

Advanced computer networking

Internet Protocols

Thomas Fuhrmann



Network Architectures
Computer Science Department
Technical University Munich

- IP is the ‚waist of the hourglass‘ in the Internet protocol stack:
 - IP connects (geographically or organizationally) local networks to form one global network
 - IP provides the least common denominator for the different organizations
- IP routing is the task to bring data packets to any destination
 - More specifically, routing determines a next hop router that is ‚closer‘ to the destination.
 - Based on the information created by the routing system, packets can be forwarded towards the destination
- Remaining issues for today & next week:
 - How do routing protocols automatically find shortest path routes?
 - What do we need to know about forwarding?
 - What happens in really large networks?



Distance Vector ./. Link State Routing

Routing in the Internet is based on two types of „self-organizing“ routing algorithms:

- **Distance Vector Routing**

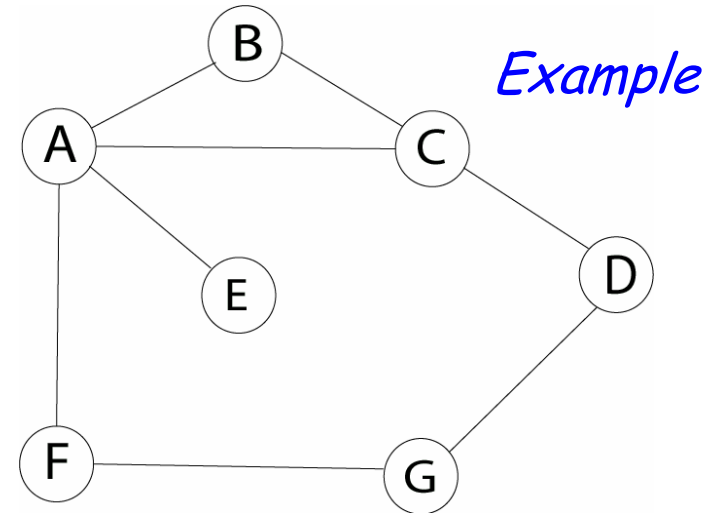
- Each Router knows the distance (e.g. in hops) to all sub-networks.
- Routers exchange these „distance vectors“ (DV) with their neighboring routers.
- Upon reception of such a DV, it is merged with the router's local DV
- Shorter paths are preferred,
- Examples: Routing Information Protocol (RIP), Distance-Vector-Routing-Protocol (DVRP)

- **Link State Routing**

- Routers exchange full topology information, i.e. which sub-networks the routers are attached to.
- Topology information can accommodate various metrics, e.g. distance, load, cost, ...
- Ideally, all routers have an identical view. They use Dijkstra's algorithm to calculate shortest paths.
- Link state routing converges more quickly than distance vector routing.
- Examples: Open Shortest Path First (OSPF), Intra-Domain Intermediate System to Intermediate System Routing Protocol (IS-IS)

Distance Vector Routing

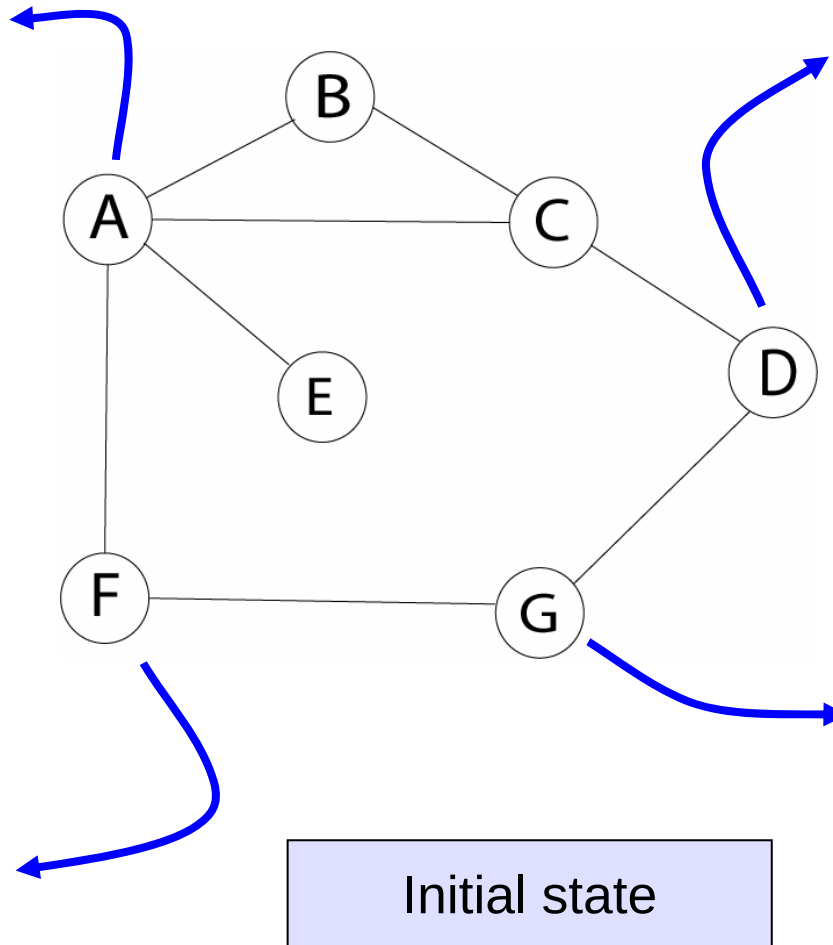
- Simplest distributed & adaptive routing
- „Routing Information Protocol“ (RIP) was the first Internet routing protocol.
- Each router maintains a distance vector (aka routing table) that contains the presumed distance to each sub-network.
- Neighboring routers exchange these tables
- Upon reception the router increases all distances in the received tables by one hop.
- If the result is better than what the router had in its own table, it stores that value in its own table.
- In practice distance vector routing converges only slowly. Especially, failing links are only noticed after several iterations.
- If a sub-network becomes unreachable, the routers need a cut-off, otherwise the algorithm would „count to infinity“.



Example: Distance Vector Routing (1)

Dst.	Hops	Next
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

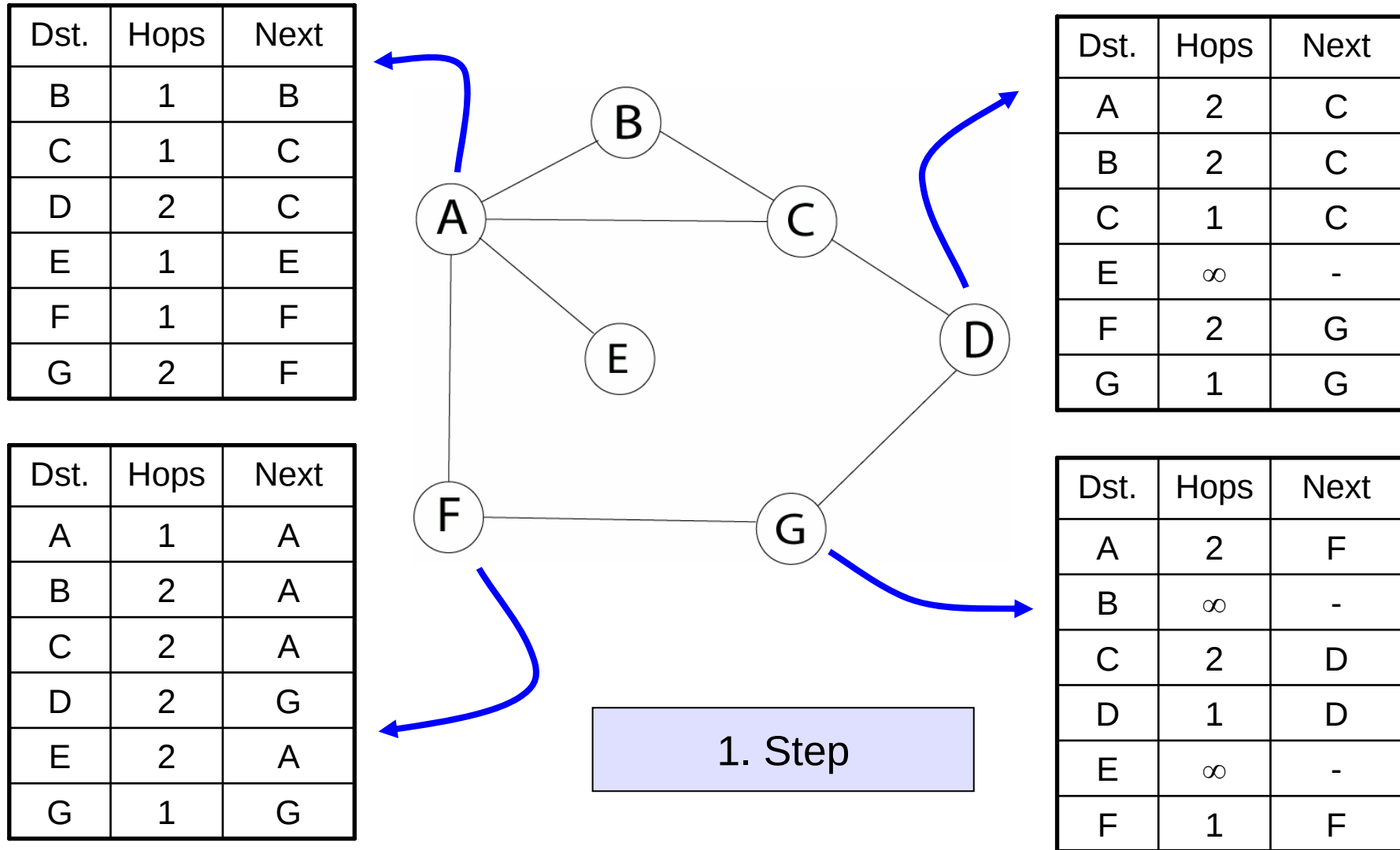
Dst.	Hops	Next
A	1	A
B	∞	-
C	∞	-
D	∞	-
E	∞	-
G	1	G



Dst.	Hops	Next
A	∞	-
B	∞	-
C	1	C
E	∞	-
F	∞	-
G	1	G

Dst.	Hops	Next
A	∞	-
B	∞	-
C	∞	-
D	1	D
E	∞	-
F	1	F

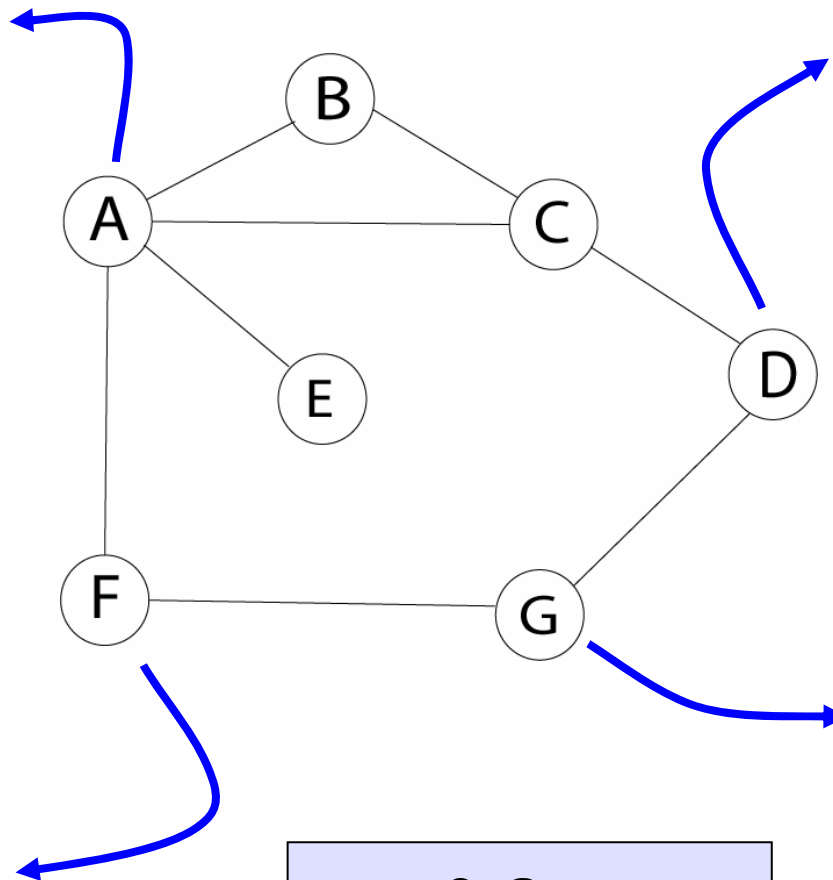
Example: Distance Vector Routing (2)



Example: Distance Vector Routing (3)

Dst.	Hops	Next
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Dst.	Hops	Next
A	1	A
B	2	A
C	2	A
D	2	G
E	2	A
G	1	G

