

8 Multiprozessorsysteme

- 1 Einführung
- 2 Prozesse und Threads
- 3 Speicherverwaltung
- 4 Dateisysteme
- 5 Eingabe und Ausgabe
- 6 Deadlocks
- 7 Virtualisierung und die Cloud
- 8 *Multiprozessorsysteme*
- 9 IT-Sicherheit
- 10 Fallstudie 1: Linux
- 11 Fallstudie 2: Windows
- 12 Entwurf von Betriebssystemen

8 Multiprozessorsysteme

8.1 Multiprozessoren

8.2 Multicomputer

8.3 Verteilte Systeme

8.1 Multiprozessoren

8.1.1 Hardware von Multiprozessoren

8.1.2 Betriebssystemarten für Multiprozessoren

8.1.3 Synchronisation in Multiprozessorsystemen

8.1.4 Multiprozessor-Scheduling

Multiprozessorsysteme

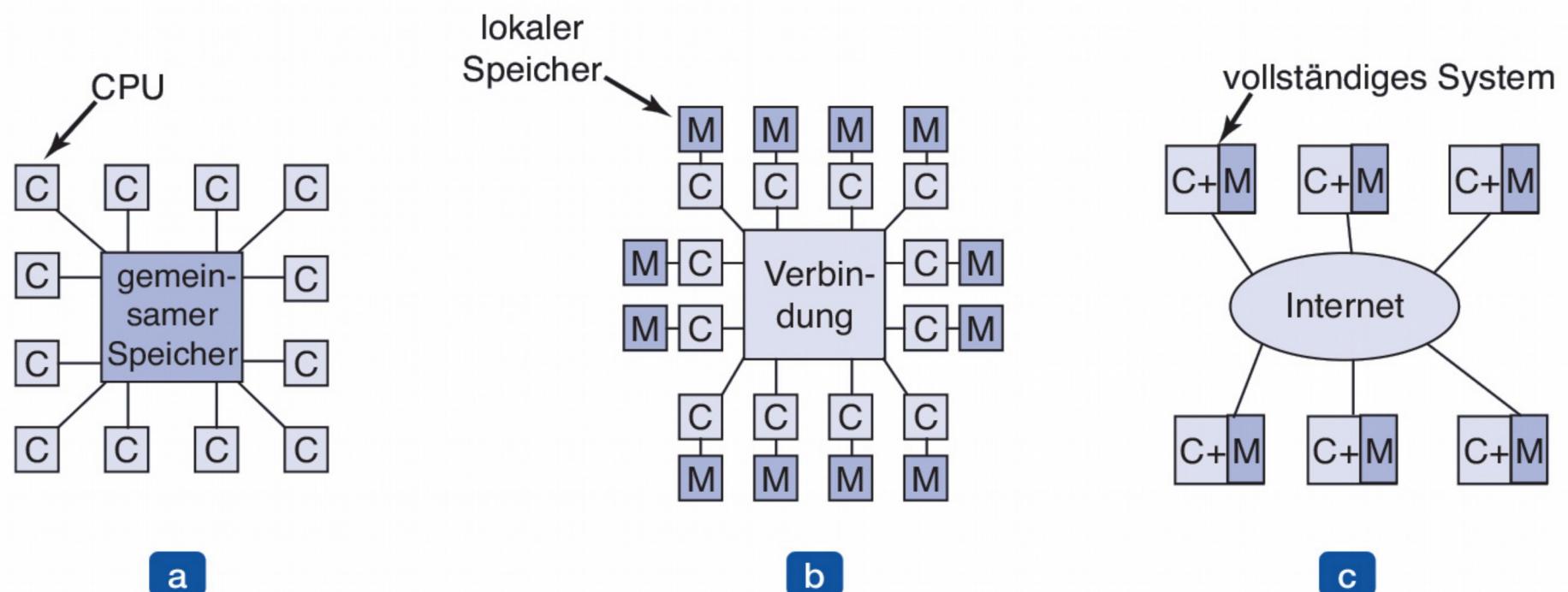


Abbildung 8.1: (a) Multiprozessorsystem mit gemeinsamem Speicher. (b) Multicomputer mit Nachrichtenaustausch. (c) Großräumig verteiltes System.

Hardware von Multiprozessoren

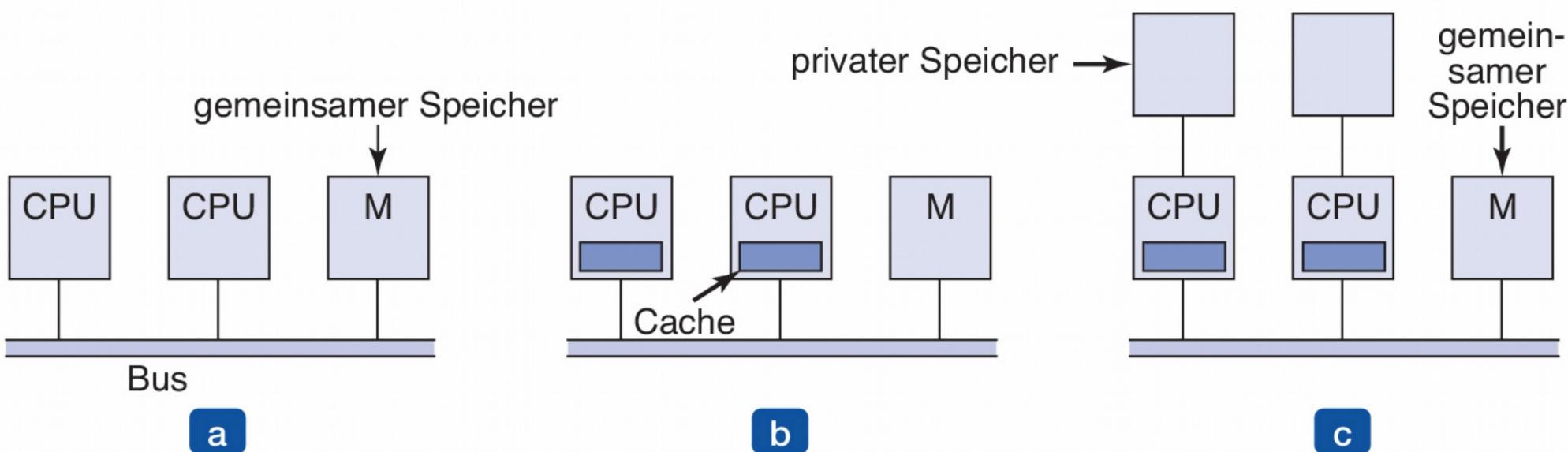


Abbildung 8.2: Drei busbasierte Multiprozessoren: (a) ohne Cache, (b) mit Cache (c), mit Cache und privatem Speicher.

UMA-Multiprozessoren mit Crossbar-Switches

Tanenbaum, A. S.; Bos, H.: Moderne Betriebssysteme. Pearson Studium 2016

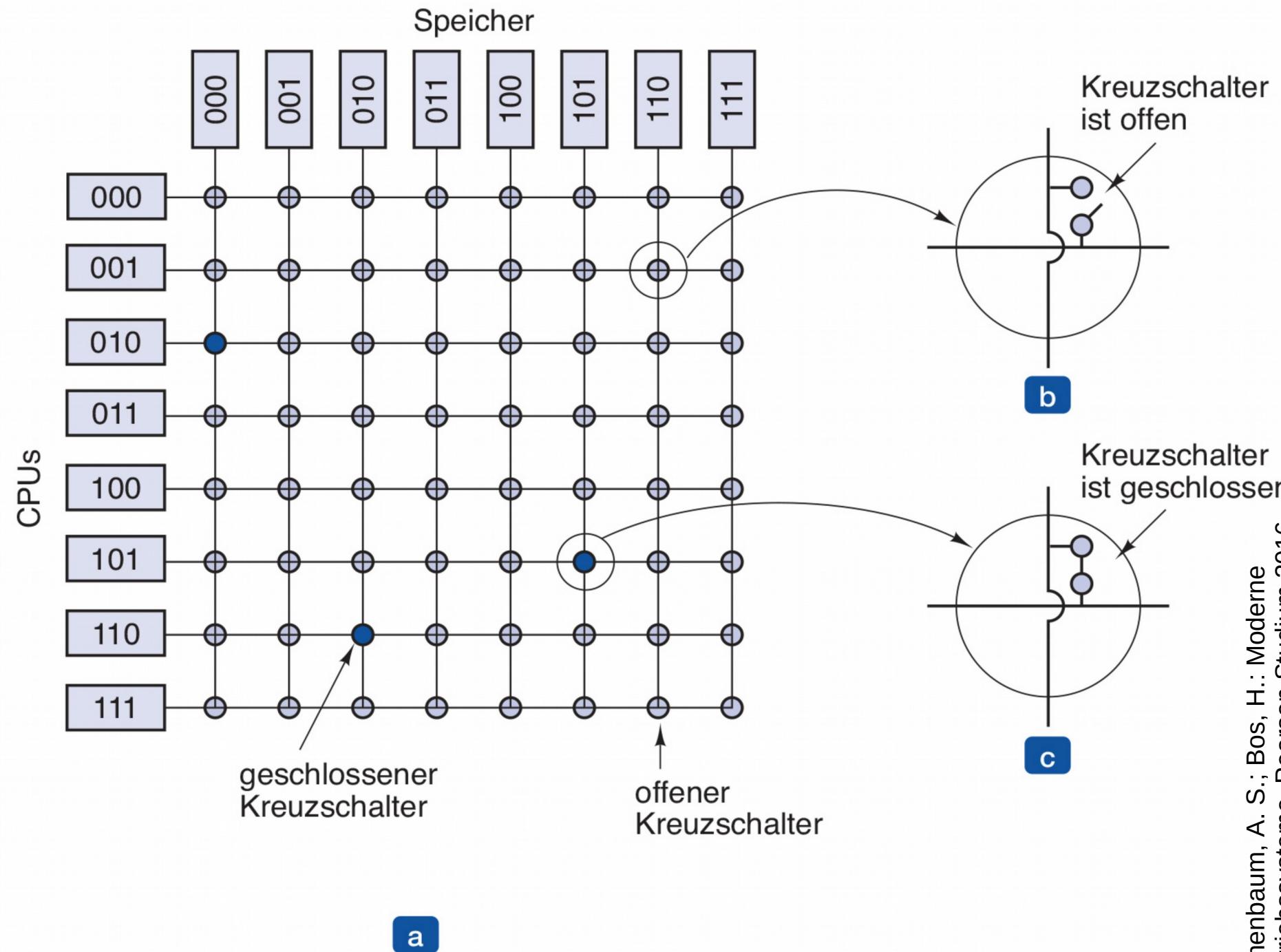


Abbildung 8.3: (a) 8x8-Koppelfeld; (b) offener Kreuzschalter; (c) geschlossener Kreuzschalter.

UMA Multiprozessoren mit mehrstufigen Vermittlungsnetzen (1)

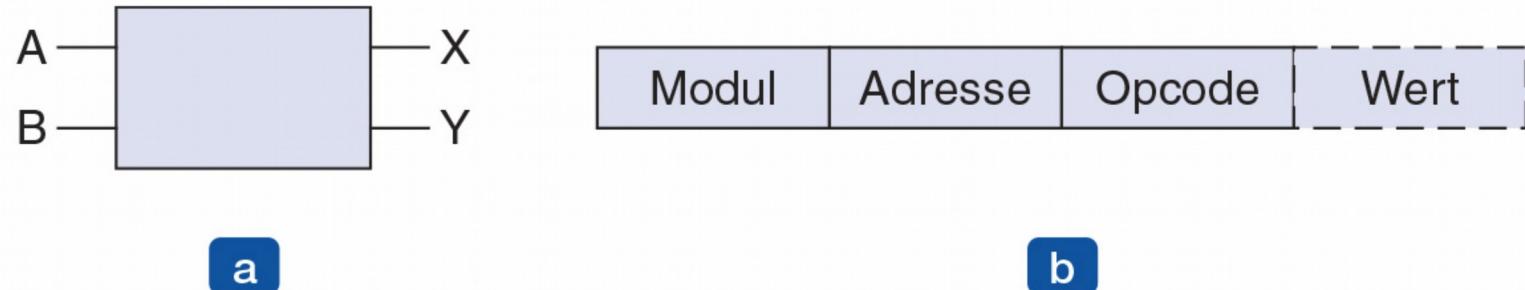


Abbildung 8.4: (a) 2×2-Schalter mit zwei Eingangsleitungen, A und B , und zwei Ausgangsleitungen, X und Y (b) Nachrichtenformat.

UMA Multiprozessoren mit mehrstufigen Vermittlungsnetzen (2)

