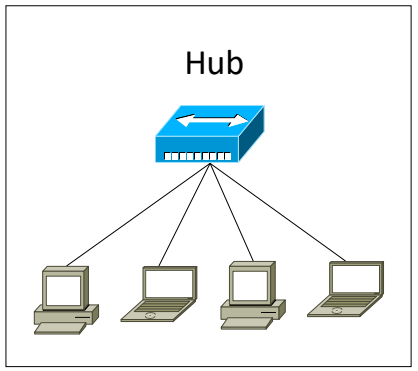
Netztechnik Teil-10

Inhalt

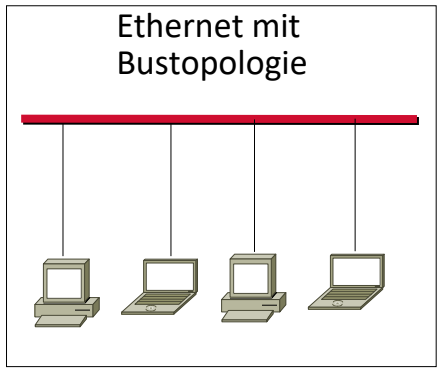
- VLANs
- SDN
- UDP
- TCP
- RIP
- OSPF
- IGMP
- Netzwerk-Management
- Anwendungsprotokolle

VLANs
Teil-1



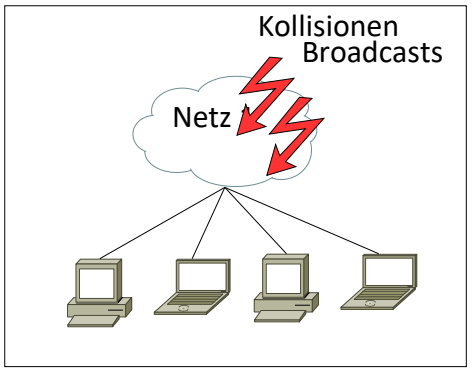
Physikalischer Aufbau

=

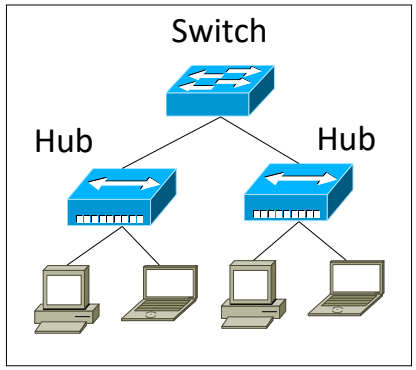


Entspricht einem Shared Media

=

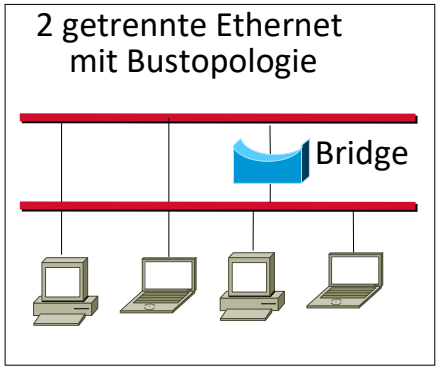


Möglicher logischer Aufbau mit Beeinflussung durch Kollisionen und Broadcasts



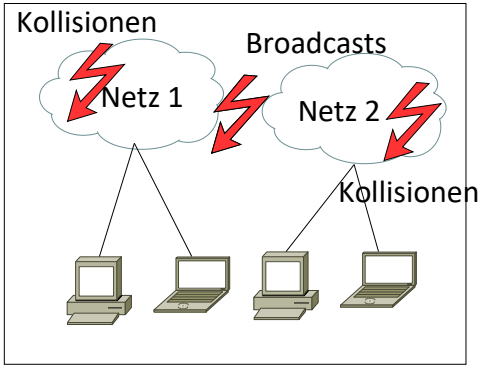
Physikalischer Aufbau mit Microsegmentierung durch Switch

=



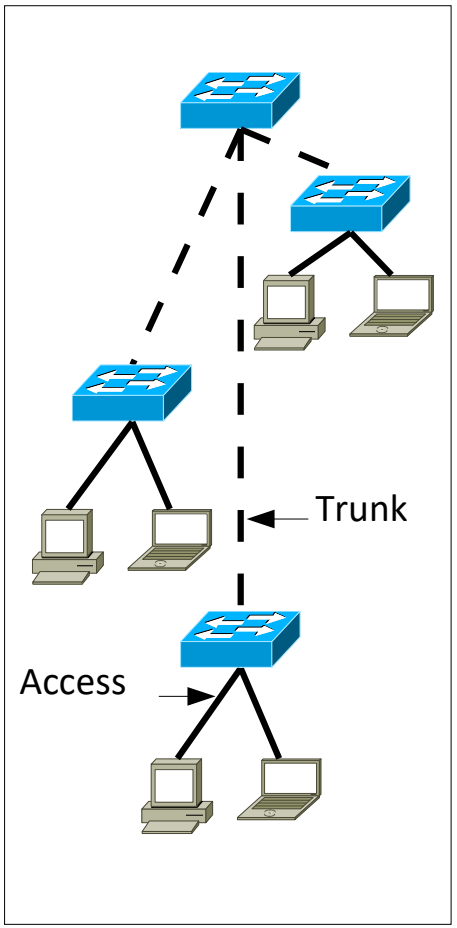
Entspricht zwei getrennten Shared Media

=

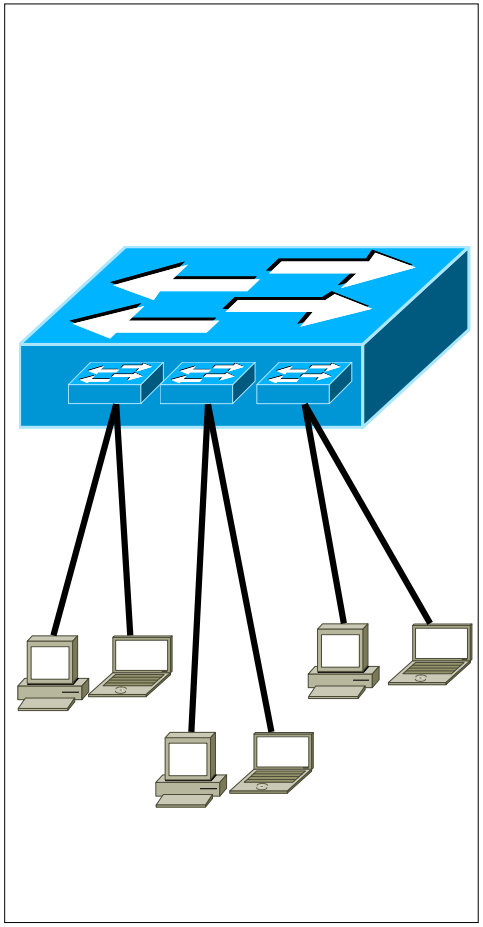


Möglicher logischer Aufbau ohne gegenseitige Beeinflussung durch Kollisionen

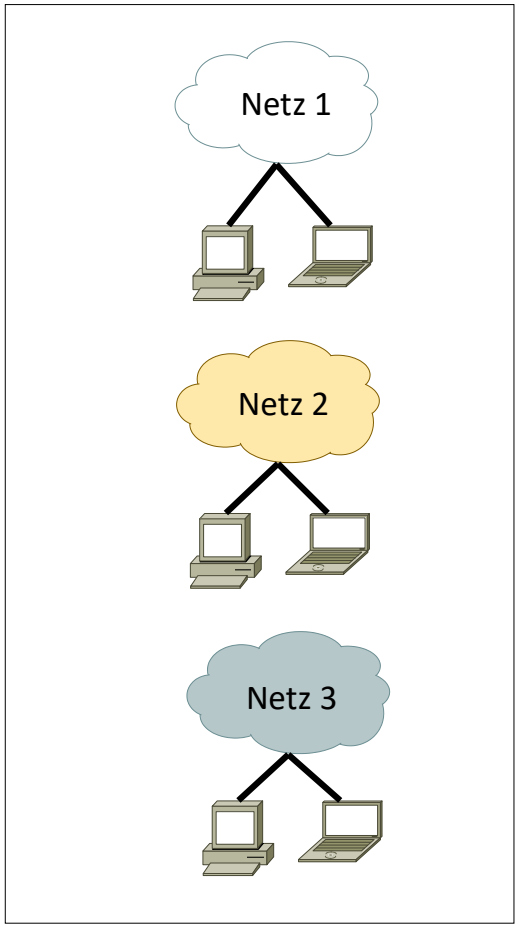
VLANs
Teil-2



=



=



Physikalischer Aufbau von VLANs mit Switches

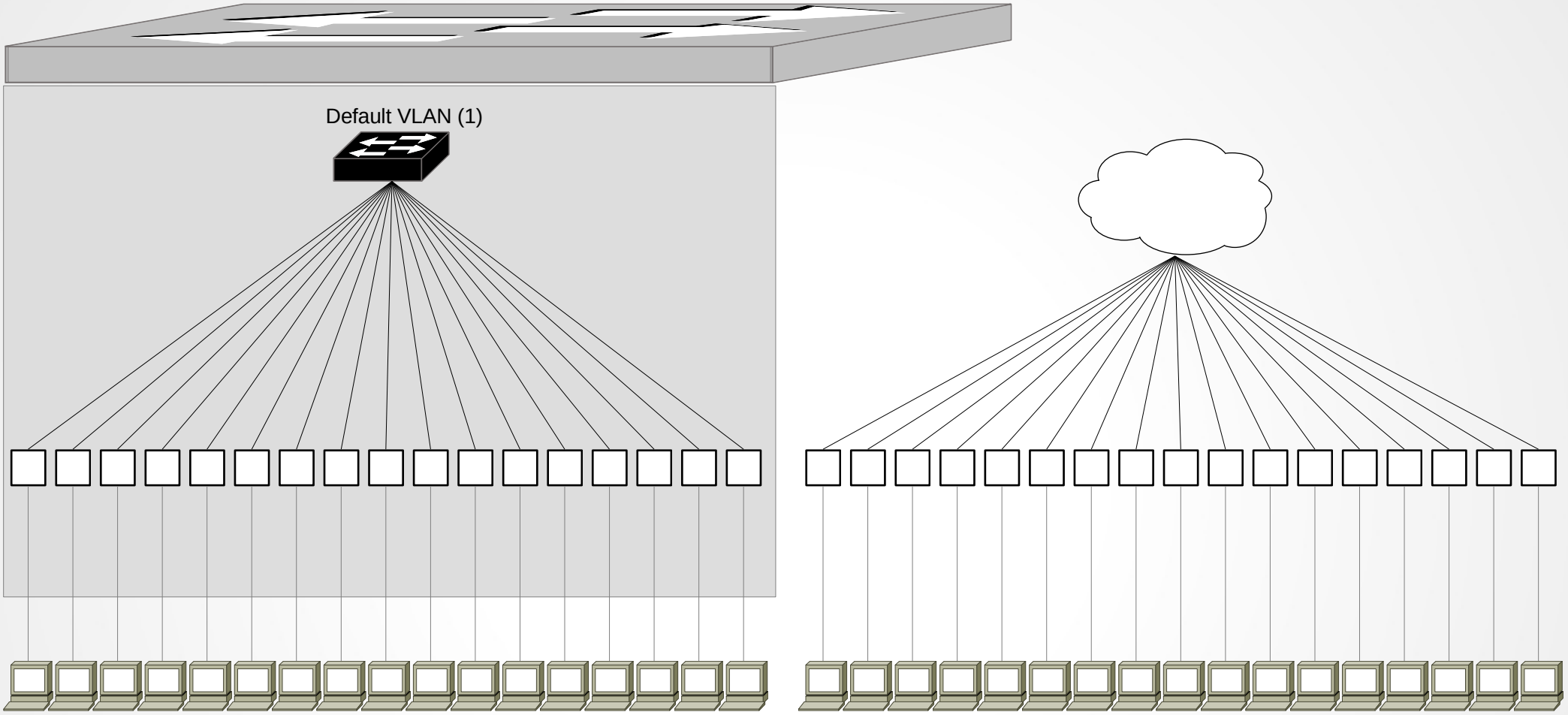
=

Entspricht drei unabhängigen geswitchten Umgebungen auf einer physikalischen Topologie

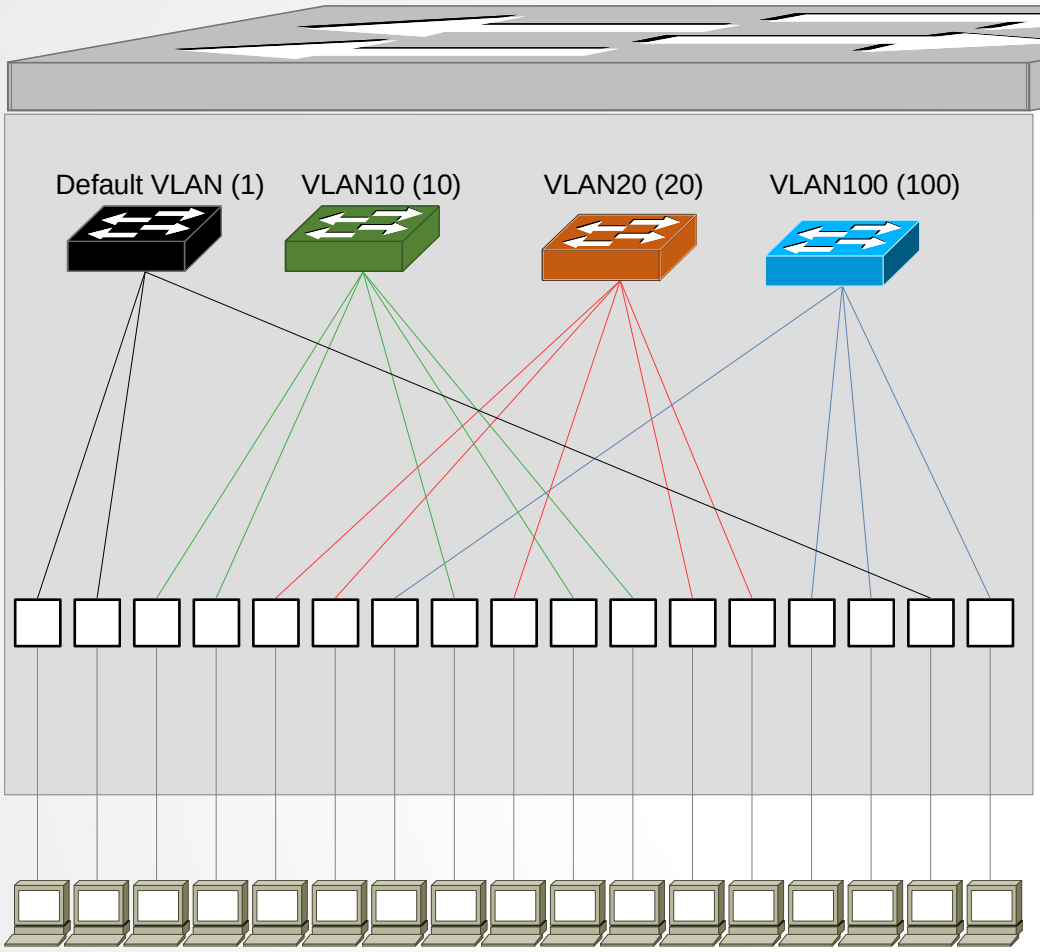
=

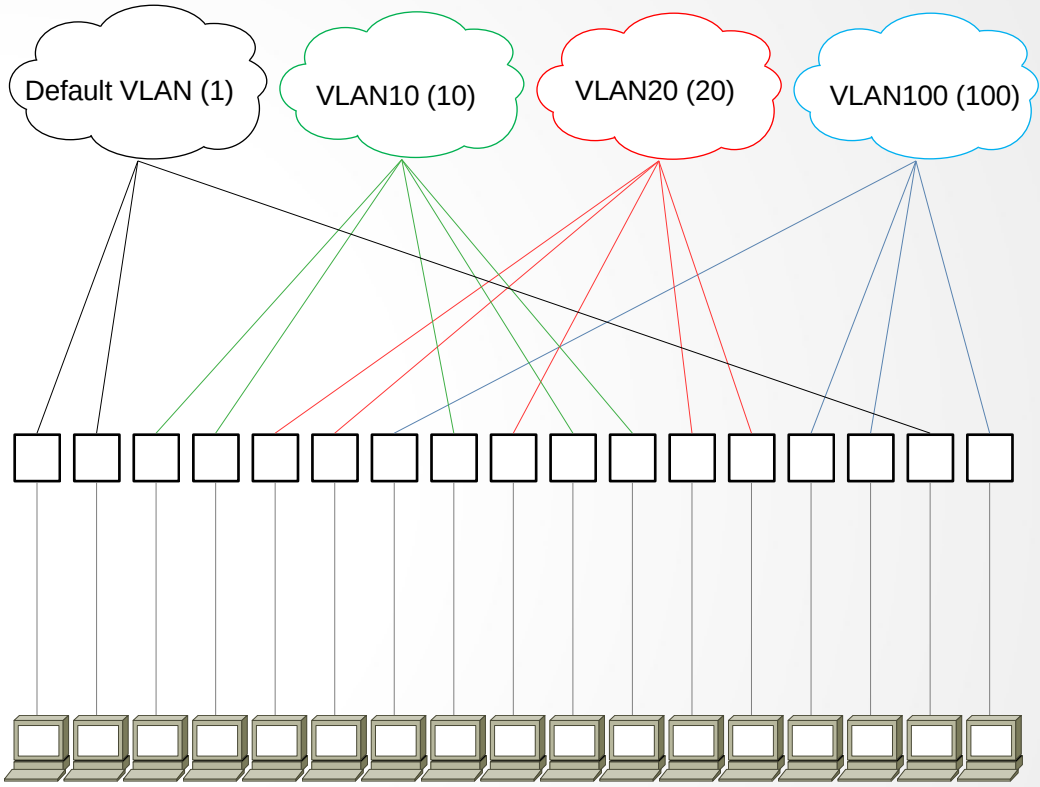
Möglicher logischer Aufbau ohne gegenseitige Beeinflussung durch Kollisionen und Broadcasts

VLANs
Teil-3

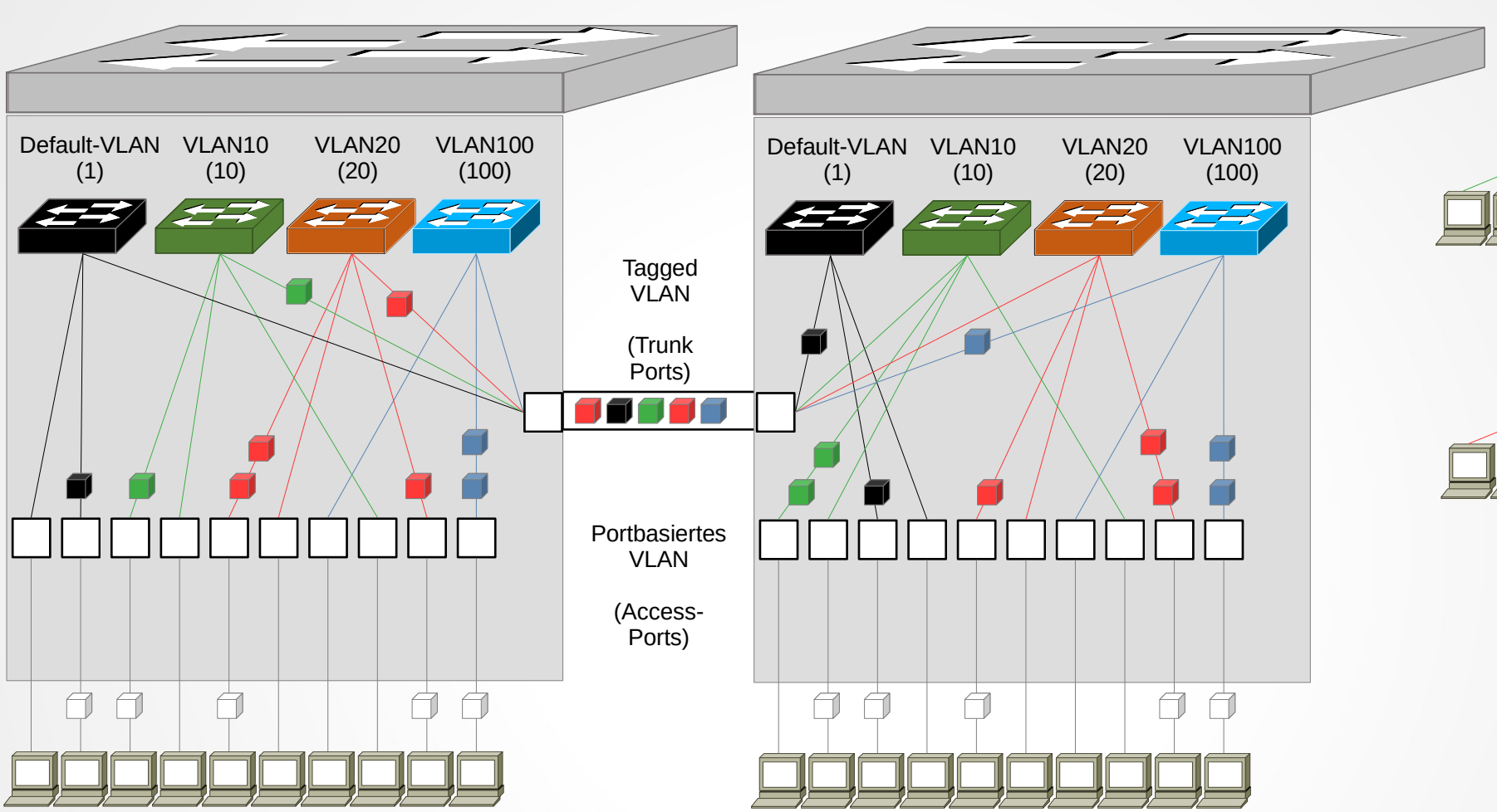


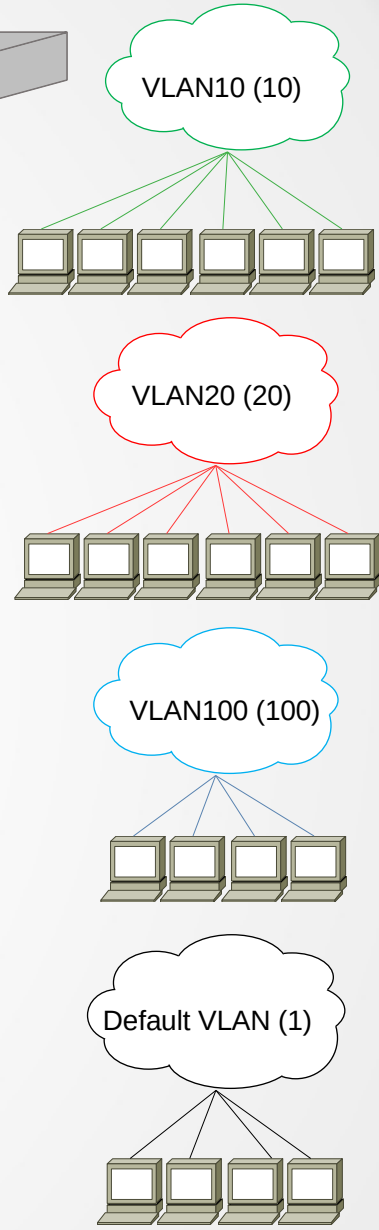
VLANs
Teil-4


Default VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)

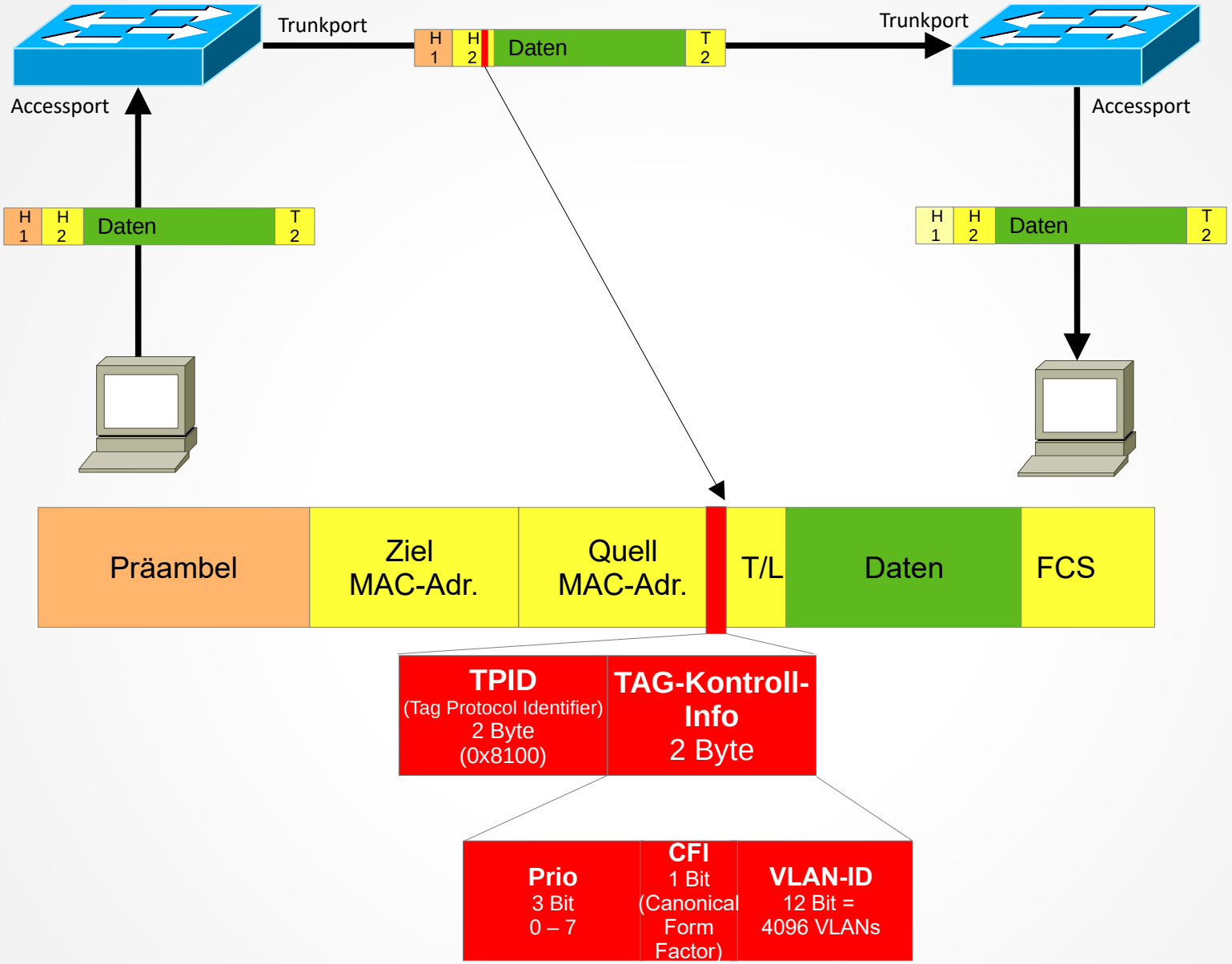

Default VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)

VLANs
Teil-5

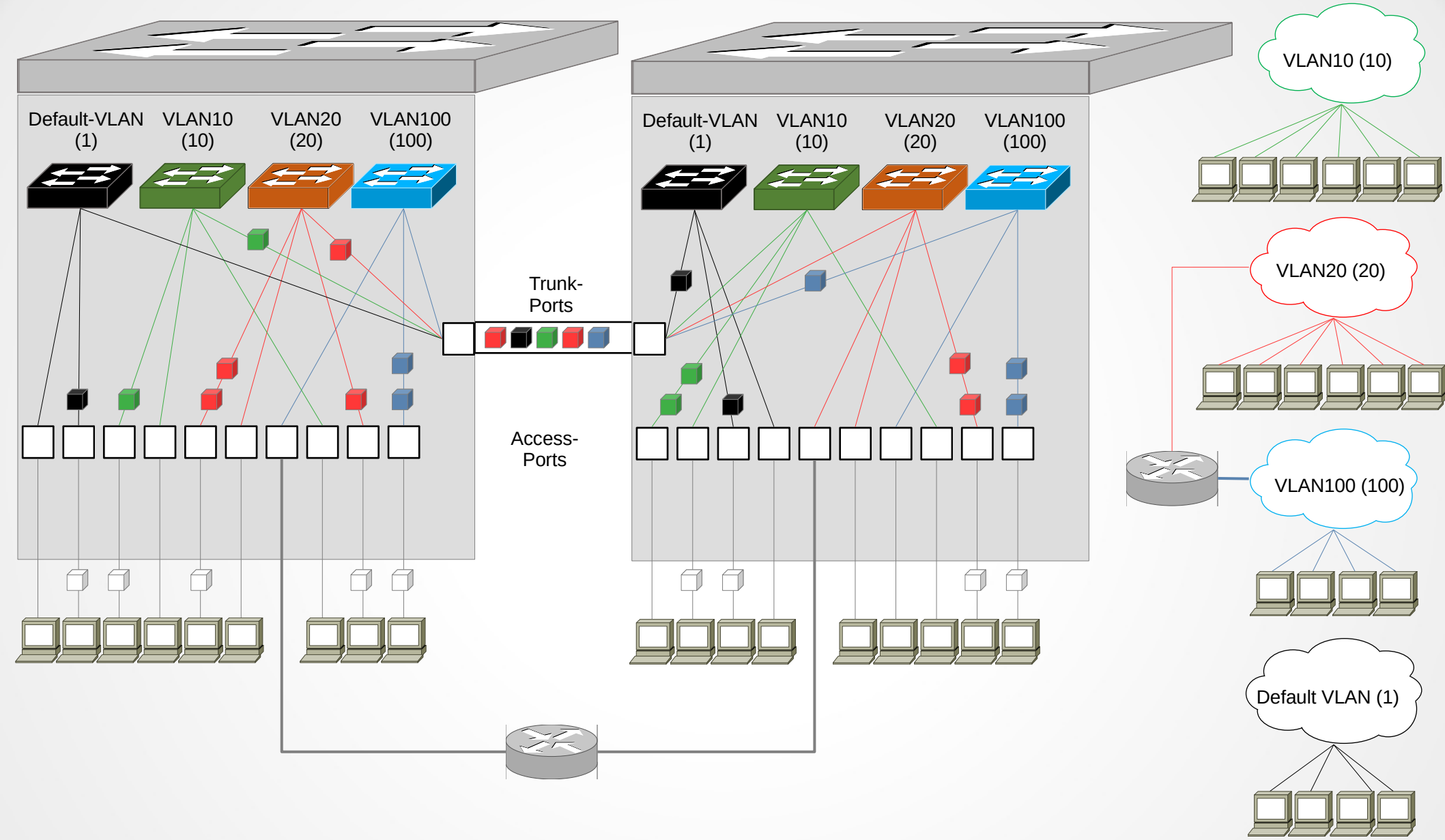

Default-VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
Tagged VLAN
(Trunk Ports)
Portbasiertes VLAN
(Access-Ports)
Default-VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)


VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
Default VLAN (1)

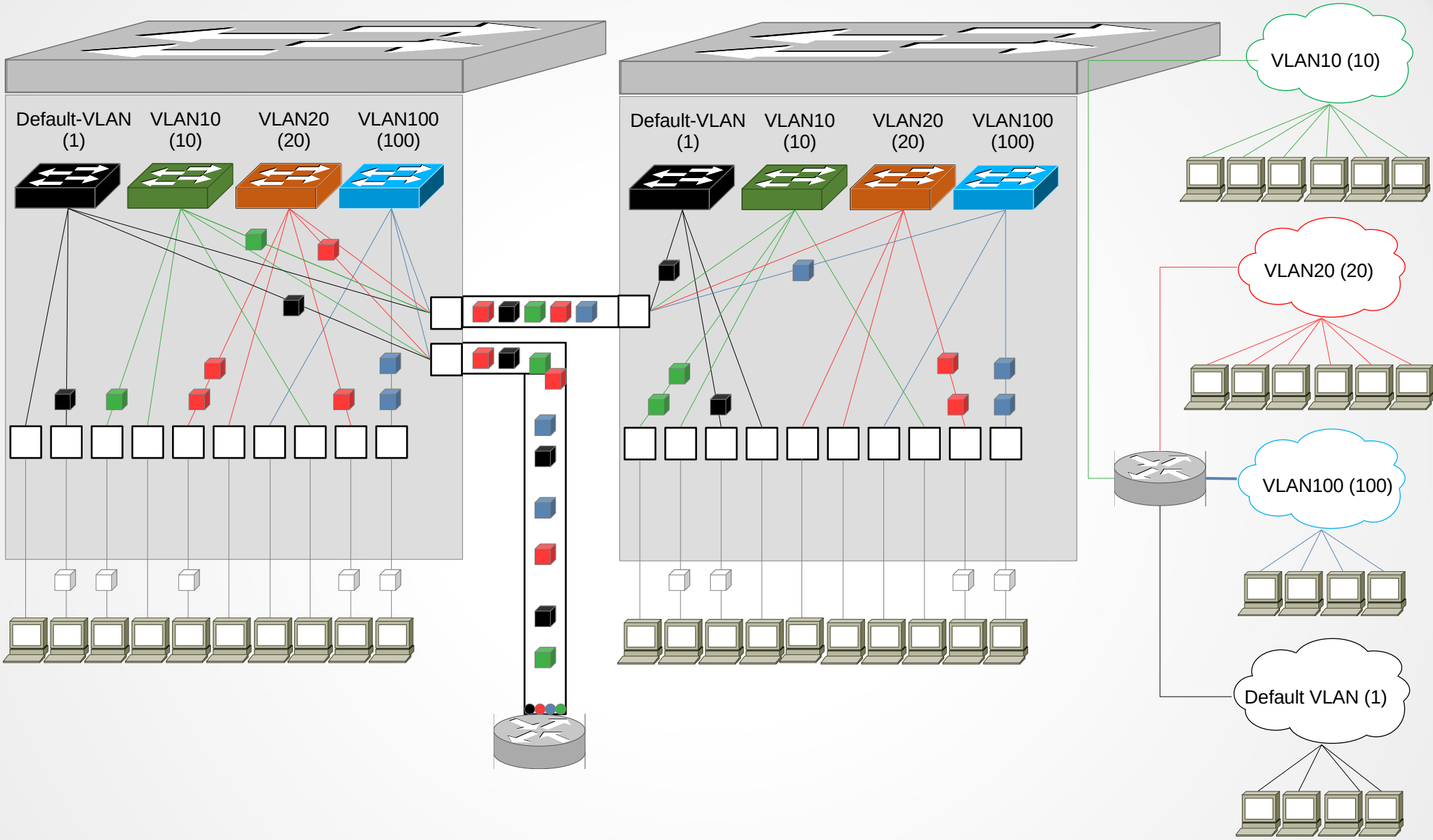
VLANs
Teil-6


Trunkport
Trunkport
Accessport
Accessport
H 1
H 2
Daten
T 2
Präambel
Ziel MAC-Adr.
Quell MAC-Adr.
T/L
Daten
FCS
TPID
(Tag Protocol Identifier)
2 Byte
(0x8100)
TAG-Kontroll-Info
2 Byte
Prio
3 Bit
0 – 7
CFI
1 Bit
(Canonical Form Factor)
VLAN-ID
12 Bit =
4096 VLANs

VLANs
Teil-7


Default-VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
Trunk-Ports
Access-Ports
Default-VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
Default VLAN (1)

VLANs
Teil-8


Default-VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
Default-VLAN (1)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
VLAN10 (10)
VLAN20 (20)
VLAN100 (100)
Default VLAN (1)

VLANs
Teil-9

VLANs können auf verschiedenen Ebenen aufgebaut werden:

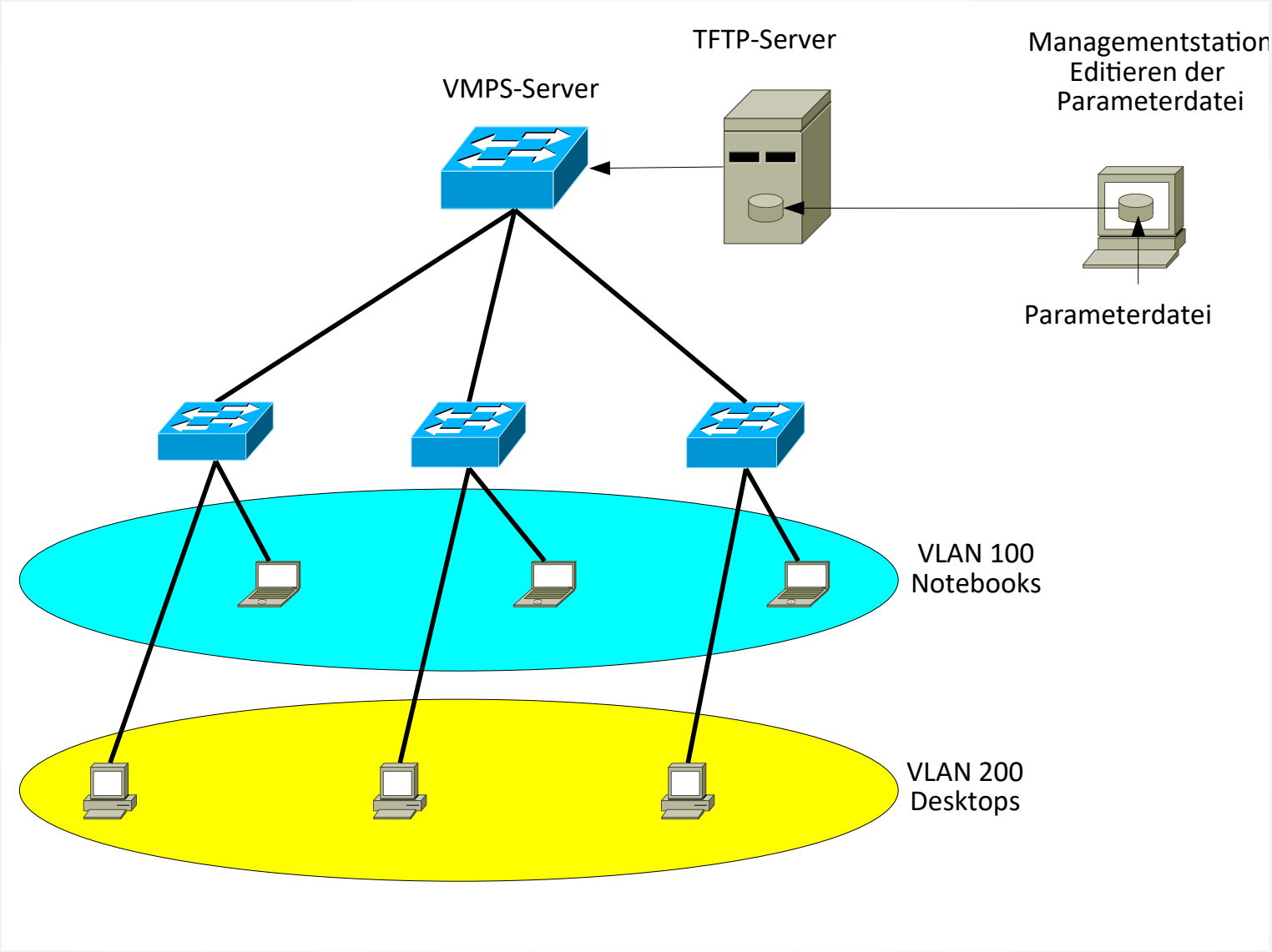
- **Layer 1**
  Switch-Port basierend
  Jedem Switchport wird durch den Administrator ein VLAN zugewiesen.
  Dies kann von einer zentralen Managementstation aus oder direkt am Switch über eine Consol-Verbindung durchgeführt werden.
  Unterlässt der Administrator dies, werden alle Ports in das Default-VLAN (VLAN 1) übernommen.
  Damit können Switches auch ohne eine VLAN-Parametrierung in Betrieb genommen werden.
  Allerdings hat man dann alle Endgeräte in der gleichen Broadcastdomain untergebracht.

