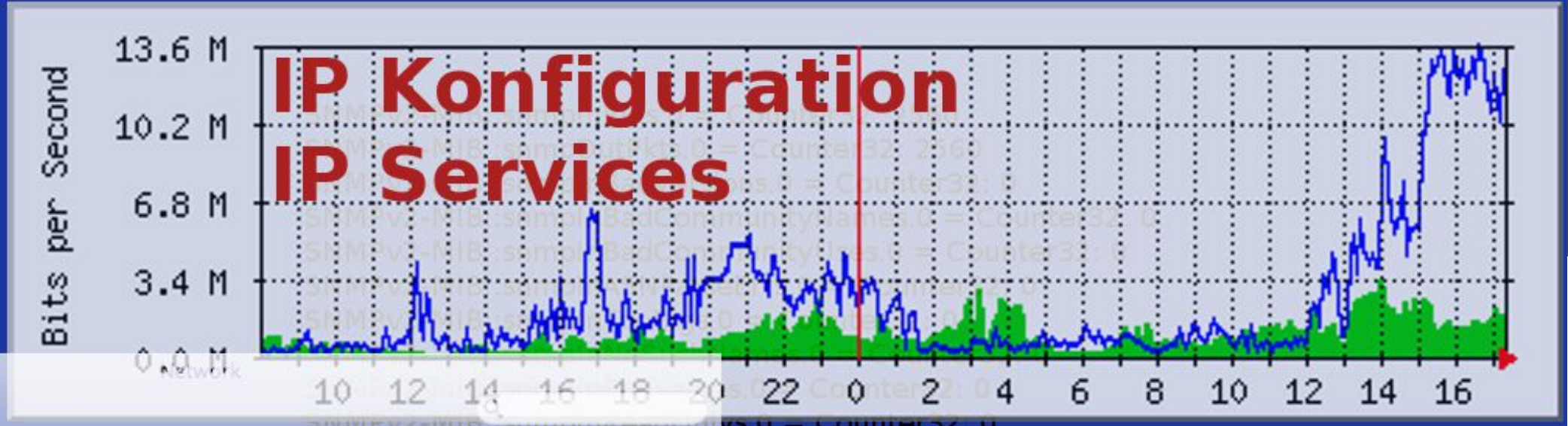
IP Konfiguration
IP Services



ocation: Automatic

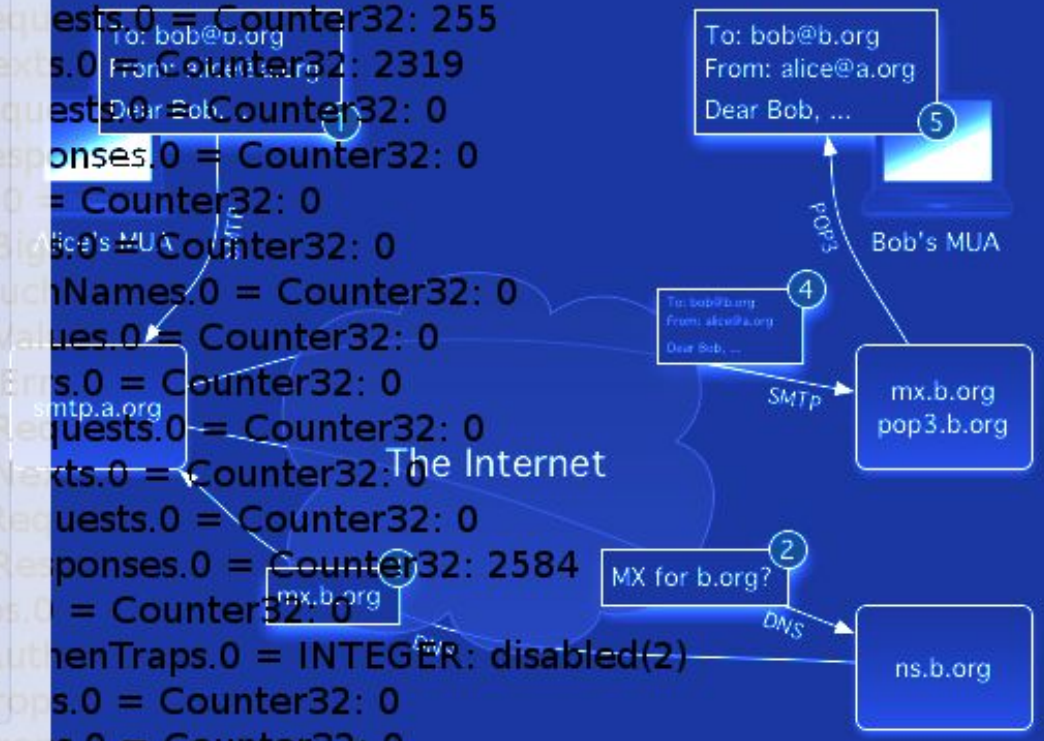
Status: **Connected**
Ethernet is currently active and has the IP address 192.168.23.23.

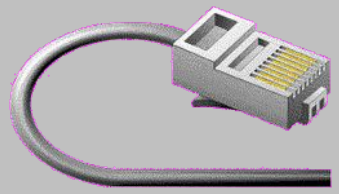
Configure: Using DHCP
IP Address: 192.168.23.23
Subnet Mask: 255.255.255.0
Router: 192.168.23.1
DNS Server: 193.246.253.10, 62.12.130.66
Search Domains: heuer.org, cyberlink.ch, magnet.ch
IPv6 Address: 2001:08a8:0030:...21b:63ff:fe33:e63a

Advanced... ?

t further changes.

Assist me... Revert Apply

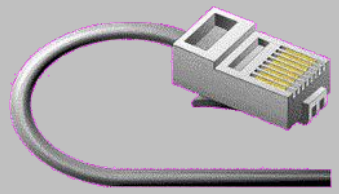




Ziele:

IP Adressen bei einem Host konfigurieren

Funktionsweise von IP basierten Services kennen
(Remote Zugriff, NTP, Mailserver, SNMP,)



IP Adressen konfigurieren

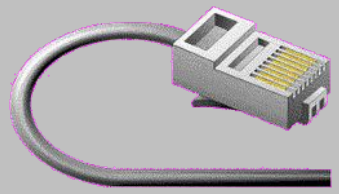
Ziele:

ARP:
MAC-Adresse und IP-Adressen verbinden

RARP:
IP-Adresse anhand der MAC-Adresse konfigurieren

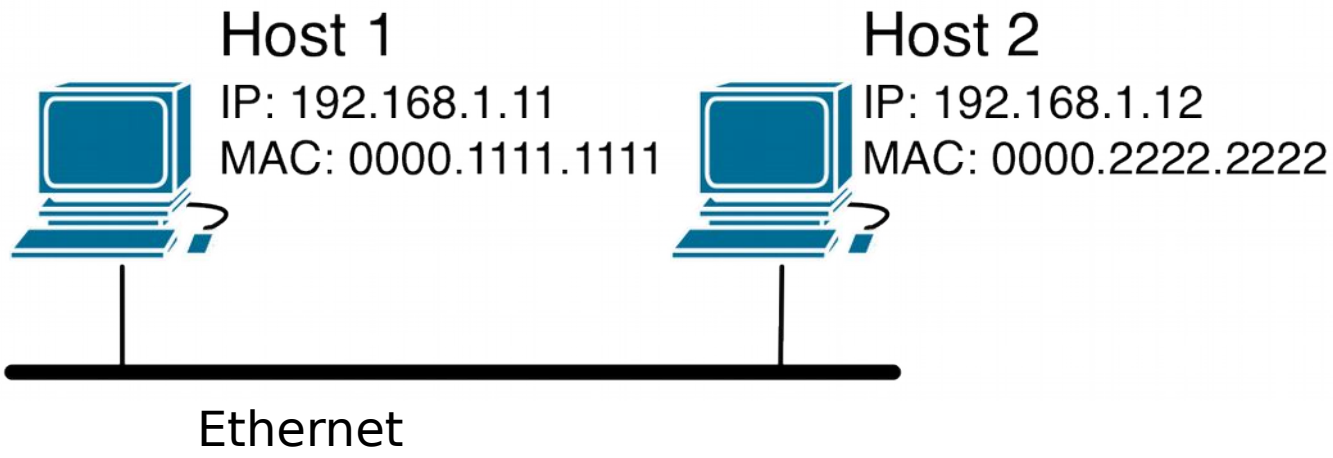
BOOTP, DHCP:
IP-Informationen anhand der MAC-Adresse konfigurieren

APIPA:
IP-Konfiguration für die Insel

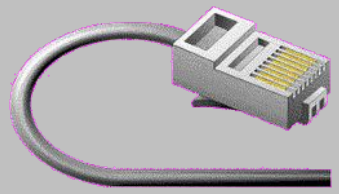


ARP

Wenn Host A an Host B eine Paket senden will, so muss Host A die MAC-Adresse von Host B kennen, um ein Paket senden zu können.

