# SYP(SEP)

* TCP/IP
* Firewalls
* Wondows Server
* Active Directory
* Virtualiesierungen
* Cluster
* Ausfallsicherheit
* Storage System / SAN
* Cloud Systeme
* Urheberrecht
* Lizensrecht
* Auschreibung

# Active Directory:

* Global Catalog

LAN/Domain

La

SAP

Mail Server

User

Domain: Alle Objecte bilden die Domain

Domain

Firma .at und Firma.local

Einkauf.firma.at

Side to Side Replication

Netzwerkberreich am Ort A ist eine Side und sie sind in der gleichen Domain.

* DNS , LDAP
* Group policies

Globale Gruppe

User

Locale Gruppe



Rechte

|  |
| --- |
| 4 |
| 3 |
| 2 |
| 1 |

|  |
| --- |
| Anwendungsschicht |
| Darstellungsschicht |
| Sitzungsschicht |
| Transportschicht |
| Vermittlungsschicht |
| Sicherheit |
| Bit übertragung |

**OSI 7-Schichten Modell**

Protocolle

* TCP/IP (OSI Schichten Modell)
* Ftp (zum file transfer)
* telnet (zb auf linux auf datein auf externem Rechner zugreifen)
* smtp
* ARPANET

Schichten Modell:

* Aufgabe Schicht 1 : Datenübertragung,

Spannungsänderungen(1 und 0)

* + Manchastercodierung (zur Vermeidung Fehlintrepration von Spannung)

Zuerst Invers (Doppelte Bitrate)

Macadresse (Switch merkt sich MA) wird übermittelt

Pakete nie gleichzeitig schicken

* + CAT 5(bis zu 1GB, 8 Leitungen, jeweiles 2 verdrillt),6,7 (Kabel)

4 Gleichzeitig

* Aufgabe Schicht 2 : Logische Absicherung : Parität(Parity bit)

BCC (64 + 16 Bits, für 64 Datenbits)

10010001 1

11010000 1

10000111 0

11101111 1

00000001 1

01110001 0

10100000 0

11111111 0

00000110

Bei mehr fehlern 🡪 langsamer übertragen

**20.09.2017**

**(zu Punkt 2) Sicherheit)**

* + CRC (Fehlerabsicherung in diesem Protokoll )

1101

X^3 + x^2 + (0\*x) +1

Restklasse

1+1 = 0 (2/2 = 0)

1-1 = 0

0+0 = 0

1. = 0

**(XOR verküpfung, nur wenn verschieden dann 1 ) / shift**

**1010010** / 1011 (sowie Primzahl , nur durch sich selbst dividierbar, also Rest)

1011000

= 0001010

1011

= 0001 🡪 1010010 (Fehlerkorrektur (0001) = **1010011**

**1010011 /1011 🡪 Ergibt 0 Rest (=Probe)**

CRC 🡪 dieser Algorithmus

* + Hemmingkodierung

**1**0**0**1**0**1**1**0**1**0**1**1**0**1**0** (4 mal 1) 🡪 0 (als Parität ergänzt)

Orange 🡪 0 (als Parität ergänzt)

**1**0**0**1**0**1**1**0**1**0**1**1**0**1**0 Blau 🡪 1 (als Parität ergänzt)**

**1**0**0**1**0**1**1**0**1**0**1**1**0**1**0 Rot 🡪 0 (als Parität ergänzt)**

* **0100 🡪 dezimalzahl 4 🡪 also 4 Bit Falsch**

**100101101011010 🡪 100001101011010**

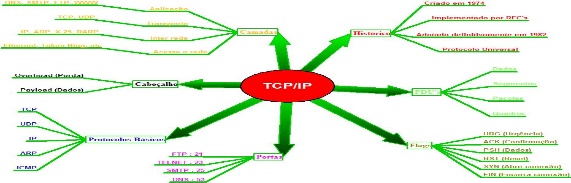
(nur möglich bei einem Fehlerbit)

# TCP

***Protocoll ( Regelwerk)***

( Gateway übersetzt Protocolle)

* Verbindungslos(fast zu 100% heute)
  + Keine Durchgehende Leitung von Sender zu Empfänger
  + Beispiel: Handynetz, Internet , ...



