Work in progress

Rechnernetze (1)

Ute Bormann, TI2

2023-10-13

Inhalt

1. Erreichen des Empfängersystems
2. Kommunizierende Anwendungen

Teil 1:
Erreichen des Empfängersystems

Systemübergreifender Nachrichtenaustausch
⇒ Rechnernetze (eigene LV im SoSe)

Zusätzliche Problembereiche:

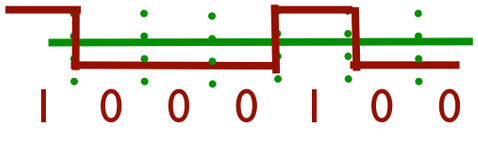
- Nachrichten müssen auf dem Medium kodiert werden
  - Abhängig vom Medium
    - Kupferkabel ⇒ elektrische Ströme/Spannungen
    - Glasfasern ⇒ Lichtimpulse
    - Funkstrecken ⇒ elektromagnetische Schwingungen

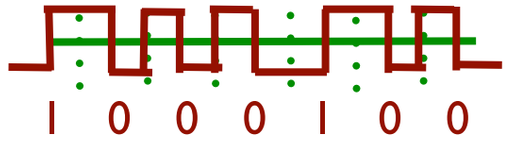
Systemübergreifender Nachrichtenaustausch
⇒ Rechnernetze (eigene LV im SoSe)

Zusätzliche Problembereiche:

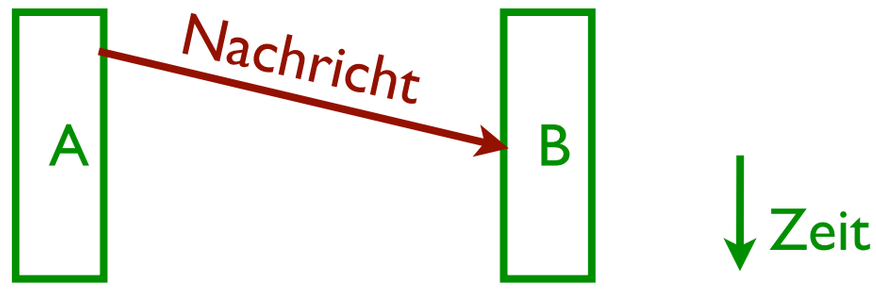
- Nachrichten müssen auf dem Medium kodiert werden
  - Abhängig vom Medium
    - Kupferkabel ⇒ elektrische Ströme/Spannungen
    - Glasfasern ⇒ Lichtimpulse
    - Funkstrecken ⇒ elektromagnetische Schwingungen
  - Signalfolgen repräsentieren im einfachsten Fall Bitströme
    ⇒ Unterscheidung von „0“ und „1“ erforderlich
  - Beispiele: (digitale Übertragung)

    ⇒ Bittakt...



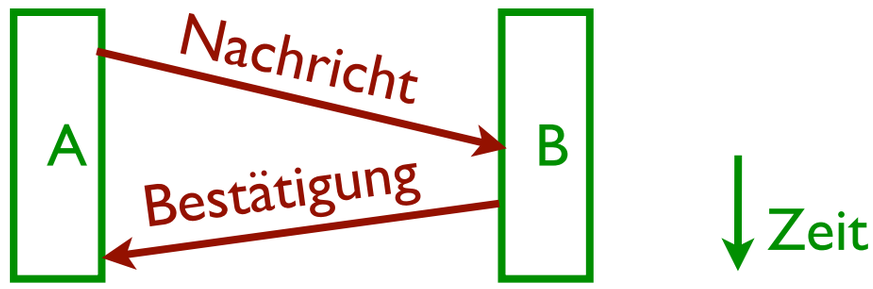


- Nachrichtenübertragung dauert Zeit
  - Kein gemeinsamer Zustand
  - Asynchrone Kommunikation



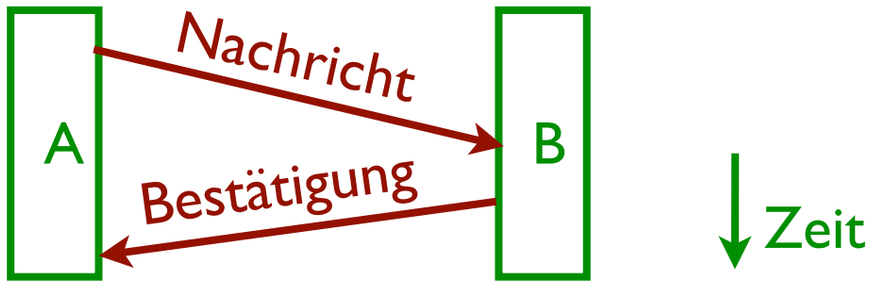
- Nachrichtenübertragung dauert Zeit
  - Kein gemeinsamer Zustand
  - Asynchrone Kommunikation

    ⇒ Herstellung der Synchronität über Bestätigungen

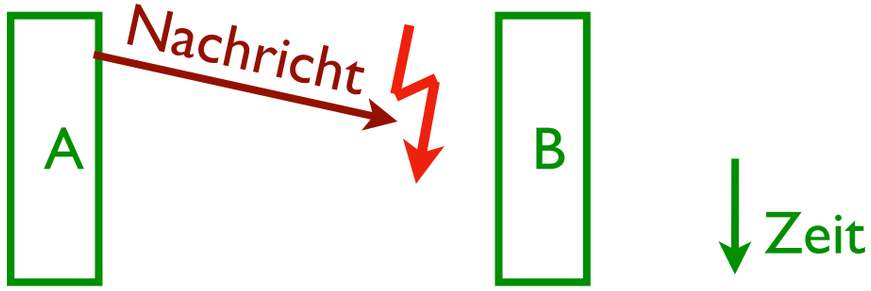


- Nachrichtenübertragung dauert Zeit
  - Kein gemeinsamer Zustand
  - Asynchrone Kommunikation

    ⇒ Herstellung der Synchronität über Bestätigungen

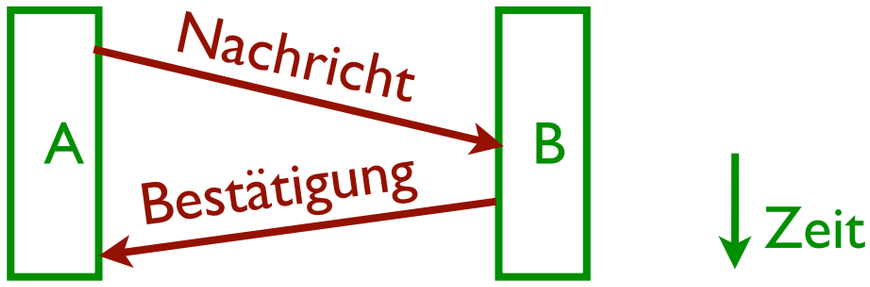


- Kommunikationskanal kann störanfällig sein
  - Übertragungsfehler („Bitkipper“)



- Nachrichtenübertragung dauert Zeit
  - Kein gemeinsamer Zustand
  - Asynchrone Kommunikation

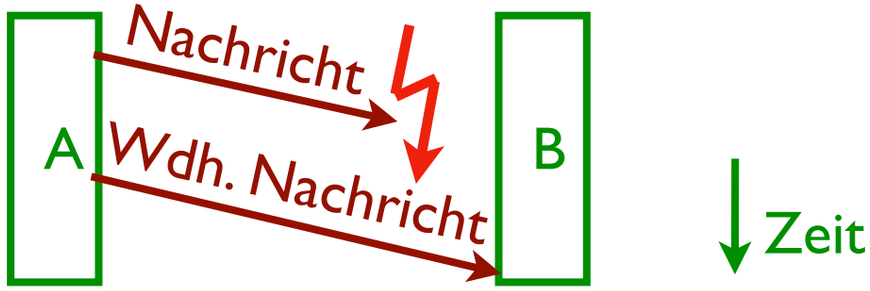
    ⇒ Herstellung der Synchronität über Bestätigungen



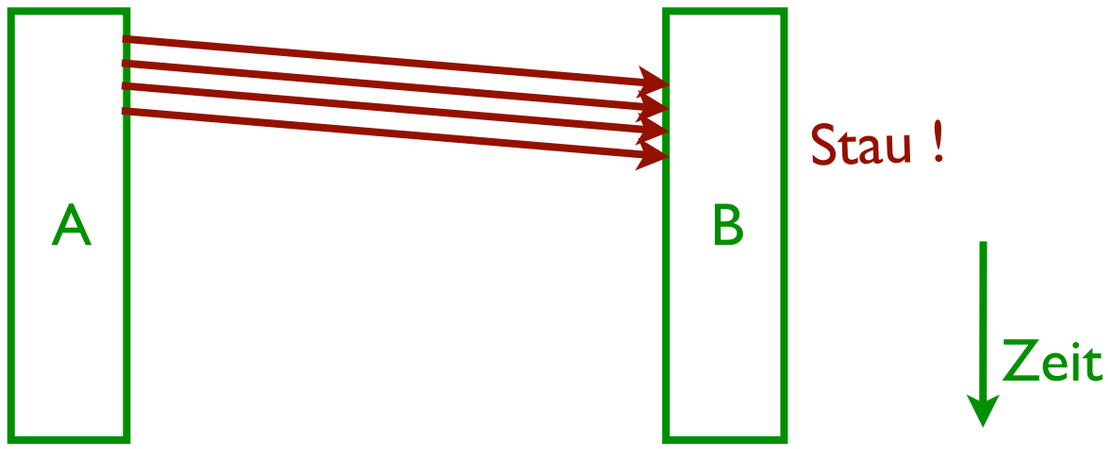
- Kommunikationskanal kann störanfällig sein
  - Übertragungsfehler („Bitkipper“)

    ⇒ Mitsenden von Redundanz (Prüfsummen)

    ⇒ Ggf. Wiederholung (bei Ablehnung oder Timeout)

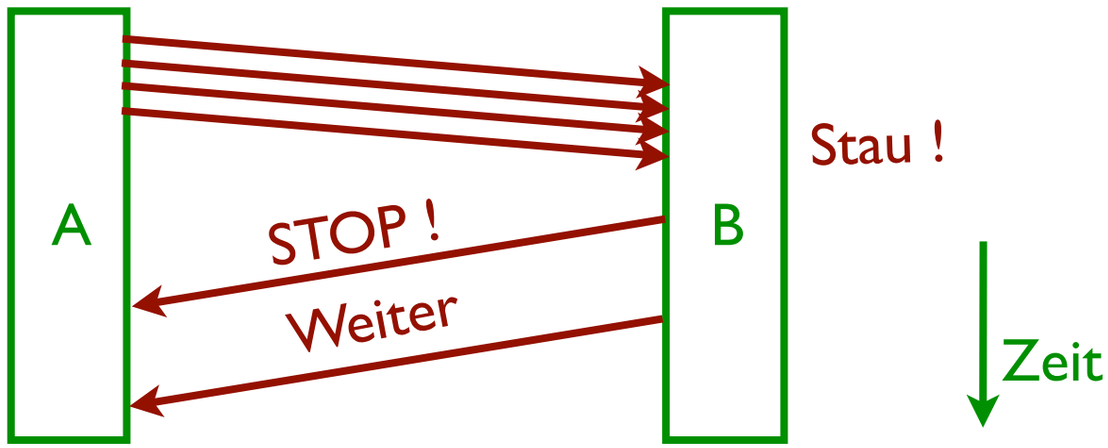


- Empfänger kann überlastet werden
  - Mehr Nachrichten senden als abgenommen werden können