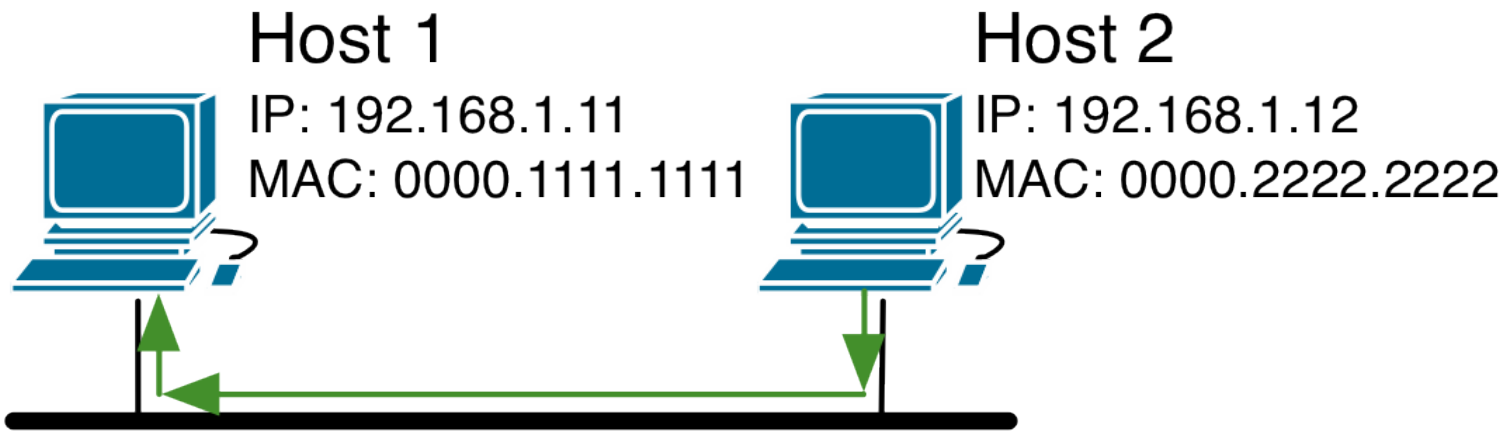


Wie kann Host A die MAC-Adresse von Host B herausfinden?

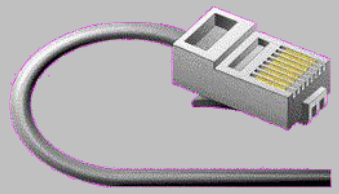


ARP-Replay

Der Rechner, der die gesuchte IP-Adresse konfiguriert hat, beantwortet die Anfrage direkt an Host A mit "192.168.1.20 is at 0000.2222.2222".

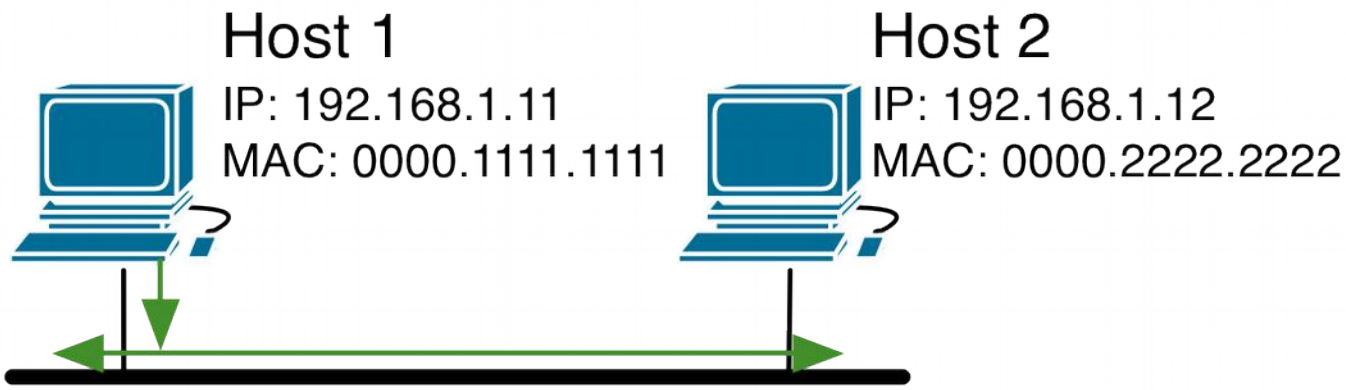


Alle anderen Rechner ignorieren die Anfrage!

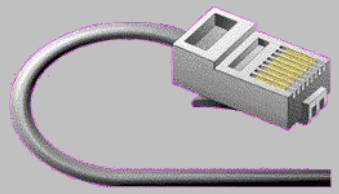


ARP-Request

Host A sendet einen ARP-Request für die IP-Adresse von Host B. Da Host A das Ziel nicht kennt, muss er diesen Request als Broadcast absenden.



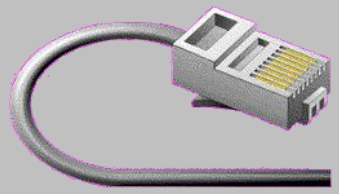
In diesem Request fragt Host A das lokale Netzwerk: "Who has 192.168.1.20?"



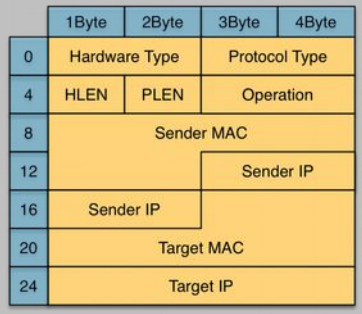
ARP Paket

Ein ARP Paket kennt folgende Felder:

	1Byte	2Byte	3Byte	4Byte
0	Hardware Type		Protocol Type	
4	HLEN	PLEN	Operation	
8	Sender MAC			
12			Sender IP	
16	Sender IP			
20	Target MAC			
24	Target IP			

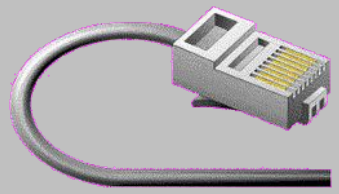


ARP Paket

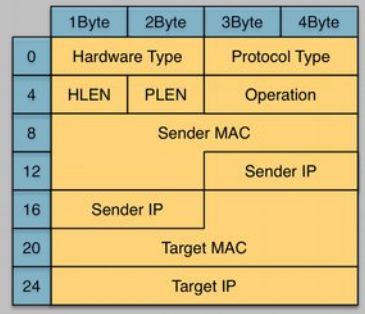


Die Felder haben folgende Bedeutung:

Hardware Type:	Ethernet := 1
Protocol Type:	IP := 0x0800
HLEN Hardware Length:	Ethernet := 6
PLEN Protocol Length:	IP := 4
Operation:	Query := 1; Replay :=2
Sender MAC:	MAC des Senders
Sender IP:	IP des Senders
Target MAC	gesuchte MAC
Target IP:	gesuchte IP



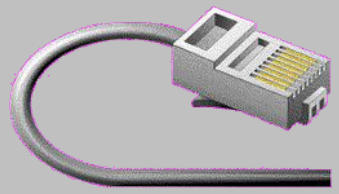
ARP Request



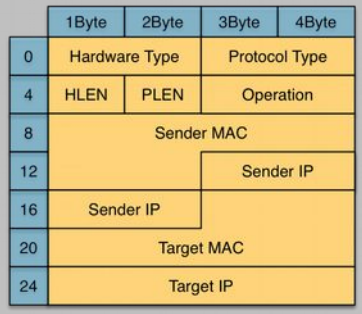
Ein ARP Request sieht wie folgt aus: (aus der Sicht des fragenden Host)

Sender MAC: die eigene MAC_Adresse
Sender IP: die eigene IP-Adresse
Target MAC: 0000.0000.0000
Target IP: die gesuchte IP-Adresse

Dieses ARP-Paket wird an die MAC-Broadcast Adresse gesendet.



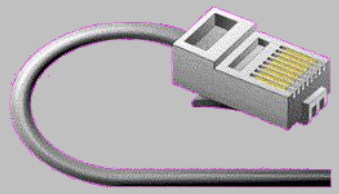
ARP Reply



Ein ARP Reply sieht wie folgt aus: (aus der Sicht des fragenden Host)

Sender MAC:	die gesuchte MAC_Adresse
Sender IP:	die gesuchte IP-Adresse
Target MAC:	die eigene MAC-Adresse
Target IP:	die eigene IP-Adresse

Dieses ARP-Paket wird an die MAC-Adresse des fragenden Host gesendet.

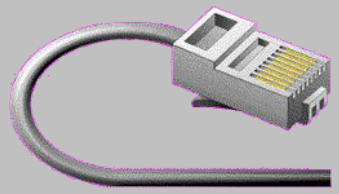


ARP-Tabelle

Damit ein Host nicht dauernd ARP-Requests senden muss speichert der Host die Antwort in seiner ARP-Tabelle ab.

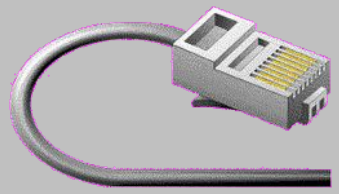
- Einträge in der ARP-Tabelle werden dynamisch eingetragen und nach einer – konfigurierbaren – Zeit wieder gelöscht.

- Je nach System ist der default ARP-Cache Timer zwischen 30 Sekunden (Linuxen) und 4 Stunden (Cisco-Router).