[This Photo](http://pt.wikipedia.org/wiki/Ficheiro:TCP-IP.jpg) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

* Verbindungsorientiert

(Beispiel : Telefonnetz (Festnetz)) / 95 %Verschwendung der Auslastung

* + Teuerer ( kein ununterbrochener Datenübertragung / nicht ausgelastet)

QoS ( Garantierte Bandbreite) 🡪 nicht immer gewährleistet

# ROUTING

(intelligente Navigierung durch Knotenpunkte)

* Statiches Routing

Lan mit routing / statische Tabellen (im Internet nur dynamisches Routen)

* Dynamisches Routing

**21.09.2017**

**Storage**

Disc Quoter / user speicherplatz begrenzen

*Disk Management*

HSM // Historical Storical Management

//Auslagerung lange nicht verwendeter Daten

EFS //Encryptment File System

//Für verschlüsselung

* Public Key

Primzahlen, Modoloalgorithmus...

Keine entschlüsselung ohne private key möglich

Nicht mathematisch lösbar

* Private Key

**Management**

* WSH // Windows Scripting Host

Zb VBScripts,JavaScripts

Für beispielsweise email erstellung Schueler

* DFS //Destributed File System

**25.09.2017**

32 Bit Datenfelder

Optionen :

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 8 | 16 | 32 |
| Version | Lage | Service Typen | Paketlänge(Byte) |
| Identifikation | | X M F | FragmentAbstand |
| Lebsenszeit | | Transport | Kopfprüfsumme |
| Senderadresse | | | |
| Empfängeradresse | | | |
| Optionen Füllzeichen | | | |

* Danach kommen die Daten

**12345678**

Priority | WDZ00

Internet Router:

* WAN
* LAN

192.168.0.0 / 16 (Präfix)

Netz ID | Host ID

**192.168.10.0**/24

In Teilnetze aufteilen(jeweils <30 in jedem Teilnetz)

**27.09.2017**

192.168.10.0/24

192.168. + 8 Bits (5 Bits 🡪 Host ID / Netz ID 🡪 3 Bits)

NetzID:

0 0 0

0 0 1

0 1 0

0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

192.168.10.00100001

**Aufgabe:**

Router ( 4 Steckplätze) 🡪 Router höchstmögliche Adresse

* 1 Netz soll 15 Rechner haben (Teilnetz) (mit switch dazwischen)
* 2 Netz soll 20 Rechner haben (Teilnetz) (mit switch dazwischen)
* 3 Netz soll Serverraum sein (mit 3 Servern)
* 4 Netz soll Internet Router sein

Adresse 🡪 192.168.10.0/24

2^5 🡪 30 Adressen in jedem TeilNetz

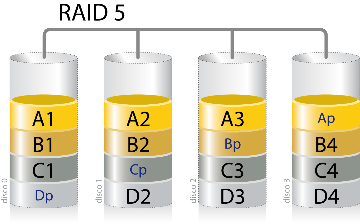
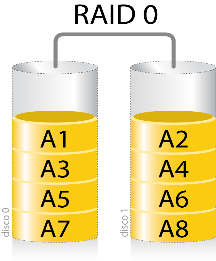
Teil-Netze:

* 1) **010** 
  + Rechner 1 : 192.168.10.01000001/27
  + Rechner 15: 192.168.10.01001111/27
  + Router : 192.168.10.01011110/27
* 2) **011**
  + Rechner 1: 192.168.10.01100001/27
  + Rechner 20: 192.168.10.01110100/27
  + Router: 192.168.10.01111110/27
* 3) **100**
  + Server 1: 192.168.10.10000001/27
  + Server 3: 192.168.10.10000100/27
  + Router: 192.168.10.10011110/27
* 4) **100**
  + Internet: 192.168.10.10100001/27
  + Router: 192.168.10.10111110/27

**28.09.2017**

Wiederholung:

AD (Active Directory)

* Carborus
  + Passwort wird nicht über das Netz übertragen
* DFS 🡪 Distributed File System
  + SAN-Systeme(Eigene Switches)
  + RAID Systeme
    - 0-6
    - 2 Festplatten zu einem Verbund zusammengeschlossen
    - 0) Keine Redundanz(Wenn einer ausfällt , kompletter Datenverlust1)
    - 1) Parrallel auf beide Festplatten geschrieben

