

Systeme II

4. Die Vermittlungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 13.06.2017

■ Circuit Switching

- Etablierung einer Verbindung zwischen lokalen Benutzern durch Schaltstellen
 - mit expliziter Zuordnung von realen Schaltkreisen
 - oder expliziter Zuordnung von virtuellen Ressourcen, z.B. Slots
- Quality of Service einfach, außer bei
 - Leitungsaufbau
 - Leitungsdauer
- Problem
 - Statische Zuordnung
 - Ineffiziente Ausnutzung des Kommunikationsmedium bei dynamischer Last
- Anwendung
 - Telefon
 - Telegraf
 - Funkverbindung

■ Packet Switching

- Grundprinzip von IP
 - Daten werden in Pakete aufgeteilt und mit Absender/Ziel-Information unabhängig versandt
- Problem: Quality of Service
 - Die Qualität der Verbindung hängt von einzelnen Paketen ab
 - Entweder Zwischenspeichern oder Paketverlust
- Vorteil:
 - Effiziente Ausnutzung des Mediums bei dynamischer Last

■ Resümee

- Packet Switching hat Circuit Switching in praktisch allen Anwendungen abgelöst
- Grund:
 - Effiziente Ausnutzung des Mediums

■ Transport

- muss gewisse Flusskontrolle gewährleisten
- z.B. Fairness zwischen gleichzeitigen Datenströmen

■ Vermittlung

- Quality of Service (virtuelles Circuit Switching)

■ Sicherung

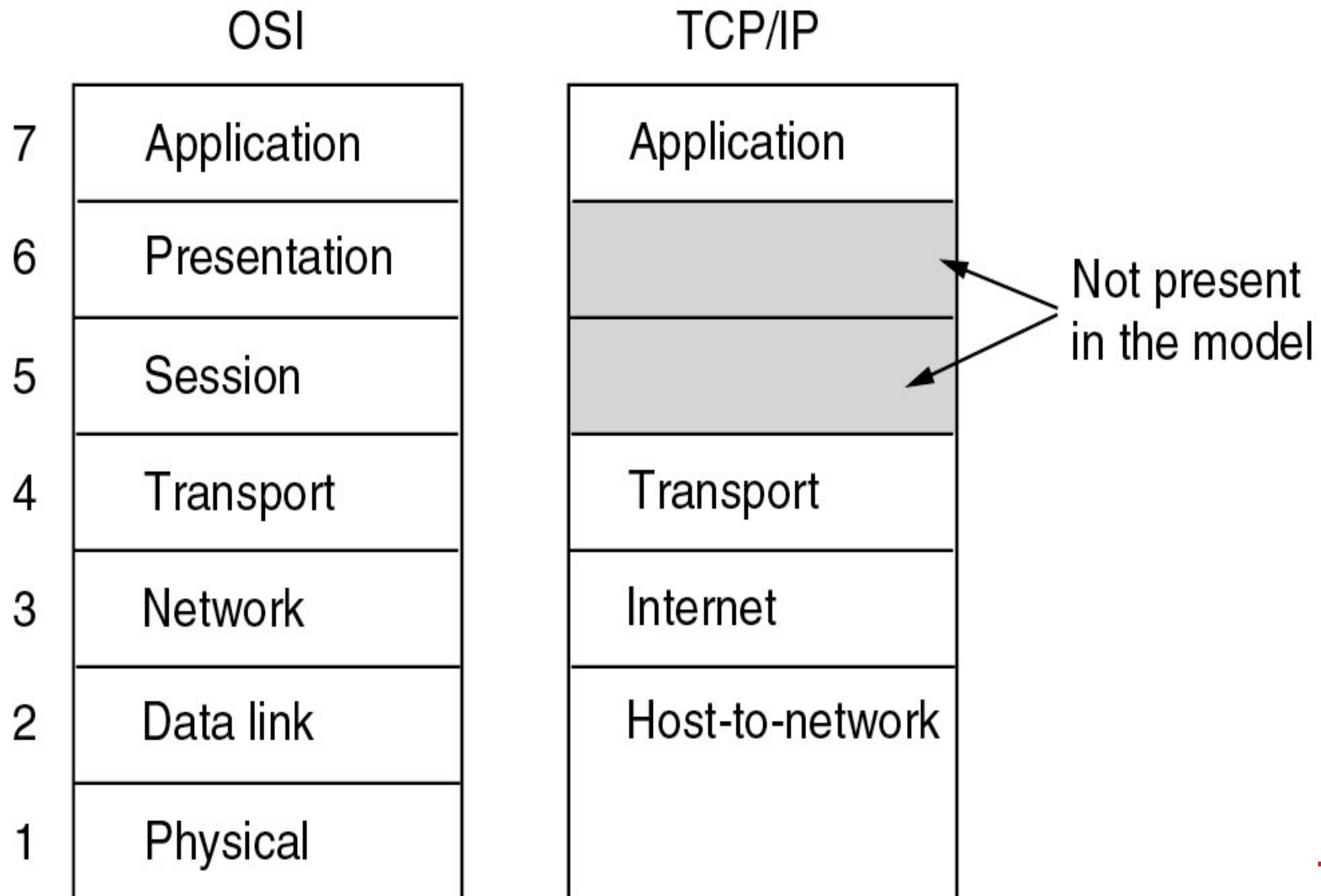
- Flusskontrolle zur Auslastung des Kanals

