

# ZSL

Zentrum für Schulqualität  
und Lehrerbildung  
Baden-Württemberg



Networking  
Academy



## STP Concepts

### Algorhyme

*I think that I shall never see  
A graph more lovely than a tree.*

*A tree whose crucial property  
Is loop-free connectivity.*

*A tree which must be sure to span  
So packets can reach every LAN.*

*First the Root must be selected.  
By ID it is elected.*

*Least cost paths from Root are traced.  
In the tree these paths are placed.*

*A mesh is made by folks like me  
Then bridges find a spanning tree.*

by Radia Perlman, 1985, Erfinderin von u.a. STP

Andreas Grupp

[Andreas.Grupp@zsl-rstue.de](mailto:Andreas.Grupp@zsl-rstue.de)

Carina Haag

[haag.c@lanz.schule](mailto:haag.c@lanz.schule)

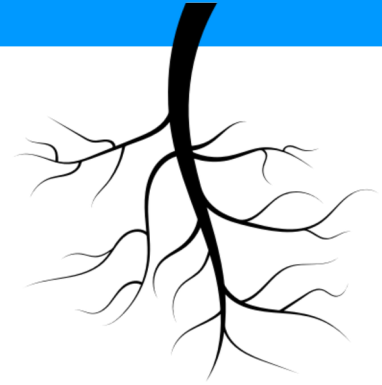
Tobias Heine

[tobias.heine@springer-schule.de](mailto:tobias.heine@springer-schule.de)

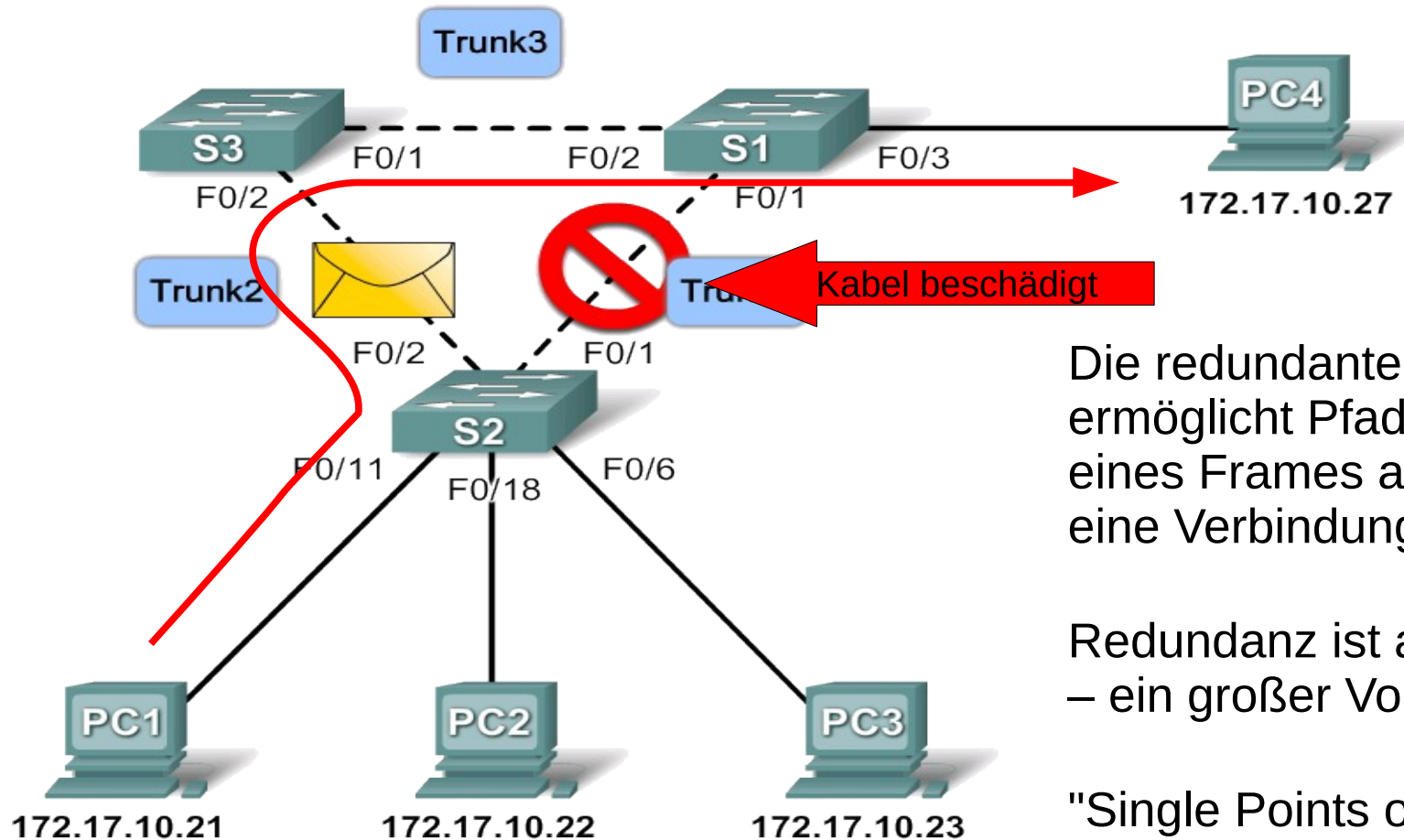
Uwe Thiessat

[uwe.thiessat@gbs-sha.de](mailto:uwe.thiessat@gbs-sha.de)

- Topologie eines Ethernet-LANs → Baumstruktur
  - Äste entsprechen Leitungen
  - An den Gabelungen sind Switches
- Fällt Leitung / Switch aus
  - nachfolgende Bereiche werden nicht mehr versorgt
- Lösung:
  - redundante An- bzw. Verbindungen wie bei IP-Netzen
  - Vermaschung
  - Quasi wie Internet



# Vorteile einer redundanten Verkabelung im LAN

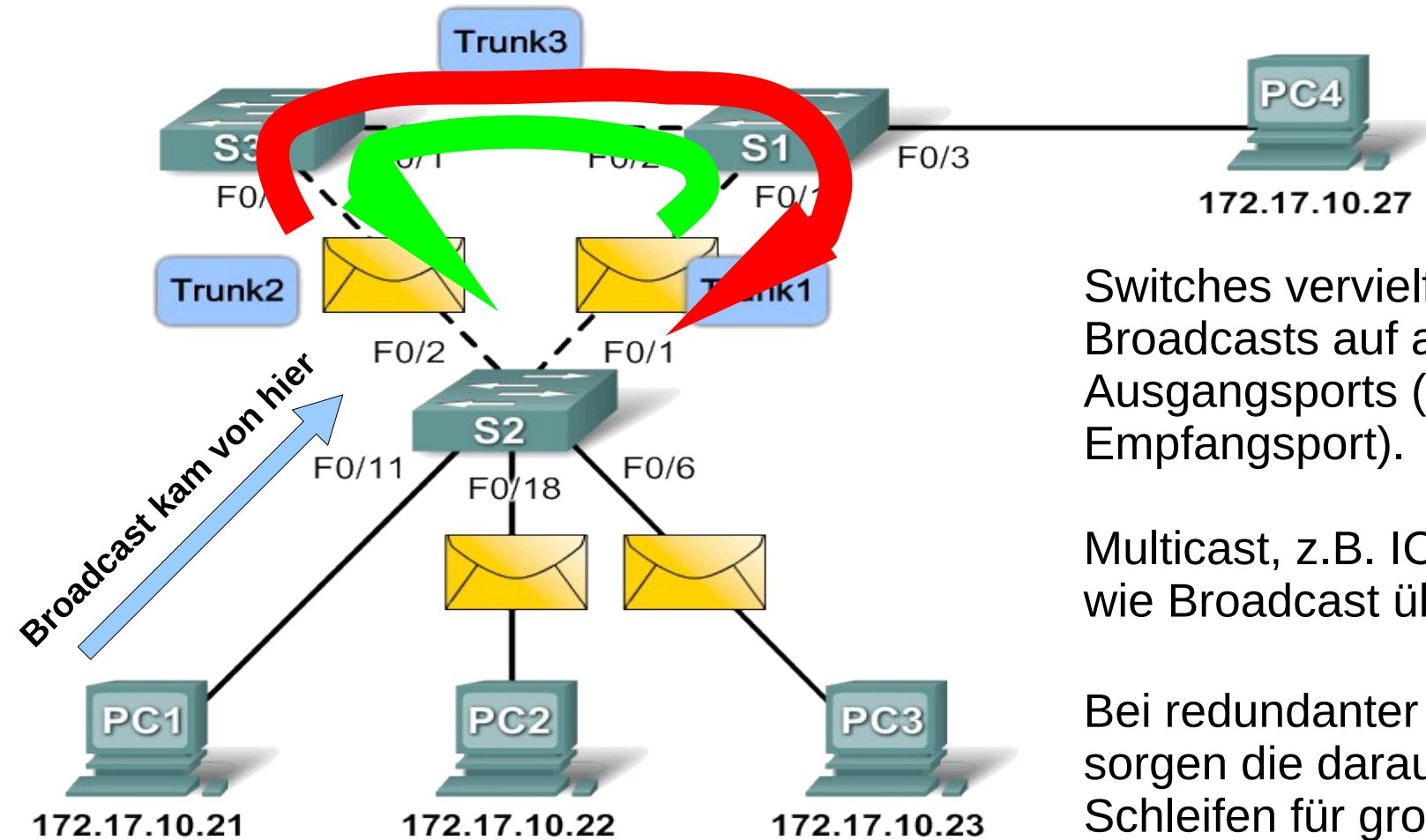


Die redundante Verkabelung ermöglicht Pfade zur Zustellung eines Frames auch dann wenn eine Verbindung ausfällt!