(Man kann von beiden Platten lesen)

* + - 5) Ab 3 platten in Form von Stripes (mit Parity Bit)

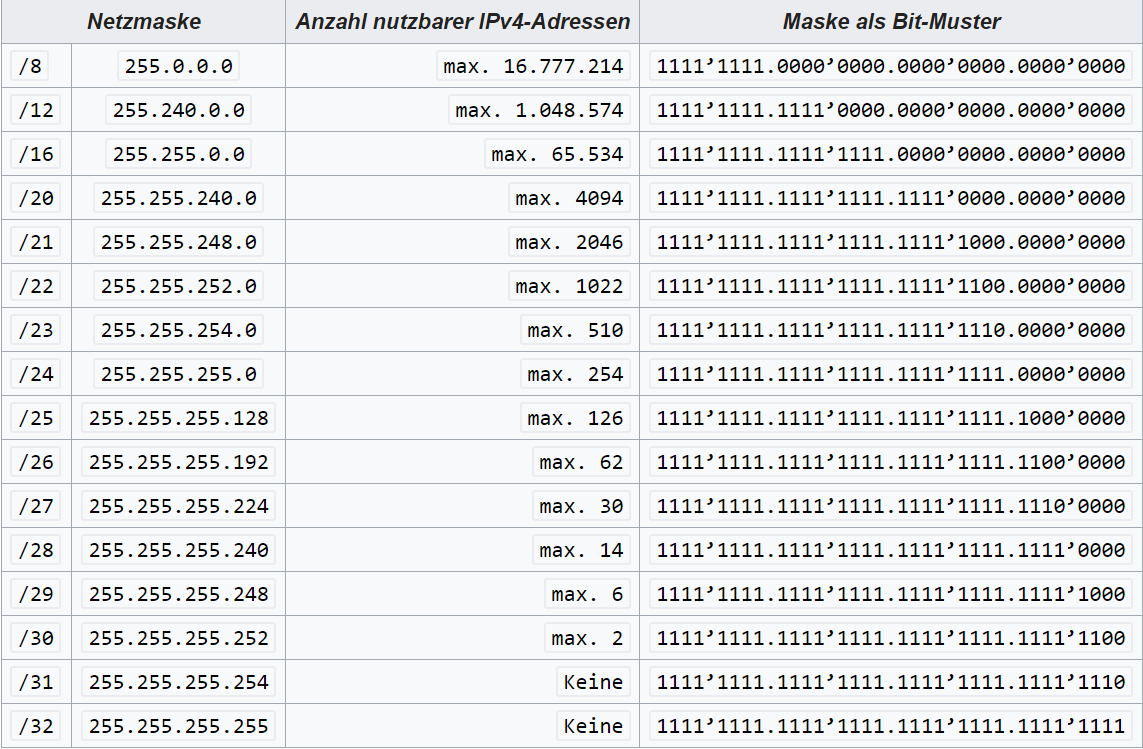
Parität is wechselsetig ( beliebig viele Platten)

* + - Sicherheits platte 🡪 (ohne: wenn 2 Platten ausfallen 🡪 aus)
    - Daten im RAID 5 System , RAID 1 🡪 Betriebssystem(Sichersten)
    - Teuer
* EFS (Encrypting File System)
* DNS für das Verwalten der Files
* Mail Server( gekoppelt mit AD)

**02.10.2017**

Private Adressen:

* 10.0.0.0 bis 10.255.255.255 //gesamter Berreich ca 16 Millionen
* Anders : 10.0.0.0/8 10.0.0.0/24 🡪 weglassen
* 172.16.0.0 bis 172.31.255.255
* 172.16.0.0/12
  + Wie viel bit von 16 bis 31
* 192.168.0.0 bis 192.168.0.0/16



Optionen

* Source Route
  + Strict

Paket muss über eine bestimmte Knotenliste laufen

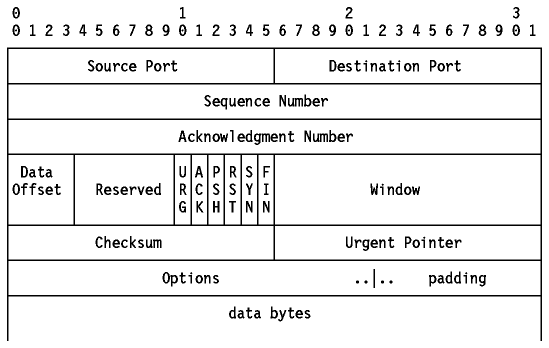
* + Loose
* Record Route
  + Zur Nachverfolgung eines Paketes

Socket Adresse: ?