ARP-Tabelle

- Einträge können auch fix eingetragen werden:
  - Um die Sicherheit zu erhöhen (denn ARP-Replay kann jeder senden ...)
  - Um ein Gerät zu konfigurieren.



ARP-Tabelle

UNIX:

```
wally: heuer$ arp -an
? (192.168.23.1) at 8:5b:e:27:49:12 on en0 ifscope [ethernet]
? (192.168.23.12) at 0:5:cd:24:c9:8 on en0 ifscope [ethernet]
```

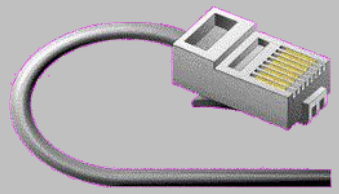
Windows:

```
C:\Users\heuer>arp -a

Interface: 192.168.23.147 --- 0x3
  Internet Address      Physical Address      Type
  192.168.23.1          08-5b-0e-27-49-12     dynamic
  192.168.23.12         00-05-cd-24-c9-08     dynamic
```

Cisco:

```
Router>show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.23.1           -   085b.0e27.4912  ARPA   FastEthernet0
Internet  192.168.23.12          0   0005.cd24.c908  ARPA   FastEthernet0
```

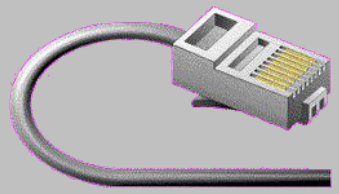


Gratuitous ARP

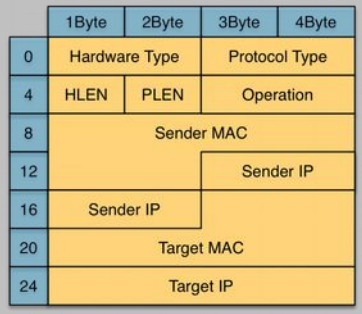
Da die Einträge in der ARP-Tabelle recht langlebig sein können, kann es sein, das ein Host mit einer geänderten MAC-Adresse – aber gleicher IP-Adresse – lange zeit nicht erreichbar ist.

Um das zu verhindern, kann der Host ARP-Requests versenden, in denen er seine neue MAC-Adresse mitteilt ohne dass vorher ein ARP-Request versendet wurde. (gratuitous ARP).

Diese gratuitous ARP Requests sind nicht authentifiziert! (jeder kann solche Pakete versenden ..., zu welchem Zweck auch immer!)



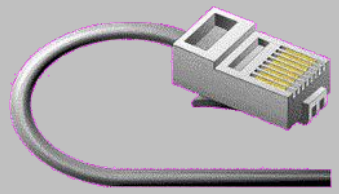
Gratuitous ARP



Ein gratuitous ARP Paket sieht wie folgt aus:

Sender MAC: die eigene MAC_Adresse
Sender IP: die eigene IP-Adresse
Target MAC: die eigene MAC-Adresse
Target IP: die eigene IP-Adresse

Dieses ARP-Paket wird an die MAC-Broadcast Adresse gesendet.

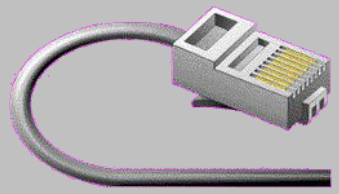


ARP Probe

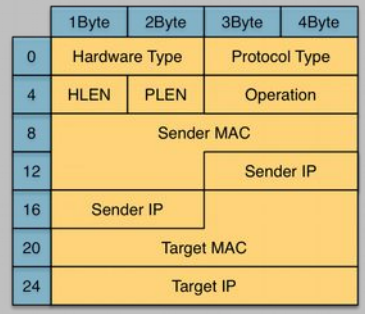
Ein DHCP Client muss testen, ob die vorgeschlagene IP-Adresse eindeutig ist. Dieser Test kann nicht mittels Ping erfolgen (Welche Source Adresse soll der Host denn verwenden?)

Mittels einer ARP Probe kann der Host testen, ob die IP-Adresse bereits verwendet wird.

Dazu wir eine ARP-Probe versendet.



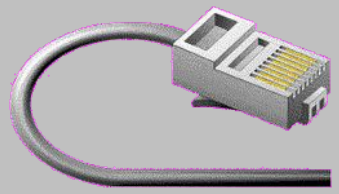
ARP Probe



Die ARP Probe sieht wie folgt aus:

Sender MAC:	die eigene MAC-Adresse
Sender IP:	0.0.0.0
Target MAC:	0000.0000.0000
Target IP:	die zu testende IP-Adresse

Dieses ARP-Paket wird an die MAC-Broadcast Adresse gesendet.



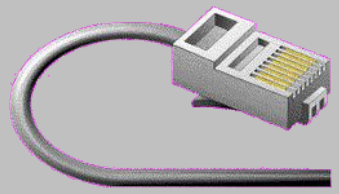
ARP

Mit ARP kann überprüft werden ob ein Host erreichbar ist, ohne dass ICMP Ping eingeschaltet sein muss (leider so eine Krankheit bei Firewalls).

Ist ein ARP-Eintrag für die Zieladresse eingetragen so besteht eine Layer 2 Verbindung zwischen den Geräten.

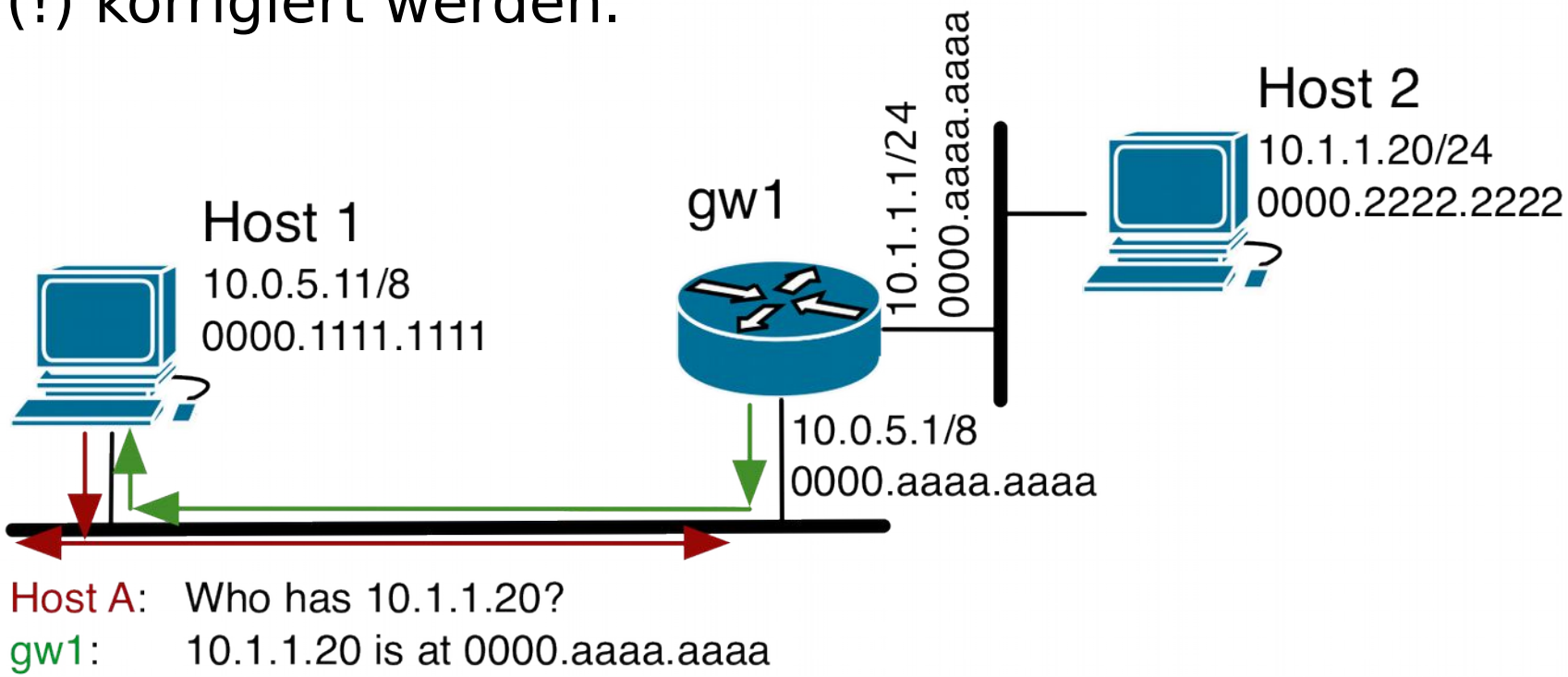
Achtung:

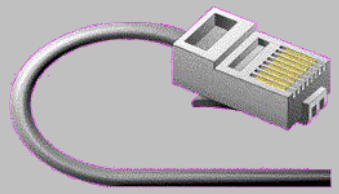
- Vor dem Test sollte ein vorhandener Eintrag zur Sicherheit gelöscht werden.
- Die Verbindung muss in beide Richtungen getestet werden!