Layer	Policies
Transport	<ul style="list-style-type: none"> • Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	<ul style="list-style-type: none"> • Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	<ul style="list-style-type: none"> • Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Die Schichtung des Internets - TCP/IP-Layer

Anwendung	Application	Telnet, FTP, HTTP, SMTP (E-Mail), ...
Transport	Transport	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Vermittlung	Network	<u>IP (Internet Protocol)</u> + ICMP (Internet Control Message Protocol) + IGMP (Internet Group Management Protocol)
Verbindung	Host-to-network	LAN (z.B. Ethernet, Token Ring etc.)

OSI versus TCP/IP



Warum eine Vermittlungsschicht

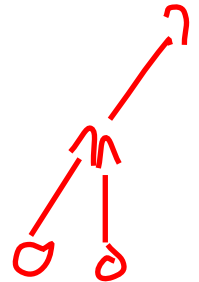
- Lokale Netzwerke können nicht nur über Hubs, Switches oder Bridges verknüpft werden
 - Hubs: Kollisionen nehmen überhand
 - Switches:
 - Routen-Information durch Beobachtung der Daten ineffizient
 - Broadcast aller Nachrichten schafft Probleme
 - Es gibt über 100 Mio. lokale Netzwerke im Internet...
- Zur Beförderung von Paketen in großen Netzwerken braucht man Routeninformationen
 - Wie baut man diese auf?
 - Wie leitet man Pakete weiter?
- Das Internet-Protokoll ist im wesentlichen ein Vermittlungsschichtprotokoll

■ IP-Routing-Tabelle

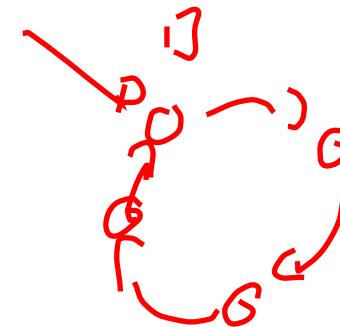
- enthält für Ziel (Destination) die Adresse des nächsten Rechners (Gateway)
- Destination kann einen Rechner oder ganze Sub-nets beschreiben
- Zusätzlich wird ein Default-Gateway angegeben

■ Packet Forwarding

- früher Packet Routing genannt
- IP-Paket (datagram) enthält Start-IP-Adresse und Ziel-IP-Adresse
 - Ist Ziel-IP-Adresse = eigene Rechneradresse dann Nachricht ausgeliefert
 - Ist Ziel-IP-Adresse in Routing-Tabelle dann leite Paket zum angegebenen Gateway
 - Ist Ziel-IP-Subnetz in Routing-Tabelle dann leite Paket zum angegebenen Gateway
 - Ansonsten leite zum Default-Gateway



- IP-Paket (datagram) enthält unter anderen
 - TTL (Time-to-Live): Anzahl der Hops *IPv6 hop-count*
 - Start-IP-Adresse
 - Ziel-IP-Adresse
- Behandlung eines Pakets
 - Verringere TTL (Time to Live) um 1
 - Falls TTL $\neq 0$ dann Packet-Forwarding aufgrund der Routing-Tabelle
 - Falls TTL = 0 oder bei Problemen in Packet-Forwarding:
 - Lösche Paket
 - Falls Paket ist kein ICMP-Paket dann
 - Sende ICMP-Paket mit
 - Start = aktuelle IP-Adresse und
 - Ziel = alte Start-IP-Adresse



Forwarding:

- Weiterleiten von Paketen

Routing:

- Erstellen Routen, d.h.
 - Erstellen der Routing-Tabelle

Statisches Routing

- Tabelle wird manuell erstellt
- sinnvoll für kleine und stabile LANs

Dynamisches Routing

- Tabellen werden durch Routing-Algorithmus erstellt
- ~~Zentraler~~ ["]Algorithmus, z.B. Link State
 - Einer/jeder kennt alle Information, muss diese erfahren
- Dezentraler Algorithmus, z.B. Distance Vector
 - arbeitet lokal in jedem Router
 - verbreitet lokale Information im Netzwerk

Dijkstra

Distance Vector Routing Protocol

Distance Table Datenstruktur

- Jeder Knoten besitzt eine
 - Zeile für jedes mögliches Ziel
 - Spalte für jeden direkten Nachbarn