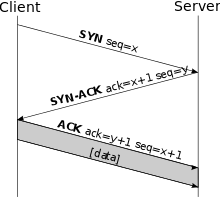
* Vollständige Adresse / Netz IP Host IP Portnummer

TCP (Ähnlich IP (übergeordnete Schicht (von Datenstrom zu Paketstrom, und umgekehrt)))

* Vollduplexfähig(senden un erhalten gleichzeitig)
* Sequenznummern (um Reihenfolge einzuhalten)
* Quittung und Fehlerüberprüfung(falls Pakete verloren gehen)
* Portnummern ( um Dienste zu adressieren //FTP ... haben Portnummern (fix vergeben)) //dynamische (immer ändernd) und statische(immer gleicher Port, zb.80 🡪 http port)

 0 32

* Data Offset: Länge des Kopfes
* Resererved: steht immer auf 0 / wird meistens nicht verwendet
* 5 Bits: Flex/Steuerbits
  + URG
  + ACK
  + PSH
  + RST
  + SYN Bestätigung des Verbindungsaufbaus/ Bestätigung der Bestätigung
  + FIN
* Window : Fenstergröße, wie viel darf ich senden (puffer(dynamisch)) (puffer wird größer bei guter Verbindung, und umgekehrt) // wie viel Platz insgesamt besteht
* Checksum: Prüfsumme//selbständig vom Protocoll generiert
* Urgent Pointer: Urgent Zeiger, ein bestimmtes Element soll vorgezogen werden // das nächste zu verrabeitende Packet
* Options: Optionen/nicht wichtig
* Padding: Füllzeichen

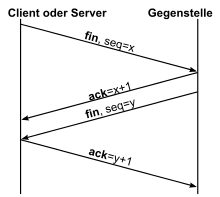
**04.10.2017**

**Protokollablauf**

1. Verbindungsaufbau
2. Seq = 100 | Flag = SYN
3. Seq = 300 | ack = 101 | Flag = SYN-ACK
4. Seq = 101 | ack = 301 | Flag = ACK (Bestätigung der

Bestätigung)

1. Datenaustausch
2. Seq = 101 | ack = 301 | Data = 10
3. Seq = 301 | ack = 111 | Data = 20
4. Seq = 111 | ack = 321 | Flag = ack



1. Verbindungsabau
2. Seq = 111 | ack = 321 | Flag = FIN
3. Seq = 321 | ack = 112 | Flag = ACK | Data = 10
4. Seq = 112 | ack = 331 | Flag = ACK
5. Seq = 331 | ack = 112 | Flag = FIN
6. Seq = 112 | ack = 332 | Flag = ACK

(nur syn und fin erhöhen seq um 1)

**Zeitüberwachung**  // Retransmisson timeout

(Timer beginnt zu laufen wenn Paket weggeschickt, wenn kein Bestätigungspaket 🡪 noch einmal wegschicken)

* Massive ausfälle 🡪 Pakete werden wieder und wieder gesendet 🡪 Ausfall

Steigerung der Performance

1. Verzögerung der Bestätigung

Chance dass Buffer schon richtig / Application schon richtiges Ergebnis / Antwort schon bereit 🡪 Einsparung von Traffic (Paketen)

1. Silly Window Avoidance Syndrom

Window = Puffer(Empfänger/Sender -Puffer)

Senden so lange Puffer nicht voll ist 🡪 Fenstergröße (Puffer) muss bekannt gegeben werden. Erst ab einer gewissen Änderung, muss die neue Fenstergröße bekannt gegeben werden ( Damit nicht bei jeder Sendung platz dafür verbraucht wird) , weniger Quittungen werden benötigt. ( Internet wurde schneller (Punkt 1. und 2. )).