Proxy ARP

Hosts können auch ARP-Requests beantworten, wenn sie eine Route für die gesuchte Adresse besitzen. Damit können fehlerhafte IP-Konfiguration teilweise (!) korrigiert werden.





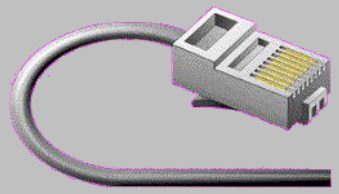
Probleme ...

Bei Verbindungs-Problemen muss sichergestellt sein, dass

- die IP-Konfiguration richtig ist (alle Rechner verwenden unterschiedliche IP-Adressen aus dem gleichen IP Subnetz)
- ARP-Einträge richtig sind.

Mit der ARP-Tabelle können verschiedene Netz-Probleme eingegrenzt werden:

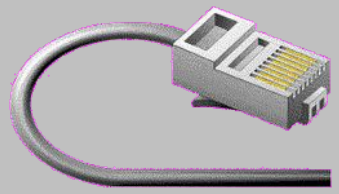
- Habe ich den korrekten ARP-Eintrag vom Ziel-Rechner bzw. default Gateway?
- Hat der Ziel-Rechner bzw. default Gateway meine eigene MAC-Adresse in der ARP-Tabelle?
- Stimmen die MAC-Adressen in der ARP-Tabelle? (wenn nicht, so sind IP-Adressen doppelt vergeben, Dann kann man in den MAC-Adresse-Tabellen der Switches das Gerät lokalisiert werden).



gut zu wissen ...

ARP verwendet einen eigenen Ethertyp (0x0806)

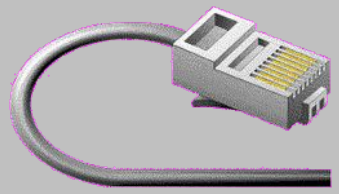
RARP verwendet einen eigenen Ethertyp (0x8035)
RARP kommt später ...



Fragen ?

I like ARP-Jokes, because it's so easy to make them appear to originate from other persons.

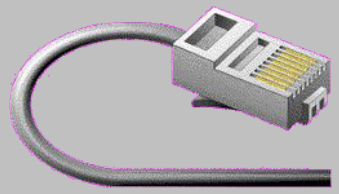
http://hardware.localhost.nl/



IP Konfiguration

Wie bringe ich die IP Konfiguration auf den Rechner?

- **Manuelle Konfiguration**
  - Handarbeit :)
- **Automatische Konfiguration**
  - RARP
  - BOOTP
  - DHCP
  - Zero Konfiguration (APIPA)



Manuelle IP Konfiguration

Unix:

```
ifconfig <iface> <ipaddr/netmask> broadcast <broadcastaddr>
ifconfig <iface> <ipaddr> netmask <netmask> broadcast <broadcastaddr>
ifconfig <iface>
```

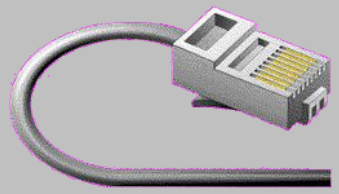
Generell müssen nur die von den IP-Klassen abweichende Netzmasken / Broadcastadressen angeben werden

Defaultrouten müssen separat (`route add default gw <IP-ADDR>`) konfiguriert werden.

DNS-Server werden in der Datei `/etc/resolv.conf` eingetragen.

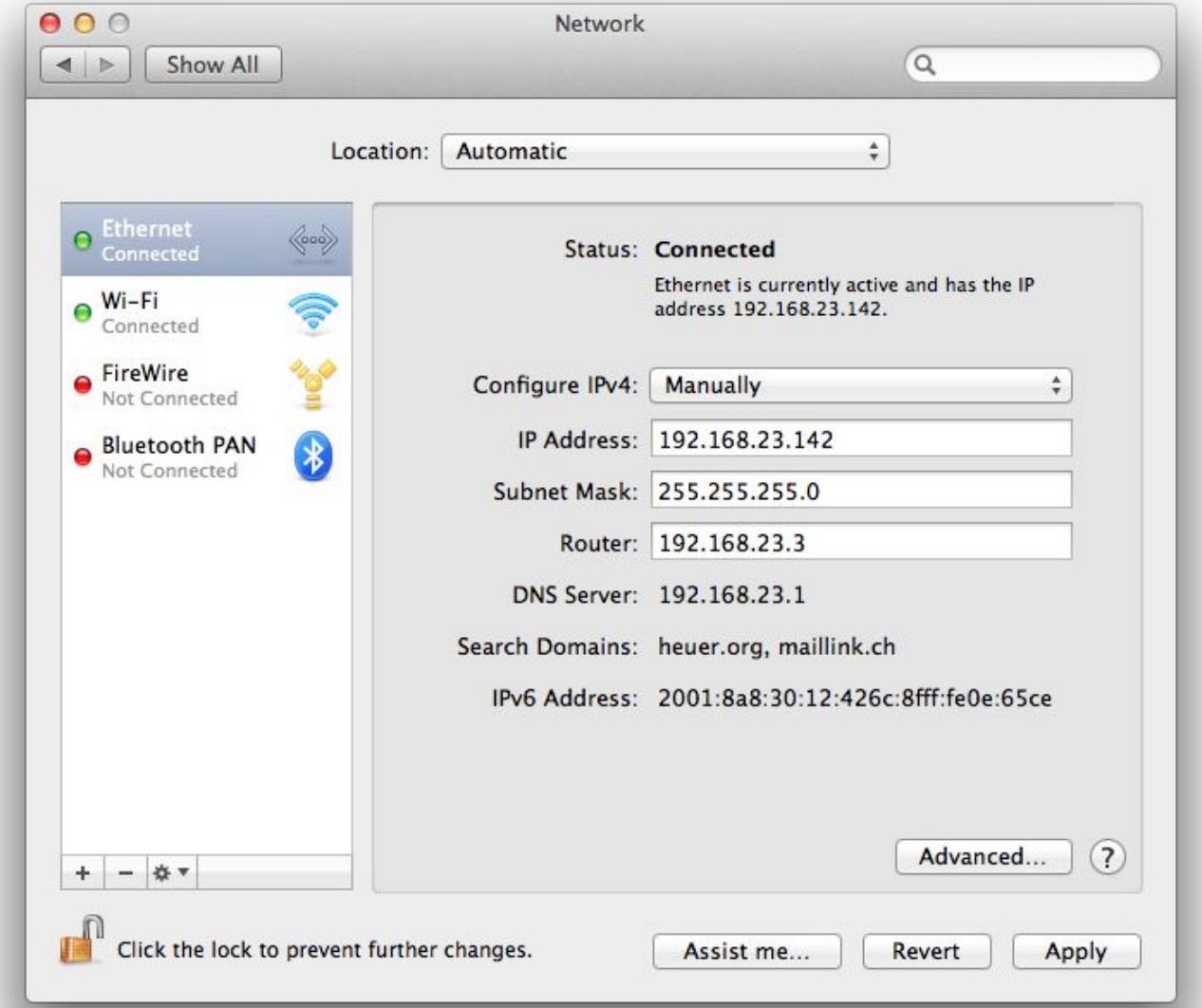
Router:

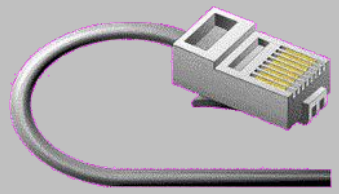
```
router# configure terminal
router(config)#interface serial 1
router(config-if)#ip address 172.26.2.1 255.255.255.252
router(config-if)#end
```