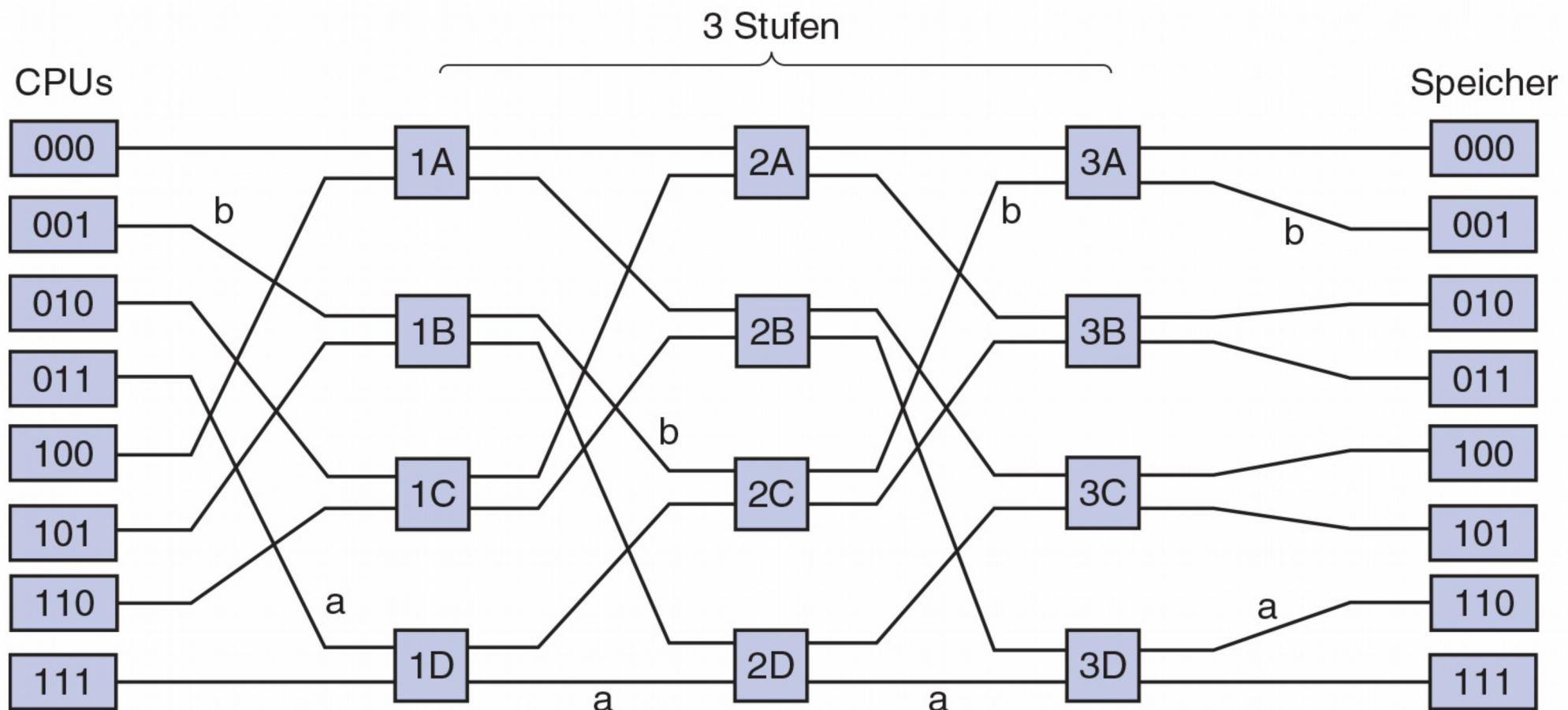


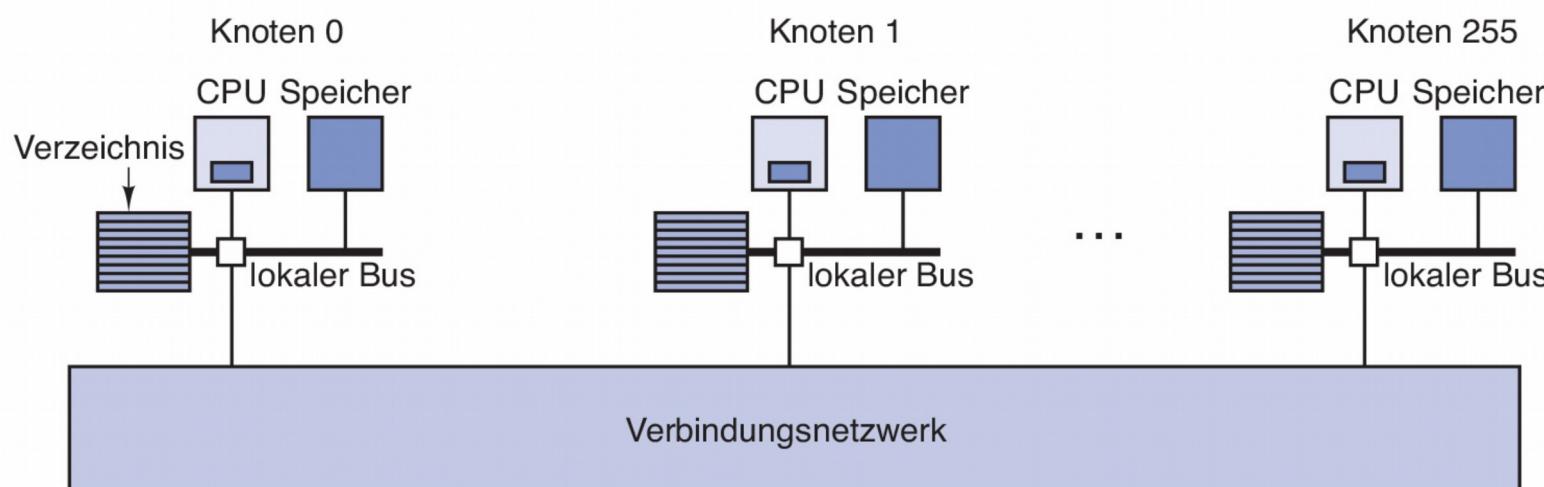
Abbildung 8.5: Omega-Schaltnetzwerk.

NUMA Multiprozessorsysteme (1)

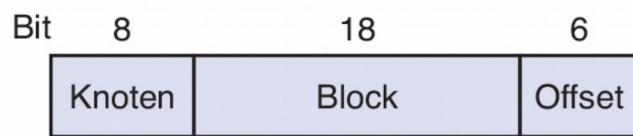
Merkmale von NUMA-Maschinen, in denen sie sich von anderen Multiprozessoren unterscheiden:

1. Es gibt einen einzigen Adressraum, der für alle CPUs sichtbar ist.
2. Der Zugriff auf den Remote-Speicher erfolgt über die Befehle LOAD und STORE.
3. Der Zugriff auf den Remote-Speicher ist langsamer als der Zugriff auf den lokalen Speicher.

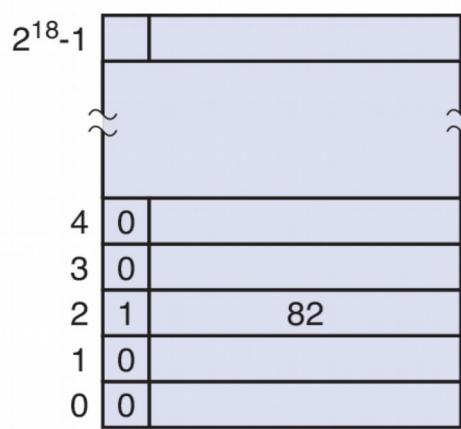
NUMA Multiprozessorsysteme (2)



a



b



c

Abbildung 8.6: (a) Verzeichnisbasierter Multiprozessor mit 256 Knoten. (b) Unterteilung einer 32-Bit-Speicheradresse in Felder. (c) Das Verzeichnis bei Knoten 36.

Betriebssystemarten für Multiprozessoren

Jeder Prozessor hat sein eigenes Betriebssystem

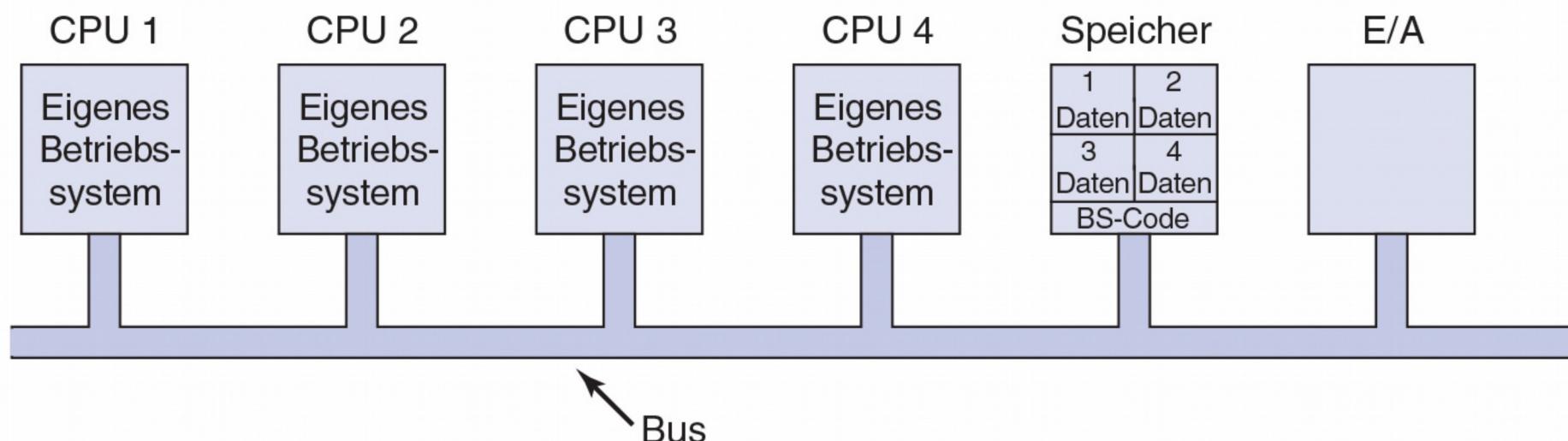


Abbildung 8.7: Aufteilung des Speichers auf vier CPUs unter Verwendung nur einer einzigen Kopie des Betriebssystemcodes. Kästen, die mit Daten beschriftet sind, stellen die privaten Daten des Betriebssystems für jede CPU dar.

Master-Slave-Multiprozessorsysteme

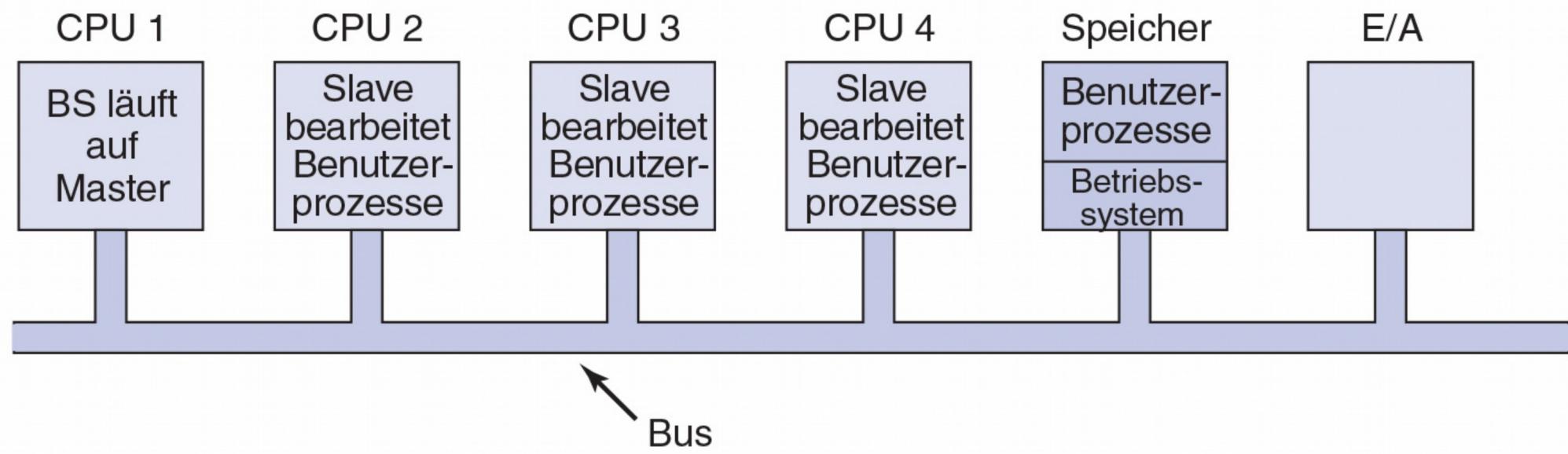


Abbildung 8.8: Master-Slave-Multiprozessormodell.

Symmetrische Multiprozessoren

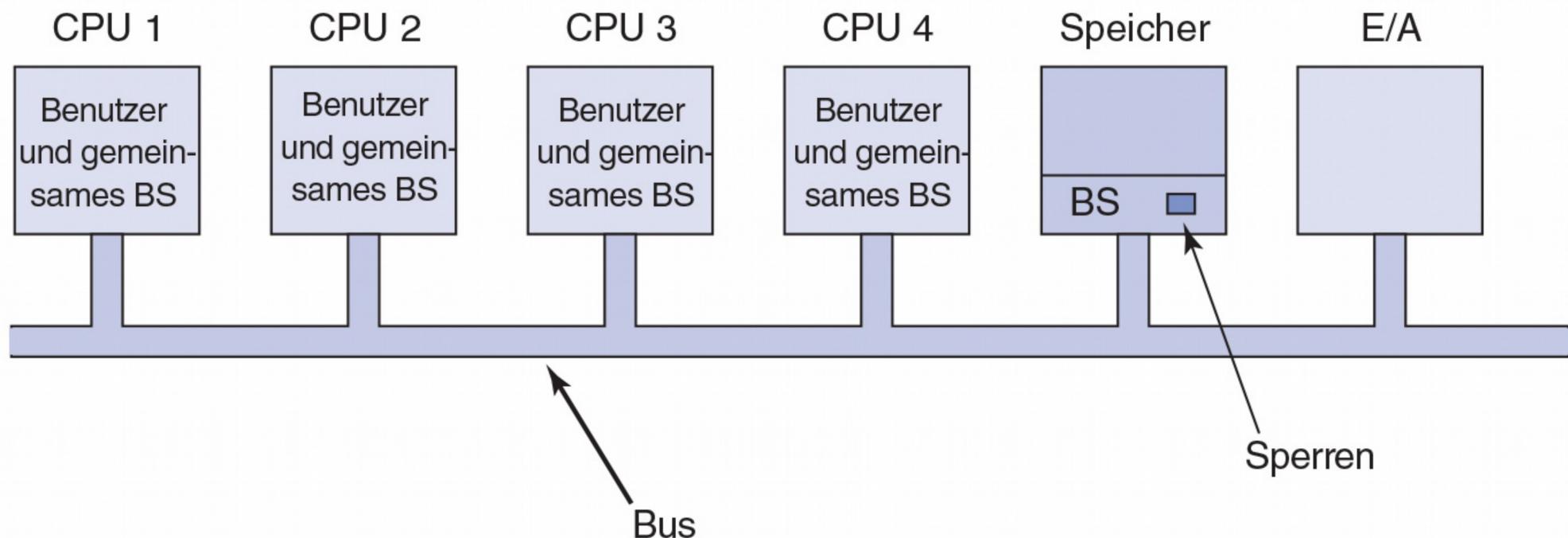


Abbildung 8.9: Das SMP-Multiprozessormodell.

Synchronisation in Multiprozessorsystemen (1)

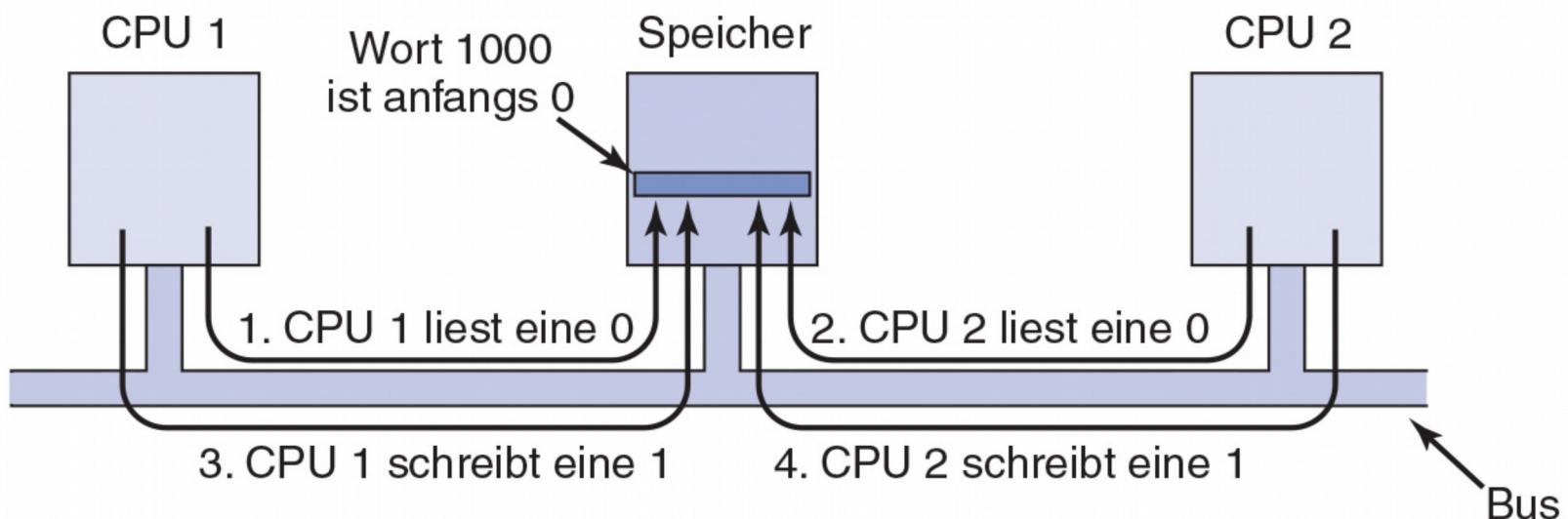


Abbildung 8.10: Der TSL-Befehl kann fehlschlagen, wenn der Bus nicht gesperrt werden kann. Diese vier Schritte zeigen eine Folge von Ereignissen, bei denen solch ein Fehler auftritt.