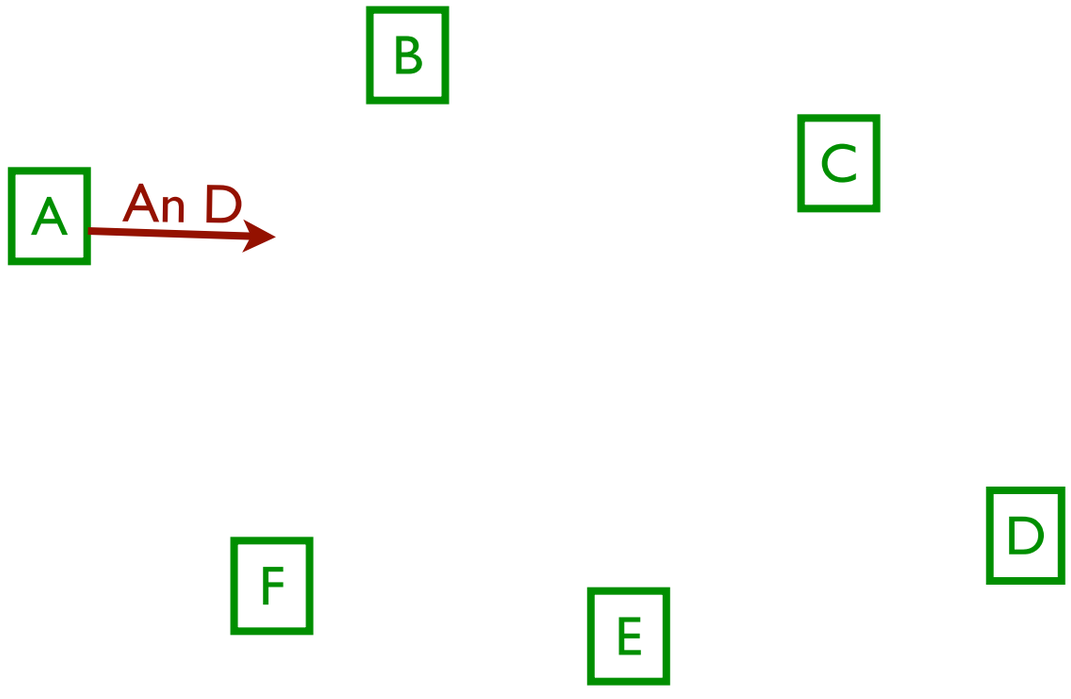


- Empfänger kann überlastet werden
  - Mehr Nachrichten senden als abgenommen werden können

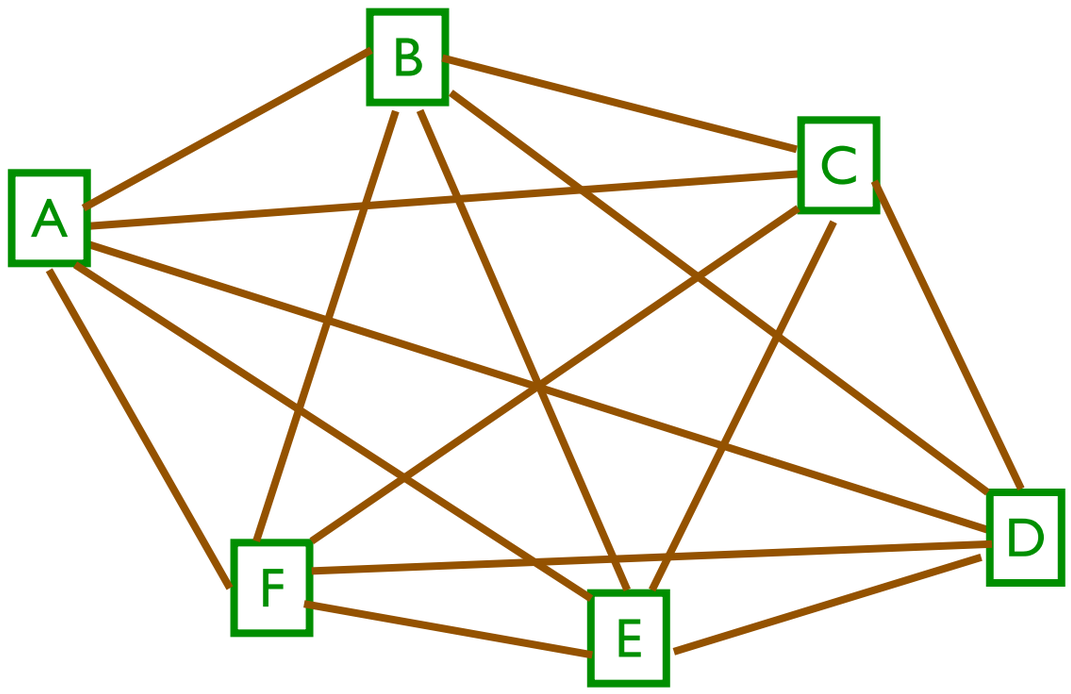
    ⇒ Flusskontrolle (z.B. durch „Staumeldung“ an Sender)



- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig

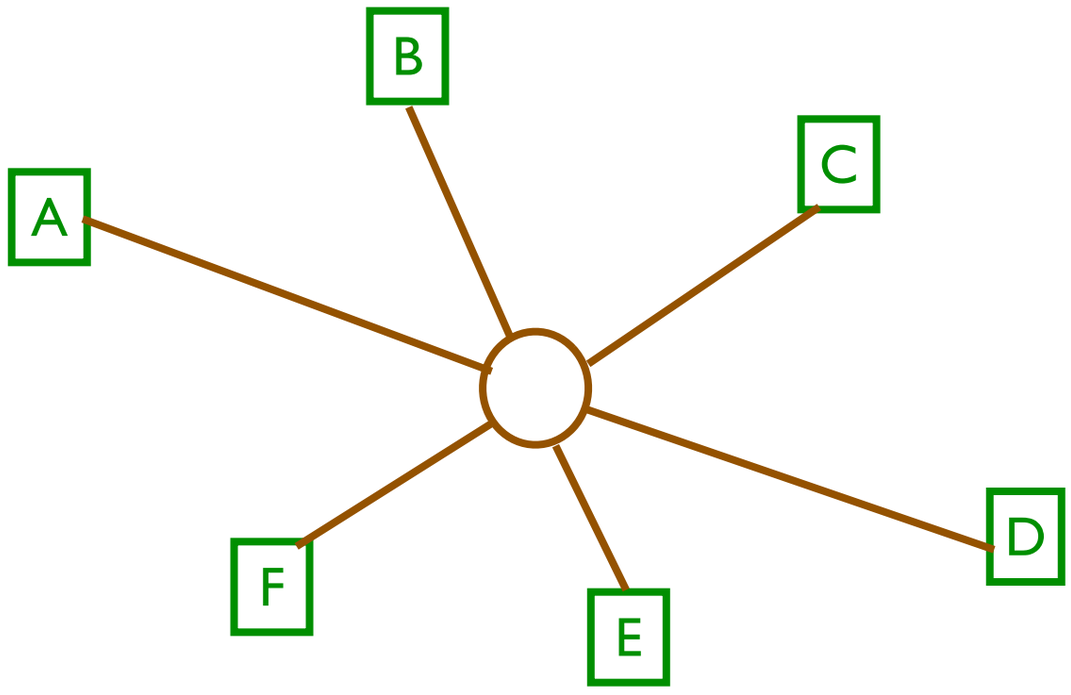


- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



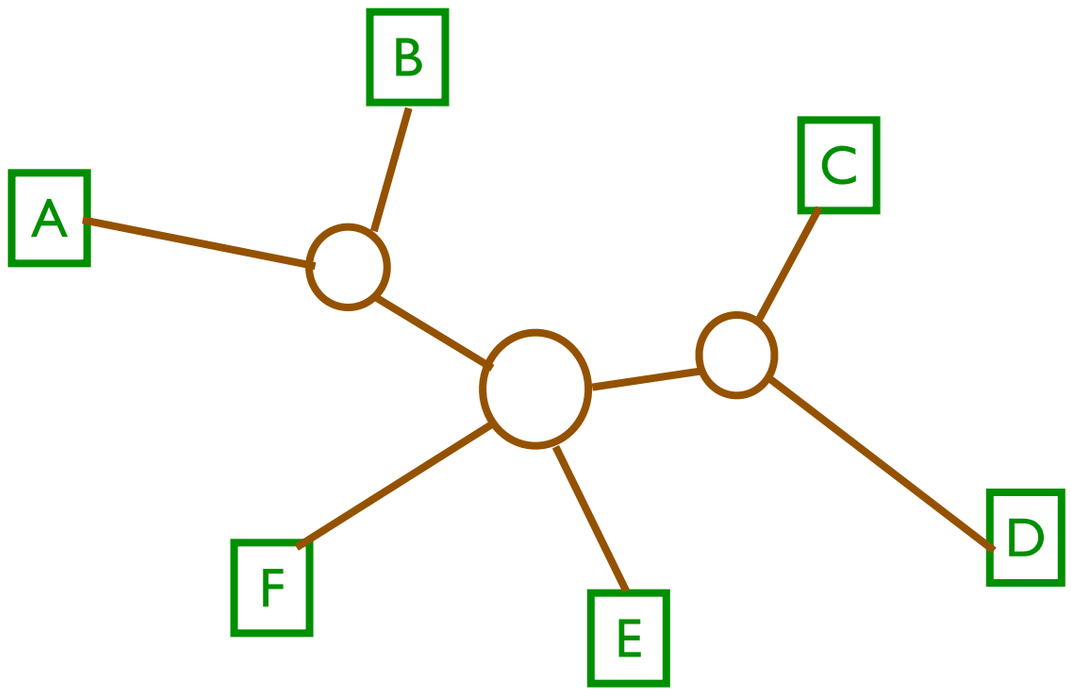
Vollständige Vermaschung

- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



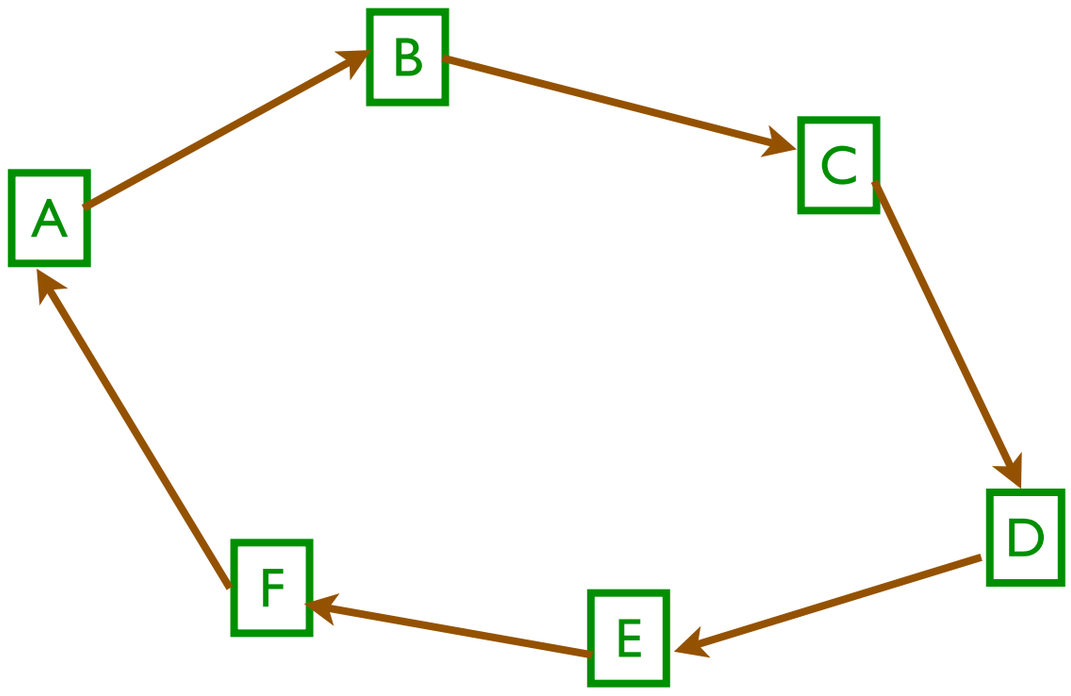
Stern

- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



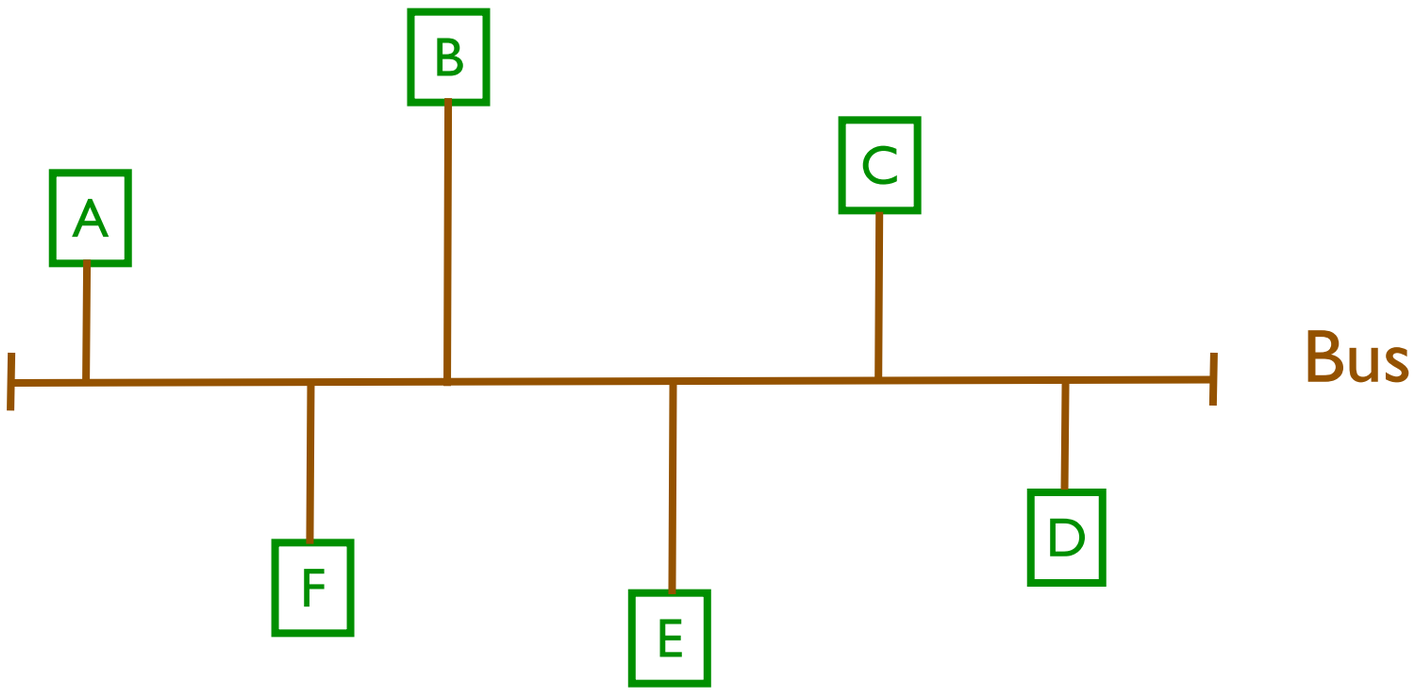
Baum

- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie

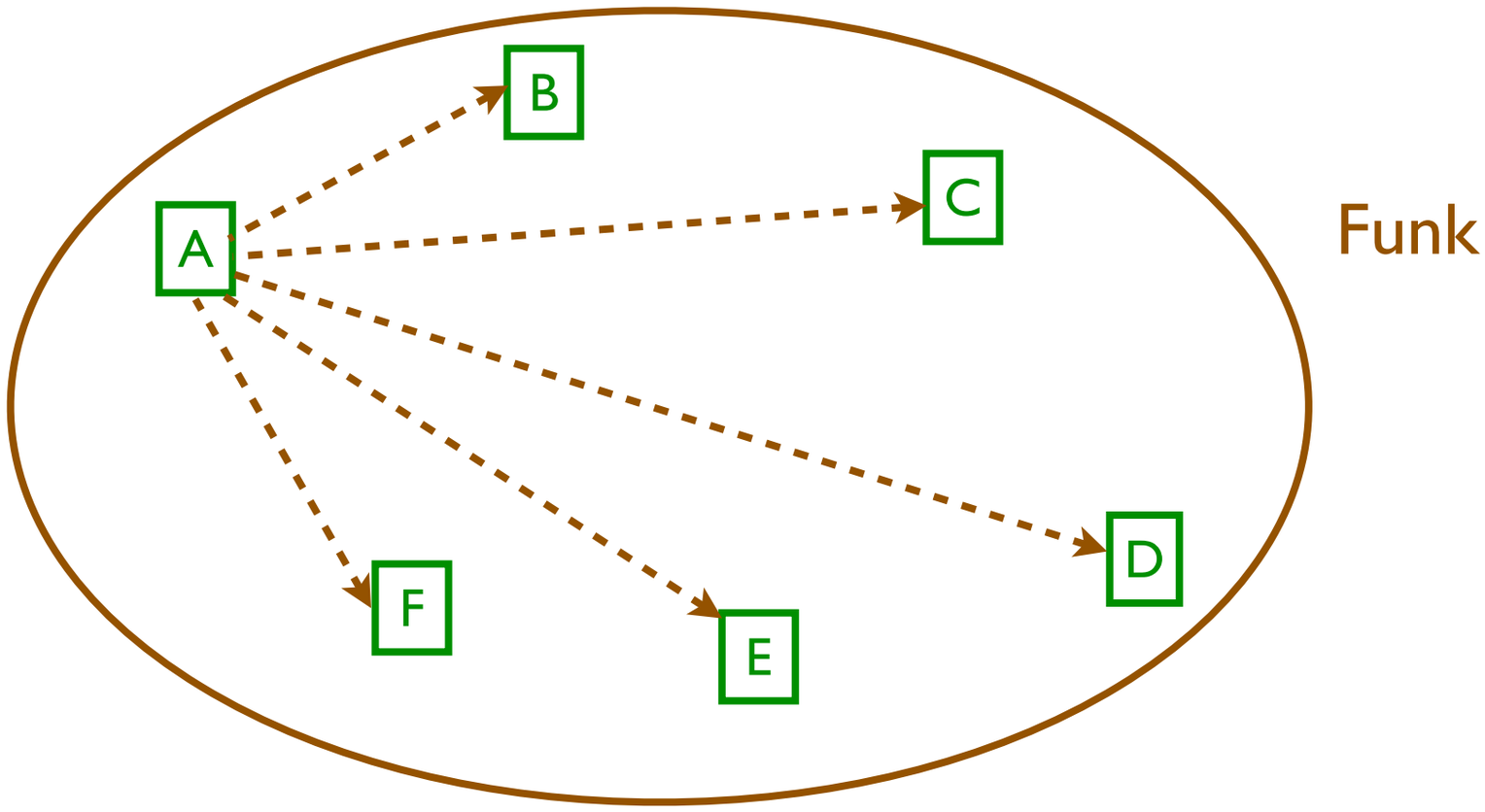


Ring

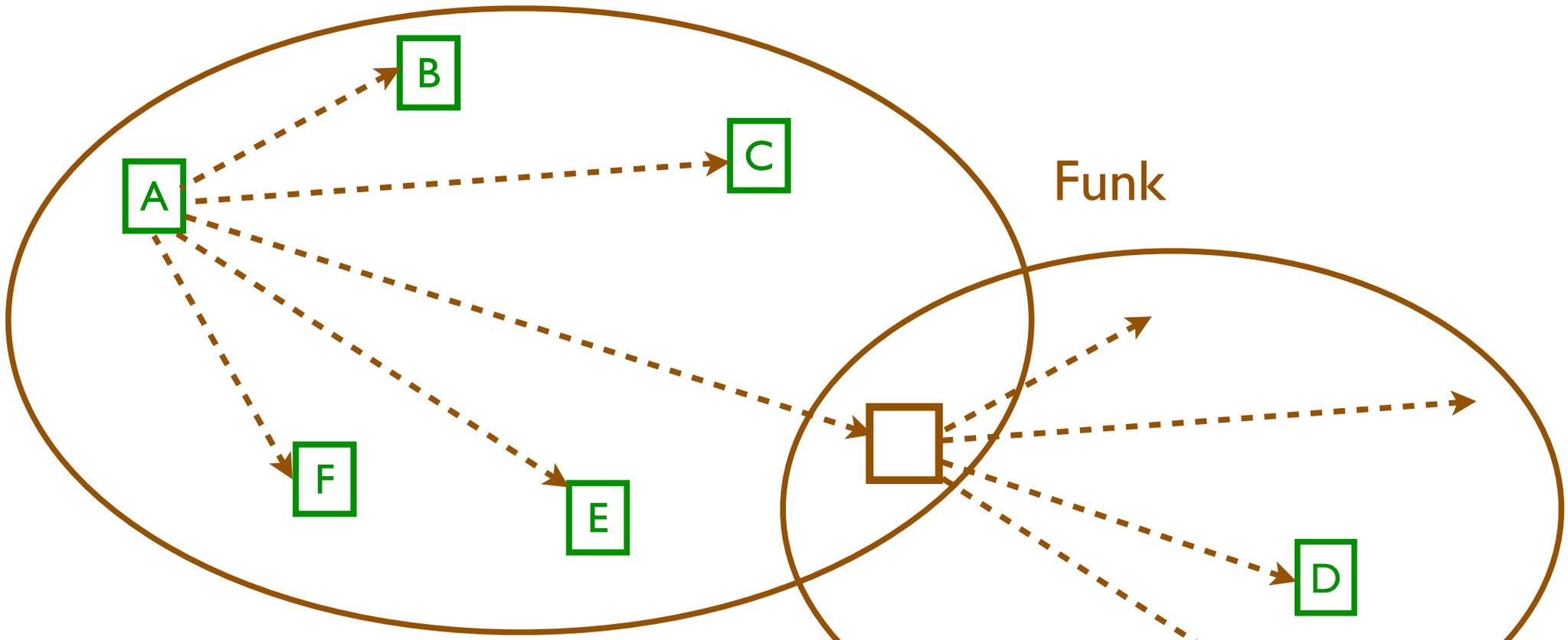
- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



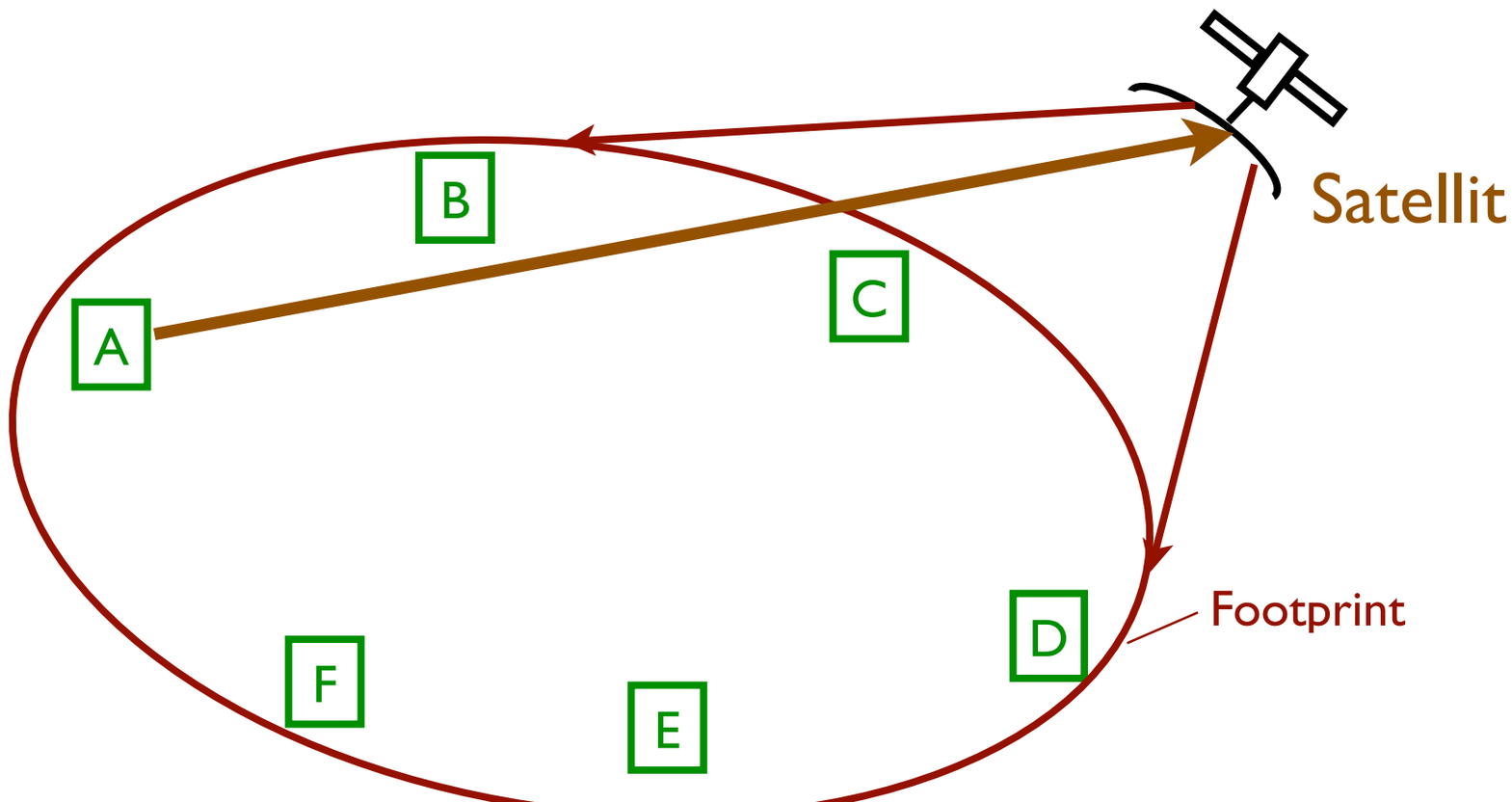
- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