1. Small Packet Avoidance

Warten bis eine gewisse Datenmenge zusammengekommen ist, before Sendung. ( Nicht immer möglich ( z.b Streaming)) 🡪 nur dann wenn keine Interaktiven ( Echtzeit) Anwendung im Spiel sind ( nur im Background ( zb. Filedownloads)

1. Jacobsen Algorithmus

Überlast von Paketen vermeiden(flow control) 🡪 Dem Sender senden langsamer zu senden 🡪 Mit Pausen senden. Router darf niemals in die Volllast gehen, Pakete können abgewiesen werden. Router sendet das Signal, Pausen zu machen.

**09.10.2017**

Persistance Timer

* Solange im empfängerfenster noch platz ist wird gesendet
* Wenn Puffer voll, warten auf Paket , mit neuer größe 🡪wenn verloren , wartet er ewig 🡪Persistance Timer schickt immer wieder Test-Packete

Quiet Time

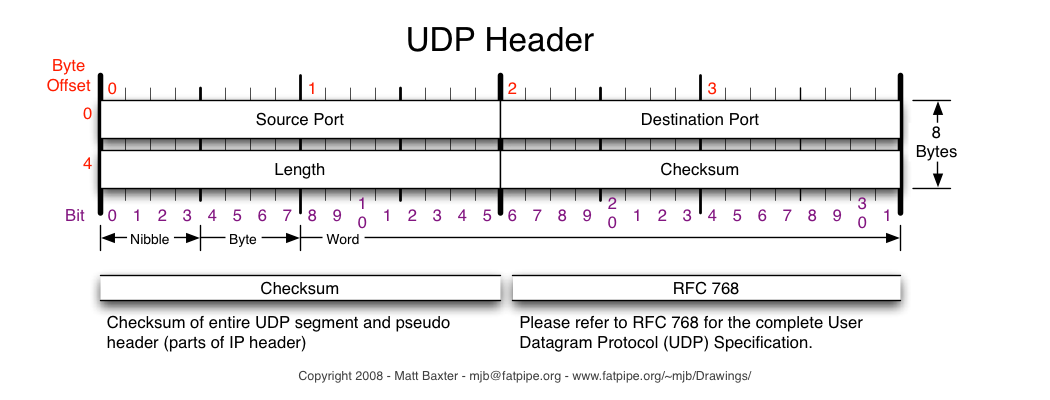
Packete mit gleicher Socket Adresse sind verboten. Nach dem Verbindungsabbruch warten, zum Verhindern. Paket kann maximal unterwegs sein 🡪 30 Sekunden 🡪 dann darf die Verbindung wieder aufgebaut werden.

Keepalive and idle timer

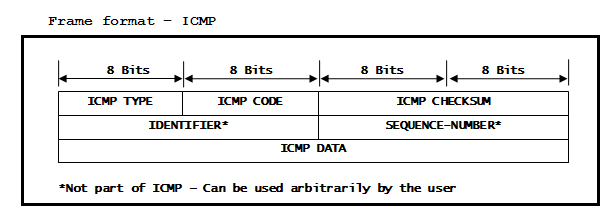
Von Zeit zu Zeit wird ein Testpacket ausgeschickt, um zu sehen, ob die Verbindung noch immer aktiv ist.

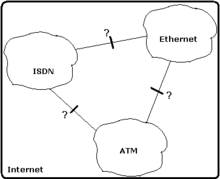
**UDP**

Ersatz von TCP. Kann nicht ohne Einschränkungen verwendet werden. Es ist schneller, Perfomanter. Funktioniert nicht im Internet, aber Lan. Es gibt

* Senderport
* Empfängerport

Wird verwendet im Localen Netzwerk.