Synchronisation in Multiprozessorsystemen (2)

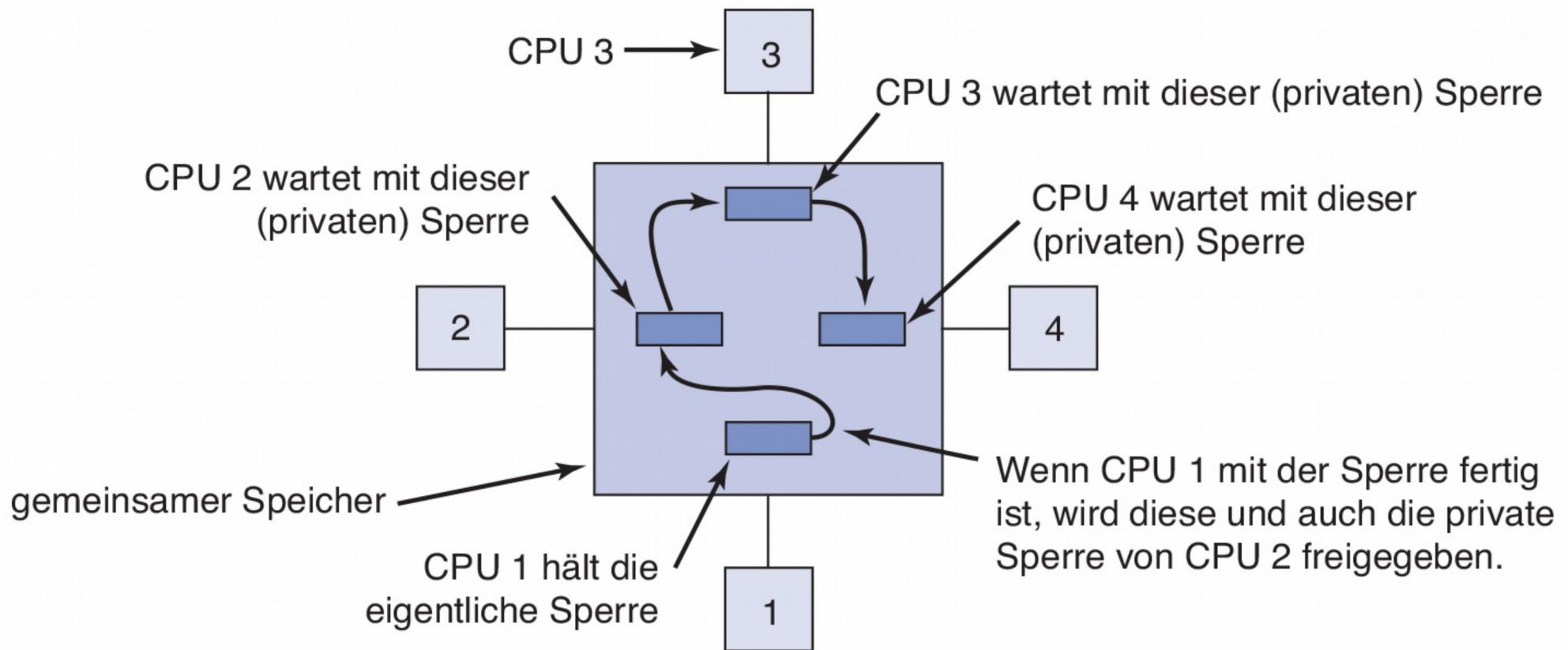


Abbildung 8.11: Einsatz von mehreren Sperren zur Vermeidung von Cache-Flattern.

Time Sharing

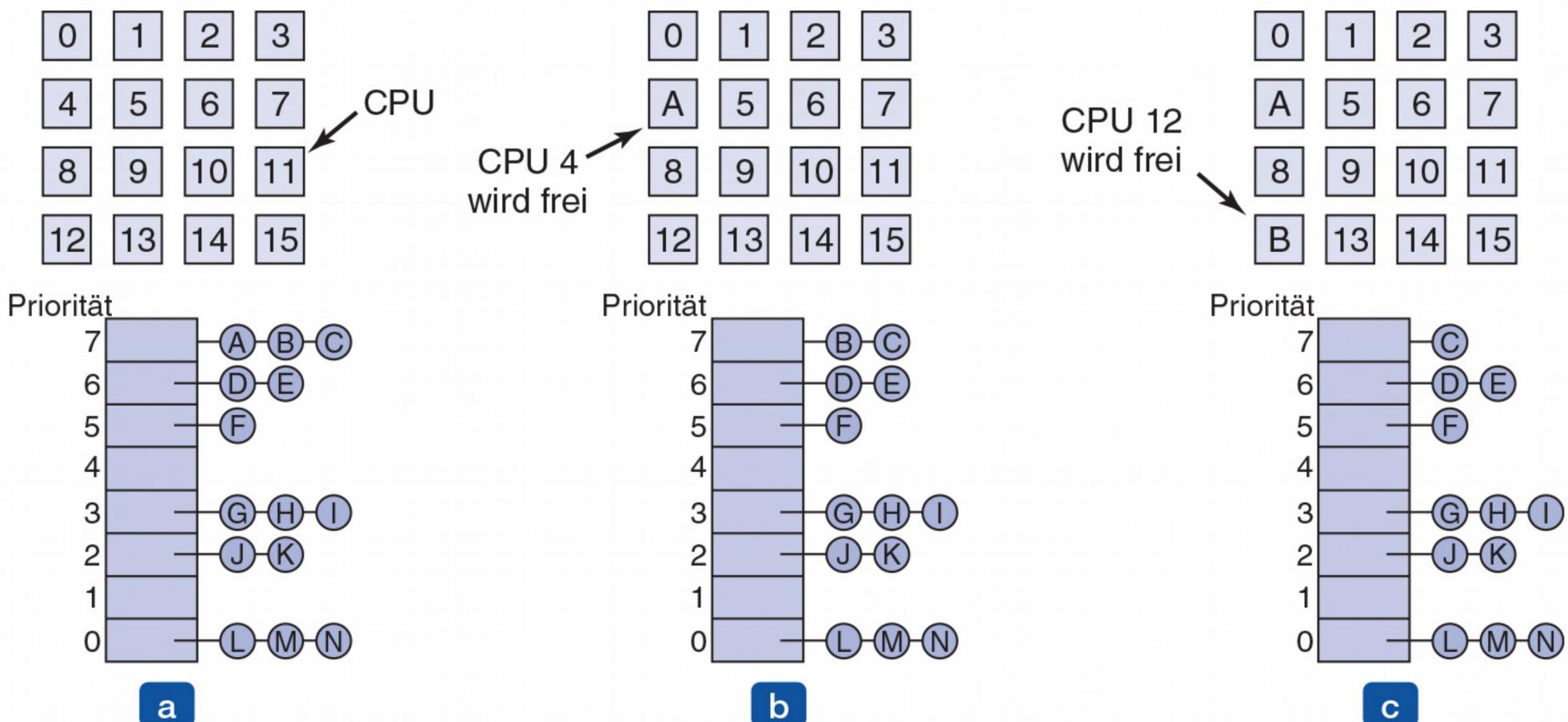


Abbildung 8.12: Verwendung einer einzigen Datenstruktur für das Scheduling auf Multiprozessorsystemen.

Space Sharing

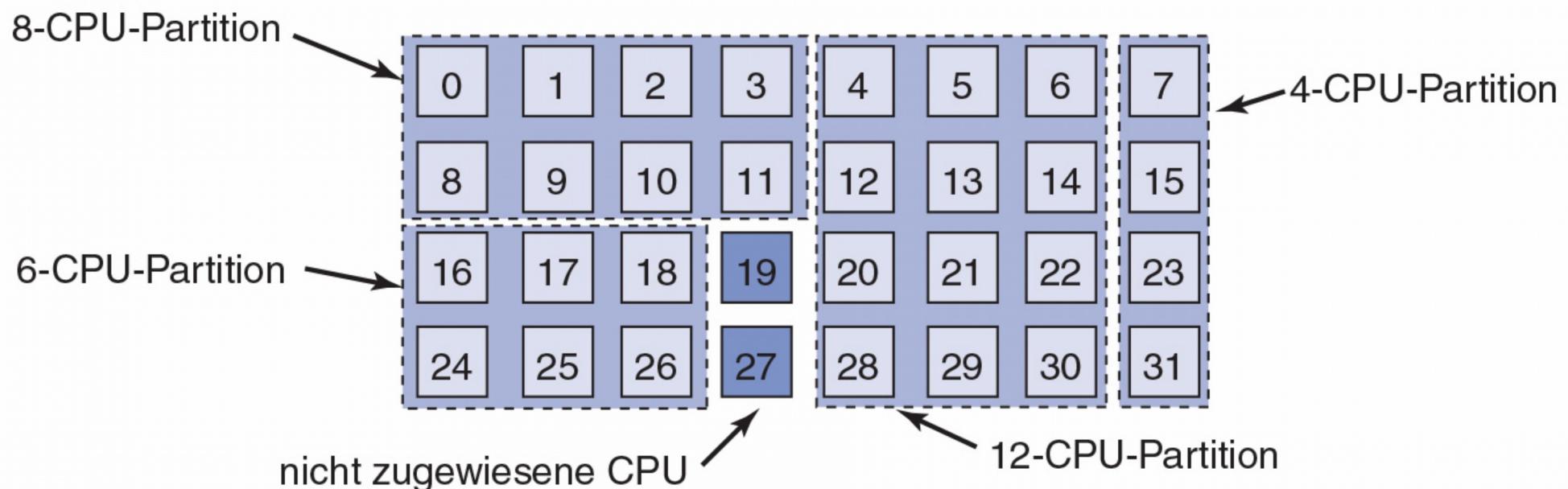


Abbildung 8.13: 32 CPUs, in vier Partitionen aufgeteilt, und zwei verfügbare CPUs.

Gang Scheduling (1)

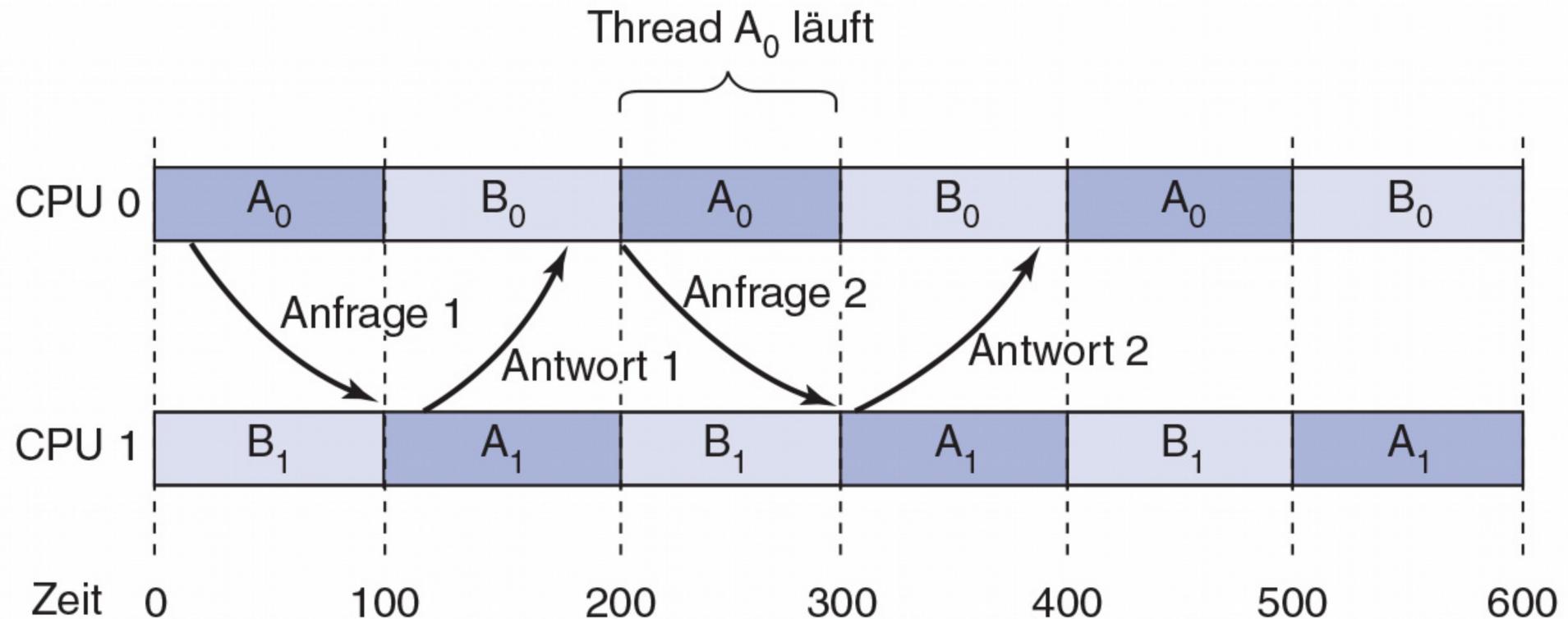


Abbildung 8.14: Kommunikation zwischen zwei Threads von A , die phasenweise verschoben sind.

Gang Scheduling (2)

Gang Scheduling hat drei Teile:

1. Gruppen verwandter Threads sind als eine Einheit geplant, eine Gruppe.
2. Alle Mitglieder einer Gruppe laufen gleichzeitig auf verschiedenen CPUs mit Timesharing.
3. Alle Gruppenmitglieder beginnen und beenden ihre Zeitscheiben zusammen.

Gang Scheduling (3)

		CPU					
		0	1	2	3	4	5
Zeitabschnitt	0	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅
	1	B ₀	B ₁	B ₂	C ₀	C ₁	C ₂
	2	D ₀	D ₁	D ₂	D ₃	D ₄	E ₀
	3	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆
	4	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅
	5	B ₀	B ₁	B ₂	C ₀	C ₁	C ₂
	6	D ₀	D ₁	D ₂	D ₃	D ₄	E ₀
	7	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆

Abbildung 8.15: Gang-Scheduling.

8.2 Multicomputer

- 8.2.1 Hardware von Multicomputern
- 8.2.2 Low-Level-Kommunikationssoftware
- 8.2.3 Kommunikationssoftware auf Benutzerebene
- 8.2.4 Entfernter Prozeduraufruf (RPC)
- 8.2.5 Distributed Shared Memory
- 8.2.6 Multicomputer-Scheduling
- 8.2.7 Lastausgleich

Verbindungstechnologien (1)

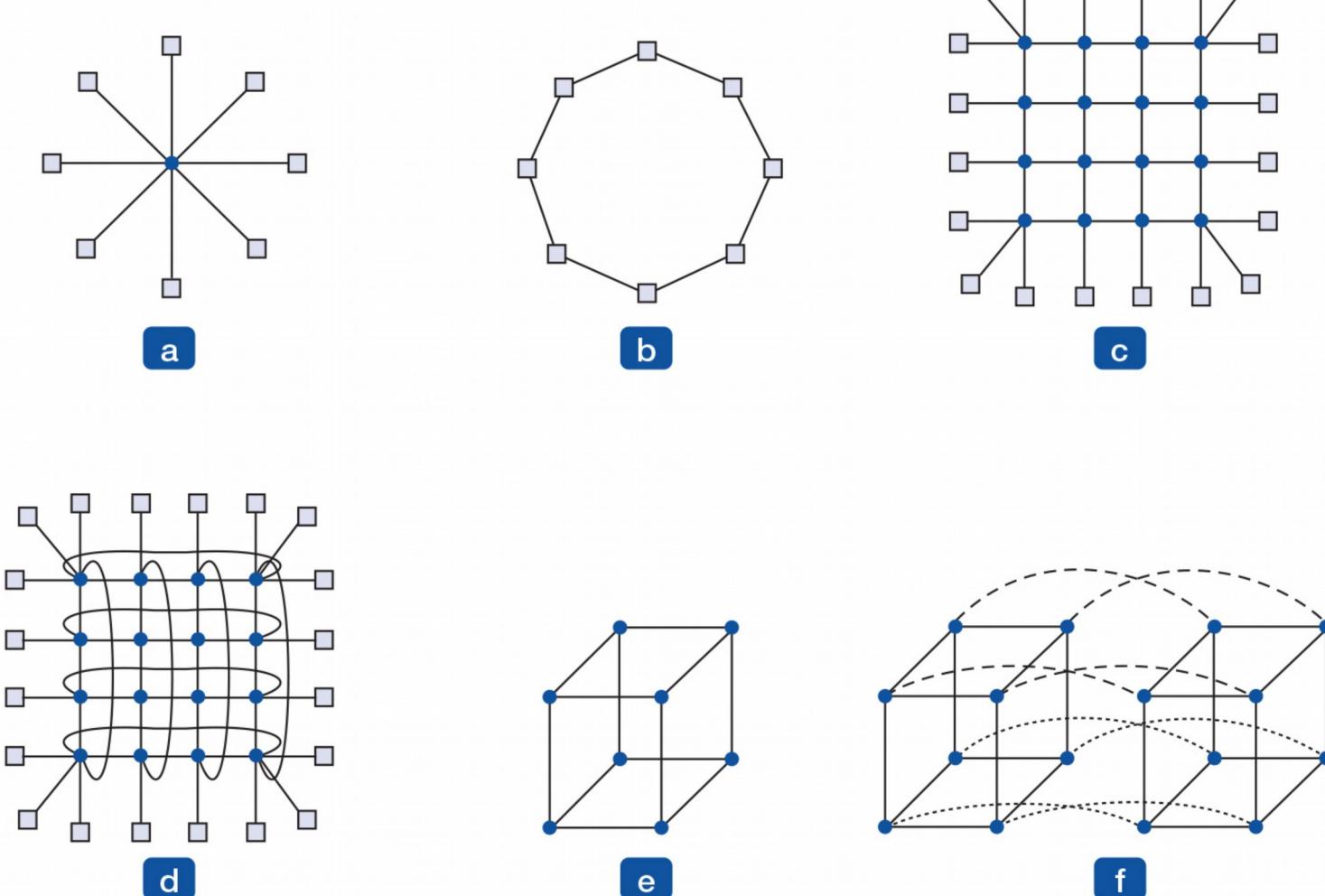


Abbildung 8.16: Verschiedene Verbindungstopologien: (a) einzelner Schalter, (b) Ring, (c) Gitter, (d) doppelter Torus, (e) Würfel, (f) 4-D-Hyperwürfel.