Verteilter Algorithmus

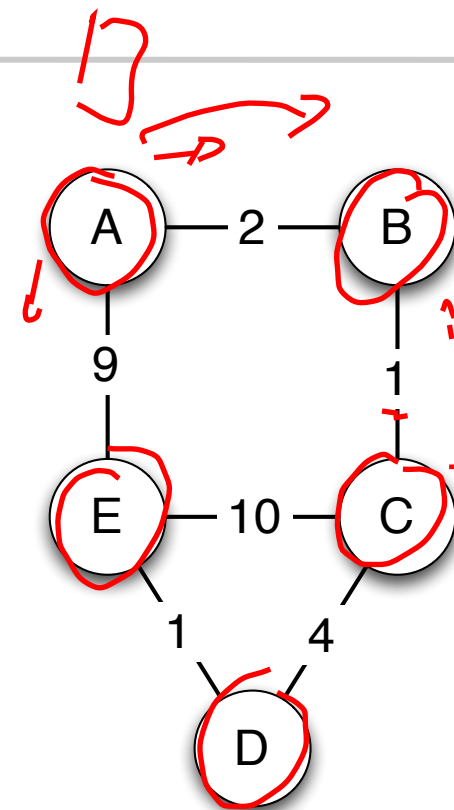
- Jeder Knoten kommuniziert nur mit seinem Nachbarn

Asynchroner Betrieb

- Knoten müssen nicht Informationen austauschen in einer Runde

Selbst Terminierend

- läuft bis die Knoten keine Informationen mehr austauschen



Distance Table für A

von A	über		Routing Tabellen Eintrag
	B	E	
nach B	2	15	B
C	3	14	B
D	7	10	B
E	8	9	E

Distance Table für C

von C	über			Routing Tabellen Eintrag
	B	D	E	
nach A	3	11	18	B
B	1	9	16	B
D	6	4	11	D
E	7	5	10	D