Redundanz ist also – wie so oft – ein großer Vorteil!

"Single Points of Failure" werden hierdurch beseitigt

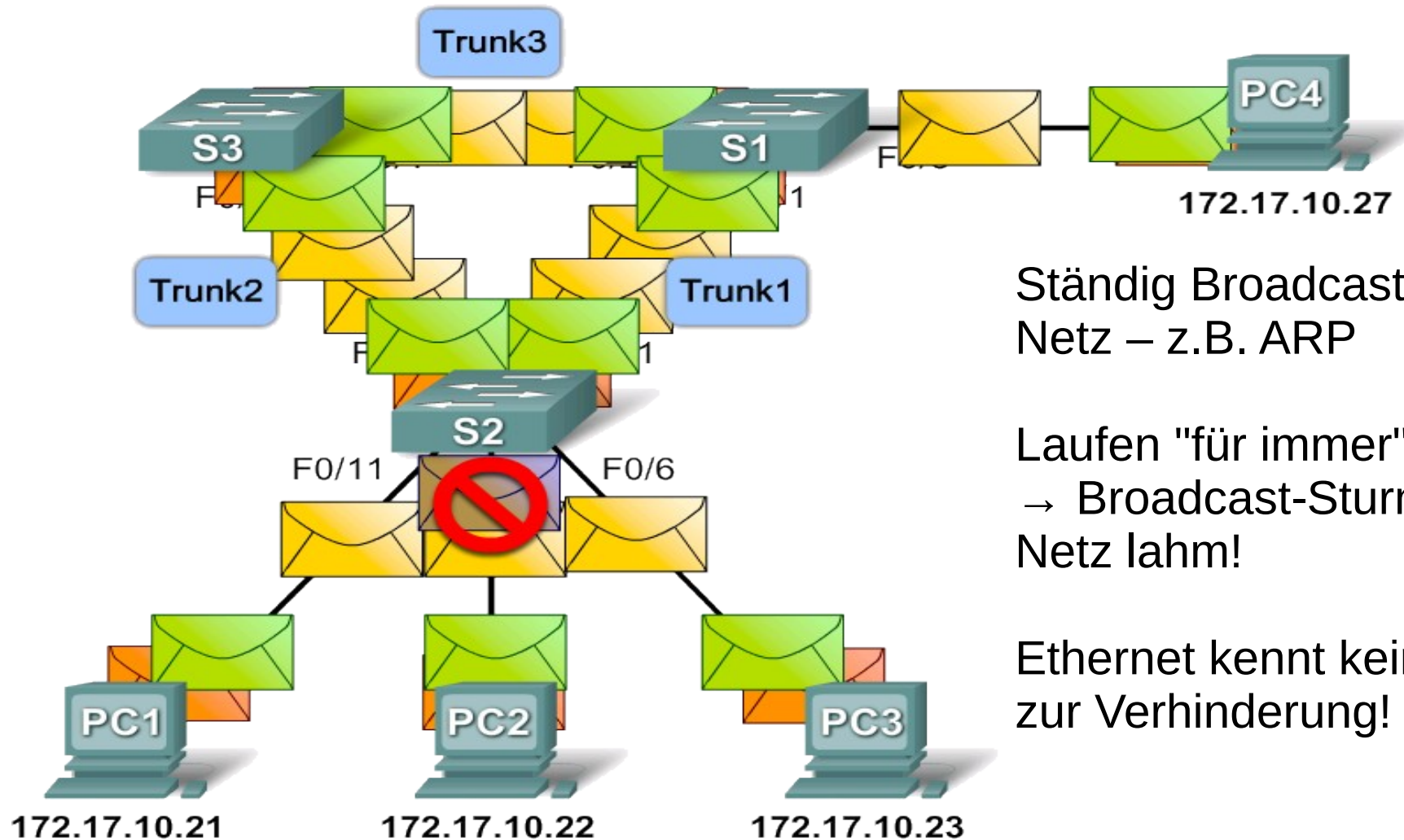
# Aber wo Licht ist, ist auch Schatten :-)



Switches vervielfältigen Broadcasts auf alle Ausgangsports (bis auf den Empfangsport).

Multicast, z.B. ICMPv6-ND, wird wie Broadcast übertragen!

Bei redundanter Verkabelung sorgen die daraus resultierenden Schleifen für große Probleme – wenn nicht Gegenmaßnahmen ergriffen werden!



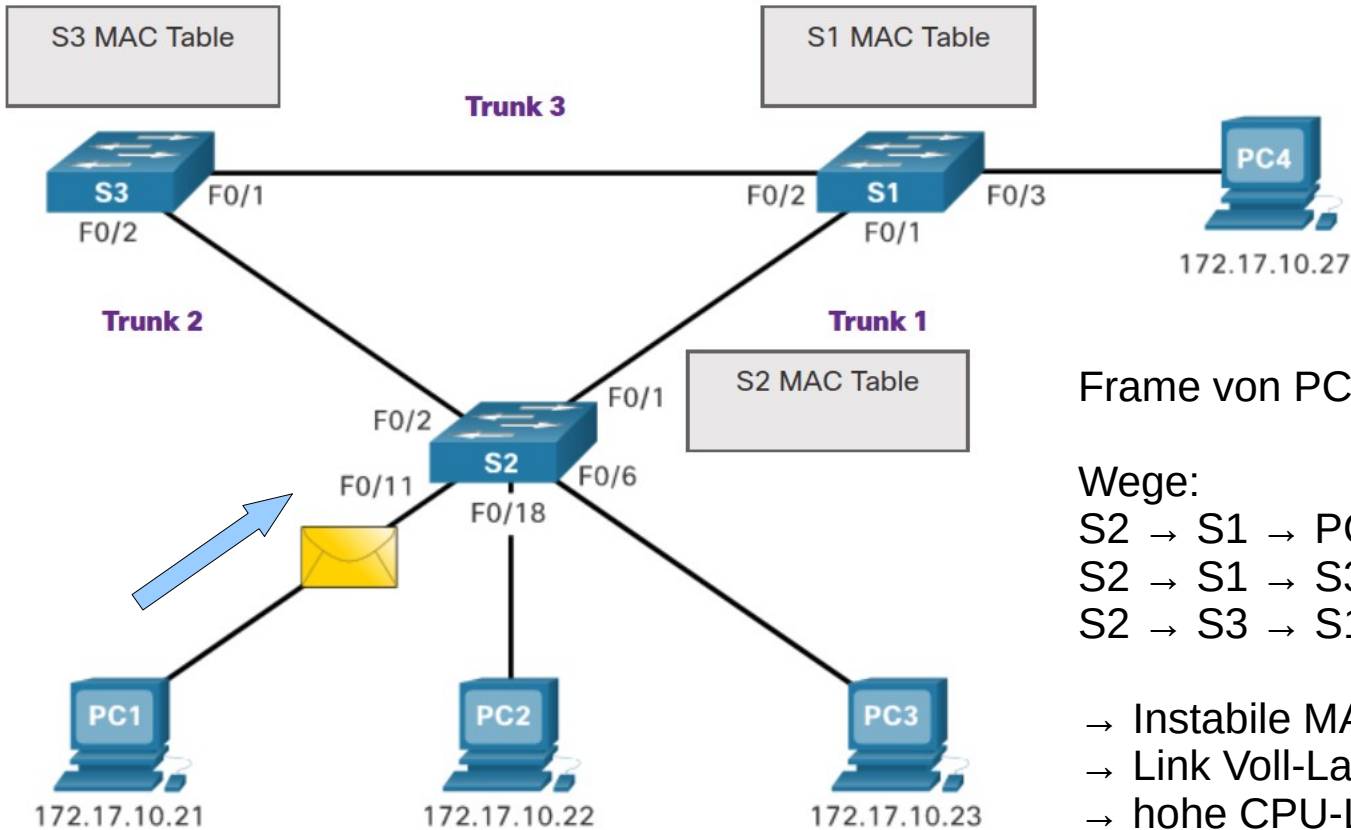
Ständig Broadcasts in einem Netz – z.B. ARP

Laufen "für immer" im Kreis  
→ Broadcast-Sturm legt Netz lahm!

Ethernet kennt keinen Weg zur Verhinderung!