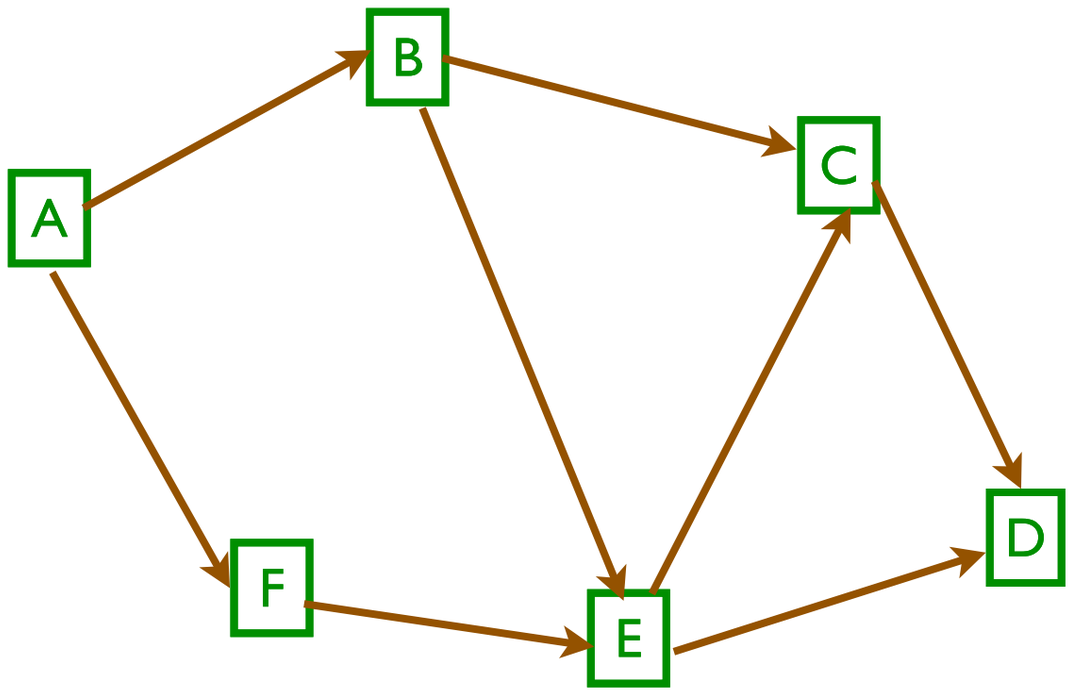
- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie

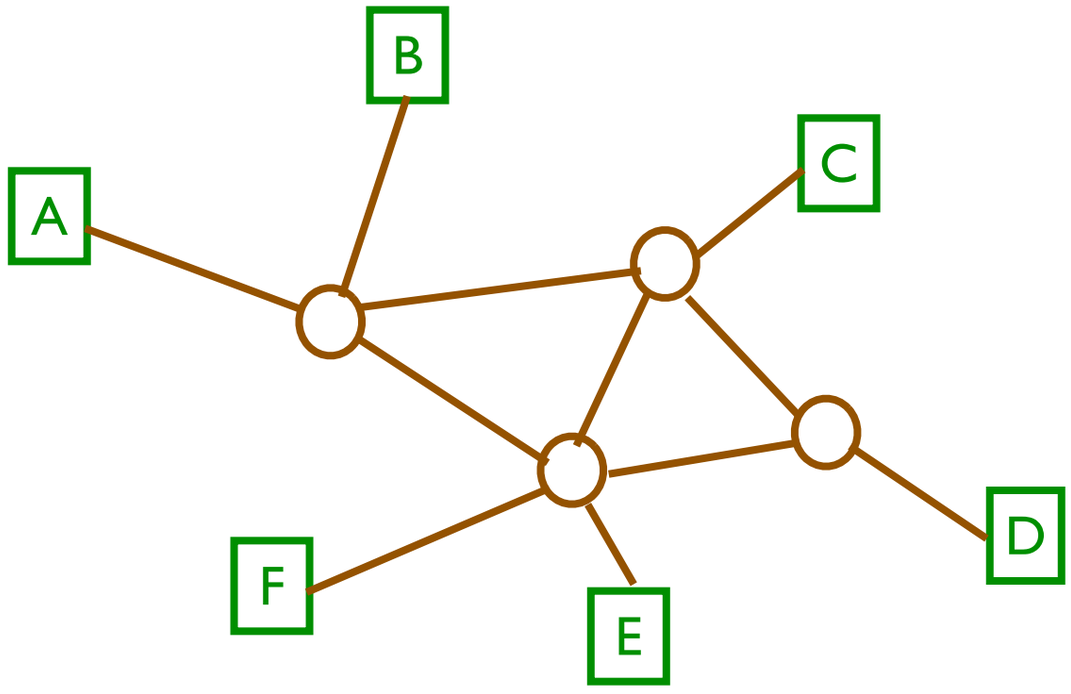


- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



gerichteter
azyklischer Graph

- Empfängersystem muss aufgefunden werden (nicht nur Prozess darin)
  - Viele potentielle Empfängersysteme (Hosts)
  - Oft kein direkter Kommunikationskanal vorhanden
    ⇒ würde zu viele „Leitungen“ erfordern
  - Folge: Adressierung der Nachrichten nötig
  - Wegewahl und Weiterleitung abhängig von Netztopologie



Allgemeines Netz

Kleine Aufgabe

Ordne die angegebenen Eigenschaften den angegebenen Netztopologien zu (Mehrfachnennungen zulässig).

Vollständige Vermaschung	Wegalternativen
Stern	Einfacher Broadcast
Ring	Leicht abhörbar
Allgemeines Netz	Hoher Aufwand
Bus	Kapazitätsengpässe
Funkzelle	Besonders störanfällig

Weiterleiten im Netz anhand der Adresse

⇒ Falls mehr als ein möglicher Weg
⇒ Wegewahl (Routing)

Weiterleiten im Netz anhand der Adresse

⇒ Falls mehr als ein möglicher Weg
⇒ Wegewahl (Routing)

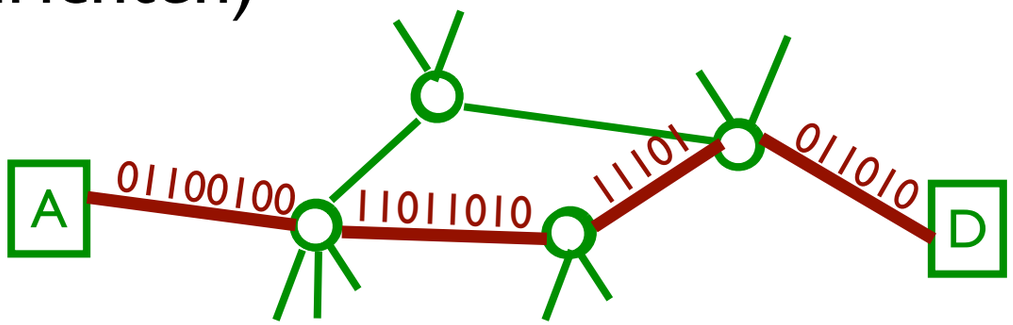
- Verbindungsorientiert

  1. Verbindungsaufbau mit Adressierung
  ⇒ „Durchschalten“ eines Wegs

  2. Datenübertragung (beliebige Nachrichten)
  ⇒ Nutzen des Wegs

  3. Verbindungsabbau
  ⇒ Freigeben des Wegs

  - Beispiel: Telefonverbindungen



Weiterleiten im Netz anhand der Adresse

⇒ Falls mehr als ein möglicher Weg
⇒ Wegewahl (Routing)

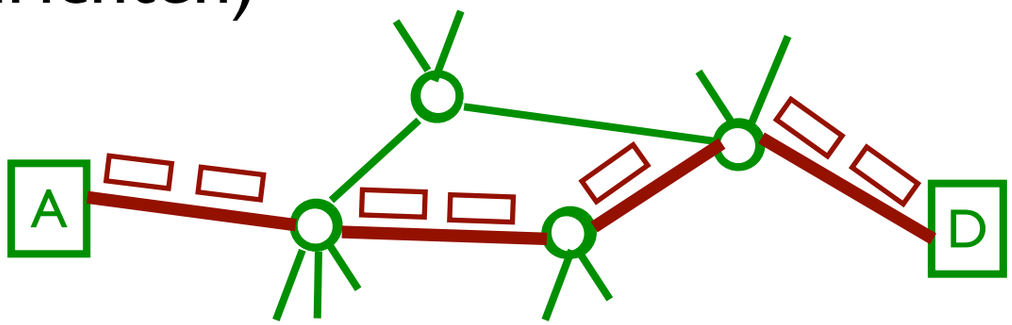
- Verbindungsorientiert

  1. Verbindungsaufbau mit Adressierung
  ⇒ „Durchschalten“ eines Wegs

  2. Datenübertragung (beliebige Nachrichten)
  ⇒ Nutzen des Wegs

  3. Verbindungsabbau
  ⇒ Freigeben des Wegs

  - Beispiel: Telefonverbindungen



Byteströme vs. Nachrichtenströme

Weiterleiten im Netz anhand der Adresse

⇒ Falls mehr als ein möglicher Weg
⇒ Wegewahl (Routing)

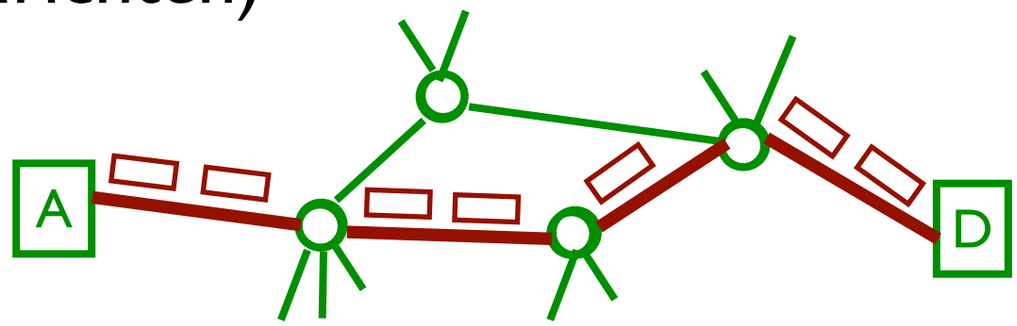
- Verbindungsorientiert

  1. Verbindungsaufbau mit Adressierung
  ⇒ „Durchschalten“ eines Wegs

  2. Datenübertragung (beliebige Nachrichten)
  ⇒ Nutzen des Wegs

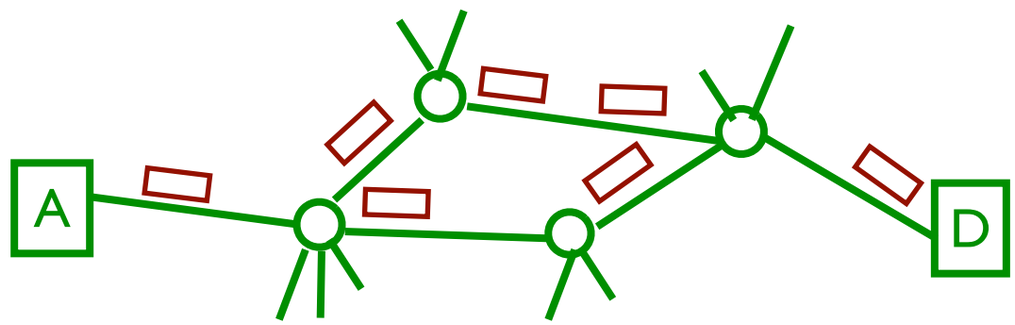
  3. Verbindungsabbau
  ⇒ Freigeben des Wegs

  - Beispiel: Telefonverbindungen



Byteströme vs. Nachrichtenströme

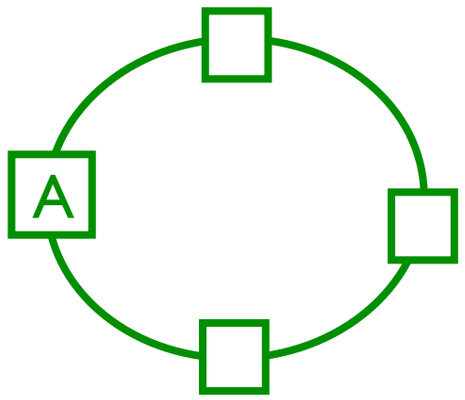
- Verbindungslos
  - Jede Nachricht enthält Adresse
    ⇒ verschiedene Wege möglich
    ⇒ Reihenfolgeerhalt?
  - Beispiel: Paketpost