- **Layer 2**
  MAC-Adressen basierend
  Alle Rechner werden an zentraler Stelle mit ihrer MAC-Adresse einem VLAN zugeordnet.
  Dazu ist auf einem Server die Zuordnungstabelle allen Switches zur Verfügung zu stellen die sich im Bedarfsfall die Tabelle vom Server beziehen.
  Sobald nun ein Rechner mit einem Switch verbunden wird, kann aufgrund der MAC-Adresse der Switchport in das zugeordnete VLAN übernommen werden.
  Hier ist z. B. Das Cisco-Protokoll VMPS (VLAN Membership Policy Server) angesiedelt.

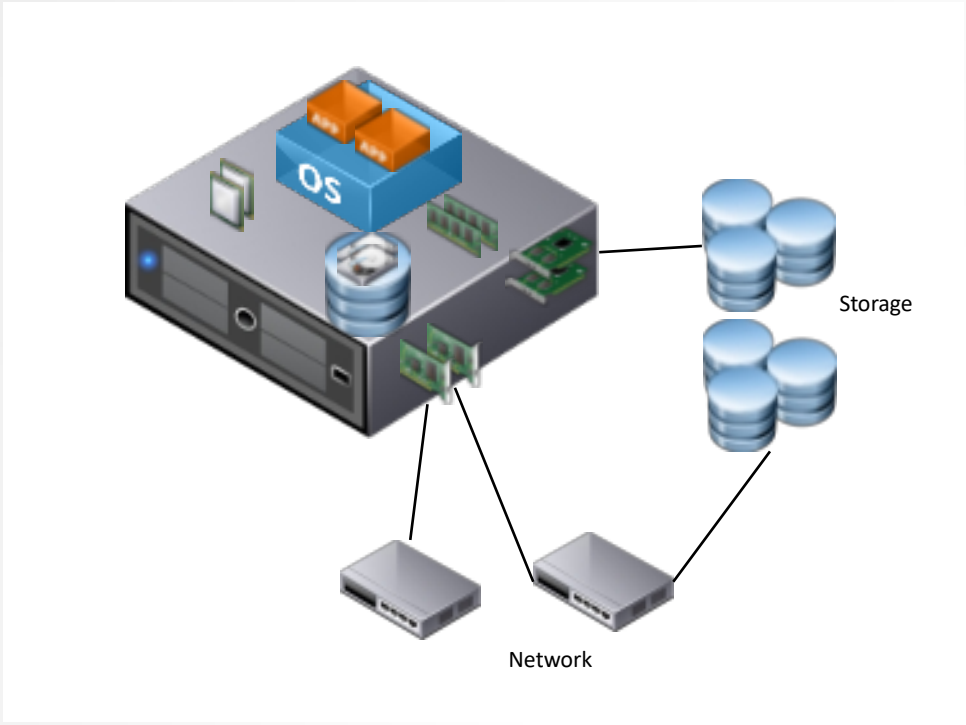
- **Layer 3**
  Protokoll basierend
  Hier werden IP-Adressen einem VLAN zugeordnet.
  Die Zuordnung der Ports zu VLANs erfolgt über ein dynamisches VLAN-Protokoll.

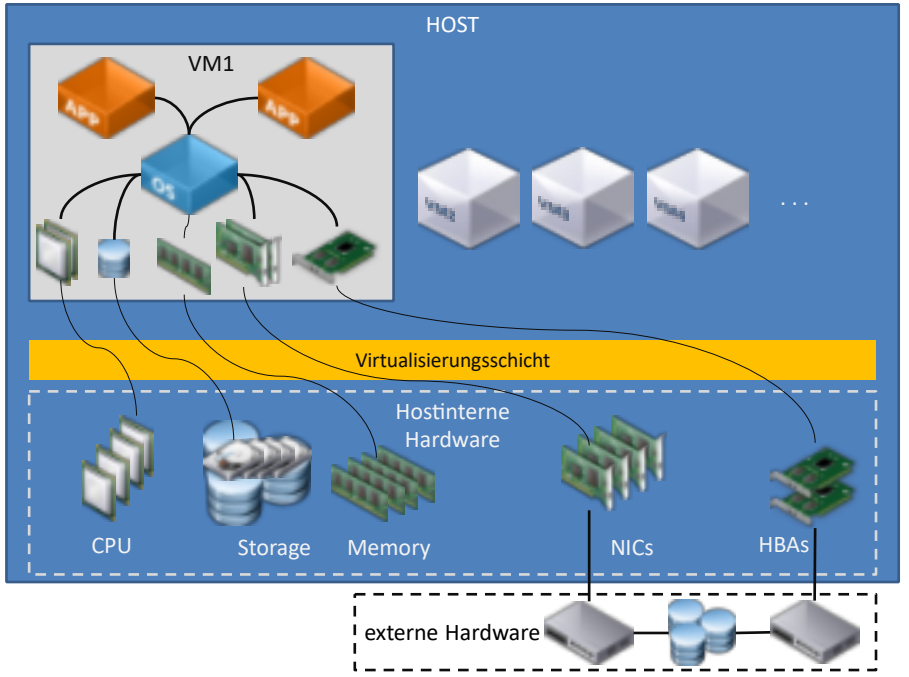
- **Layer 4**
  TCP/IP-Port basierend
  Hier werden TCP- oder UDP-Ports einem VLAN zugeordnet.
  Die Zuordnung der Ports zu VLANs erfolgt über ein dynamisches VLAN-Protokoll.

VLANs
Teil-10

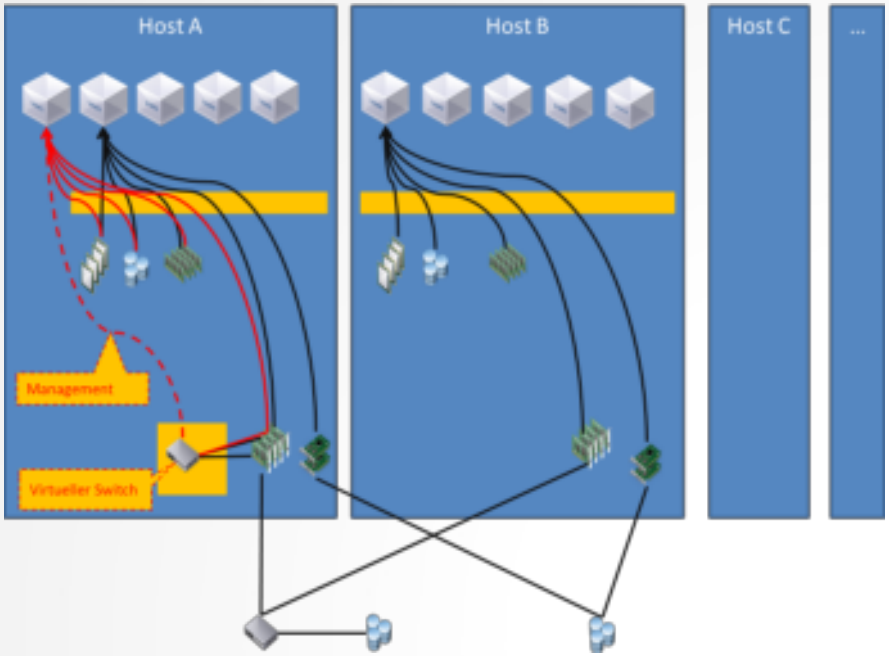


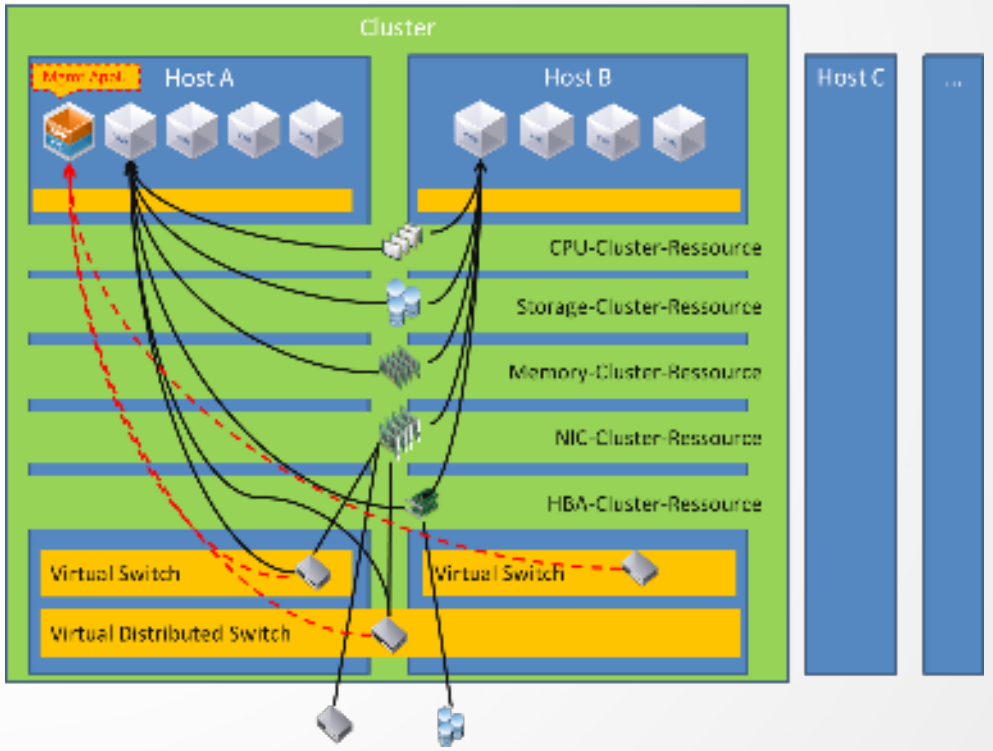
SDN
Teil-1


Os
Storage
Network

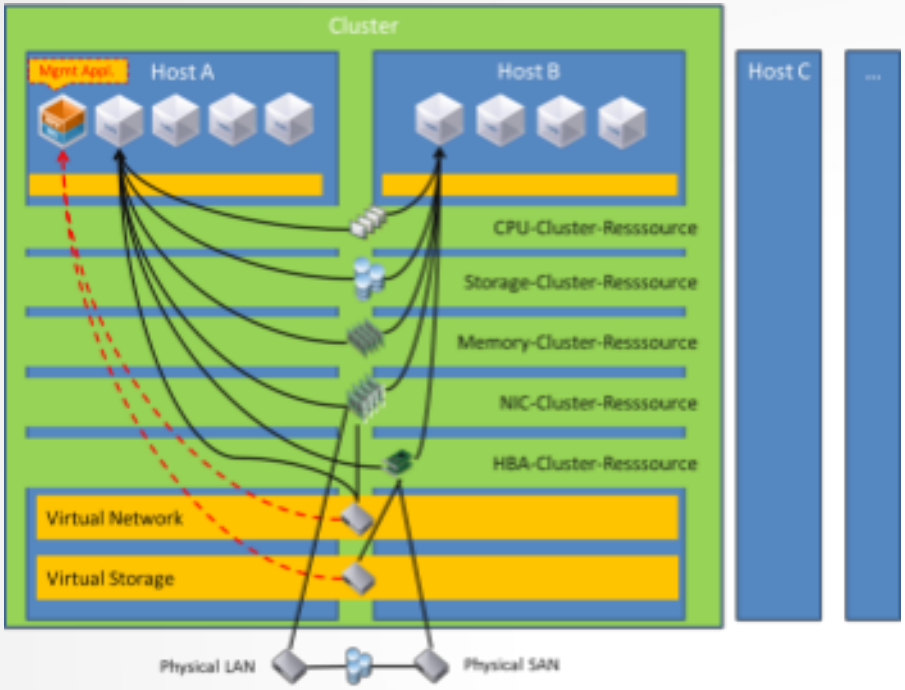

HOST
VM1
APP
APP
OS
Virtualisierungsschicht
Hostinterne
Hardware
CPU
Storage
Memory
NICs
HBAs
externe Hardware

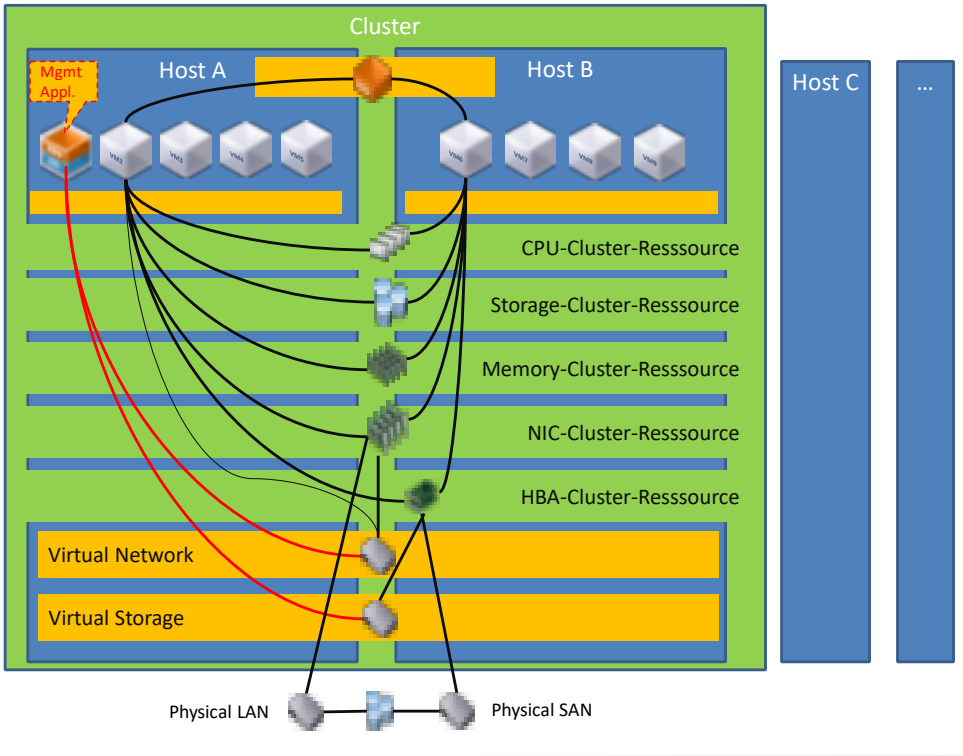
SDN
Teil-2


Host A
Host B
Host C
...
Management
Virtueller Switch

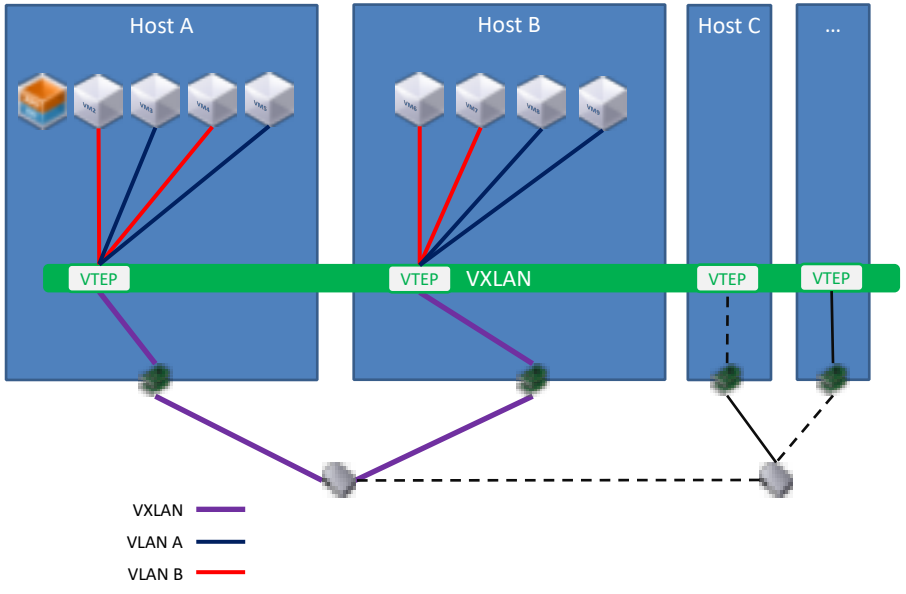

Cluster
Host A
Host B
Host C
...
CPU-Cluster-Ressource
Storage-Cluster-Ressource
Memory-Cluster-Ressource
NIC-Cluster-Ressource
HBA-Cluster-Ressource
Virtual Switch
Virtual Switch
Virtual Distributed Switch

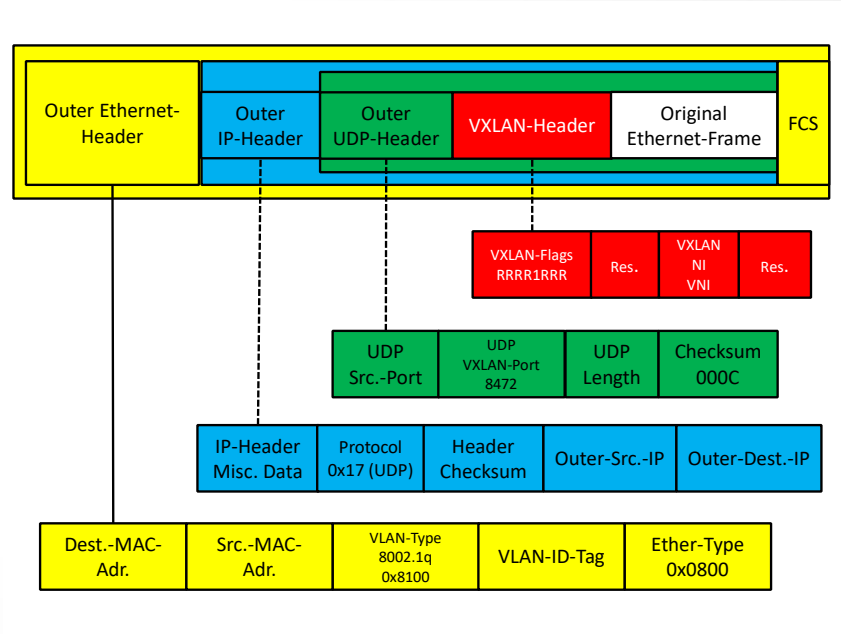
SDN
Teil-3


Cluster
Mgmt Appl.
Host A
Host B
Host C
...
CPU-Cluster-Resssource
Storage-Cluster-Resssource
Memory-Cluster-Resssource
NIC-Cluster-Resssource
HBA-Cluster-Resssource
Virtual Network
Virtual Storage
Physical LAN
Physical SAN

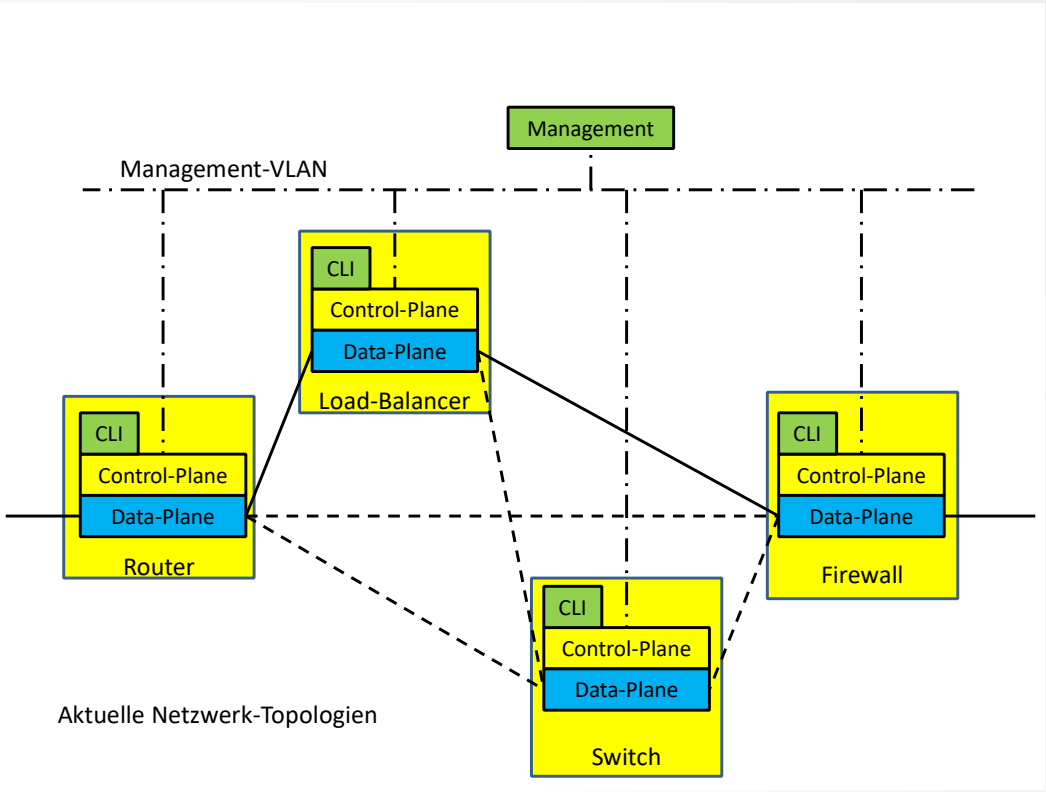

Cluster
Mgmt Appl.
Host A
Host B
Host C
...
CPU-Cluster-Resssource
Storage-Cluster-Resssource
Memory-Cluster-Resssource
NIC-Cluster-Resssource
HBA-Cluster-Resssource
Virtual Network
Virtual Storage
Physical LAN
Physical SAN

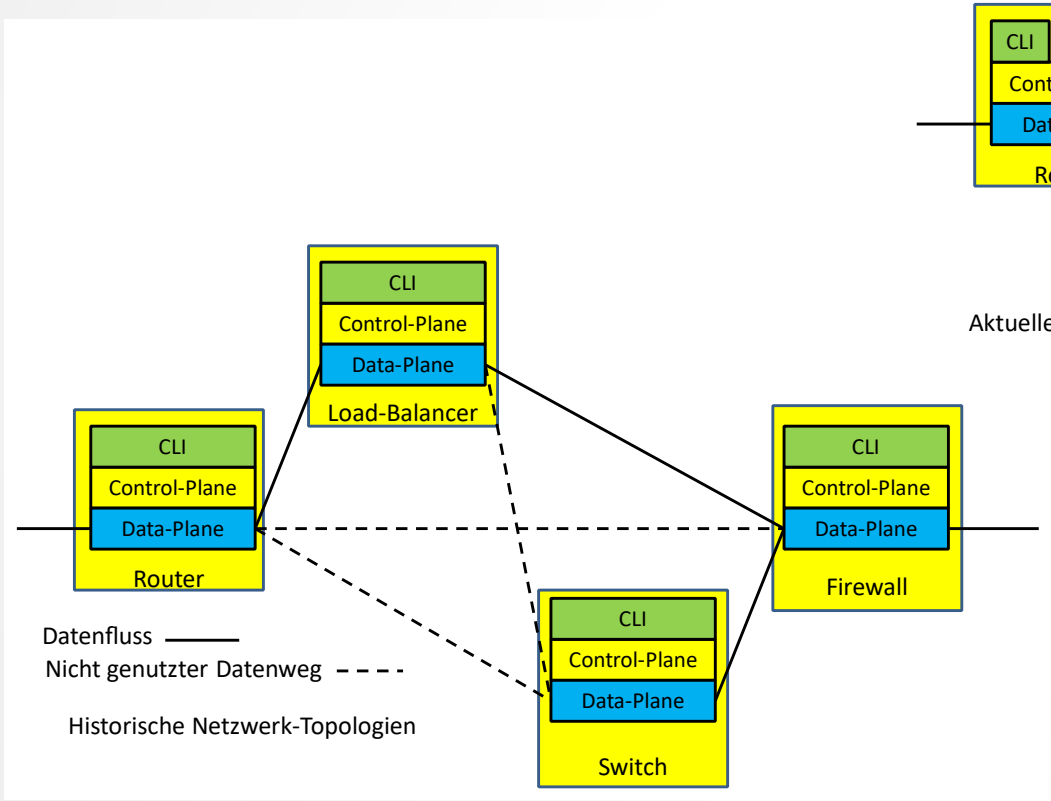
SDN
Teil-4


Host A
Host B
Host C
...
VTEP
VTEP
VXLAN
VTEP
VTEP
VXLAN
VLAN A
VLAN B

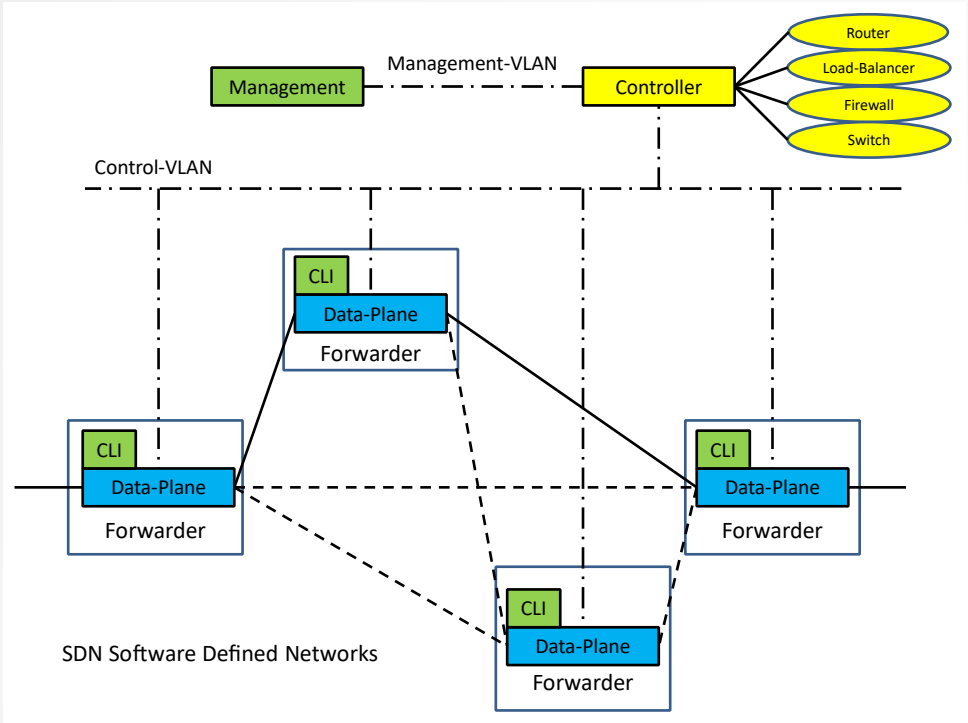

Outer Ethernet-Header
Outer IP-Header
Outer UDP-Header
VXLAN-Header
Original Ethernet-Frame
FCS
VXLAN-Flags RRRR1RRR
Res.
VXLAN NI VNI
Res.
UDP Src.-Port
UDP VXLAN-Port 8472
UDP Length
Checksum 000C
IP-Header Misc. Data
Protocol 0x17 (UDP)
Header Checksum
Outer-Src.-IP
Outer-Dest.-IP
Dest.-MAC-Adr.
Src.-MAC-Adr.
VLAN-Type 8002.1q 0x8100
VLAN-ID-Tag
Ether-Type 0x0800

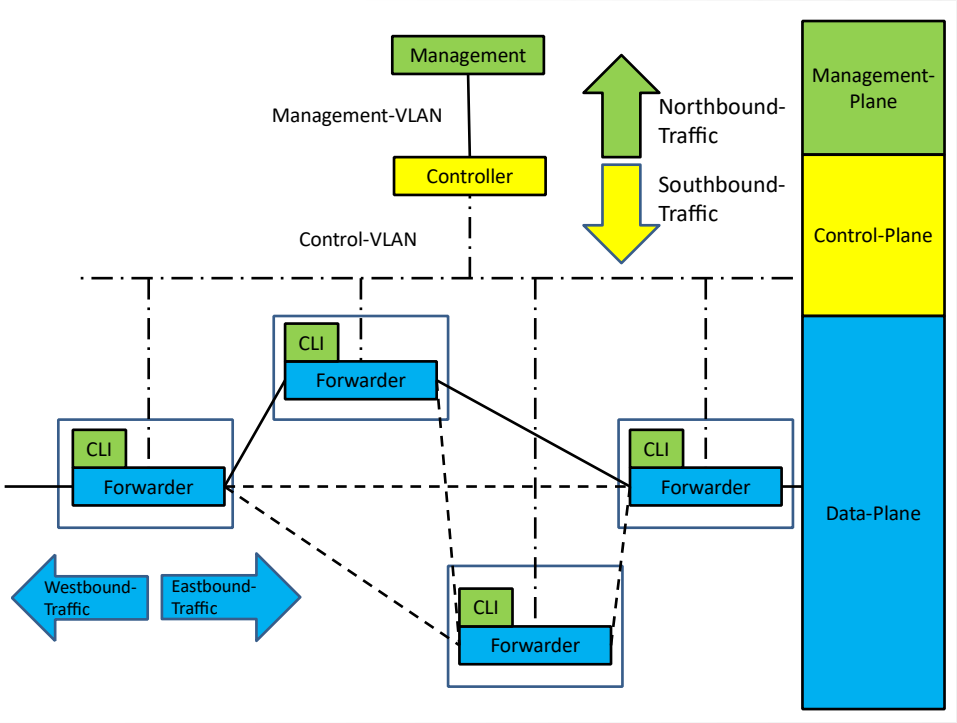
SDN
Teil-5


Management
Management-VLAN
CLI
Control-Plane
Data-Plane
Load-Balancer
CLI
Control-Plane
Data-Plane
Router
CLI
Control-Plane
Data-Plane
Firewall
CLI
Control-Plane
Data-Plane
Switch
Aktuelle Netzwerk-Topologien


CLI
Control-Plane
Data-Plane
Load-Balancer
CLI
Control-Plane
Data-Plane
Router
CLI
Control-Plane
Data-Plane
Firewall
CLI
Control-Plane
Data-Plane
Switch
Datenfluss
Nicht genutzter Datenweg
Historische Netzwerk-Topologien

SDN
Teil-6


Management
Management-VLAN
Controller
Router
Load-Balancer
Firewall
Switch
Control-VLAN
CLI
Data-Plane
Forwarder
CLI
Data-Plane
Forwarder
CLI
Data-Plane
Forwarder
CLI
Data-Plane
Forwarder
SDN Software Defined Networks


Management
Management-VLAN
Controller
Control-VLAN
Northbound-Traffic
Southbound-Traffic
Management-Plane
Control-Plane
Data-Plane
CLI
Forwarder
CLI
Forwarder
CLI
Forwarder
CLI
Forwarder
Westbound-Traffic
Eastbound-Traffic

UDP
Einsortierung im ISO-7Schichten-Modell / Header

Dienst	Ebene
SNMP,DHCP, BOOTP, NTP, TFTP, Rservices, DNS, RPC, usw.	> 4
UDP	4
IP	3

Verbindungslose Kommunikation

2 Bytes	2 Bytes	2 Bytes	2 Bytes
Source-Port	Destination-Port	Message-Length	Checksum

TCP
Einsortierung im ISO-7Schichten-Modell / Header

Dienst	Ebene
HTTP, TELNET, FTP SMTP, Rservices, RFC1006, RPC, usw.	> 4
TCP	4
IP	<4