# Probleme bei „Unknown Unicasts“ (Ziel nicht in MAC-Table)



Frame von PC1 → PC4

Wege:

S2 → S1 → PC4, aber auch  
S2 → S1 → S3 → S2, oder  
S2 → S3 → S1 → S2

- Instabile MAC Tables / Fehl-Forwarding
- Link Voll-Last / Nicht-Erreichbarkeit
- hohe CPU-Last bei Switches
- hohe CPU-Last bei Endgeräten

- **Ziel:** Schleifenfreie logische Topologie durch Baumstruktur

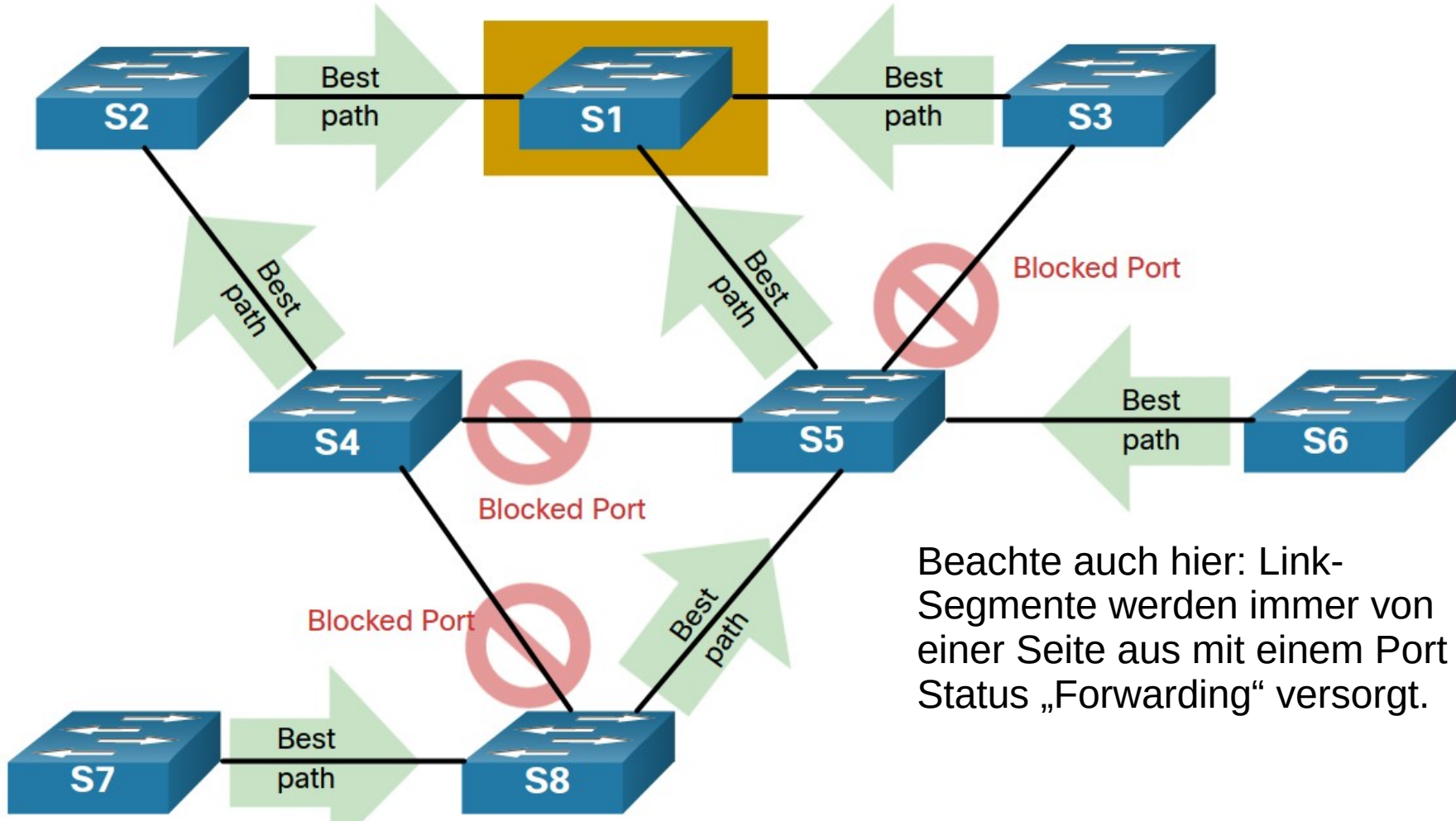
„logisch“ = physikalisch weiterhin mit Schleifen, aber für Datenverkehr (logische Verbindungen) keine Schleifen mehr

STP-Algorithmus basiert auf Arbeit von Radia Perlman

- **Vorgehen:**

- Switches "wählen Baumwurzel" → Root-Bridge
- Es werden zuerst nur die Pfade aktiviert die eine Erreichbarkeit der Root-Bridge mit minimalen „Kosten“ ermöglichen
- Bei Links zwischen zwei Switches die für die Anbindung zur Root-Bridge verworfen wurden, muss eine Seite das Segment "versorgen".
  - Dies erledigt der Switch mit den geringsten Kosten zur Root-Bridge. Bei gleichen Kosten entscheidet die kleinere Bridge-ID.
  - Gegenseite / Port auf der gegenüberliegenden Linkseite blockiert!

# Endzustand nach STA, incl. Ports im Status „Blocking“



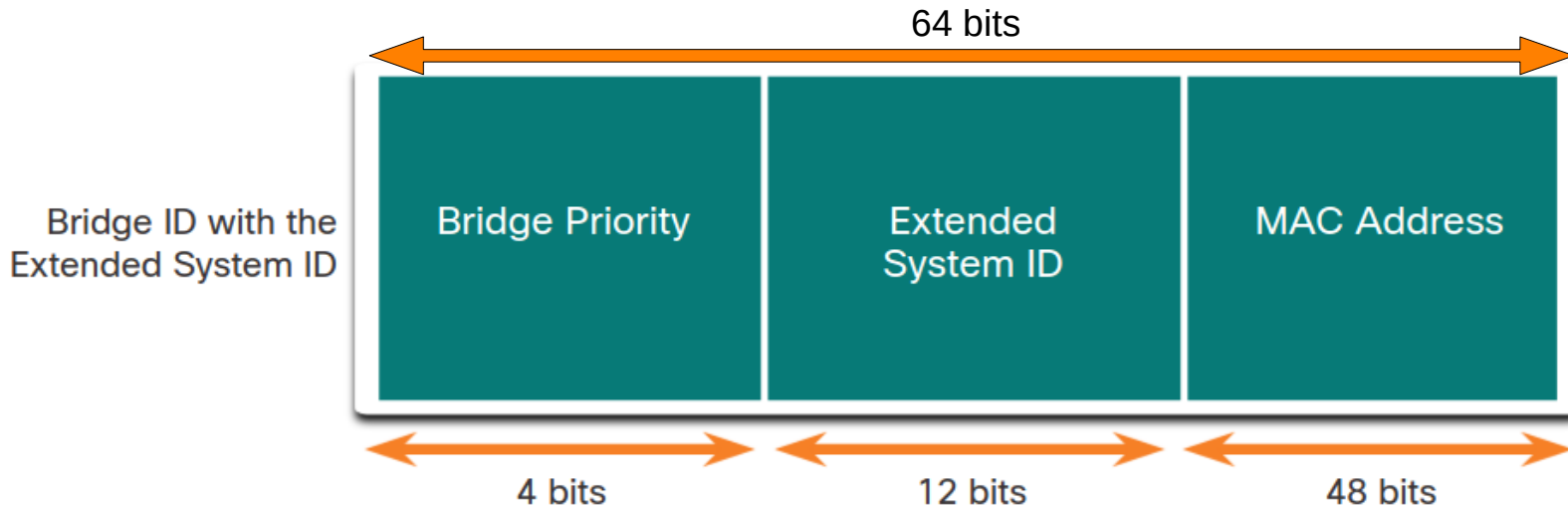
Beachte auch hier: Link-Segmente werden immer von einer Seite aus mit einem Port im Status „Forwarding“ versorgt.



# Wie macht das der „Spanning Tree Algorithmus (STA)“ nun?

- Sicherstellen einer „logisch schleifenfreien LAN-Topologie“.
- Schritte im Überblick:
  - Wahl einer Root Bridge
  - Festlegen der Root Ports
  - Auswahl der Designated Ports
  - Festlegung der Alternate (Blocked) Ports
- STA / STP verwenden zur Kommunikation „Bridge Protocol Data Units (BPDUs)“ die u.a. die Bridge-Identity enthalten
- Zentrales Kriterium → die Bridge-Identity (Bridge-ID)
  - Bridge = alte Begrifflichkeit. Funktion war ursprünglich in Software realisiert

# Die Bridge-Identity – zentraler Entscheidungswert des STA



**Bridge Priority:** Bei Wertangabe wird, im Gegensatz zur Grafik, die 12 Bit ExtSysID mit eingerechnet → Defaultwert 32768 nur in 4096-er Schritten veränderbar.

**Extended System ID:** Entspricht VLAN-ID → Pro VLAN andere Root-Bridge möglich. Bridge-Priority für VLAN 1 als Default-Wert somit 32769

**MAC-Adresse:** Ja, managebare und STP-Switches haben MAC-Adressen

**Bridge-ID:** Kombination der drei Werte.

**ACHTUNG:** Kleinere ID-Werte sind hinsichtlich STA besser! Ohne Konfiguration ist damit die „zufällige“ MAC-Adresse für den STA relevant. Besser: **Priority** konfigurieren!