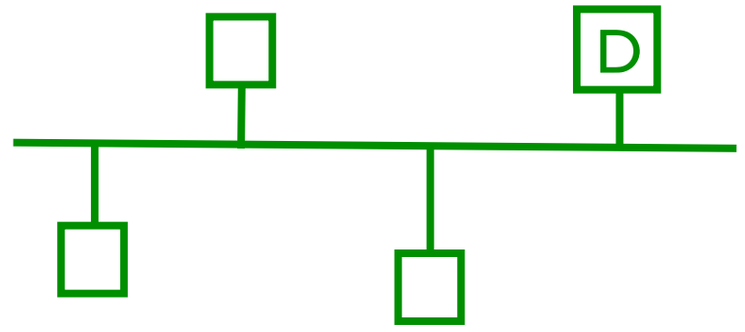


- Empfängersystem kann in andersartigem Netz liegen

  ⇒ Kopplung von heterogenen Netzen

- Beispiel:

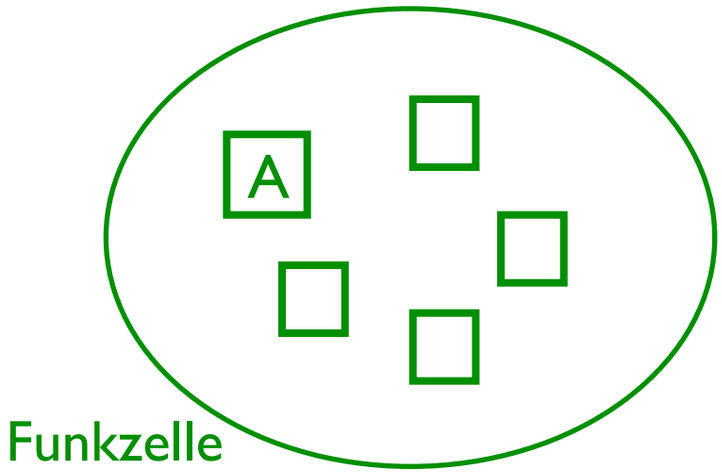


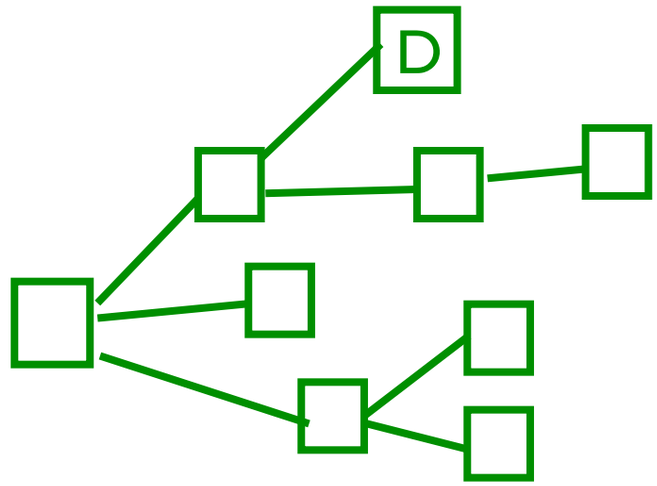


- Empfängersystem kann in andersartigem Netz liegen

  ⇒ Kopplung von heterogenen Netzen

- Beispiel:

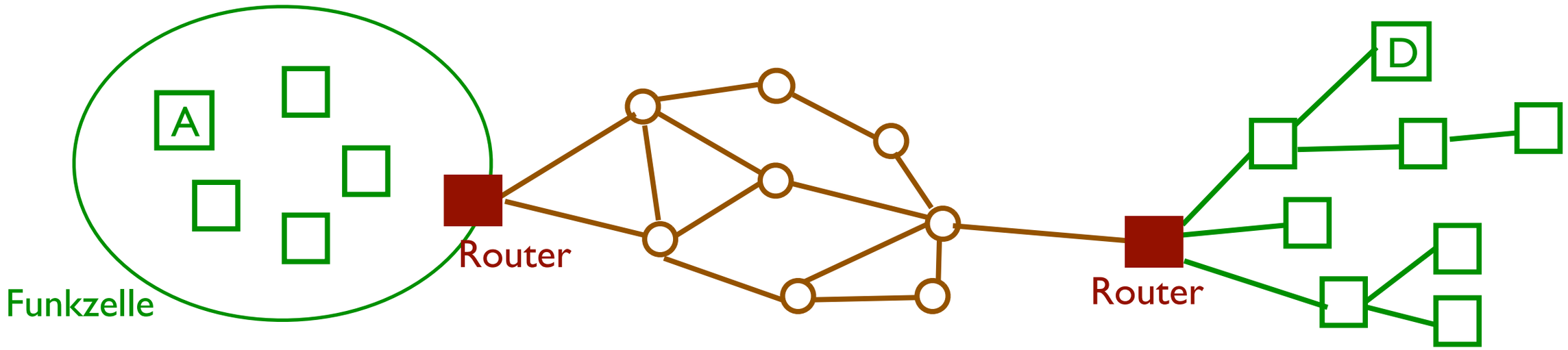




- Empfängersystem kann in andersartigem Netz liegen

  ⇒ Kopplung von heterogenen Netzen

- Beispiel:

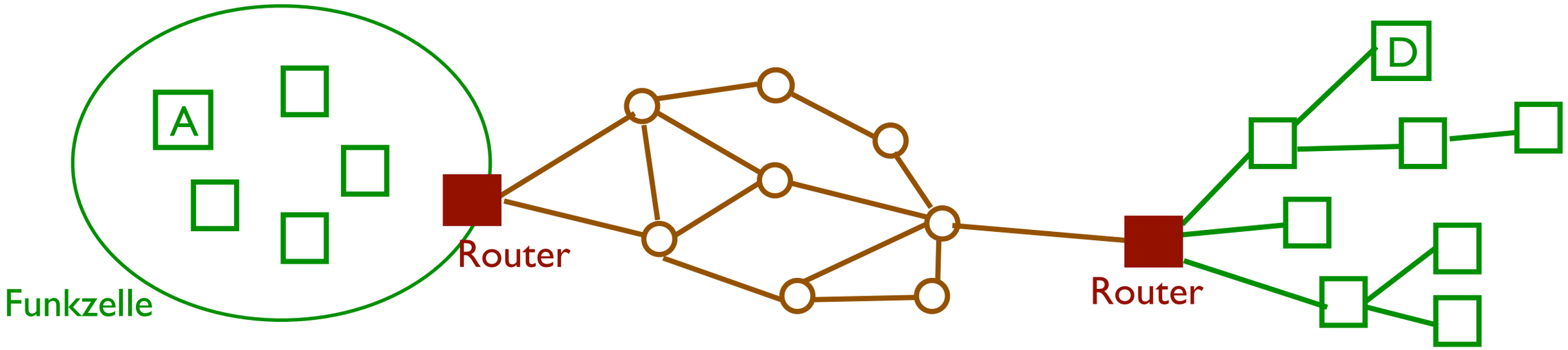


– Lokales Netz	– Weitverkehrsnetz	– Lokales Netz

- Empfängersystem kann in andersartigem Netz liegen

  ⇒ Kopplung von heterogenen Netzen

- Beispiel:



– Lokales Netz	– Weitverkehrsnetz	– Lokales Netz
– 48-Bit-Adressen	– z.B. Telefonnummern	– 48-Bit-Adressen
– verbindungslos	– z.T. verbindungsorientiert	– verbindungslos
– z.B. 100 Mbit/s	– i.d.R. Anteil von Breitbandlinks	– z.B. 1 Gbit/s

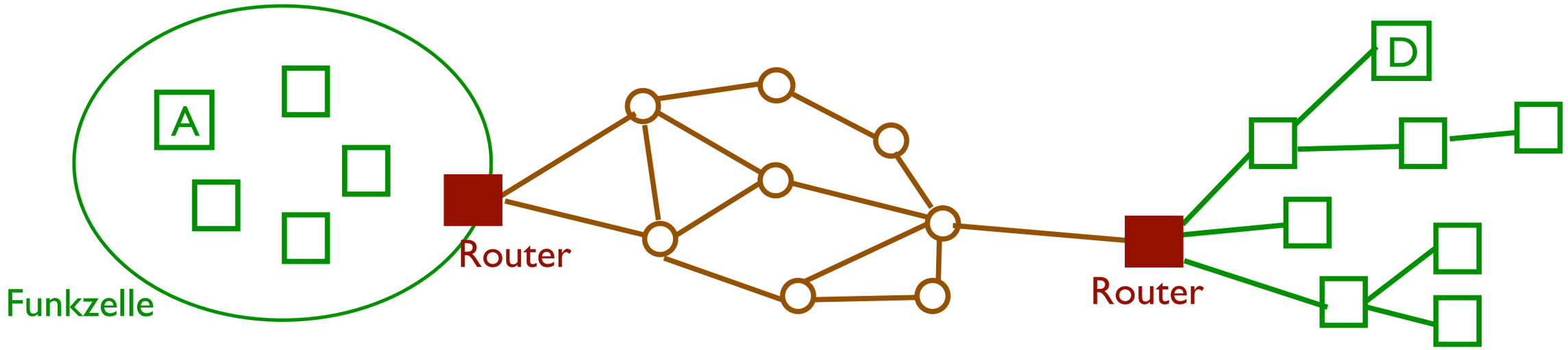
⇒ Anpassungen/Umwandlungen erforderlich (z.B. in Routern)

\+ netzübergreifende Adressierung (IP-Adressen)

- Empfängersystem kann in andersartigem Netz liegen

  ⇒ Kopplung von heterogenen Netzen

- Beispiel:



– Lokales Netz	– Weitverkehrsnetz	– Lokales Netz
– 48-Bit-Adressen	– z.B. Telefonnummern	– 48-Bit-Adressen
– verbindungslos	– z.T. verbindungsorientiert	– verbindungslos
– z.B. 100 Mbit/s	– i.d.R. Anteil von Breitbandlinks	– z.B. 1 Gbit/s

⇒ Anpassungen/Umwandlungen erforderlich (z.B. in Routern)
+ netzübergreifende Adressierung (IP-Adressen)

- + Sonderfall NAT

Fragen – Teil 1

- Skizziere kurz einige typische Kommunikationsprobleme und je eine mögliche Lösung.
- Skizziere einige Eigenschaften typischer *Netztopologien*.
- Welche besondere Bedeutung kommt IP zu?

Teil 2:
Kommunizierende Anwendungen

23 Rechnernetze (1), Ute Bormann

Bisher betrachtet:

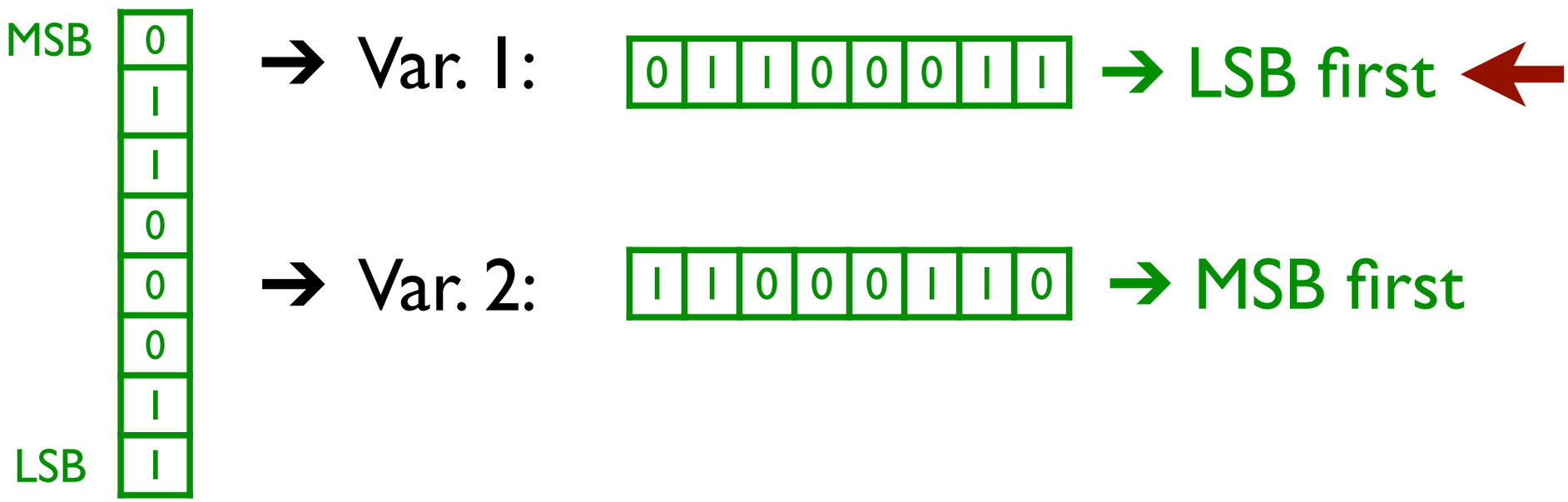
- Kommunikation über asynchronen Nachrichtenaustausch
- Viele Problembereiche, z.B.:
  - Kodierung von Bitströmen auf dem Medium
  - Erkennung und Behebung von Übertragungsfehlern
  - Schutz vor Überlastung
  - Adressierung und Wegewahl im Netz

- Nachrichten müssen repräsentiert werden
  - Alle Meldungen benötigen eindeutige Repräsentation

    ⇒ sonst Missverständnisse möglich
  - Heterogenität der systeminternen Informationsrepräsentation

- Nachrichten müssen repräsentiert werden
  - Alle Meldungen benötigen eindeutige Repräsentation

    ⇒ sonst Missverständnisse möglich
  - Heterogenität der systeminternen Informationsrepräsentation
- Beispiel 1: Bit Order?