Via

B	A	C	
A	2	4	A
C	5	1	C
D	9	5	C
E	10	6	C

A ∞
A ∞
A ∞
A ∞

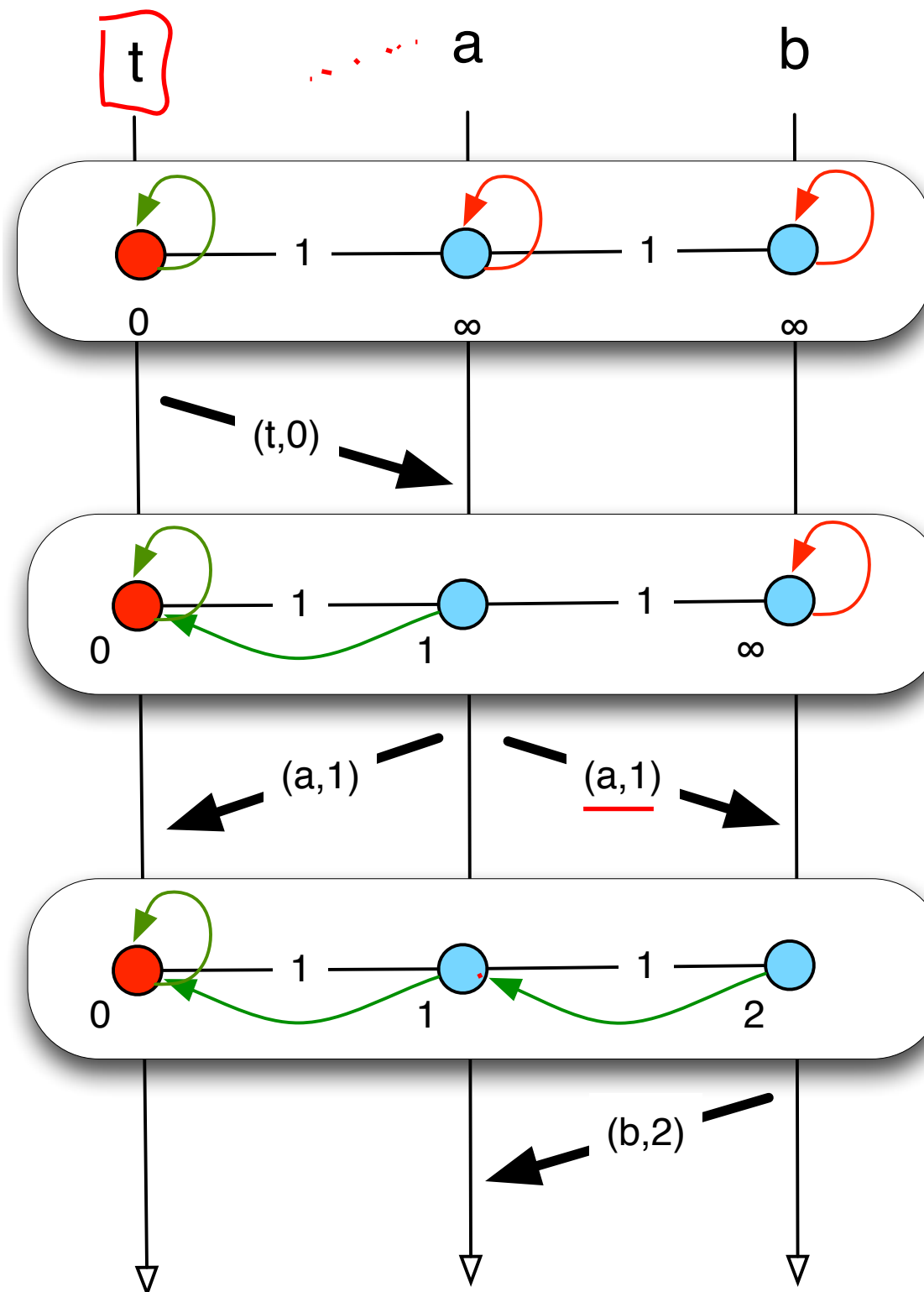
Beispiel für Distance-Vector für Ziel t

Via

	a	t	b
t	∞	∞	
b	∞	∞	∞

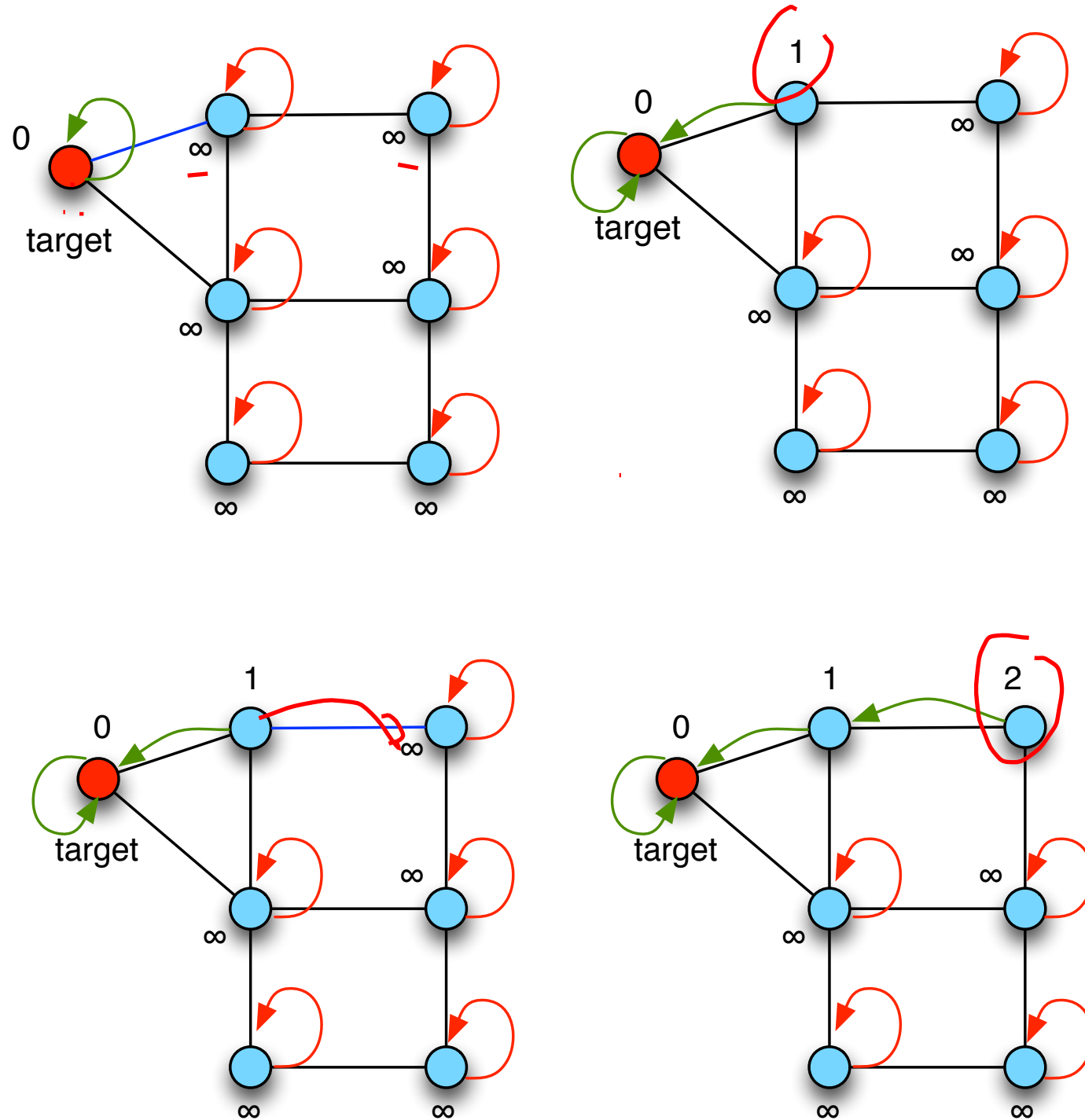
Via

	a	t	b
t	1	∞	
b	3	2	∞

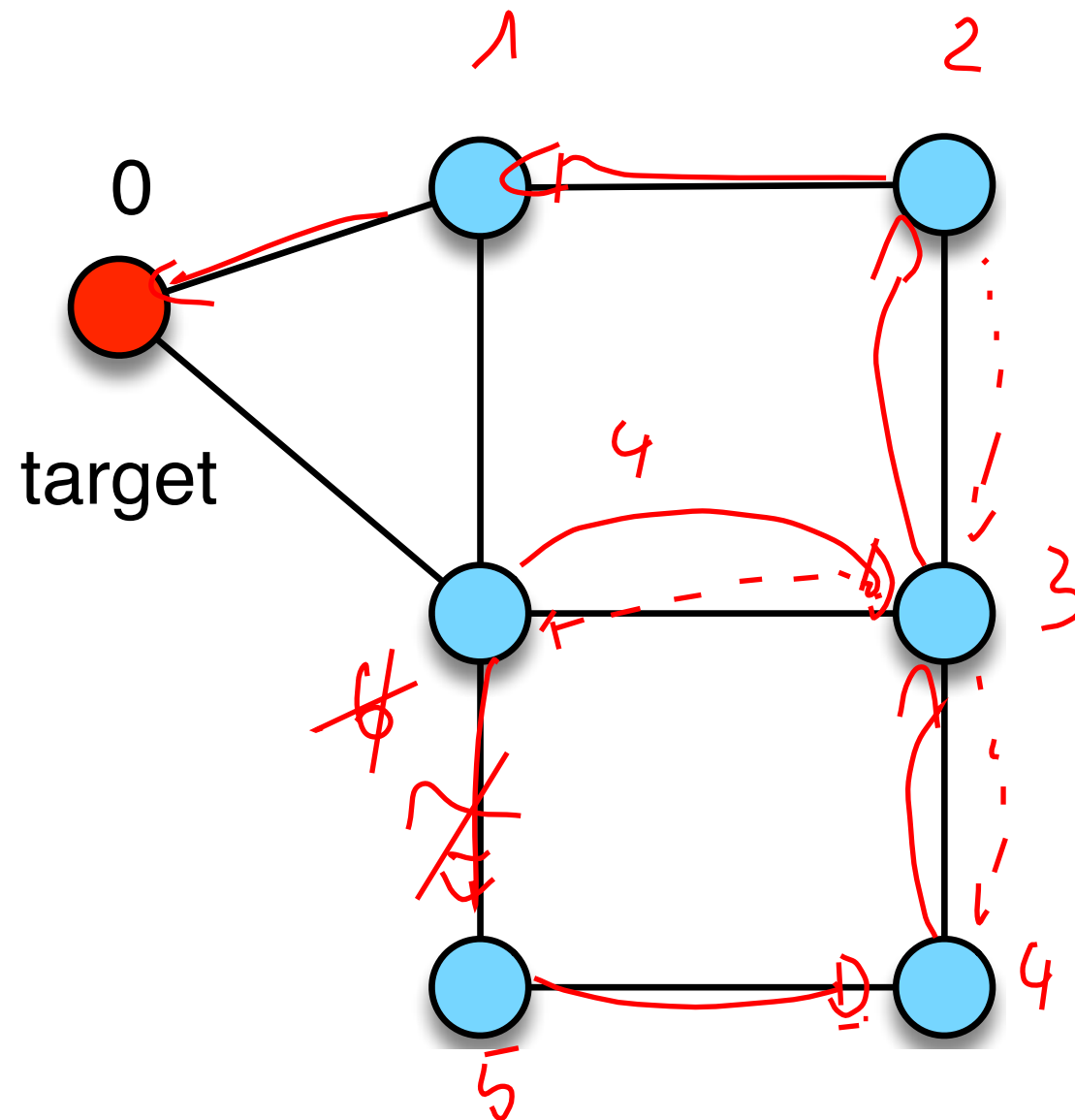


	b	a
t	1	2