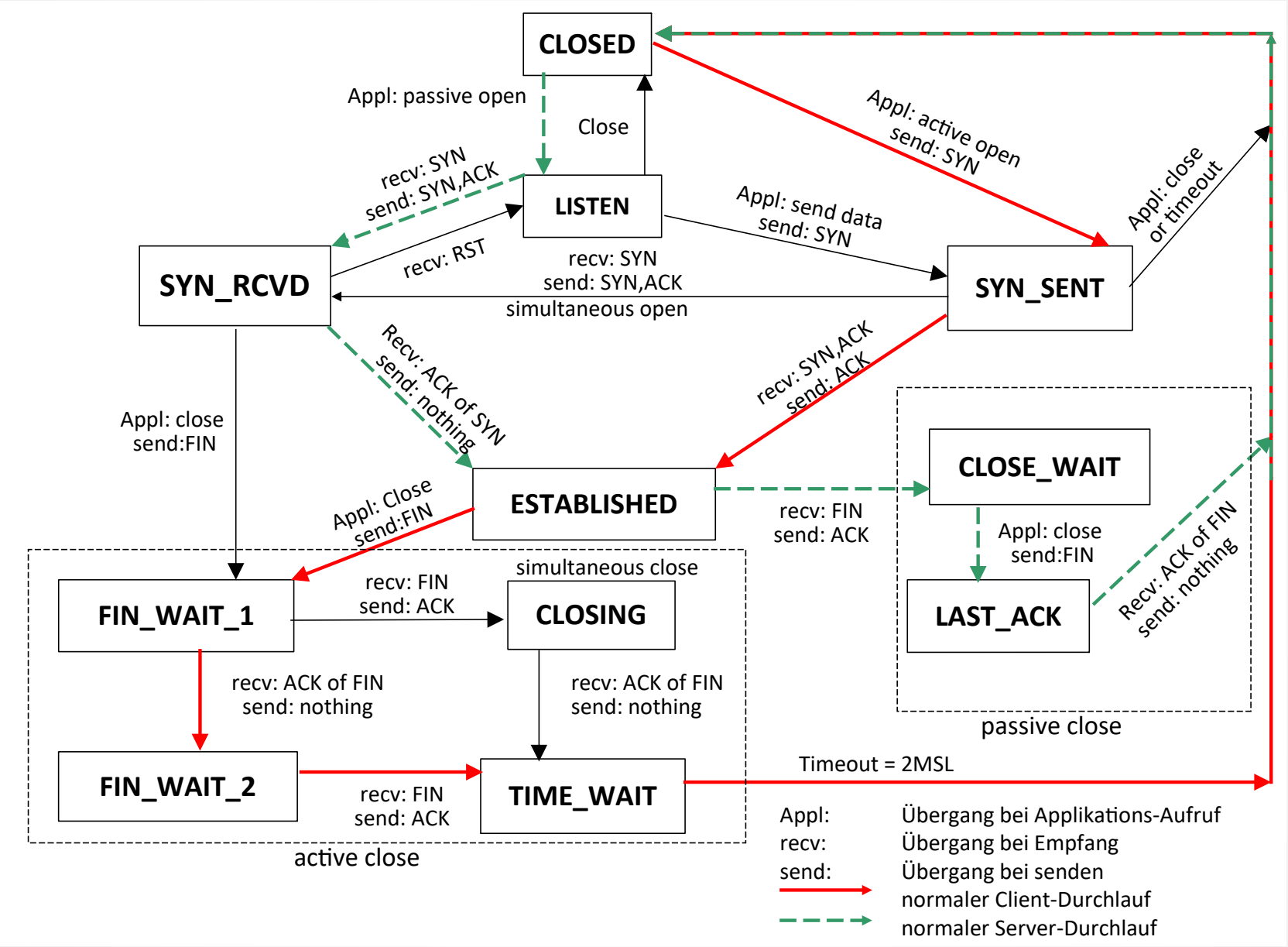
Verbindungsorientierte Kommunikation

Source-Port (16Bit)								Destination-Port (16Bit)
Sequence-Number (32 Bit)								
Acknowledgement-Number (32 Bit)								
Header-Length (4Bit)	Reserved (6Bit)	URG	ACK	PSH	RST	SYN	FIN	Window Size (16 Bit)
Check-Sum (16 Bit)								Urgent-Pointer (16 Bit)
Options (falls vorhanden)								
Daten								

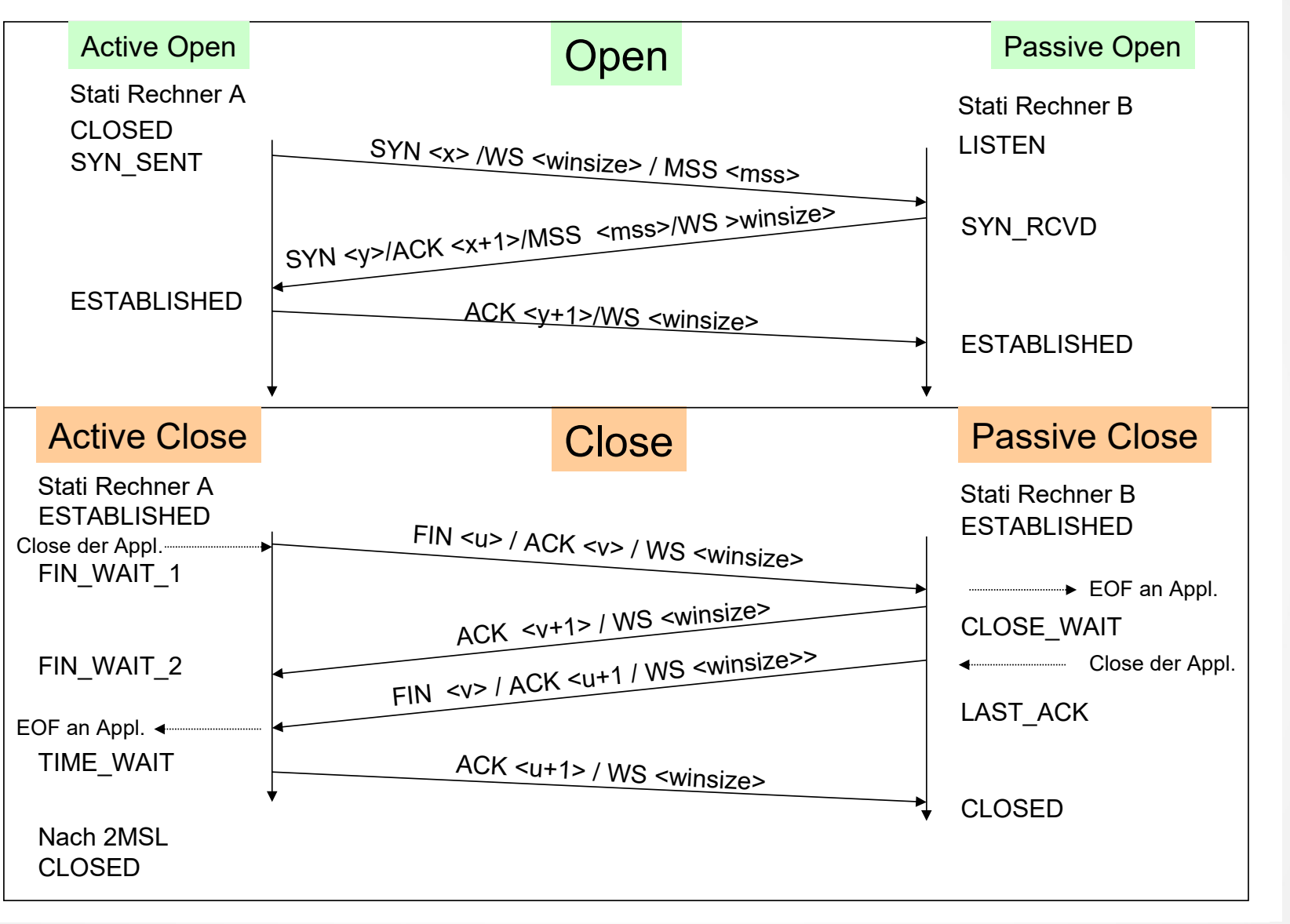
TCP
Flags

- **URG Flag → Urgent-Bearbeitung**
- **ACK Flag → Quittungsbearbeitung**
- **PSH Flag → Datenübertragung**
- **RST Flag → Reset der Verbindung**
- **SYN Flag → Verbindungsaufbau**
- **FIN Flag → Verbindungsabbau**

TCP
Verbindungsstati



TCP
Open / Close



TCP
RST

RST(Reset)
Wird ein SYN-Segment gesendet, ohne dass ein Port auf der anderen Seite dafür geöffnet wurde oder gleichzeitig ein SYN von der anderen Seite gesendet wird, wird der Verbindungsaufbau-Versuch zurückgewiesen.
Dies wird mit einem RST-Segment das an den SYN-Sender zurückgegeben wird durchgeführt.
Ein RST-Segment wird auch gesendet, wenn ein Segment für eine nicht etablierte Verbindung eintrifft.
Zu einer bestehenden Verbindung gehören jeweils zwei IP-Adressen und die zugehörigen Ports.
Bei UDP wird in diesem Fall eine ICMP-Meldung zurückgesendet.

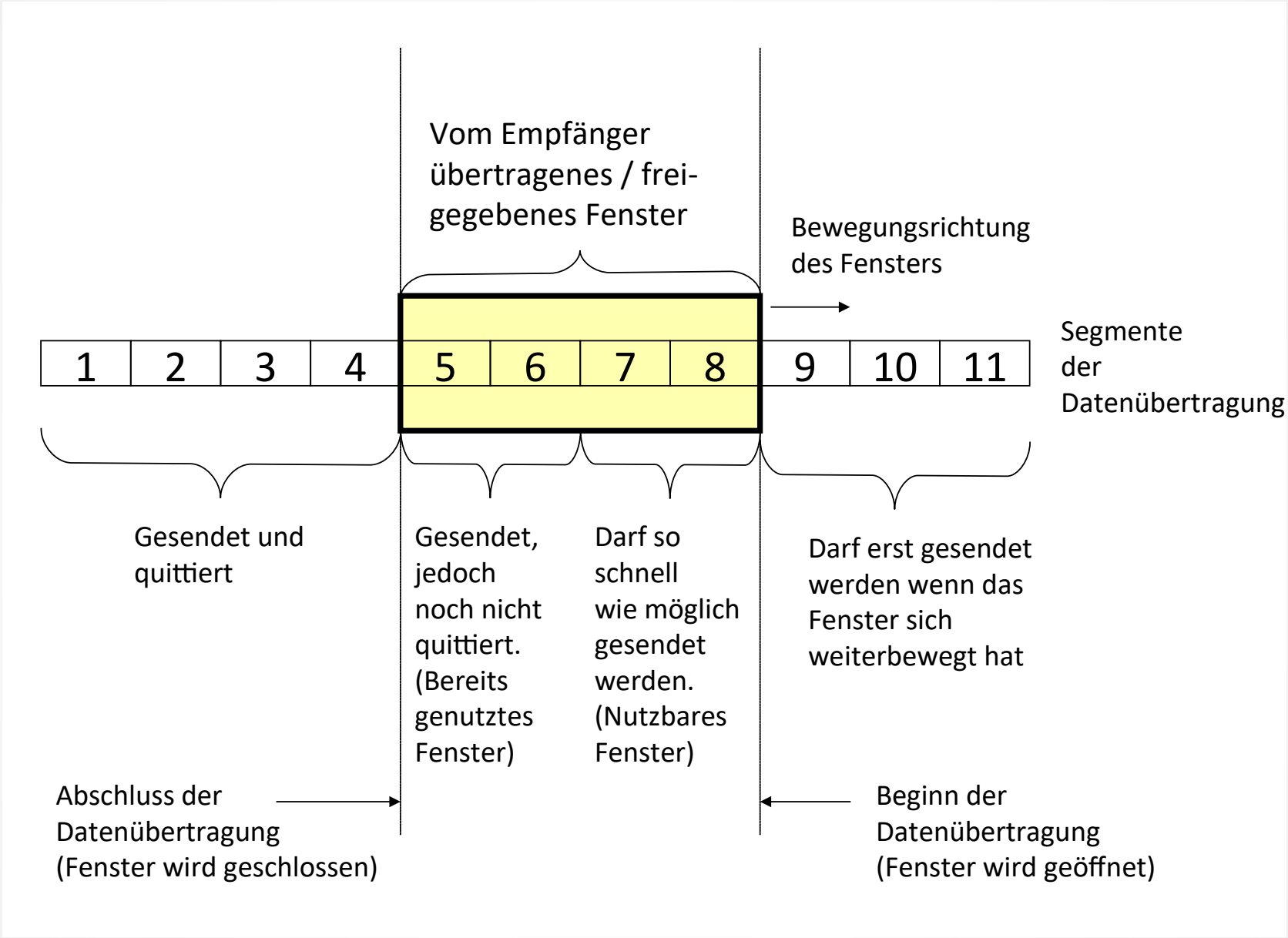
SYN SNR<x> WIN<y> MSS<z> ->
<- RST SNR 0 ACK <x+1> WIN 0

Ein RST auf ein SYN bedeutet somit, dass bei dem Empfänger entweder die IP-Adresse nicht stimmt, oder dass auf dem Port kein Partner hört.

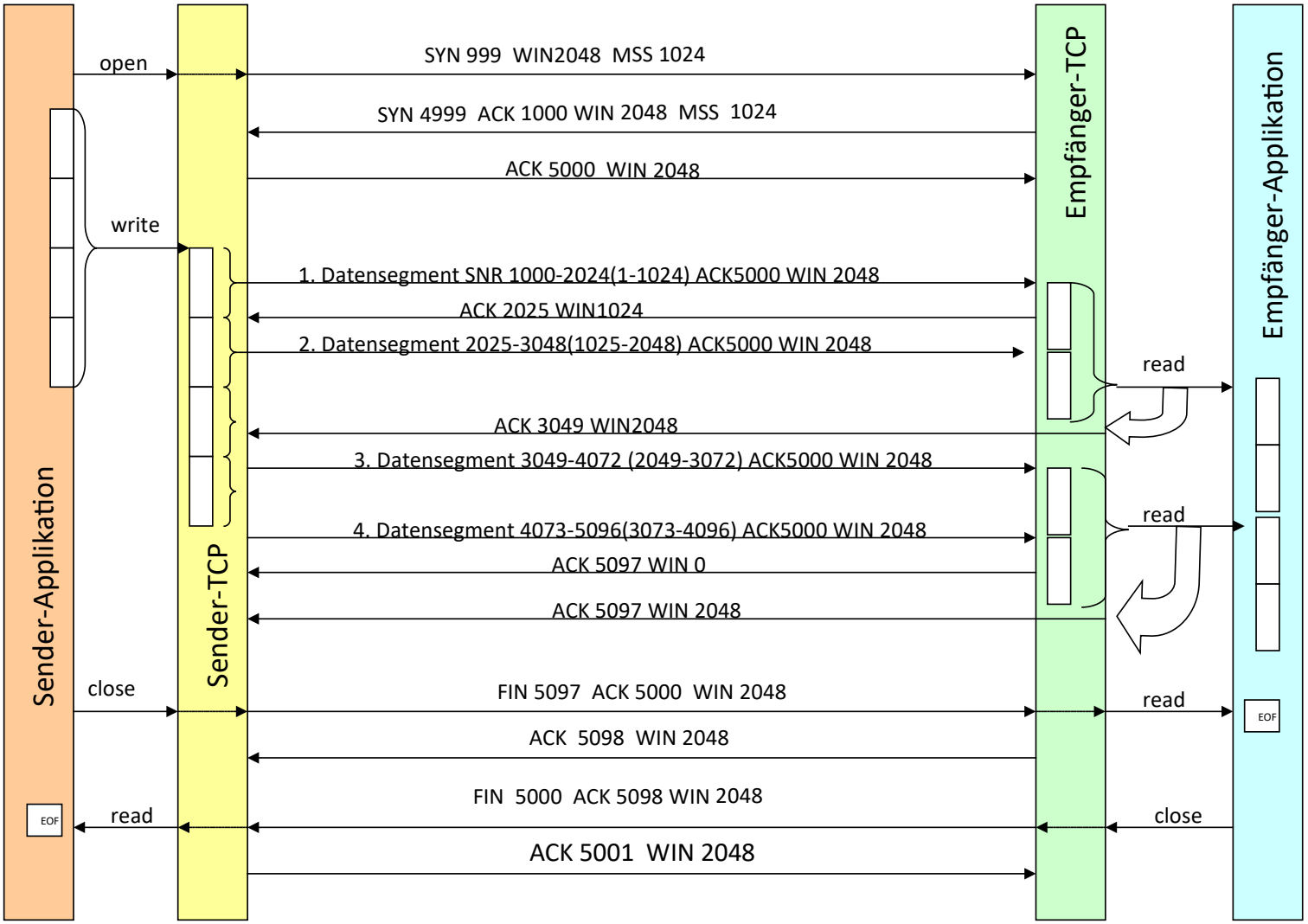
Abbrechen einer Verbindung
Normalerweise wird eine Verbindung mit FIN ... beendet.
Dies wird orderly release genannt.

Es ist auch möglich, eine Verbindung mit einem RST –Segment anstelle einer FIN-Sequenz zu beenden.
Dies wird abortive release genannt.
Dabei werden die Daten in den Puffern verworfen und ein RST-Segment wird gesendet.
Dies kann mit der SO_LINGER - Socket-Option gemacht werden.
Dabei wird mit einer linger-time (deutsch: Verzögerungszeit) von 0 genau dies gemacht.

TCP
Sliding Window



TCP
Datenübertragung-1



TCP
Datenübertragung-2

Segment	Inhalt/Bedeutung
1	Das erste Datensegment wird gesendet.
2	Die Daten werden von TCP mit einem ACK quittiert. Die Windowgröße wird von 2048 auf 1024 reduziert, da TCP die Daten der Applikation noch nicht weitergegeben hat.
3	Das nächste Datensegment wird gesendet. Der Empfangspuffer des Empfängers ist jetzt voll. Der Sender weiß dies und muss warten, bis der Empfänger mit einer Windowgröße > 0 einen neuen Empfangs-Puffer zur Verfügung stellt.
4	TCP übergibt die Daten der Empfänger-Applikation. Damit kann TCP den Empfangs-Puffer räumen. I ACK wird dies durch die Windowgröße 2048 mitgeteilt. Der Sender kann jetzt wieder Daten senden.
5	Das 3. Datensegment wird übertragen.
6	Da noch Platz im Empfangs-Puffer sein muss, kann jetzt gleich das 4. Datensegment gesendet werden. Jetzt sind keine weiteren Daten mehr übertragbar, der Empfangs-Puffer ist voll.
7	TCP auf der Empfängerseite teilt dem TCP auf der Sender-Seite mit, dass die Daten einschließlich dem 4. Segment empfangen wurden und im Empfangs-Puffer stehen. Da dort kein Platz mehr für weitere Daten ist, wird die Windowgröße 0 in der Quittung mitgeteilt.
8	Die Daten werden der Empfänger-Applikation übergeben und somit ist der TCP-Empfangs-Puffer räumbar. Sobald der Puffer geräumt ist, wird dem Sender mitgeteilt, dass wieder ein Empfangs-Puffer > 0 zur Verfügung steht. Dies wird in einem so genannten Window-Update – Segment gemacht
9	Da die Sender-Applikation keine Daten mehr zu senden hat, durchläuft sie einen Close-Aufruf. TCP sendet daraufhin das FIN-Segment. Der Sender ist daraufhin im FIN_WAIT_1-Status
10	Das FIN-Segment wird vom Empfänger-TCP mit einem ACK bestätigt. Somit ist der Empfänger nun im CLOSE_WAIT-Status. Der Sender ist daraufhin im FIN_WAIT_2-Status. EOF wird an die Empfänger-Applikation übergeben, die daraufhin ebenfalls einen Close-Aufruf durchläuft.
11	Der Close-Aufruf auf der Empfängerseite erzeugt nun ein FIN-Segment. Der Empfänger ist im LAST_ACK-Status.
12	Der Sender-TCP quittiert den Empfang des FIN mit einem ACK und ist nun im TIME_WAIT-Status. Dieser geht nach 2MSL in den CLOSED-Status über. Nach dem Empfang des ACK ist die Empfänger-Verbindung im Status CLOSED

TCP
Bandwidth-Delay-Produkt

Bandwidth-Delay-Product

Um nun die optimale Fenster-Größe, auch Verbindungs-Kapazität genannt, zu ermitteln, kann man das Bandwidth-Delay-Product (deutsch: Bandbreiten-Verzögerungszeit-Produkt) anwenden. Dies geschieht folgendermaßen:

Verbindungs-Kapazität [Bytes] = Bandbreite [Bits/Sec] * RTT [Sec]) / 8 [Bits /Byte]

Beispiel:
10Mbps Datenverbindung
5ms RTT

Verbindungskapazität = (10.000.000 * 0.005) / 8 = 6250 Bytes

TCP
Slow-Start

Mit dem Slow Start (deutsch: Langsamer Anfang) ist eine Datenübertragung gemeint, die sich langsam an die optimale Daten-Übertragungs-Abwicklung annähert.
Besonders bei Verbindungen über WAN-Strecken hinweg, kann es vorkommen, dass die Segmente in den Routern zwischengespeichert werden müssen. Würden bereits zu Beginn alle möglichen Daten-Segmente (bis die Window-Size erreicht ist) vom Sender ausgesandt werden, könnte es vorkommen, dass in den Routern Datenpakete mangels Ressourcen verworfen werden müssten.
Deshalb gibt es nicht nur ein Window, welches vom Empfänger vorgegeben wird, sondern es gibt noch ein CWND (Congestion Window; deutsch Überlast-/Daten-Stau-Fenster), welches vom Sender gepflegt wird.
Dabei wird zu Beginn einer Datenübertragung das CWND auf die Größe eines Segments gesetzt.
Dies ist im Normalfall die MSS, welche beim Verbindungsaufbau ausgehandelt wird.
Jedes Mal, wenn ein ACK empfangen wird, wird die CWND um einen Segment-Größe vergrößert.
Der Sender kann dann bis zum Minimum von CWND oder der Windowgröße Daten übertragen.

CWND ist eine Fluss-Kontrolle, die vom Sender aus kontrolliert wird.
WIN ist eine Fluss-Kontrolle, die vom Empfänger aus kontrolliert wird.

TCP
Timer

Retransmission Timer
RTT = aRTT + (1 - a) M

Persist Timer
Der Persist Timer wird beim Erreichen der Windowgröße von 0 von seinem Gegenüber gesetzt.
Ist der Persist Timer abgelaufen, wird ein so genanntes Window-Probe-Paket (ACK) gesendet, um beim Gegenüber nach der Window-Größe nachzufragen.
Damit kann aus dieser Deadlock-Situation herausgefunden werden.

Keepalive Timer
Ein Keepalive Timer ist nicht Teil der TCP-Spezifikation.
Eine TCP-Verbindung, die keine Daten austauscht, wechselt keine Pakete miteinander aus.
Trotzdem haben viele TCP-Implementierungen einen Keepalive Timer.
NFS setzt z. B. bei Verwendung von TCP immer den Keepalive Timer auf Client- und Server-Seite.
Bei Rlogin und Telnet wird nur auf der Serverseite der Keepalive Timer gesetzt.

2MSL Timer
Überwachte Zeit im Status TIME-WAIT-State beim Verbindugnsabbau.
Bei einem Active Close verweilt der schließende Kommunikationspartner im TIME-WAIT-State und wartet auf einen Final ACK.
Nun muss genügend Zeit sein, damit ein Final ACK vom anderen Kommunikationspartner verloren gehen und wiederholt werden kann.

Routing-Protokolle
RIPv1

1. Byte	2. Byte	3. Byte	4.Byte
Comand	Version	0	
Family of Net 1		0	
IP-Address of Net 1			
0			
0			
Distancte to Net 1 (Hops)			
Family of Net 1		0	
IP-Address of Net 2			
0			
0			
Distancte to Net 2 (Hops)			
...			

Erster Versuch des Routing Information Protocol (RIP)

RIP ist ein Interior Gateway Protokollen, da es im LAN-Umfeld agiert.
Zustandsabhängig
Als Metrik wird die Anzahl der Hops
(Anzahl der Router bis zum Ziel) verwendet.
→ Distance Vector Routingprotokoll

Dabei handelt es sich um einen Bellman-Ford-Algorithnmus

Wird als Broadcast gesendet. (→ betrifft alle Netzwerk-Teilnehmer!)
Kein Subnetting, da keine Subnet-Mask-Informationen
Maximal 14 Hops (15 Hops gilt als nicht erreichbar)
Updates sind Timer-gesteuert → langsam bei Änderungen

Routing-Protokolle
RIPv1-Ablauf-1

R1-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
a	direct	0
b	direct	0

R2-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
b	direct	0
c	direct	0

R3-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
c	direct	0
d	direct	0

R2-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
b	direct	0
c	direct	0
a	R1	1

R1-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
a	direct	0
b	direct	0
c	R2	1

R3-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
a	R2	2
b	R2	1
c	direct	0
d	direct	0

R2-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
b	direct	0
c	direct	0
a	R1	1
d	R3	1

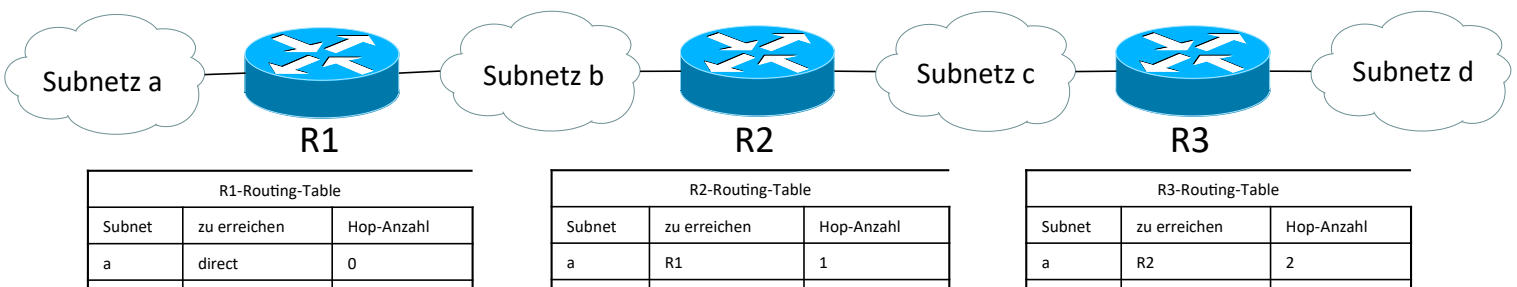
R1-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
a	direct	0
b	direct	0
c	R2	1
d	R2	2

R3-Routing-Table

Subnet	zu erreichen	Hop-Anzahl
a	R2	2
b	R2	1
c	direct	0
d	direct	0

Routing-Protokolle
RIPv1-Ausfall einer Route



R1-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	direct	0
b	direct	0
c	R2	1
d	R2	2

R2-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	R1	1
b	direct	0
c	direct	0
d	R3	1

R3-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	R2	2
b	R2	1
c	direct	0
d	direct	0

R1-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	direct	15
b	direct	0
c	R2	1
d	R2	2

R2-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	R1	1
b	direct	0
c	direct	0
d	R3	1

R3-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	R2	2
b	R2	1
c	direct	0
d	direct	0

R1-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	R2	2
b	direct	0
c	R2	1
d	R2	2

R2-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	R1	3
b	Direct	0
c	Direct	0
d	R3	1

R3-Routing-Table		
Subnet	zu erreichen	Hop-Anzahl
a	R2	4
b	R2	1
c	direct	0
d	direct	0

Routing-Protokolle
RIPv1-beschleunigungsmaßnahmen

Zur Beschleunigung der Konvergenzzeit wurden folgende Maßnahmen definiert.

Split Horizon (deutsch: geteilter Horizont)
Ein Router wird Informationen nur an die Subnetze weitergeben aus denen er die Informationen nicht bekommen hat. Das bedeutet, dass wenn er an einem Port ein Subnetz mitgeteilt bekommen hat, wird er dieses Subnetz an diesem Port nicht wieder weiter verkünden.

Split Horizon und Poison Reverse (deutsch: vergifteter Rückweg)
Hierbei werden wieder alle Subnetze auf allen Ports angekündigt. Allerdings werden die Subnetze mit dem Hopcount 15 zurückgegeben aus denen sie gekommen sind.

Triggered Updates (deutsch: ausgelöste Aktualisierungen)
Hierbei wird nicht bis zum nächsten Timeralarm gewartet bis eine neuen Information verbreitet wird. Es wird sofort nachdem ein Port als „Down“ erkannt wird, die Information weitergeleitet.

Zusätzliche Maßnahmen
Routen die über RIP gelernt wurden haben nur eine Lebensdauer von 3 Minuten. Sollte in der Zwischenzeit kein Update von einem anderen Router erfolgen bekommt die Route die Metrik 16.

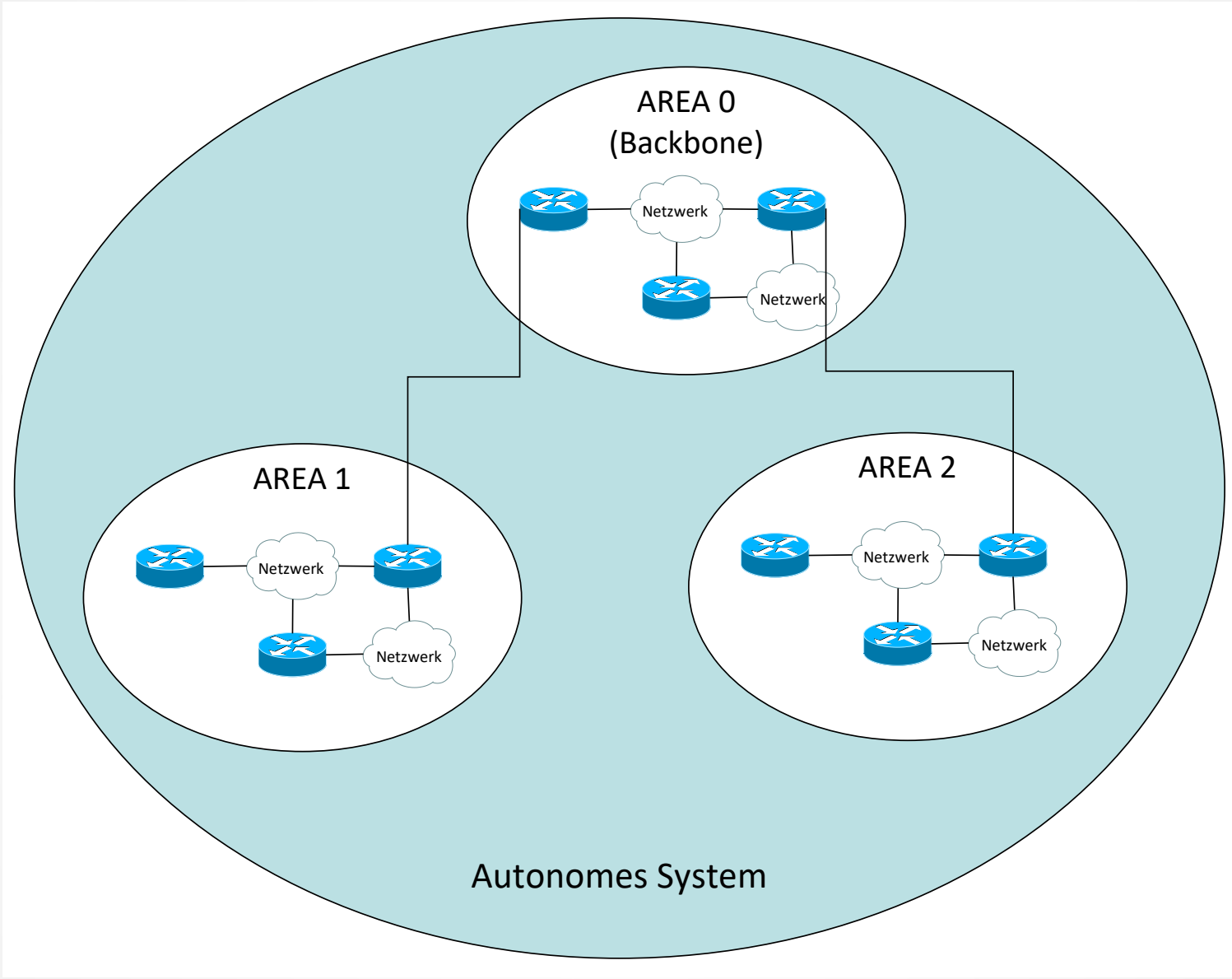
Routing-Protokolle
RIPv2

1. Byte	2. Byte	3. Byte	4.Byte
Comand	Version	Pasword	
Family of Net 1		0	
IP-Address of Net 1			
Subnet-Mask of Net 1			
0			
Distancte to Net 1 (Hops)			
Family of Net 1		0	
IP-Address of Net 2			
Subnet-Mask of Net 2			
0			
Distancte to Net 2 (Hops)			
...			

Verbesserungen gegenüber der Version v1:

- Multicast (IP-Multicast-Adresse 224.0.0.9)
  → Nur noch Router beschäftigen sich damit.
- Subnetzmasken-Information wird übermittelt
  → (VLMS = Variable Length of Subnet Mask).
  Damit ist klassenlose IP-Adressierung / Subnetting möglich.
- Es werden bis zu 25 Netzwerke in einem RIP-Paket mitgeteilt.
- Die maximal gültige Hop-Anzahl ist 14.
  Ein Netzwerk mit 15 Hops wird als nicht erreichbar übermittelt.

Routing-Protokolle
OSPF



Routing-Protokolle
OSPF-Eigenschaften

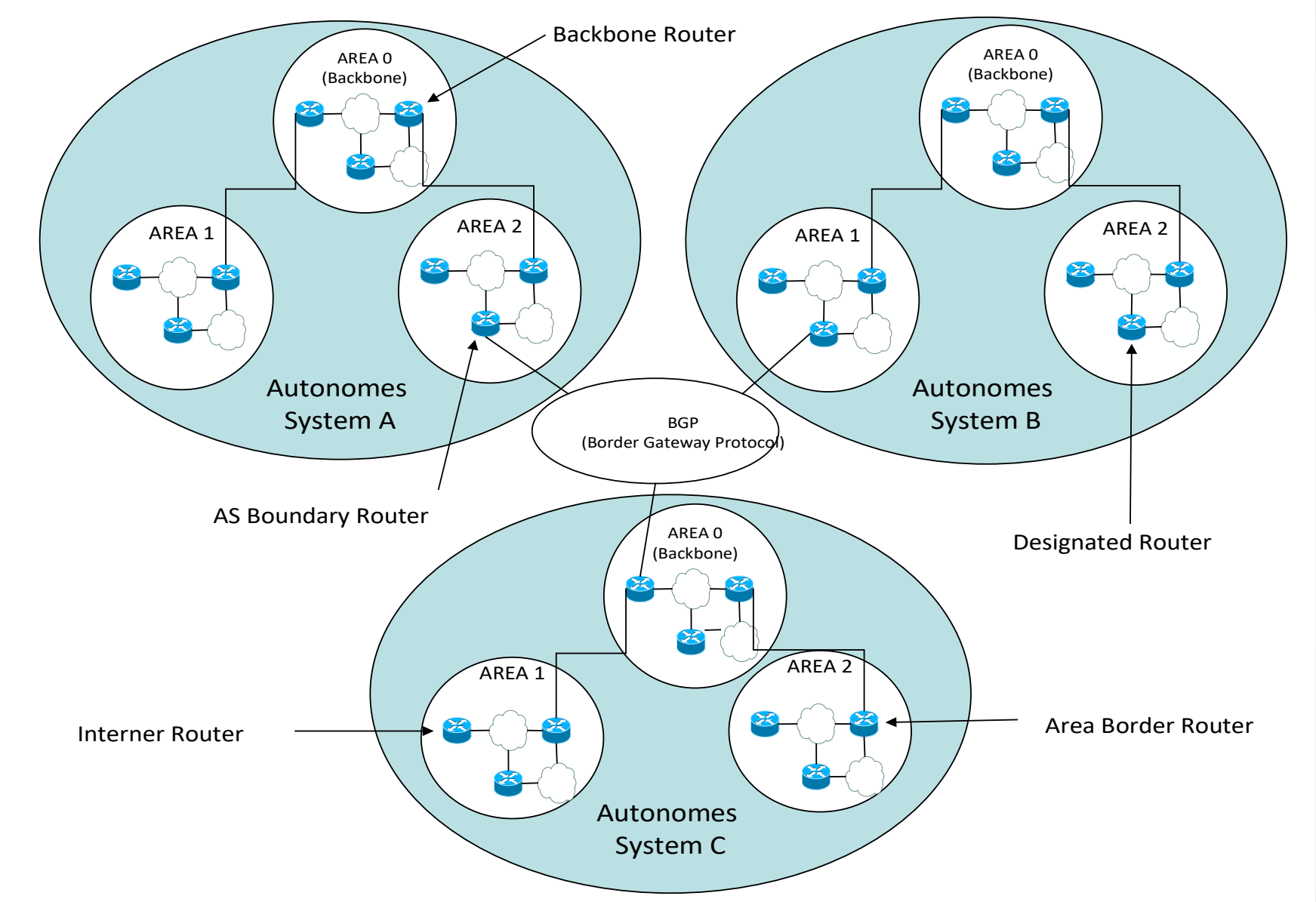
Eine OSPF-Topologie ist in mehrere Hierarchie-Ebenen eingeteilt:
1) Autonomes System → Gesamtheit aller über ein Backbone verbundenen Areas
2) Backbone → Verbindung von Areas
3) Area → Gruppierung von Netzwerken
4) Netzwerk

OSPF unterstützt drei Arten von Verbindungen zwischen Netzwerken:
Punkt-zu-Punkt-Verbindungen zwischen zwei Routern.
Mehrfachzugriffsnetzwerke mit Broadcasting. Das sind die meisten LANs.
Mehrfachzugriffsnetzwerke ohne Broadcasting. Das sind die meisten paketvermittelnden WANs.