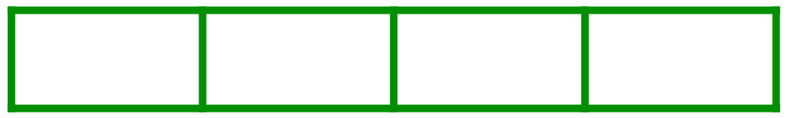


- Beispiel 2: Byte Order?

System A (z.B. Sun)
Most significant byte first (MSB first)

32-Bit-Wert

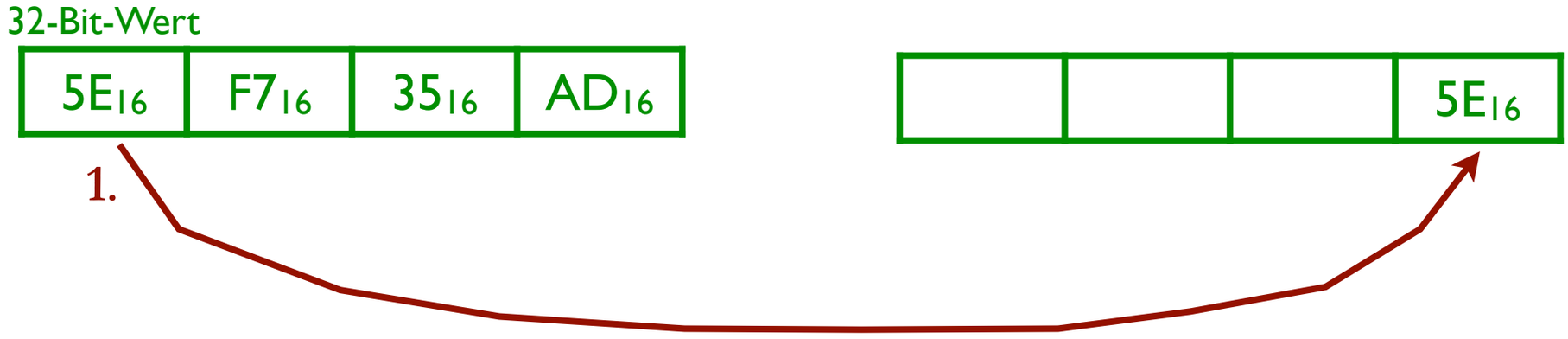
System B (z.B.Intel)
Least significant byte first (LSB first)



- Beispiel 2: Byte Order?

System A (z.B. Sun)
Most significant byte first (MSB first)

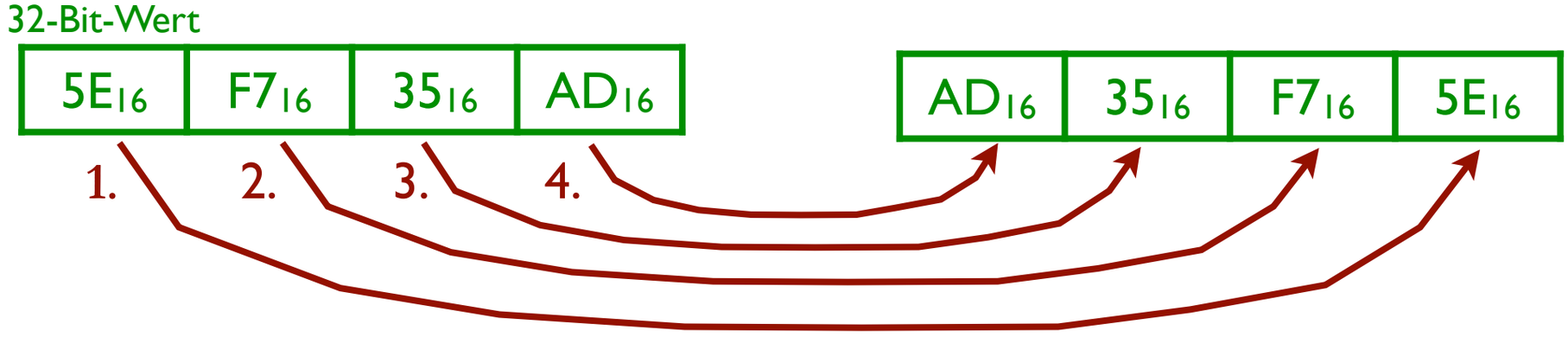
System B (z.B.Intel)
Least significant byte first (LSB first)



- Beispiel 2: Byte Order?

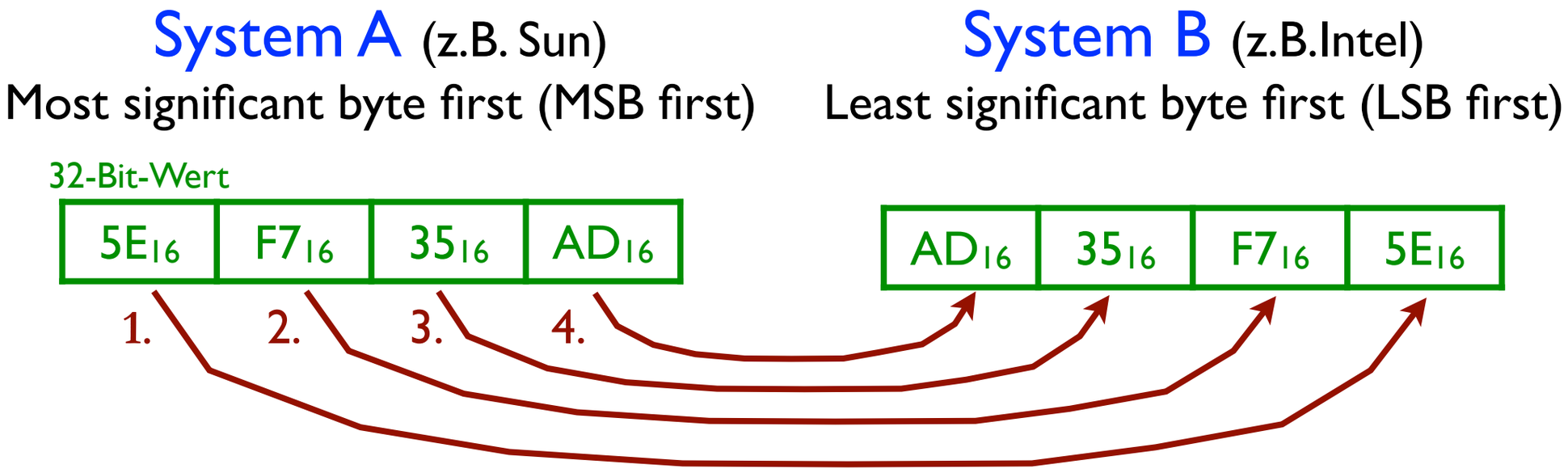
System A (z.B. Sun)
Most significant byte first (MSB first)

System B (z.B.Intel)
Least significant byte first (LSB first)



⇒ Sender und Empfänger müssen sich im Vorfeld über Byteorder beim Austausch einigen (i.d.R. MSB first)

- Beispiel 2: Byte Order?

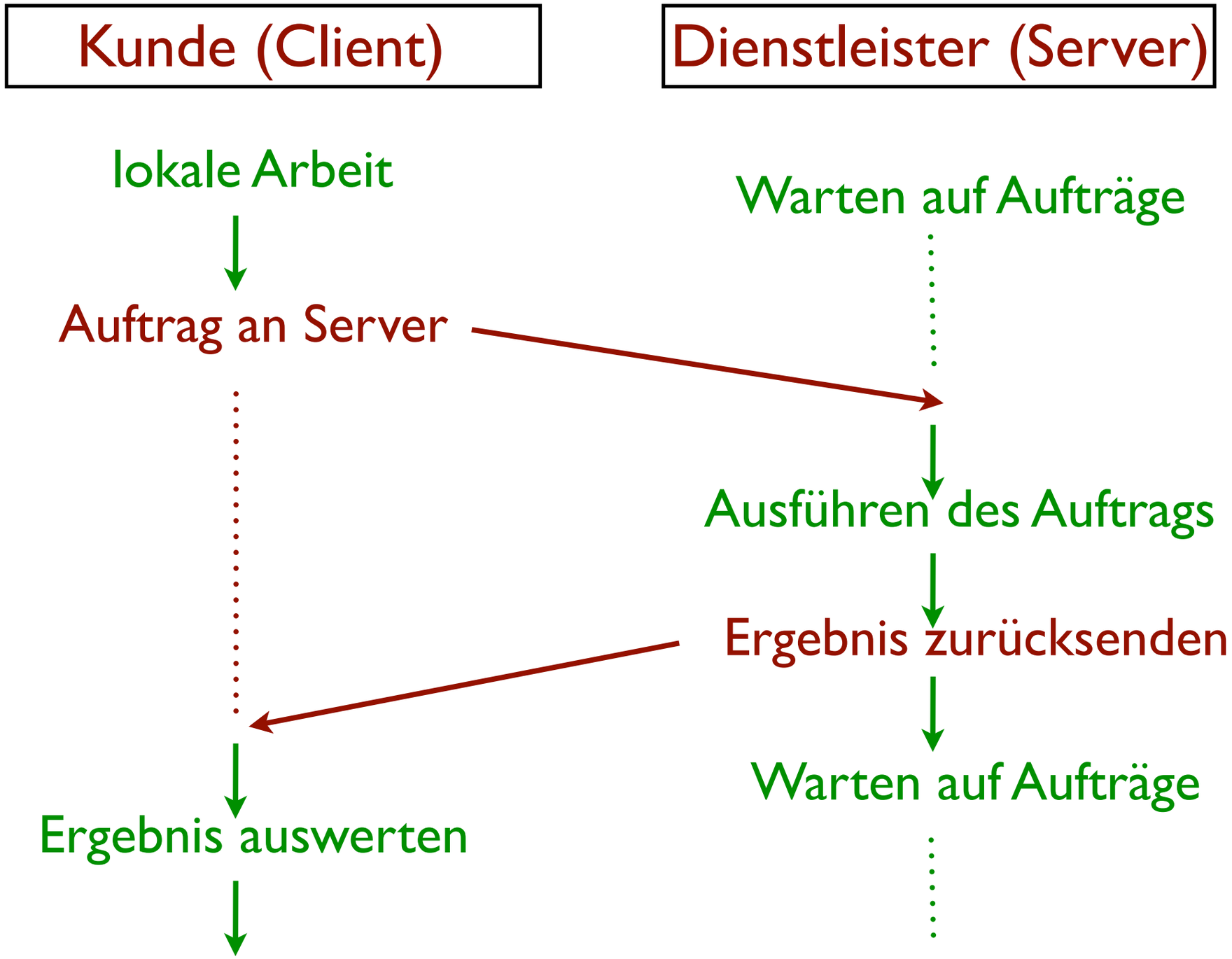


⇒ Sender und Empfänger müssen sich im Vorfeld über Byteorder beim Austausch einigen (i.d.R. MSB first)