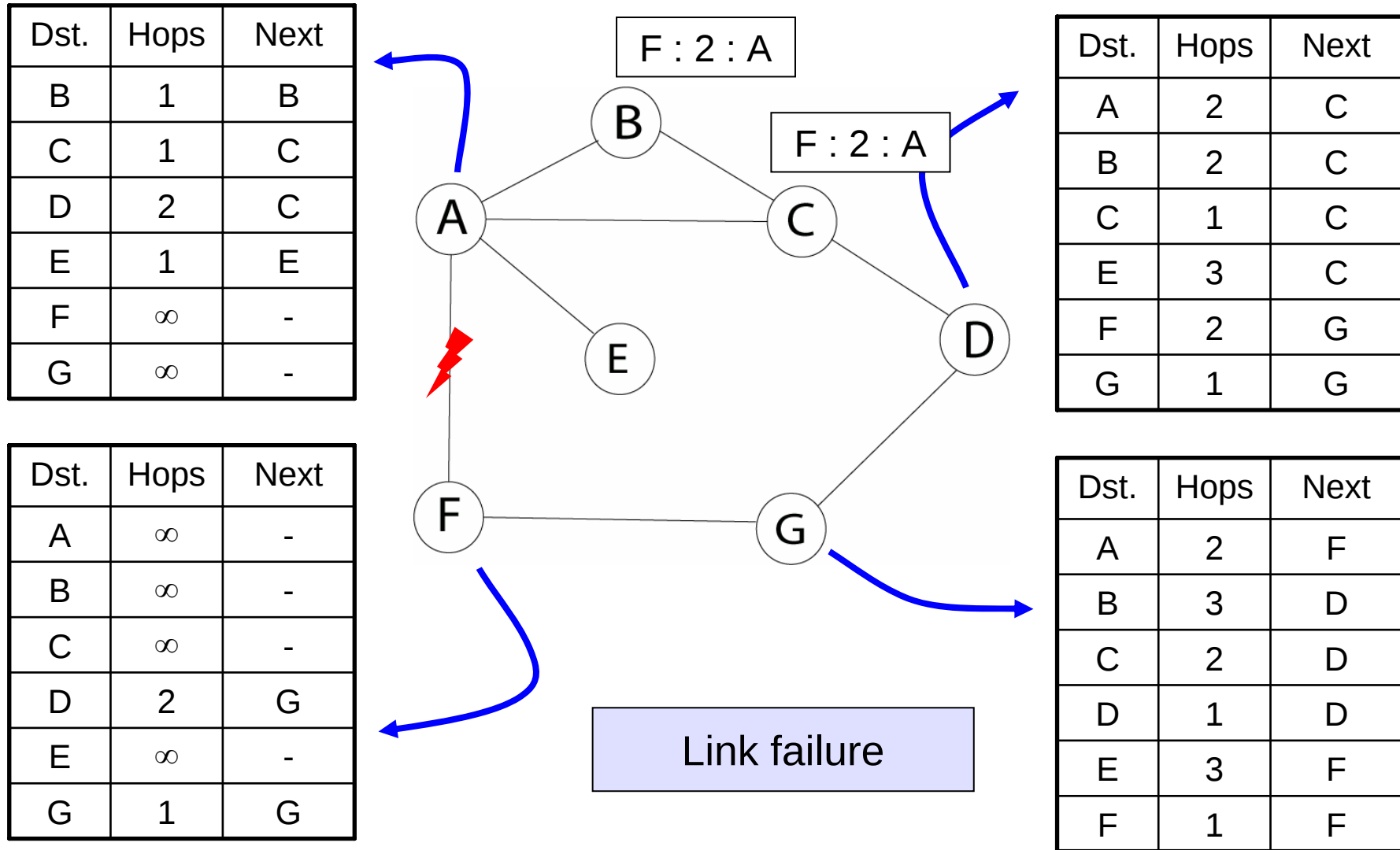


2. Step

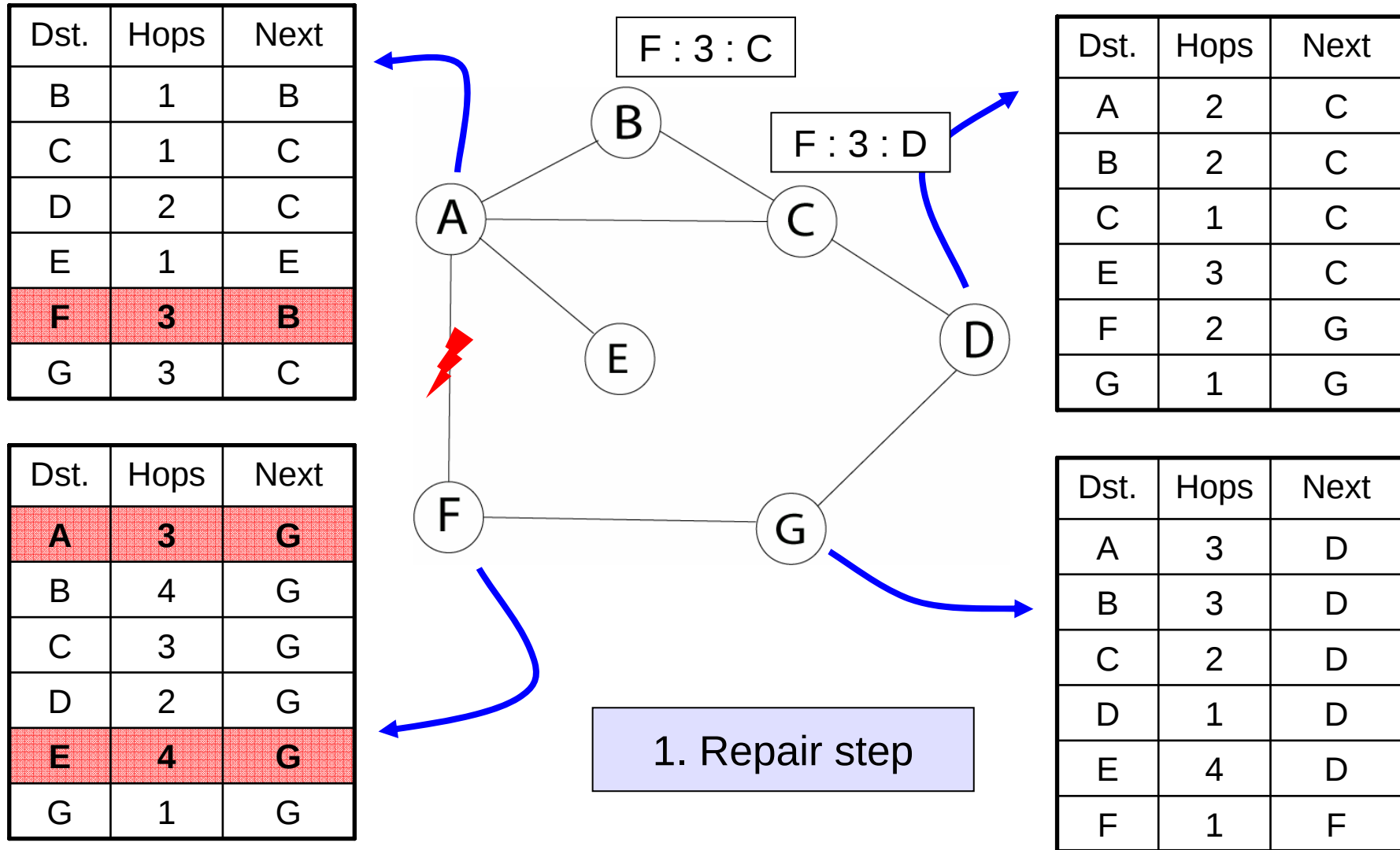
Dst.	Hops	Next
A	2	C
B	2	C
C	1	C
E	3	C
F	2	G
G	1	G

Dst.	Hops	Next
A	2	F
B	3	D
C	2	D
D	1	D
E	3	F
F	1	F

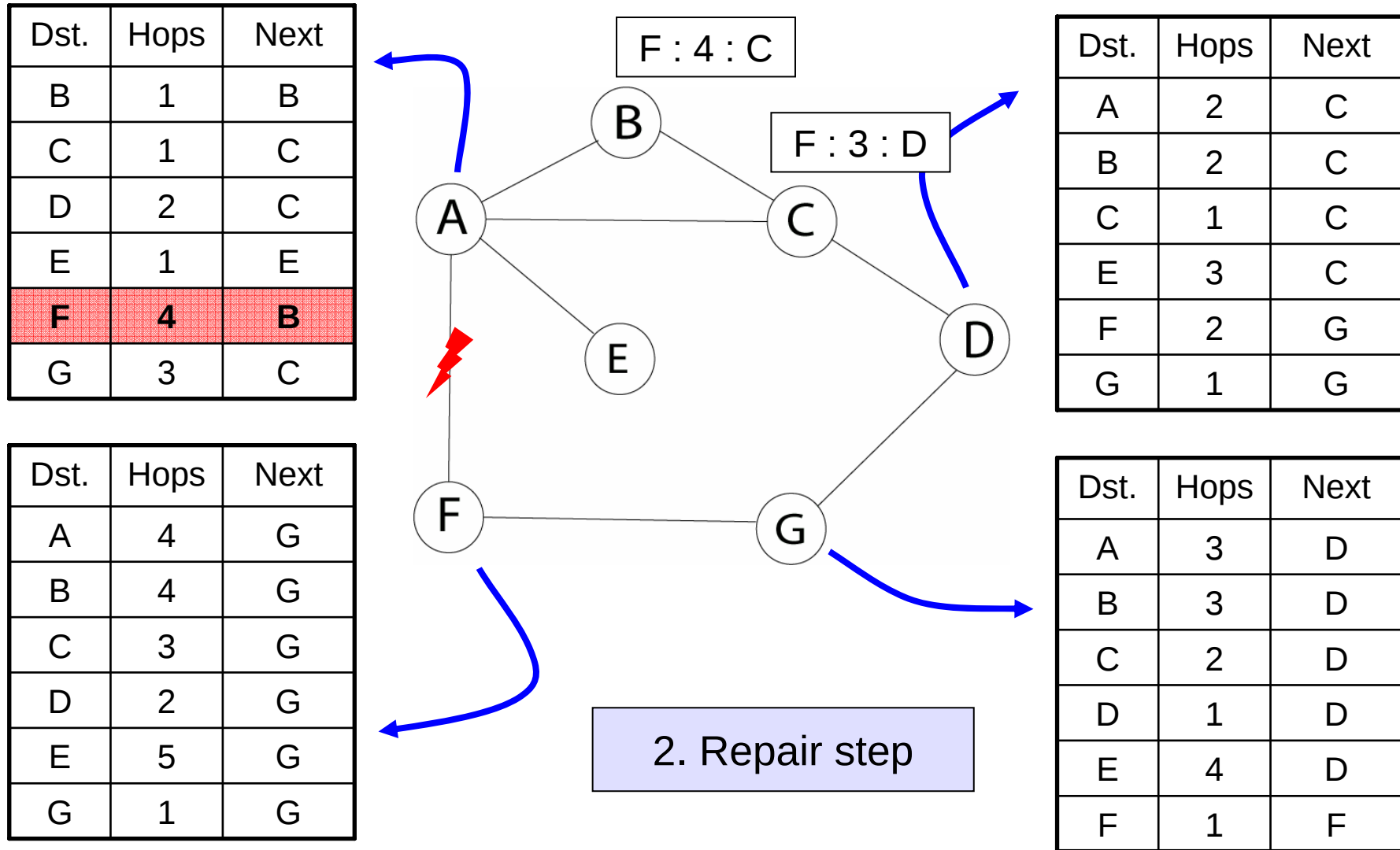
Example: Distance Vector Routing (4)



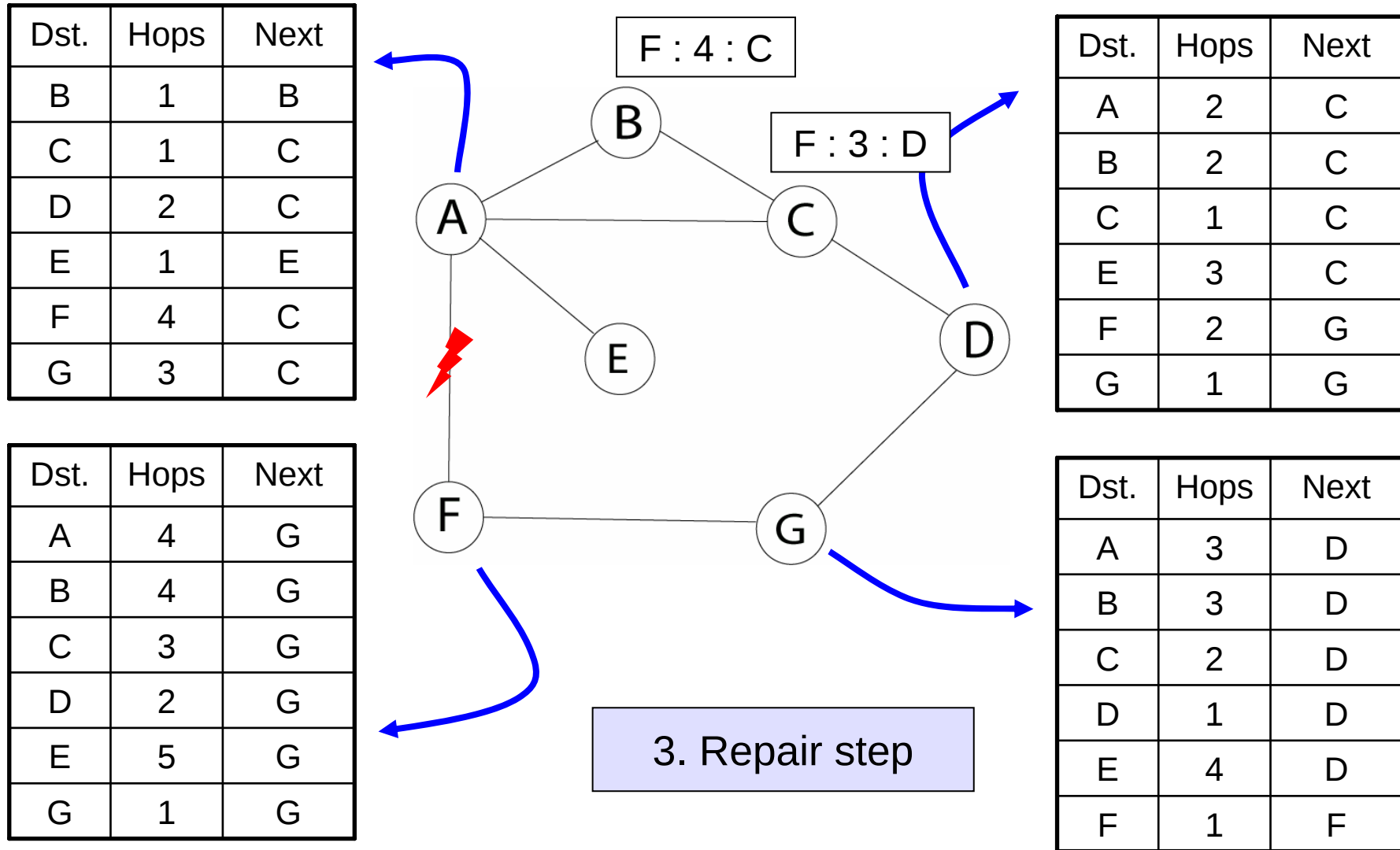
Example: Distance Vector Routing (5)