Verbindungstechnologien (2)

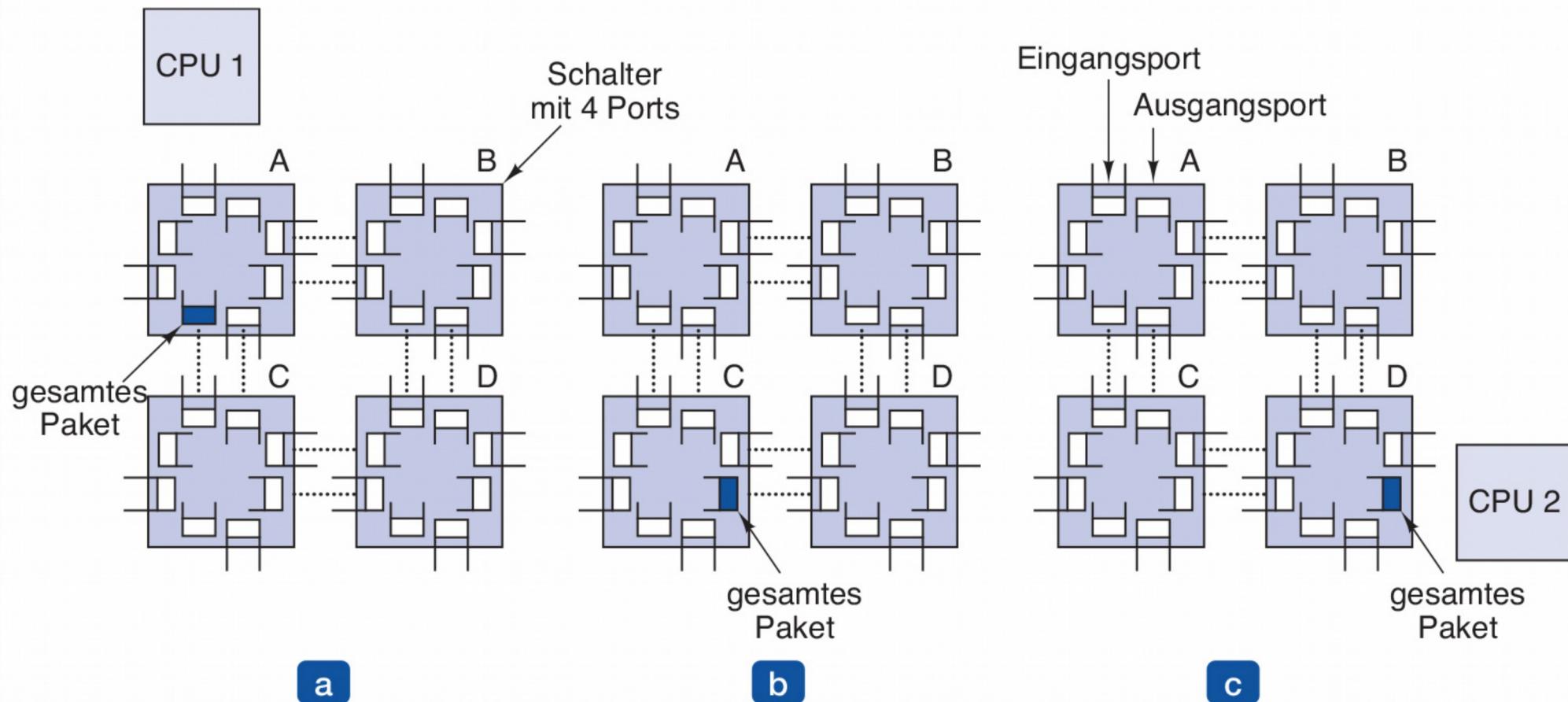


Abbildung 8.17: Store-and-Forward-Packet-Switching.

Netzwerkschnittstellen

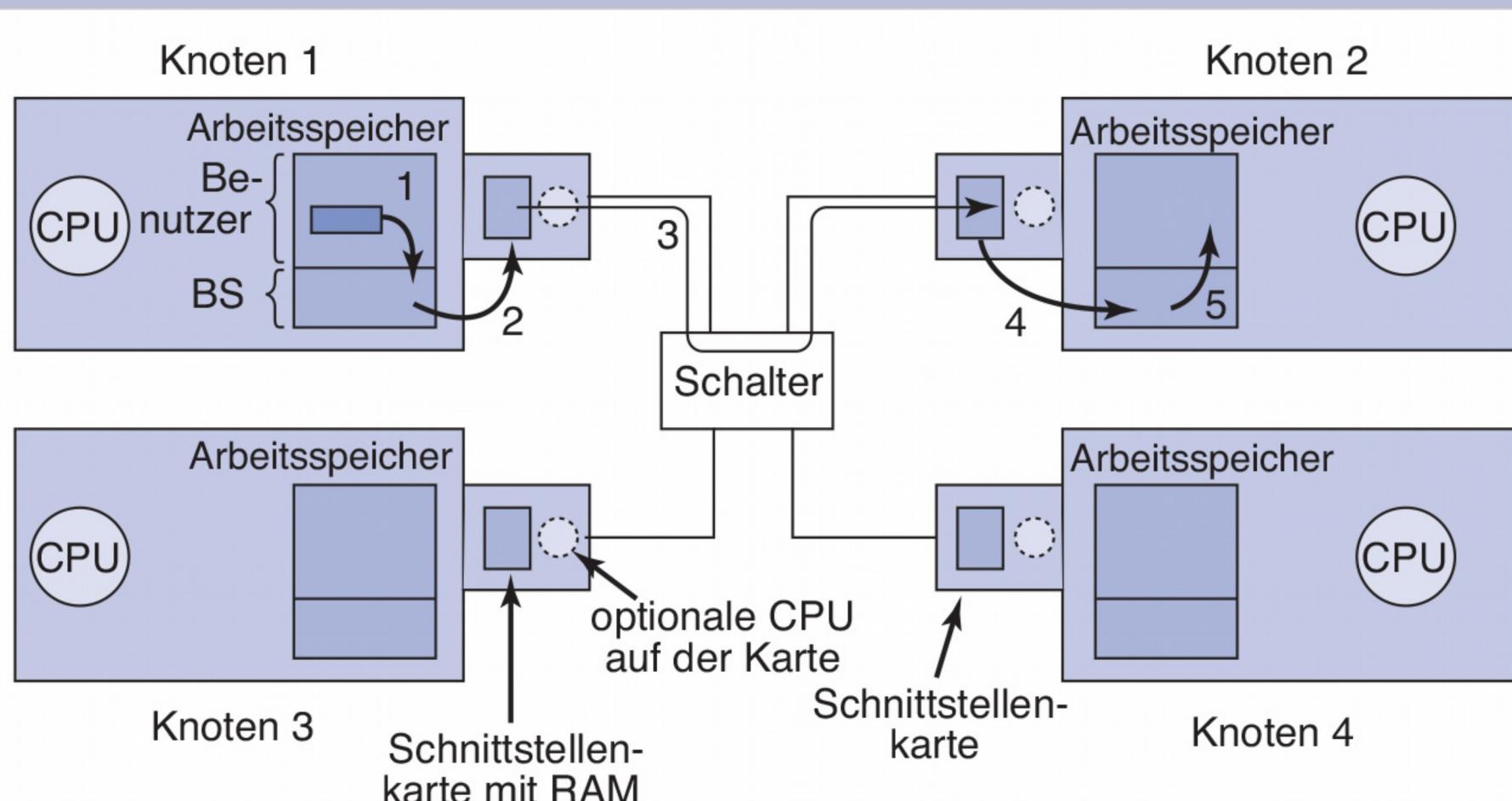
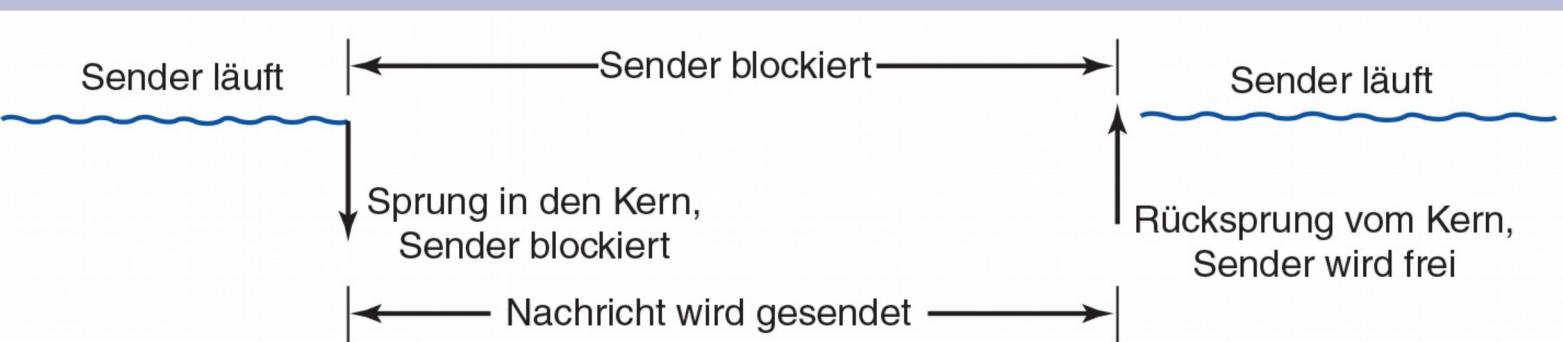


Abbildung 8.18: Position der Netzwerkschnittstelle innerhalb eines Multicomputers.

Blockierende und nicht blockierende Aufrufe (1)



a

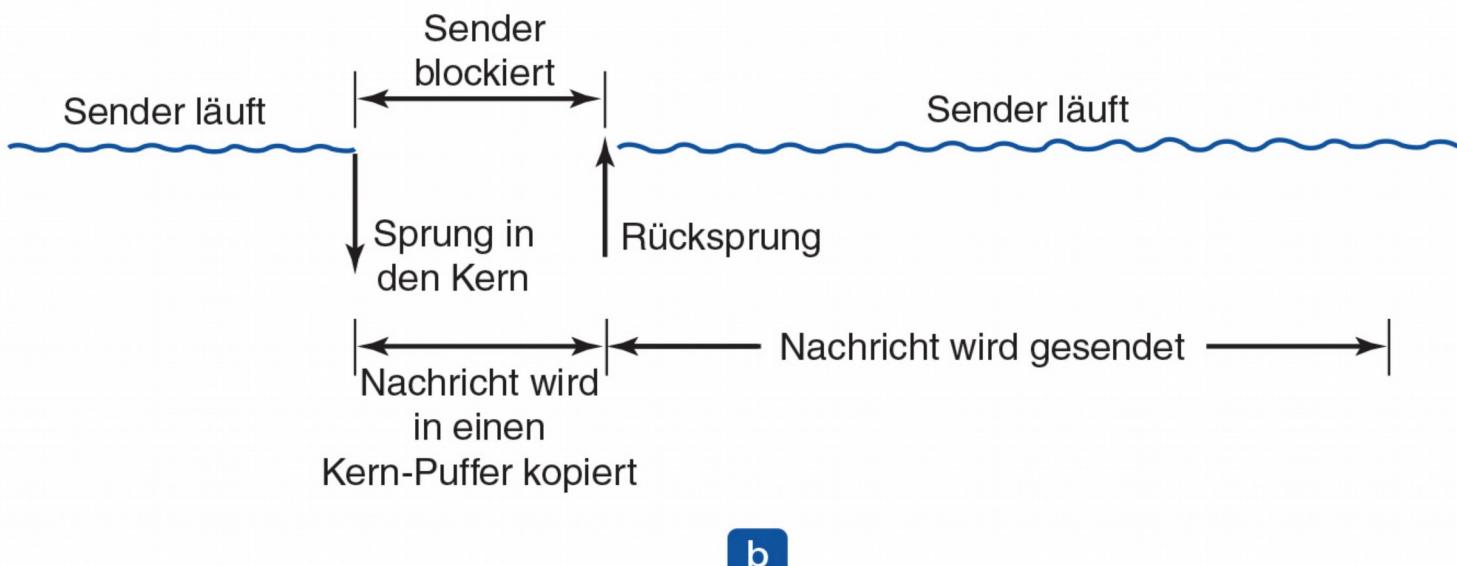


Abbildung 8.19: (a) Blockierender *send*-Aufruf. (b) Nicht blockierender *send*-Aufruf.

Blockierende und nicht blockierende Aufrufe (2)

Wahlmöglichkeiten auf der Sendeseite:

1. Senden blockieren (CPU-Leerlauf während der Übertragung).
2. Nicht blockierendes Senden mit Kopie (CPU-Zeit wird für die zusätzliche Kopie verschwendet).
3. Nicht blockierendes Senden mit Interrupt (erschwert die Programmierung).
4. Kopie beim Schreiben (extra Kopie wird wahrscheinlich irgendwann benötigt).