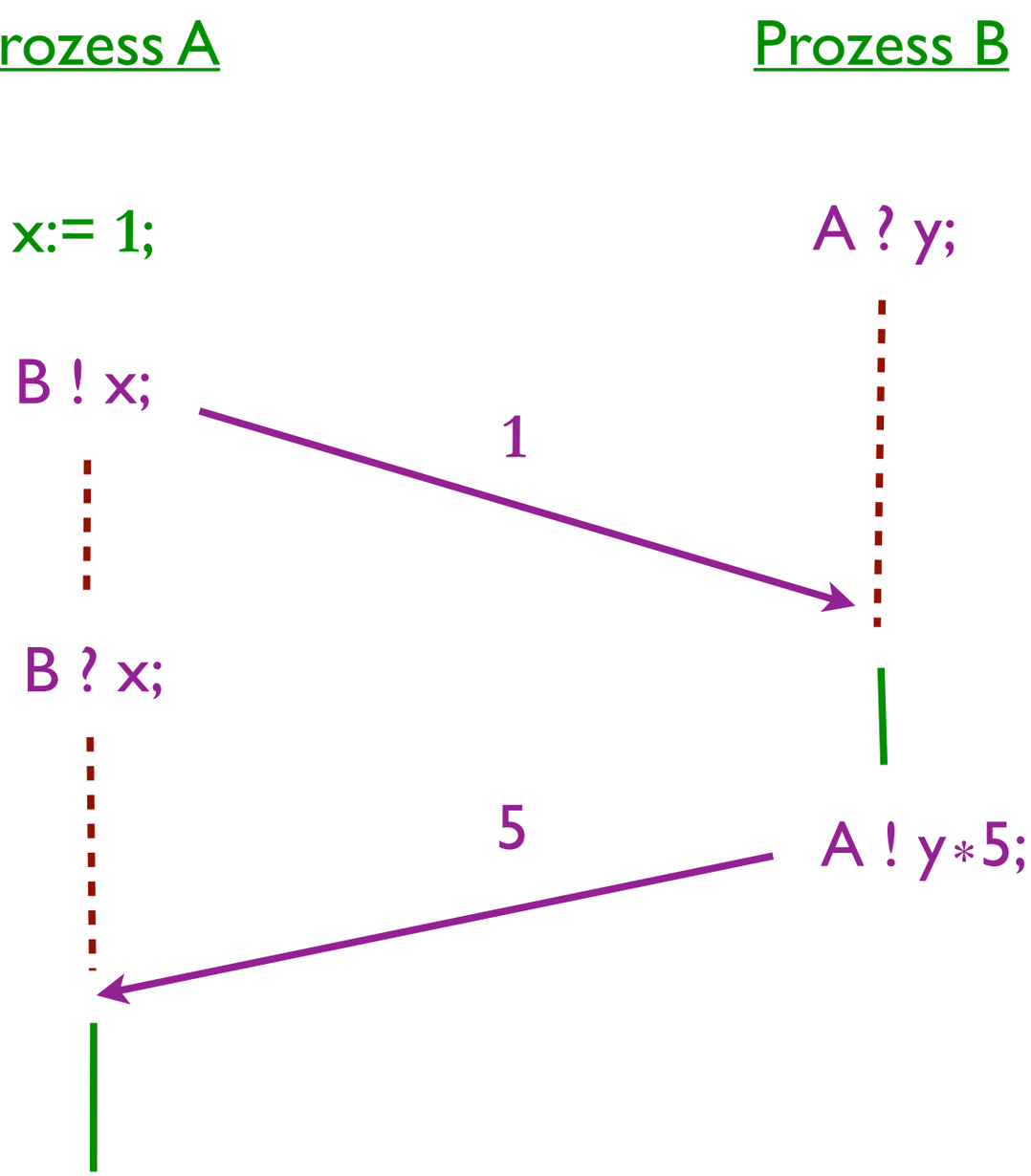
- Weitere potentielle Repräsentationsunterschiede:
  - Zeichencodes
  - ...

Das Client-/Server-Modell

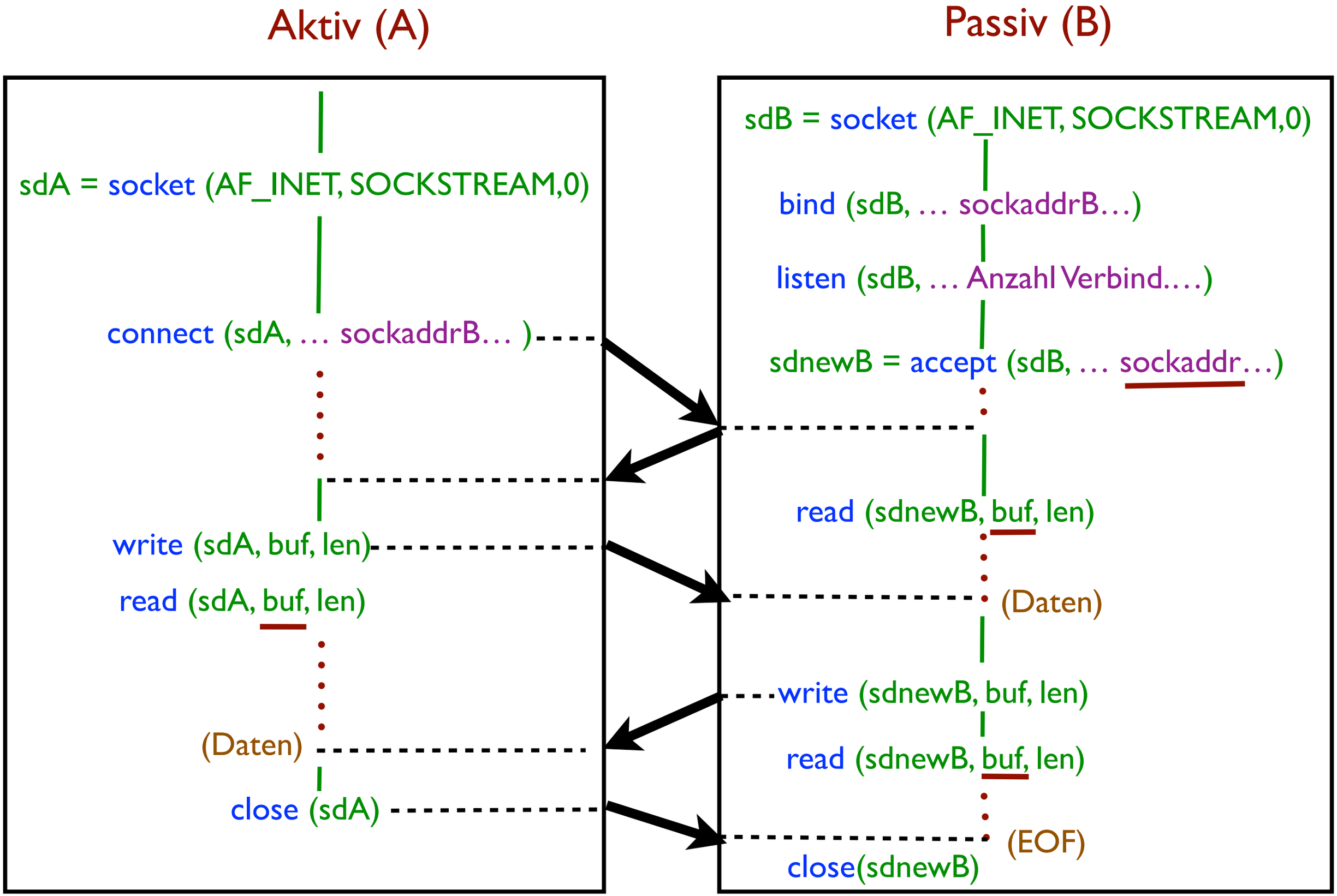
- Viele Kommunikationsbeziehungen stark asymmetrisch



Wdh.: Kommunikationsbeispiel in CSP ⇒ Client/Server



Wdh.: Kommunikationsbeispiel mit Sockets ⇒ Client/Server



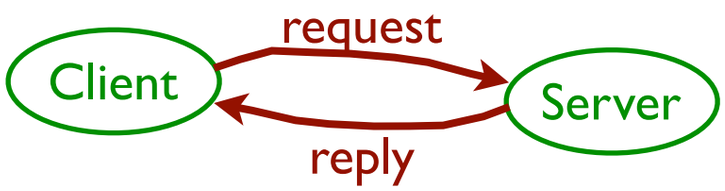
Prozedurfernaufrufe

- Client-/Server-Modell erinnert an lokale Prozeduraufrufe

  ⇒ Konzept verallgemeinern

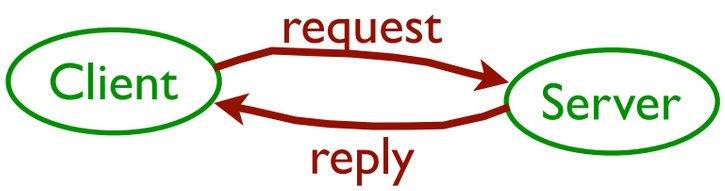
  ⇒ Remote Procedure Calls (RPC)

  ⇒ Aufruf bei anderem Prozess (evtl. auf anderem Rechner)

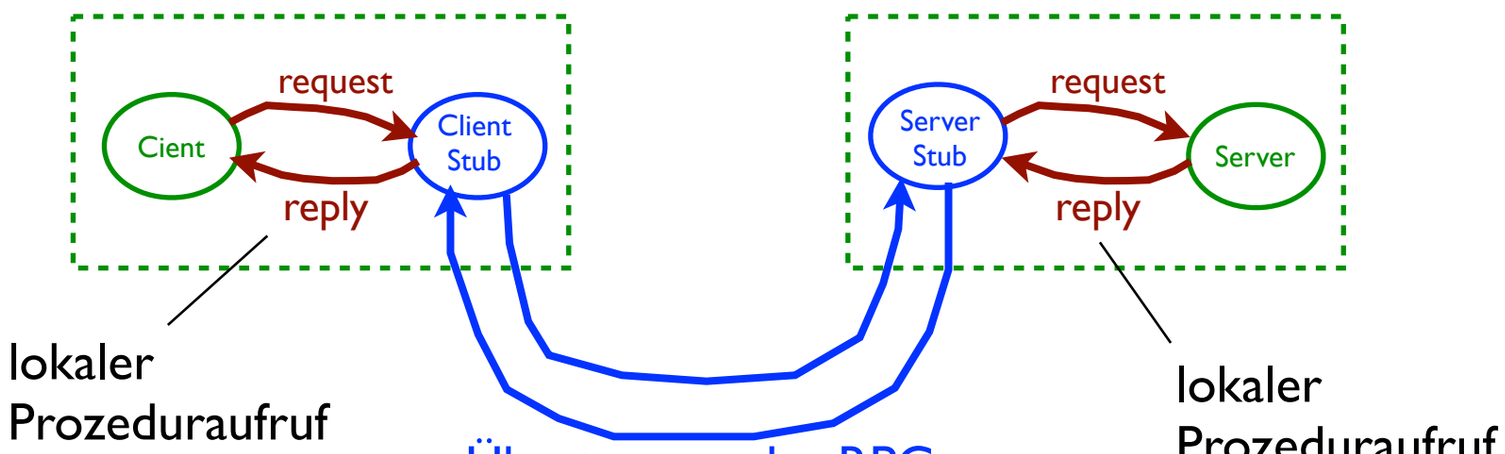


Prozedurfernaufrufe

- Client-/Server-Modell erinnert an lokale Prozeduraufrufe
  ⇒ Konzept verallgemeinern
  ⇒ Remote Procedure Calls (RPC)
  ⇒ Aufruf bei anderem Prozess (evtl. auf anderem Rechner)



- Weitgehende Abstraktion von unterliegenden Kommunikationseigenschaften (keine expliziten Nachrichten)



Übertragung des RPC:
– Verpacken der Nachricht
– Versenden über Netz
– Auspacken beim Empfänger

- RPCs können synchron (Client wartet auf Antwort) oder asynchron sein (jeweils auf Basis von asynchronem Nachrichtenaustausch)