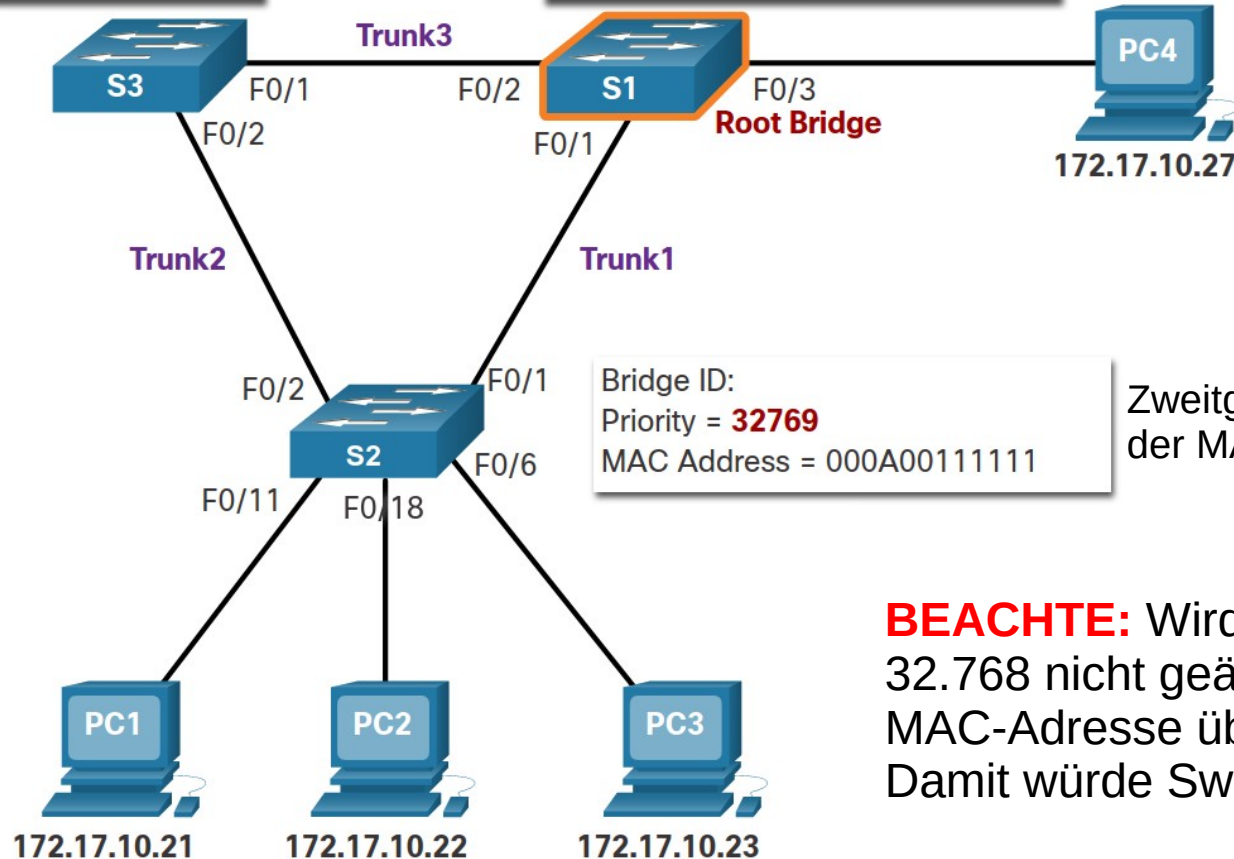
- Nach dem Booten sendet ein Switch alle 2 Sekunden BPDU-Frames in die angeschlossenen Segmente
- Inhalt eigene Bridge-ID und Bridge-ID der Root-Bridge (wobei zu Beginn erst einmal die eigene ID verwendet wird)
- Gleichzeitig werden BPDUs von den Nachbarn empfangen
- Enthält Nachbar-BPDU eine kleinere Root-Bridge-ID → neue Root-Bridge ID übernehmen und außerdem die Kosten zur Root-Bridge aktualisieren
- Endzustand: Alle Switches in der Broadcast-Domäne kennen einheitlich die BID der Root-Bridge. Daten werden bislang noch **nicht** weitergeleitet!

# Bestimmung der Root-Bridge => Kleinste BID!

Bridge ID:  
Priority = **32769**  
MAC Address = 000A00222222

Bridge ID:  
Priority = **24577**  
MAC Address = 000A00333333

Durch die „kleinere“ Priorität  
hier die kleinste BID



**BEACHT:** Wird die Default-Priorität von 32.768 nicht geändert, entscheidet die MAC-Adresse über die kleinste BID. Damit würde Switch ... die Root-Bridge!

# "Kosten" bei Spanning-Tree aus Link-Speed abgeleitet

- Durch IEEE in 802.1d und 802.1w (Rapid-STP) unterschiedlich festgelegt
- Defaultwerte nach 802.1D sind IOS bekannt. "Kosten" aber auch manuell änderbar!

Path cost for different port speed and STP variation

Data rate (Link bandwidth)	STP cost (802.1D-1998)	RSTP cost <sup>[5]:154</sup> (802.1W-2004 default value)
4 Mbit/s	250	5,000,000
10 Mbit/s	100	2,000,000
16 Mbit/s	62	1,250,000
100 Mbit/s	19	200,000
1 Gbit/s	4	20,000
2 Gbit/s	3	10,000
10 Gbit/s	2	2,000
100 Gbit/s	N/A	200
1 Tbit/s	N/A	20

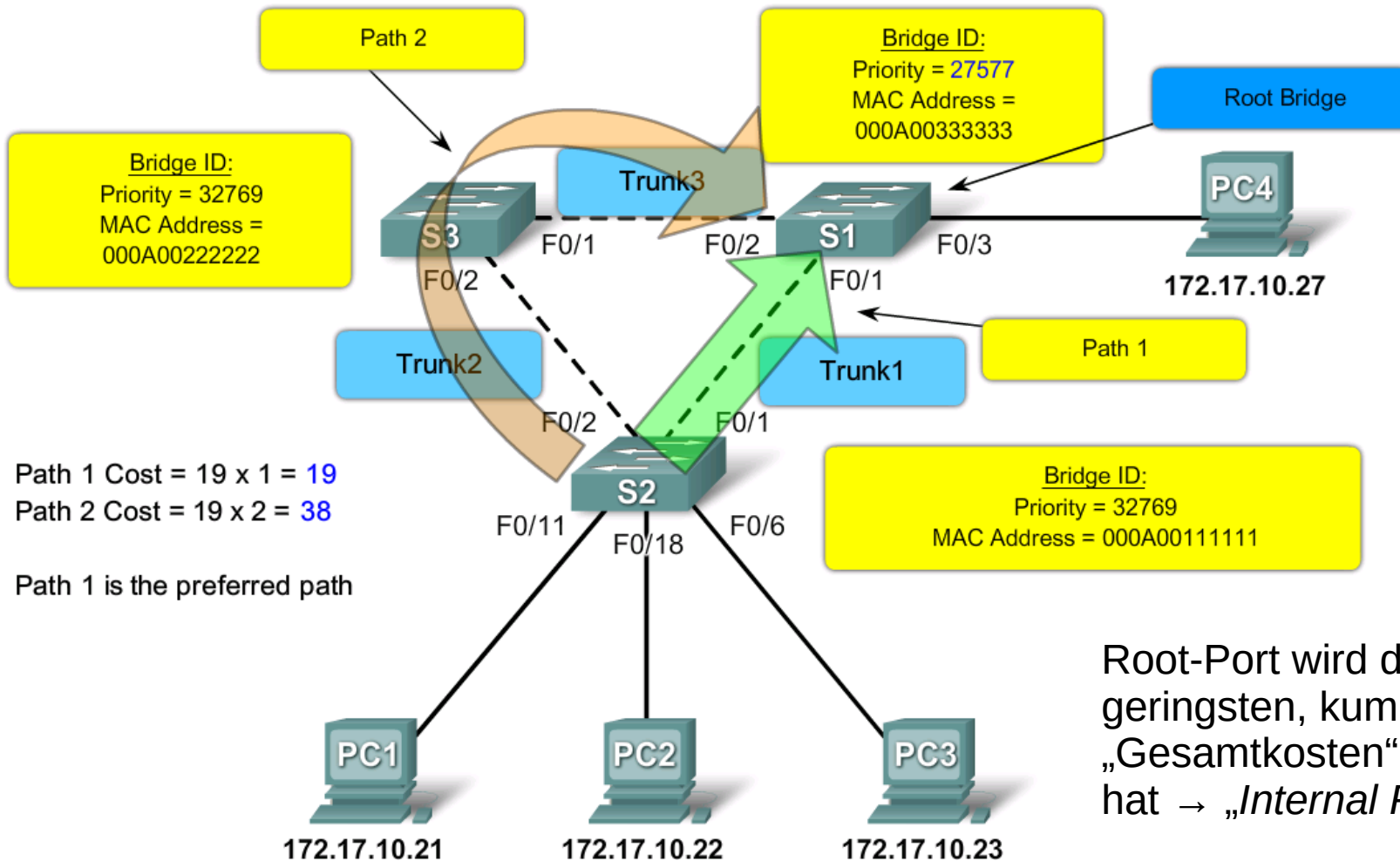
```
S2#configure terminal
S2 (config)#interface Fa0/18
S2 (config-if)#spanning-tree cost 25
S2 (config-if)#no spanning-tree cost
S2 (config-if)#end
```

