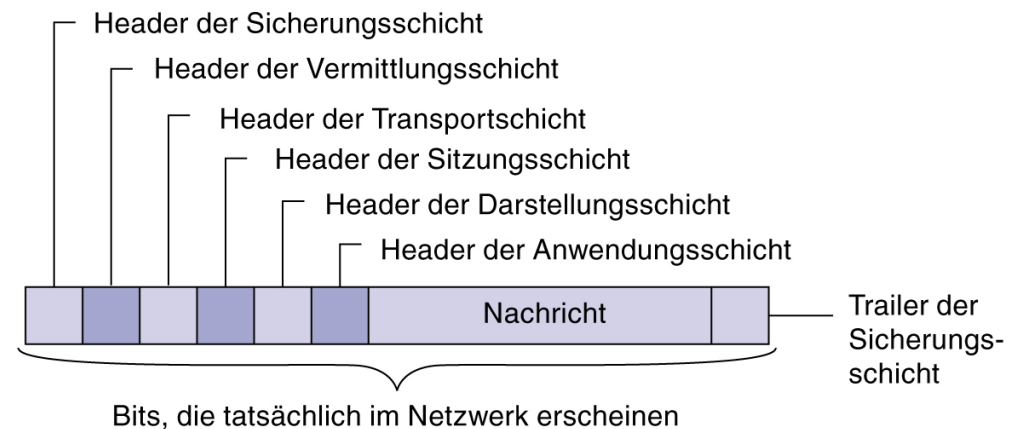
Kommunikation

Grundlagen

- OSI-Modell (Open Systems Interconnection Reference Model)



– Protokollstapel



Protokolle der unteren Schicht

•Bitübertragungsschicht

- Übermittlung von Bits
- Ausnutzung der Physik und entsprechender Analysemethoden (z.B. Fourier-Analyse)

•Sicherungsschicht

- Gruppierung der Bits in Rahmen
- Prüfsummenabgleich

•Vermittlungsschicht (Netzwerkschicht)

- Verantwortlich für Routing
- Weiterleitung von IP-Paketen
- (IP entspricht verbindungslosem Protokoll)

Transportprotokolle

.Transportschicht

- Nutzbarmachung des zugrundeliegenden Netzwerkes für Anwendungsentwickler
- Bereitstellung einer verlust-/fehlerlosen Verbindung
 - Fehlerkorrektur
 - (z.B. Hamming-Code
<https://www.youtube.com/watch?v=X8jsijhIII A>)
- Oberhalb eines verbindungsorientierten oder verbindungslosen Netzwerkdiensten
 - TCP – verbindungsorientiert
 - UDP – verbindungslos

Protokolle der höheren Schichten

•Sitzungsschicht

- z.B. für Dialogkontrolle oder für Synchronisation
- Konzept der Sitzung in Middleware-Lösungen, aber Protokolle der Sitzungsschicht nicht im Praxiseinsatz

•Darstellungsschicht

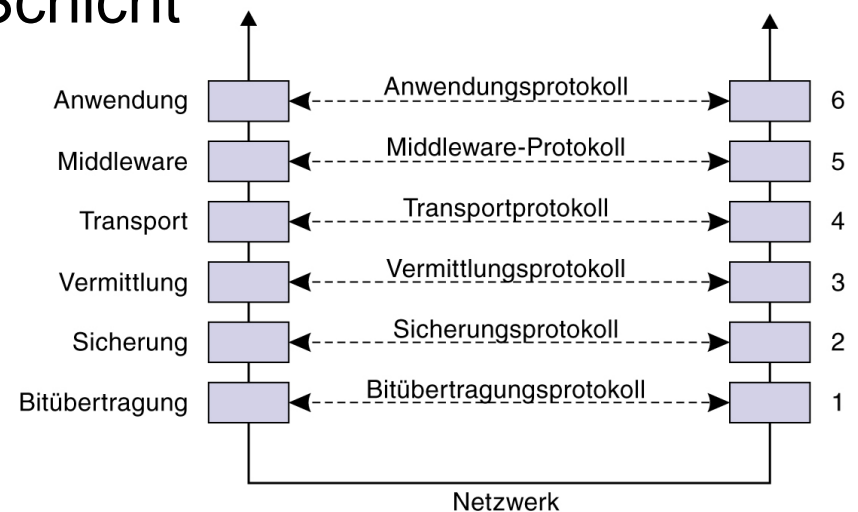
- Beschäftigungsfeld: Bedeutung der Bits
- Erleichterung der Kommunikation durch Definition bzw. Umwandlung von Formaten

•Anwendungsschicht

- Schnittstelle für Anwendungen
- z.B. HTTP, FTP, ...

Hinweise zum OSI-Modell

- Protokolle der Sitzungsschicht und Darstellungsschicht werden nicht verwendet
- Existenz weiterer Protokolle für allgemeine Zwecke, aber
 - Kein Transportprotokoll
 - Kein Anwendungsprotokoll
 - Kein Protokoll der Sitzungs- oder Darstellungsschicht
- Modifikation des OSI-Modells
 - Einführung einer Middleware-Schicht



Middleware-Protokolle

• Protokolle der Anwendungsschicht, aber für allgemeine Zwecke

• Beispiele

- Authentifizierungsprotokolle
- Commit-Protokolle (für die Ausführung von Transaktionen)
- Sperrprotokolle
- Kommunikationsprotokolle
 - z.B. für Zugriff auf entfernte Ressourcen
 - z.B. für Übertragung von Echtzeitdaten (Streams)

Arten der Kommunikation - Fragen

.Was verstehen Sie unter

- flüchtiger Kommunikation
- persistenter Kommunikation?

.Was verstehen Sie unter

- asynchroner Kommunikation
- synchroner Kommunikation?

.Was verstehen Sie unter

- diskreter Kommunikation
- fließender Kommunikation?

Arten der Kommunikation

.Persistente Kommunikation

- Middleware speichert Nachricht, bis sie beim Empfänger angekommen ist

.Flüchtige (transiente) Kommunikation

- Nachricht nur solange gespeichert, wie sendende und empfangende Anwendung in Ausführung

Arten der Kommunikation

•Asynchrone Kommunikation

- Sender fährt fort, nach Abgabe der Nachricht

•Synchrone Kommunikation

- Sender ist gesperrt bis Anforderung akzeptiert
- (Sperre bis Annahme durch Middleware oder Empfang oder nach vollständiger Bearbeitung der Anforderung)

Arten der Kommunikation

.Diskrete Kommunikation

- Jede Nachricht = vollständige Kommunikationseinheit

.Fließende (streaming) Kommunikation

- Senden vieler Nachrichten mit relevanter Reihenfolge

Kommunikation - Fragen

- Wie kann man entfernte Prozeduraufrufe gestalten?
- Welche Arten von flüchtiger Nachrichtenkommunikation können Sie sich vorstellen?
- Welche Art von persistenter Nachrichtenkommunikation kennen Sie?
- Welche Probleme kann es bei der Kommunikation zwischen mehreren Systemen / Anwendungen geben?