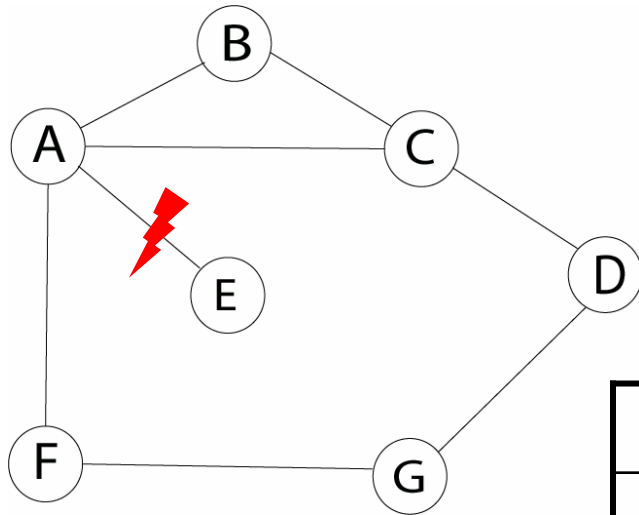
Example: Distance Vector Routing (6)



Example: Distance Vector Routing (7)



Count-to-Infinity Problem



Example

Max. network diameter
is assumed to be
“Infinity”

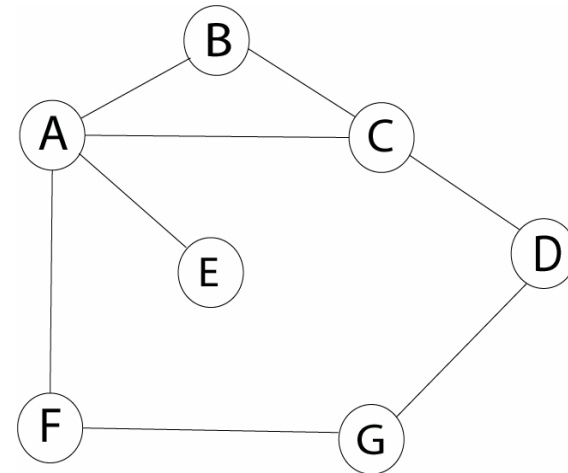
Typical value: 16

- Distance vector routing cannot (always) discover when a sub-network becomes unreachable.
- The only indication is that in such a case the distance increases step-by-step.

	A	B	C	D	F	G
	1 : E	2 : A	2 : A	3 : C	2 : A	3 : F
Link fails	∞	2 : A	2 : A	3 : C	2 : A	3 : F
1st Step	3 : B	3 : C	3 : B	3 : C	4 : G	3 : F
2nd Step	4 : B	4 : A	4 : A	4 : C	4 : A	5 : F
3rd Step	5 : B	5 : A	5 : A	5 : C	5 : A	5 : F
4th Step	6 : B	6 : A	6 : A	6 : C	6 : A	6 : F
5th Step	7 : B	7 : A	7 : A	7 : C	7 : A	7 : F

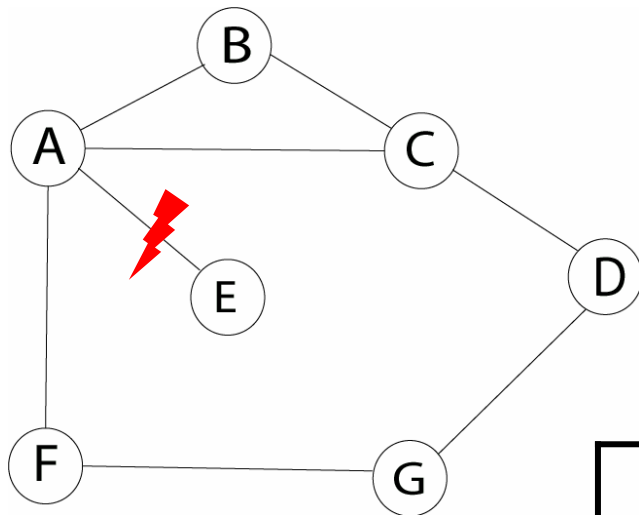
Split Horizon & Poison Reverse

- Distanz Vektor Protokolle konvergieren langsam, da sich Informationen nur Schritt für Schritt verbreiten
- Aber: Werden die Tabellen „einfach so“ übernommen, verbreiten sich sogar sinnlose Informationen (vgl. Beispiel)
- Lösung: Gebe nur Informationen weiter, die für den Nachbarn sinnvoll sein könnten:
 - **Split Horizon:** Nachbar X erhält nur Tabelleneinträge die nicht über X selbst als Next Hop führen.
 - **Poison Reverse:** Für Nachbar X werden alle Einträge, die über X selbst führen auf „Unendlich“ gesetzt.



Ziel	Hops	Next	Weiter-leiten an...
B	1	B	C, E, F
C	1	C	B, E, F
D	2	C	B, E, F
E	1	E	B, C, F
F	1	F	B, C, E
G	2	F	B, C, E

Beispiel: Poison Reverse



Beispielnetz

Auch mit Poison Reverse konvergieren Distanzvektorprotokolle nur langsam!

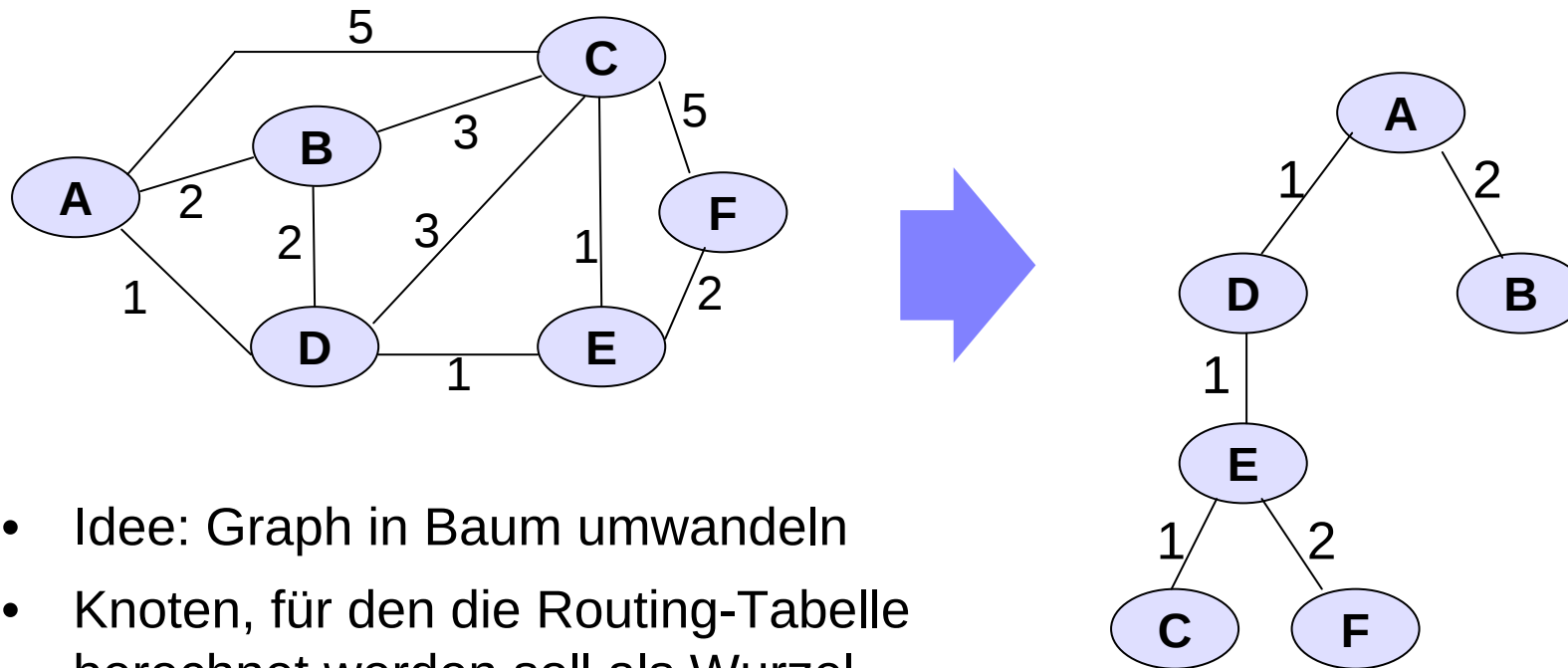
- Mit Poison Reverse verbreitet sich die Unerreichbarkeit von Knoten vergleichsweise rasch.
- Veränderungen im Abstand n Hops werden i.d.R. nach n Schritten erkannt.

	A	B	C	D	F	G
	1 : E	2 : A	2 : A	3 : C	2 : A	3 : F
Ausfall	∞	2 : A	2 : A	3 : C	2 : A	3 : F
1. Schritt	∞	3 : C	3 : B	3 : C	∞	3 : F
2. Schritt	∞	∞	∞	4 : C	∞	4 : D
3. Schritt	∞	∞	∞	∞	∞	5 : D
4. Schritt	∞	∞	∞	∞	∞	∞

Link State Routing

- *Verteiltes, adaptives Routing*
- Grundidee: Jeder Router kennt den Zustand des gesamten Netzes und berechnet sich daraus die Routing-Tabelle
- Beispiele: OSPF (Open Shortest Path First) und IS-IS (Intermediate System - Intermediate System)
- Algorithmus:
 - Entdecken neuer Nachbarn (mit HELLO-Paket)
 - Messung der Verzögerung bzw. Kosten zu jedem Nachbarn (ECHO-Paket misst Umlaufzeit)
 - Erstellen eines „Link-State“-Pakets mit allen gelernten Daten (Sender, Liste der Nachbarn mit Verzögerung, Zeitstempel)
 - Periodisches oder ereignisgesteuertes Fluten dieses Paketes im Netz (z.B. bei neuem Nachbar, Leitungsausfall, etc.)
 - Auf jedem Router Berechnung des kürzesten Pfades zu allen anderen Routern (mit z.B. Dijkstra-Algorithmus)

Dijkstra-Algorithmus



- Idee: Graph in Baum umwandeln
- Knoten, für den die Routing-Tabelle berechnet werden soll als Wurzel eines Baums
- Knoten mit der kleinsten Entfernung hinzufügen
- Schrittweise weiter, bis alle Knoten im Baum sind
- Entfernung zu einem Knoten = Summe entlang des Pfades im Baum

- Oszillation der Verkehrsströme
 - Hängen die Kosten eines Links von der Verkehrslast ab, können Oszillationen entstehen
 - Lösung wäre Lastverteilung auf verschiedene Routen mit ähnlichen Kosten. (Wird kaum in der Praxis angewandt.)
- Oszillation durch Austausch der Routing-Informationen
 - Werden die Link-State-Informationen oder die Routing-Tabellen (bei Distanzvektor-Verfahren) in konstanten Zeitintervallen verteilt, synchronisiert sich das ganze Netz.
 - Leitungen periodisch durch Informationsaustausch belastet.
 - Router periodisch durch Routenberechnung überlastet.