Wertebereich  
1 - 200.000.000

Defaulteinst. wieder  
aktivieren



# Root-Ports festlegen – STA-Ergebnis im vorliegenden Beispiel

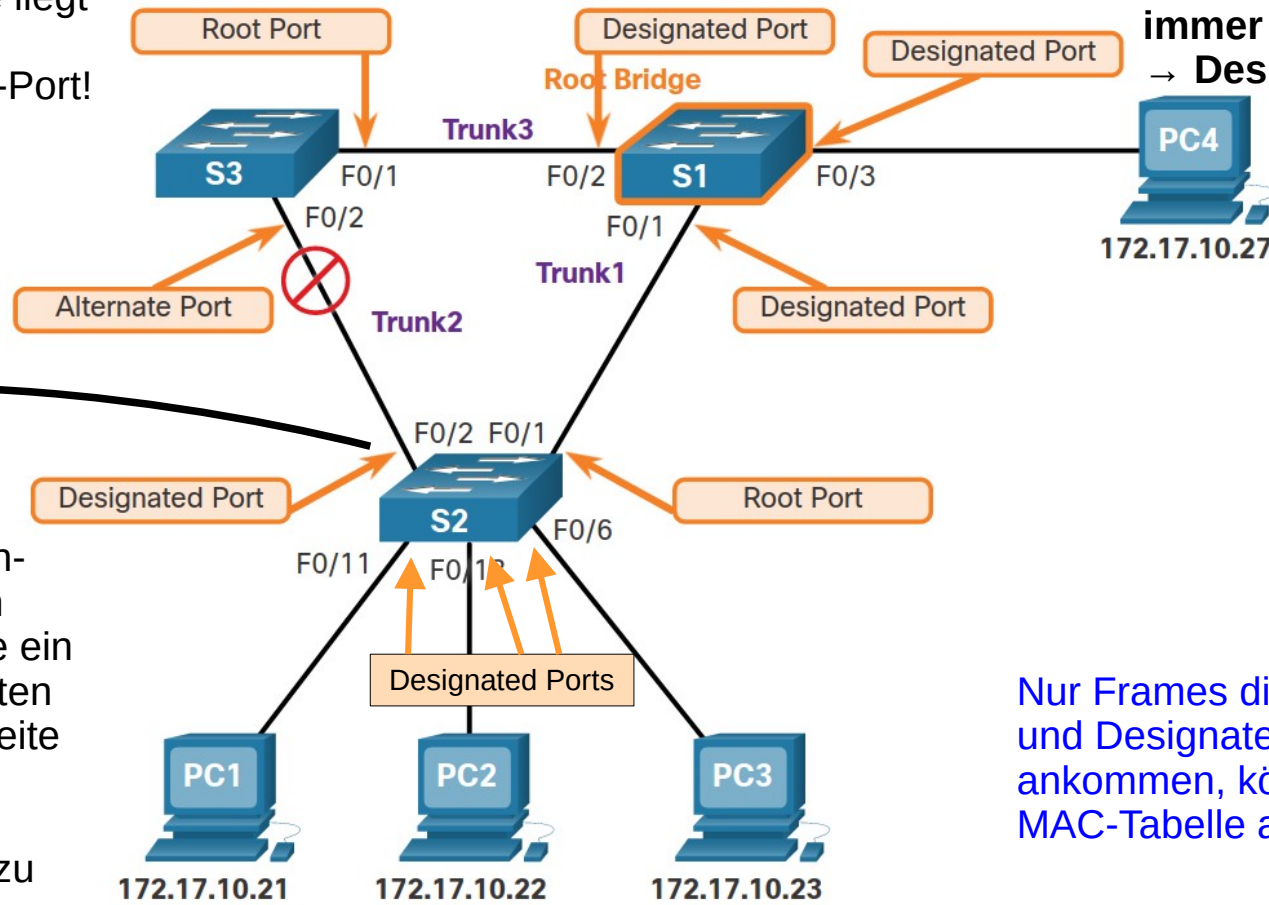


Root-Port wird der Port der die geringsten, kumulierten „Gesamtkosten“ zur Root-Bridge hat → „*Internal Root Path Cost*“

# Spanning Tree Protocol (STP) Ergebnis & Port-Rollen

Alle anderen Switches aktivieren Port der am "nächsten" zur Root-Bridge liegt  
→ **Root-Ports**  
Pro Switch genau ein Root-Port!

**Root-Bridge** hat die kleinste BID. Aktiviert **immer** alle ihre Ports  
→ **Designated Ports!**



Verbindungen zw. zwei Non-Root-Bridges müssen auch versorgt werden! Es könnte ein Geräte ohne STP-Fähigkeiten da sein! Deshalb ist eine Seite dafür zuständig mit Rolle **"Designated Port"** (Status Forwarding) das Segment zu versorgen. Die andere Seite hat die Rolle → **"Alternate Port"** (Status Blocking), hört u. Verarbeitet aber nach wie vor BPDUs

Nur Frames die via Root- und Designated-Ports ankommen, können die MAC-Tabelle aktualisieren.

# Welcher „Tie break“ löst „equal cost“-Pfade zur Root-Bridge?

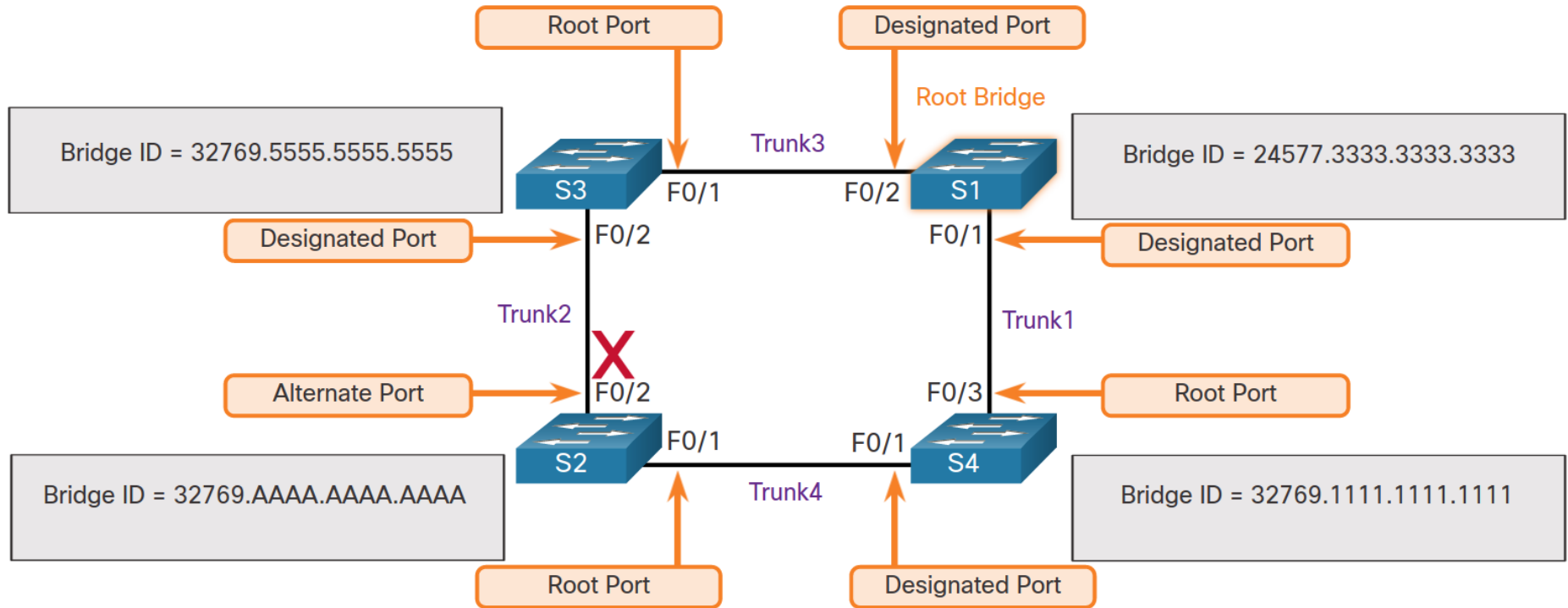
- Fall ist: Ein Switch hat bei mehreren Ports die gleichen Kosten zur Root-Bridge! Was nun?
- Hier sind erneut die von der / den Partnern gesendeten Werte relevant – nicht die eigenen!