Themenüberblick

.Kommunikation

- **Entfernter Prozeduraufruf (Remote Procedure Call, RPC)**
- Nachrichtenorientierte Kommunikation
- Streamorientierte Kommunikation
- Multicast-Kommunikation

Remote Procedure Call (RPC)

•Ziel: Verbergen des expliziten Kommunikationsaustausches (Zugriffstransparenz)

- Prozeduren: *send* und *receive*

•Herkömmlicher Prozeduraufruf

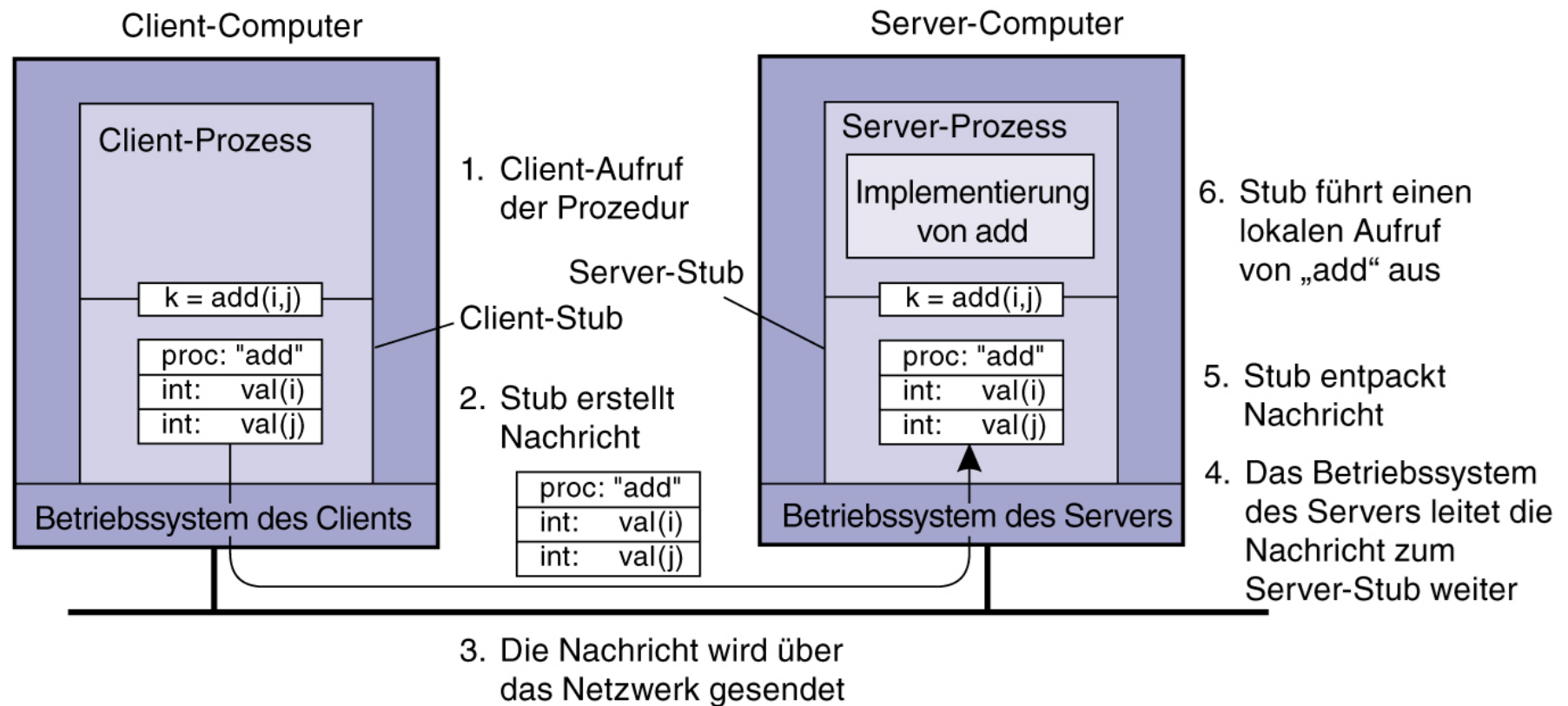
- Parameter werden auf Stack gelegt
- Aufruf einer Bibliotheksfunktion / Systemaufruf
- Bearbeitung und Rückgabe des Ergebnis
- Hinweis zum Parameterruf
 - Übergabe als Wert (Call-by-value)
 - Übergabe als Verweis (Call-by-reference)

Remote Procedure Call (RPC)

• Verbergen der Kommunikationsaustausches durch Arbeit mit Client- und Server-Stub

- Stub (engl. für Stummel, Stumpf) – kurzer relativ einfacher Programmcode

• Vorgehensweise:



Remote Procedure Call (RPC)

.Vorgehensweise

- 1) Client-Prozedur ruft Client-Stub auf
- 2) Client-Stub: Zusammenstellung der Nachricht, Aufruf des Betriebssystems (BS)
- 3) Client-BS sende Nachricht an Server-BS
- 4) Server-BS: Weitergabe der Nachricht an Server-Stub
- 5) Server-Stub: Entpackung der Parameter, Aufruf des Servers
- 6) Bearbeitung des Server, Rückgabe des Ergebnisses an Server-Stub
- 7) Server-Stub: Zusammenstellung der Nachricht, Aufruf des Server-BS
- 8) Server-BS sendet Nachricht an Client-BS
- 9) Client-BS: Weitergabe der Nachricht an Client-Stub
- 10) Client-Stub: Entpackung der Nachricht und Rückgabe an Client

Remote Procedure Call (RPC)

.Übergabe von Parametern als Wert

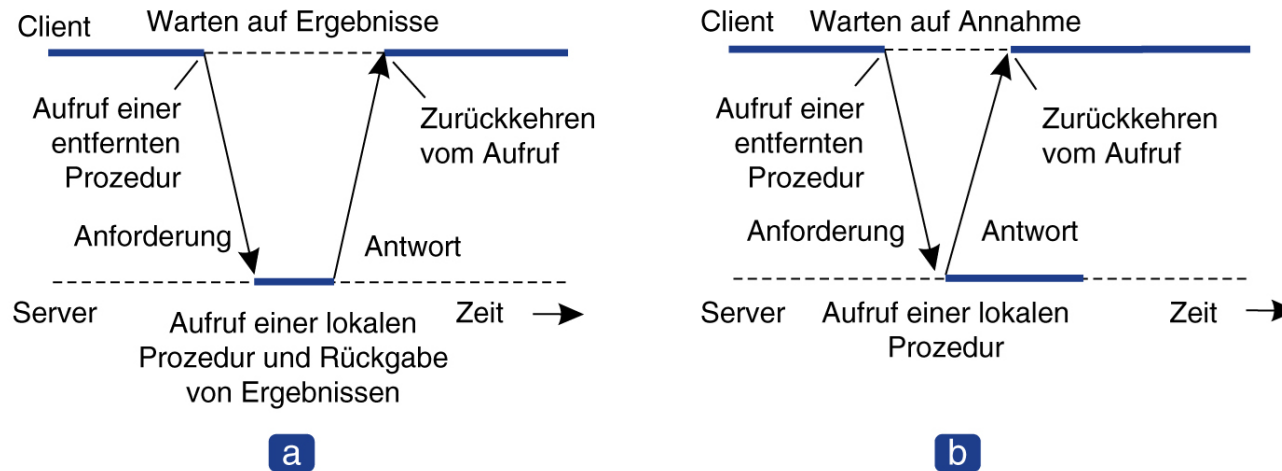
- Problem: Unterschiedliche Darstellungen von Zahlen, Zeichen etc. auf unterschiedlichen Rechnern
- Beispiel: Little Endian und Big Endian (unterschiedliche Reihenfolge bei Binärzahlen)
 - Little Endian – niedrigste Bit wird zuerst übertragen
 - Bsp.: $1110\ 0101_2 = 167_{10}$
 - Big Endian – höchstes Bit wird zuerst übertragen
 - Bsp.: $1010\ 0111_2 = 167_{10}$