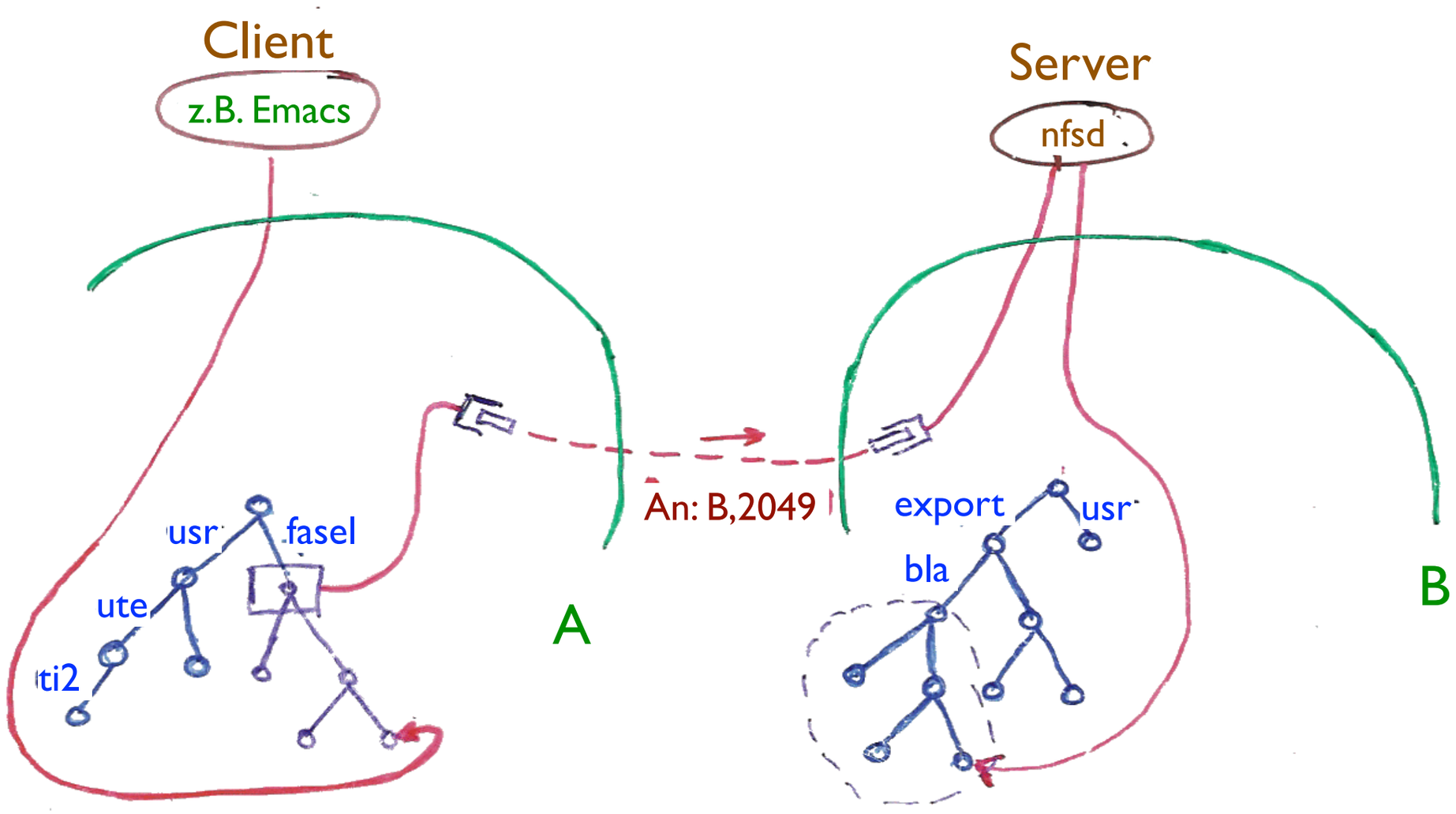
- Parameter eines RPC-Requests:
  - Identifizierung des Programms (+ Version)
  - Identifizierung der Prozedur
  - Eingabeparameter
  - ggf. Authentisierungsinfos
  - ggf. Aufrufnummer (falls asynchron)

- Parameter eines RPC-Replys:
  - ggf. Rückgabewert
  - ggf. Fehlermeldung
  - ggf. Bezug auf Aufrufnummer

- (Es gibt diverse Varianten eines RPC-ähnlichen Kommunikationsverfahrens)

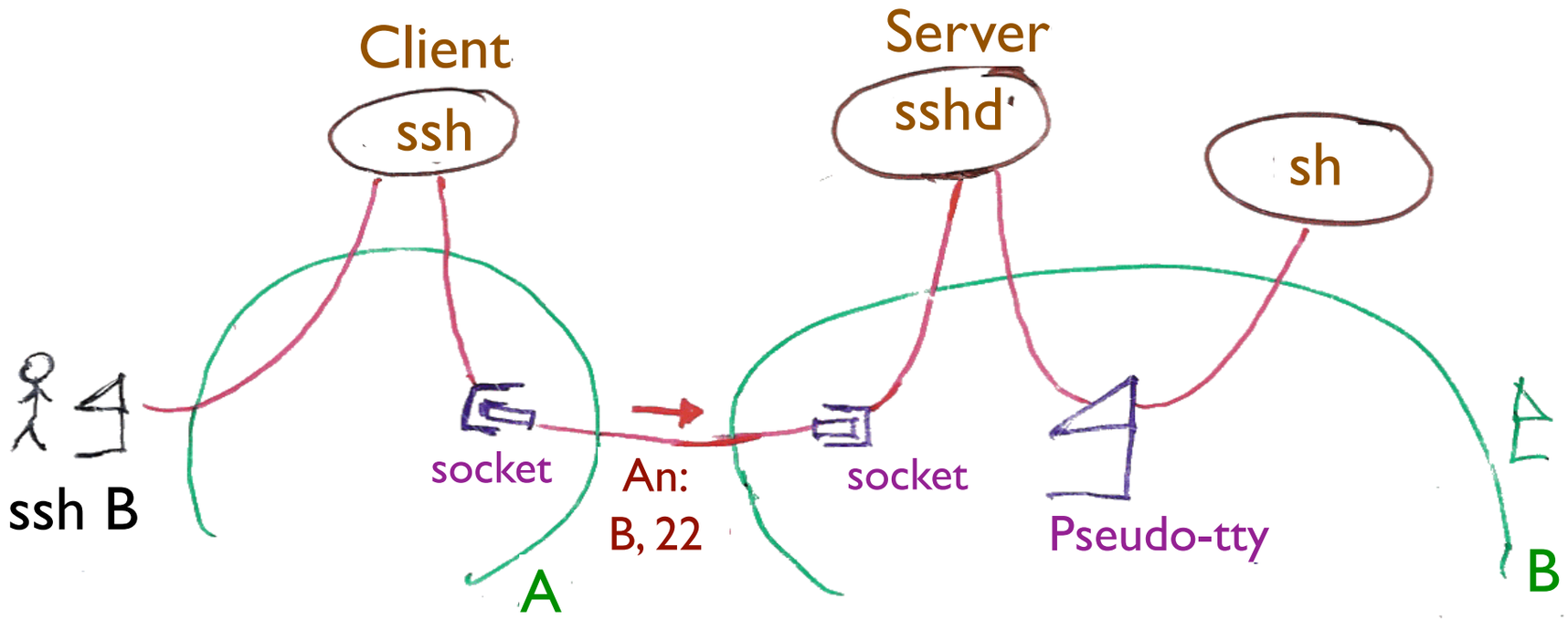
Beispiele für Kommunikationsanwendungen

1) Zugriff auf entfernte Dateisysteme (Beispiel: NFS)



mount B:/export/bla /fasel

2) Entferntes Einloggen (Beispiel: ssh)



- Durch ssh-Kommando wird Client-Prozess gestartet
- Der baut Verbindung zu Rechner B auf
  (22 ist die feste Portnummer des ssh-Dämons)
- Auf Rechner B empfängt ssh-Dämon die Nachricht (+ Authentisierung)
- Startet Shell-Prozess und kommuniziert mit ihm über ein Pseudo-tty
  („virtuelles“ Terminal)
- Shell weiss nicht, dass ein Fernzugriff erfolgt
  (normale Terminal-Schnittstelle)

Kommunikationsprotokolle

- Absprachen zwischen Kommunikationspartnern über Kommunikationsablauf, Nachrichtenformate, ...

  ⇒ sollten möglichst universell einsetzbar sein
  ⇒ internationale Standardisierung

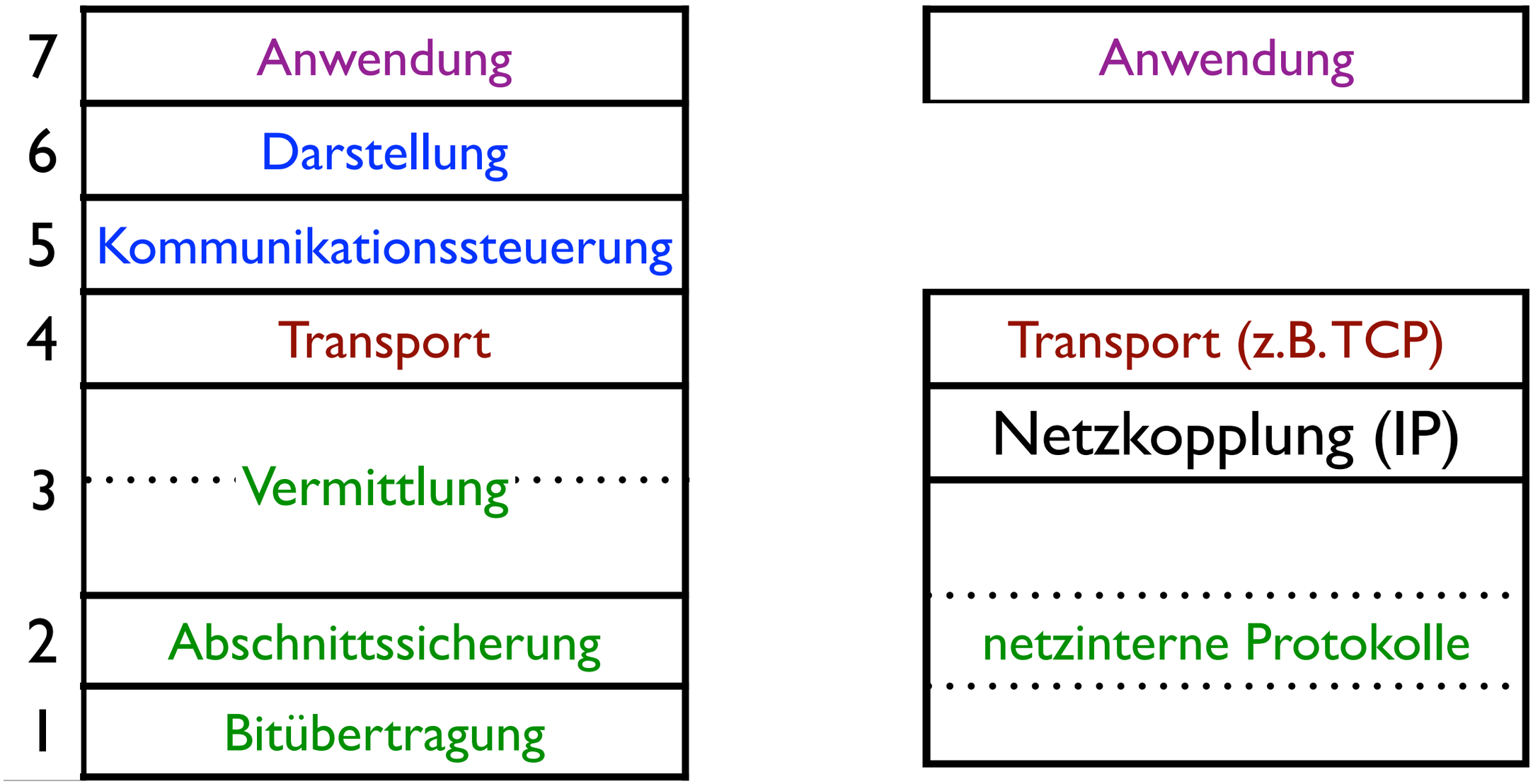
- Umfassen Mechanismen zur:
  - Informationsrepräsentation
  - Fehlersicherung
  - Flusskontrolle
  - Adressierung
  - Netzkopplung
  - ...

⇒ zu kompliziert für ein einziges Protokoll
⇒ Hierarchie von Protokollen

Kommunikationsarchitektur

OSI-Modell

Kommunikationsarchitektur



Fragen – Teil 2

- Was ist ein *Remote Procedure Call (RPC)*, und welche Parameter wird man typischerweise dabei übergeben?
- Was ist ein *(Kommunikations-)Protokoll*?

Zusammenfassung

- Kommunikation über asynchronen Nachrichtenaustausch
- Viele Problembereiche, z.B.:
  - Kodierung von Bitströmen auf dem Medium
  - Erkennung und Behebung von Übertragungsfehlern
  - Schutz vor Überlastung
  - Adressierung und Wegewahl im Netz
  - Repräsentation der Nachrichten
  - Diverse Anwendungsprotokolle...
- Protokolle
- Kommunikationsarchitekturen: OSI vs. Internet

Rechnernetze 1 – Fragen

1. Skizziere kurz einige typische Kommunikationsprobleme und je eine mögliche Lösung..
2. Skizziere einige Eigenschaften typischer *Netztopologien*.
3. Welche besondere Bedeutung kommt IP zu?
4. Was ist ein *Remote Procedure Call (RPC)*, und welche Parameter wird man typischerweise dabei übergeben?
5. Was ist ein *(Kommunikations-)Protokoll*?