Der Fall-Back-Mechanismus sieht wie folgt aus:

- Port dessen Gegenseite die kleinste Bridge-ID (BID) hat
- Bei Gleichheit – Port mit kleinster Port Priority der Gegenseite
- Bei Gleichheit – Port mit kleinster Port ID der Gegenseite

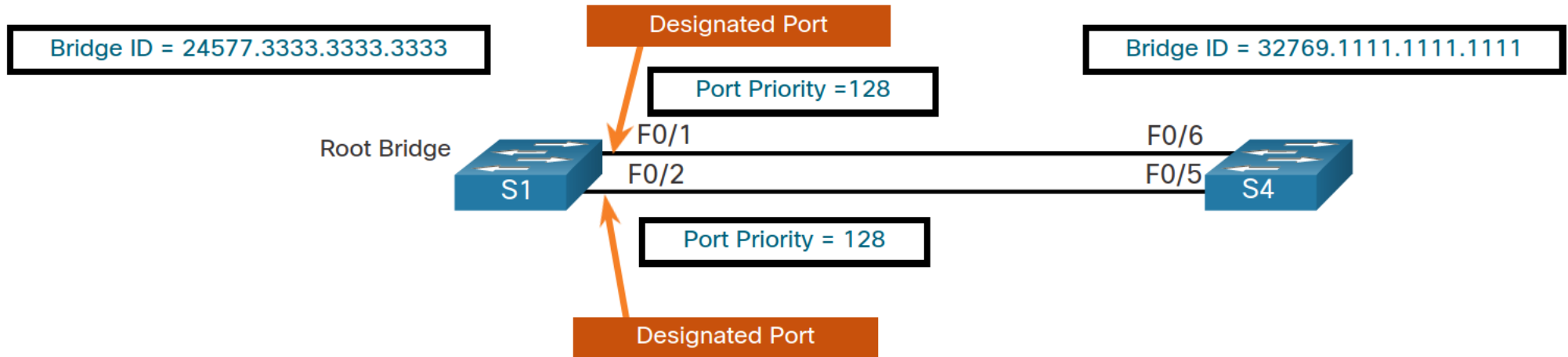
Das schauen wir uns an Fallbeispielen an ...

# „Equal cost Path“ – Tie Break #1



- Switch S2 hat das Entscheidungsproblem! Welcher Port soll Root Port werden?
- Relevant sind die BID-Werte der Switches S3 und S4 (die sogenannten *Sender*)
- S4 hat kleinere BID → F0/1 wird Root Port

# „Equal cost Path“ – Tie Break #2



- Switch S4 hat das Entscheidungsproblem! Welcher Port soll Root Port werden?
- Relevant sind die Port-Priority-Werte von Switch S1 (dem *Sender*)
  - Bei Gleichheit geht es zu Tie Break #3 (nächste Folie)
  - Falls ungleich, z.B. mittels ...

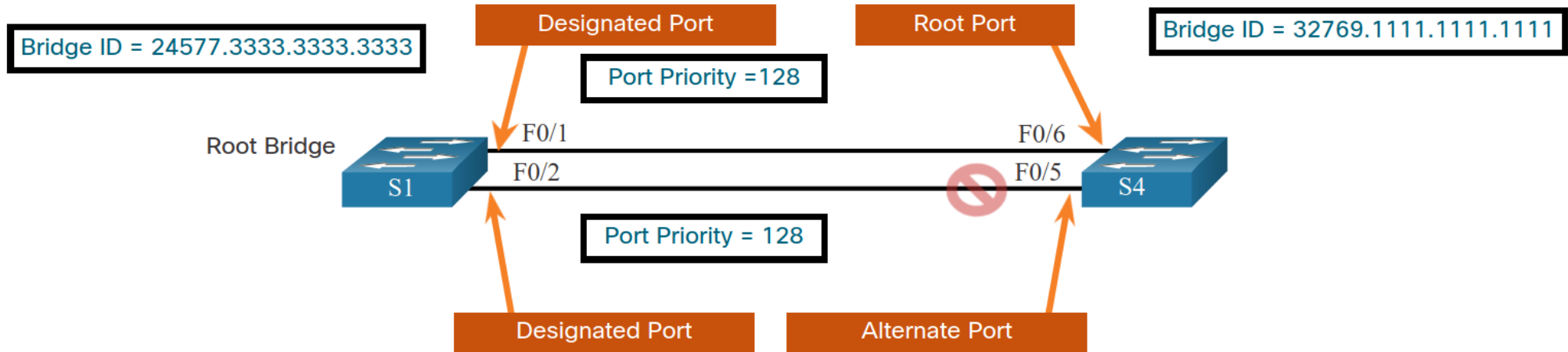
```
S1(config)#interface Fa0/2
```

```
S1(config-if)# spanning-tree port-priority 112
```

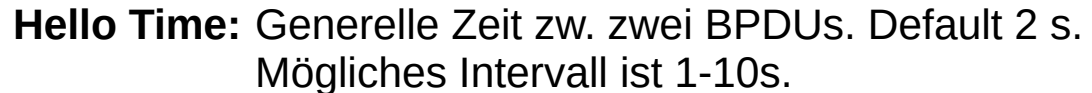
wird hier bei S4 der Port Fa0/5 zum Root Port (kleinere Priority)



# „Equal cost Path“ – Tie Break #3



- Switch S4 hat das Entscheidungsproblem! Welcher Port soll Root Port werden?
- Bei gleichen Port-Priority-Werten von Switch S1 entscheidet nun dessen Port Identities:
  - Switch S1, Port ID F0/1 ist kleiner als S1-F0/2  
somit wird hier bei S4 der Port Fa0/6 zum Root Port (kleinere Port-ID!)
  - Port-ID ist im realen STP eine Zahl



# Zu unterscheiden: „Port Rolle“ und „Port Zustand“

## Gelernte Port Roles wären:

- Root Port → im Regelbetrieb im Forwarding State
- Designated Port → im Regelbetrieb im Forwarding State
- Alternate Port → Blocking State

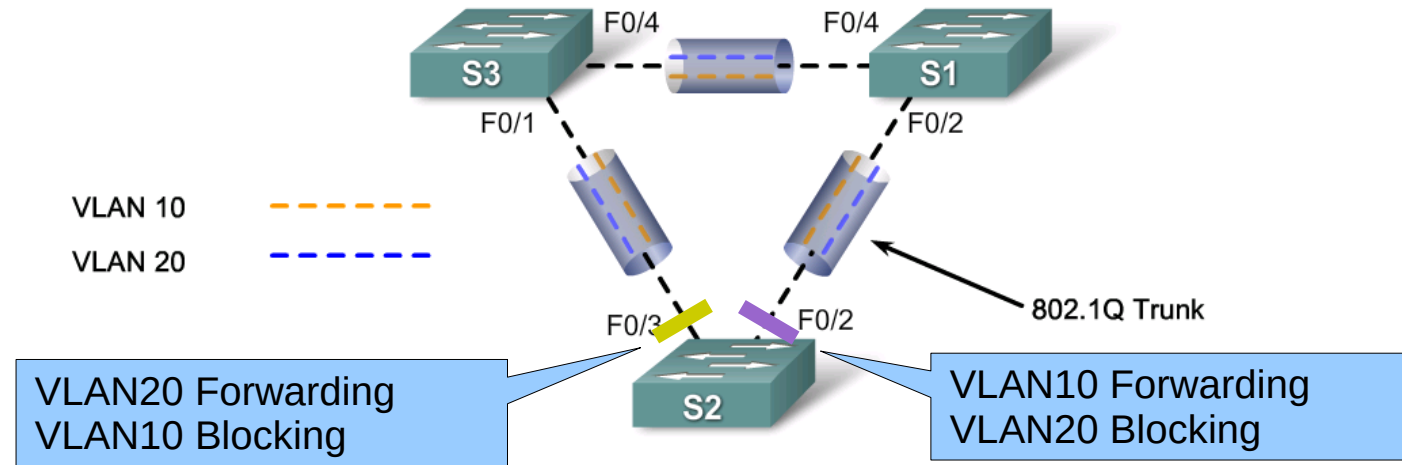
## Port States können sein:

Port State	BPDU	MAC Address Table	Forwarding Data Frames
Blocking	Receive only	No update	No
Listening	Receive and send	No update	No
Learning	Receive and send	Updating table	No
Forwarding	Receive and send	Updating table	Yes
Disabled	None sent or received	No update	No

# Per VLAN Spanning-Tree (PVST)

Primary root bridge for VLAN 20  
Secondary root bridge for VLAN 10

Primary root bridge for VLAN 10  
Secondary root bridge for VLAN 20



- Pro VLAN ein eigener STP-Prozess
- Durch geschickte Platzierung der Root-Bridge für die einzelnen VLANs Lastverteilung auf den Trunks
- Nachteil: Erhöhter Bandbreitenbedarf durch die BPDUs die jetzt "per VLAN" unterwegs sind.