.Übergabe von Parametern als Verweis

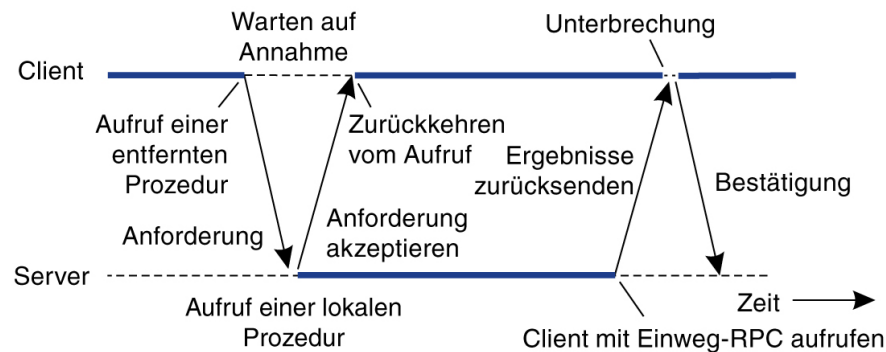
- Allgemein nicht möglich
- Aber: Wenn Feldgröße bekannt – Übergabe des Feldes
- Prinzip: Kopieren / Wiederherstellen

Remote Procedure Call (RPC)

.Arbeitsweisen von RPCs



(a) Die Wechselwirkung zwischen Client und Server in einem herkömmlichen RPC (b); die Wechselwirkung bei asynchronem RPC



Ein Client und ein Server kommunizieren über zwei asynchrone RPCs.

Themenüberblick

.Kommunikation

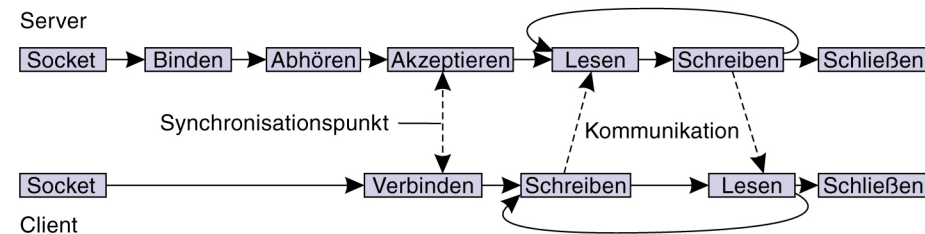
- Entfernter Prozeduraufruf (Remote Procedure Call, RPC)
- **Nachrichtenorientierte Kommunikation**
- Streamorientierte Kommunikation
- Multicast-Kommunikation

Nachrichtenorientierte Kommunikation

• Arten der nachrichtenorientierten Kommunikation

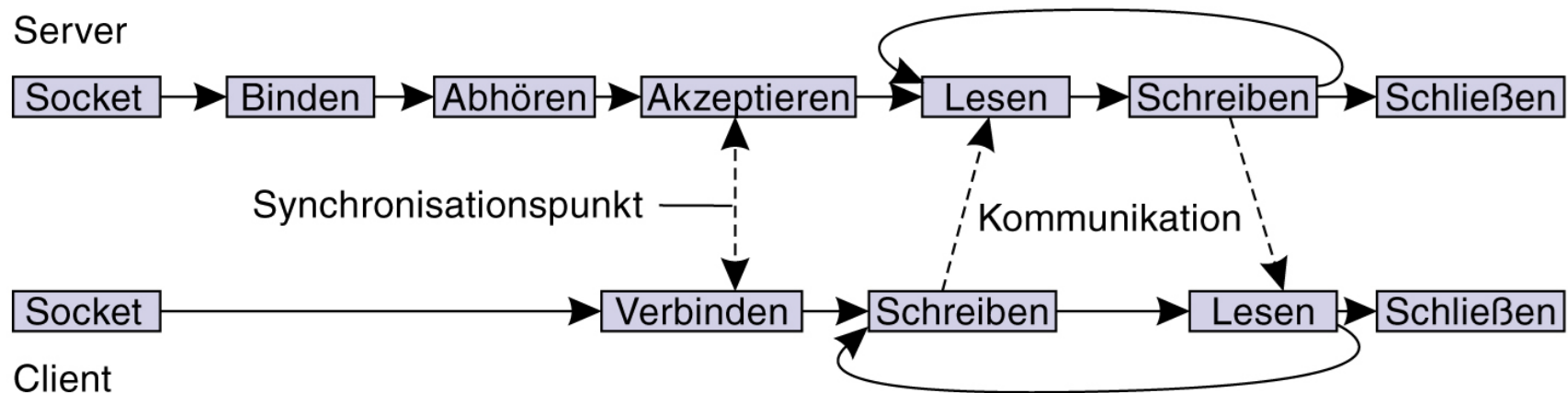
– **Flüchtige Kommunikation**

- Sender und Empfänger in Ausführung
- Kein Zwischenspeicherung
- Beispiel 1) Berkley Sockets
 - Schnittstelle oberhalb des Kommunikationsendpunktes seitens des Betriebssystems
 - Kommunikationsendpunkt für Anwendungsprogramme
 - Nutzung einer einfachen Menge von Primitiven
- Beispiel 2) MPI (Message-Passing-Interface)
 - Für parallele Anwendungen (für hochleistungsfähiger Multicomputer)
 - Annahme: Kommunikation zwischen bekannter Anzahl von Prozessen