**ICMP**

* Typ (z.b Destination unreachable)
* Code
* Prüfsumme
* Verschiedenes (z.b Pausenlänge)
* Verschiedenes
* IP-Kopf + 8 Daten- Bytes

**Internetworking**

* Autonome Zone

Eine voll arbeitsfähige Zone. Mindestens 1 DNS, 1 Gateway nach ausen, 1 Router von innen

* Router

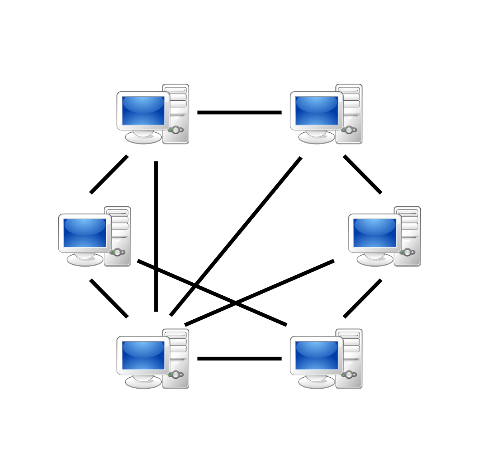
Immer das selbe Protocoll. ( Auf Ip – Ebene Weiterleitung)

* Gateway

Ein Kommunikationsrechner, der verschiedene rechennetze miteinander verbindet. Kann das Protocoll umsetzen. ( Teilweise andere Protocolle ). Verschiedene Protocolle können miteinander kommunizieren.

* Bridge

Verbindung von 2 Netzen untereinander. Es können andere Protocolle verwendet werden. ( Um Entfernungen zu überbrücken)



[This Photo](https://es.wikipedia.org/wiki/Peer-to-peer) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

Routing

* Was ist der schnellste Weg ( durchs Netz ) ? (Routing Strategie)
* Lawinen Routing. Jeder hat an alle Nachbarn gesendet, außer an den,

von dem er gekommen ist.

* Dynamisches Routen

Jeder Router sendet Testpackete aus. Anhand dessen wird entschieden, welcher Weg genommen wird. ( Matrix , Verbindungsgeschwindigkeiten werden gespeichert)

(Ipv6 🡪 bessere Information , 3-4 hubs / ipv4 5-6 hubs)

Im Lan🡪 statisch ( fixe Routing Tabellen ) ( bei Installation müssen alle statisch nachgezogen werden)

Zeitaufwendigkeit: Routing > Datenübertragung (Wegen der vielen Knoten)

**11.10.2017**

Internet Working 🡪 Zellen die für sich abgeschlossen sind🡪innerhalb der Zelle wird mit einem dynamischen Routing gearbeitet.

Lan 🡪 Ethernet Protocoll (Transport)

Kopf

Ethernet

Daten

TCP / UDP

IP

Ethernet übertragungsprotocoll 🡪 (alt ( nur mehr aus kompatibilitätsgründen) 🡪 CSNA/CD ( Carrier Sense / Collision Detection) ).

Neu: Jeder Rechner hat seine eigene Leitung 🡪 flow Control / auf dieser Leitung kann keine Koallision auftreten.

Im Hintergrund : Bus system ( WLAN : immer nur einer Senden, einer Warten // Performance schlecht)

**16.10.2017**

Script HUI

**18.10.2017**

Script HUI

**23.10.2017**

Script HUI

**25.10.2017**

Ausschreibung:

* Offenes Verfahren   
  muss wirtschaftlich und technisch möglich sein
* Nicht offenes Verfahren(beschränkt ausschreiben)
* Freihändige Vergabe  
  nicht Ausgeschrieben 🡪 bestellt / nur bei geringen Beträgen
* Sonderregelung bei geistiger Wertschöpfung

Offennes / nicht offenes Verfahren auch als Verhandlungsverfahren

**30.10.2017**

Preisarten :

* Einheitspreis
* Bauschpreis
* Regiepreis

Alle obrigen vorhanden als :

* Fix
* Veränderlich ( wird angepasst an die Marktsituation)

Stoff

* Windows server
* Active Directory ( Catalog)
* Replication
* FSMO
* DNS
* Group polices
* ISO Osi Modell mit Schicht Ethernet (Zusammenfassung HUI (Ethernet)) (nur 1-2 Schicht)
* Switches...
* Sicherungsschicht
* Parity prüfung
* CrC
* Hemming Codierung
* Doppelstrom
* HSM
* Management im Windows server
* Derouted file system
* Kat5 /kat 6