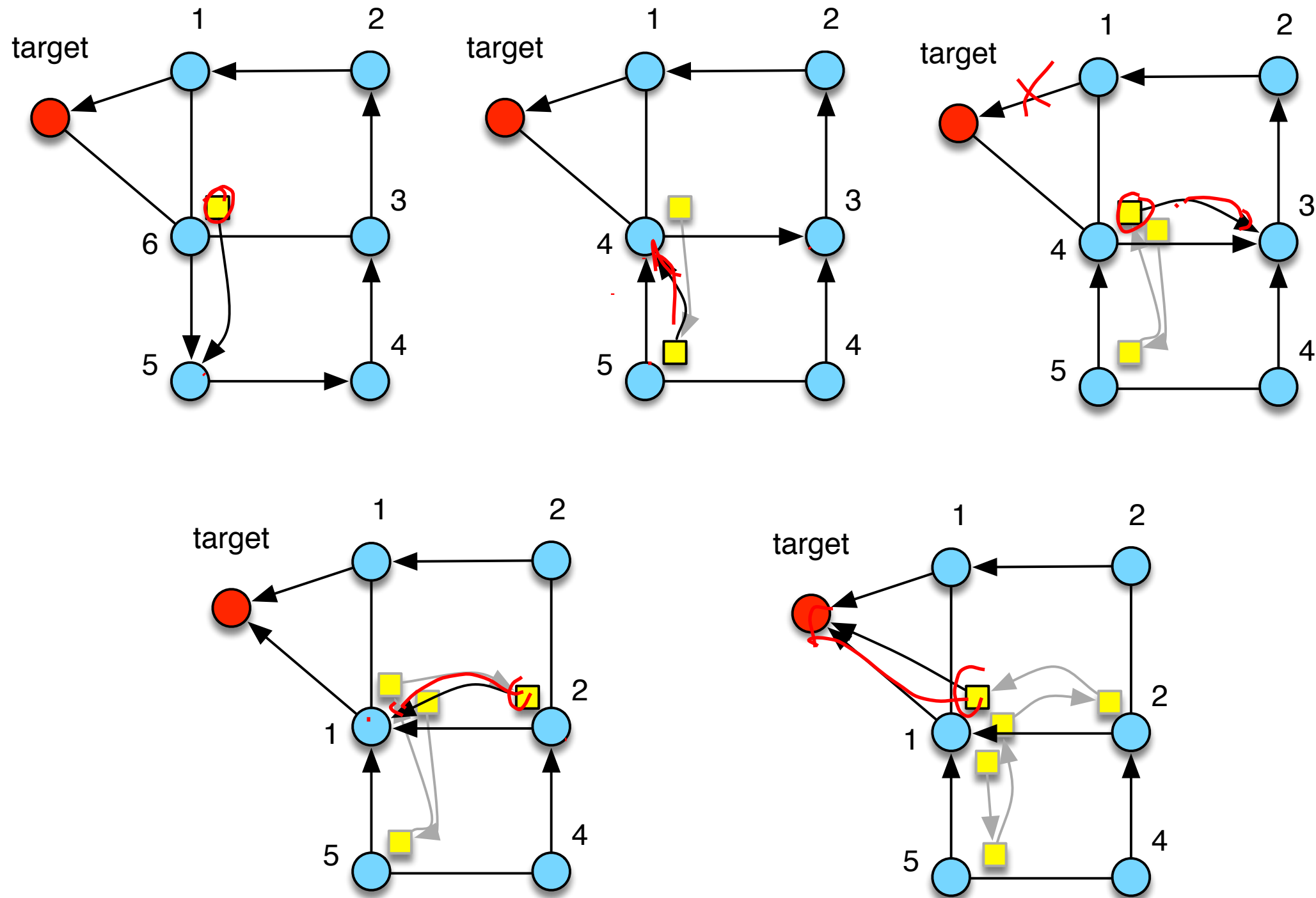
Distance-Vector für ein Ziel



Distance-Vector für ein Ziel

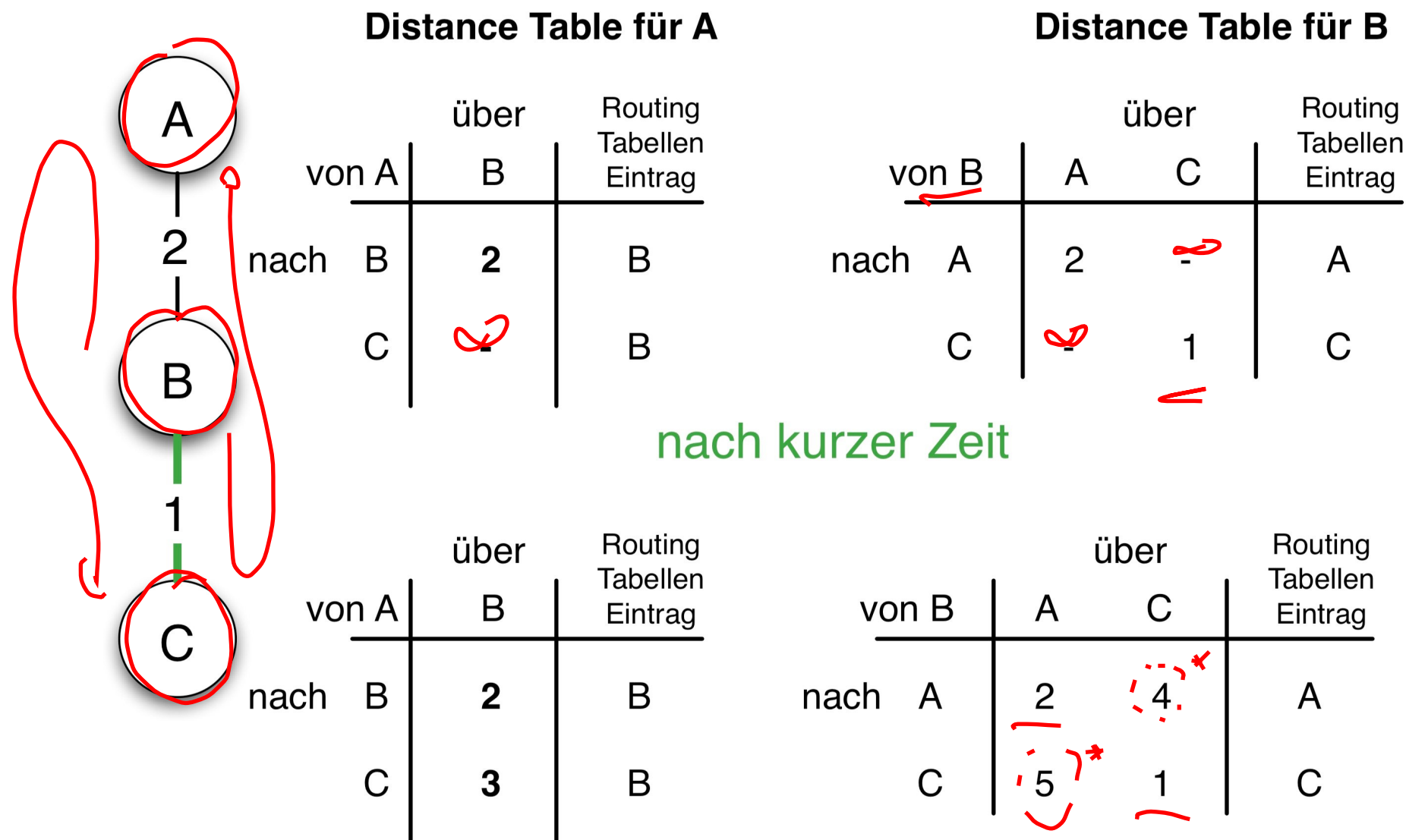


Irrlichter im Routing



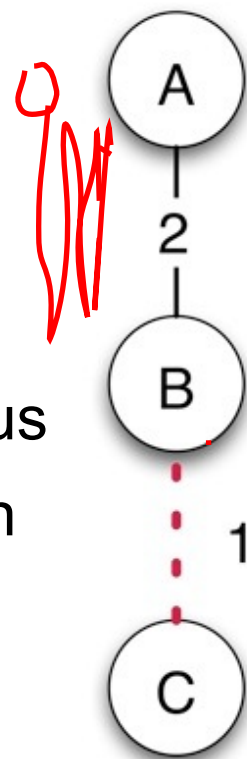
Das “Count to Infinity” - Problem

- Gute Nachrichten verbreiten sich schnell
 - Neue Verbindung wird schnell veröffentlicht