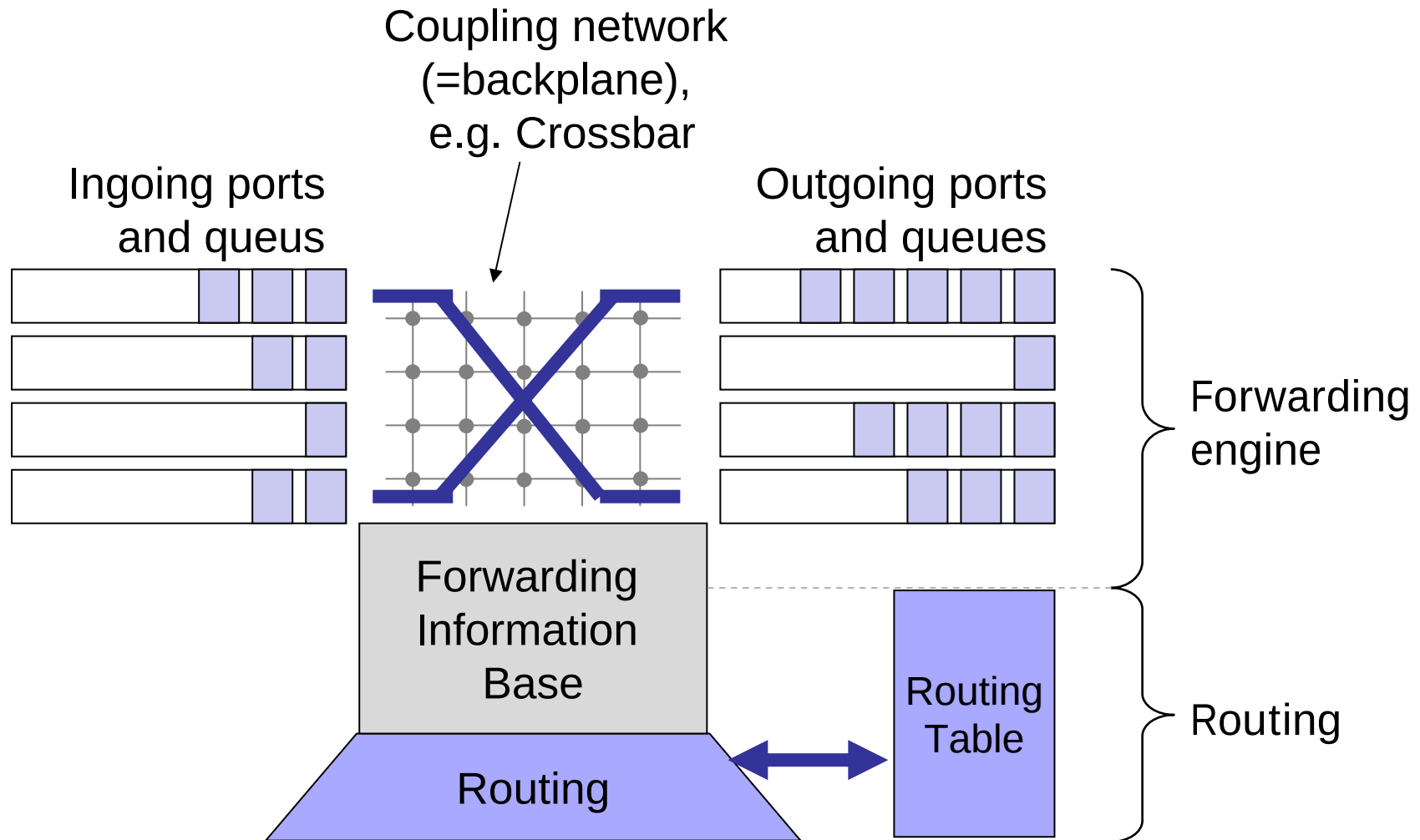
Link-State vs. Distanz-Vektor

- Komplexität der Kontroll-Dateneinheiten
 - Bei Link-State muss jeder Knoten die Kosten aller Links kennen.
 - Bei Distanz-Vektor werden nur zusammengefasste Informationen an benachbarte Knoten weitergegeben.
- Konvergenzgeschwindigkeit
 - Link-State: Schnelle Konvergenz zu konsistentem Zustand.
 - Distanz-Vektor: Langsame Konvergenz. Count-to-Infinity-Problem!
- Robustheit
 - Link-State berechnet Routen verteilt, Fehler bleibt ggf. begrenzt bzw. leichter zu erkennen
 - Bei Distanz-Vektor-Algorithmen kann ein Knoten inkorrekte Pfade zu allen Zielen verbreiten
- Gewinner?
 - Link-State konvergiert schneller und ist robuster
 - Distanz-Vektor einfacher zu implementieren

Forwarding packets in the Internet

A thick, blue, hand-drawn brushstroke underline is positioned below the title, extending across most of the width of the slide.

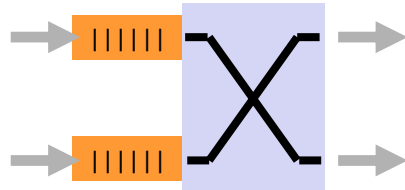
Router Overview



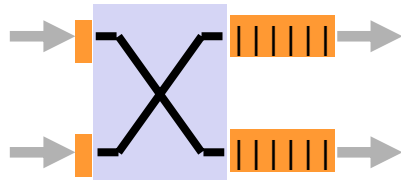
Routing versus Forwarding

- Routing produces forwarding information,
 - E.g. longest prefix match lookup yields an outgoing interface.
 - Note: Routing determines the next hop router, not only the interface, but with point-to-point links (which are typical for the Internet core) this is a one-to-one mapping.
 - Thus, here we assume that we only determine the outgoing interface.
- Forwarding needs to
 - Bring packets from the ingoing interface to the respective outgoing interface (→coupling network)
 - Buffer the packets until outgoing interface is available (→blocking, queuing)

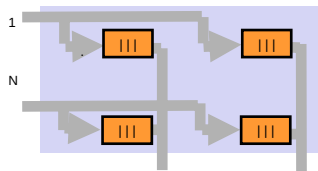
Where to buffer the packets?



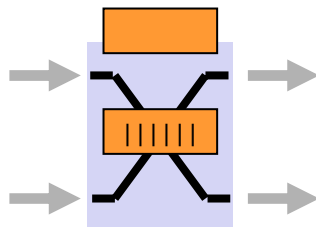
- + Simple (→buffer on the line card)
- Head-of-Line Blocking
- Throughput 60% - 75%



- + Throughput 80%- 85%
- Requires fast memory access
- Requires additional input buffer

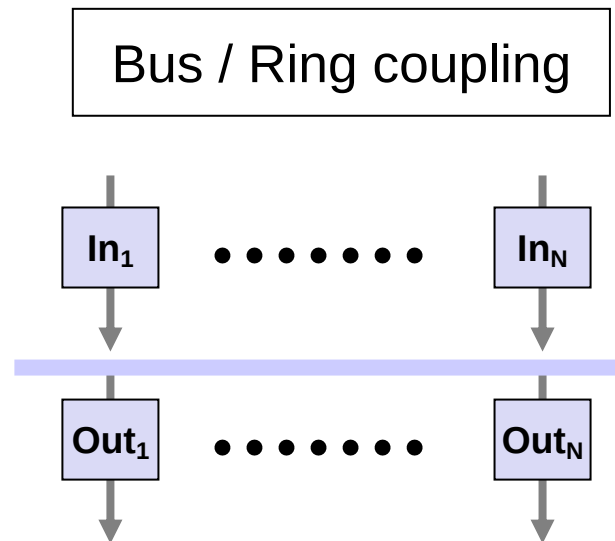


- + Optimal throughput
- High memory demand

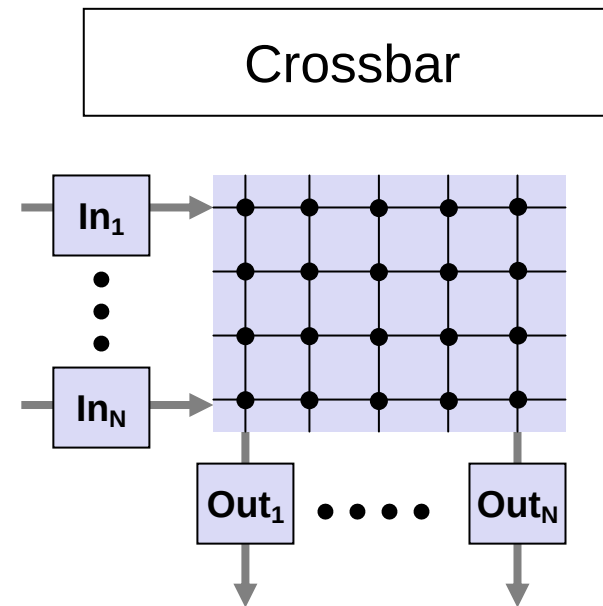


- + Low memory demand
- Requires fast memory access

How to move the packets? (1)



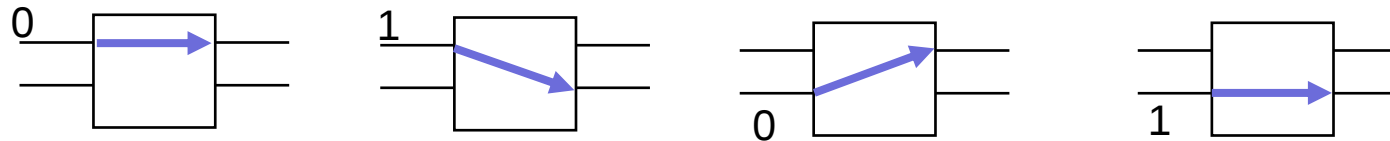
- + Easily extensible
- + Simple support for broadcast & multicast
- Overall capacity limited by backplane capacity



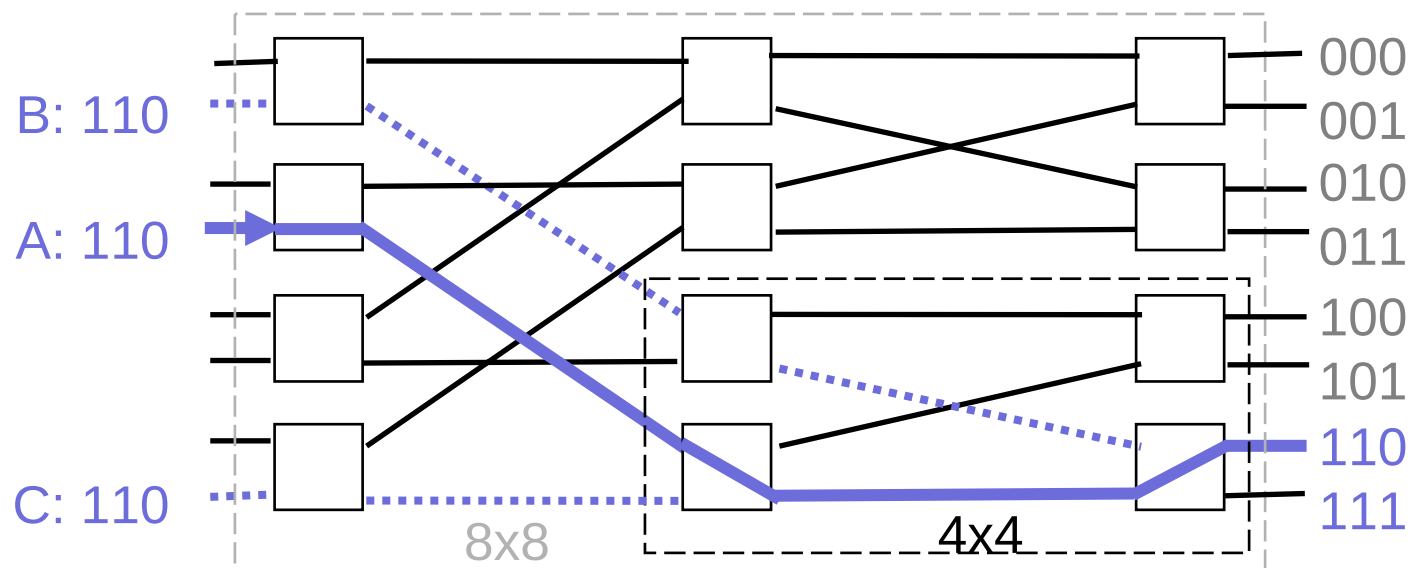
- + Packets switched in parallel
- + High throughput
- Expensive technology
- Not scalable

How to move the packets? (2)

Elementary Switching Elements:



3-stage Banyan Network:

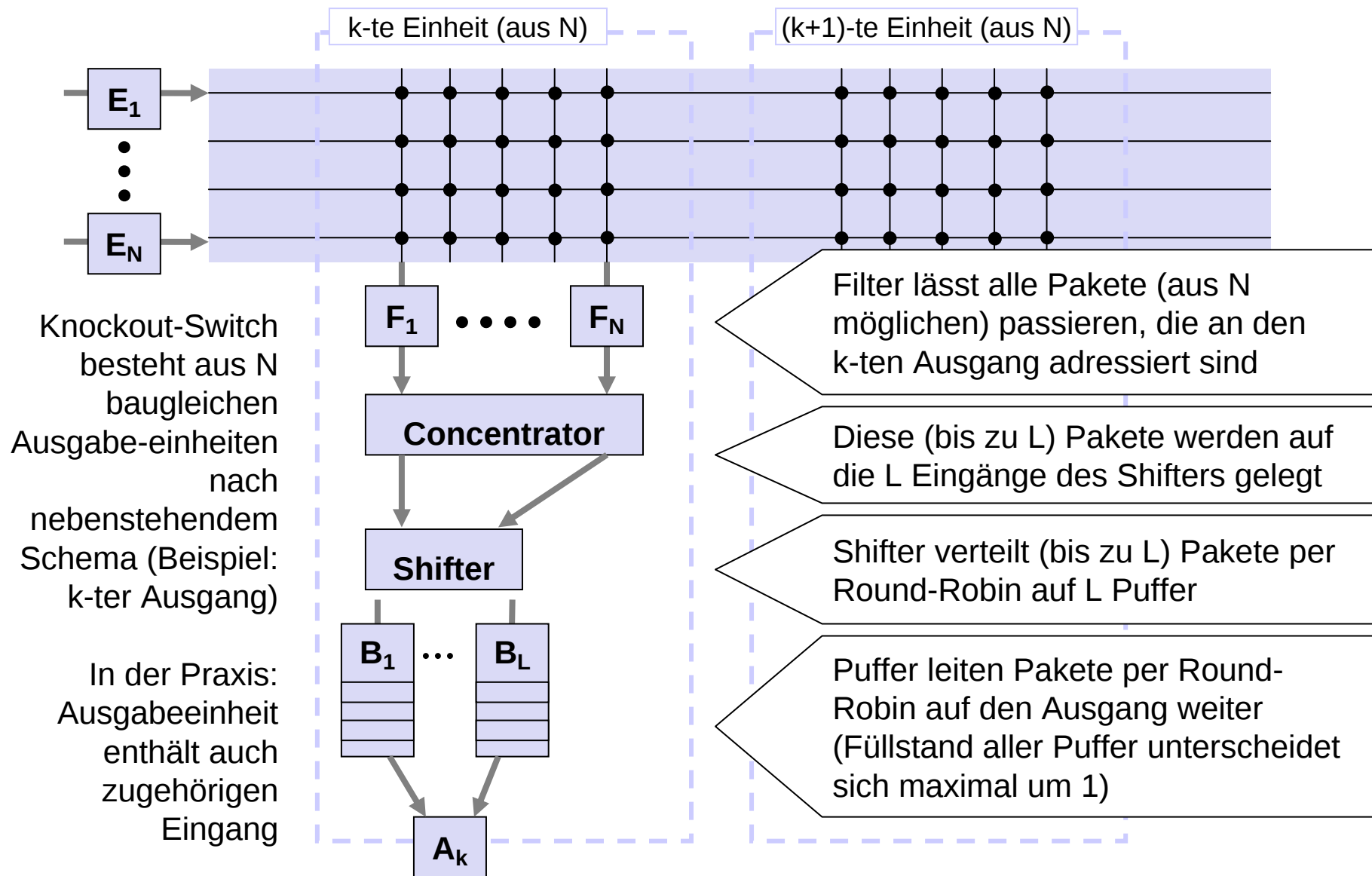


Path of packet A

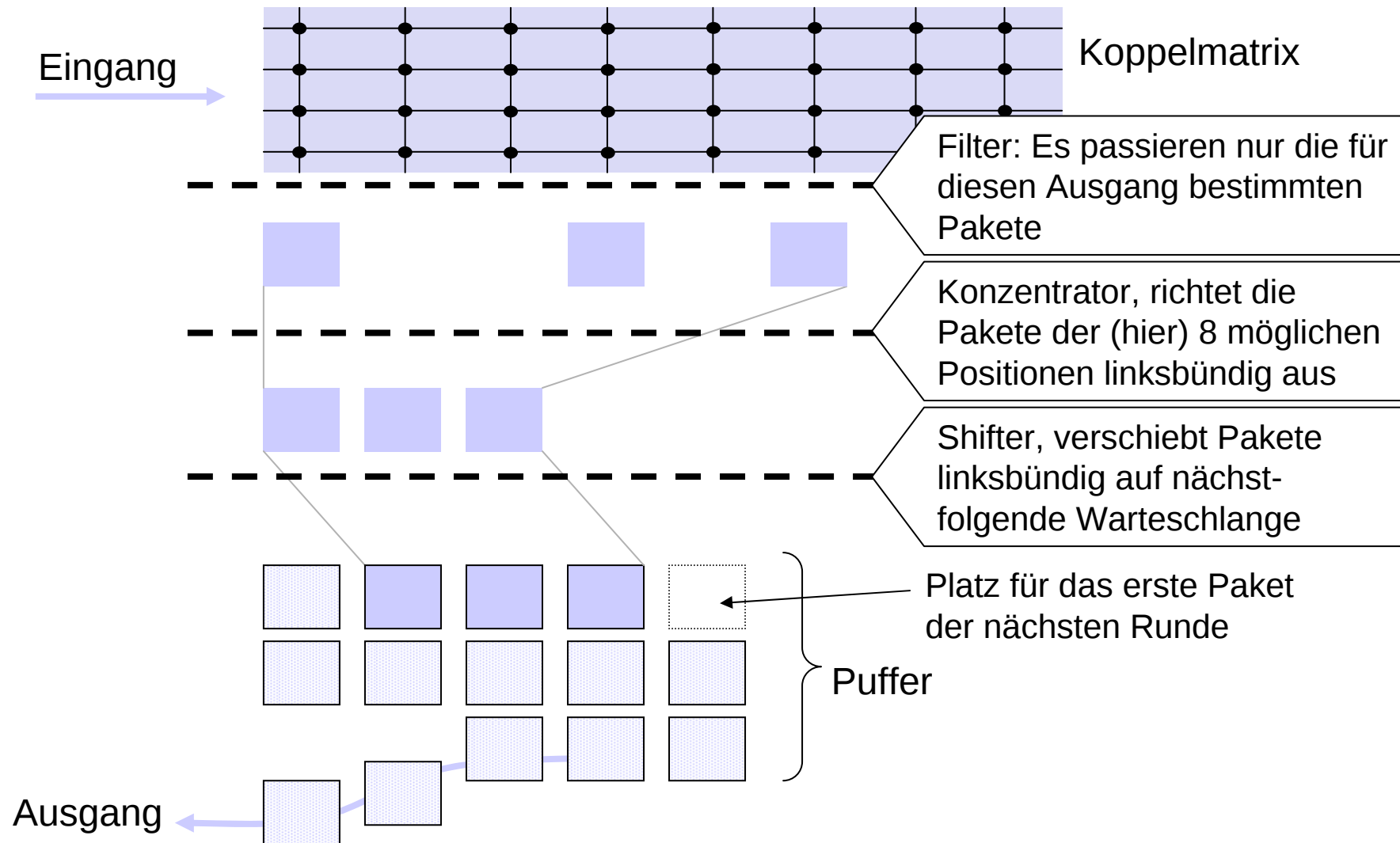
Paths of packets B and C

Other links in the switch

Knockout-Switch (1) - Architektur



Knockout-Switch (2) - Ablauf



Routers in Practice (1)



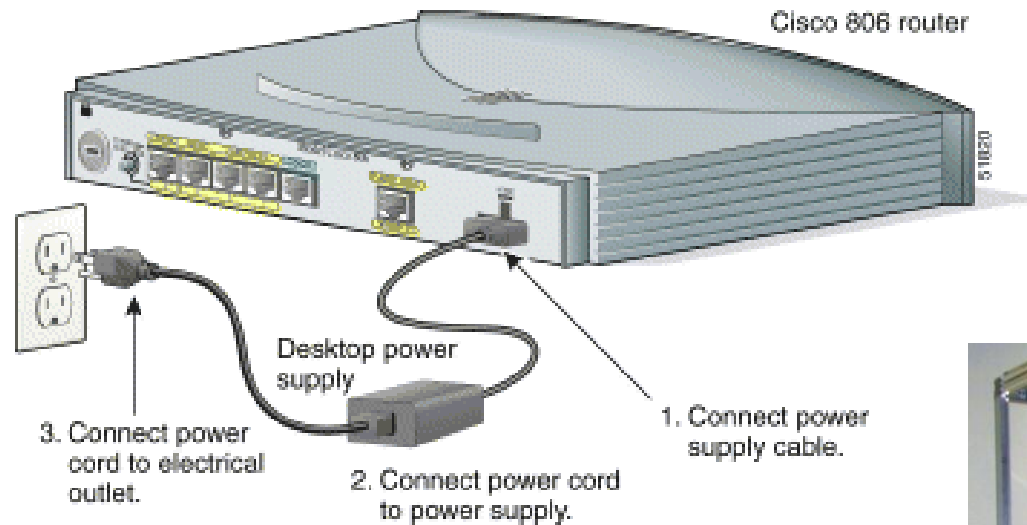
2.4GBit/s Backbone (Cisco 12000)



Campus (Cisco 7500)

* Uni Karlsruhe Networks incl. Baden-Württemberg research backbone, 2003

Routers in Practice (2)

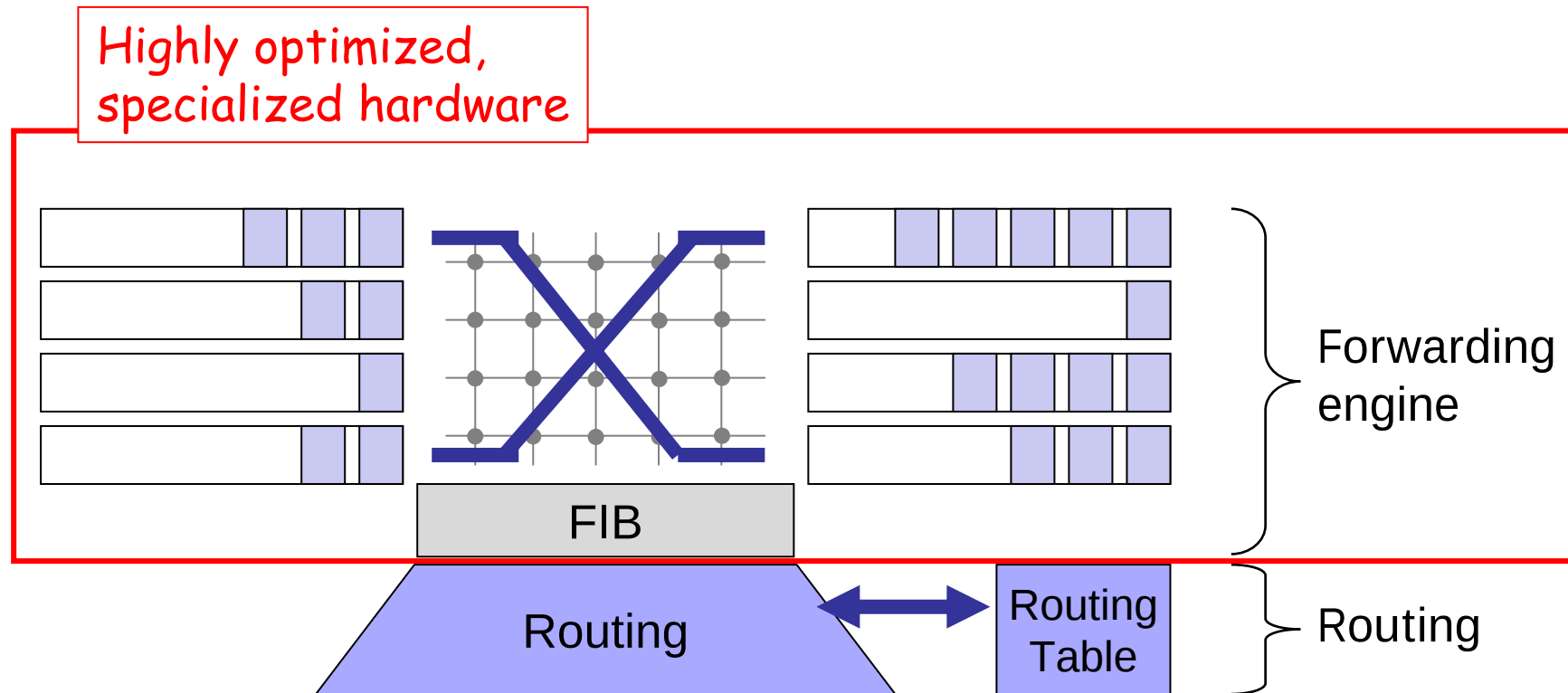


Small office & home office
(Cisco 806)

Local network at a CS chair
(cheap PCs)



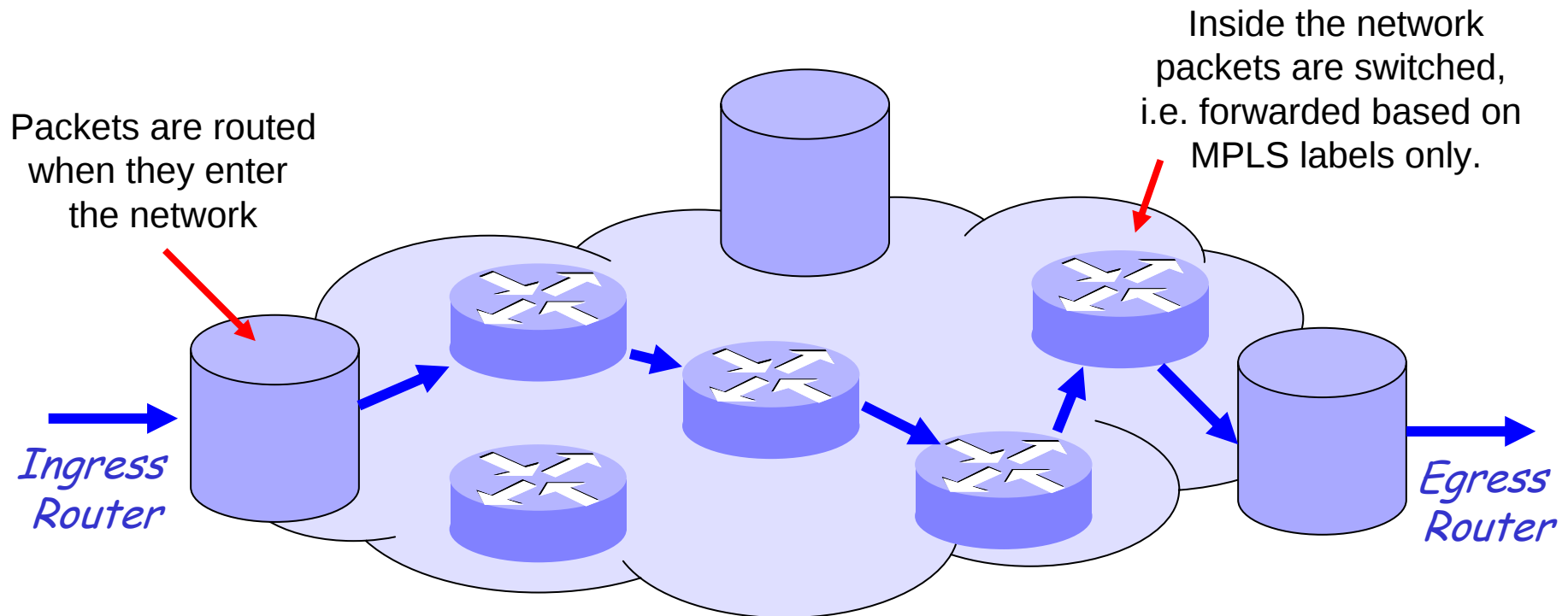
Router Architecture Overview



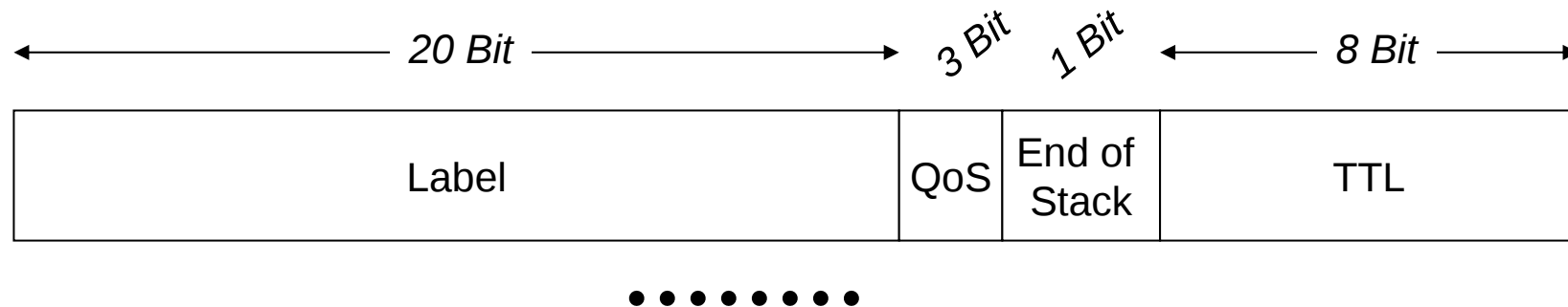
General purpose computing hardware
with operation system, running routing protocols.