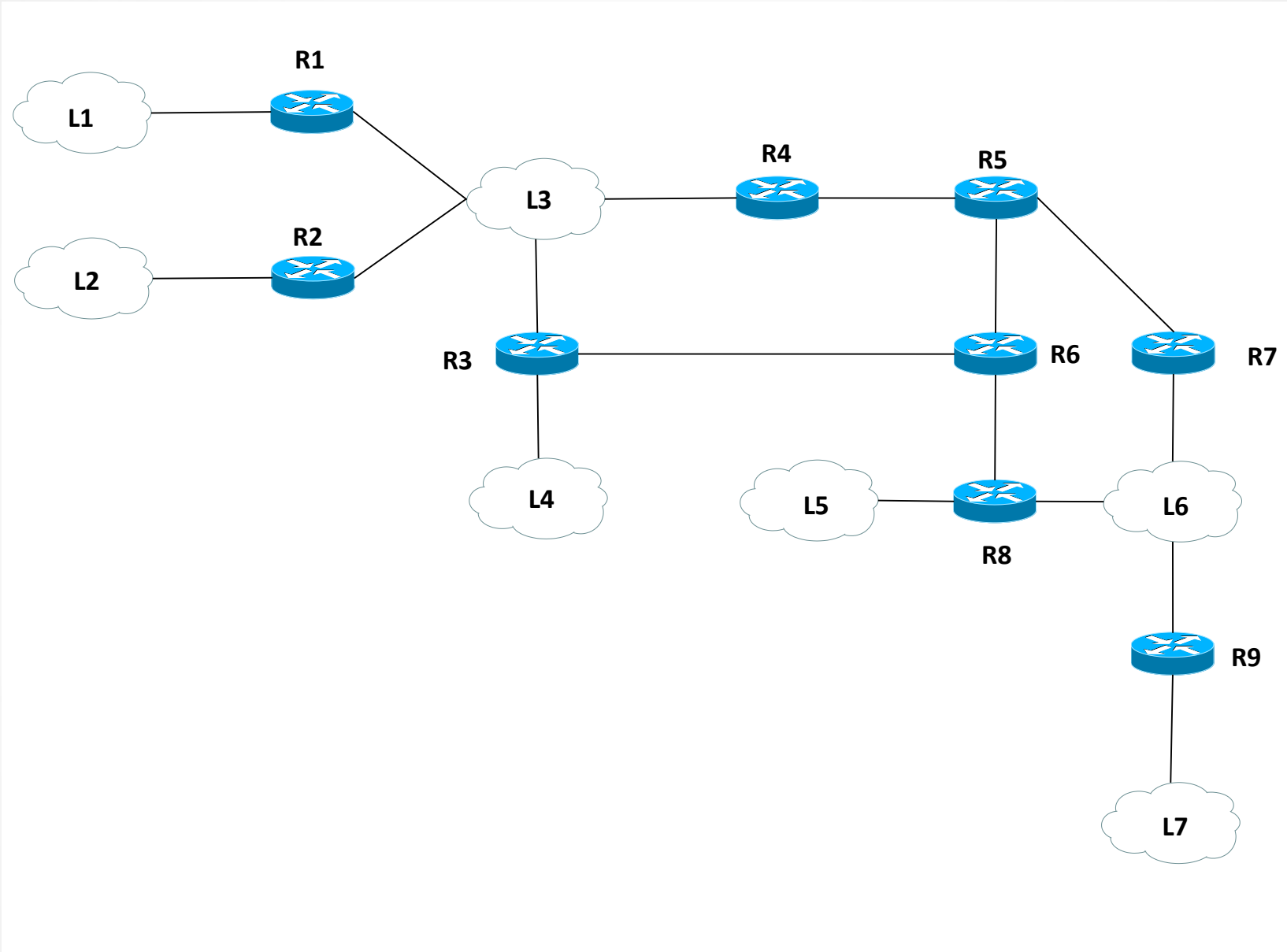
OSPF ist ein Link-State-Algorithmus.
Die Routing Entscheidung wird demnach aufgrund des Link-Status ermittelt bzw. korrigiert.

- Es können Netzwerke über mehr als 14 Zwischen-Systemen erreicht werden.
- OSPF konvergiert bei Netzwerk-Änderungen schneller
- Geringerer Overhead
- Unterstützung hierarchischer Netzwerk-Strukturen
- Unterstützung zur Authentifizierung

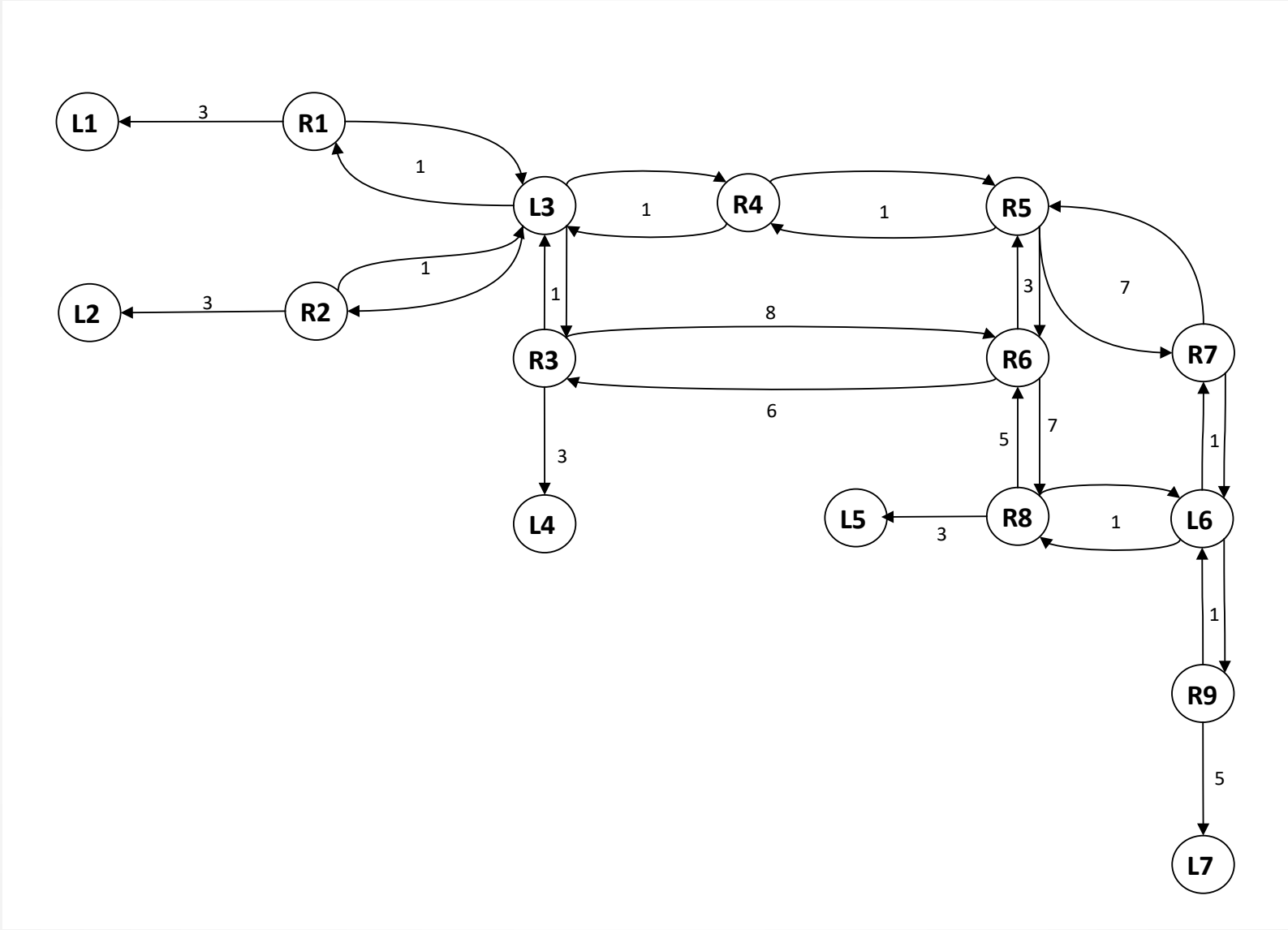
Routing-Protokolle
OSPF-Router-Rollen



Routing-Protokolle
OSPF-Ablauf-1



Routing-Protokolle
OSPF-Ablauf-2



Routing-Protokolle
OSPF-Ablauf-3

Router	Subnetze mit Kosten
R1	L1 = 3, L3 = 1
R2	L2 = 3, L3 = 1
R3	L3 = 1, L4 = 3
R4	L3 = 1
R5	-
R6	-
R7	L6 = 1
R8	L5 = 3, L6 = 1
R9	L6 = 1, L7 = 5

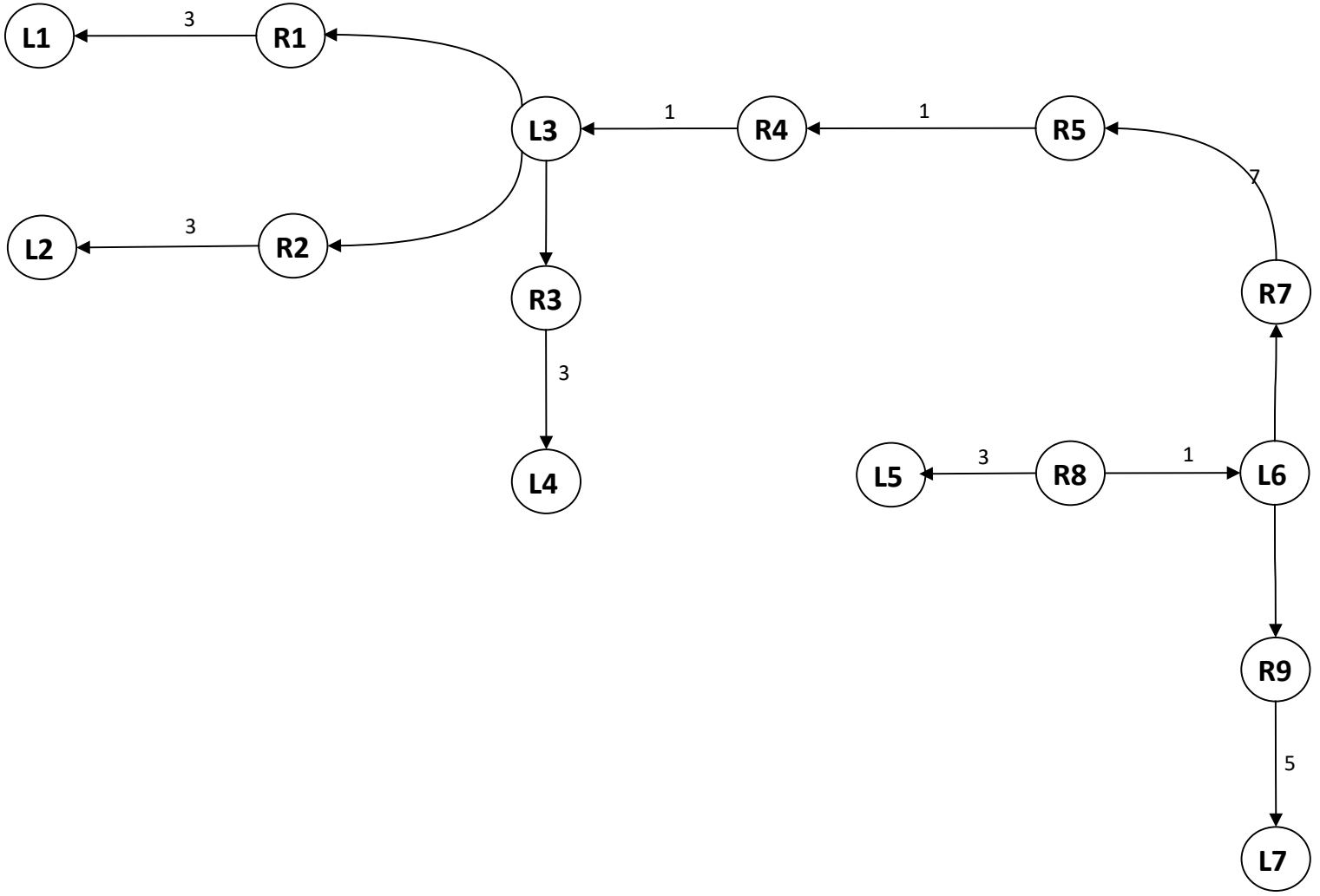
Für die einzelnen Router ergeben sich damit folgende angeschlossenen Netzwerke mit den zugehörigen Kosten. Damit kann für alle Router die LSDB (Link State Data Base) aufgebaut werden.

Die Datenbank ist vollständig, wenn jeder Router von jedem Router eine gültige Liste empfangen hat.

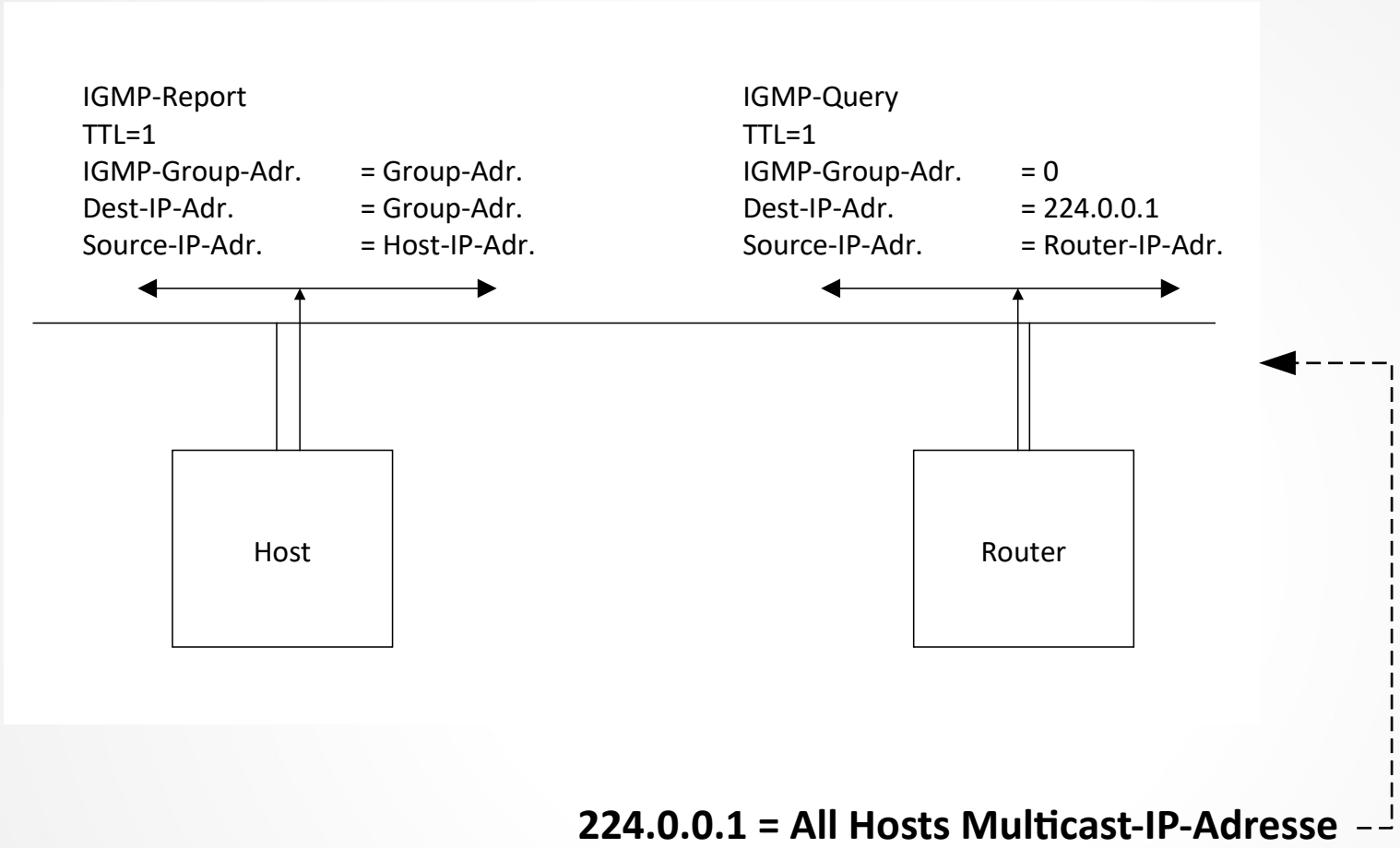
Z. B. Kostentabelle von R8

Ziel-Netzwerk	Next Hop	Distance (Kosten)
L1	R7	13
L2	R7	13
L3	R7	10
L4	R7	13
L5	direct	3
L6	direct	1
L7	R9	6

Routing-Protokolle
OSPF-Ablauf-4

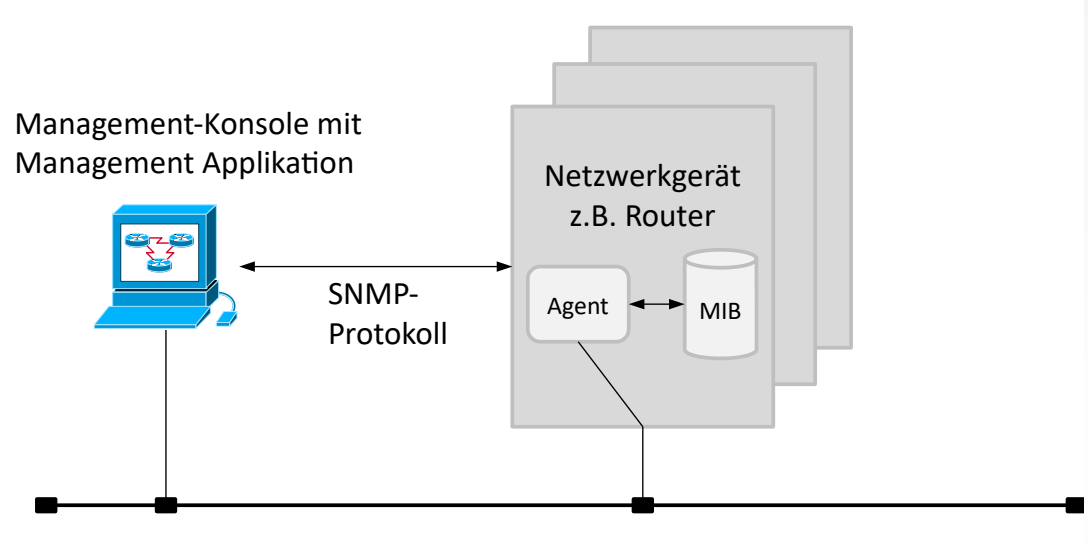


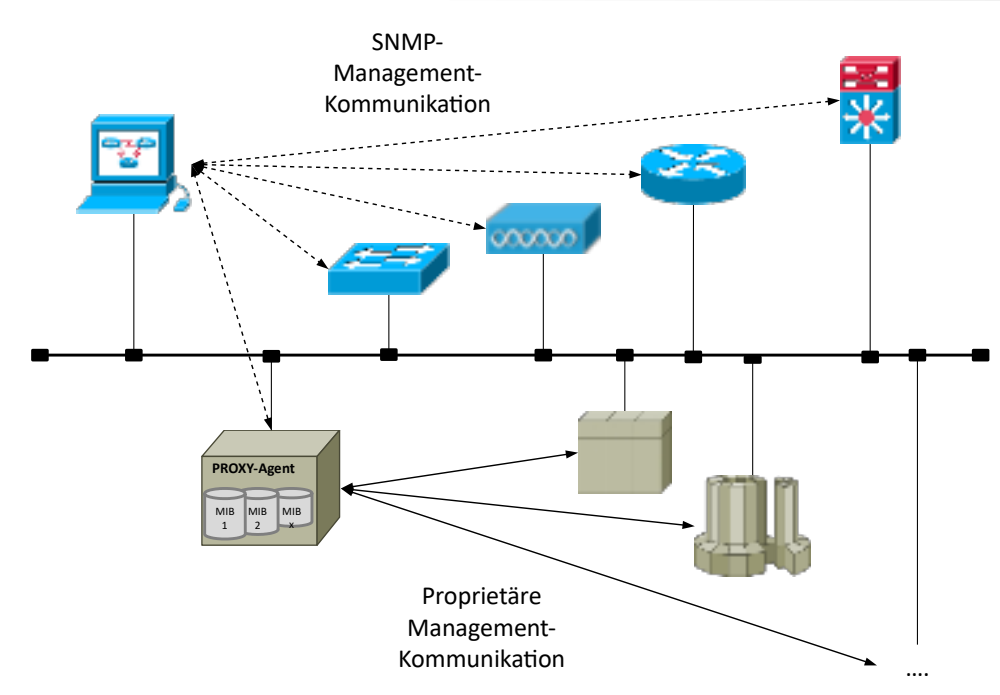
IGMP


IGMP-Report
TTL=1
IGMP-Group-Adr. = Group-Adr.
Dest-IP-Adr. = Group-Adr.
Source-IP-Adr. = Host-IP-Adr.
IGMP-Query
TTL=1
IGMP-Group-Adr. = 0
Dest-IP-Adr. = 224.0.0.1
Source-IP-Adr. = Router-IP-Adr.
Host
Router
224.0.0.1 = All Hosts Multicast-IP-Adresse

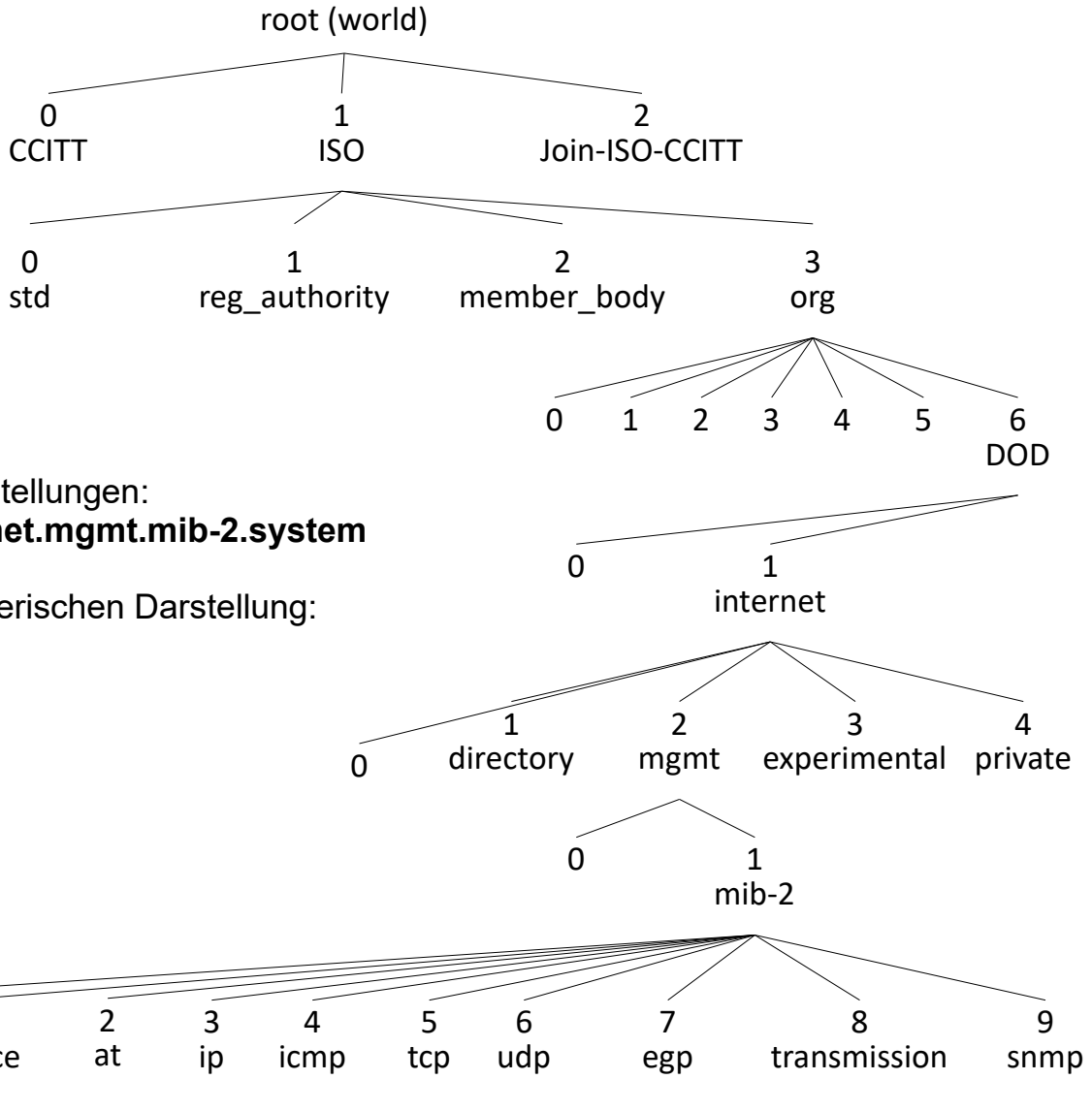
Netzwerk-Management
Übersicht

-	SNMP
TCP	UDP
IP	
Ebene 2	
Ebene 1	





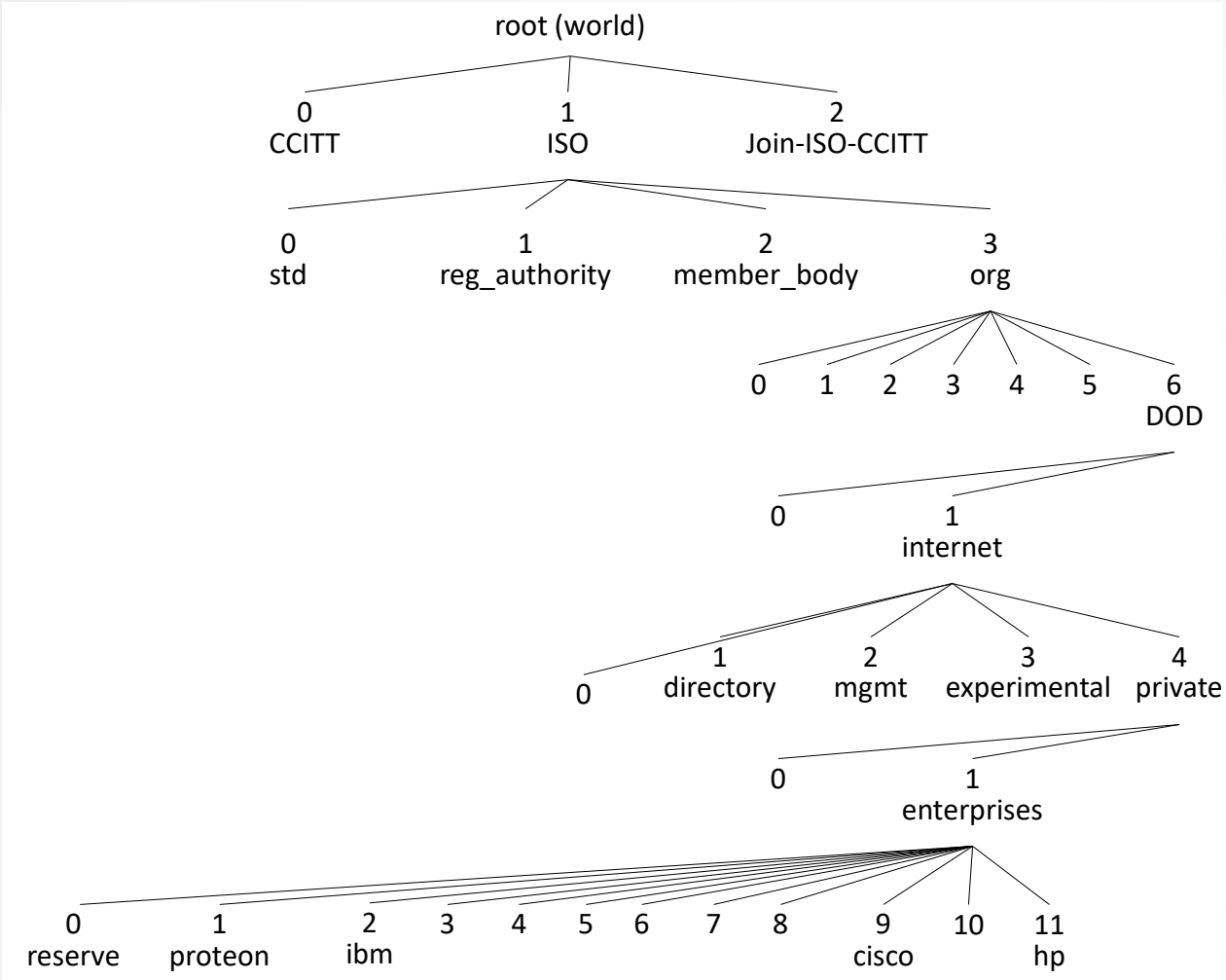
Netzwerk-Management
MIB-Aufbau



Verschiedene Darstellungen:
iso.org.dod.internet.mgmt.mib-2.system

entspricht der numerischen Darstellung:
.1.3.6.1.2.1.0

Netzwerk-Management
MIB-Herstellerteil



Kennung	Hersteller
0	
1	proteon
2	ibm
9	cisco
11	hp
23	novellMib
119	nec
197	kalpana
353	atmForum
437	grandjuction
494	madge
711	lightstream

Netzwerk-Management
SNMPv1 / SNMPv2c

SNMPv1

Aufruf	Bedeutung	Richtung	Port
get	Anforderung einer MIB-Variablen	Manager -> Remote-Gerät	161
getnext	Anforderung der lexikographisch nächsten Variablen	Manager -> Remote-Gerät	
response	Antwort auf einen get/getnext-Telegramm	Remote-Gerät -> Manager	
set	Setzen einer MIB-Variablen	Manager -> Remote-Gerät	
trap	Information an den Manager	Remote-Gerät -> Manager	162

SNMPv2c

Aufruf	Bedeutung	Richtung
get	Anforderung einer MIB-Variablen	Manager -> Remote-Gerät
getnext	Anforderung der lexikografisch nächsten Variablen	Manager -> Remote-Gerät
getbulk	Anforderung großer MIB-Bereiche	Manager -> Remote-Gerät
response	Antwort auf einen get/getnext-Telegramm	Remote-Gerät -> Manager
set	Setzen einer MIB-Variablen	Manager -> Remote-Gerät
inform	Versand bestätigter Meldungen Kommunikation zwischen Managern	Remote-Gerät<-> Manager Manager<-> Manager
trap	Information an den Manager	Remote-Gerät -> Manager

Netzwerk-Management
RMON / SMON

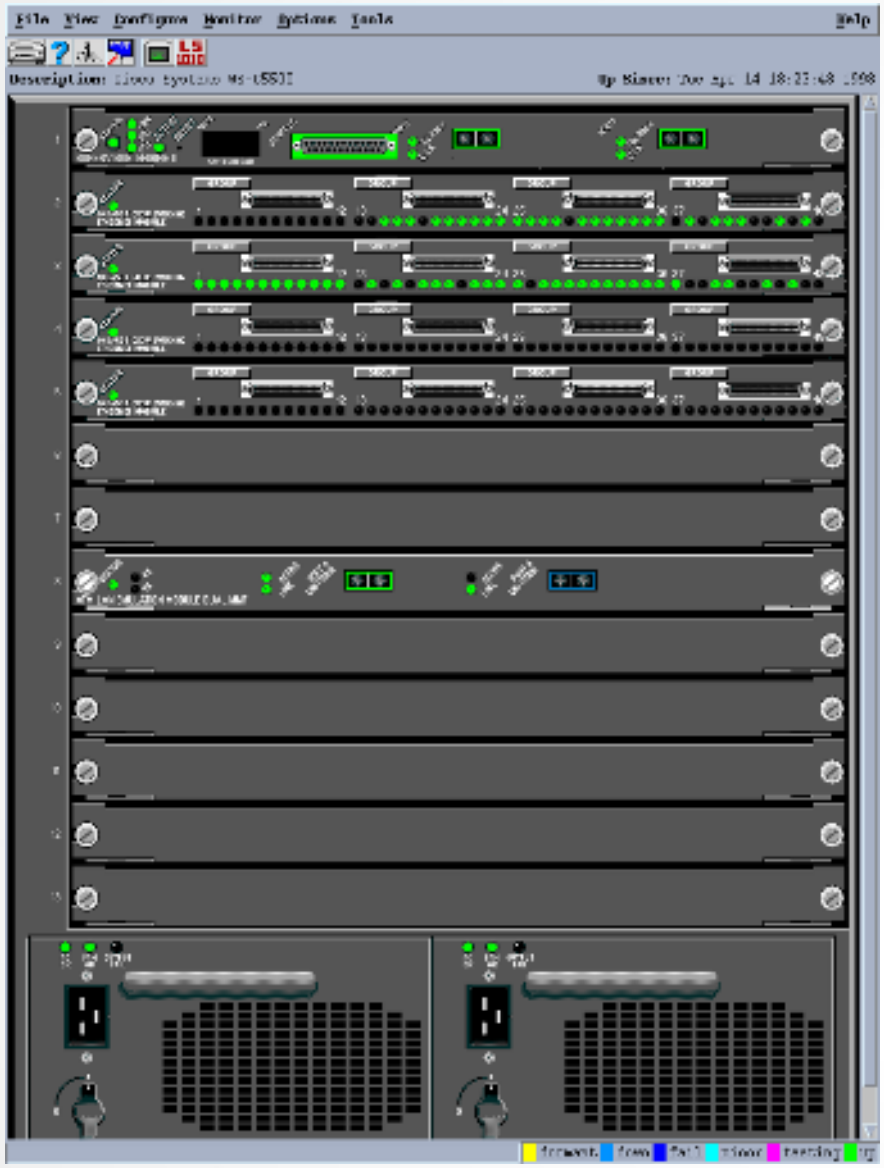
Im MIB-Baum wurde **RMON** unterhalb des MGMT-Knotens angesiedelt.
RMON enthält folgende Klassen:

RMON-Klasse	Bedeutung
1	Statistics
2	History
3	Alarms
4	Hosts
5	Host Top10
6	Matrix
7	Filters
8	Capture
9	Events
10	Token Ring

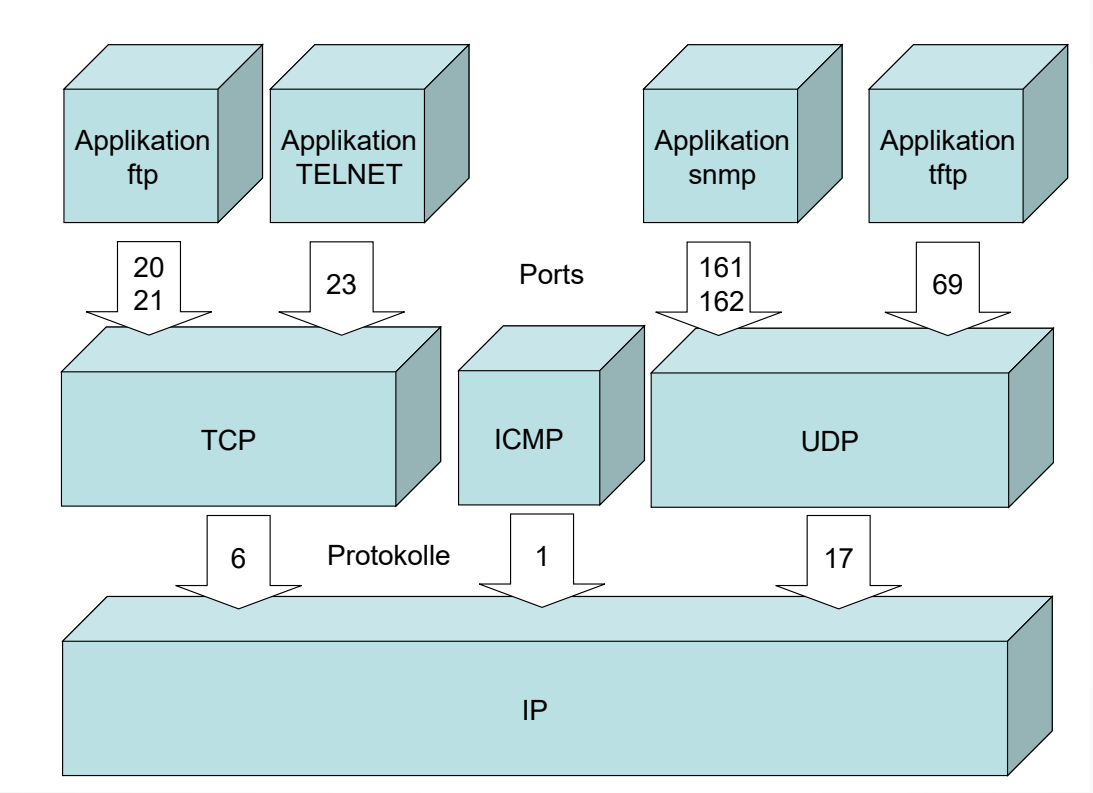
RMON-Klasse	Bedeutung
11	Protocol Directory
12	Protocol Distribution
13	Address Map
14	Network Layer Host
15	Network Layer Matrix
16	Application Layer Host
17	Application Layer Matrix
18	User History
19	Probe Configuration
20	Conformance

SMON
Speziell für Switches wurden die Grundlagen für die Belange von Switches erweitert und in einer eigenen Struktur festgelegt

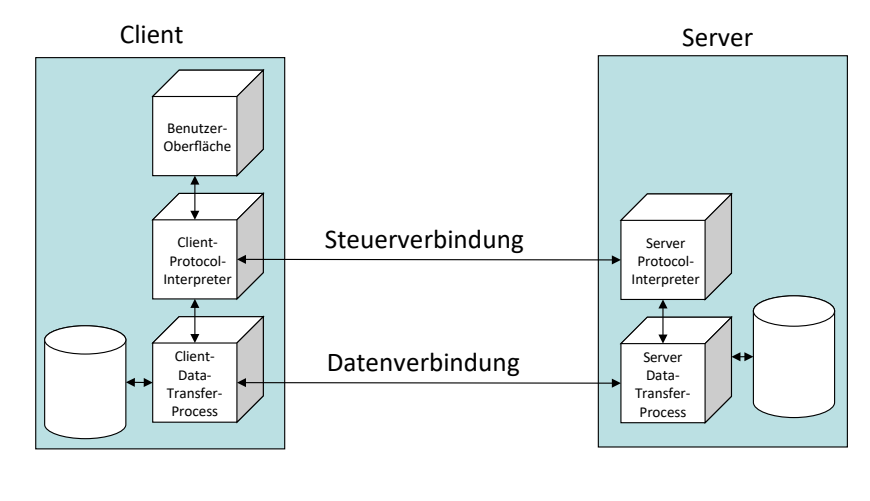
Netzwerk-Management
Beispiel: Cisco-View

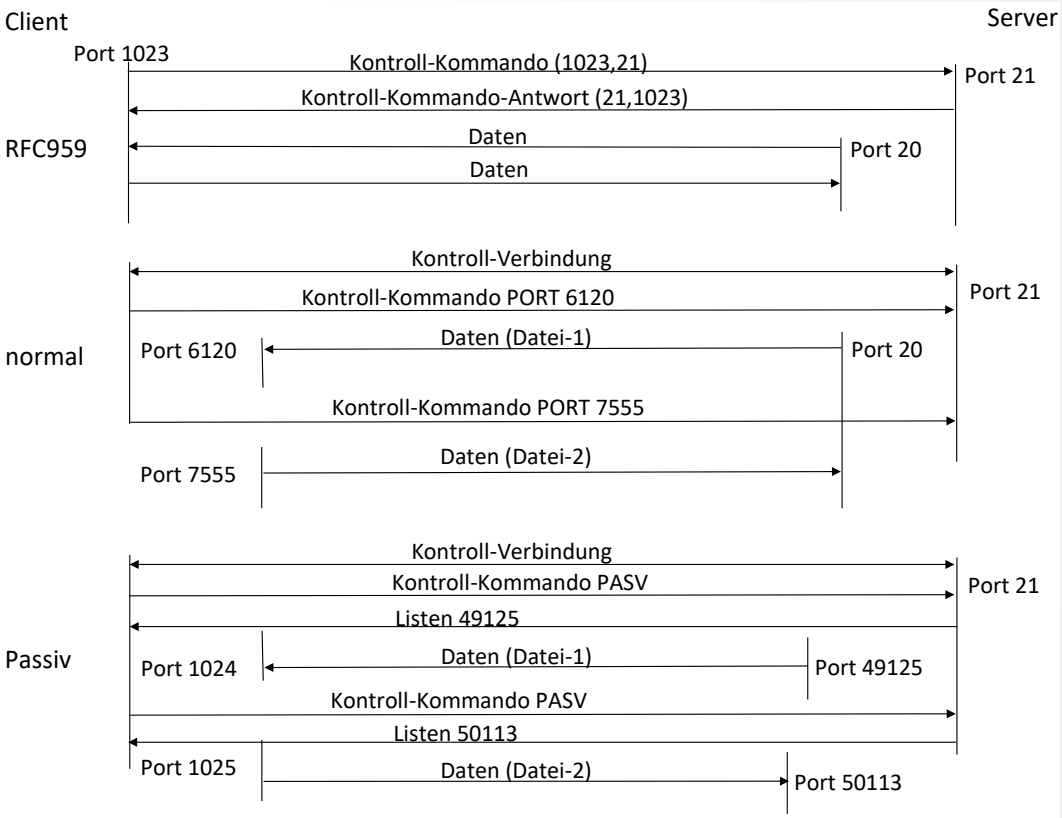


Anwendungsprotokolle
Übersicht

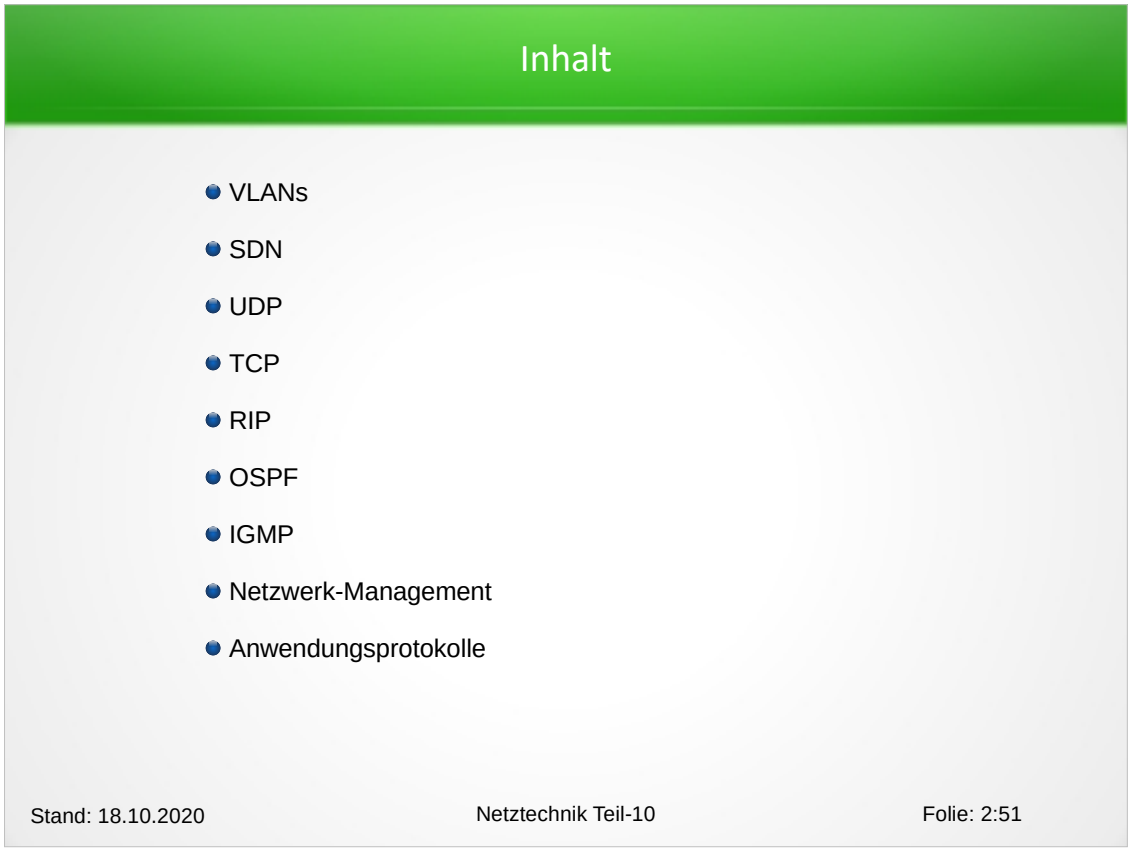


Anwendungsprotokolle
Beispiel FTP





Netztechnik Teil-10



Vor der Betrachtung der Ebenen 4 und höher sollen die Ebenen 2 und 3 im Rahmen der VLANs nochmals zusammengefasst werden.
Mit SDN wird die weitere Entwicklung der Netzwerke durch Virtualisierung beschrieben.

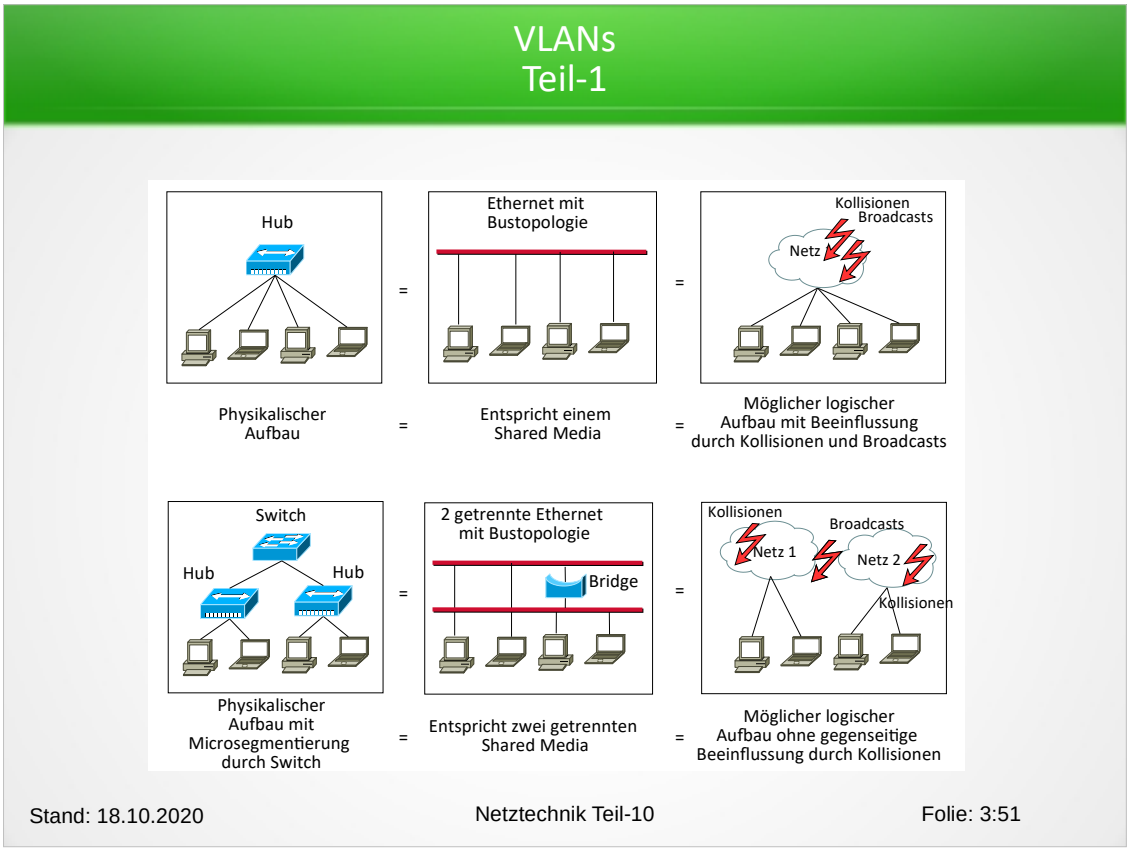
UDP und TCP sind die Protokolle der Transportebene (4)

RIP und OSPF sind Beispiele für Routingprotokolle.

IGMP beleuchtet ein Protokoll für Multicasts

Netzwerk-Management beschäftigt sich mit den Protokollen und Tools zur Verwaltung und Monitoring von Netzwerk-Geräten.

Die Anwendungsprotokolle runden den Protokollstack nach oben hin ab.

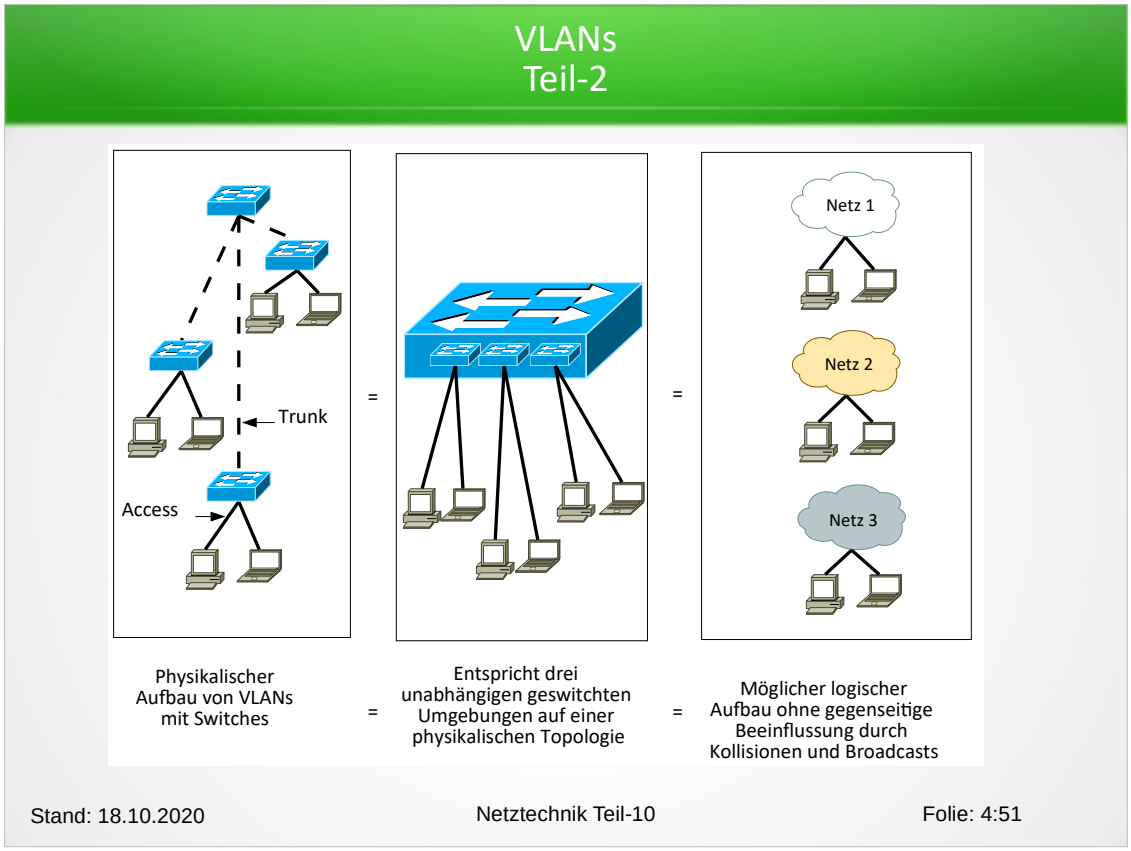


Bei der Verwendung von Hubs hängen alle angeschlossenen Geräte an einem Shared-Media in einer Collision- / Broadcast Domain mit den bekannten negativen Auswirkungen.

Werden die Hubs mit Brücken oder Switches getrennt, wird das Netzwerk in so genannten Mikrosegmente zerlegt.
Damit werden zumindest die Collision-Domains verkleinert. Die Kollisionen beschränken sich nur auf kleinere Netzwerke!

Das Broadcast-Problem bleibt bestehen.

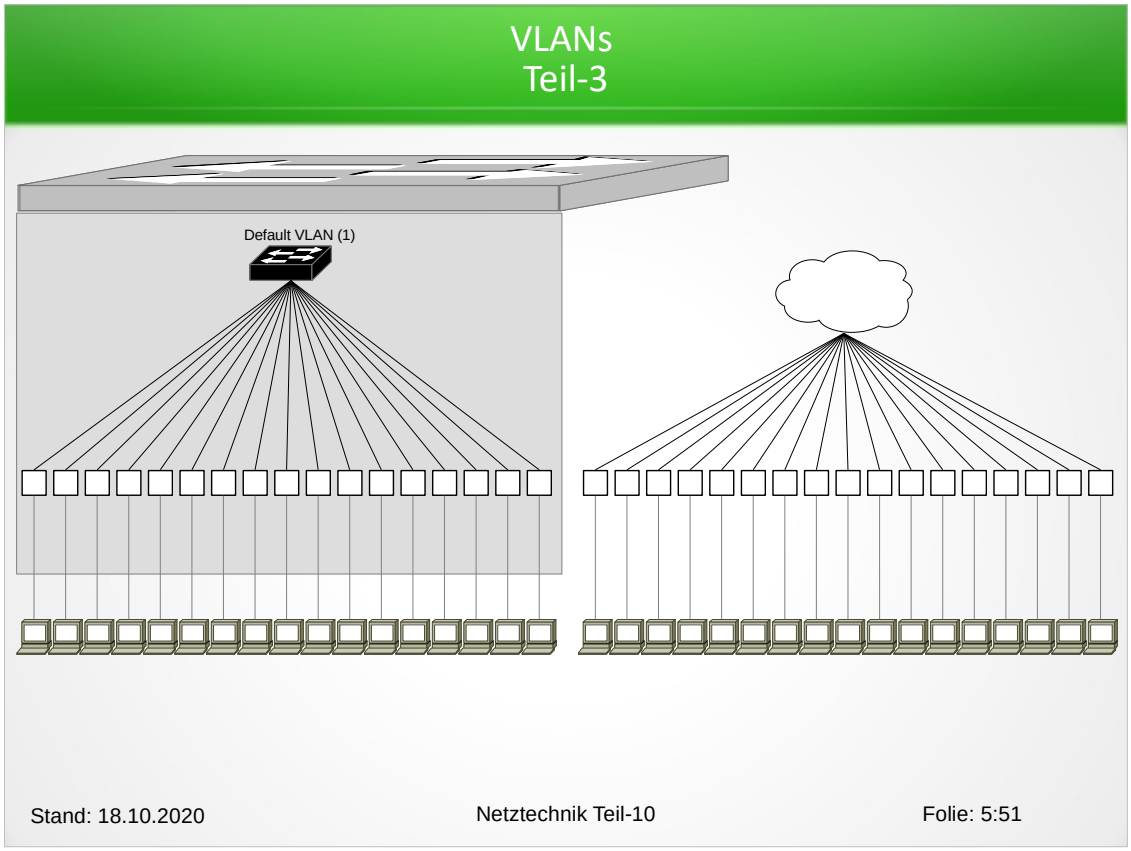
Logisch stellen alle Mikrosegmente zusammen immer noch ein Netzwerk dar. Die Hosts in allen Mikrosegmenten können sich gegenseitig sehen und adressieren.



Um die gegenseitigen Beeinflussungen zu reduzieren wurden die virtuellen LANs (VLANs) entwickelt.

Auf einer physikalischen Infrastruktur können damit mehrere logisch voneinander getrennte Netzwerke realisiert werden.

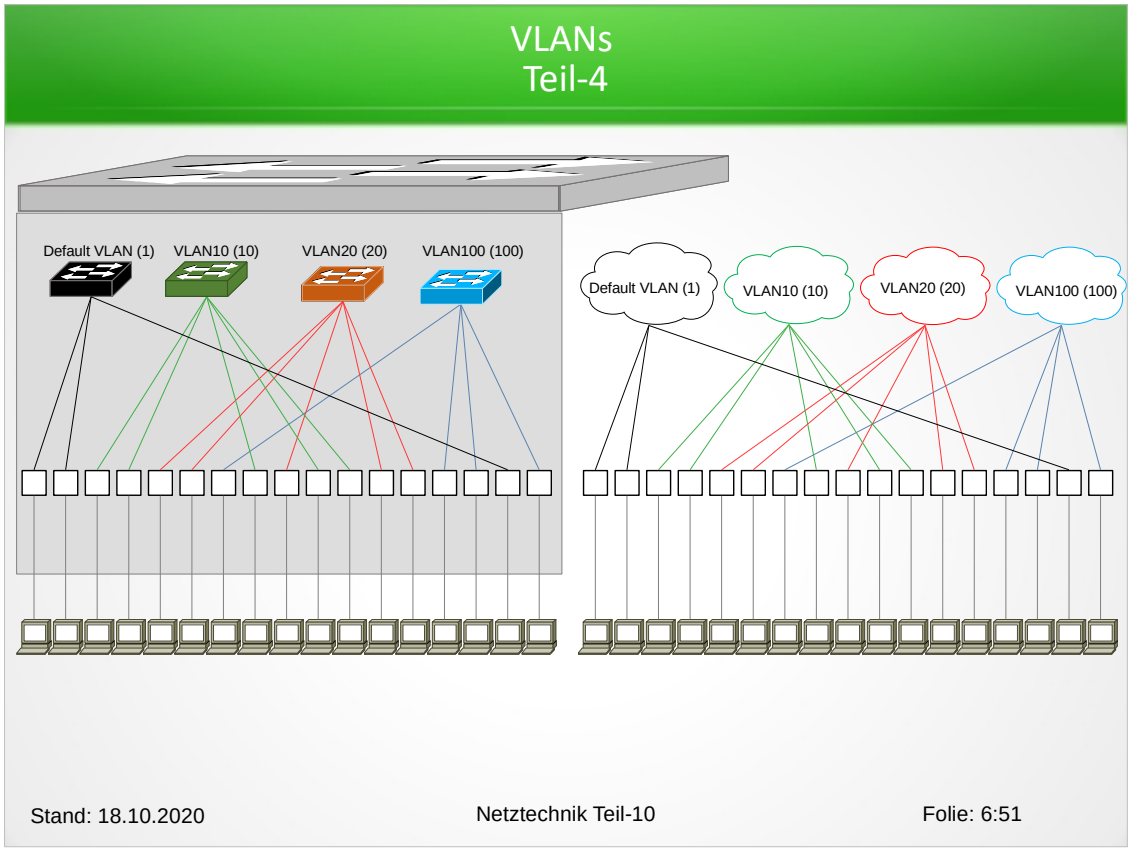
Damit wird das Broadcast-Problem reduziert. Die Host können sich nur noch innerhalb eines Netzwerks (VLANs) sehen.



Eine Bearbeitung von VLANs setzt managebare Switches voraus um die VLANS konfigurieren zu können.
Damit kommen Switches aus dem SOHO (Small Office Home Office) Bereich für den Einsatz bei VLANs nicht in Frage.

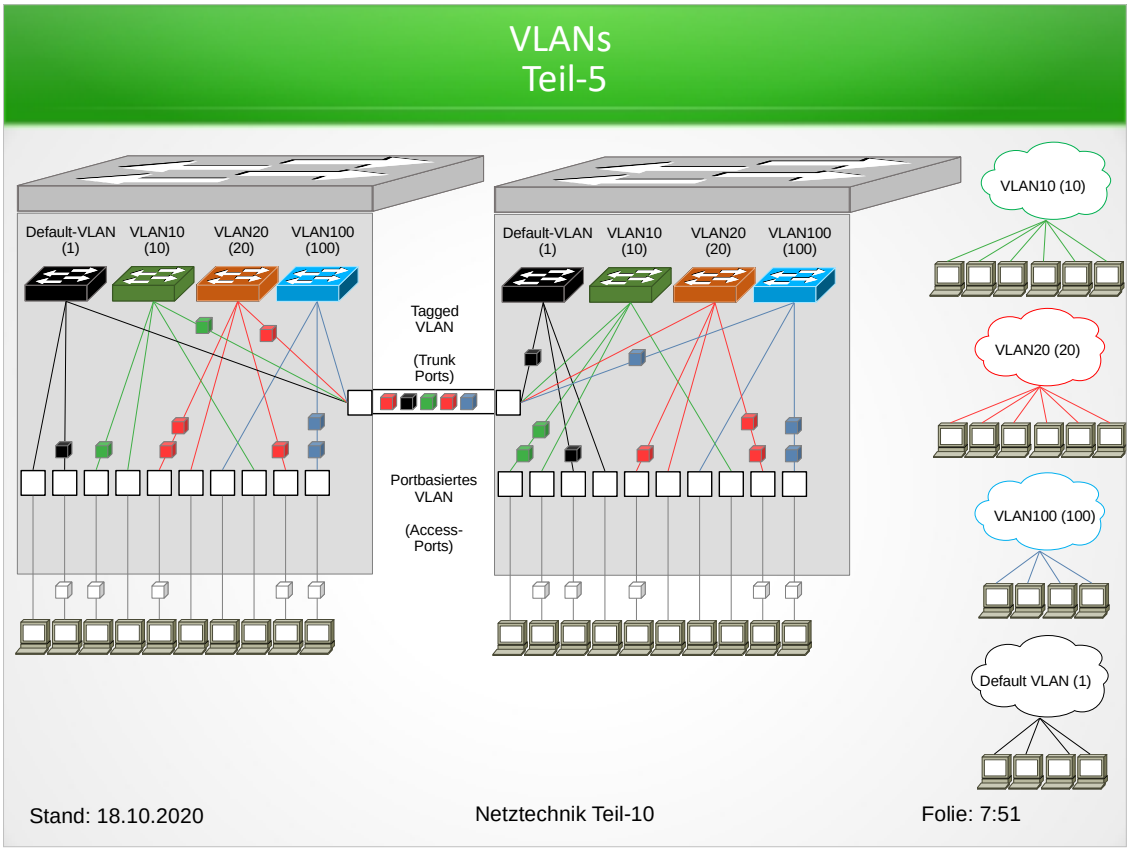
Im Auslieferungszustand sind alle Ports auf das VLAN1 (Default-VLAN) konfiguriert.

Dies entspricht einem geschlossenen Netzwerk an das alle Ports angeschlossen sind.



Konfiguriert man nun weitere VLANS (VLAN10, VLAN20 und VLAN100 dazu, können die Ports auf die VLANs verteilt/zugeordnet werden.

Damit entstehen mehrere kleine Netzwerke, die voneinander getrennt sind. Jedes dieser Netzwerke stellt eine eigene Broadcast-Domain dar.
Nur die Geräte innerhalb eines solchen Netzwerks können sich gegenseitig sehen und miteinander kommunizieren.

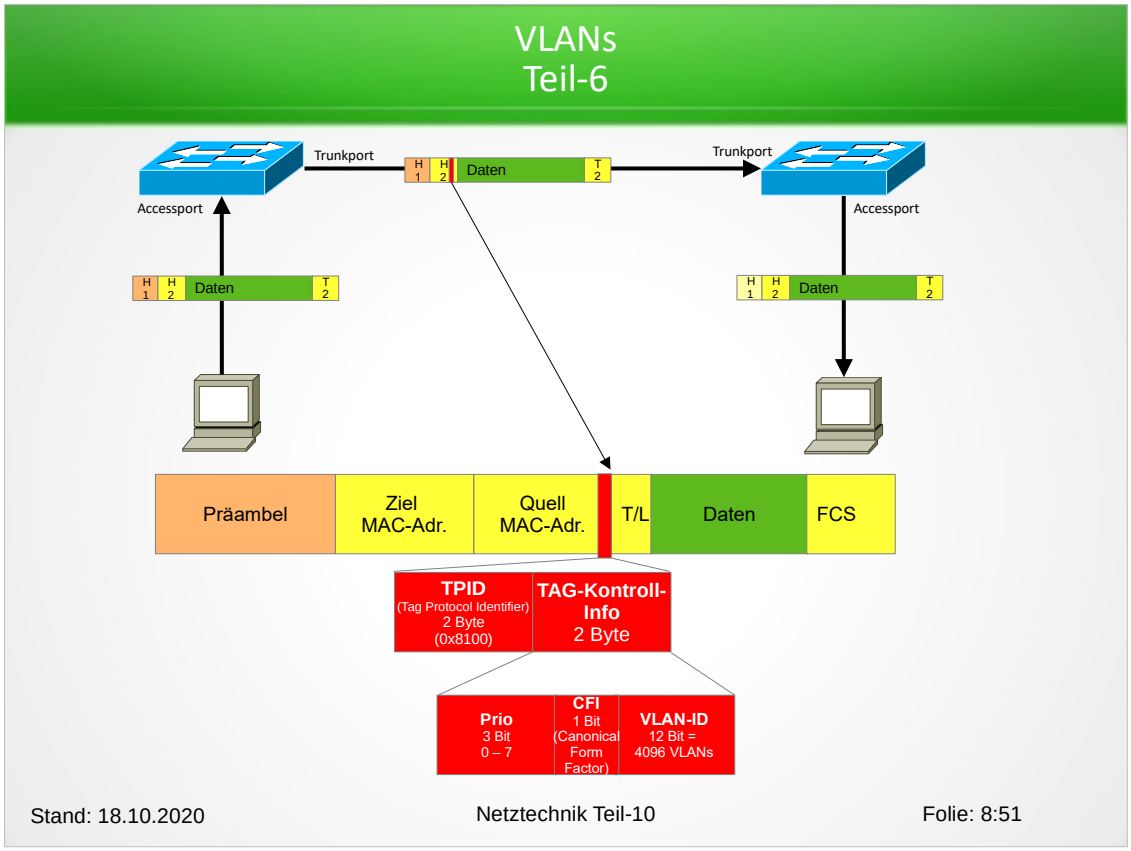


Reicht die Port-Anzahl nicht mehr aus, kann ein weiterer Switch mit den entsprechenden VLANs konfiguriert werden.

Um die beiden Switches miteinander zu verbinden könnte für jedes VLAN ein Port für die Verbindung zum Nachbarswitch konfiguriert werden.
Es geht aber noch einfacher. Man kann mehrere/alle VLANs an einem Trunk-Port zusammenfassen. Über den Trunk können gleichzeitig mehrere/alle VLANs transportieren werden.

An den Access-Ports hängen Geräte die keine Information zu den VLANs haben.

Sobald ein Paket an einem Access-Port ankommt, wird es um die VLAN-Information ergänzt (eingefärbt) und kann damit über einen Trunk transportiert werden. Verlässt das Paket den letzten Switch, wird am Access-Port die VLAN-Information wieder entfernt. Damit haben die Endgeräte keine Informationen über die VLANs in denen ihre Daten transportiert werden.



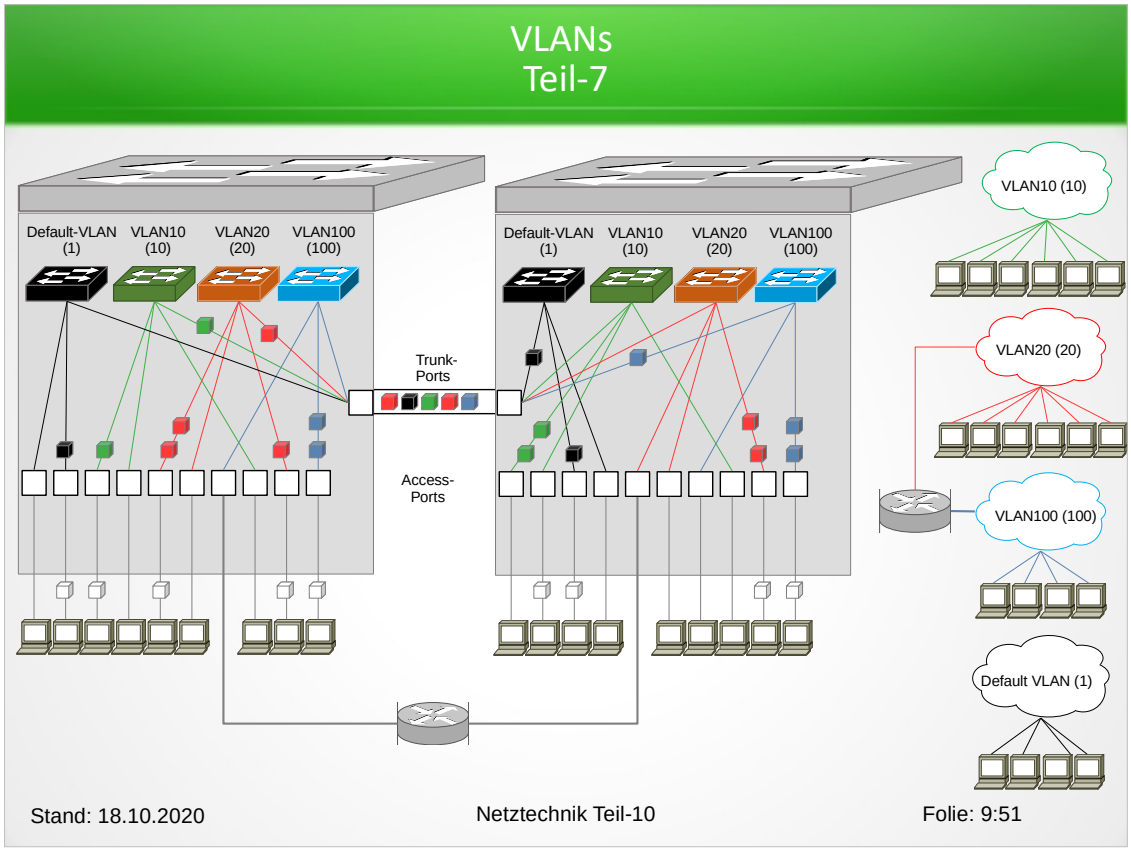
Am Accessport wird einem Frame der Tag eingefügt.