Das “Count to Infinity” - Problem

- Schlechte Nachrichten verbreiten sich langsam
 - Verbindung fällt aus
 - Nachbarn erhöhen wechselseitig ihre Entfernung
 - “Count to Infinity”-Problem



		über	Routing Tabellen Eintrag
von A		B	
nach	B	2	B
	C	3	B

		über	Routing Tabellen Eintrag
von A		B	
nach	B	2	B
	C	7	B

		über	Routing Tabellen Eintrag
von A		B	
nach	B	2	B
	C	7	B

		über	Routing Tabellen Eintrag
von B		A C	
nach	A	2 -	A
	C	5 -	A

		über	Routing Tabellen Eintrag
von B		A C	
nach	A	2 -	A
	C	5 -	A

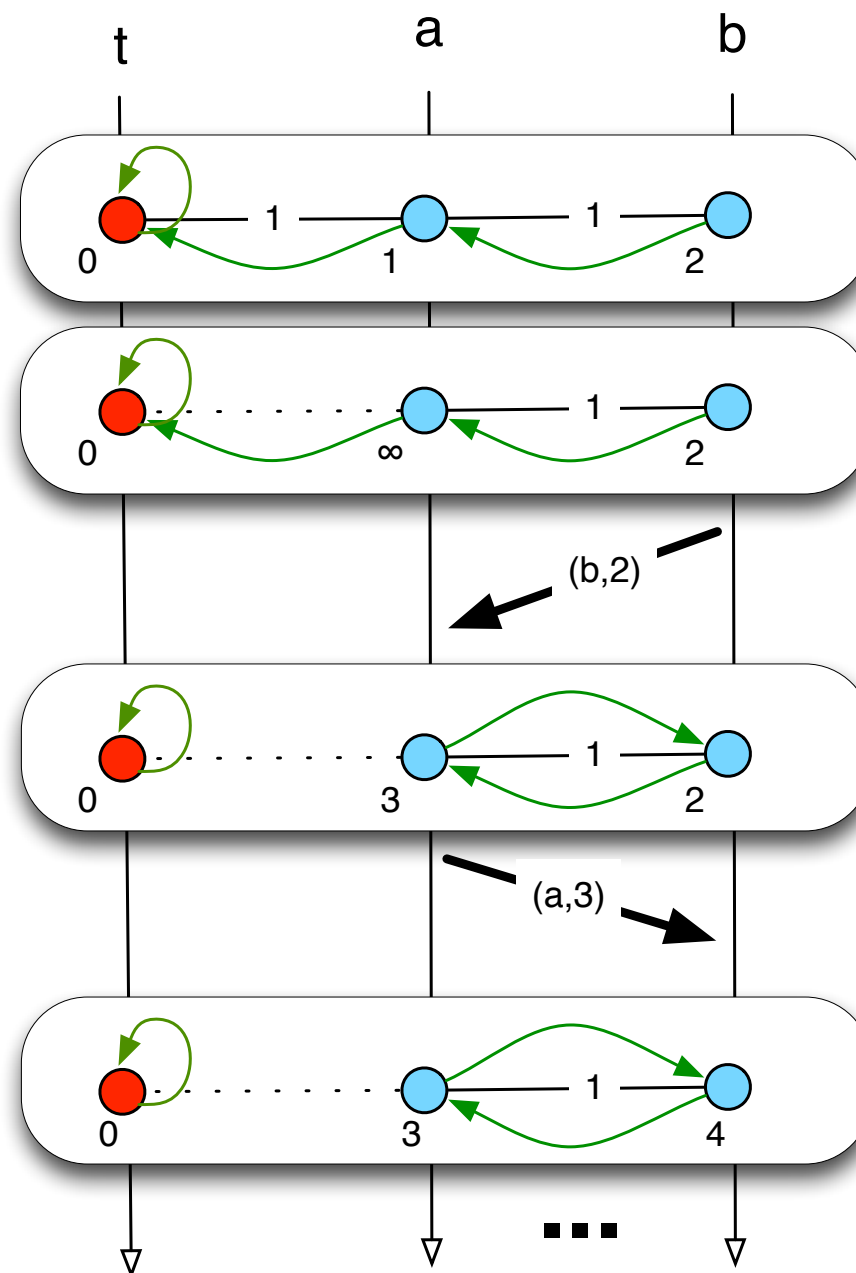
		über	Routing Tabellen Eintrag
von B		A C	
nach	A	2 -	A
	C	9 -	A