Multiprotocol Label Switching

- Traditionally, Internet routers combine both the routing and the forwarding functionality in one device.
- Thus, routers are ,omnipotent‘ but expensive.
- Today (RFC 3031, Jan 2001), multi-protocol label switching (MPLS) is a mechanism to separate routing and forwarding.



MPLS Header

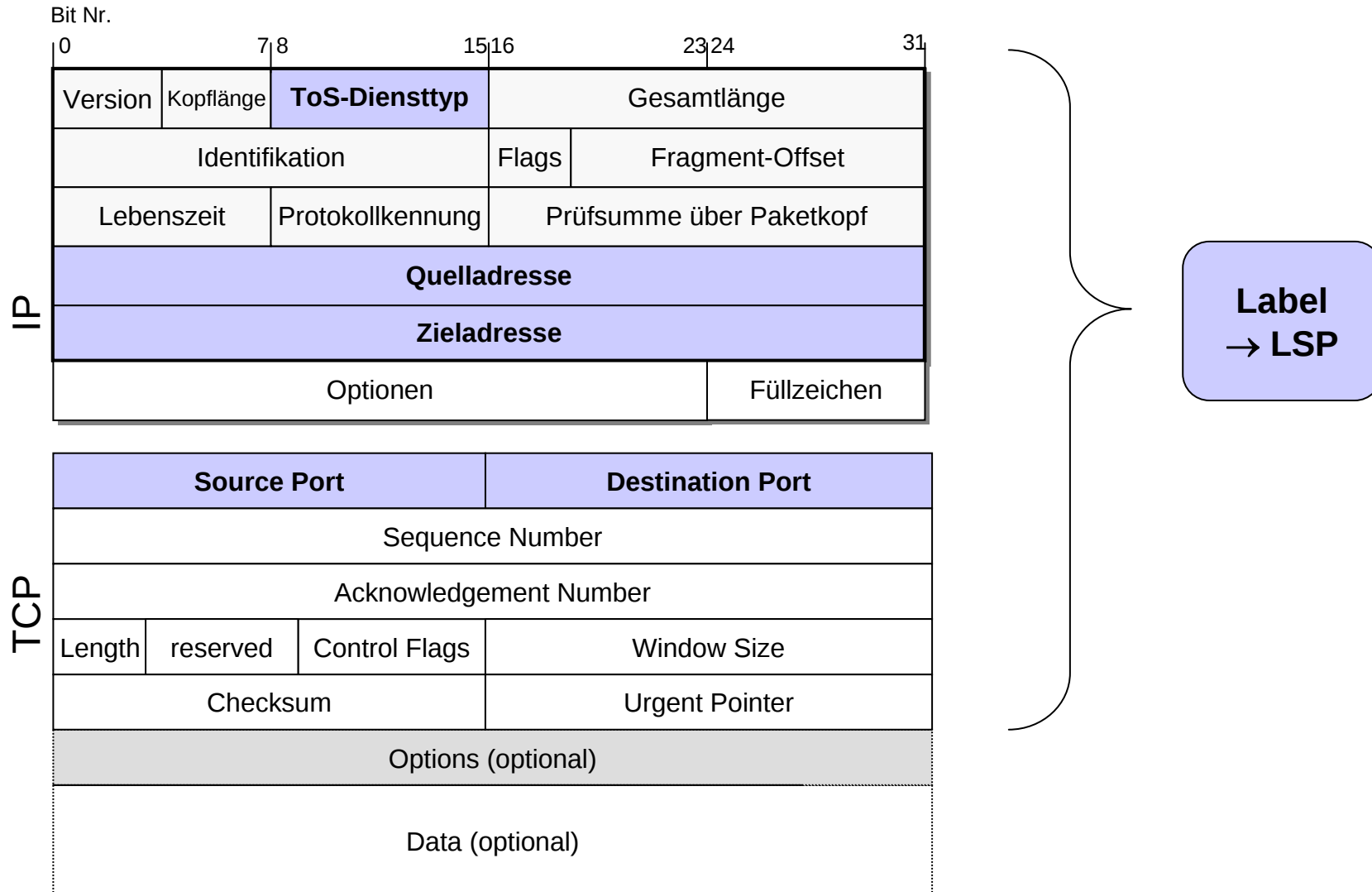


- The MPLS header is a stack of 32 bit entries between the data link header and the network header (→ layer 2.5)
- The MPLS layer is considered so small and lightweight that it is called ‚shim‘ (=dt. Unterlegscheibe) layer.
- Ingress router copies TTL from IP header; egress router copies TTL into IP header.

MPLS Terminology

- **Forwarding Equivalence Class (FEC)**
 - Gebildet anhand Eigenschaften von Datenpaketen (z.B. Ziel-IP-Adresse)
 - Alle Pakete einer FEC erhalten identisches Label und somit identische Behandlung
- **Label Switch Router (LSR)**
 - Im Kern eines MPLS-Netzes
 - Betrachten nur den MPLS Shim Header
- **Label Edge Router (LER)**
 - Router am Ein-/Ausgang eines MPLS-Netzes
 - Klassifiziert Pakete in eine FEC und fügt MPLS Shim Header mit entsprechendem Label ein
- **Label Distribution Protocol (LDP)**
 - Austausch von Labels / Label-Bindings
 - Router werden als Label Distribution Peers bezeichnet
 - Mehrere LDPs existieren, z.B. für BGP, RSVP, MPLS-LDP
- **Label Information Base (LIB)**

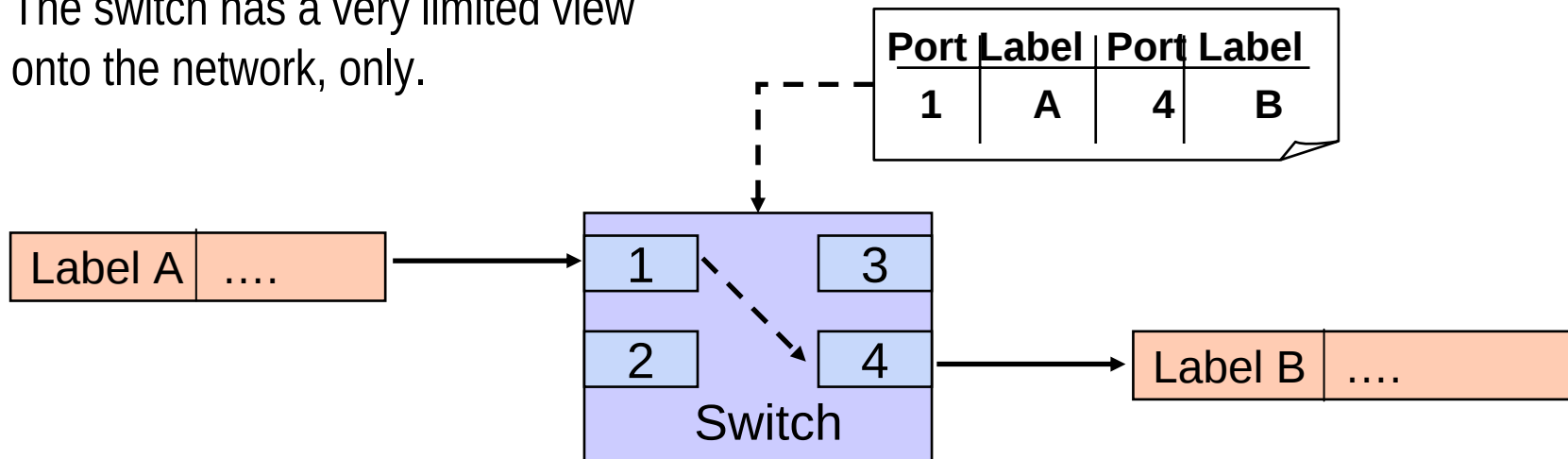
FEC Mapping



- The FEC could be derived from the destination address alone.
- Using other header fields allows traffic engineering:
 - Different sources can be treated differently.
 - E.g. only traffic from the ISPs own customers are forwarded over the new high speed path, traffic from other networks is forwarded over the old switching hardware, etc.
- Stacking the MPLS header allows tunneling of several FEC over the same path in one part of the network.
- Within the tunnel, the tunneled traffic is isolated from the rest of the traffic.
- This can be used to easily build up virtual private networks.

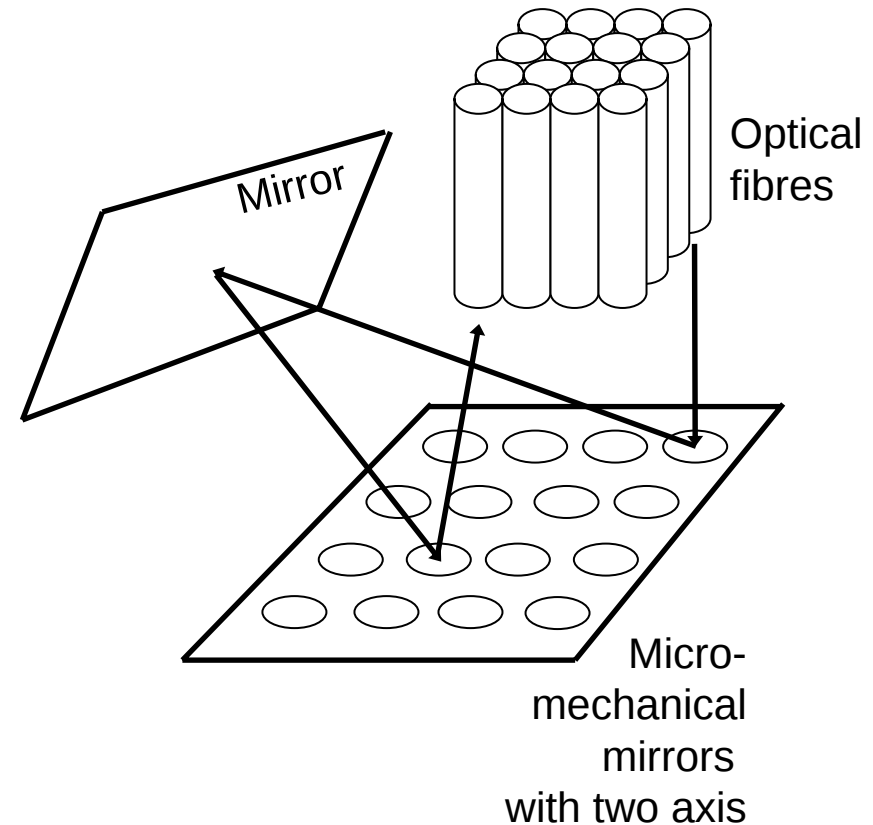
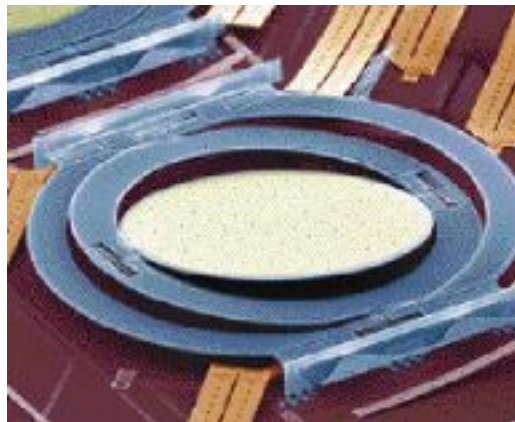
Label Swapping

- Edge routers send label swapping tables to all switches in the network
- Switches look up the incoming port/label combination to obtain an outgoing port/label combination.
 - The outgoing label is copied into the shim header, and
 - The packet is sent to the given outgoing port.
- These operations are cheap and fast.
- The switch has a very limited view onto the network, only.



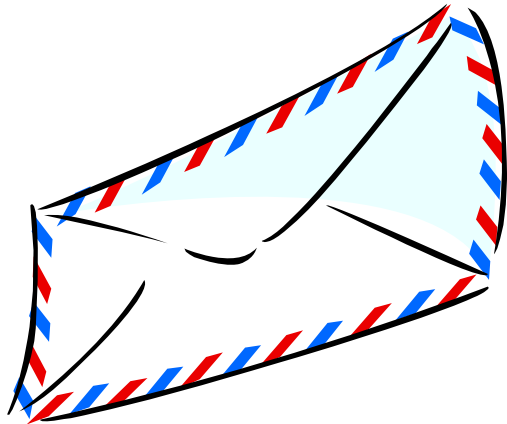
Lambda Switching

Instead of swapping labels in the LSRs the LERs could set up the path so that the respective label port pair is unique at each switch. Think of the labels as different fibers or wavelengths in the fibers.



Scalability in today's Internet

A thick, blue, hand-drawn brushstroke underline is positioned below the title, extending across most of the slide's width.