Remote Procedure Call (RPC)

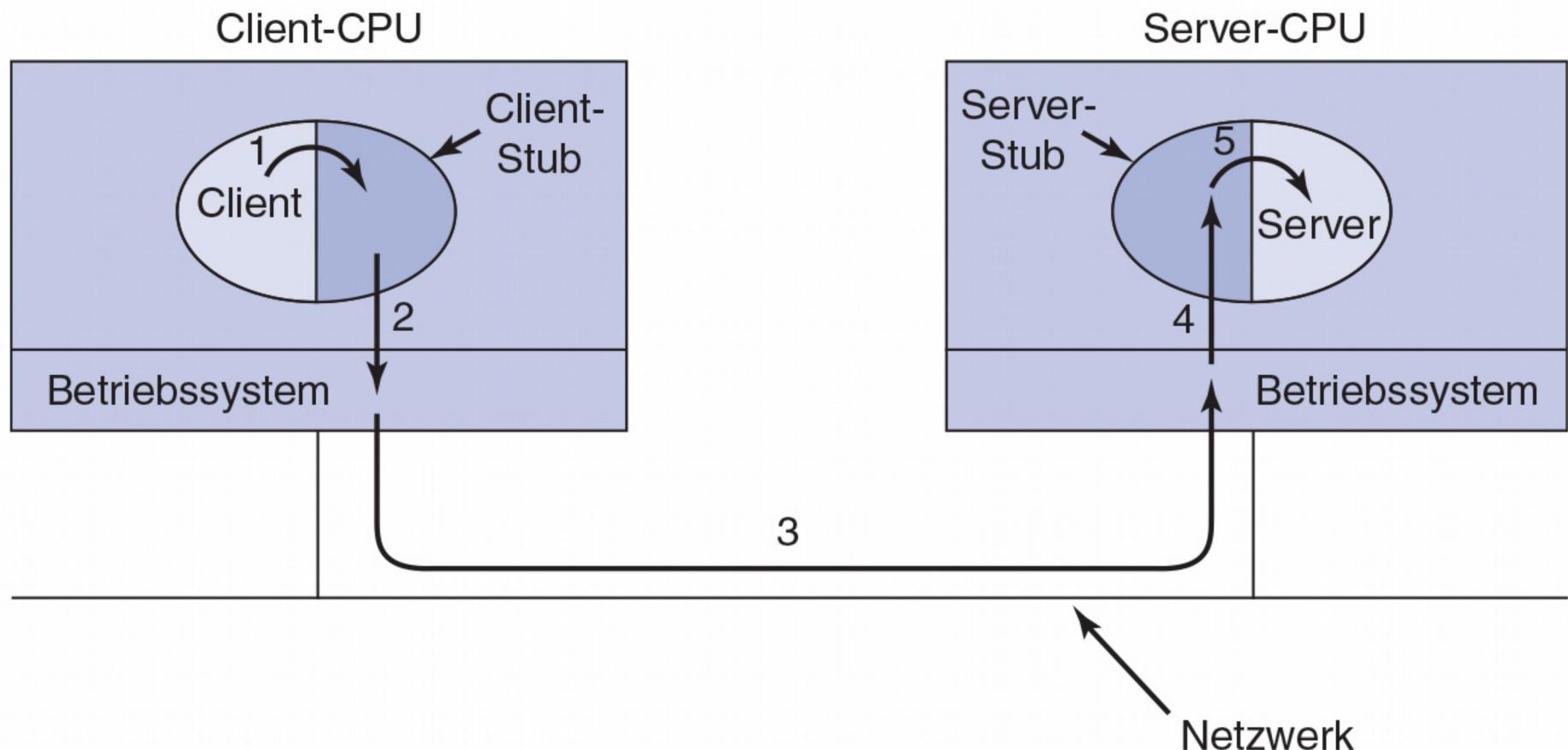


Abbildung 8.20: Schritte auf dem Weg zum entfernten Prozeduraufruf. Die Stubs sind dunkler dargestellt.

Distributed Shared Memory (1)

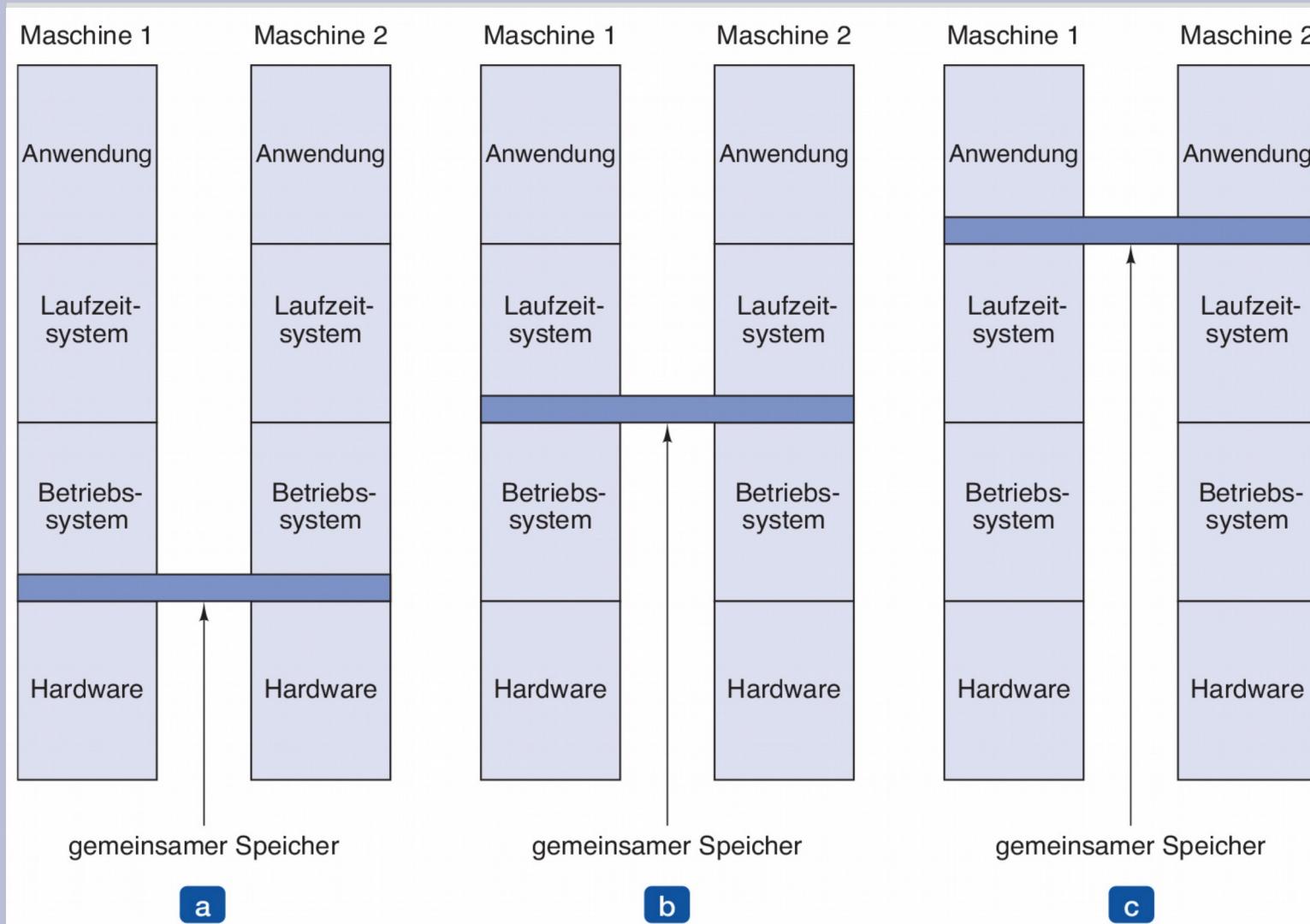
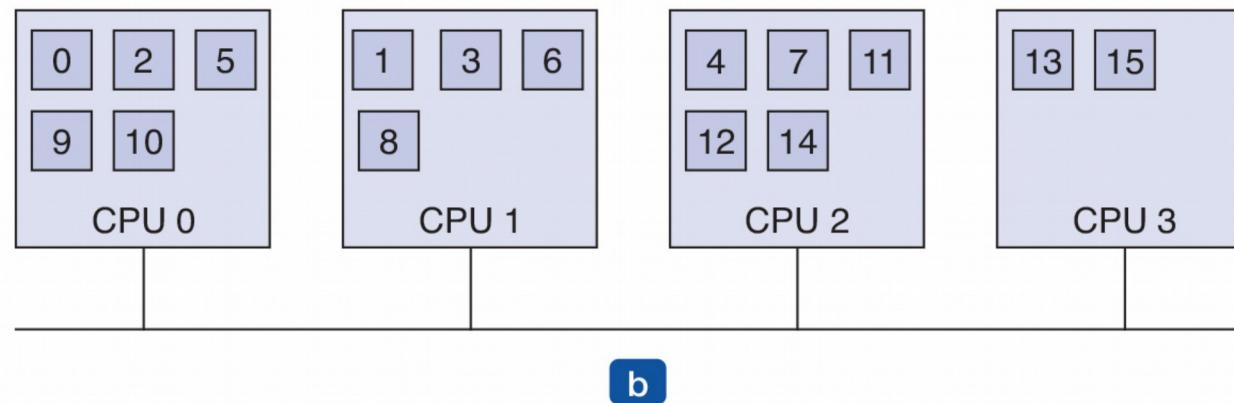
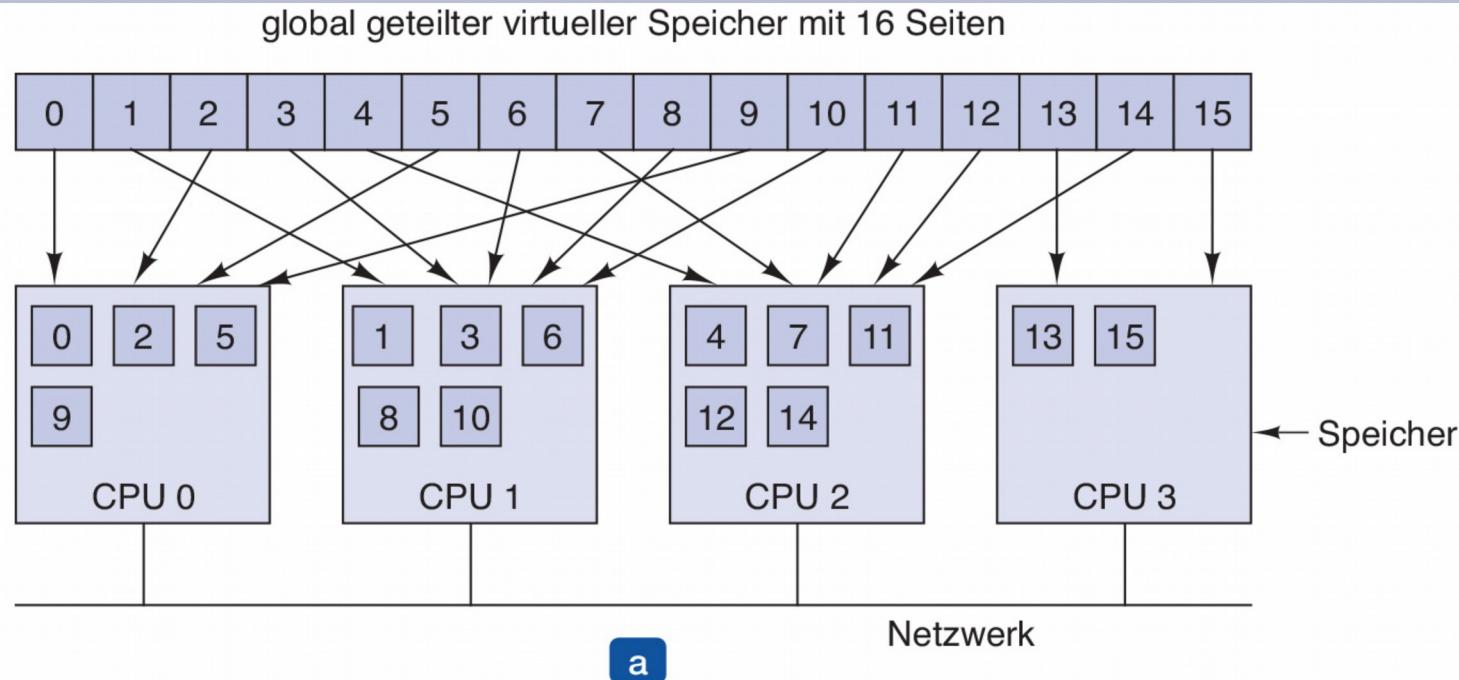


Abbildung 8.21: Verschiedene Schichten, in denen der gemeinsame Speicher implementiert werden kann: (a) Hardware, (b) Betriebssystem, (c) Software auf der Benutzerebene.

Distributed Shared Memory (2)



Distributed Shared Memory (3)

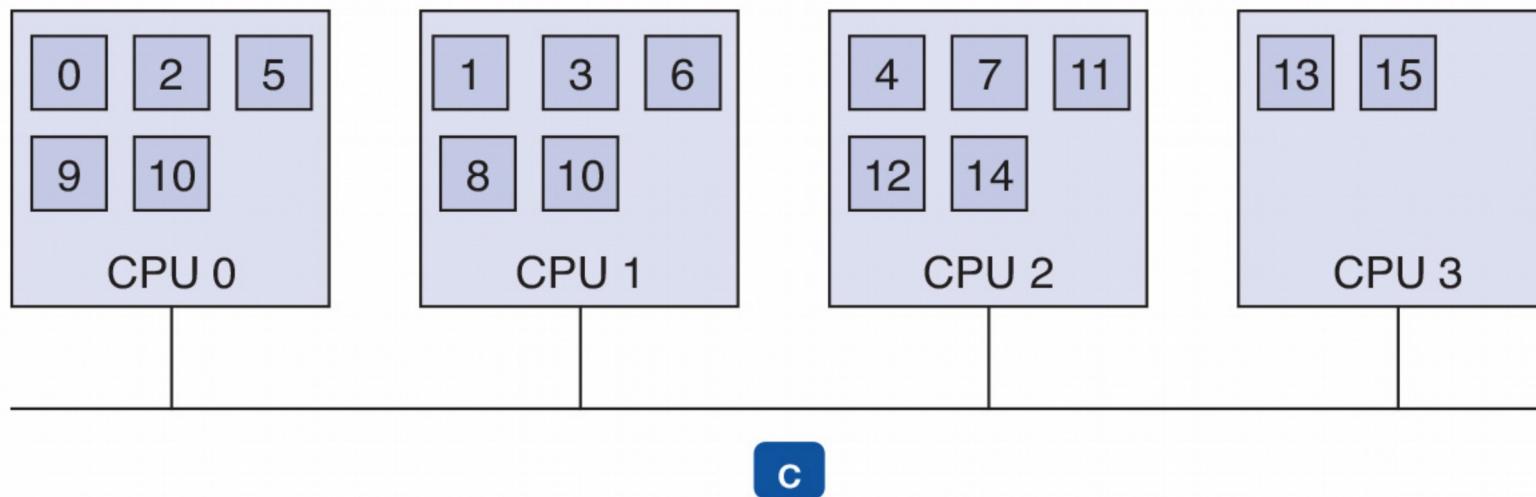


Abbildung 8.22: (a) Speicherseiten des auf vier Maschinen verteilten Adressraums. (b) Situation, nachdem CPU 0 auf Seite 10 zugegriffen hat und die Seite dorthin verschoben wurde. (c) Situation, wenn Seite 10 nur zum Lesen ist und Replikation verwendet wurde.