11

15

13

Das “Count to Infinity” - Problem für Ziel t





■ Link State Router

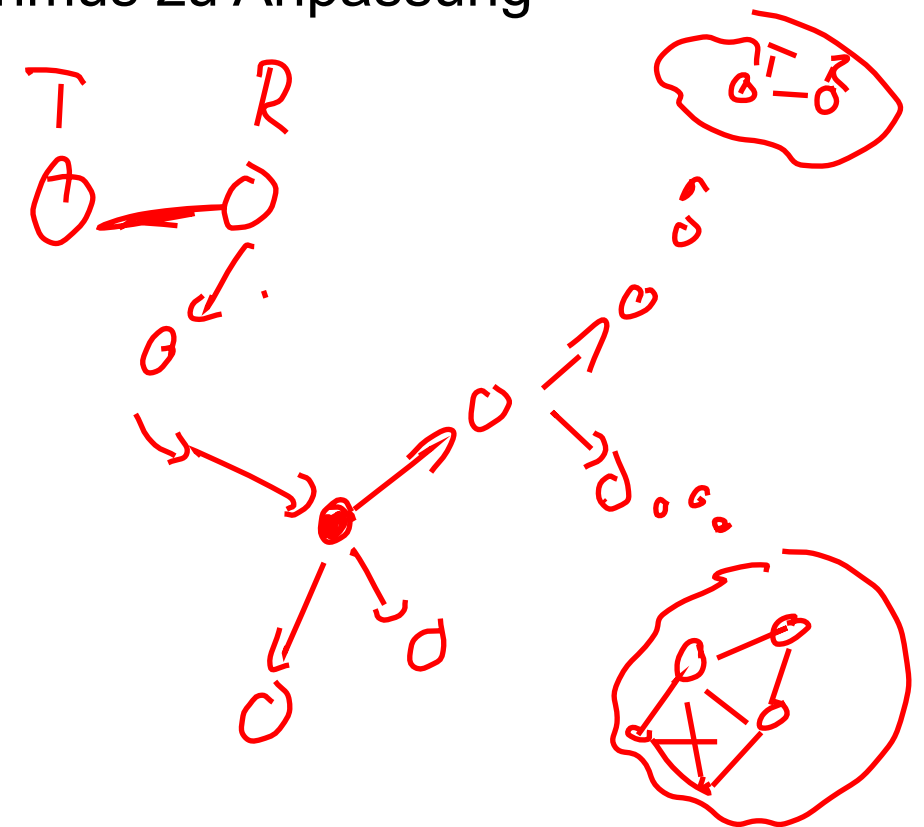
- tauschen Information mittels Link State Packets (LSP) aus
- Jeder verwendet einen eigenen Kürzeste-Wege-Algorithmus zu Anpassung der Routing-Tabelle

■ LSP enthält

- ID des LSP erzeugenden Knotens
- Kosten dieses Knotens zu jedem direkten Nachbarn
- Sequenznr. (SEQNO)
- TTL-Feld für dieses Feld (time to live)

■ Verlässliches Fluten (Reliable Flooding)

- Die aktuellen LSP jedes Knoten werden gespeichert
- Weiterleitung der LSP zu allen Nachbarn
 - bis auf den Knoten der diese ausgeliefert hat
- Periodisches Erzeugen neuer LSPs
 - mit steigender SEQNOs
- Verringern der TTL bei jedem Weiterleiten



Die Grenzen des flachen Routing

- Link State Routing
 - benötigt $O(g \cdot n)$ Einträge für n Router mit maximalen Grad g
 - Jeder Knoten muss an jeden anderen seine Informationen senden
- Distance Vector
 - benötigt $O(g \cdot n)$ Einträge
 - kann Schleifen einrichten *← temporär*
 - Konvergenzzeit steigt mit Netzwerkgröße
- Im Internet gibt es mehr als 10^7 Router
 - damit sind diese so genannten flachen Verfahren nicht einsetzbar
- Lösung:
 - Hierarchisches Routing

- Autonomous System (AS)
 - liefert ein zwei Schichten-Modell des Routing im Internet
 - Beispiele für AS:
 - uni-freiburg.de
- Intra-AS-Routing (Interior Gateway Protocol)
 - ist Routing innerhalb der AS
 - z.B. RIP, OSPF, IGRP, ... ⁴
- Inter-AS-Routing (Exterior Gateway Protocol)
 - Übergabepunkte sind Gateways
 - ist vollkommen dezentrales Routing
 - Jeder kann seine Optimierungskriterien vorgeben
 - z.B. EGP (früher), BGP