Im Tag ist zuerst der spezielle Typ-Code (0x8100) hinterlegt. Damit ist das Paket um 4 Bytes erweitert worden und getaggt.

Die Priorität ermöglicht es Pakete eines bestimmten VLANs zu bevorzugen.

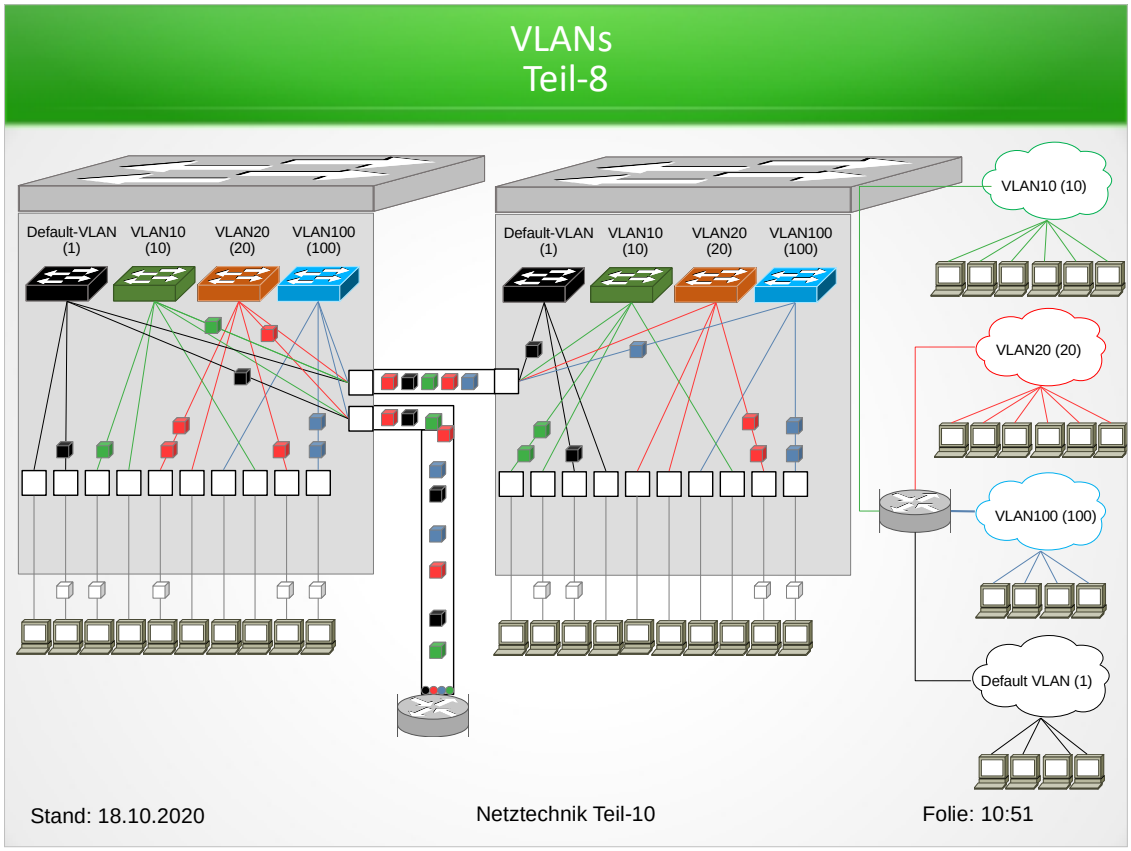
Die CFI klärt über die Darstellungsform auf.

Die VLAN-ID klärt auf zu welchem VLAN das Paket gehört. Aufgrund der 12 Bits sind maximal 4096 VLANs unterscheidbar. Oft können die Switches nur 2048 VLANs verwalten. Das ist natürlich eine Resourcenfrage. So kann z.B. jedes VLAN einen eigenen Spanning-Tree zu verwalten.



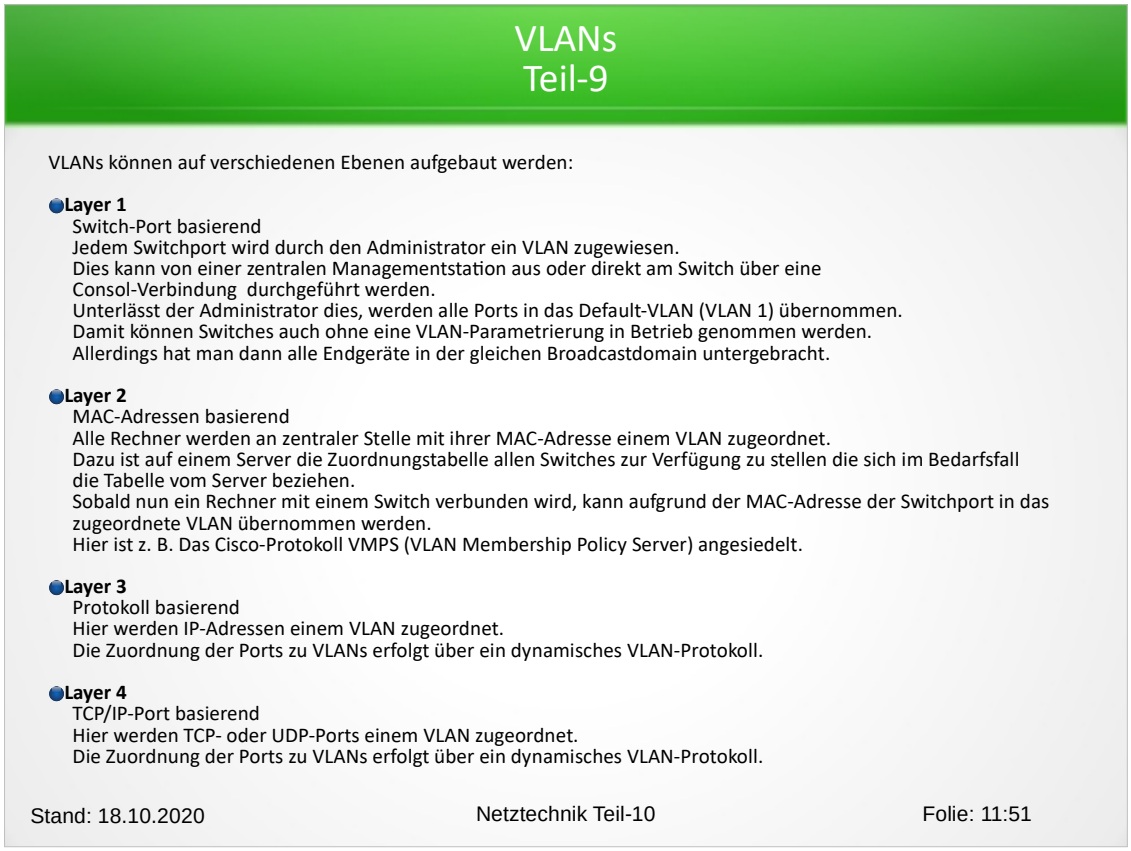
Sollen die Geräte aus VLAN20 mit den Geräten aus VLAN100 miteinander kommunizieren können, muss eine Routing-Instanz die beiden bisher unabhängigen Netzwerke miteinander verbinden.

Dies kann entweder über dedizierte Ports auf welche die VLANs konfiguriert wurden (siehe oben), oder mit einem Trunkport erfolgen.



Bei Verwendung von einem Trunk wird nur ein Port benötigt. Am Router ist dann für jedes VLAN ein Subinterface zu Konfigurieren.

Bei Trunks können die VLANs dediziert zugelassen oder ausgeschlossen werden



VLANs sind auf unterschiedlichen Ebenen des ISO-7-Schichten-Modells möglich.

Ebene1:

Jeder einzelne physikalische Port kann einem beliebigen bereits angelegten VLAN zugeordnet werden.

Ebene2:

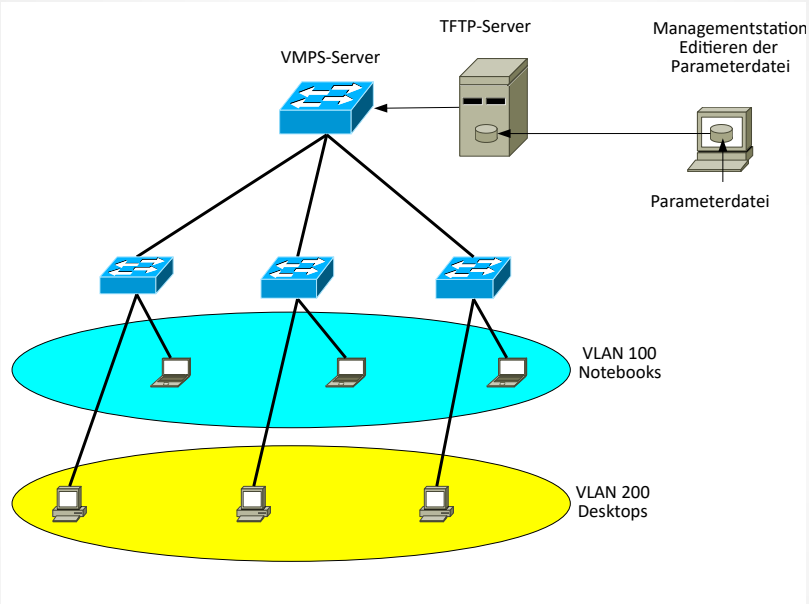
Gibt es zentrale Zuordnung der Geräte-MAC-Adressen zu VLANs können die Ports, aufgrund der MAC-Adresse des angeschlossenen Geräts, einem VLAN zugeordnet werden. Dies setzt allerdings voraus dass an einem solchen Port nur ein Gerät also auch nur eine MAC-Adresse angeschlossen ist. Eine Erweiterung mit einem Switch ist an einem Access-Port deshalb nicht zulässig!

Soll ein Switch mit einem zusätzlichen Switch erweitert werden, muss der Port, an dem erweitert wird, ein Trunk-Port sein!

Ebene3:

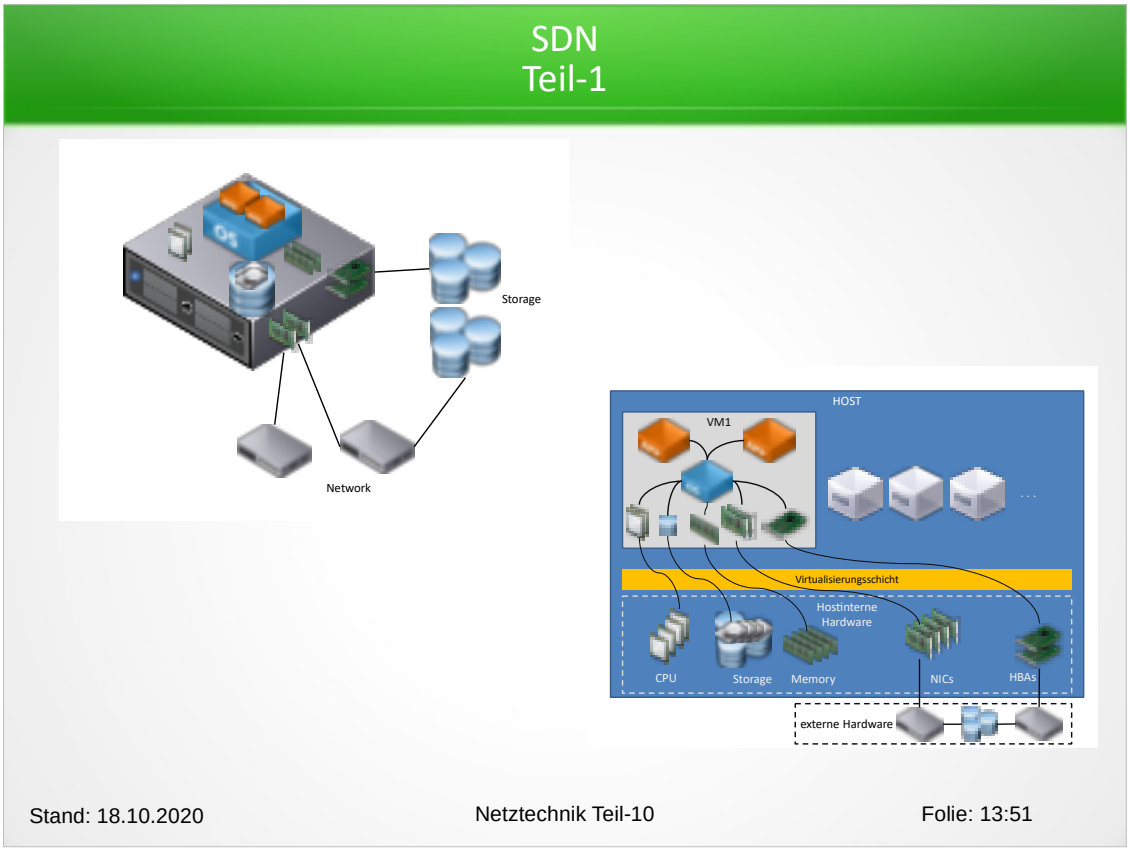
Wenn die Geräte bereits eine IP-Adresse zugeordnet bekommen haben, können sie automatisch beim Verbinden an einem Switchport einem VLAN zugeordnet und durchgeschaltet werden.

VLANs
Teil-10



Ein VMPS-Server, der auf einem zentralen Switch angelegt sein kann, wird die Zuordnung von MAC-Adresse zu VLAN verwaltet. Die Konfiguration kann von einem zentral abgelegten File per TFTP geladen werden.

Die Access-Port-Switches holen sich beim zentralen Switch im Bedarfsfall die Zuordnungen ab.



In klassischen Servern sind die Hardware-Komponenten Motherboard, CPU, Speicher, SSDs und Festplatten sind in einem Gehäuse verbaut.

Extern sind Netzwerke und Storage-Systeme über die entsprechenden Controller (NICs und HBAs) anschließbar

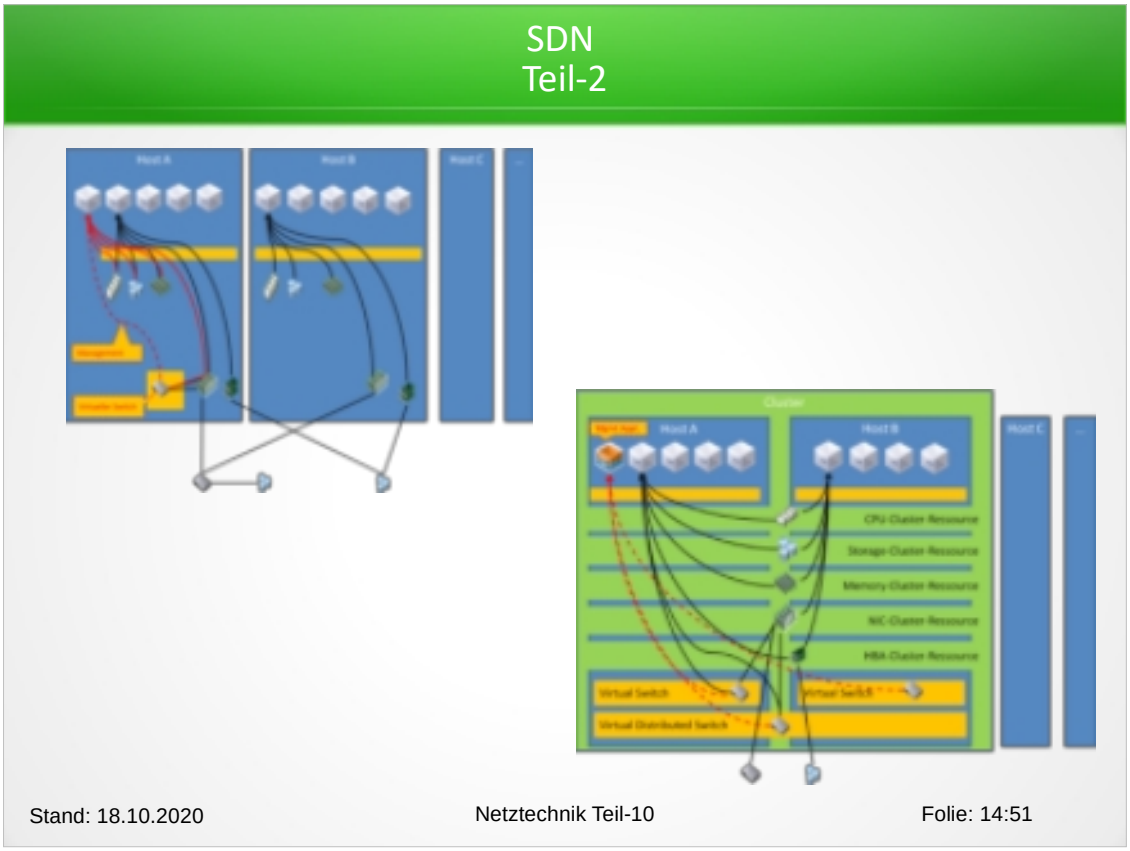
Die Applikationen sind über das Betriebssystem mit den Hardwarekomponenten und den externen Systemen verbunden

Durch Virtualisierung können auf einem Host-System unterschiedliche Mengen der vorhandenen Host internen Hardware-Komponenten einer virtuellen Maschine (VM) zugewiesen werden.

Die Applikationen können über das Betriebssystem (OS) auf die Komponenten der VM zugreifen.

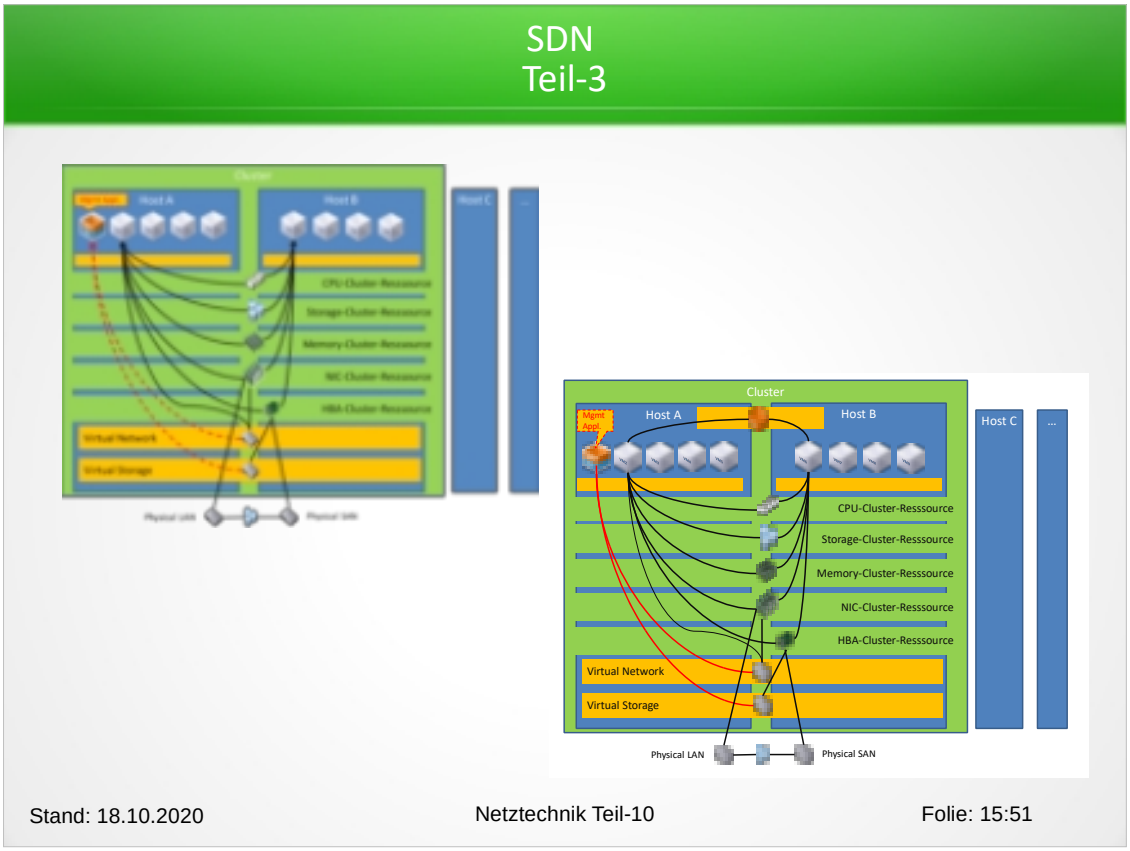
Aus Sicht der Applikationen verhält sich das OS auf der virtuellen Maschine wie auf einer physikalischen Maschine.

Die Verbindung zur Außenwelt wird mittels der NICs und HBAs (die zugewiesen wurden) hergestellt.



Die Virtualisierung kann Funktionen von externen Komponenten wie z. B. Switches auch in virtueller Form auf einem Host zur Verfügung stellen. Damit ist es möglich auf einem Host virtuelle Switches zu definieren, die den unterschiedlichen VMs unterschiedliche VLANs zur Verfügung stellen können. Die Verbindungen nach Außen werden über NICs und HBAs ermöglicht.

Mehrere Hosts können zu einem Cluster zusammengeschaltet werden. Dadurch erhöht sich sowohl die Anzahl der Ressourcen als auch die Verfügbarkeit (z. B. im Falle eines Netzteil- oder Motherboard-Ausfalls) Die Switches können auf einen Host begrenzt sein (Virtual Standard Switch (VSS)), oder über mehrere Hosts hinweg (Virtual Distributed Switch (VDS)) die VLANs den VMs zur Verfügung stellen.



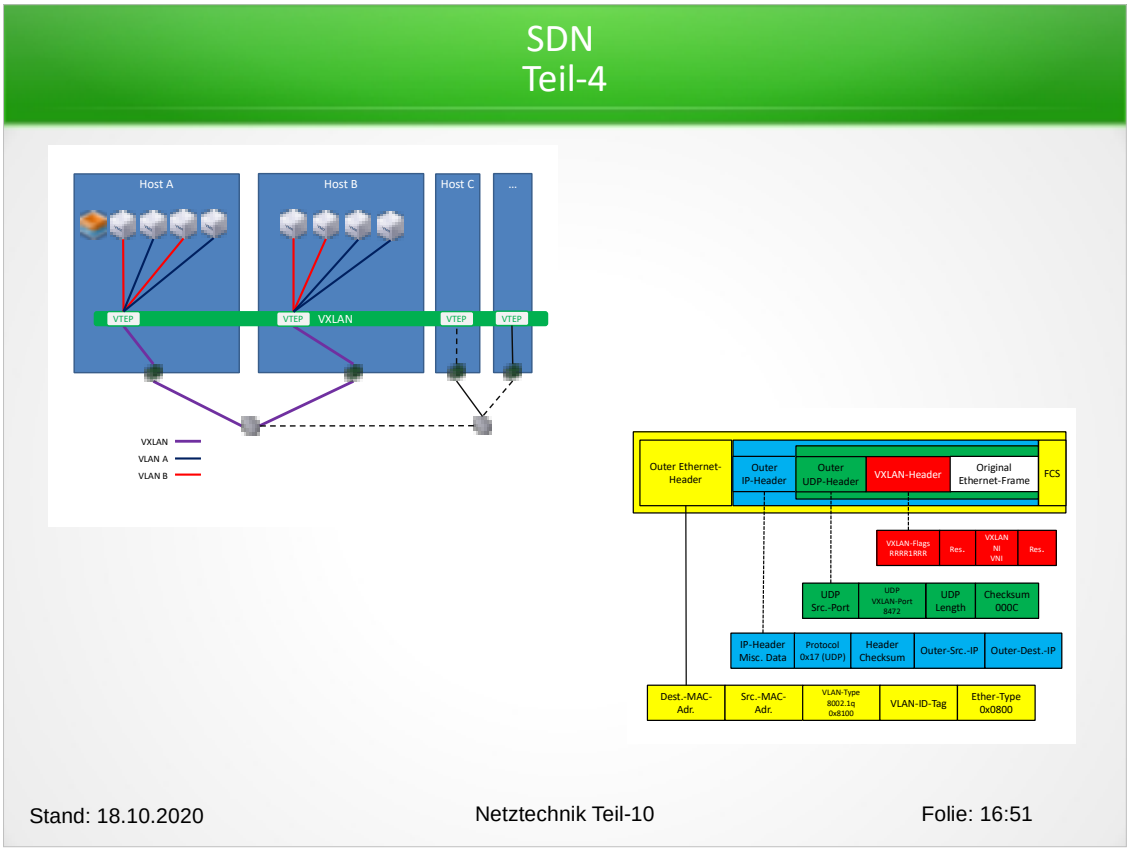
Die Verwaltung der virtuellen Switches kann mittels einer Applikation auf einer VM bewerkstelligt werden. Voraussetzung ist, dass die Hosts über die NICs miteinander verbunden sind.

Nach der Virtualisierung des Netzwerks ist noch nicht Schluss.

So ist auch eine Virtualisierung des Storage möglich. Dabei ist es unerheblich, ob das Storage über NICs oder HBAs verbunden wird.

Damit werden die Host internen Ressourcen um die Externen Ressourcen erweitert.

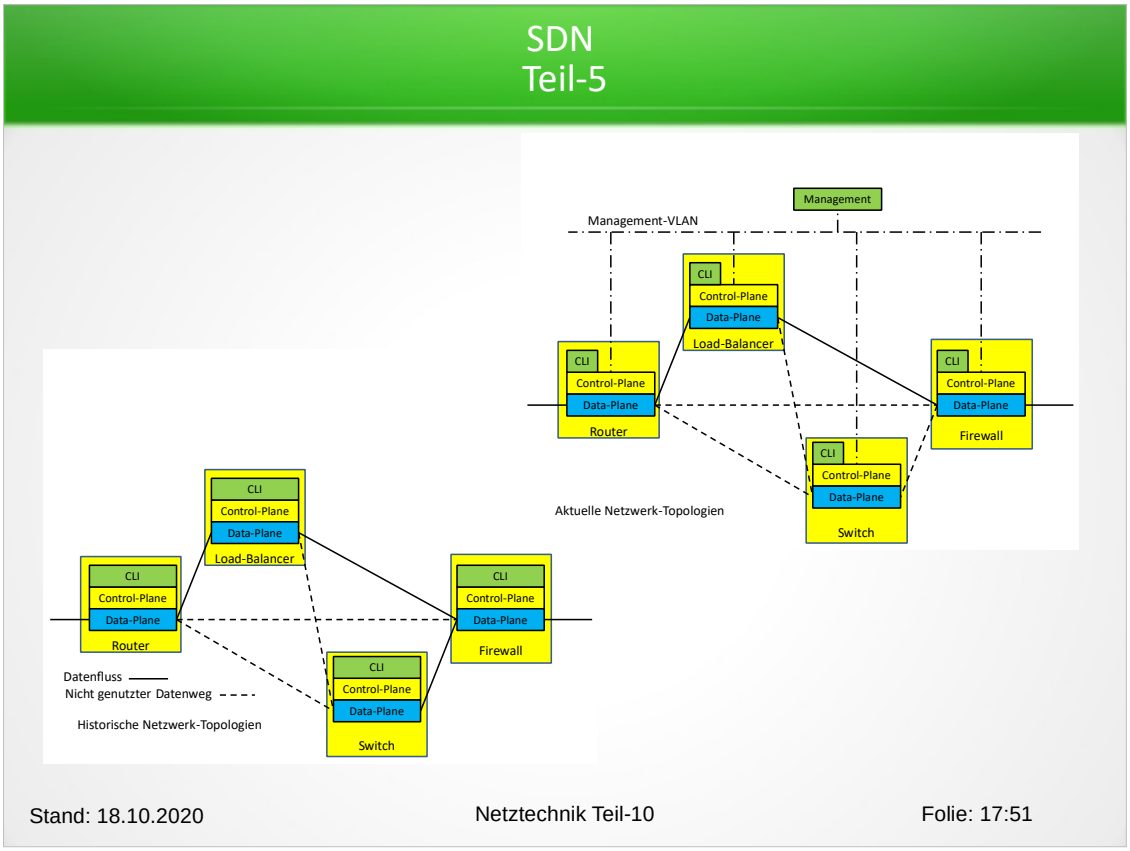
Der Aufwand wird betrieben um die vorhandenen Ressourcen besser zu nutzen und die Verfügbarkeit zu erhöhen, denn mit einer so geschaffenen Infrastruktur ist es einfacher Applikationen schnell zwischen VMs, Hosts und somit auch Standorten wechseln zu lassen.



So kann nach und nach eine Cloud aufgebaut werden.
Da nur 4096 unterschiedliche VLANs (in IEEE802.1q) möglich sind, würden z. B. Cloud-Provider schnell Grenzen stoßen.

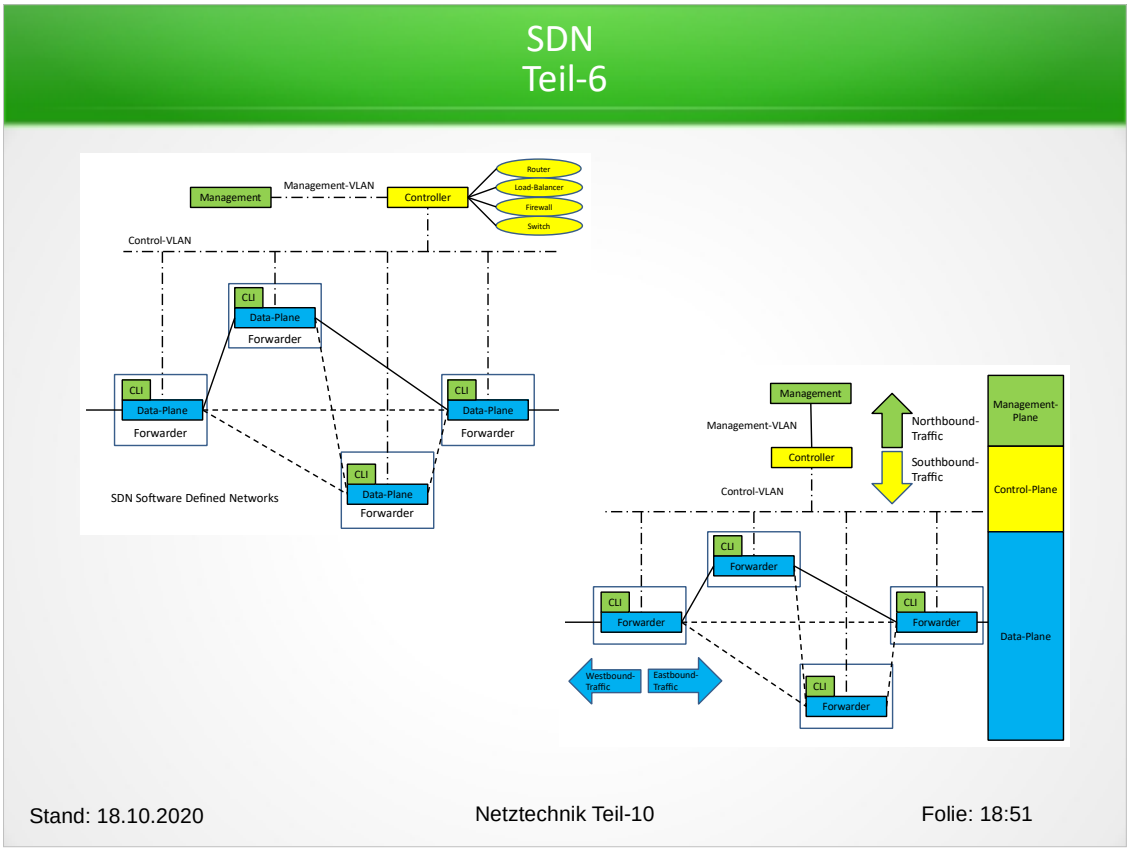
Um diese Grenze zu überwinden kommt VXLAN (Virtual Extensible VLAN) zum Einsatz.
Dabei wird ein VLAN-Rahmen in einen VXLAN-Rahmen eingepackt.
Da die VXLAN-ID mit 24 Bits mehr als 16 Millionen VXLANs ermöglicht, ist es möglich über Server- / RZ- / Standote-Grenzen hinweg L2-Netzwerke aufzuspannen.

Die Endpunkte der VXLANs bilden die VTEP (Virtual Tunnel End Point)
Hinter den VTEPs können dann die VLANs wie gewohnt genutzt werden.



Veränderung der Netzwerk-Komponenten unter SDN
Klassisch gesehen hatte die Netzwerk-Komponenten wie Switches Router usw. ein Command Line Interface (CLI) mit dem der Administrator die Konfiguration des Gerätes vornehmen konnte

Durch die Einführung von Management-Systemen konnte von einer zentralen Managementstation die Konfiguration der Netzwerk-Komponenten vorgenommen werden. Dadurch konnten Konfigurations-Stände und Betriebssystemversionen zentral verwaltet werden.
Zusätzlich konnte durch Realisierung des Management-Netzwerks als eigenständiges und damit geschütztes VLAN die Sicherheit des Managements gewährleistet werden.
Das CLI dient bei Ausfall des Managements als Fallback-Lösung.



Durch die Einführung einer Control-Plane konnte die Funktion der Steuerung eines Netzwerk-Gerätes abstrahiert , und im Controller zentralisiert werden.

Damit findet eine Dreiteilung statt.
Über die Northbound-API kann ein Management-System die Konfiguration des Netzwerks bewerkstelligen.

Die eigentlichen Funktionen finden in den Controllern statt.

Die Geräte werden zu dummen Forwardern reduziert. Jedes eintreffende Paket wird auf einen Eintrag in einer Forwarding-Liste hin überprüft. In der Liste ist der Port , an dem das Paket weiter zu leiten ist, hinterlegt.

Fehlt der Eintrag, da ein Paket für diese Verbindung noch nicht am Gerät vorbei gekommen ist, muss beim Controller über die Southbound Schnittstelle nachgefragt werden. Der aktualisiert die Forwarding-Liste des Gerätes.

Damit sind in der Data-Plane nur noch einfache Forwarder erforderlich. Der eigentliche Datenverkehr wird auch als Westbound/Eastbound-Traffic bezeichnet.

Funktionen wie Routing, Firewalling oder Load-Balancing werden im Management konfiguriert und im Controller abgebildet und verwaltet. Um die Forwarder aufzusetzen wird ein CLI noch mitgeliefert.