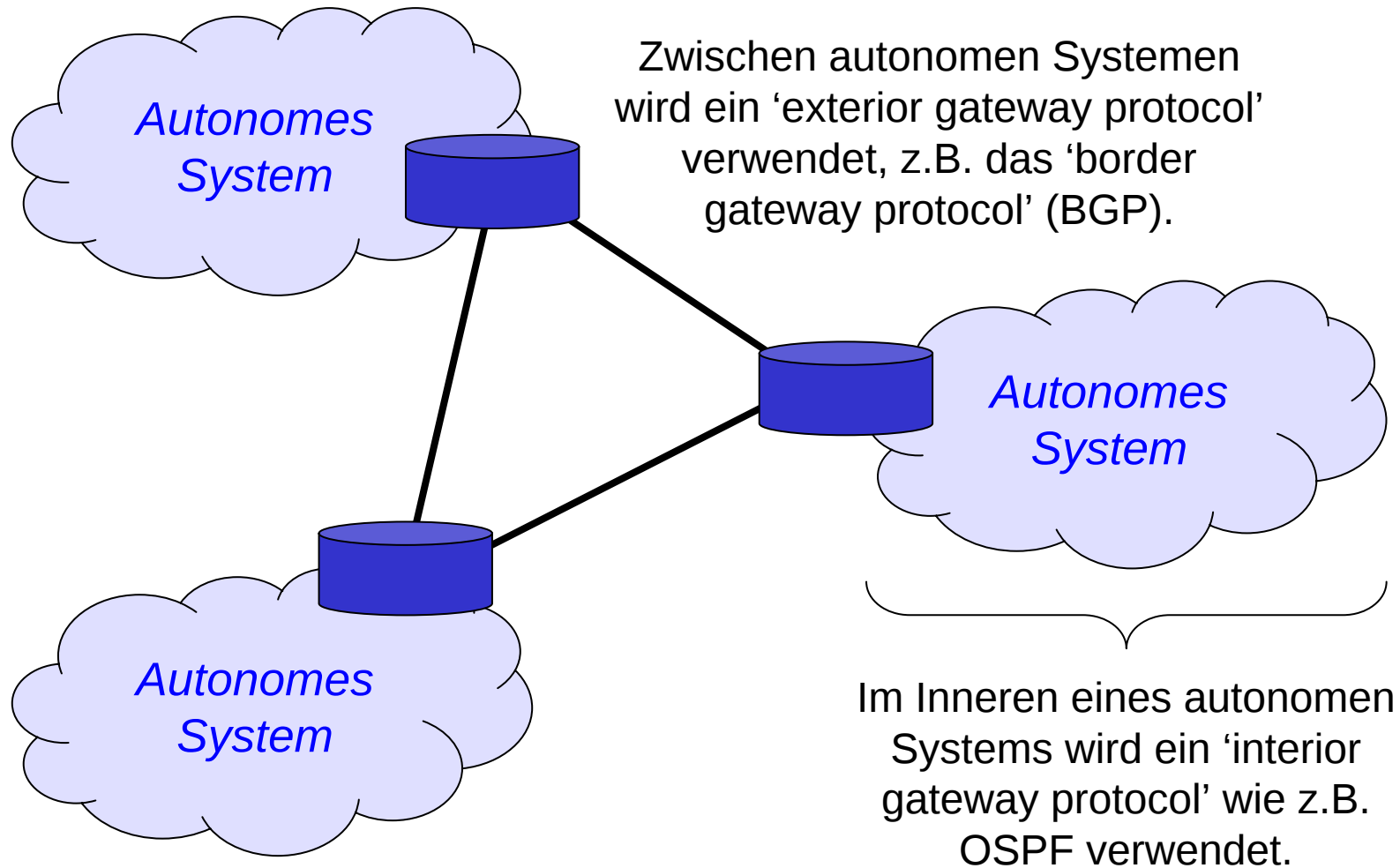
Back-of-the-Envelope Rechnung:

(= schnelle Rechnung ohne Details zu beachten)

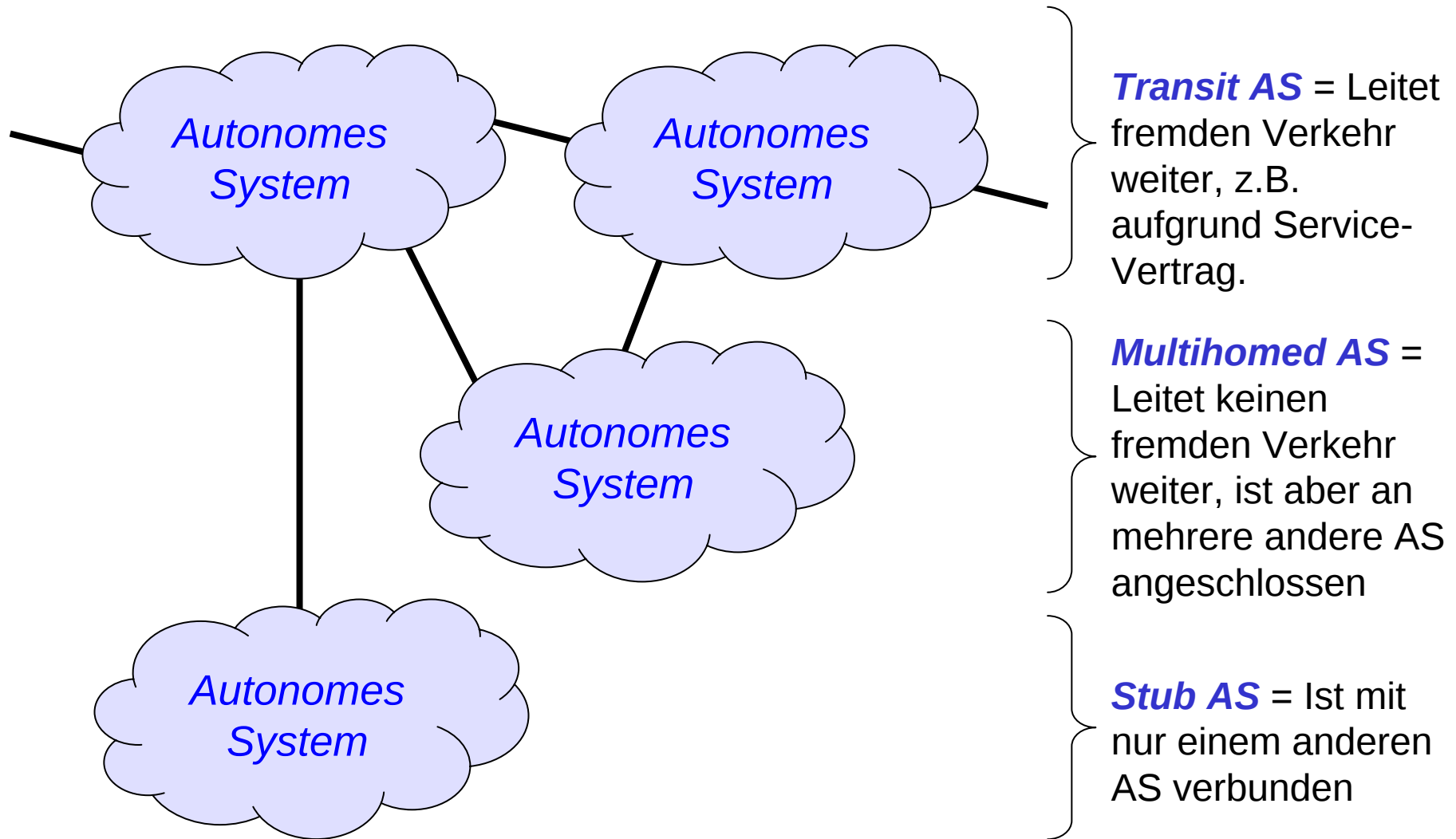
- Im Internet gibt es ca. 800 Mio Computer
- Angenommen ca. 200 Computer pro Subnetz
→ 4 Mio Subnetze
- Angenommen ca. einmal im Jahr ändert sich die Anbindung des Subnetzes → 10.000 Routenänderungen pro Tag, d.h. alle 9 Sekunden muss neue Routing-Tabelle mit 4 Mio Einträgen berechnet werden.

Das Internet funktioniert nur, weil es eine weitere Hierarchiestufe oberhalb der Subnetze gibt.

Autonome Systeme (1)



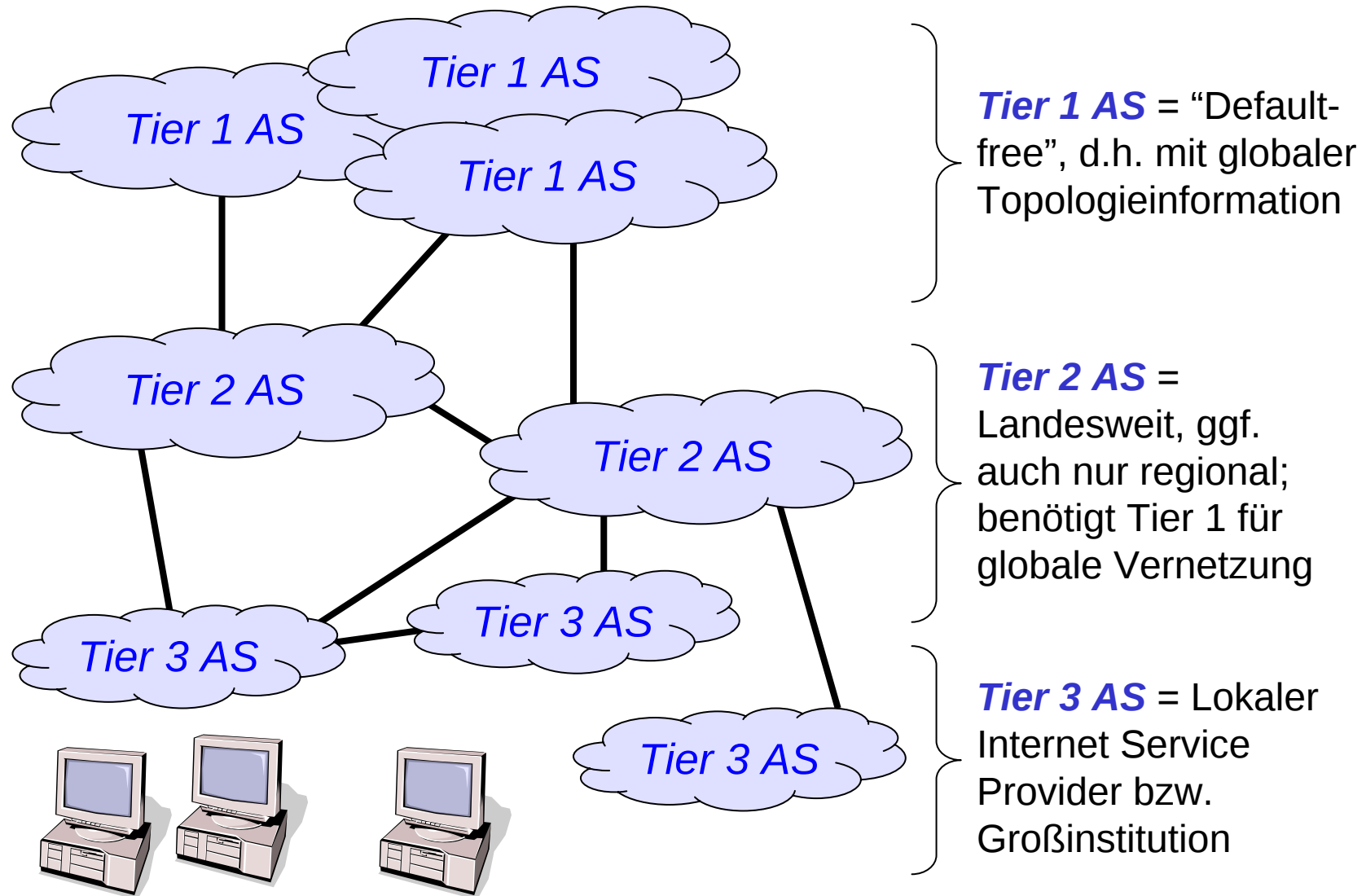
Autonome Systeme (2)