Nachrichtenorientierte Kommunikation

.Flüchtige Kommunikation – Bsp. Sockets



Verbindungsorientierte Kommunikationsmuster unter Verwendung von Sockets

Nachrichtenorientierte Kommunikation

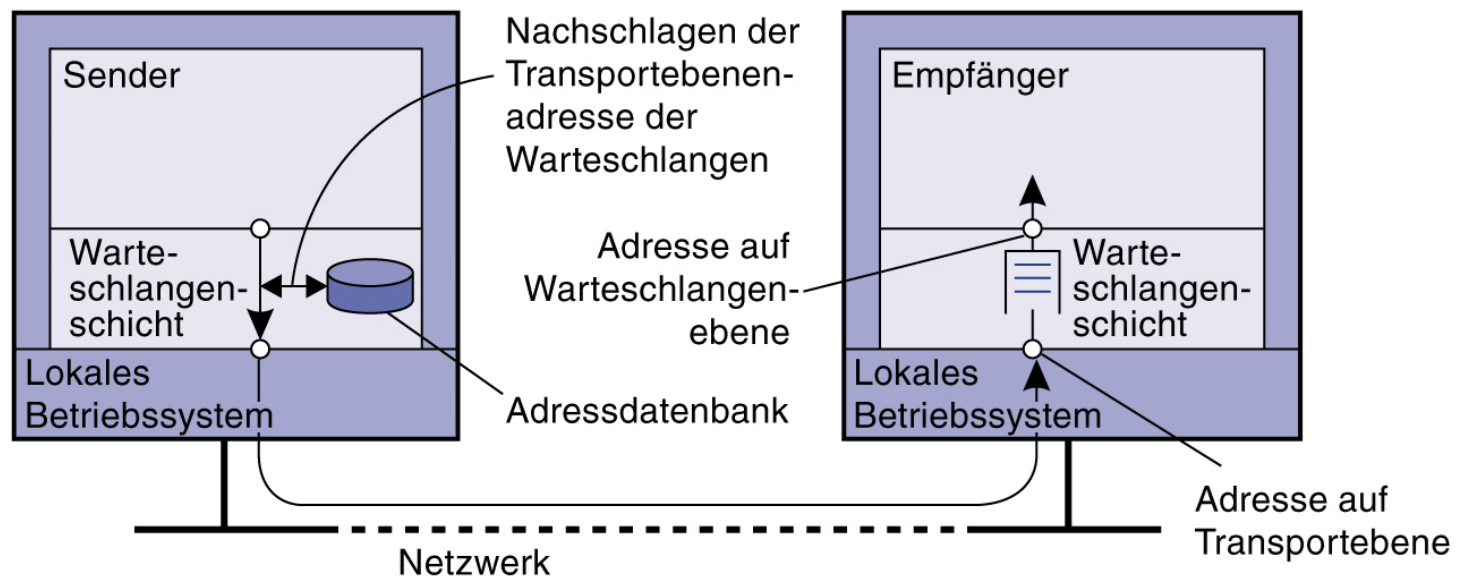
.Persistente Kommunikation

.Warteschlangensysteme (Message-Queuing Systems) bzw. nachrichtenorientierte Middleware (Message-Oriented Middleware - MOM)

- Zeitlich lose gekoppelte Kommunikation – Zwischenspeicherung der Nachrichten
 - Beispiel: E-Mail
- Grundkonzept
 - Nachricht in Warteschlange
 - Weiterleitung über Kommunikationsserver
 - Auslieferung
- Vorteile:
 - Empfänger kann beim Aussenden ausgeschaltet sein
 - Sender braucht bei Weiterleitung nicht mehr aktiv zu sein

Nachrichtenorientierte Kommunikation

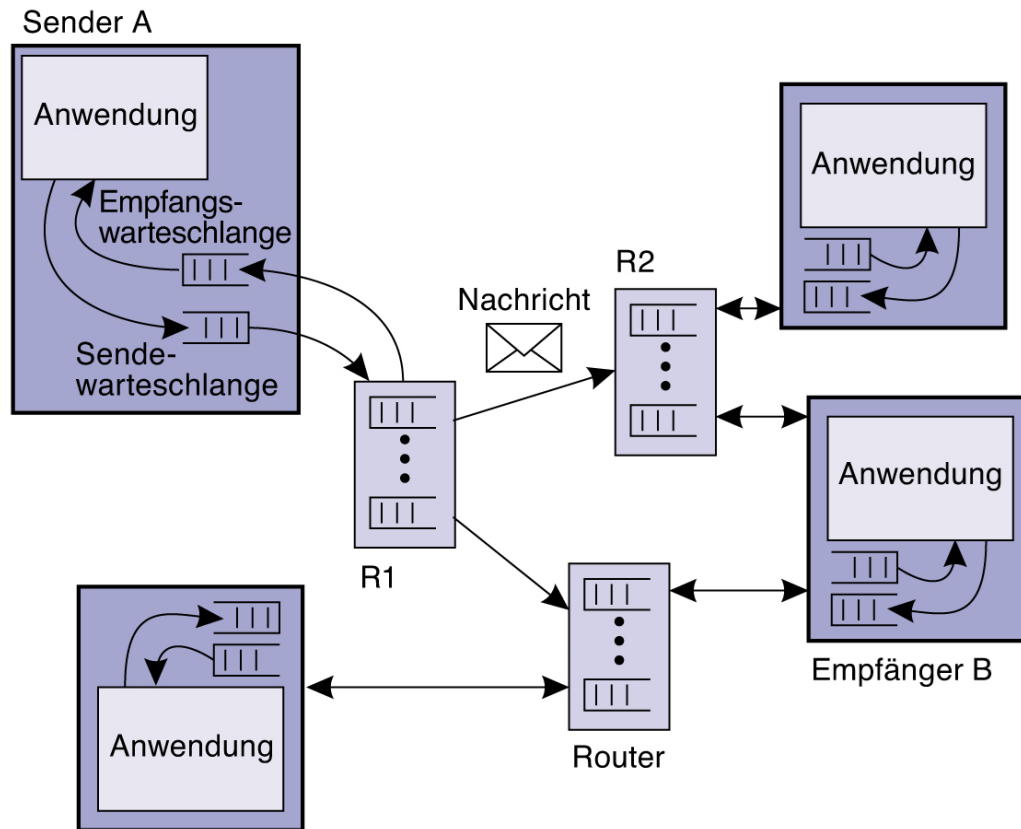
.Persistente Kommunikation - Warteschlangensystem



Die Beziehung zwischen der Adressierung auf Warteschlangen- und auf Netzwerkebene

Nachrichtenorientierte Kommunikation

.Persistente Kommunikation - Warteschlangensystem



Allgemeine Aufbau
eines Warteschlangensystems
mit Routern

Nachrichtenorientierte Kommunikation

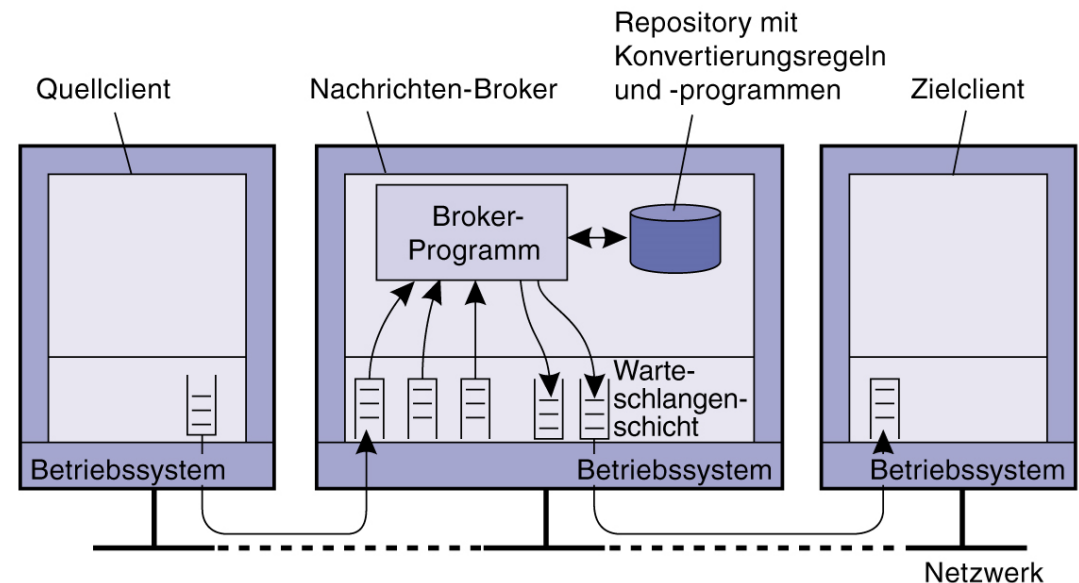
.Persistente Kommunikation – Warteschlangensystem

- Anwendungsbereich: Integration vorhandener und neuer Anwendungen in einzelnes, zusammenhängendes Informationssystem
- Problematik: Gleiches Format bei Sender und Empfänger
 - Ansatz 1: neue Anwendung = neues Format
 - Ansatz 2: Definition eines allgemeinen Nachrichtenformates
 - Ansatz 3: ...