UDP
Einsortierung im ISO-7Schichten-Modell / Header

Dienst	Ebene
SNMP,DHCP, BOOTP, NTP, TFTP, Rservices, DNS, RPC, usw.	> 4
UDP	4
IP	3

Verbindungslose Kommunikation

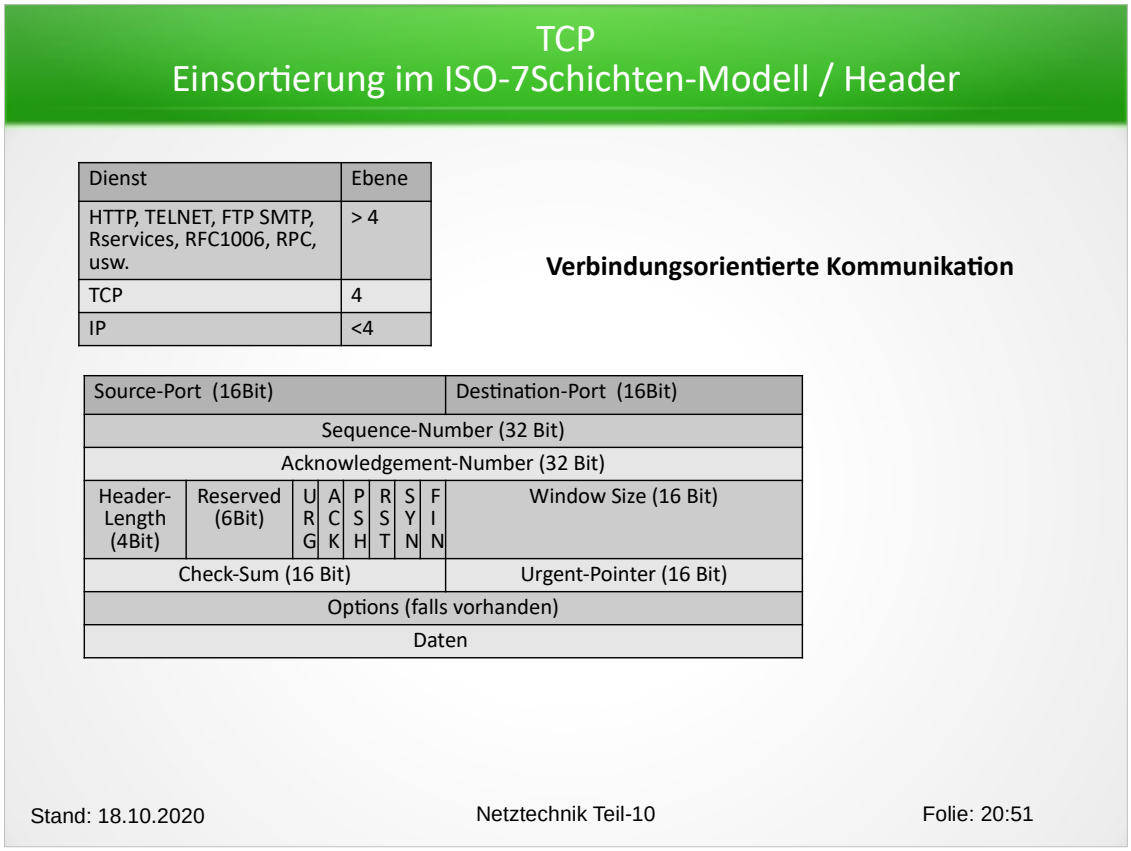
2 Bytes	2 Bytes	2 Bytes	2 Bytes
Source-Port	Destination-Port	Message-Length	Checksum

Stand: 18.10.2020 Netztechnik Teil-10 Folie: 19:51

UDP bedeutet **User Datagramm Protokoll**.
UDP ist im Internet entstanden und dort neben TCP für die Bearbeitung der Ebene-4 zuständig.

UDP wird im **RFC 768** beschrieben und arbeitet im Gegensatz zu TCP **verbindungslos**. Dies bedeutet, dass bei einem Datenaustausch die Daten auf das Netz gegeben werden ohne zu wissen, ob jemand auf die Daten wartet.
Deshalb muss auch kein Verbindungsaufbau und Verbindungsabbau stattfinden. Dies vereinfacht und beschleunigt die Datenübertragung erheblich. Deshalb verwenden Datenbanken zur Datenübertragung UDP. Allerdings gibt UDP keine Gewähr, ob die Daten auch beim Empfänger ankommen. Darum müssen sich bei Verwendung von UDP die höheren Schichten kümmern.
UDP stellt eine ungesicherte Verbindung dar. Übertragene Daten werden nicht quittiert.

Deshalb werden sie auch bei **Multicasts** und **Broadcasts** verwendet



Source-Port
Portnummer der Quelle
Destination-Port
Portnummer des Ziels
Sequence –Number
Sequenznummer des Datenpakets (entspricht der Anzahl der bisher gesendeten Daten + Initialisierungs-Sequenznummer)
Acknowledgement-Number (ACK)
Quittung für die zuvor gesendete Sequenznummer. Die ACK-Sequenz-Nummer gibt dem Partner an, ab welcher Position im Datenstrom die nächste Sequenz erwartet wird.
Header-Length (4 Bit)
FLAGS
Normalerweise ist nur ein (maximal 2)Flag(s) gesetzt. Pakete, bei denen alle Flags gesetzt sind, werden Kamikaze-Paket, nastygram, Christmas tree packet oder Lamptest genannt. Wird zum Test von TCP-Stacks verwendet. Siehe hierzu RFC1025 (TCP and IP Bake Off)
Im tcpdump-Output werden die Flags durch einen Buchstaben gekennzeichnet:
U=URG / S=SYN / F=FIN / R=RST / P=PSH / .=Kein Flag gesetzt
Window-Size (16 Bit)
Deutsch: Fenster-Größe. Damit teilt ein Kommunikationspartner seinem Gegenüber mit, wie viel Platz in seinem Empfangs-Puffer noch frei ist. Diese Datenmenge kann in den nächsten Segmenten maximal übertragen werden, ohne dass eine Quittung erforderlich ist.
Checksum (16 Bit)
Prüfsumme
Urgent-Pointer (16 Bit)
Zeiger auf das Ende des dringlich zu behandelnden Datenteils.
Options
Im ursprünglichen TCP-RFC 793 werden nur 3 Optionen definiert:
End of option list / No option / MSS
Diese wurden im RFC 1323 um 2 Optionen erweitert:
Window Scale Factor / Timestamp

URG Flag. (1 Bit)
URG bedeutet urgent; deutsch dringend. Damit wird das Paket vom Sender als dringlich eingestuft. Der Urgent-Pointer ist damit gültig.

ACK Flag. (1 Bit)
Acknowledgement; deutsch Bestätigung. Damit wird der korrekte Datenempfang bestätigt.

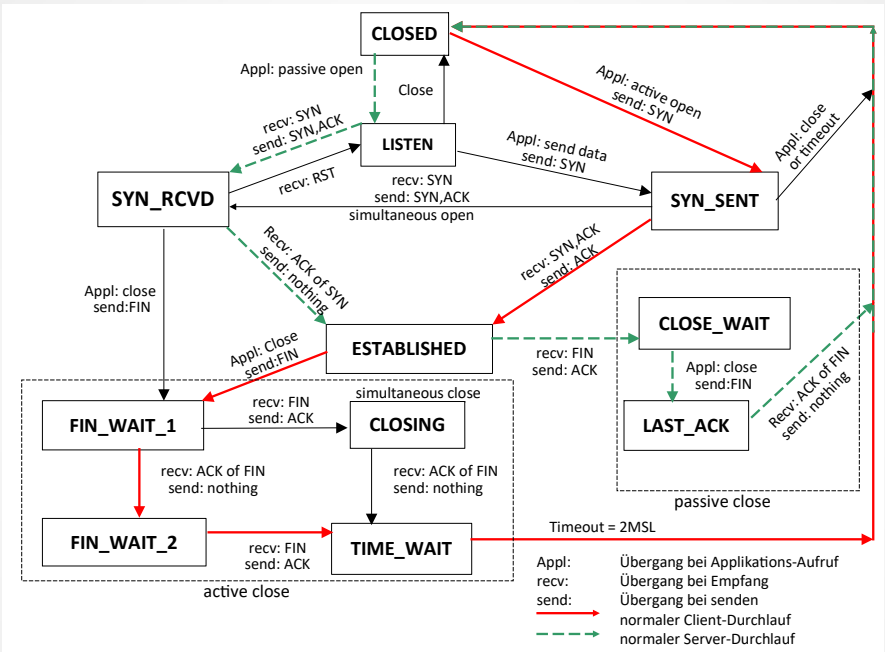
PSH Flag. (1 Bit)
PSH bedeutet push; deutsch Anstoß oder Initiative. Damit wird die empfangende TCP-Seite angewiesen, die Daten auf dem schnellsten Weg der Empfänger-Applikation zu übergeben. Das PSH-Flag wird von TCP gesetzt, wenn der Sendepuffer beim Senden geleert wurde, was meistens bei den Datenübertragungen der Fall ist. Es gibt auch TCP Implementierungen, die grundsätzlich das PSH-Flag beim Senden von Daten setzen. In modernen APIs ist es dem Programmierer nicht möglich das PSH-Flag zu setzen.

RST Flag.(1 Bit)
RST bedeutet reset; deutsch zurücksetzen. Damit wird eine Kommunikations Beziehung abgebrochen. Alle Puffer werden geleert bzw. freigegeben.

SYN Flag. (1 Bit)
SYN bedeutet synchronize; deutsch synchronisieren. Dies ist eine Verbindungsaufbau-Anforderung.

FIN Flag. (1 Bit)
FIN bedeutet finalize; deutsch beenden. Damit wird einseitig eine Kommunikations-Beziehung abgebaut.

TCP
Verbindungsstati

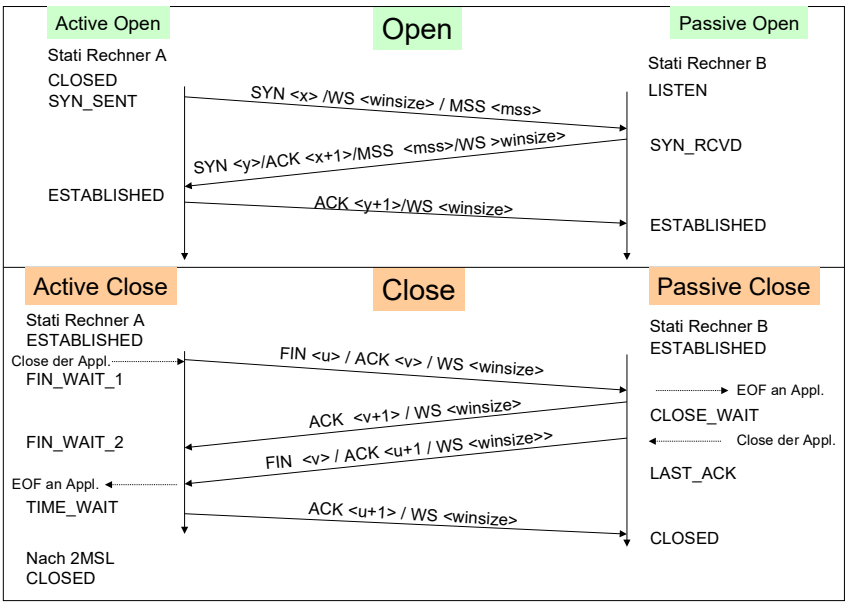


Es gibt zwei Sichten auf die Stati:

- Client-Sicht
- Server-Sicht

Beide starten und enden im Status Closed

TCP
Open / Close



TCP
RST

RST(Reset)
Wird ein SYN-Segment gesendet, ohne dass ein Port auf der anderen Seite dafür geöffnet wurde oder gleichzeitig ein SYN von der anderen Seite gesendet wird, wird der Verbindungsaufbau-Versuch zurückgewiesen.
Dies wird mit einem RST-Segment das an den SYN-Sender zurückgegeben wird durchgeführt.
Ein RST-Segment wird auch gesendet, wenn ein Segment für eine nicht etablierte Verbindung eintrifft.
Zu einer bestehenden Verbindung gehören jeweils zwei IP-Adressen und die zugehörigen Ports.
Bei UDP wird in diesem Fall eine ICMP-Meldung zurückgesendet.

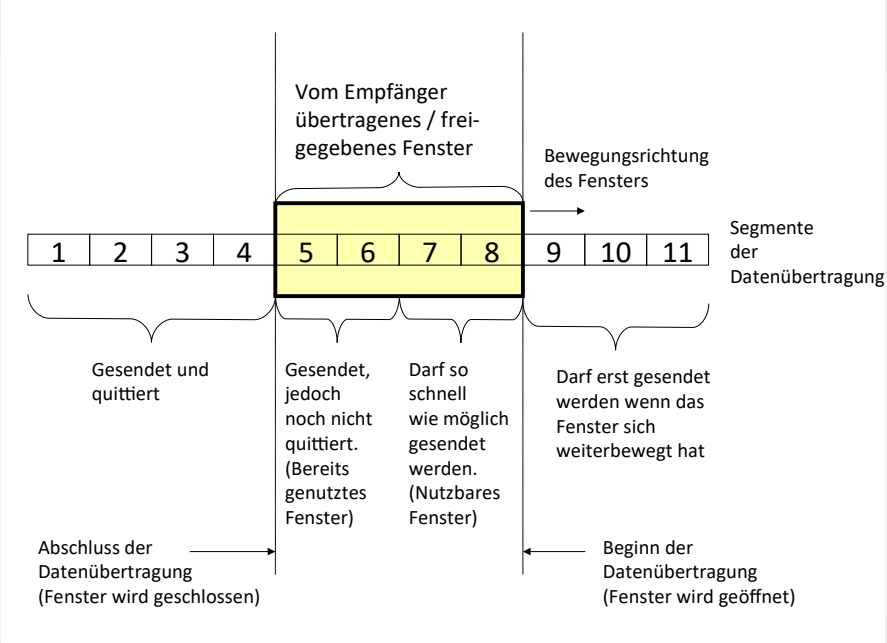
SYN SNR<x> WIN<y> MSS<z> ->
<- RST SNR 0 ACK <x+1> WIN 0

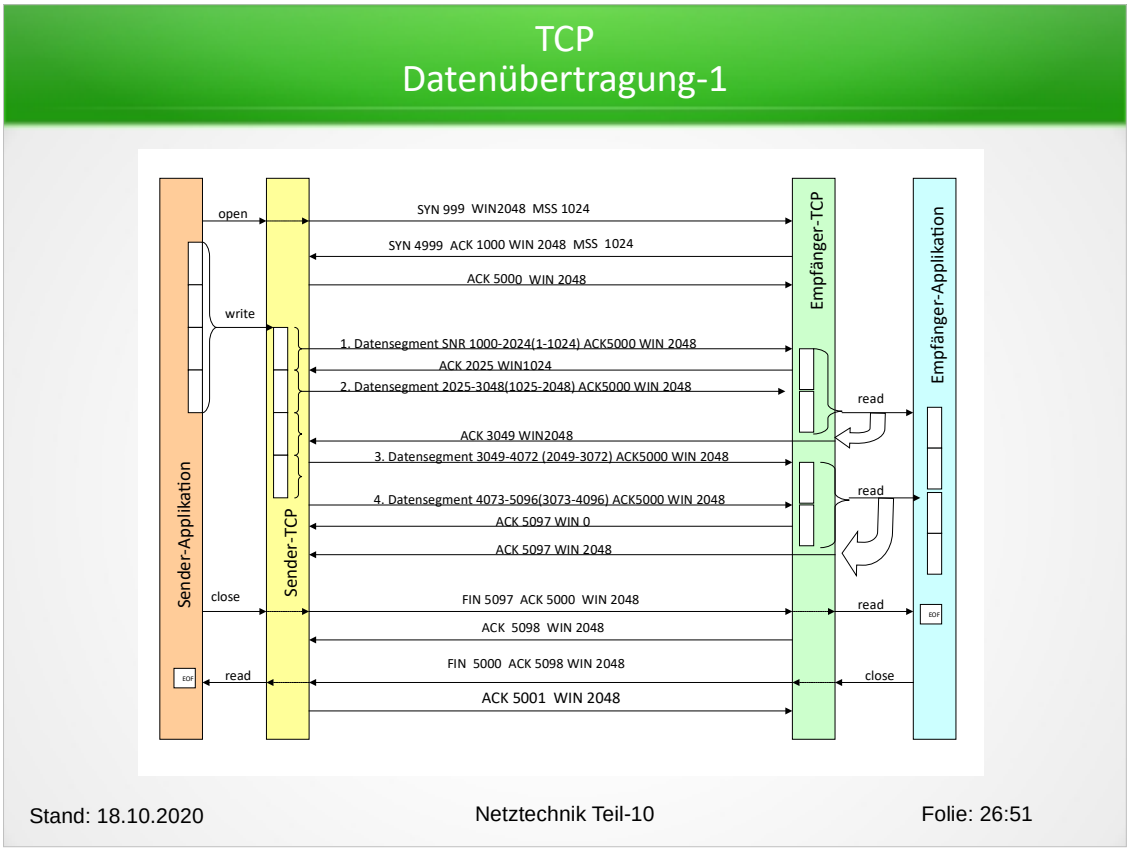
Ein RST auf ein SYN bedeutet somit, dass bei dem Empfänger entweder die IP-Adresse nicht stimmt, oder dass auf dem Port kein Partner hört.

Abbrechen einer Verbindung
Normalerweise wird eine Verbindung mit FIN ... beendet.
Dies wird orderly release genannt.

Es ist auch möglich, eine Verbindung mit einem RST –Segment anstelle einer FIN-Sequenz zu beenden.
Dies wird abortive release genannt.
Dabei werden die Daten in den Puffern verworfen und ein RST-Segment wird gesendet.
Dies kann mit der SO_LINGER - Socket-Option gemacht werden.
Dabei wird mit einer linger-time (deutsch: Verzögerungszeit) von 0 genau dies gemacht.

TCP
Sliding Window





Beim Verbindungsaufbau wird die MSS (Maximum Segment Size; deutsch: Maximale Segment-Größe) ausgehandelt, mit der Daten übertragen werden. Dieser Wert muss für beide Partner gleich sein. Bei ungleiche Vorschlägen wird Kleinste genommen. Zusätzlich teilt jeder Partner seinem Gegenüber mit, wie groß sein Empfangs-Puffer ist (Window-Size) ist. Dies kann bei jedem anders sein.

Damit ist die maximale Segment-Größe und die maximale Anzahl von Segmenten hintereinander, ohne dass eine Quittung erfolgen muss, festgelegt.

Im folgenden Beispiel gilt:
MSS = 1024
WIN=2048 für beide Seiten
Zu übertragende Datenmenge 4096 Byte
Der Start der Sequenznummern ist zufällig gewählt.

TCP
Datenübertragung-2

Segment	Inhalt/Bedeutung
1	Das erste Datensegment wird gesendet.
2	Die Daten werden von TCP mit einem ACK quittiert. Die Windowgröße wird von 2048 auf 1024 reduziert, da TCP die Daten der Applikation noch nicht weitergegeben hat.
3	Das nächste Datensegment wird gesendet. Der Empfangspuffer des Empfängers ist jetzt voll. Der Sender weiß dies und muss warten, bis der Empfänger mit einer Windowgröße > 0 einen neuen Empfangs-Puffer zur Verfügung stellt.
4	TCP übergibt die Daten der Empfänger-Applikation. Damit kann TCP den Empfangs-Puffer räumen. I ACK wird dies durch die Windowgröße 2048 mitgeteilt. Der Sender kann jetzt wieder Daten senden.
5	Das 3. Datensegment wird übertragen.
6	Da noch Platz im Empfangs-Puffer sein muss, kann jetzt gleich das 4. Datensegment gesendet werden. Jetzt sind keine weiteren Daten mehr übertragbar, der Empfangs-Puffer ist voll.
7	TCP auf der Empfängerseite teilt dem TCP auf der Sender-Seite mit, dass die Daten einschließlich dem 4. Segment empfangen wurden und im Empfangs-Puffer stehen. Da dort kein Platz mehr für weitere Daten ist, wird die Windowgröße 0 in der Quittung mitgeteilt.
8	Die Daten werden der Empfänger-Applikation übergeben und somit ist der TCP-Empfangs-Puffer räumbar. Sobald der Puffer geräumt ist, wird dem Sender mitgeteilt, dass wieder ein Empfangs-Puffer > 0 zur Verfügung steht. Dies wird in einem so genannten Window-Update – Segment gemacht
9	Da die Sender-Applikation keine Daten mehr zu senden hat, durchläuft sie einen Close-Aufruf. TCP sendet daraufhin das FIN-Segment. Der Sender ist daraufhin im FIN_WAIT_1-Status
10	Das FIN-Segment wird vom Empfänger-TCP mit einem ACK bestätigt. Somit ist der Empfänger nun im CLOSE_WAIT-Status. Der Sender ist daraufhin im FIN_WAIT_2-Status. EOF wird an die Empfänger-Applikation übergeben, die daraufhin ebenfalls einen Close-Aufruf durchläuft.
11	Der Close-Aufruf auf der Empfängerseite erzeugt nun ein FIN-Segment. Der Empfänger ist im LAST_ACK-Status.
12	Der Sender-TCP quittiert den Empfang des FIN mit einem ACK und ist nun im TIME_WAIT-Status. Dieser geht nach 2MSL in den CLOSED-Status über. Nach dem Empfang des ACK ist die Empfänger-Verbindung im Status CLOSED

TCP
Bandwidth-Delay-Produkt

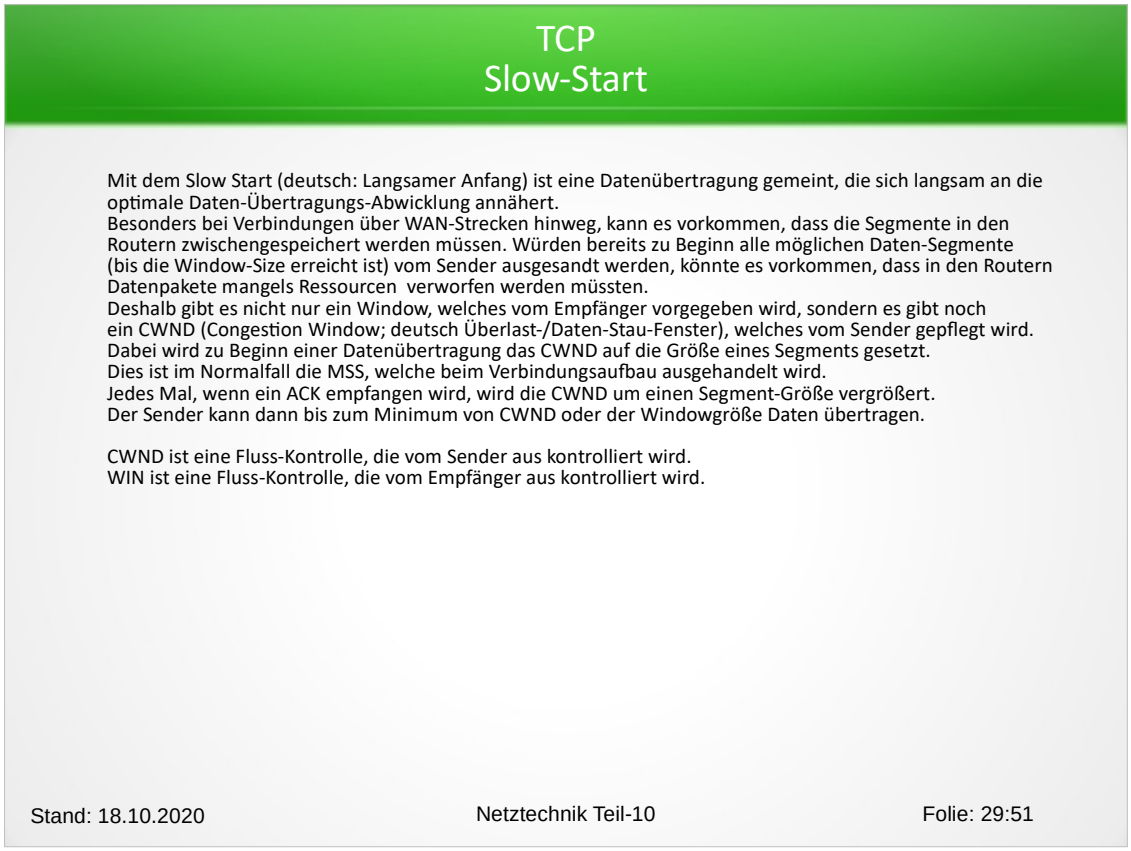
Bandwidth-Delay-Product

Um nun die optimale Fenster-Größe, auch Verbindungs-Kapazität genannt, zu ermitteln, kann man das Bandwidth-Delay-Product (deutsch: Bandbreiten-Verzögerungszeit-Produkt) anwenden. Dies geschieht folgendermaßen:

Verbindungs-Kapazität [Bytes] = Bandbreite [Bits/Sec] * RTT [Sec]) / 8 [Bits /Byte]

Beispiel:
10Mbps Datenverbindung
5ms RTT

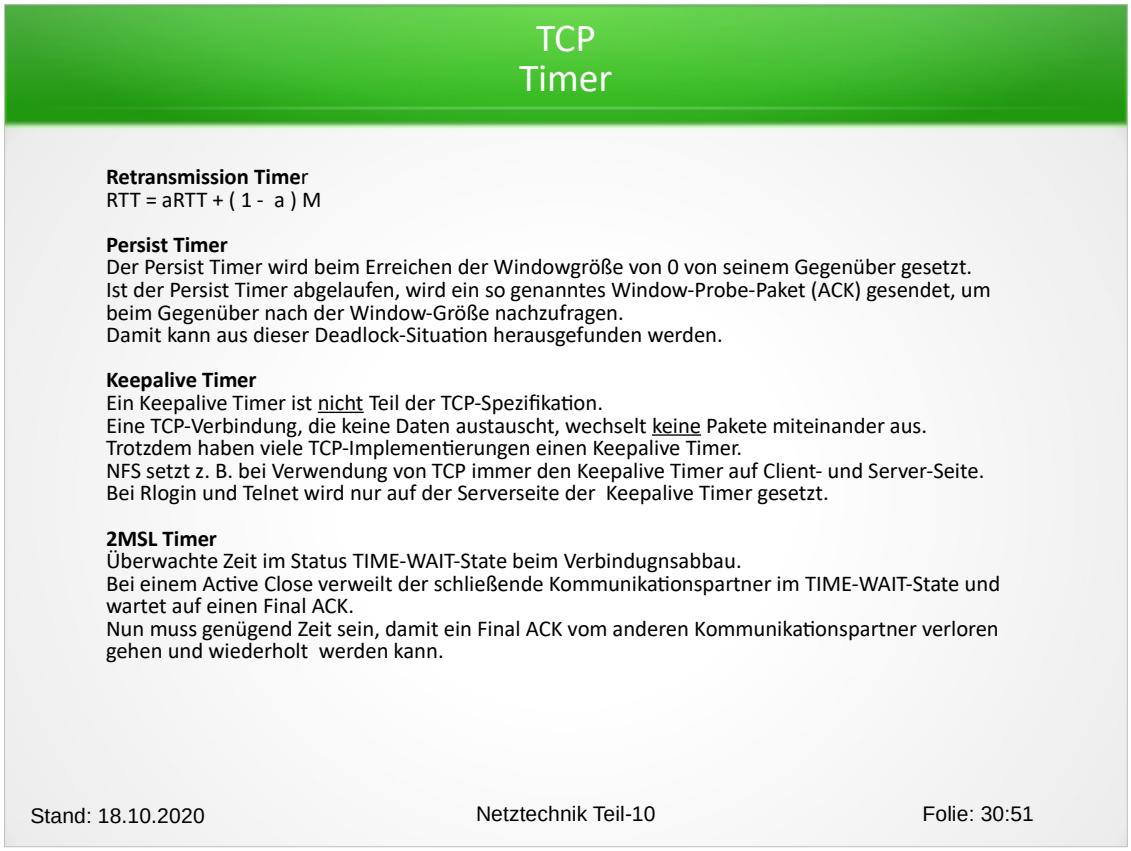
Verbindungskapazität = (10.000.000 * 0.005) / 8 = 6250 Bytes



Damit wird beim Senden zuerst ein Daten-Segment gesendet. Sobald der zugehörige ACK eingegangen ist, werden zwei Daten-Segmente hintereinander gesendet und auf ein ACK gewartet. Sobald auch hier der zugehörige ACK eingegangen ist, werden 3 Daten-Segmente hintereinander gesendet.....

Es wird so lange gesendet, bis das Minimum von CWND oder WIN erreicht ist. Sobald der CWND-Wert die maximale Windowgröße erreicht hat, gilt nur noch die Windowgröße, da der CWND-Wert immer weiter hoch gezählt wird, jedoch das Minimum der beiden Werte relevant ist. In diesem Zustand bleibt die Datenübertragung so lange, wie es zu keinen weiteren Beeinträchtigungen kommt.

Anmerkung :
Die ICMP-Meldung Source-Quench (Überlastung bei einem Empfänger/Router) führt beim Sender dazu, dass der CWND-Wert auf 1 zurückgesetzt wird. Dies ist der Anfang eines neuen Slow-Start. Damit kann eine Überlastungsmeldung zu einer Entlastung der Datenübertragung führen.



Steht für die Zeitspanne, innerhalb der ein Datenpaket quittiert werden muss. Läuft der Timer ab, ohne, dass eine Quittung dafür empfangen wurde, wird das Senden der Daten wiederholt. Bei allen weiteren Versuchen wird die Wartezeit dazwischen verdoppelt. (exponentieller backoff)

Nach 12 Versuchen wird die Verbindung mit RST abgebaut.

Grundlegend für den Timeout und die wiederholten Versuche ist die RTT (round trip time; deutsch: Zeit für den Hin- und Rückweg für Datensegment und Quittung) Da sich die Wege zwischen zwei Endgeräten ändern können (Wechsel von Routen zum Ziel), kann sich die Zeit für eine Quittung für ein Datensegment ändern. Deshalb hat TCP für die Ermittlung einen Anpassungs-Mechanismus vorgesehen. Die Anpassung wird allerdings langsam, bzw. geglättet vorgenommen. Der angeglichene, neu ermittelte RTT wird auf einem Tiefpass-Filter erzeugt.

$$RTT = aRTT + (1 - a) M$$

Der Glättungs-Faktor a wird mit 0,9 empfohlen. Damit wird der neue RTT zu 90% aus dem bisherigen RTT und zu 10% aus dem neu ermittelten (gemessenen) M zusammengesetzt.

Der RFC793 empfiehlt daraus einen RTO (Retransmission Timeout; deutsch Sendewiederholungs-Überwachungszeit) von

$$RTO = RTT + Rb * RTT$$

Rb ist der Verzögerungs-Faktor. Er wird mit einem Wert von 4 empfohlen.

Anmerkung :
Die ICMP-Meldung host unreachable oder network unreachable wird von TCP ignoriert. Beide Meldungen können auftreten, wenn ein Router ausfällt und die Verbindung neu aufgebaut werden muss. TCP versucht durch Wiederholungen (engl: retransmissions) bis zu 9 Minuten lang die Daten doch noch zu übertragen. Hier sind je nach Implementierung unterschiedliche Werte anzutreffen.

Routing-Protokolle
RIPv1

1. Byte	2. Byte	3. Byte	4.Byte
Comand	Version	0	
Family of Net 1		0	
IP-Address of Net 1			
0			
0			
Distancte to Net 1 (Hops)			
Family of Net 1		0	
IP-Address of Net 2			
0			
0			
Distancte to Net 2 (Hops)			
...			

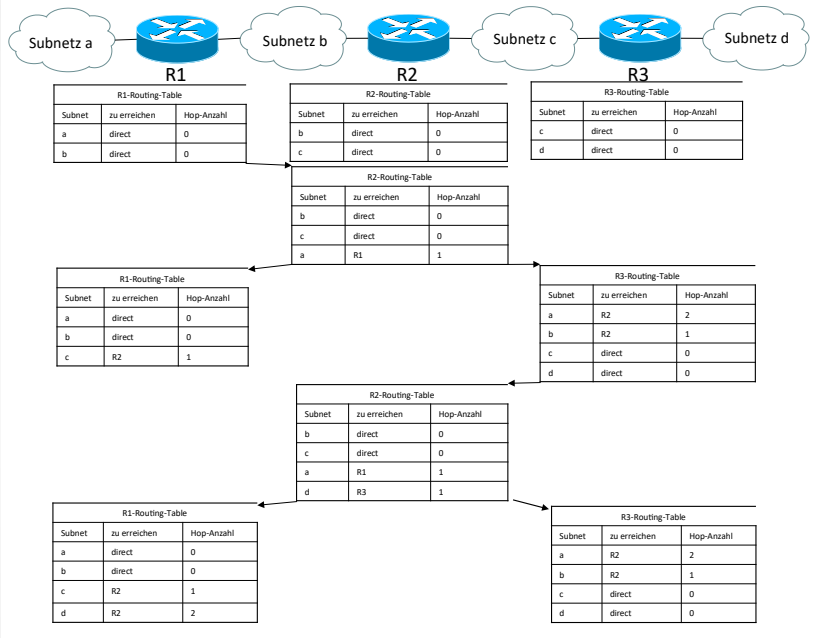
Erster Versuch des Routing Information Protocol (RIP)

RIP ist ein Interior Gateway Protokollen, da es im LAN-Umfeld agiert.
Zustandsabhängig
Als Metrik wird die Anzahl der Hops
(Anzahl der Router bis zum Ziel) verwendet.
→ Distance Vector Routingprotokoll

Dabei handelt es sich um einen Bellman-Ford-Algorithnmus

Wird als Broadcast gesendet. (→ betrifft alle Netzwerk-Teilnehmer!)
Kein Subnetting, da keine Subnet-Mask-Informationen
Maximal 14 Hops (15 Hops gilt als nicht erreichbar)
Updates sind Timer-gesteuert → langsam bei Änderungen

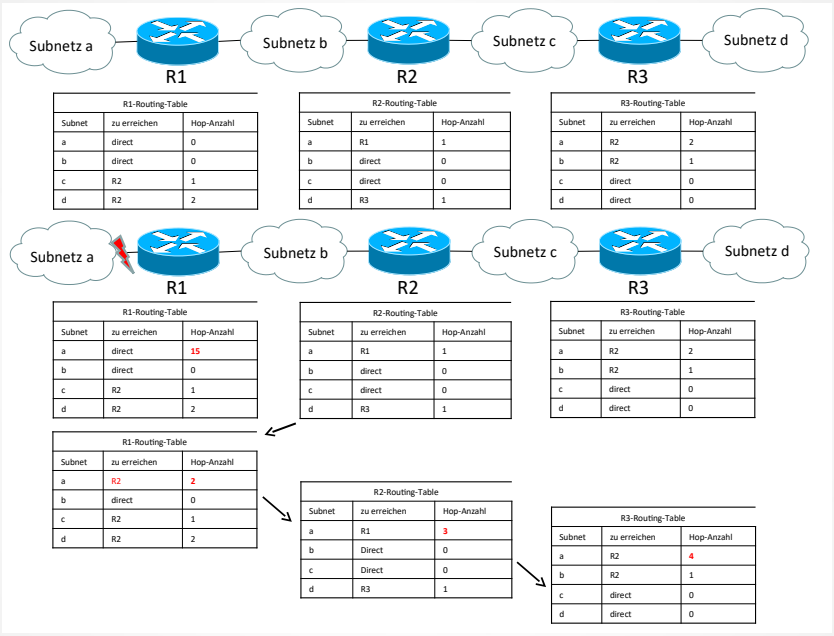
Routing-Protokolle
RIPv1-Ablauf-1



Sobald ein Router gebootet wird, ermittelt er die direkt an ihn angeschlossen Netzwerke und teilt diese seinen Router-Nachbarn in Form eines Broadcast mit. Von den anderen Routern erhält er die Informationen über die restlichen Netzwerke. Wie die Router sich gegenseitig mit Informationen versorgen ist am folgenden Beispiel zu Sehen.

Hier wird klar, dass es lange dauert, bis alle Router die Informationen über alle Subnetze erhalten haben. Im obigen Beispiel dauert es bis zu 2 Minuten, um alle Subnetze auf allen Routern bekannt zu manchen. Diese Konvergenzzeit ist der Nachteil der timerbasierten Informationsverteilung.

Routing-Protokolle
RIPv1-Ausfall einer Route



Wenn nun eine Route ausfällt (z. B. der Anschluss an das Subnetz a am Router 1) trägt der angeschlossene Router (R1) für das Subnetz a die Hop-Anzahl 15 ein.

Da er auf den Timer warten muss bis er diese Information an andere Router weitergeben kann ist es möglich, dass er vom Router 2 die Information bekommt dass der Router 2 das Subnetz mit 2 Hops erreichen kann.

Deshalb trägt der Router 1 das Subnetz a als erreichbar über den Router 2 mit der Hopanzahl 3 ein.
Diese Information überträgt er an den Router 2 beim nächsten Timer-Intervall.

Damit zählen die Router 1 und 2 so lange die Hopcount-Zähler für das Subnetz a hoch bis der Wert 15 (also unerreichbar) erreicht wird.

Auch hier wird ersichtlich, dass die Konvergenzzeit einige Minuten dauern kann.