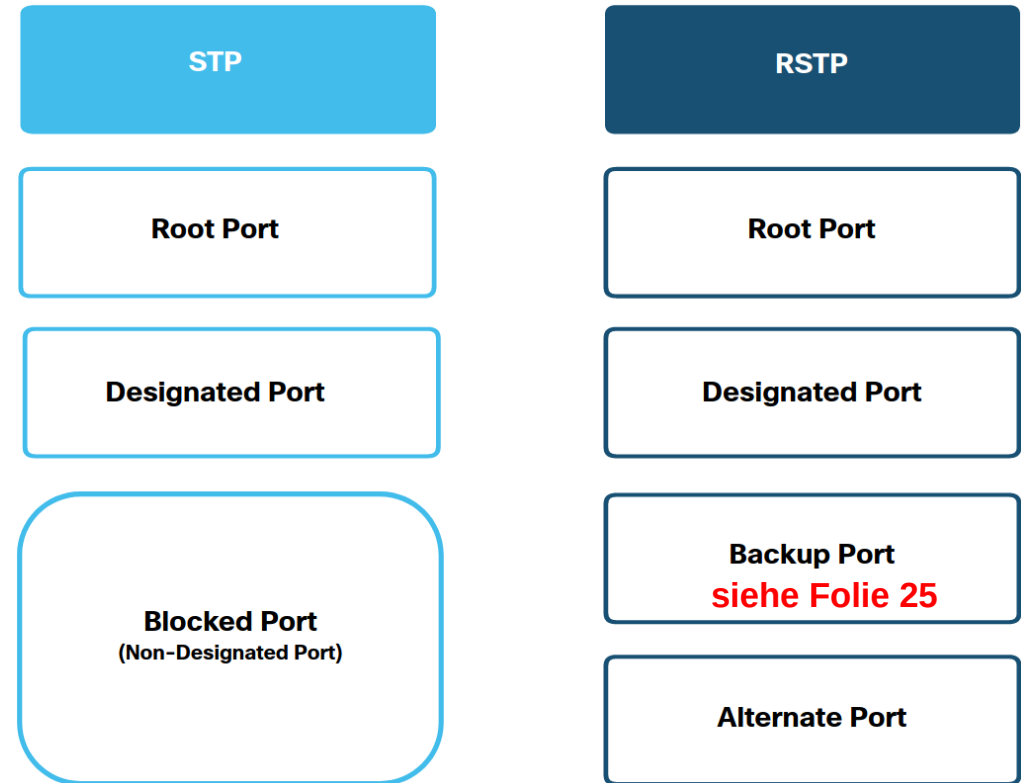
- Am Anfang war ... 802.1D – **STP**
- Die Weiterentwicklung ist **Rapid STP (RSTP)** IEEE 802.1w
- Standard IEEE-802-1D-2004 übernimmt RSTP als STP-Default
- Dann **Multiple Spanning Tree Protocol (MSTP)** von IEEE, als Erweiterung von STP & RSTP in IEEE 802.1Q-2005 (per VLAN)
- Und Cisco hat jeweils eine verbesserte Version davon, mit Features wie: PortFast, UplinkFast, BackboneFast, BPDU guard, BPDU filter, root guard, loop guard
  - STP → PVSTP+  
(Default auf den Switches mit IOS 15)
  - RSTP / IEEE-802-1D-2004 → Rapid PVSTP+
  - MSTP → MST



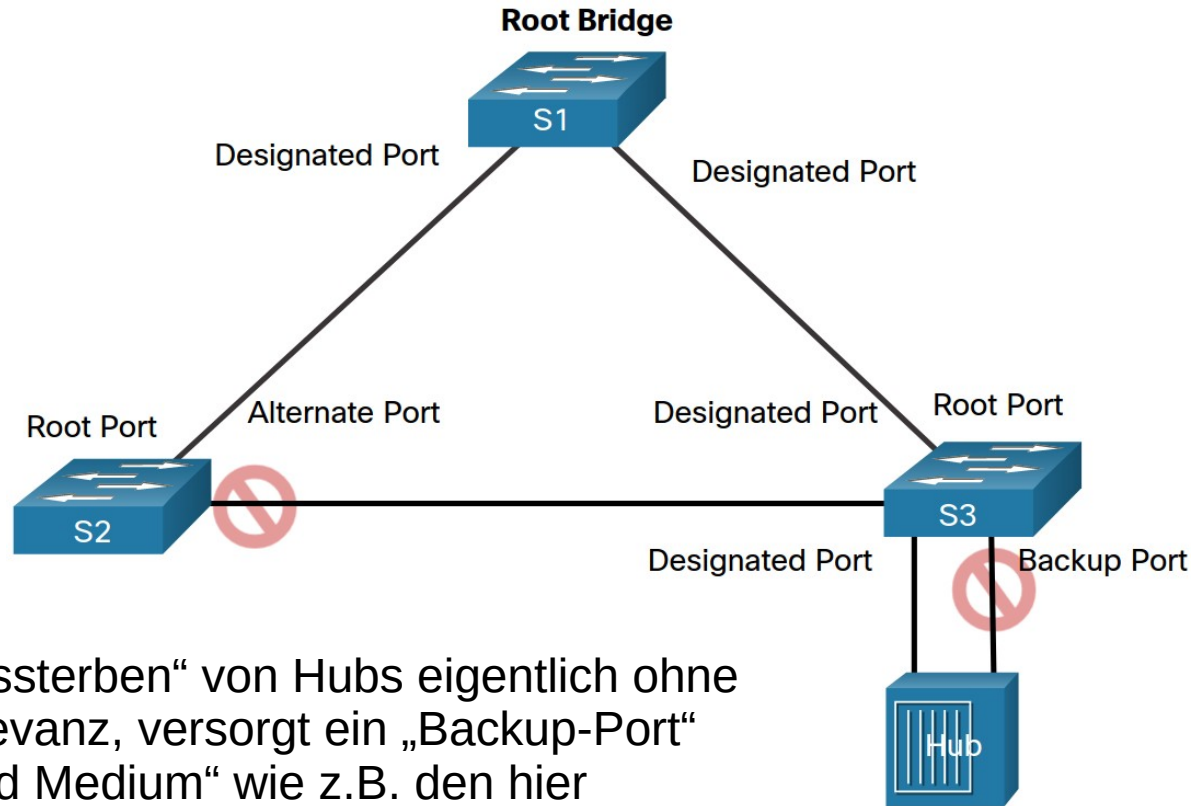
## Port Zustände / States



## Port Rollen / Roles

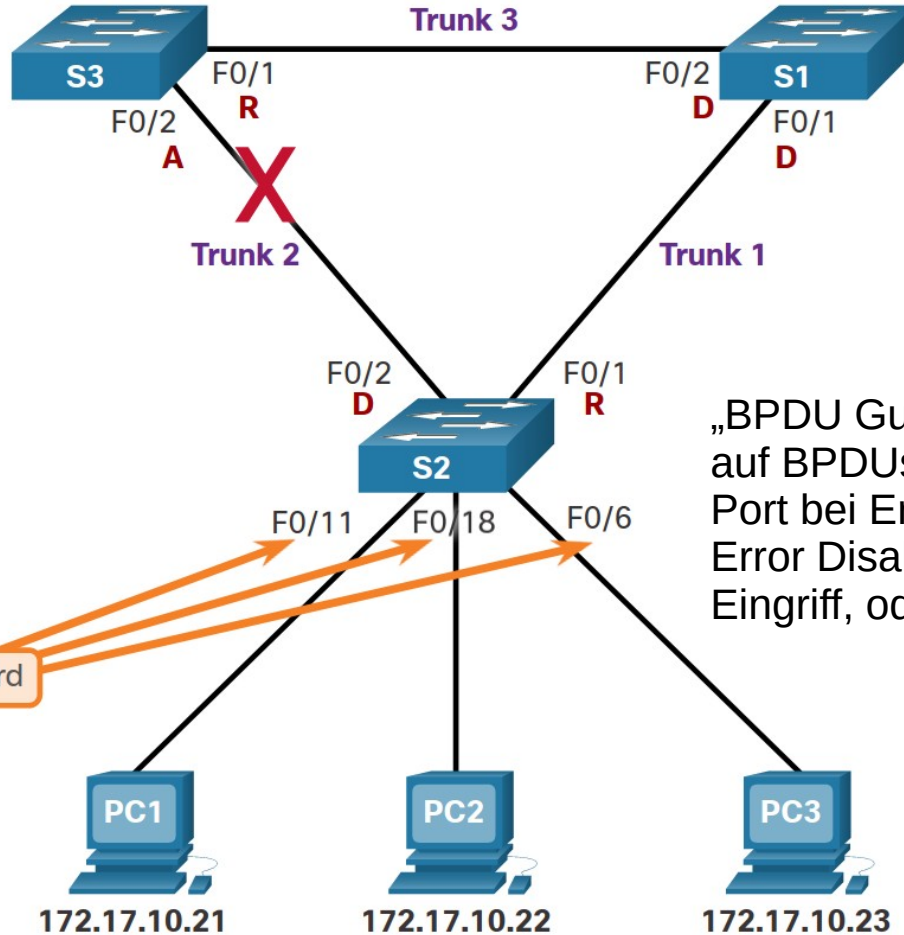


Curriculum mischt Begriffe leider etwas. Alternate Port schon in Erklärung von STP, obwohl RSTP



Durch „Aussterben“ von Hubs eigentlich ohne große Relevanz, versorgt ein „Backup-Port“ ein „Shared Medium“ wie z.B. den hier gezeigten Hub.

# PortFast und BPDU Guard, z.B. bei STP für DHCPv4



Bei Client-Start dauert es mit STP  
min. 30 Sekunden bis Link für  
Daten operativ ist!