Nachrichtenorientierte Kommunikation

.Persistente Kommunikation – Warteschlangensystem

- Ansatz 1 und 2 nicht praktikabel
- Ansatz 3: Unterschiedliche Formate akzeptieren, aber Nutzung eines Nachrichten-Brokers (Filter)
 - Umwandlung der Nachrichten
 - i.A. nicht Teil des
 - Warteschlangen-
 - systems



Themenüberblick

.Kommunikation

- Entfernter Prozeduraufruf (Remote Procedure Call, RPC)
- Nachrichtenorientierte Kommunikation
- **Interludium / Zwischenfragen**
- Streamorientierte Kommunikation
- Multicast-Kommunikation

Erinnerungsfragen

Kommunikationssysteme

• Was verstehen Sie unter

- Unicast?
- Anycast?
- Broadcast?
- Multicast?

• Was wissen Sie über die Dienstgüte in einem Netzwerk?

- Welche Parameter sind für die Dienstgüte wichtig?
- Welche Dienstgüteparameter sind für welche Dienste wichtig?

„Casting“ in Kommunikationssystemen

•Unicast

- Senden an ein Ziel

•Anycast

- Senden an ein Ziel aus einer Gruppe von Zielen (meist das nächstgelegene Ziel)

•Broadcast

- Senden an alle Ziele

•Multicast

- Senden an eine Gruppe von Zielen

Dienstgüte in Kommunikationssystemen

•Dienstgüte (Quality of Service)

- Bandbreite
 - Physikalisch-technische Obergrenze für den Durchsatz
- Übertragungsverzögerung
 - Zeit zwischen Versenden und Ankunft einer Nachricht
- Jitter
 - Schwankungen (Standardabweichung) bei einer Übertragungsverzögerung
- Verlustrate

Dienstgüte in Kommunikationssystemen

•Anforderungen an die Dienstgüte

Anwendung	Bandbreite	Übertragungsverzögerung	Jitter	Verlustrate
E-Mail	Niedrig	Niedrig	Niedrig	Mittel
Dateiaustausch	Hoch	Niedrig	Niedrig	Mittel
Webzugriff	Mittel	Mittel	Niedrig	Mittel
Entfernte Anmeldung	Niedrig	Mittel	Mittel	Mittel
Audio on Demand	Niedrig	Niedrig	Hoch	Niedrig
Video on Demand	Hoch	Niedrig	Hoch	Niedrig
Telefonie	Niedrig	Hoch	Hoch	Niedrig
Videokonferenzen	Hoch	Hoch	Hoch	Niedrig

Abbildung 5.27: Stringenz der Anforderungen für die Dienstgüte verschiedener Anwendungen.

Themenüberblick

.Kommunikation

- Entfernter Prozeduraufruf (Remote Procedure Call, RPC)
- Nachrichtenorientierte Kommunikation
- **Streamorientierte Kommunikation**
- Multicast-Kommunikation

Streamorientierte Kommunikation

Einstiegsfragen

.Was verstehen Sie unter diskreten und kontinuierlichen (Darstellungs-)Medien?

.Was ist ein (Daten)Stream? Was für Unterscheidungsmerkmale könnte es geben?

.Was gilt es alles bei Audio- und Videostrams zu beachten?

Streamorientierte Kommunikation

.Unterscheidung in

- Diskrete (Darstellungs-)Medien
 - Keine zeitliche Beziehung zwischen Dateneinheiten für die richtige Interpretation
 - Bsp.: (unbewegliche) Bilder, ausführbare Dateien, ...
- Kontinuierliche (Darstellungs-)Medien
 - Zeitliche Beziehung wichtig für richtige Interpretation
 - Bsp.: Audio und Video

Streamorientierte Kommunikation

.Datenstream = Folge von Dateneinheiten

- Anwendung auf diskrete oder kontinuierliche Datenstreams

.Unterscheidung der Datenstreams

- Asynchroner Übertragungsmodus
 - Ohne zeitliche Beschränkung, Übertragung nacheinander
 - Bsp.: Datei als Datenstream
- Synchroner Übertragungsmodus
 - Existenz einer maximalen Ende-zu-Ende-Verzögerung
 - (Prinzip: Schneller ist stets erlaubt.)
 - Bsp.: Datenübermittlung bei Sensorknoten
- Isochroner Übertragungsmodus
 - Ankunft zur rechten Zeit ist wichtig
 - Bedingung: minimale und maximale Verzögerung, beschränkter Jitter
 - Bsp.: Verteilte Multimediasysteme

Streamorientierte Kommunikation

.Unterteilung von Streams in

- Einfache Streams
 - Nur eine Datenfolge
- Komplexe Streams
 - Zusammensetzung aus mehreren Substreams
 - Zeitabhängigkeit auch zwischen Substreams
 - Bsp.: Stereo-Audio-Stream, Video-Stream, Untertitel

.Hinweis für folgende Betrachtung

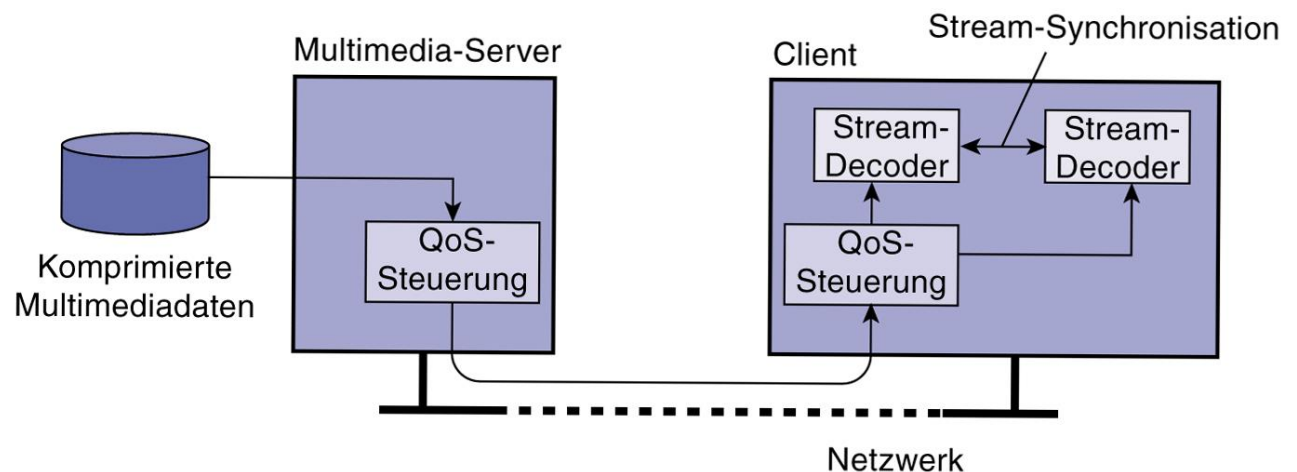
- Fokus auf Streaming gespeicherter Daten
- (Keine Betrachtung von Live-Streams)
- Aber Betrachtung: Qualität der Übertragung und Synchronisierung

Streams und Dienstgüte

- Anforderung an zeitliche Steuerung = Dienstgüte
 - Wichtig ist: Pünktlichkeit, Umfang, Zuverlässigkeit
- Eigenschaften für Dienstgüte:
 - Erforderliche Bit-Rate zur Übertragung
 - Maximale Verzögerung für Verbindungsaufbau
 - Maximale Ende-zu-Ende-Verzögerung
 - Maximale Verzögerungsvarianz (Jitter)
 - Maximale Umlaufverzögerung (round-trip delay)

Frage:

Wie lässt sich die
Dienstgüte steuern?