Routing-Protokolle
RIPv1-beschleunigungsmaßnahmen

Zur Beschleunigung der Konvergenzzeit wurden folgende Maßnahmen definiert.

Split Horizon (deutsch: geteilter Horizont)
Ein Router wird Informationen nur an die Subnetze weitergeben aus denen er die Informationen nicht bekommen hat. Das bedeutet, dass wenn er an einem Port ein Subnetz mitgeteilt bekommen hat, wird er dieses Subnetz an diesem Port nicht wieder weiter verkünden.

Split Horizon und Poison Reverse (deutsch: vergifteter Rückweg)
Hierbei werden wieder alle Subnetze auf allen Ports angekündigt. Allerdings werden die Subnetze mit dem Hopcount 15 zurückgegeben aus denen sie gekommen sind.

Triggered Updates (deutsch: ausgelöste Aktualisierungen)
Hierbei wird nicht bis zum nächsten Timeralarm gewartet bis eine neuen Information verbreitet wird. Es wird sofort nachdem ein Port als „Down“ erkannt wird, die Information weitergeleitet.

Zusätzliche Maßnahmen
Routen die über RIP gelernt wurden haben nur eine Lebensdauer von 3 Minuten. Sollte in der Zwischenzeit kein Update von einem anderen Router erfolgen bekommt die Route die Metrik 16.

Routing-Protokolle
RIPv2

1. Byte	2. Byte	3. Byte	4.Byte
Comand	Version	Pasword	
Family of Net 1		0	
IP-Address of Net 1			
Subnet-Mask of Net 1			
0			
Distancte to Net 1 (Hops)			
Family of Net 1		0	
IP-Address of Net 2			
Subnet-Mask of Net 2			
0			
Distancte to Net 2 (Hops)			
...			

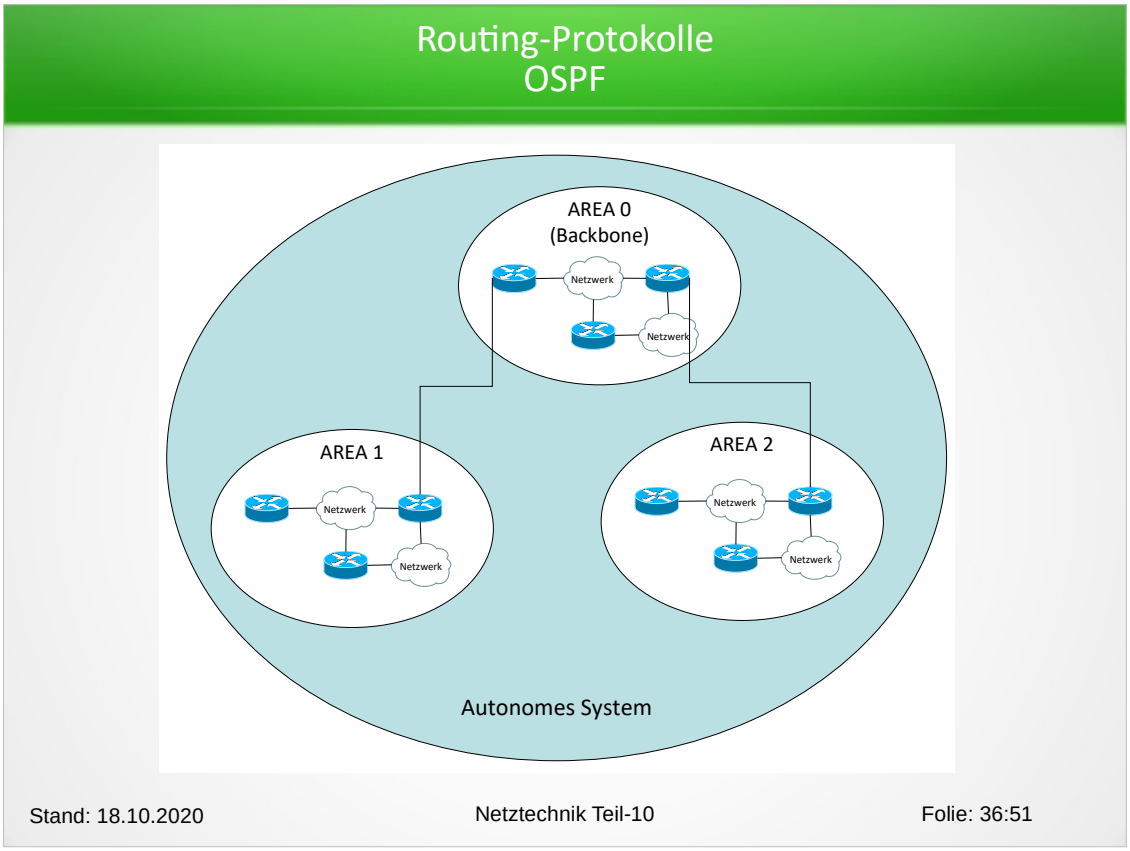
Verbesserungen gegenüber der Version v1:

- Multicast (IP-Multicast-Adresse 224.0.0.9)
  → Nur noch Router beschäftigen sich damit.
- Subnetzmasken-Information wird übermittelt
  → (VLMS = Variable Length of Subnet Mask).
  Damit ist klassenlose IP-Adressierung / Subnetting möglich.
- Es werden bis zu 25 Netzwerke in einem RIP-Paket mitgeteilt.
- Die maximal gültige Hop-Anzahl ist 14.
  Ein Netzwerk mit 15 Hops wird als nicht erreichbar übermittelt.

Die maximal gültige Hop-Anzahl ist 14. Ein Netzwerk mit 15 Hops wird als nicht erreichbar übermittelt.

Extended RIP erlaubt eine Hop-Anzahl von 127 Hops. Dabei bedeutet die Hop-Anzahl von 128, dass das Netzwerk nicht mehr erreichbar ist.

Ein Router sendet bei Hochlauf einmal seine bekannten Routen. Danach alle 30 Sekunden. (Kann bei BAY-Networks z. B. für ISDN-Verbindungen auf etwa eine Std. korrigiert werden)



Autonome Systeme

Bei OSPF sind die Router mit ihren Netzwerken zu einem Autonomen System (AS) zusammengefasst.
Ein Autonomes System ist dabei eine **Verwaltungseinheit**, die es dem Administrator ermöglicht mit beliebigen hierarchischen Routing Architekturen zu arbeiten und nach außen hin definierte Schnittstellen zu haben.

OSPF ist als IGP (**Interior Gateway Protocol**) konzipiert und regelt somit das Routing innerhalb eines Autonomen Systems.

Für das Routing zwischen autonomen Systemen sind Algorithmen nach dem EGP (Exterior Gateway Protokoll) zuständig.
Im Internet wird das BGP (Border Gateway Protocol) verwendet.

Routing-Protokolle
OSPF-Eigenschaften

Eine OSPF-Topologie ist in mehrere Hierarchie-Ebenen eingeteilt:
1) Autonomes System → Gesamtheit aller über ein Backbone verbundenen Areas
2) Backbone → Verbindung von Areas
3) Area → Gruppierung von Netzwerken
4) Netzwerk

OSPF unterstützt drei Arten von Verbindungen zwischen Netzwerken:
Punkt-zu-Punkt-Verbindungen zwischen zwei Routern.
Mehrfachzugriffsnetzwerke mit Broadcasting. Das sind die meisten LANs.
Mehrfachzugriffsnetzwerke ohne Broadcasting. Das sind die meisten paketvermittelnden WANs.

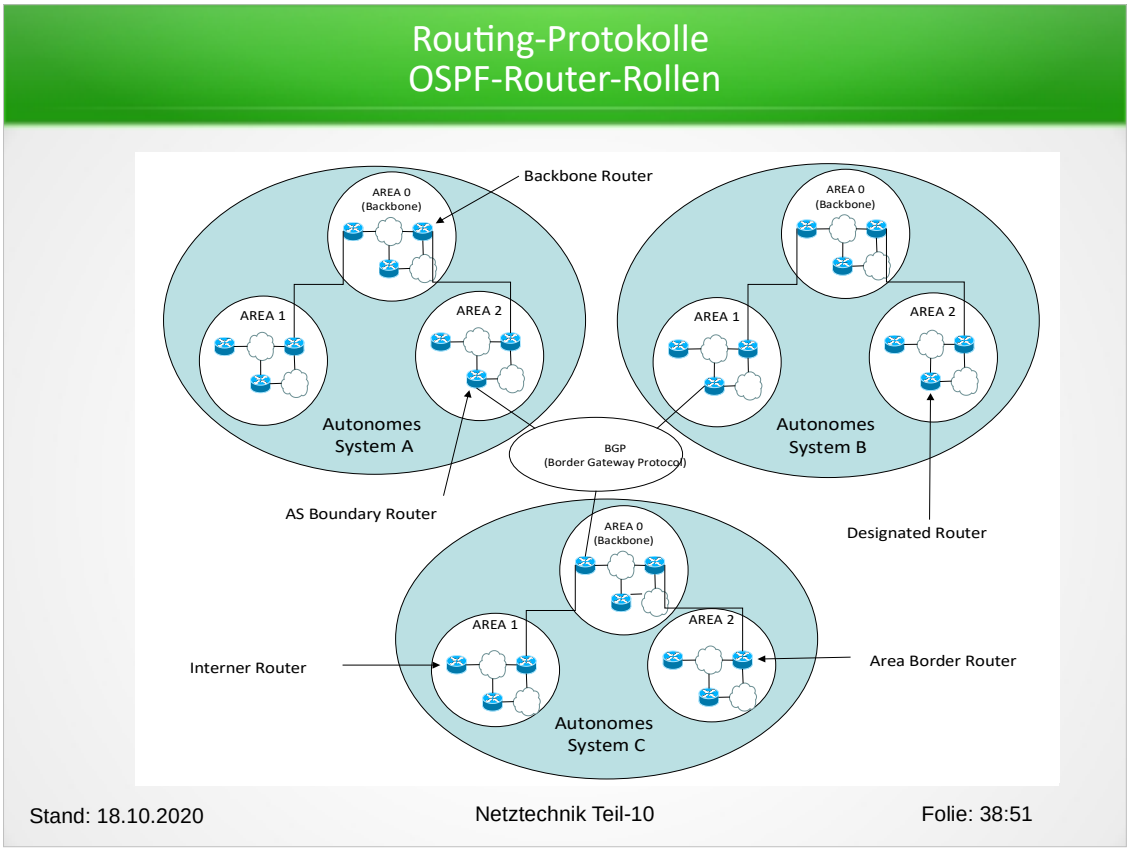
OSPF ist ein Link-State-Algorithmus.
Die Routing Entscheidung wird demnach aufgrund des Link-Status ermittelt bzw. korrigiert.

- Es können Netzwerke über mehr als 14 Zwischen-Systemen erreicht werden.
- OSPF konvergiert bei Netzwerk-Änderungen schneller
- Geringerer Overhead
- Unterstützung hierarchischer Netzwerk-Strukturen
- Unterstützung zur Authentifizierung

Innerhalb eines AS sind die Areas hierarchisch organisiert.

Ganz oben ist die Backbone Area mit der Area-ID = 0 zu definieren.

Alle anderen Areas werden an diese Area angeschlossen



In den Areas haben die Router, je nach ihrer Position in der Area, unterschiedliche Klassen:

- Designierte Router

Ausgewählter Router der stellvertretend für alle Router einer Area mit anderen Areas Informationen austauscht. Er kann über die Multicast Adresse 224.0.0.6 adressiert werden. Dieser Router wird über das HELLO-Protokoll aus allen Routern einer Area ausgewählt.

- Interne Router

Diese Router arbeiten innerhalb eines AS oder eines Backbones.

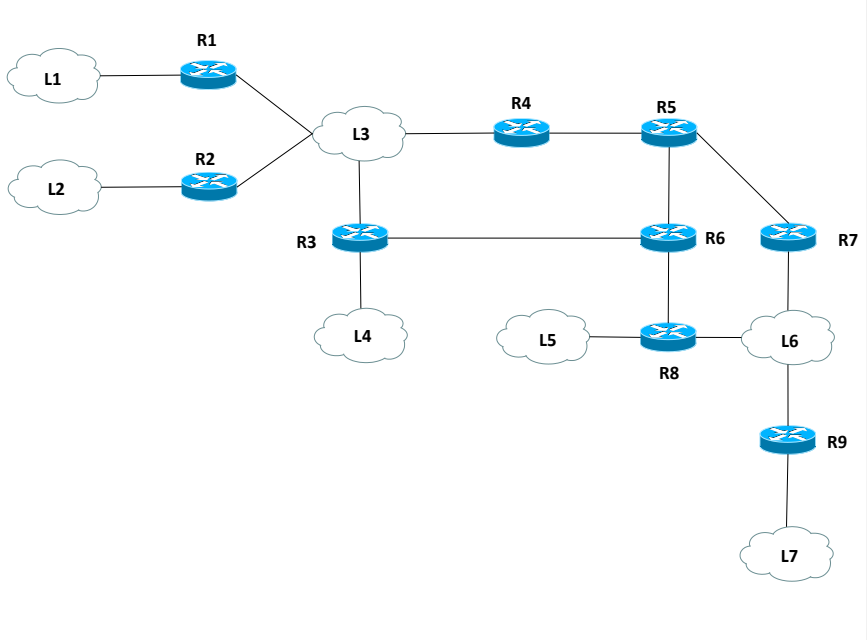
- Area Border Router

Dieser Router verbindet zwei Areas oder eine Area mit der Backbone Area.

- AS Boundary Router

Diese Router verbinden autonome Systeme miteinander.

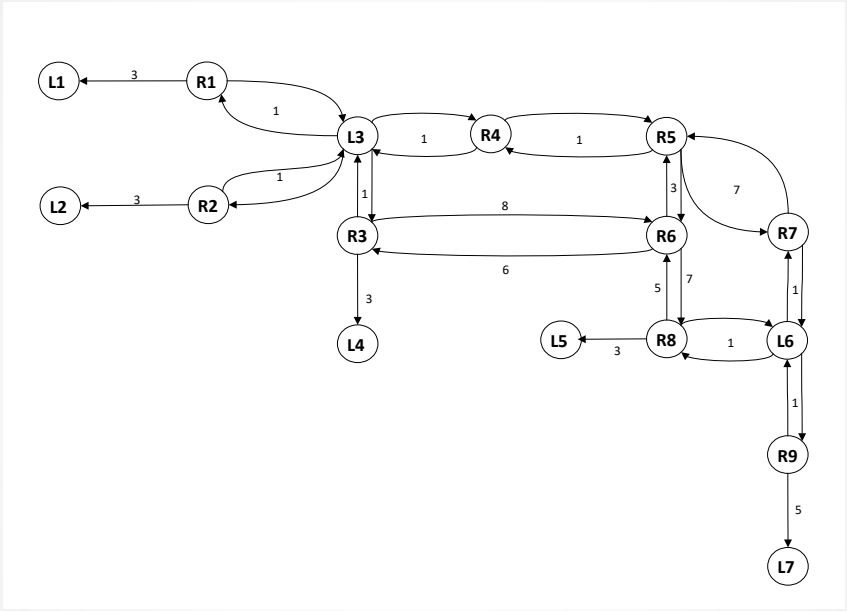
Routing-Protokolle
OSPF-Ablauf-1



Hier sind sowohl LANs als auch WAN-Verbindungen (Direkte Verbindungen zwischen den Routern) dargestellt.

Die WANVerbindungen werden hier wie ein LAN mit zwei Netzwerkadressen behandelt.

Routing-Protokolle
OSPF-Ablauf-2



In der Graphen-Darstellung werden die Kosten in die einzelnen Netzwerke dargestellt.

Für die spätere Berechnung werde nur die Kosten in die Netzwerke verwendet! Deshalb ist bei Netzwerken aus denen keine Routing-Information kommen kann nur der Weg in das LAN mit Kosten bewertet. (Z.B. R1 -> L1 = 3)

Je niedriger die Kosten sind, desto attraktiver ist es für einen Router den Weg auszuwählen.

Sind mehrere Router an einem LAN angeschlossen bildet das LAN einen eigenen Knoten. Bei Punkt-zu-Punkt Verbindungen ist das dazwischen liegende Netzwerk nicht als eigener Knoten ausgebildet.

Die Wege können unterschiedlich bewertet sein. In der obigen Abbildung sind, bis auf die Ausnahme der Verbindung R3 <-> R6, die Kosten sowohl in das Netzwerk als auch aus dem Netzwerk heraus gleich.

Werden die Kosten über redundante Wege gleich gesetzt, erhält man ein einfaches Loadbalancing.

Routing-Protokolle
OSPF-Ablauf-3

Router	Subnetze mit Kosten
R1	L1 = 3, L3 = 1
R2	L2 = 3, L3 = 1
R3	L3 = 1, L4 = 3
R4	L3 = 1
R5	-
R6	-
R7	L6 = 1
R8	L5 = 3, L6 = 1
R9	L6 = 1, L7 = 5

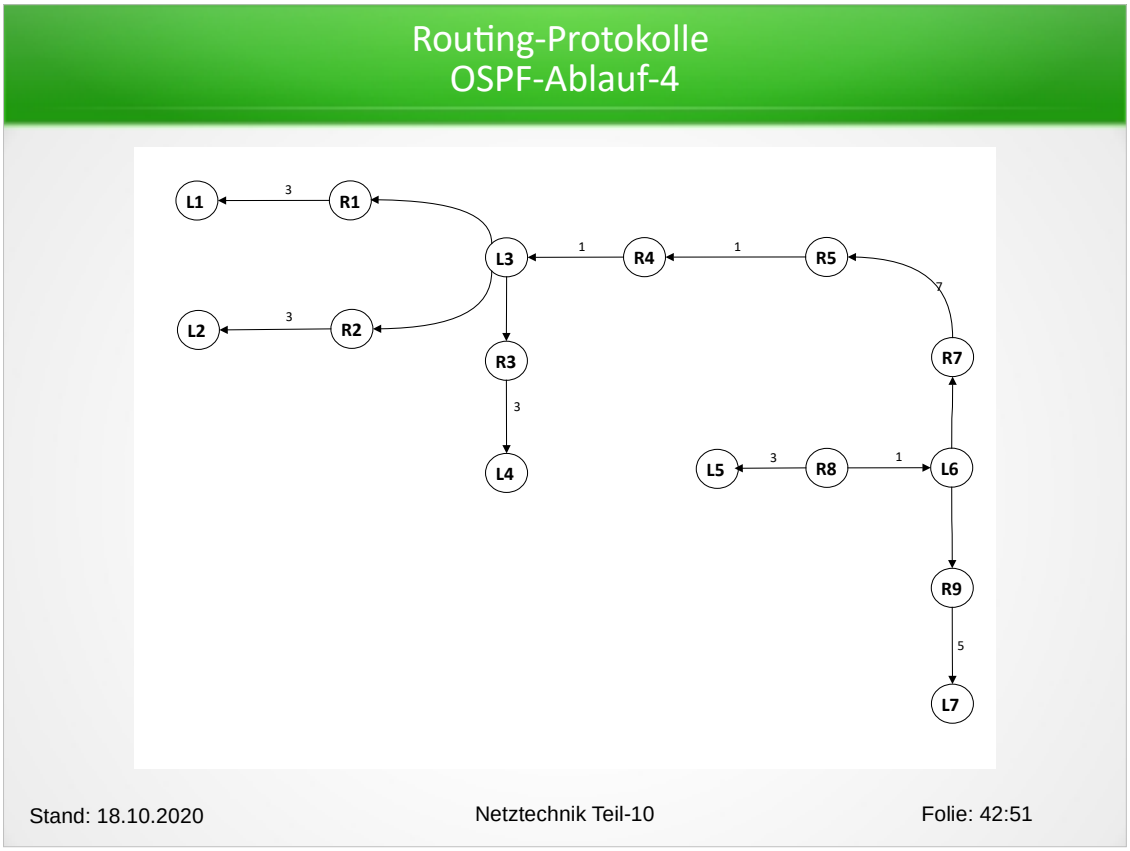
Für die einzelnen Router ergeben sich damit folgende angeschlossenen Netzwerke mit den zugehörigen Kosten. Damit kann für alle Router die LSDB (Link State Data Base) aufgebaut werden.

Die Datenbank ist vollständig, wenn jeder Router von jedem Router eine gültige Liste empfangen hat.

Z. B. Kostentabelle von R8

Ziel-Netzwerk	Next Hop	Distance (Kosten)
L1	R7	13
L2	R7	13
L3	R7	10
L4	R7	13
L5	direct	3
L6	direct	1
L7	R9	6

Mit der Kostentabelle kann der Router R8 einen Graphenbaum aufbauen. Auffällig ist dabei, dass es in jedes LAN nur einen Weg gibt. (Siehe auch Spanning Tree) Er sieht folgendermaßen aus:



In der Graphen-Darstellung werden die Kosten in die einzelnen Netzwerke dargestellt.

Für die spätere Berechnung werde nur die Kosten in die Netzwerke verwendet! Deshalb ist bei Netzwerken aus denen keine Routing-Information kommen kann nur der Weg in das LAN mit Kosten bewertet. (Z.B. R1 -> L1 = 3)

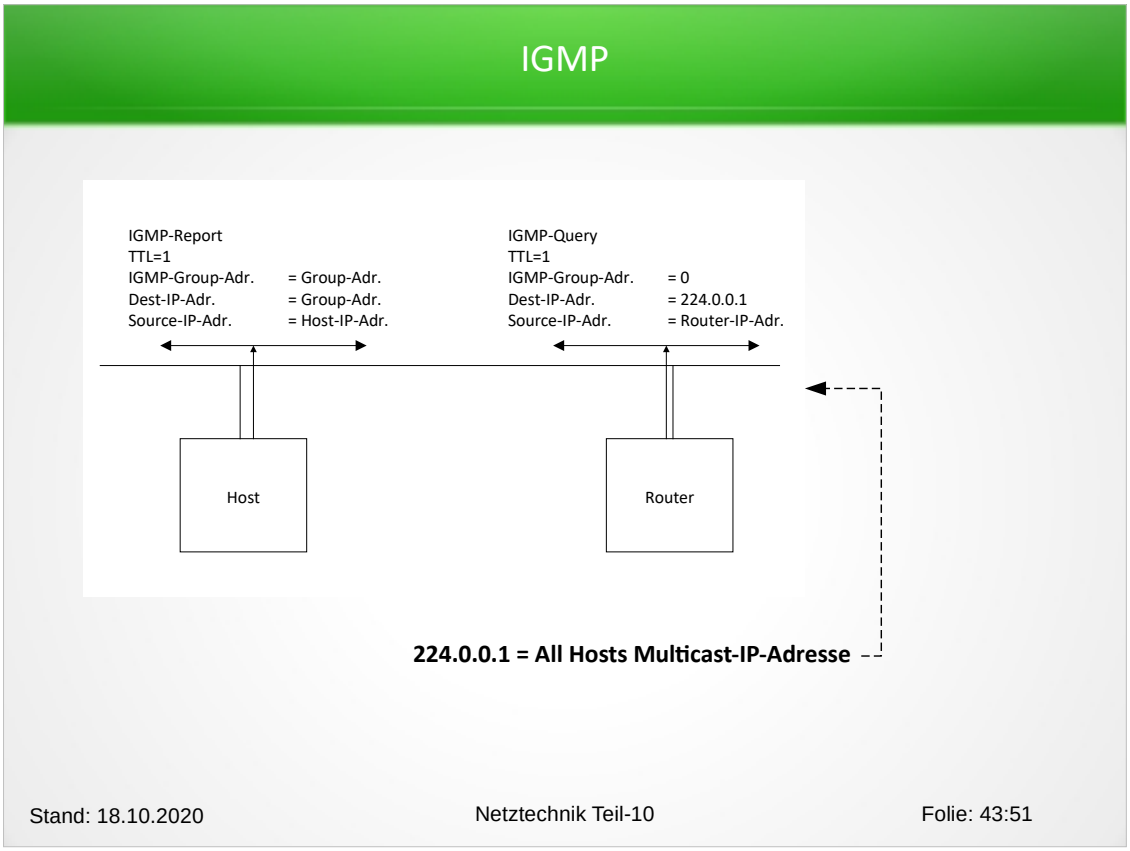
Je niedriger die Kosten sind, desto attraktiver ist es für einen Router den Weg auszuwählen.

Sind mehrere Router an einem LAN angeschlossen bildet das LAN einen eigenen Knoten. Bei Punkt-zu-Punkt Verbindungen ist das dazwischen liegende Netzwerk nicht als eigener Knoten ausgebildet.

Die Wege können unterschiedlich bewertet sein. In der obigen Abbildung sind, bis auf die Ausnahme der Verbindung R3 <-> R6, die Kosten sowohl in das Netzwerk als auch aus dem Netzwerk heraus gleich.

Werden die Kosten über redundante Wege gleich gesetzt, erhält man ein einfaches Loadbalancing.

Mit der Kostentabelle kann der Router R8 einen Graphenbaum aufbauen. Auffällig ist dabei, dass es in jedes LAN nur einen Weg gibt. (Siehe auch Spanning Tree) Er sieht folgendermaßen aus:



Um mit Multicasts arbeiten zu können, muss sich ein Gerät bei der Multicast-Gruppe anmelden. Die Mitgliedschaft in einer Multicast-Gruppe ist dynamisch. Ein Prozess kann einer Multicast-Gruppe beitreten und sie wieder verlassen. Dies wird von einem entsprechenden Multicast-API unterstützt. Eine Multicast-Gruppe wird an eine Interfacekarte gebunden. Ein Prozess kann derselben Multicast-Gruppe auf mehreren Interfaces beitreten. Ein Host erkennt eine Gruppe an der Gruppen-Adresse an dem zugehörigen Interface. Die Verwaltung der Gruppen und Interfaces, führt jeder Host für sich einmal in einer Tabelle.

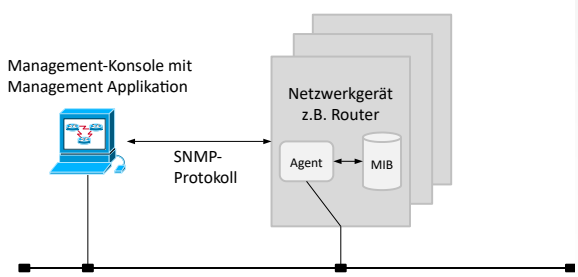
Wenn ein Host einer Gruppe bei tritt, sendet er einen IGMP-Report. Dies geschieht nur beim ersten Beitritt zu dieser Gruppe an diesem Port. (Auch bei mehreren Prozessen) Der Report wird an dem Port ausgesendet, an dem die Gruppen-Zugehörigkeit gelten soll.
Beim Verlassen des Hosts von der Gruppe wird kein Report gesendet.

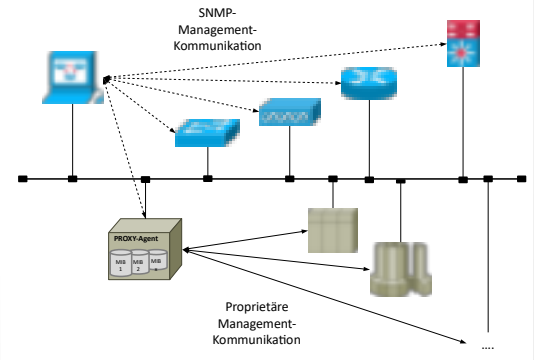
Ein Multicast Router sendet zyklisch IGMP-Query's auf allen parametrierten Interfaces um zu ermitteln, ob sich ein Multicast-Host an diesem Interface befindet. Auf diesen IGMP-Query meldet sich ein Multicast-Host mit einem IGMP-Report für jede Gruppe, die mindestens noch einen Prozess auf diese Gruppe gebunden hat. Dies geschieht mit einer zufälligen Verzögerung, damit nicht mit einer Report-Lawine auf einen Query reagiert wird und von evtl. mehreren Hosts gleichzeitig das Netzsegment blockiert wird. Außerdem kann ein Host, der einen Report senden müsste, und auf seinem Interface von einem anderen Host einen Report für seine Gruppe erkennt, auf seinen Report verzichten. Dem Router reicht es zu wissen, dass an einen Interface mindestens ein Multicast-Host angeschlossen ist. Damit baut der Multicast-Router eine Tabelle auf. Sobald ein Router einen Multicast empfängt, überprüft er anhand dieser Tabelle, auf welchen Interface er den Multicast weitergeben muss.

Bei einem einfachen Netzwerk (ohne Router) werden nur einmal IGMP-Reports gesendet wenn sich ein Prozess bei der Multicast-Gruppe anmeldet.
Der TTL-Wert ist normalerweise auf 1 gesetzt. Somit beschränkt sich die IGMP-Meldung auf ein Subnetz. Von den Routern werden keine ICMP-TTL-EXCEEDED-Meldungen erzeugt.
Für bestimmte Server gibt es den Sonderfall des Expanding Ring Search (deutsch:

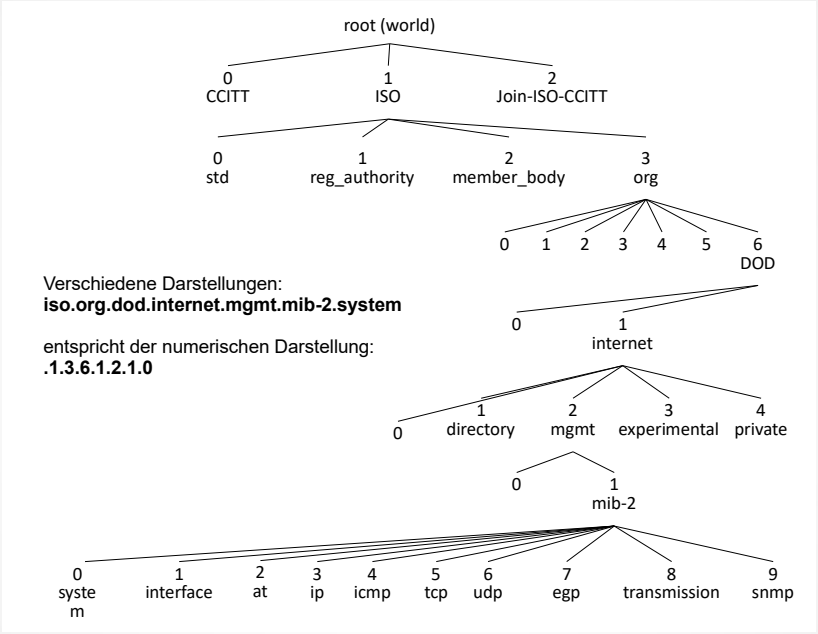
Netzwerk-Management
Übersicht

-	SNMP
TCP	UDP
IP	
Ebene 2	
Ebene 1	





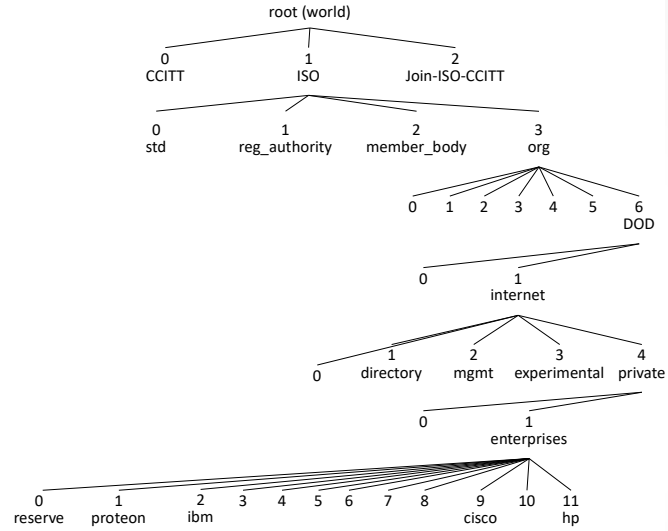
Netzwerk-Management
MIB-Aufbau



Verschiedene Darstellungen:
iso.org.dod.internet.mgmt.mib-2.system

entspricht der numerischen Darstellung:
.1.3.6.1.2.1.0

Netzwerk-Management
MIB-Herstellerteil



Kennung	Hersteller
0	
1	proteon
2	ibm
9	cisco
11	hp
23	novellMib
119	nec
197	kalpana
353	atmForum
437	grandjuction
494	madge
711	lightstream

Netzwerk-Management
SNMPv1 / SNMPv2c

SNMPv1

Aufruf	Bedeutung	Richtung	Port
get	Anforderung einer MIB-Variablen	Manager -> Remote-Gerät	161
getnext	Anforderung der lexikographisch nächsten Variablen	Manager -> Remote-Gerät	
response	Antwort auf einen get/getnext-Telegramm	Remote-Gerät -> Manager	
set	Setzen einer MIB-Variablen	Manager -> Remote-Gerät	
trap	Information an den Manager	Remote-Gerät -> Manager	162

SNMPv2c

Aufruf	Bedeutung	Richtung
get	Anforderung einer MIB-Variablen	Manager -> Remote-Gerät
getnext	Anforderung der lexikografisch nächsten Variablen	Manager -> Remote-Gerät
getbulk	Anforderung großer MIB-Bereiche	Manager -> Remote-Gerät
response	Antwort auf einen get/getnext-Telegramm	Remote-Gerät -> Manager
set	Setzen einer MIB-Variablen	Manager -> Remote-Gerät
inform	Versand bestätigter Meldungen Kommunikation zwischen Managern	Remote-Gerät<-> Manager Manager<-> Manager
trap	Information an den Manager	Remote-Gerät -> Manager

SNMPv3

Allgemeines

Die Sicherheitsmechanismen bei SNMP in den beiden ersten Versionen waren so schlecht, dass SNMP für „Security is not my problem“ stand. Mit der Version 3 wurden sie Sicherheitsmechanismen verbessert. Durch die komplexe Schlüsselverwaltung hat SNMPv3 jedoch noch keine große Verbreitung gefunden.

Netzwerk-Management
RMON / SMON

Im MIB-Baum wurde **RMON** unterhalb des MGMT-Knotens angesiedelt.
RMON enthält folgende Klassen:

RMON-Klasse	Bedeutung
1	Statistics
2	History
3	Alarms
4	Hosts
5	Host Top10
6	Matrix
7	Filters
8	Capture
9	Events
10	Token Ring

RMON-Klasse	Bedeutung
11	Protocol Directory
12	Protocol Distribution
13	Address Map
14	Network Layer Host
15	Network Layer Matrix
16	Application Layer Host
17	Application Layer Matrix
18	User History
19	Probe Configuration
20	Conformance

SMON
Speziell für Switches wurden die Grundlagen für die Belange von Switches erweitert und in einer eigenen Struktur festgelegt

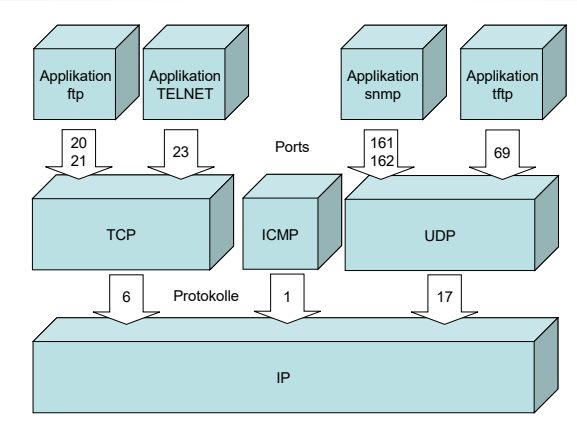
Netzwerk-Management
Beispiel: Cisco-View

CiscoView

Dient zur graphischen Ausgabe des Gerätes. Außerdem kann hier, je nach Gerät, eine Vielzahl von Änderungen durchgeführt werden. Z. B. Ändern von VLAN-Portzuordnungen.

Bekannte weitere Beispiele:
NAGIOS (Open-Source)
Icinga
PRTG
Shinken
...

Anwendungsprotokolle
Übersicht



Anwendungsprotokolle
Beispiel FTP