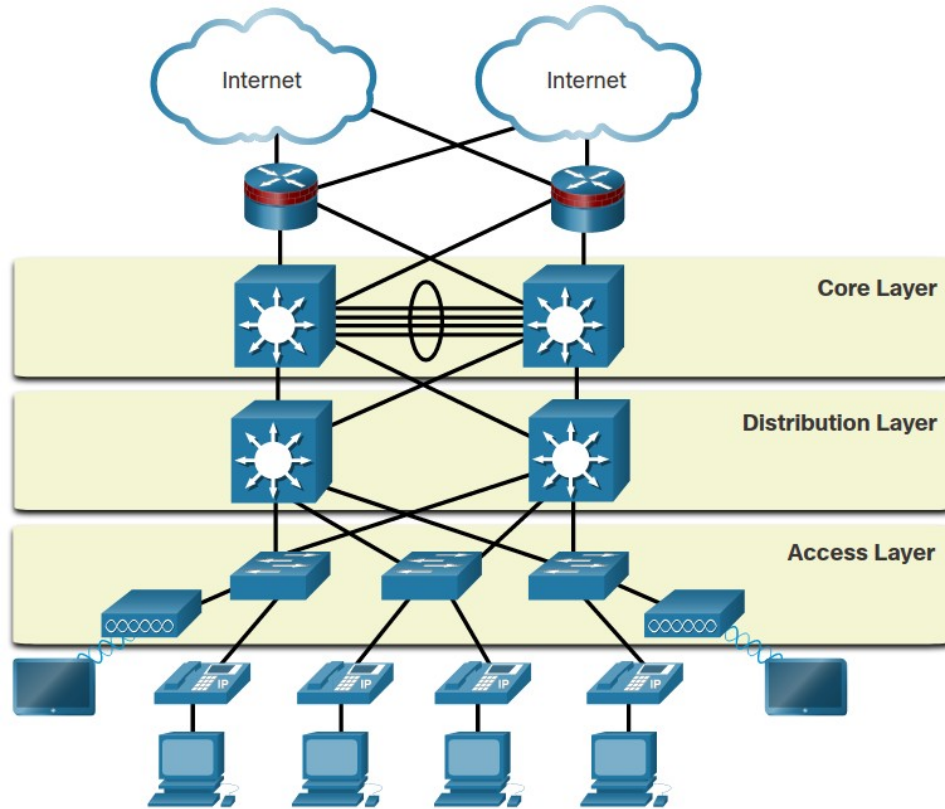
**Für DHCP bei IPv4 ggf. zu spät!**  
Bei IPv6 weniger kritisch. Warum?

„BPDU Guard“, überwacht Port  
auf BPDUs, und deaktiviert ggf.  
Port bei Erhalt einer solchen!  
Error Disabled – erfordert  
Eingriff, oder Auto-Recovery.

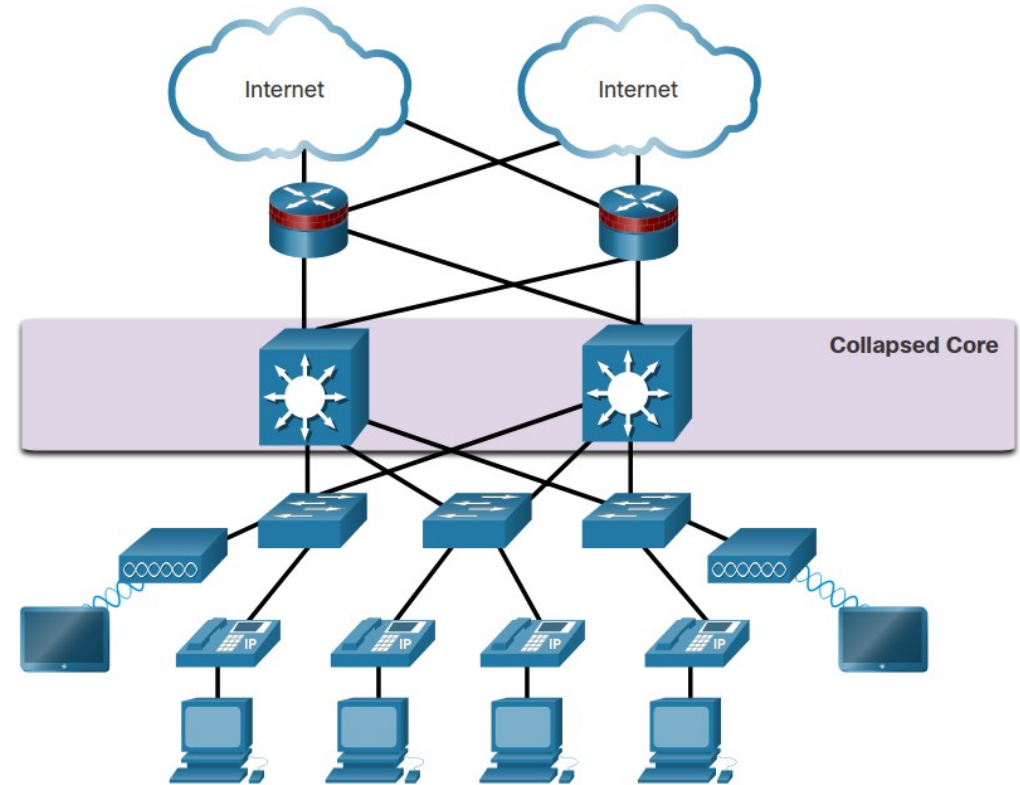
PortFast and BPDU Guard

„Portfast“ aktiviert Ports, unter  
Umgehung des STP sofort!  
Gefahr einer Loop-Bildung! Nur  
auf Access Ports aktivieren!

# Spanning Tree im hierarchischen Netz Design – geht das noch?

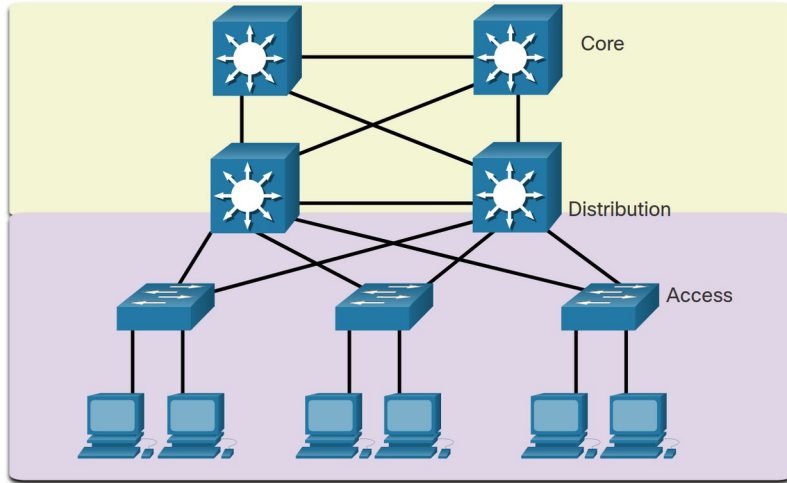


Hierarchische Netze sind heute Standardblöcke im Netzdesign – umfassen ggf. Hunderte Switches und Tausende VLANs.



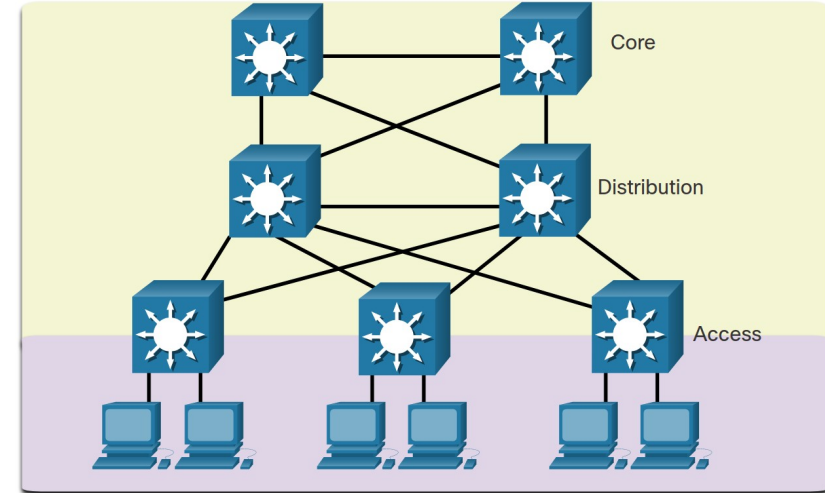
Sowohl im 3-Layer Design, als auch im 2-Layer „collapsed Core“, geht STP grundsätzlich, stößt aber an Grenzen. Deshalb hier ggf. andere Protokolle, mehr Routing statt Switching

# Layer 3 Routing benötigt kein STP



Layer 3 Routing

Layer 2 Switching



Layer 3 Routing

Layer 2 Switching

Das „hierarchische Block-Modell“ verwendet ab „Distribution-Layer“ IP-basierendes Routing.

Damit sind dort redundante Wege auf Basis von Routing möglich.

Dynamisches Routing ersetzt STP

Genereller Lösungsansatz: Da Layer 3 Routing explizit kein Problem mit redundanten Pfaden / Schleifen hat → Verschiebung des Layer 3 Switchings bis in den Access Layer!

Neben STP auch andere Techniken wie M(C)LAG, SPB oder TRILL im Einsatz (kein CCNA-Thema mehr)



## ▪ 5.1.9 - Packet Tracer - Investigate STP Loop Prevention

Modul enthält keine Konfiguration, keine Übungen. Außer den Befehlen in obiger PT-Aktivität keine weitere Praxis.

Für Interessierte / für selbstdefinierte Übungen finden Sie in der Präsentation zum Exploration-Course 3, Kapitel 5, die Konfigurationsbefehle zu den hier eingeführten Konzepten