Streams und Dienstgüte

.Problematik:

- Verwendung des Internetprotokollstapels
- Kommunikationbasis: IP-Protokoll

.IP bietet aber etwas Unterstützung hinsichtlich der Dienstgüte (QoS = Quality of Service)

- IP: Existenz von differentierbaren Diensten
- Entsprechende Klassifizierung der Pakete nötig

Streams und Dienstgüte

(Exkurs Kommunikationssysteme)

•Fair Queuing

- Idee: Aufteilung des Datenflusses pro Ausgangsleitung (Warteschlangen)
- Einführung einer virtuellen Zeit
- Je Warteschlange
 - Bestimmung der Ankunftszeit und Berechnung der Beendigungszeit (ggf. Wichtung)
- Ausgabe nach Beendigungszeiten (frühestes Ende zuerst)
- Idee: Byte-Abfluss je Kanal gleich (bzw. abhängig von Wichtung)

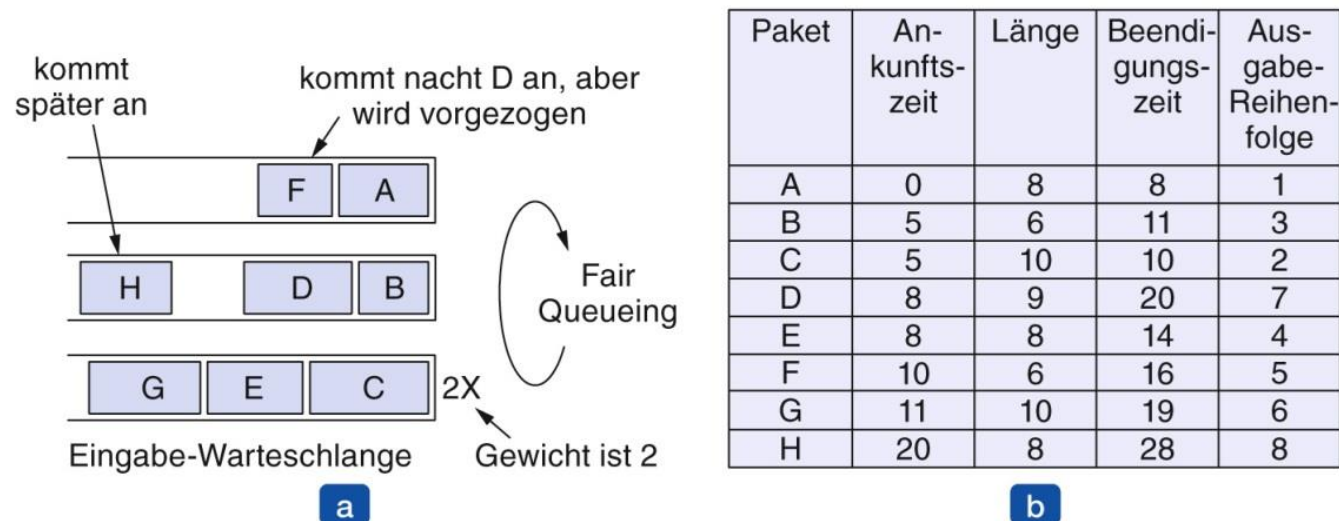
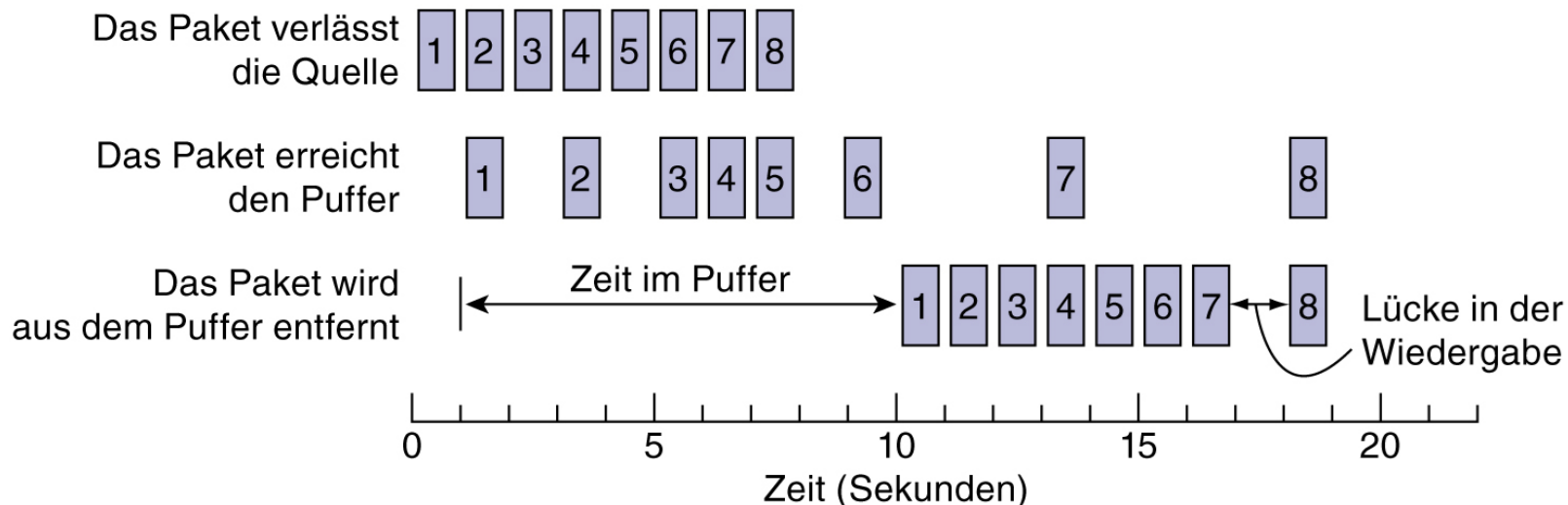


Abbildung 5.31: (a) Gewichtetes Fair Queueing. (b) Beendigungszeiten der Pakete.

Streams und Dienstgüte

.Weitere Anpassung:

- Einführung eines Puffers (zwecks Verringerung des Jitters)
- Speicherung der Datenpakete für gewisse Zeit
- Übergabe an Empfänger in einer regelmäßigen Frequenz