False Sharing

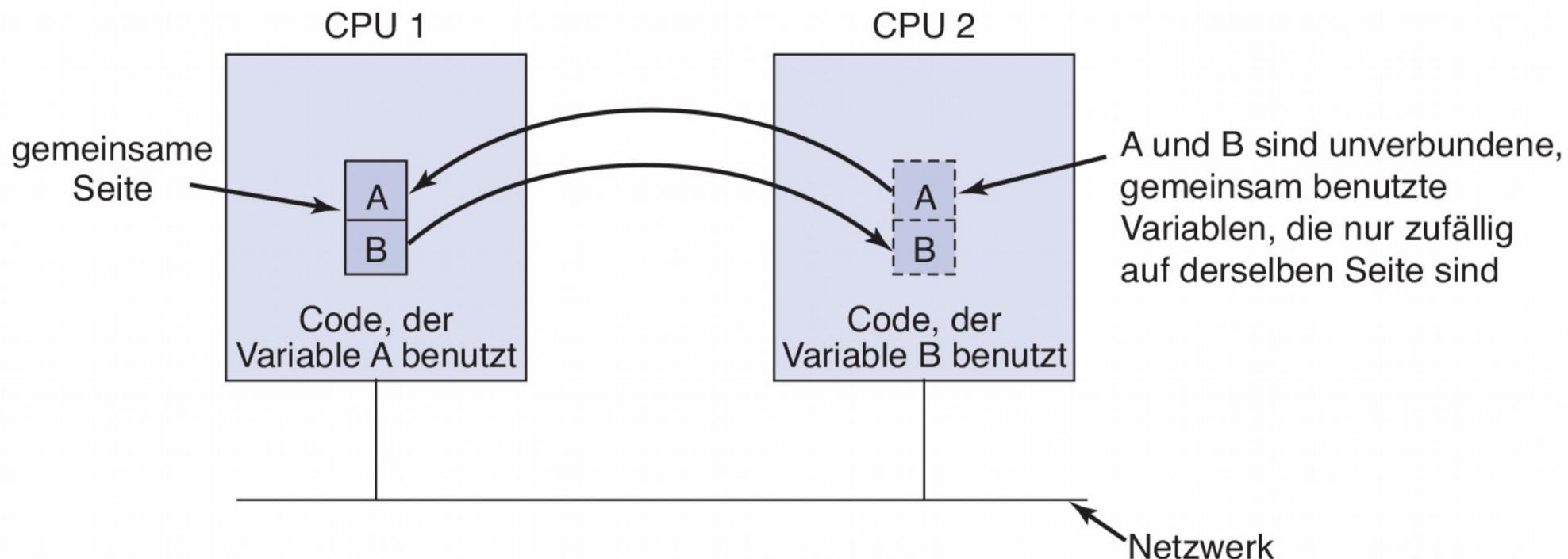


Abbildung 8.23: False-Sharing einer Seite mit zwei unabhängigen Variablen.

Ein deterministischer, graphentheoretischer Algorithmus

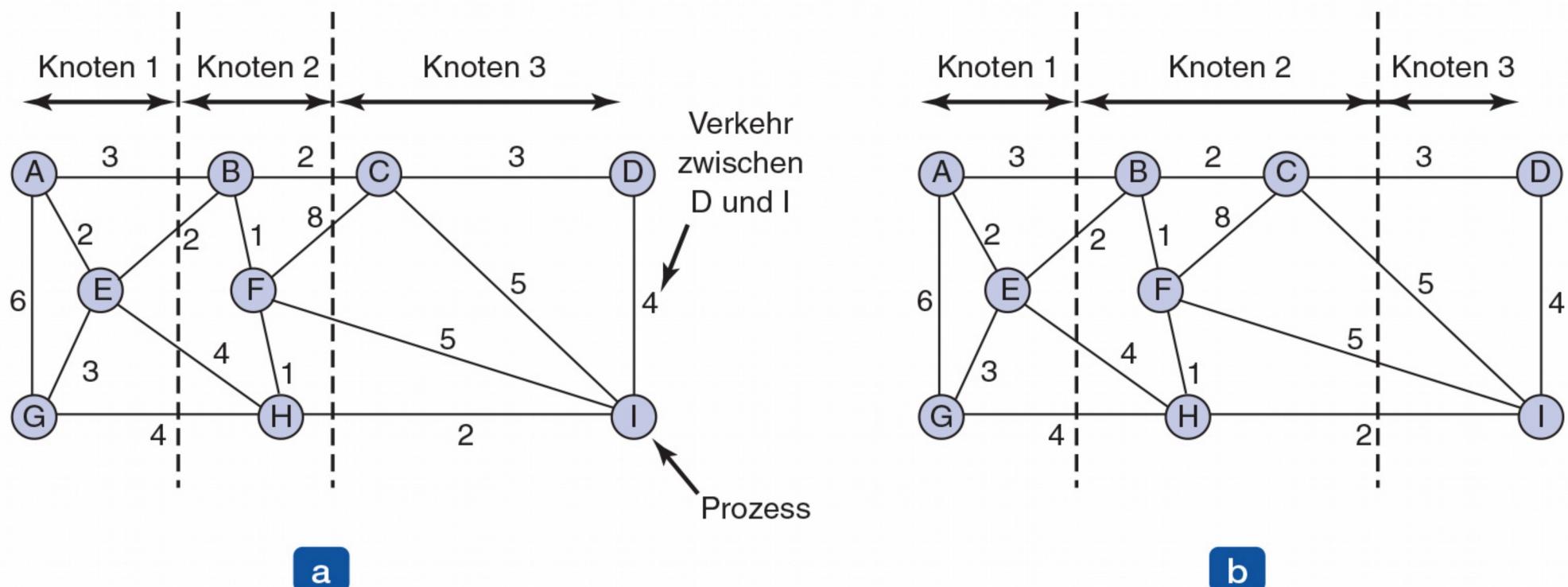


Abbildung 8.24: Zwei Möglichkeiten, um neun Prozesse auf drei Knoten zu verteilen.

Ein heuristischer, senderinitierter verteilter Algorithmus

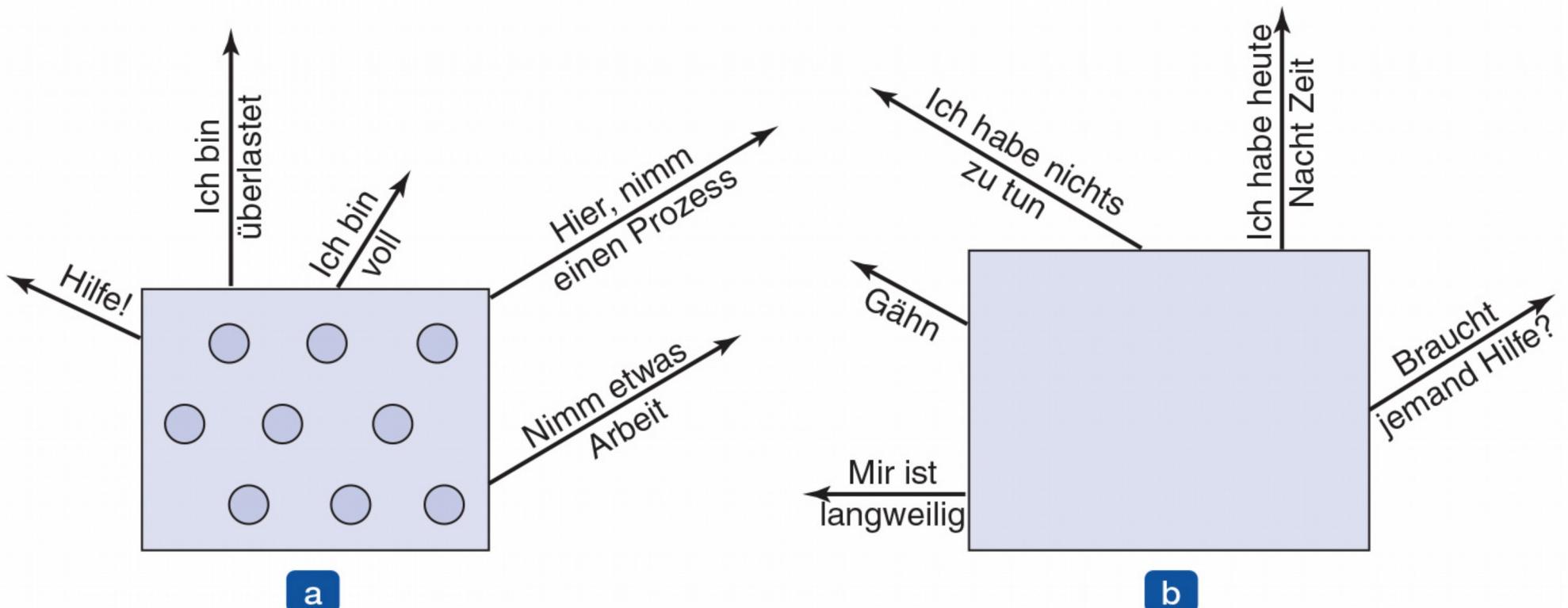


Abbildung 8.25: (a) Ein überlasteter Knoten sucht nach einem weniger belasteten Knoten, um Prozesse abgeben zu können. (b) Ein freier Knoten sucht nach Arbeit.

8.3 Verteilte Systeme

8.3.1 Netzwerkhardware

8.3.2 Netzwerkdienste und -protokolle

8.3.3 Dokumentenbasierte Middleware

8.3.4 Dateisystembasierte Middleware

8.3.5 Objektbasierte Middleware

8.3.6 Koordinationsbasierte Middleware

Verteilte Systeme (1)

Element	Multiprozessor	Multicomputer	Verteiltes System
Knotenkonfiguration	CPU	CPU, Speicher, Netzwerk	Vollständiger Rechner
Knotenperipherie	Alle gemeinsam genutzt	Gemeinsam, außer eventuell Platte	Vollständiger Satz pro Knoten
Standort	Gleiches Gehäuse	Gleicher Raum	Eventuell weltweit
Kommunikation der Knoten	Gemeinsamer Speicher	Spezielle Verbindung	Herkömmliches Netz
Betriebssysteme	Eines, gemeinsam genutzt	Viele, dasselbe	Eventuell alle verschieden
Dateisysteme	Eines, gemeinsam genutzt	Eines, gemeinsam genutzt	Eigenes pro Knoten
Verwaltung	Eine Organisation	Eine Organisation	Viele Organisationen

Abbildung 8.26: Vergleich dreier Arten von Mehrprozessorsystemen.

Verteilte Systeme (2)

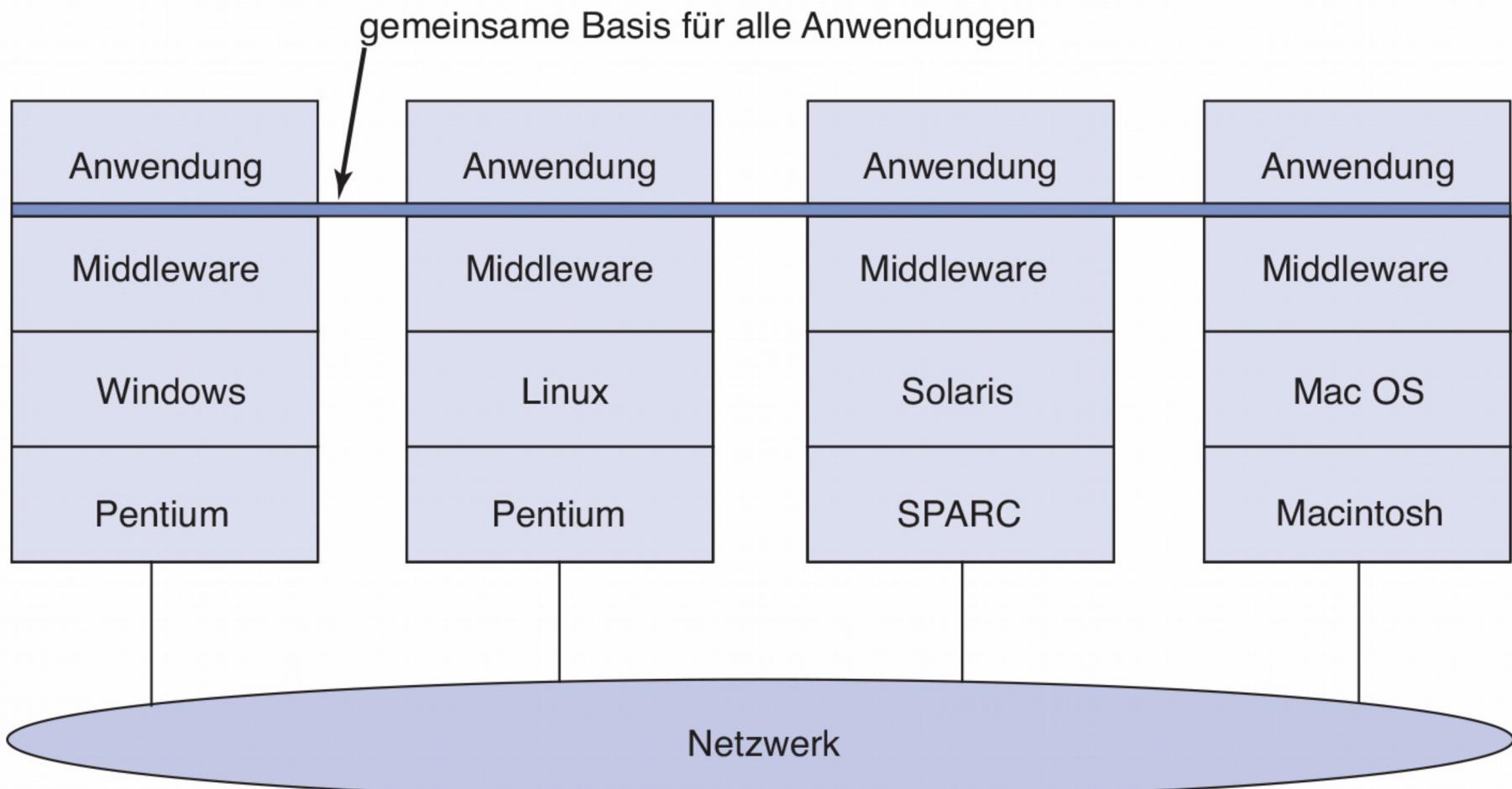


Abbildung 8.27: Position der Middleware in einem verteilten System.

Netzwerk Hardware Ethernet

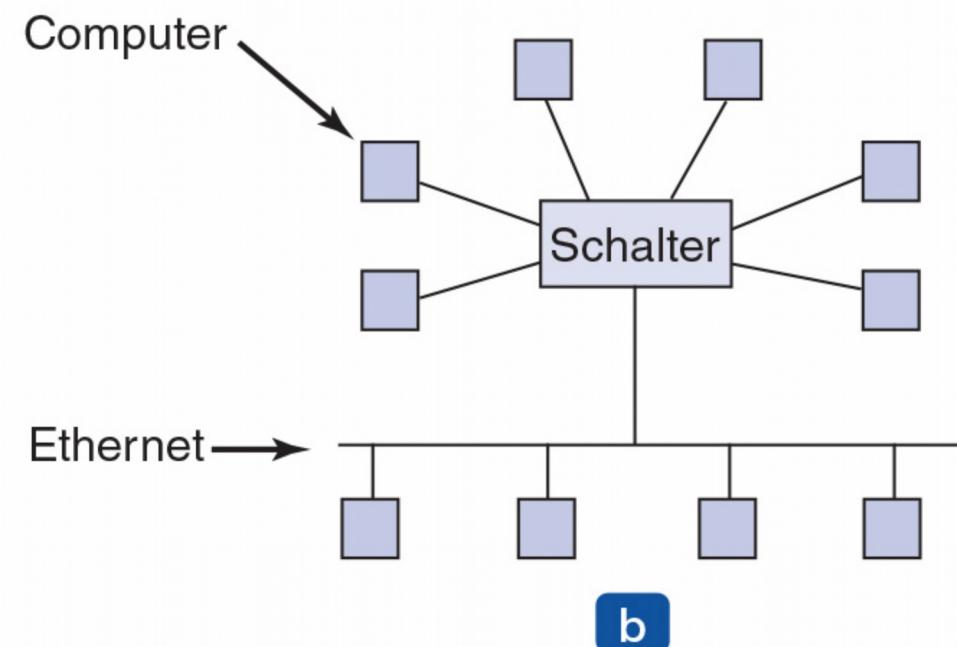
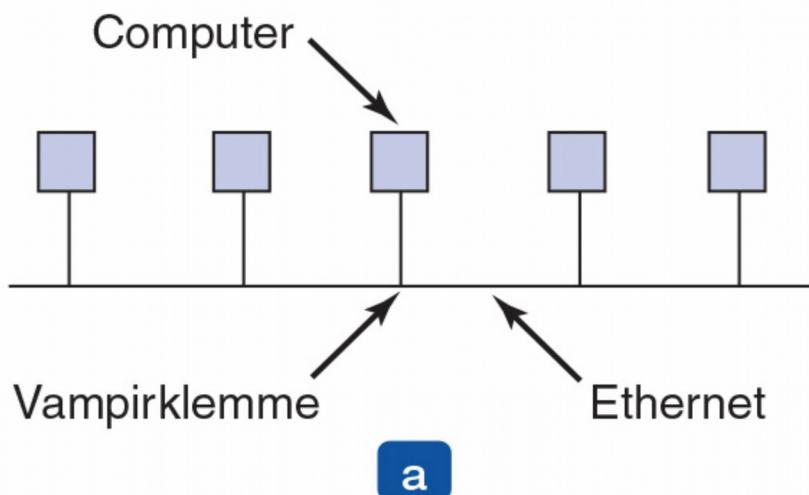


Abbildung 8.28: (a) Klassisches Ethernet. (b) Switched-Ethernet.