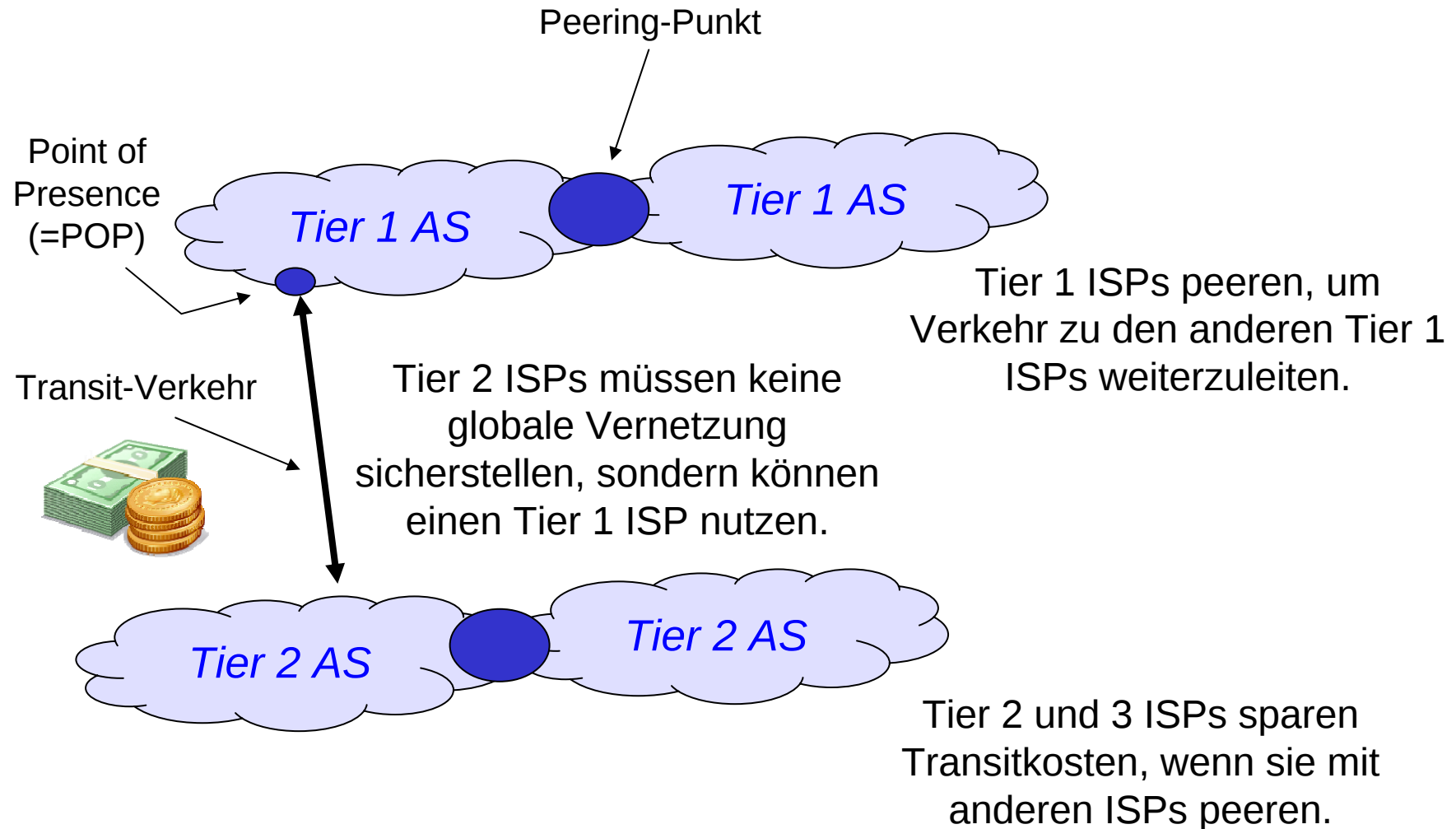
Autonome Systeme (3)

- Die Einteilung in Autonome Systeme (AS) reduziert die Komplexität des Internets:
 - Innere Router kennen nicht die gesamte Netzwerktopologie, sondern nur die ihres eigenen AS
 - Innere Router kennen die über die Grenzrouter erreichbaren Präfixe
 - Grenzrouter kennen die Präfixe der anderen AS
- Auswahl des Next-Hop-AS oft aufgrund von Policies
 - Verkehrsverträge regeln das Geschäft zwischen den Providern
 - Policies spiegeln z.B. Transitkosten oder Peering-Abkommen wider
 - Policies können auch Sicherheitsaspekte reflektieren, z.B. kein Regierungsverkehr durch das feindliche Elbonien weiterleiten.
- Struktur der AS-Beziehungen meist durch finanzielle Beziehungen geregelt (vgl. Policies)
 - Nationale oder lokale Internet-Service-Provider stellen die Versorgung vor Ort sicher und kaufen Konnektivität von globalen Anbietern
 - Internet-Service-Provider schließen sich an Peering-Punkten zusammen, z.B. um Kosten zu sparen.

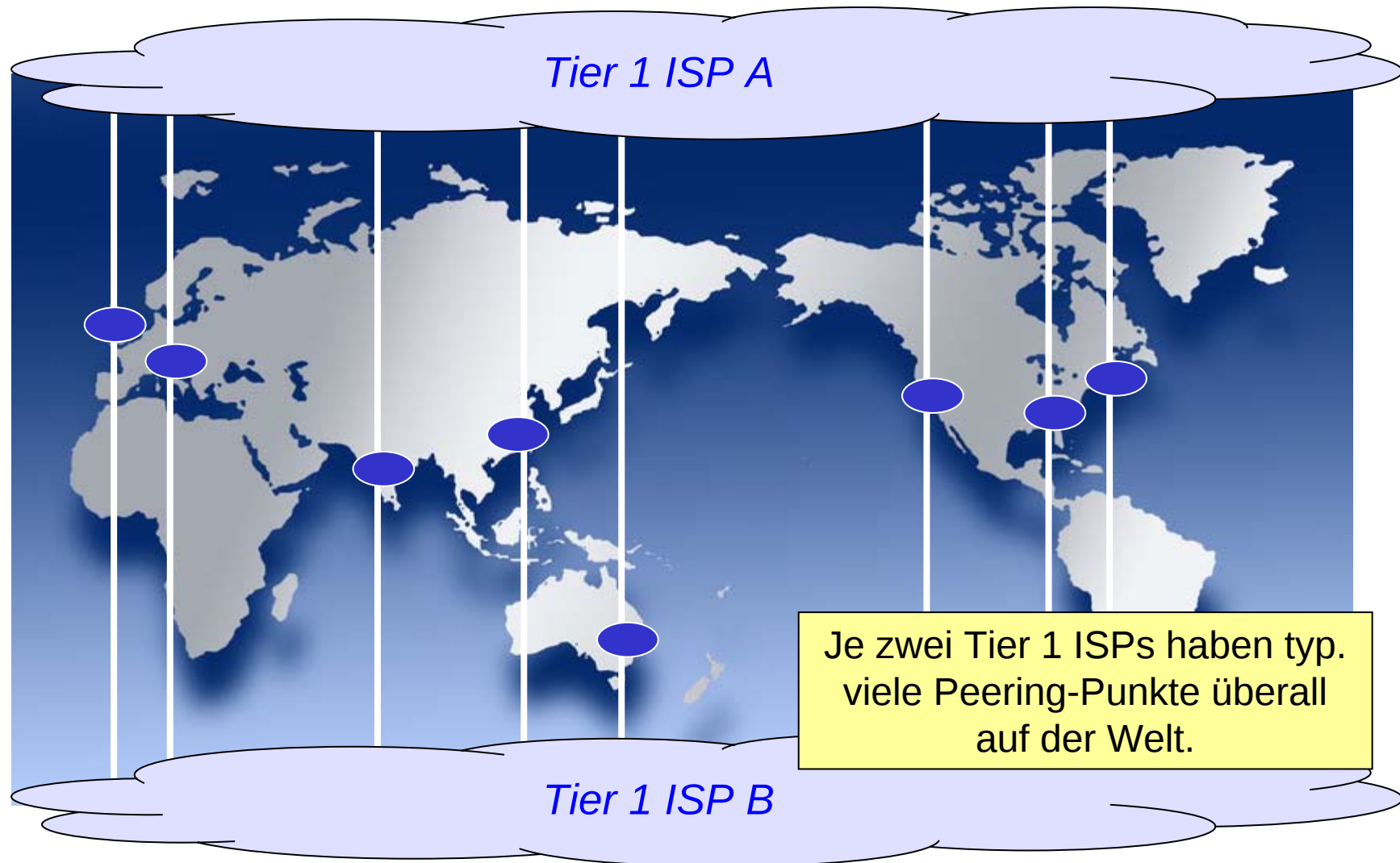
Tier 1, 2 und 3 Netze



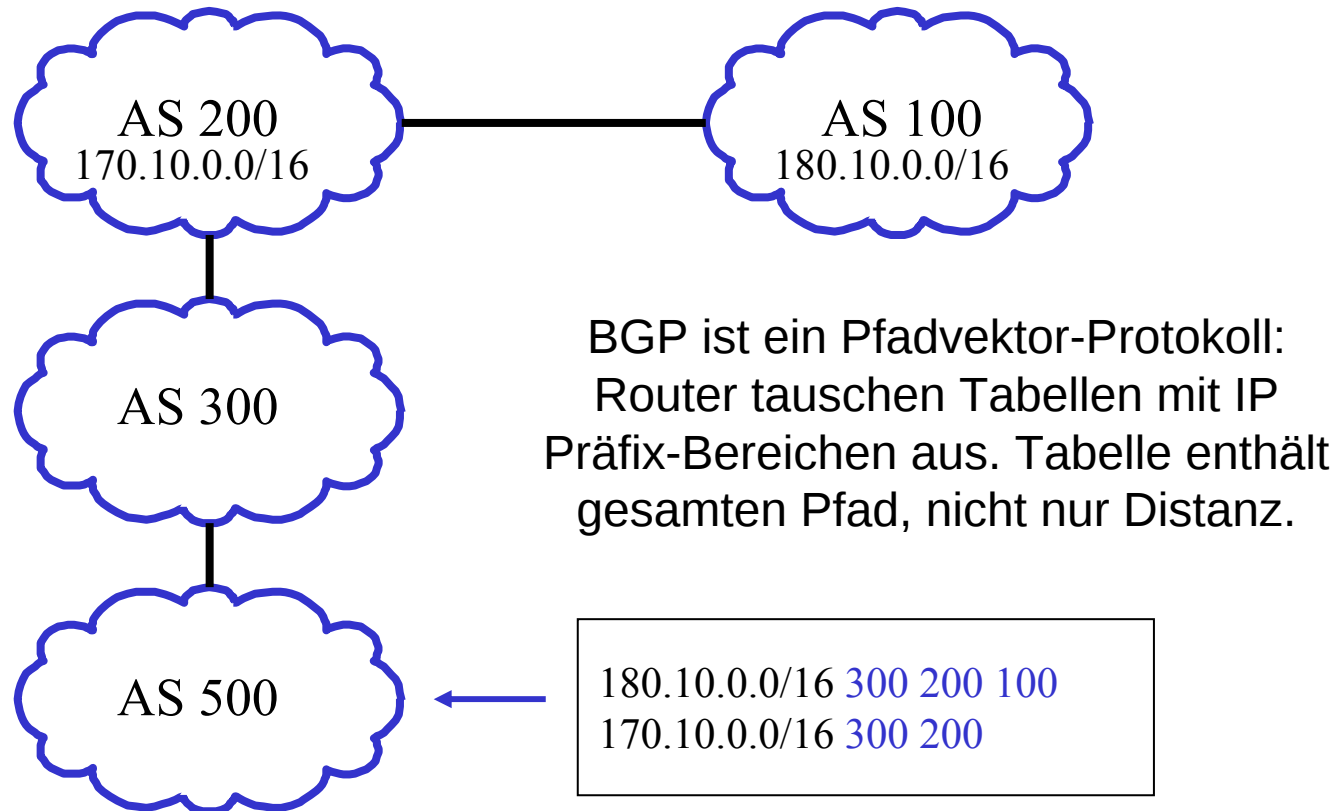
Peering zwischen ISPs (1)



Peering zwischen ISPs (2)

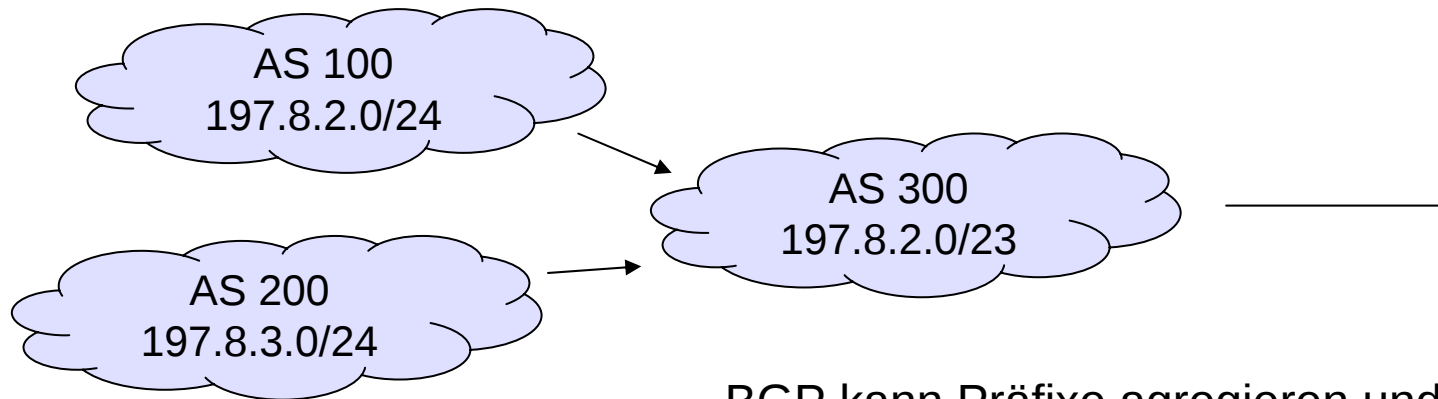


Border Gateway Protocol (1)



Beachte: AS tragen eigene 16-Bit Adressen und kündigen ggf. mehrere IP-Adressbereiche an.

Border Gateway Protocol (2)



BGP kann Präfixe agregieren und so die Größe der Tabellen beschränken. Policies schränken ggf. die Größe weiter ein.

AS und BGP – Zusammenfassung

- Das Internet muss aufgrund seiner Größe in autonome Systeme gegliedert werden.
- Die autonomen Systeme spiegeln die Organisationsstruktur wieder
 - Tier 1 AS erstrecken sich typischerweise über die gesamte Welt
 - In einer Stadt gibt es typischerweise mehrere AS
- Verschiedene AS sind über Points of Presence (POP) oder Peering Punkte verbunden.
 - Zwei AS können an mehreren Punkten verbunden sein
 - Verbindungen spiegeln die Vertragsbeziehungen wider
 - Peering = gegenseitiger Nutzen
 - POP = Allgemeiner, z.B. Dienstleistungsvertrag

Questions?



Thomas Fuhrmann

Department of Informatics
Self-Organizing Systems Group
c/o I8 Network Architectures and Services
Technical University Munich, Germany

fuhrmann@net.in.tum.de