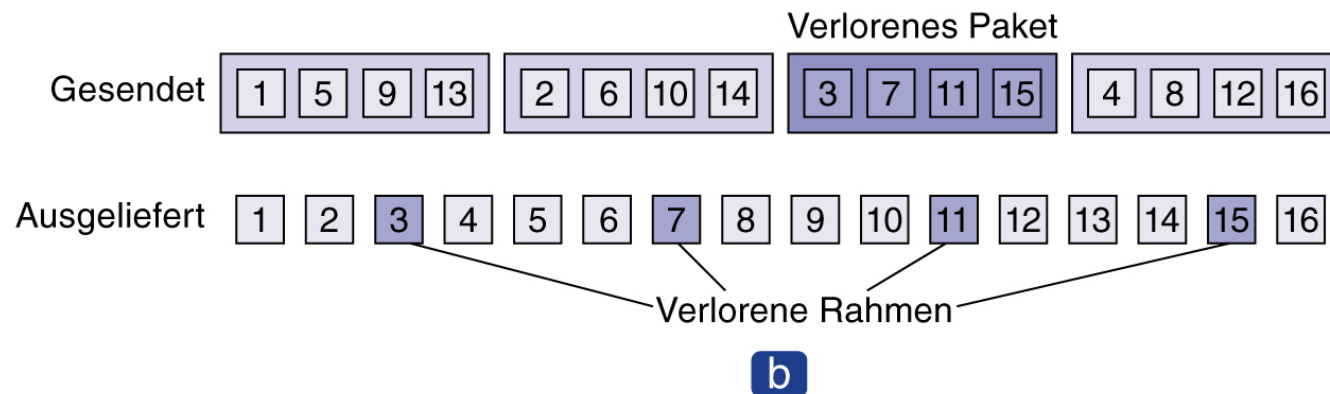
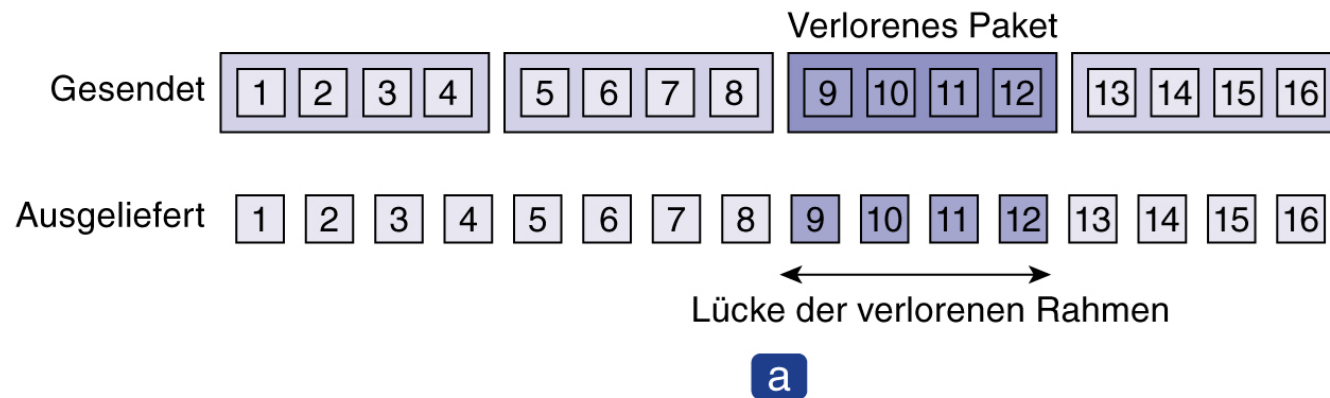
Streams und Dienstgüte

.Weitere Anpassung (Teil 2):

- Umgang mit verloren gegangenen Paketen
 - Neuübertragung keine Lösung
- Vorwärtsgerichtete Fehlerkorrektur
 - Idee ist, dass k von n übertragenen Paketen ausreicht für die richtige Darstellung ($k < n$)
 - Problem nur wenn Paket mehrere Audio- und Video-Rahmen enthält, aber ...

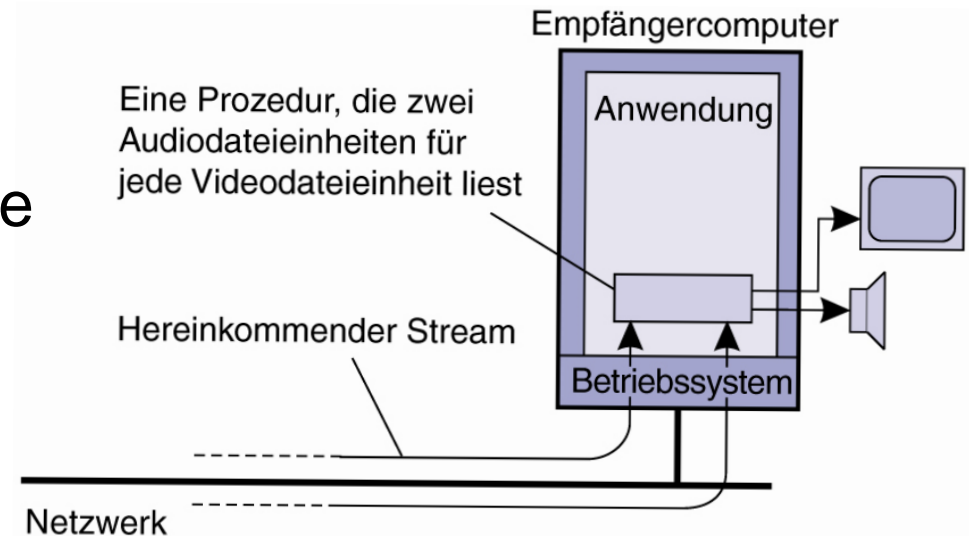
Streams und Dienstgüte

- Vorwärtsgerichtete Fehlerkorrektur
 - Aufteilung der Rahmen auf verschiedene Pakete
 - Problem: Startverzögerung

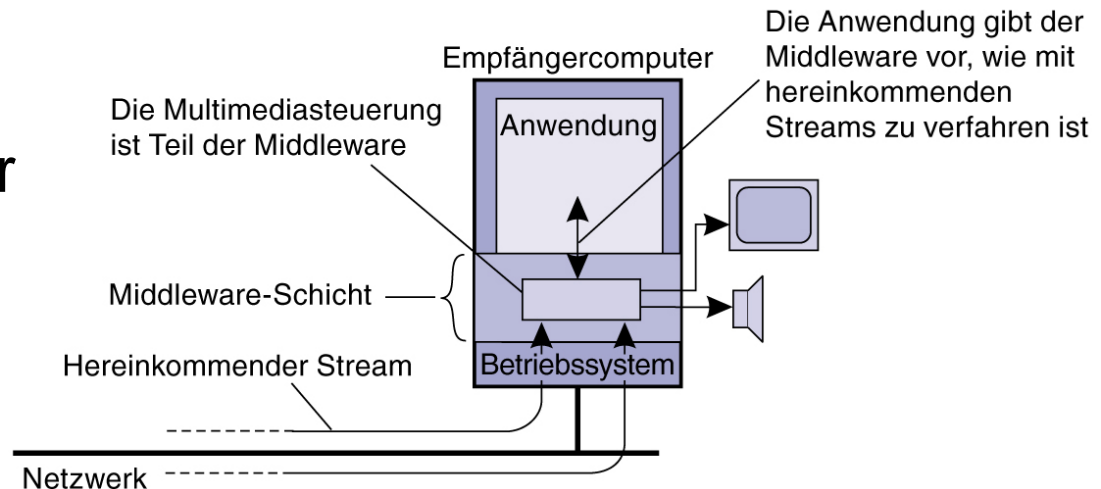


Stream: Synchronisierungsmechanismen

•Prinzip der expliziten
Synchronisierung auf der Ebene
der Dateneinheiten



•Synchronisierungsprinzip unter
Verwendung von High-Level-
Schnittstellen



Themenüberblick

.Kommunikation

- Entfernter Prozeduraufruf (Remote Procedure Call, RPC)
- Nachrichtenorientierte Kommunikation
- Streamorientierte Kommunikation
- **Multicast-Kommunikation**

Multicast-Kommunikation

- Wie ließe sich ein Multicast in einem Peer-to-Peer-System realisieren?
- Welche Schwierigkeiten könnten man beim Multicast in Peer-to-Peer-Systemen begegnen?
- Wie könnten Sie herausbekommen, ob Ihre Lösung auch effizient ist?