Netzwerk Hardware Ethernet

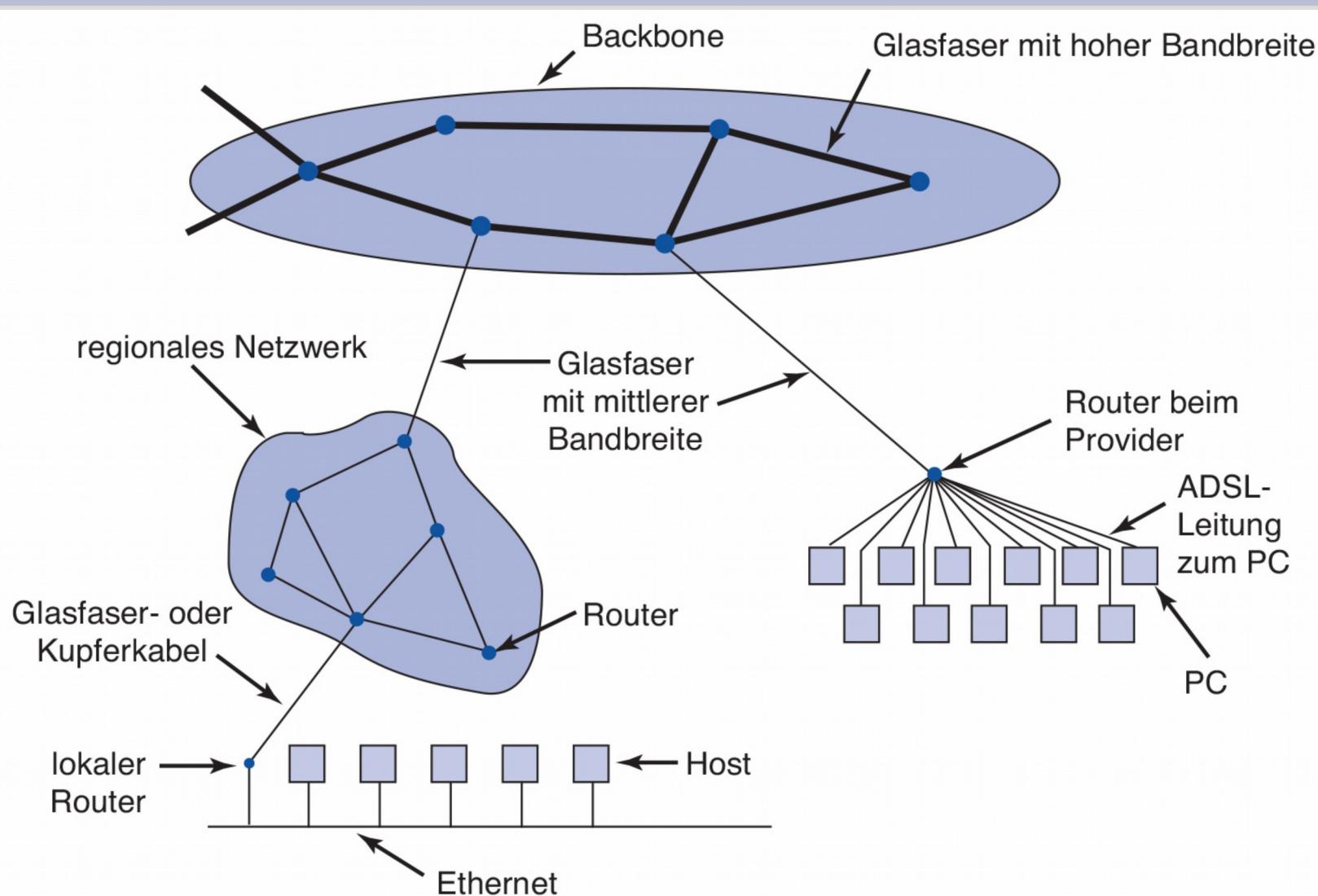


Abbildung 8.29: Ausschnitt des Internets.

Netzwerkdienste

	Dienst	Beispiel
verbindungsorientiert	Zuverlässiger Nachrichtenstrom	Folge von Buchseiten
	Zuverlässiger Zeichenstrom	Entfernter Login
	Unzuverlässige Verbindung	Digitalisierte Sprache
verbindungslos	Unzuverlässiges Datagramm	Netzwerk-Testpakete
	Bestätigtes Datagramm	E-Mail mit Empfangsbestätigung
	Anfrage/Antwort	Datenbankanfrage

Abbildung 8.30: Sechs verschiedene Typen von Netzwerkdiensten.

Netzwerkprotokolle

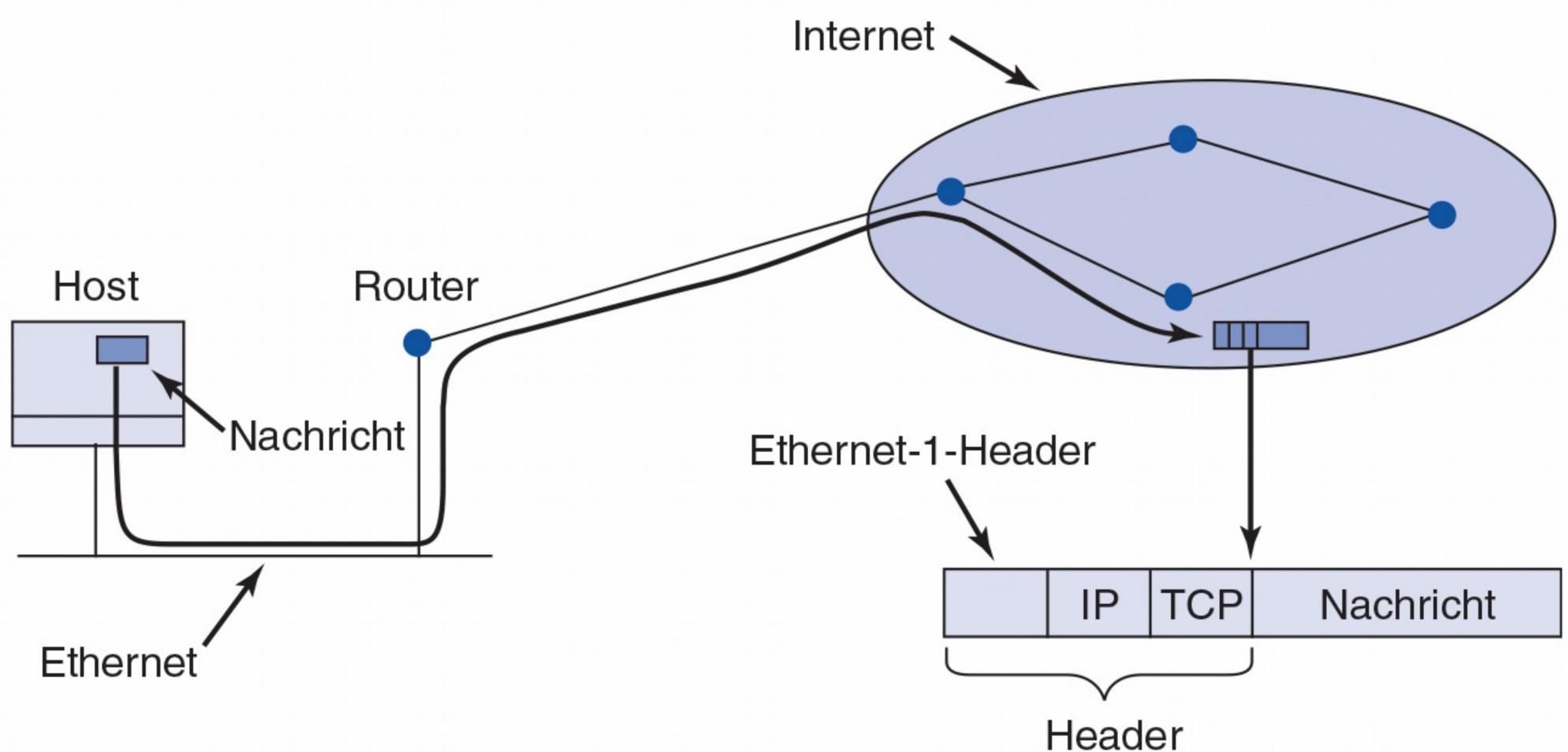


Abbildung 8.31: Die Anhäufung der Paket-Header.

Dokumentenbasierte Middleware (1)

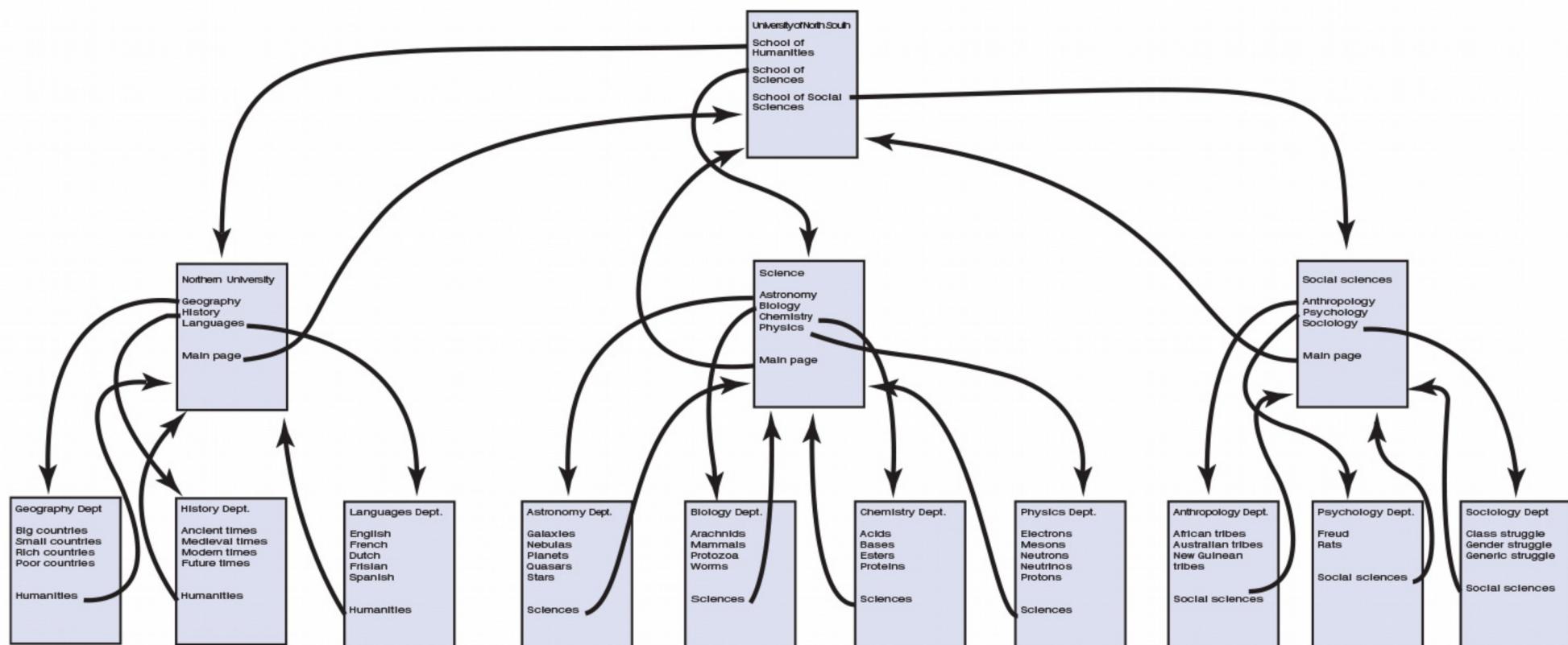


Abbildung 8.32: Das Web ist ein großer gerichteter Graph von Dokumenten.

Dokumentenbasierte Middleware (2)

Eingabe im Browser, um die Seite zu erhalten:

<http://www.minix3.org/getting-started/index.html>

1. Browser fragt DNS nach der IP-Adresse von
www.minix3.org
2. DNS antwortet mit 66.147.238.215
3. Der Browser stellt eine TCP-Verbindung zu Port 80 unter 66.147.238.215 her
4. Der Browser sendet eine Anfrage nach der Datei „getting-started / index.html“

Dokumentenbasierte Middleware (3)

5. Der www.minix3.org Server sendet die Datei „getting-started / index.html“
6. Der Browser zeigt den gesamten Text der Datei „getting-started / index.html“ an.
7. Der Browser ruft auch alle Bilder auf der Seite ab und zeigt sie an
8. Die TCP-Verbindung wird freigegeben

Dateisystembasierte Middleware Transfer Modell

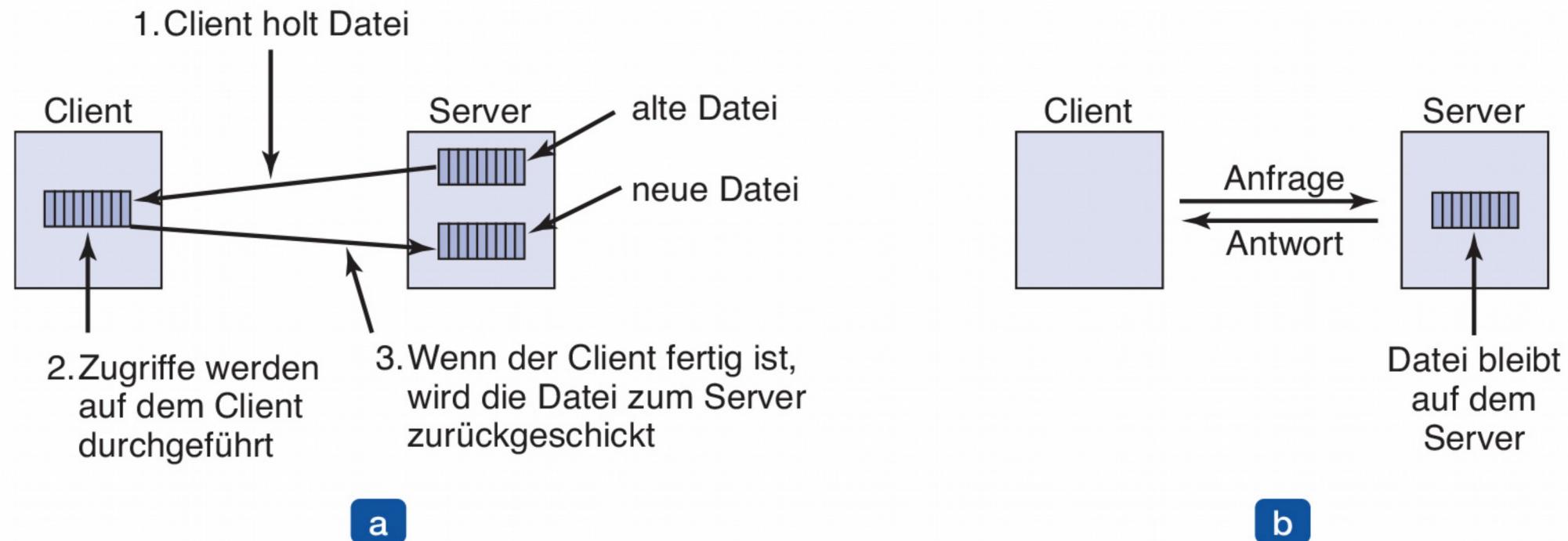


Abbildung 8.33: (a) Das Upload-Download-Modell. (b) Das Remote-Access-Modell.