Manuelle IP Konfiguration

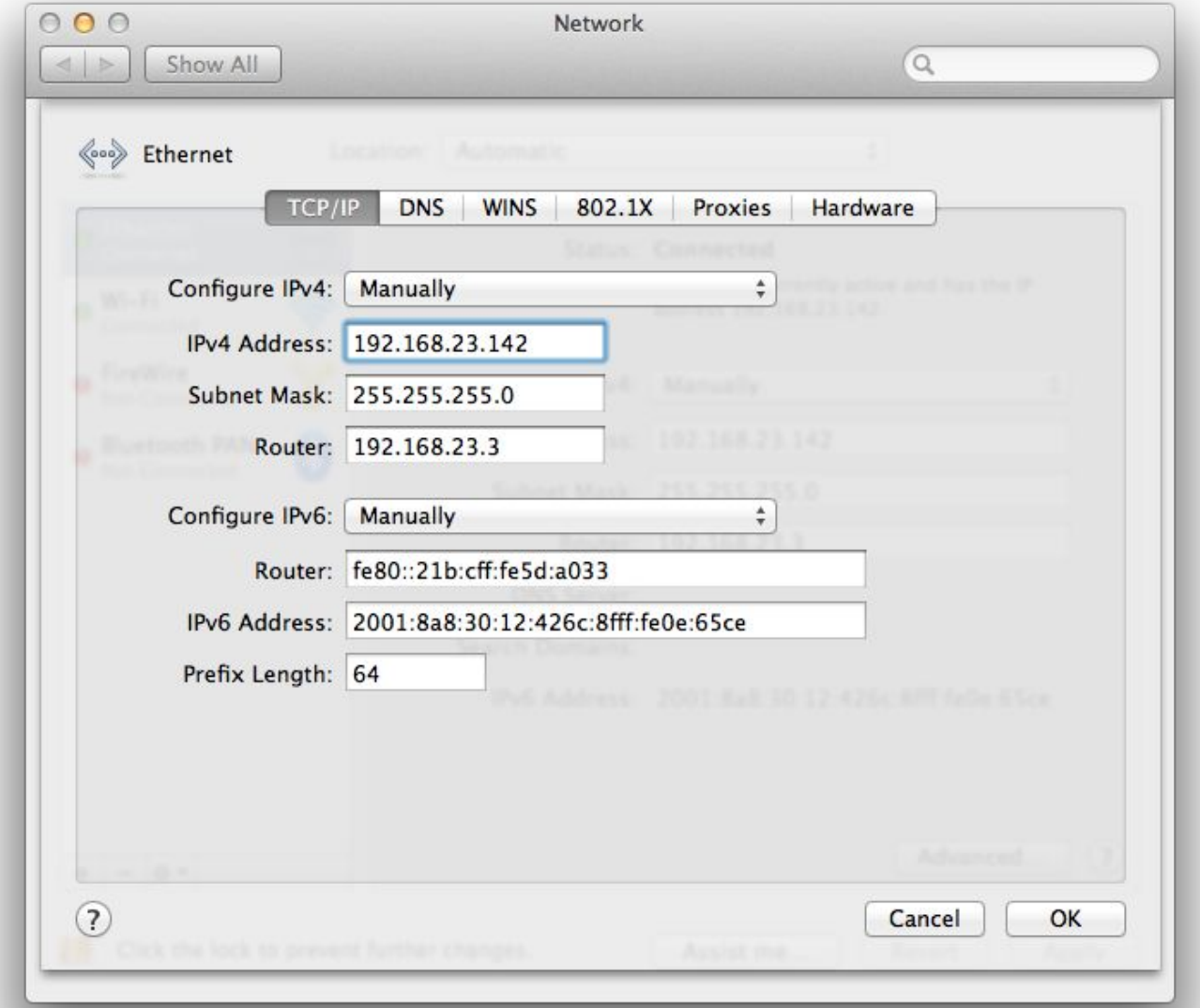
Mac OS X:

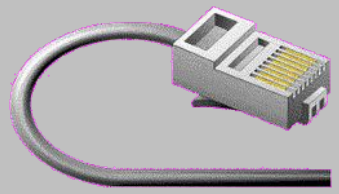




Manuelle IP Konfiguration

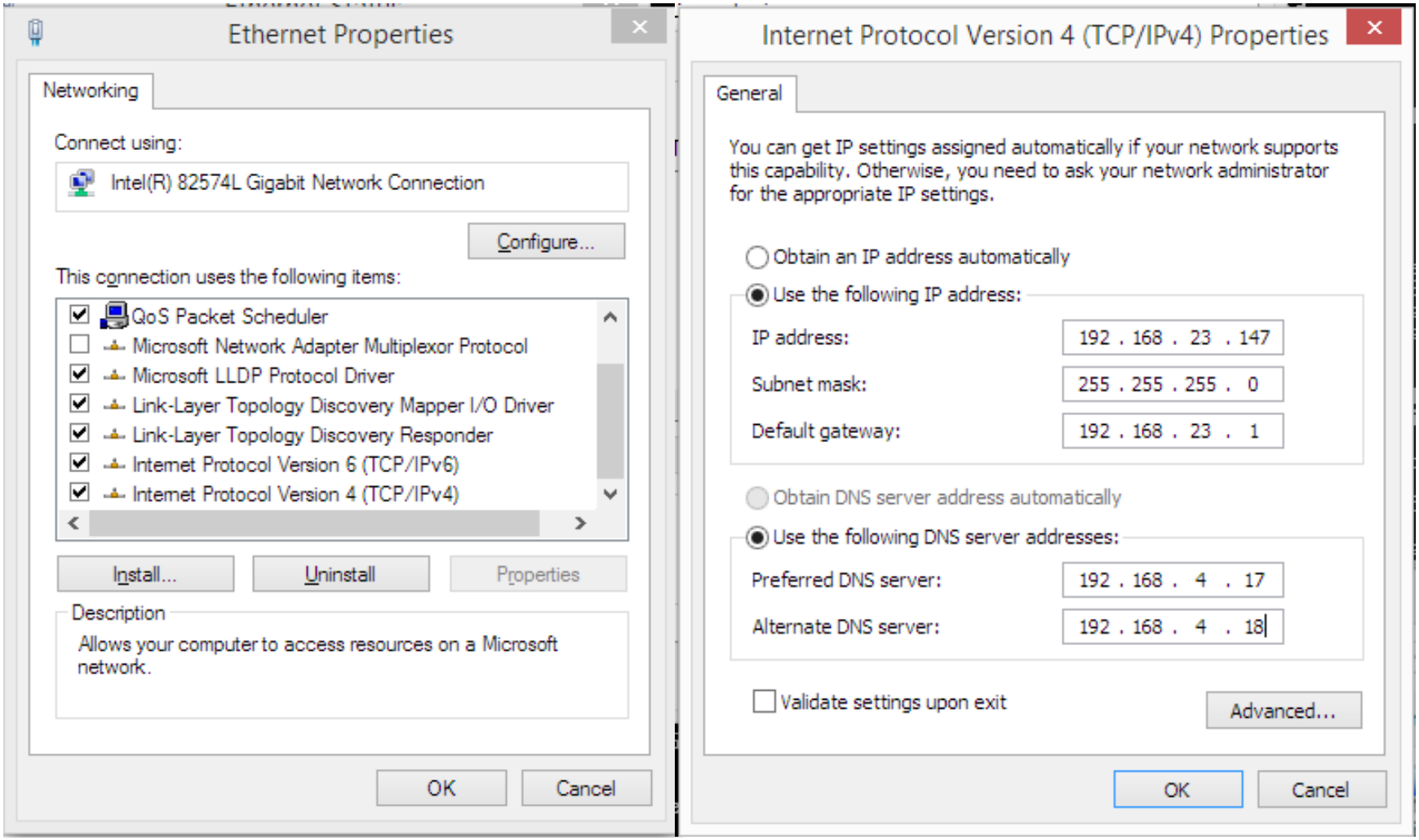
Mac OS X:

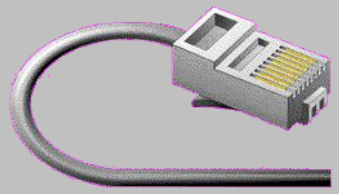




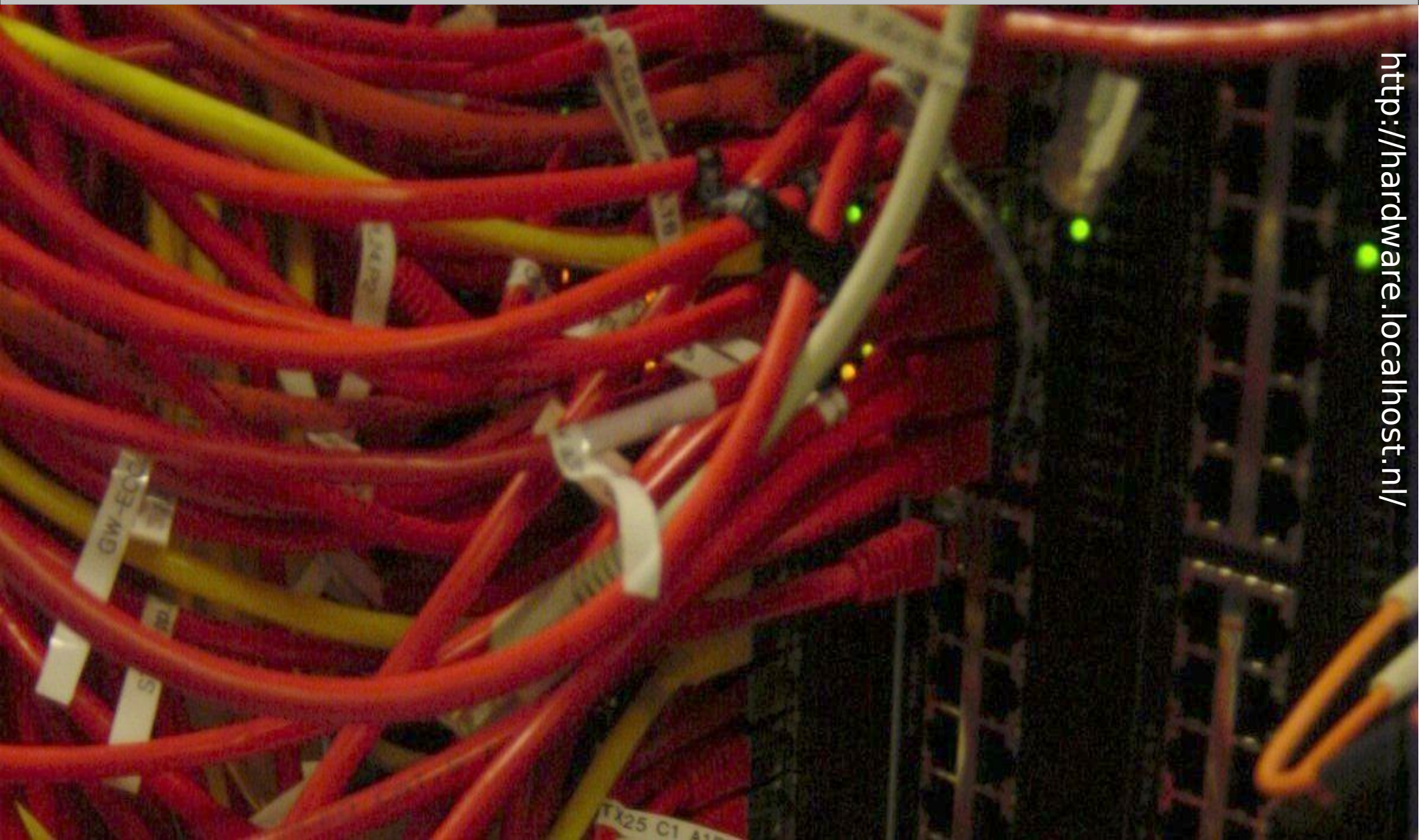
Manuelle IP Konfiguration

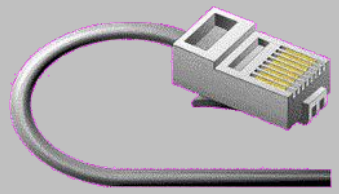
Windows:





Fragen ?





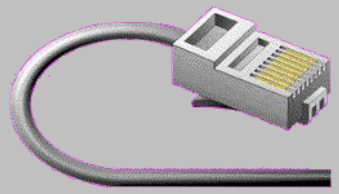
Reverse ARP

RARP

In der Regel verwendet ein Rechner ARP um die MAC-Adresse eines anderen Rechners zu finden. wenn der Rechner nach seiner eigenen IP-Adresse sucht, so kann er das **Reverse Address Resolution Protocol** (**RARP**) verwenden.

Damit RARP funktioniert, muss innerhalb der Ethernet Broadcast-Domain ein RARP-Server installiert und konfiguriert sein.

RARP kann nur die IP-Adresse übermitteln. Alle zusätzlichen Informationen müssen anschliessen von Hand konfiguriert werden!

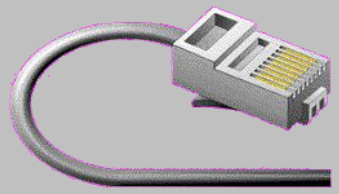


Bootstrap Protocol (BOOTP)

RARP ist sehr stark limitiert (Nur IP-Adresse, nur im lokalen Netz). Um diese Einschränkungen zu umgehen wurde BOOTP erfunden.

Der Client sendet einen BOOTP-Request als UDP Broadcast. Der BOOTP-Server beantwortet den Request mit einem BOOTP-Replay – sofern er den Client in seiner Datenbank findet.

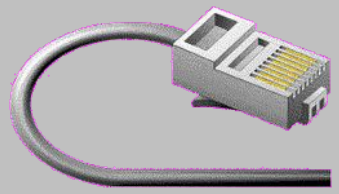
Das BOOTP-Replay Paket kann neben der IP-Adresse auch die Netzmaske, DNS-Server, Default Gateway und viele andere Parameter übertragen.



Bootstrap Protocol (BOOTP)

Da der BOOTP-Request als UDP-Broadcast gesendet wird, ist es möglich, dass dieses Paket an einen entfernten BOOTP-Server weiter geleitet werden kann. Es ist pro LAN Segment (Broadcast Domain) nicht mehr ein einzelner, dedizierter Server notwendig. Dazu muss der Router im Segment konfiguriert sein.

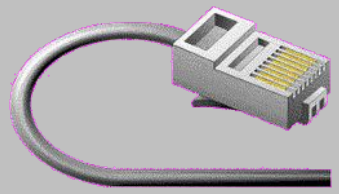
Der Administrator vom BOOTP-Server muss jede MAC-Adresse **vorgängig** in der Konfiguration vom BOOTP-Server registrieren sein um die notwendigen Parameter zuweisen.



Bootstrap Protocol (BOOTP)

BOOTP verwendet einen Schlüssel um den Klient zu identifizieren. In 99.9% aller Fälle ist das die Ethernet-Adresse der Netzwerk-Schnittstelle, die konfiguriert werden soll.

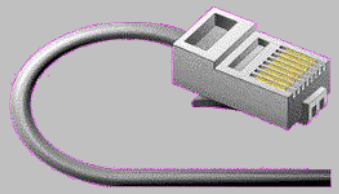
Wird ein Rechner in ein anderes Netz-Segment gezügelt, so muss dies auch beim BOOTP-Server angepasst werden, da der Rechner sonst die falschen IP-Parameter (die vom alten Standort) zugewiesen bekommt.



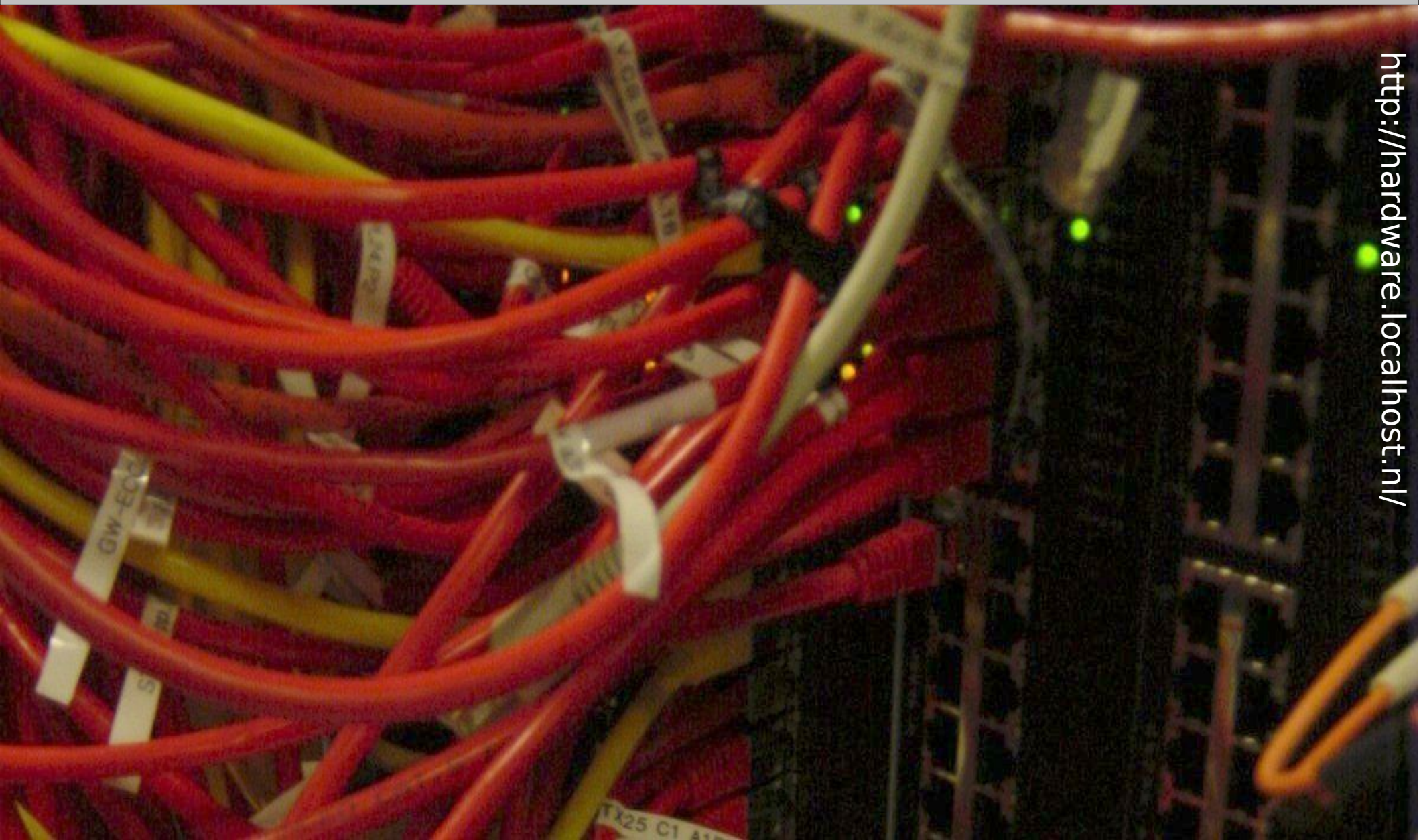
BOOTP Ablauf

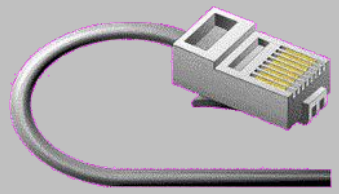
Ablauf einer BOOTP Session:

159	2012-03-24 15:12:27	0.0.0.0	255.255.255.255	BOOTP	Boot Request from 40:6c:8
160	2012-03-24 15:12:27	192.168.23.3	192.168.23.145	BOOTP	Boot Reply



Fragen?