Multicast-Kommunikation

•Multicast-Kommunikation → Senden von Daten an mehrere Empfänger

•Fokus:

- Multicast-Kommunikation in Peer-to-Peer-Netzwerken

•Problematik:

- Knoten gliedern sich in einem Overlay-Netzwerk
- Aber Netzwerkknoten nicht Teil des Overlay-Netzwerks
- Nachrichten-Routing nutzt mitunter keine optimalen Routen.

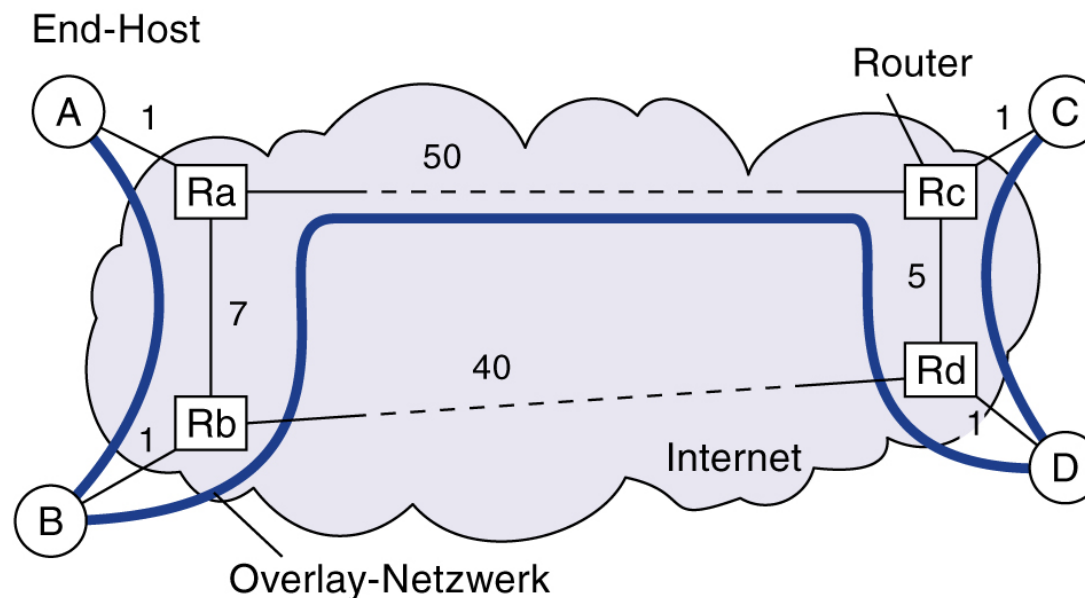
Multicast-Kommunikation

- Entwurfsfrage hinsichtlich Aufbau des Overlay-Netzwerkes
 - Baumstruktur
 - Maschennetzwerk
- Einfacher Ansatz – Fokus: Baumstruktur
 - Senden einer Nachricht: Weitergabe der Nachricht an die Wurzel des Multicast-Baumes
 - Teilnahmeanfrage: Kontaktierung der nächsten Knoten auf dem Weg zur Wurzel
 - Knoten nicht Teil des Baumes: Einstufung als Weiterleiter
 - Knoten Teil des Baumes: Anfrager wird Kind des Knotens

Multicast-Kommunikation Overlay Konstruktion

.Problematik

- A-B-D-C = Overlay-Netzwerk
- Ra, Rb, Rc, Rd = Netzwerkrouter
- Mehrere Pfade werden doppelt gegangen!!



Multicast-Kommunikation

Overlay Konstruktion

• Metriken für Qualität des Multicast-Baumes

- Link-Stress
 - Wie oft kreuzt ein Paket eine Verbindung (Link)?
- Ausdehnung / relative Verzögerung (Relative Delay Penalty)
 - Verhältnis zwischen Verzögerung zweier Knoten im Overlay-Netzwerk zum zugrundeliegenden Netzwerk (Zeit im Overlay-Netzwerk geteilt durch Zeit im zugrundeliegenden Netzwerk)
- Baumkosten
 - Globale Metrik zur Minimierung der gesammelten Verbindungskosten
 - Ziel: Minimaler „Spanning Tree“ = Gesamtzeit zur Informationsverbreitung an alle Knoten minimal

Multicast-Kommunikation

Overlay Konstruktion

• Lösungsmöglichkeiten

- Wurzel als einzige Quelle => Sterntopologie
 - Überlastung der Quelle möglich
 - Lösung: nur k Nachbarn erlaubt
- Switch Trees
 - Wechsel der Eltern
 - Bedingung: neue Eltern kein Mitglied des Unterbaums
 - Idee: Austausch von Infos zwischen Knoten
 - Achtung: Gleichzeitiger Elternwechsel 2er Knoten verbieten
 - Vermeidung von Schleifen im Multicast-Baum

Multicast-Kommunikation

Weitere Fragen

.Ziel:

- Informationsverbreitung auf allen Knoten eines Peer-to-Peer-Systems

.Wie könnten Sie eine solche Informationsverbreitung realisieren?

Multicast-Kommunikation

Gossip-basierte Datenverbreitung

.Idee:

- Informationsverbreitung basiert auf epidemischen Verhalten (wie Ansteckung von Krankheiten)

.Einteilung der Knoten

- „infizierte Knoten“ → Knoten, der Daten enthält, die verteilt werden sollen
- „anfällige Knoten“ → Knoten, der Daten nicht kennt
- „entfernter Knoten“ → aktualisierter Knoten, der Daten nicht verteilen will oder kann

Multicast-Kommunikation

Gossip-basierte Datenverbreitung

•Verbreitungsmodelle

- Anti-Entropie
- Gerüchteverbreitung (Gossip)

•Anti-Entropie

- P wählt zufälligen Knoten Q und
 - Push: P überträgt nur seine eigene Aktualisierung an Q
 - Pull: P erhält nur neue Aktualisierungen von Q
 - Push-Pull: Gegenseitiger Austausch der Aktualisierungen
- Push alleine ineffizient / Kombination mit Pull
- Aktualisierung aller Knoten in Größenordnung $\log(N)$ -Runden (N=Anzahl der Knoten im System)

Multicast-Kommunikation

Gossip-basierte Datenverbreitung

.Gerüchteverbreitung (Gossip)

- P wählt zufälligen Knoten Q um Aktualisierung weiter zu geben
 - Wenn Q „anfällig“ → suche nächsten Knoten
 - Wenn Q „infiziert“ → P reicht Aktualisierung mit einer bestimmten Wahrscheinlichkeit nicht mehr weiter
- Problem: Gewisse Menge wird „anfällig“ Knoten bleiben.
- Lösung: Kombination mit Anti-Entropie

.Stand: Weitergabe von Aktualisierungen möglich, aber wie funktioniert die Weitergabe von Löschungen?

Multicast-Kommunikation

Gossip-basierte Datenverbreitung

•Verbreitung von Löschungen

- Ummünzung: Löschen entspricht Aktualisierung
- Verteilen eines Löschzertifikates („Totenschein“)
- Aufräumen der Löschzertifikate nötig
- Sicherstellung des Löschens: Wenige Knoten halten schlafende Löschzertifikate, die nicht verworfen werden

•Hinweis zur Anwendung Gossip-basierter Datenverarbeitung:

- Abschätzung der Größe eines Peer-To-Peer-Netzwerkes