Die Verzeichnis-Hierarchie

Tanenbaum, A. S.; Bos, H.: Moderne Betriebssysteme. Pearson Studium 2016

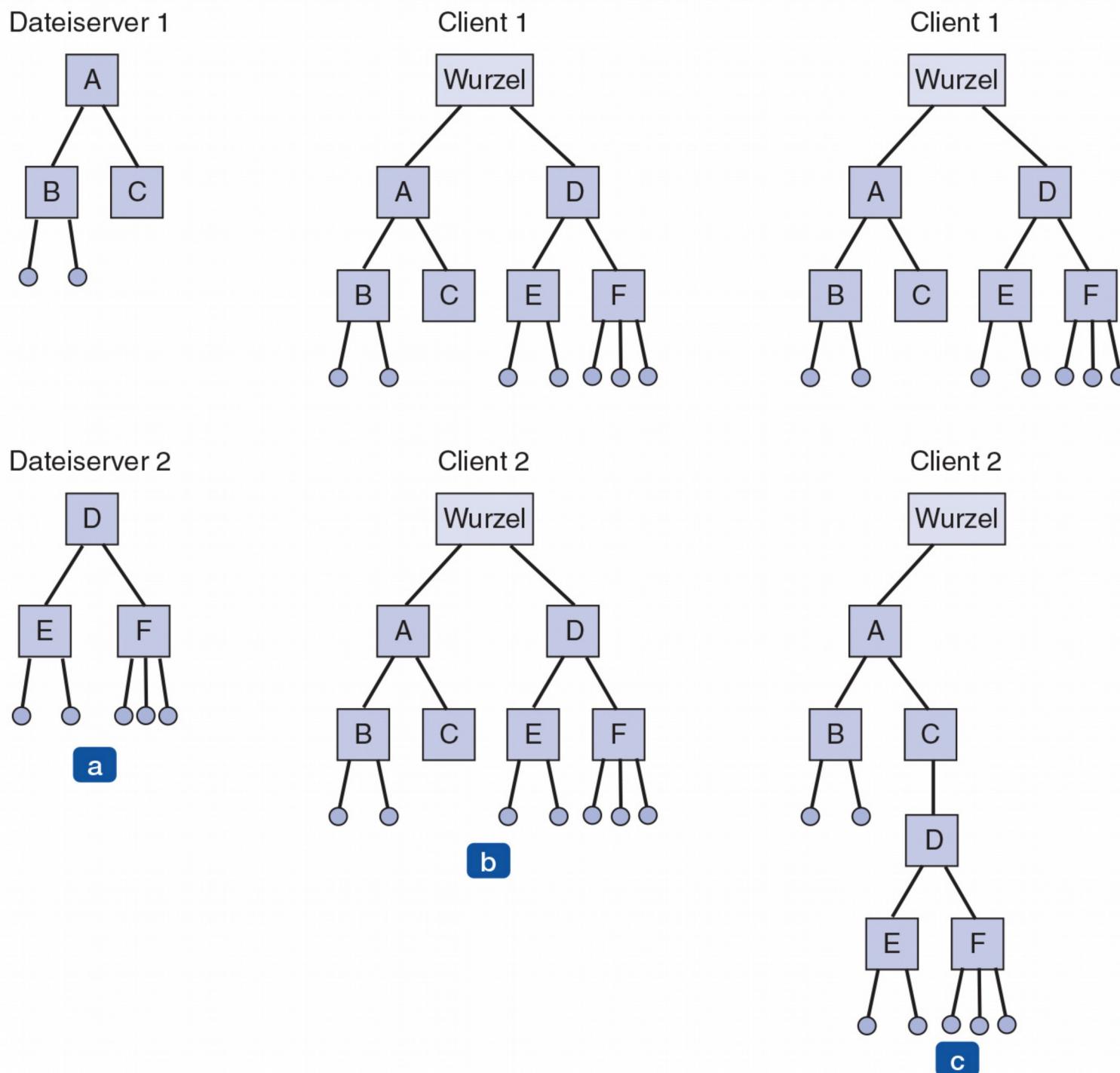


Abbildung 8.34: (a) Zwei Dateiserver. Die Rechtecke stellen die Verzeichnisse dar, die Kreise symbolisieren die Dateien. (b) Ein System, bei dem alle Clients die gleiche Sicht auf das Dateisystem haben. (c) Ein System, in dem verschiedene Clients unterschiedliche Sichten auf das Dateisystem haben.

Transparente Benennung

Häufige Ansätze für die Benennung von Dateien und Verzeichnissen in einem verteilten System:

1. Name der Maschine + Pfad, z. B. /Maschine/Pfad oder Maschine: Pfad.
2. Mounten von Remote-Dateisystemen in die lokale Dateihierarchie.
3. Ein einzelner Namensraum, der auf allen Computern gleich aussieht.

Die Semantik der Dateifreigabe

Tanenbaum, A. S.; Bos, H.: Moderne Betriebssysteme. Pearson Studium 2016

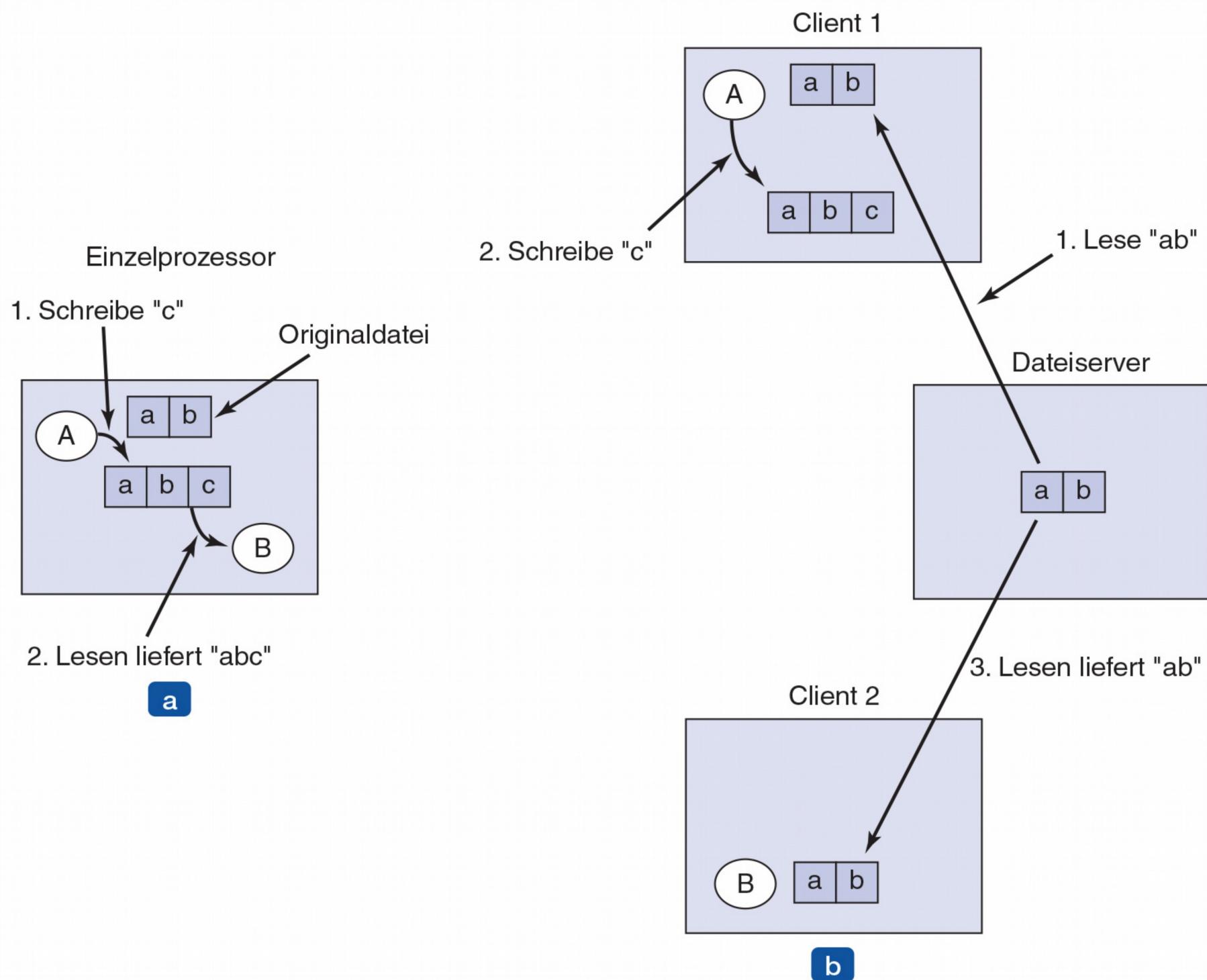


Abbildung 8.35: (a) Sequenzielle Konsistenz. (b) In einem verteilten System mit Caching kann der Lesezugriff veraltete Werte zurückliefern.

Objektbasierte Middleware

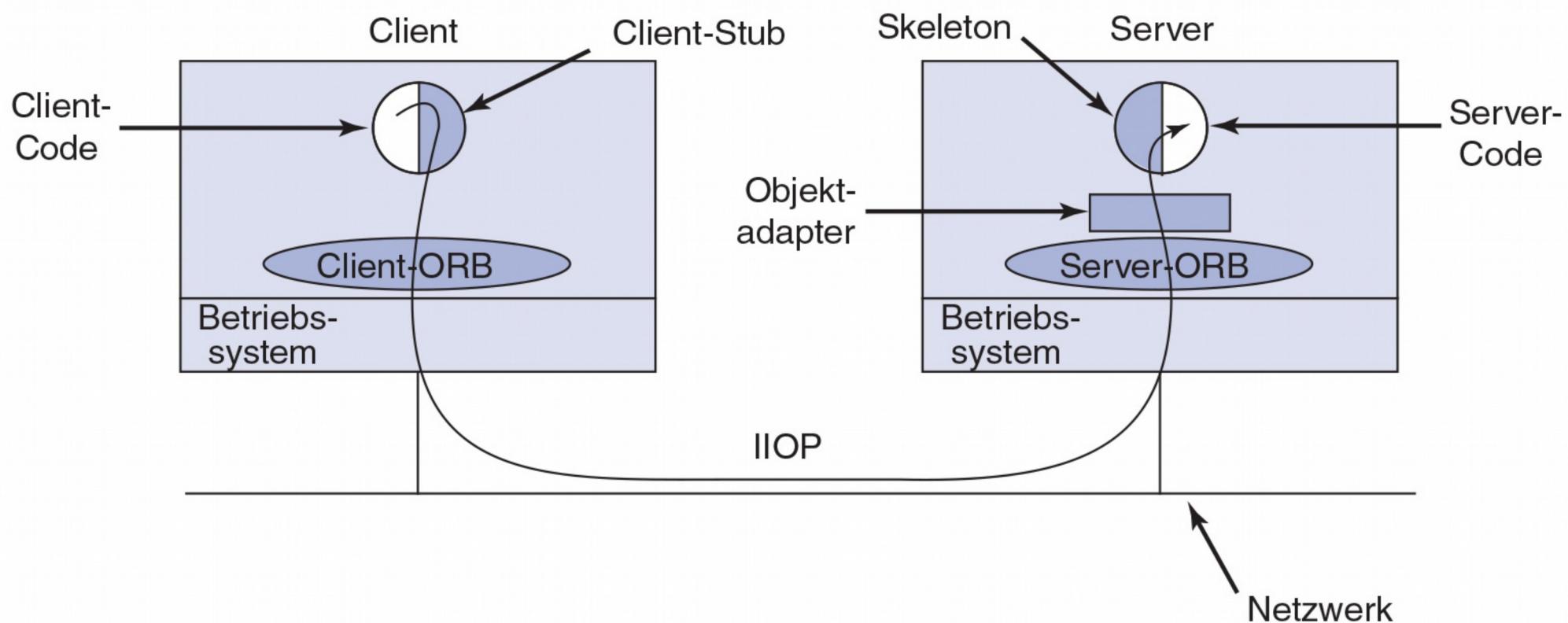


Abbildung 8.36: Hauptelemente eines verteilten Systems, das auf CORBA basiert. Die CORBA-Teile sind dunkler dargestellt.

Koordinationsbasierte Middleware (1)

```
("abc", 2, 5)
("matrix-1", 1, 6, 3.14)
("familie", "ist-schwester", "Stefanie", "Roberta")
```

Abbildung 8.37: Drei Linda-Tupel.

Koordinationsbasierte Middleware (2)

Eine Übereinstimmung tritt auf, wenn die folgenden drei Bedingungen alle erfüllt sind:

1. Template und Tupel haben die gleiche Anzahl von Feldern.
2. Typen von einander entsprechenden Feldern sind gleich.
3. Jede Konstante oder Variable in der Vorlage stimmen mit ihrem Tupelfeld überein.

Die Publish/Subscribe-Architektur

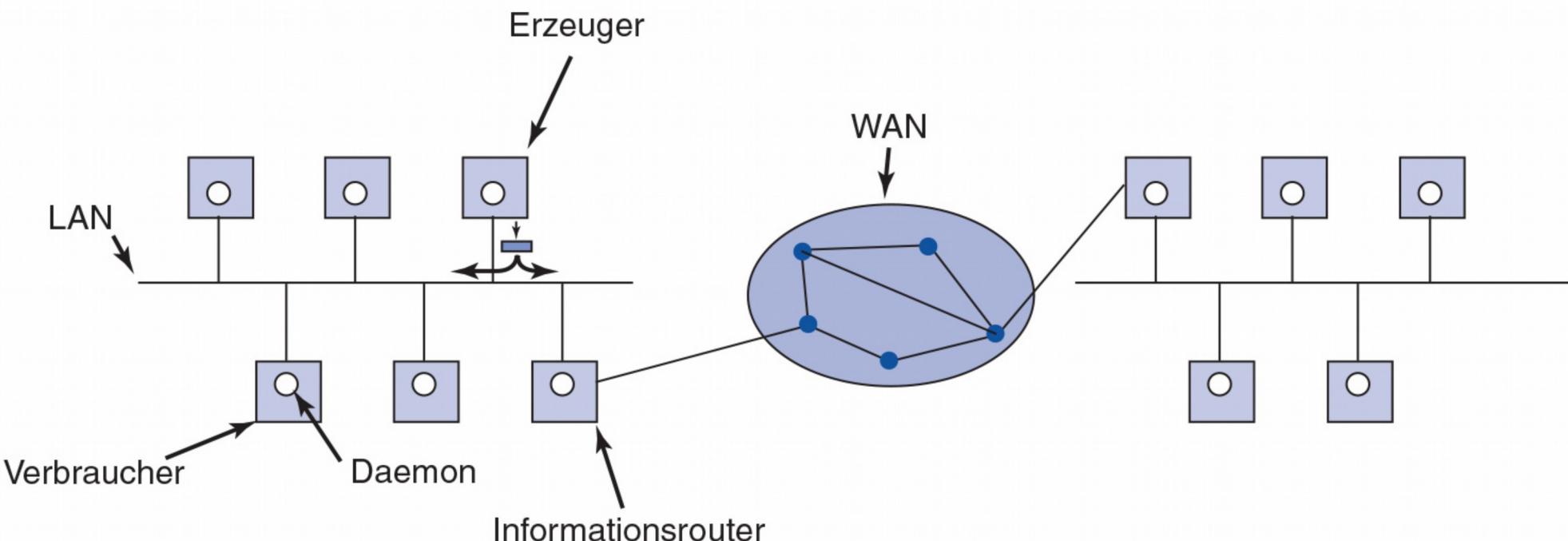


Abbildung 8.38: Die Publish/Subscribe-Architektur.