DHCP

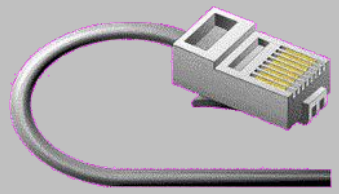
DHCP basiert auf BOOTP.

Der Unterschied ist, dass der DHCP-Server IP-Adressen an die Clients dynamisch verteilen kann.

Die Clients müssen nicht mehr im voraus beim Server registriert sein.

Der Server muss jedoch IP-Pools besitzen aus denen er die IP-Adressen 'verteilen' kann.

Da die IP-Adresse nicht mehr fix zugeteilt ist, ist das Protokoll zwischen dem DHCP-Client und -Server komplizierter als bei BOOTP oder RARP:



DHCP Ablauf

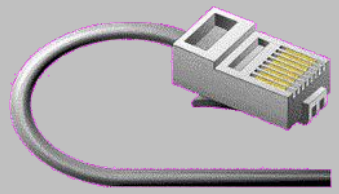
Der Client sendet eine DHCP-**DISCOVERY** Meldung

Der DHCP-Server antwortet mit einer DHCP-**OFFER** Meldung

Der Client überprüft, ob die IP-Adresse aus der OFFER Meldung brauchbar ist. Wenn die IP OK ist, so sendet der Client dem Server eine DHCP-**REQUEST** Meldung zu.

Der Server bestätigt dem Client die Parameter mit einer DHCP-**ACK** Meldung. Der Client kann die Parameter verwenden.

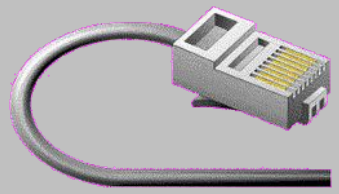
Ist die IP-Adresse aus der DHCP-OFFER nicht verwendbar, so sendet der Client eine DHCP-**DECLINE** Meldung zurück.



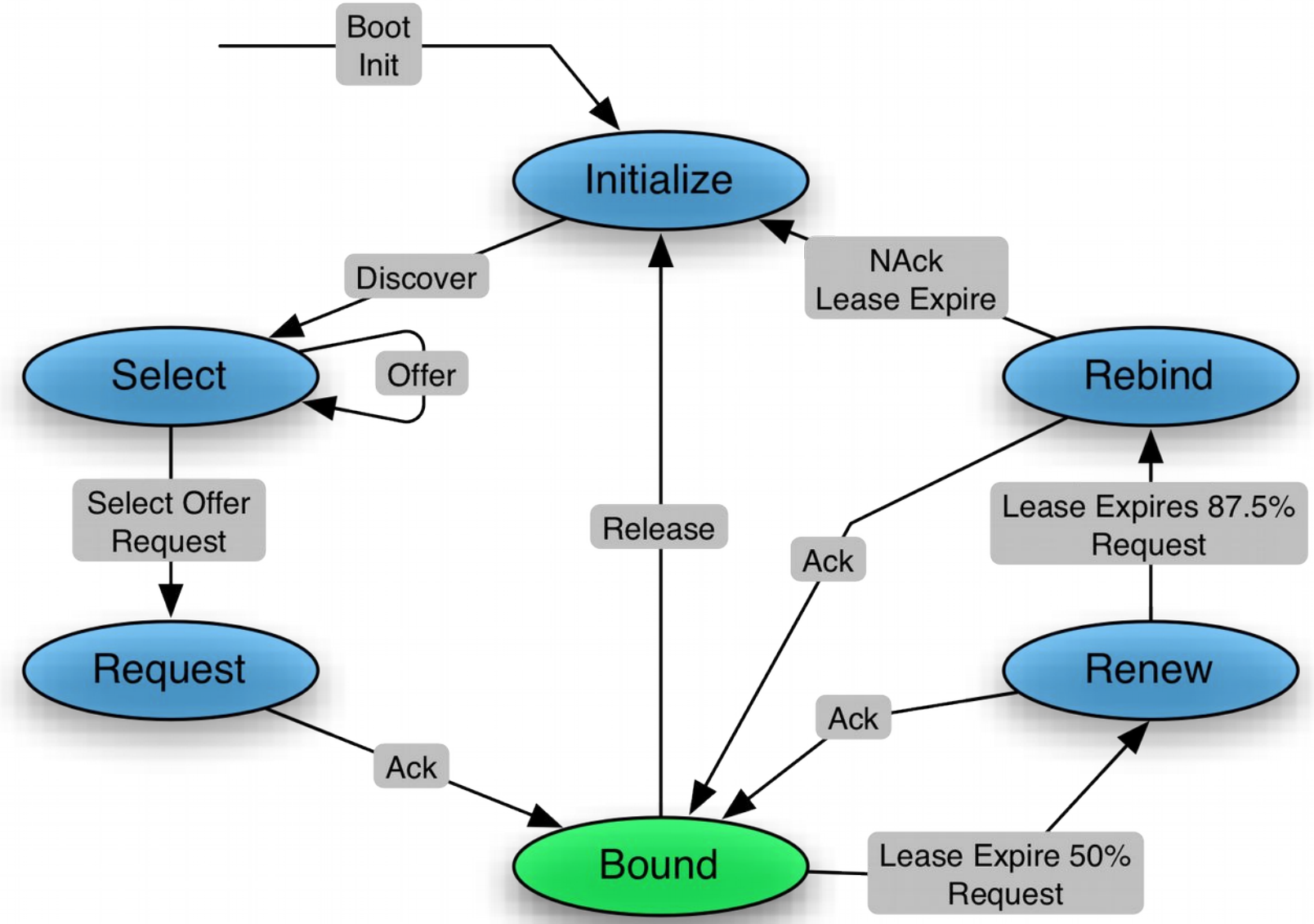
DHCP Ablauf

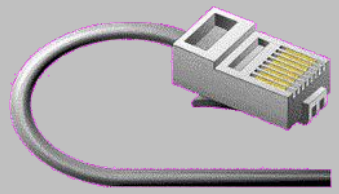
Bekommt der DHCP-Client mehrere DHCP-OFFER Meldungen - weil beispielsweise mehrerer DHCP-Server am Netz sind - so gewinnt der schnellste Server, der eine passende Offer sendet.

Tipp: Versuchen sie mal mit Wireshark einen solchen Ablauf aufzuzeichnen. Starten sie Wireshark und geben sie ihre DHCP-Lease mit `ipconfig /release` frei und holen sie ich mit `ipconfig /renew` eine 'neue' IP-Adresse



DHCP States

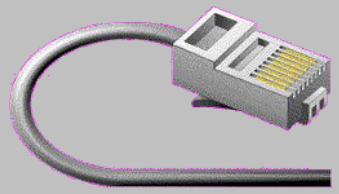

Boot
Init
Initialize
Discover
NAck
Lease Expire
Select
Offer
Rebind
Select Offer
Request
Release
Ack
Lease Expires 87.5%
Request
Request
Renew
Ack
Ack
Bound
Lease Expire 50%
Request



DHCP Meldungen

DHCP kennt folgende Meldungs Typen:

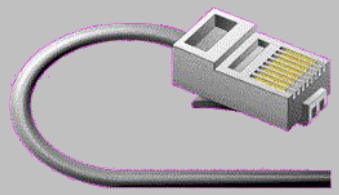
DHCP Discover	(Client > Server)
DHCP Offer	(Server > Client)
DHCP Request	(Client > Server)
DHCP Decline	(Client > Server)
DHCP Ack	(Server > Client)
DHCK NAck	(Server > Client)
DHCP Release	(Client > Server)



DHCP Ablauf

Ablauf einer DHCP Boot Session

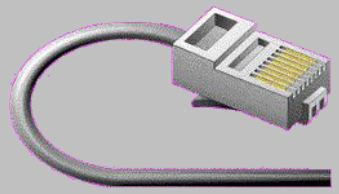
1966	2012-03-24 15:18:03	0.0.0.0	255.255.255.255	DHCP	DHCP Discover
1969	2012-03-24 15:18:03	192.168.23.3	192.168.23.148	DHCP	DHCP Offer
1974	2012-03-24 15:18:04	0.0.0.0	255.255.255.255	DHCP	DHCP Request
1975	2012-03-24 15:18:04	192.168.23.3	192.168.23.148	DHCP	DHCP ACK



DHCP Discover

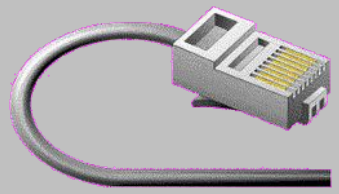
```
▷ Frame 1959: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
▷ Ethernet II, Src: Apple_0e:65:ce (40:6c:8f:0e:65:ce), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
▷ User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
▽ Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x43041c4a
    Seconds elapsed: 0
  ▷ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: Apple_0e:65:ce (40:6c:8f:0e:65:ce)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▷ Option: (t=53,l=1) DHCP Message Type = DHCP Discover
  ▷ Option: (t=55,l=9) Parameter Request List
  ▷ Option: (t=57,l=2) Maximum DHCP Message Size = 1500
  ▷ Option: (t=61,l=9) Client identifier
  ▷ Option: (t=51,l=4) IP Address Lease Time = 90 days
  ▷ Option: (t=12,l=5) Host Name = "wally"
    End Option
    Padding
```

5



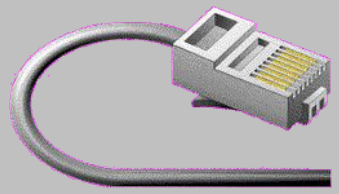
DHCP Offer

```
▷ Frame 1969: 349 bytes on wire (2792 bits), 349 bytes captured (2792 bits)
▷ Ethernet II, Src: Cisco_5d:a0:33 (00:1b:0c:5d:a0:33), Dst: Apple_0e:65:ce (40:6c:8f:0e:65:ce)
▷ Internet Protocol Version 4, Src: 192.168.23.3 (192.168.23.3), Dst: 192.168.23.148 (192.168.23.148)
▷ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
▽ Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x43041c4a
    Seconds elapsed: 0
  ▷ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.23.148 (192.168.23.148)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: Apple_0e:65:ce (40:6c:8f:0e:65:ce)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▷ Option: (t=53,l=1) DHCP Message Type = DHCP Offer
  ▷ Option: (t=54,l=4) DHCP Server Identifier = 192.168.23.3
  ▷ Option: (t=51,l=4) IP Address Lease Time = 1 day
  ▷ Option: (t=58,l=4) Renewal Time Value = 12 hours
  ▷ Option: (t=59,l=4) Rebinding Time Value = 21 hours
  ▷ Option: (t=1,l=4) Subnet Mask = 255.255.255.0
  ▷ Option: (t=15,l=11) Domain Name = "maillink.ch"
  ▷ Option: (t=3,l=4) Router = 192.168.23.3
  ▷ Option: (t=6,l=12) Domain Name Server
    End Option
```



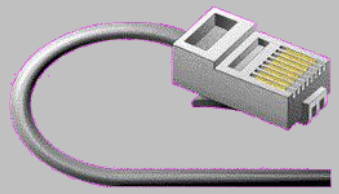
DHCP Request

```
▷ Frame 1974: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
▷ Ethernet II, Src: Apple_0e:65:ce (40:6c:8f:0e:65:ce), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
▷ User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
▽ Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x43041c4a
    Seconds elapsed: 3
  ▷ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: Apple_0e:65:ce (40:6c:8f:0e:65:ce)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▷ Option: (t=53,l=1) DHCP Message Type = DHCP Request
  ▷ Option: (t=55,l=9) Parameter Request List
  ▷ Option: (t=57,l=2) Maximum DHCP Message Size = 1500
  ▷ Option: (t=61,l=9) Client identifier
  ▷ Option: (t=50,l=4) Requested IP Address = 192.168.23.148
  ▷ Option: (t=54,l=4) DHCP Server Identifier = 192.168.23.3
  ▷ Option: (t=12,l=5) Host Name = "wally"
    End Option
    Padding
```



DHCP Ack

```
▷ Frame 1975: 349 bytes on wire (2792 bits), 349 bytes captured (2792 bits)
▷ Ethernet II, Src: Cisco_5d:a0:33 (00:1b:0c:5d:a0:33), Dst: Apple_0e:65:ce (40:6c:8f:0e:65:ce)
▷ Internet Protocol Version 4, Src: 192.168.23.3 (192.168.23.3), Dst: 192.168.23.148 (192.168.23.148)
▷ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
▽ Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x43041c4a
    Seconds elapsed: 0
  ▷ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.23.148 (192.168.23.148)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: Apple_0e:65:ce (40:6c:8f:0e:65:ce)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▷ Option: (t=53,l=1) DHCP Message Type = DHCP ACK
  ▷ Option: (t=54,l=4) DHCP Server Identifier = 192.168.23.3
  ▷ Option: (t=51,l=4) IP Address Lease Time = 1 day
  ▷ Option: (t=58,l=4) Renewal Time Value = 12 hours
  ▷ Option: (t=59,l=4) Rebinding Time Value = 21 hours
  ▷ Option: (t=1,l=4) Subnet Mask = 255.255.255.0
  ▷ Option: (t=15,l=11) Domain Name = "maillink.ch"
  ▷ Option: (t=3,l=4) Router = 192.168.23.3
  ▷ Option: (t=6,l=12) Domain Name Server
    End Option
```



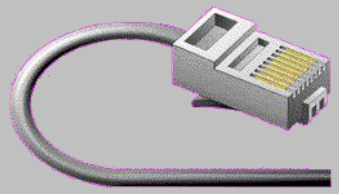
DHCP-Server

DHCP Server müssen nicht mehr jeden Client im voraus zu kennen. Die DHCP-Server brauchen mindestens einen IP-Pool um IP-Adresse zu verteilen.

In den folgenden Beispielen ist der Pool immer 172.16.84.32 - 172.16.84.63

Zyxel Router:

```
DHCP Setup
 DHCP= Server
 Client IP Pool Starting Address= 172.16.84.32
 Size of Client IP Pool= 32
 Primary DNS Server= 62.12.130.66
 Secondary DNS Server= 193.246.253.10
 Remote DHCP Server= N/A
```



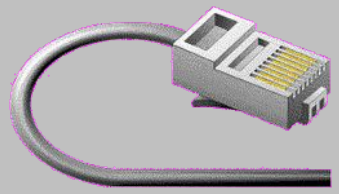
DHCP-Server

Cisco:

```
ip dhcp exclude-address 172.16.1.0 172.16.1.31
ip dhcp exclude-address 172.16.84.64 172.16.84.255
ip dhcp pool users
   network 172.16.84.0 255.255.252.0
   domain-name example.com
   default-router 172.16.84.1
   dns-server 62.12.130.66 193.246.253.10
```

Unix:

```
subnet 172.16.84.0 netmask 255.255.255.0 {
  option domain-name "example.com";
  range 172.16.84.32 172.16.84.63;
  option broadcast-address 172.16.84.255;
  option routers 172.16.84.1;
  option subnet-mask 255.255.255.0;
  max-lease-time 7200;
}
```



DHCP-Server (Probleme)

Wen im Netz ein falsch konfiguriert DHCP-Server zu finden ist, führt das zu grossen Problemen. Von

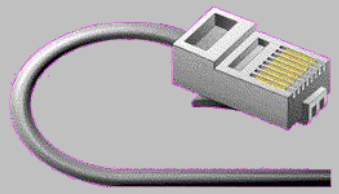
Der Client bekommt eine falsche IP-Adresse, die Kommunikation funktioniert nicht.

Bis zu

Der Traffic kann abgehört werden,

Ist das ganze Spektrum möglich!

Moderne, managed Switches können dies verhindern!



DHCP-Server Relay

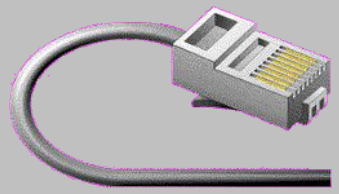
DHCP Server müssen nicht zwingend im selben Netz wie die Clients sein.

Damit dies funktioniert muss in dem Client Netz ein DHCP Relay vorhanden sein. Dies ist meistens der Router in dem Netz.

Das Relay leitet die DHCP-Discover Meldungen als unicast an seine konfigurierten DHCP-Server weiter.

Die DHCP Server senden die Antwort an das Relay zurück.

Das Relay leitet die Antwort an den Client weiter

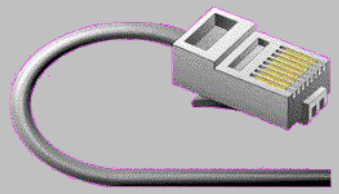


DHCP-Server Relay

Damit der DHCP-Server weiss für welches Netz er eine Lease vergeben muss, fügt das DHCP-Relay die Interface Adresse in die Option Relay agent IP Address (GIADDR) des DHCP-Recover Paktes ein.

Der DHCP-Server muss pro Netz einen passenden DHCP-Pool konfiguriert haben (die IP, die im Feld GIADDR steht muss in einen Pool passen).

Es können mehrere DHCP Server angegeben werden um eine Redundanz der DHCP-Server zu implementieren.



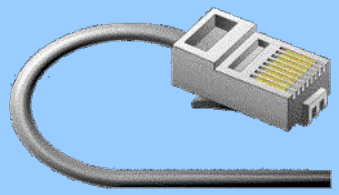
APIPA / Zerokonfig

Wenn ein DHCP-Client keine IP-Adresse beziehen kann, so vergibt sich der DHCP-Client eine beliebige Adresse aus dem Netz 169.254.0.0/16.

Der ganze Vorgang ist sehr träge, da verschiedene Time-Outs abgewartet werden müssen.

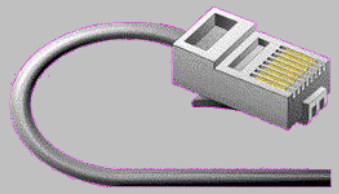
Da sich der Klient eine einzige Adresse selber zu-ordnet, eignet sich diese Methode nur für kleine, iso-lierte Netze. Zusätzlich müssen Dienste vorhanden sein, die die Host-, Service-Namen via Broad-/ Multi-cast announcen.

Das Netz 169.254.0.0/16 ist wie die RFC-1918 Netze nicht in der globalen Routing Tabelle enthalten.



IP Math Hausaufgaben

Lösen sie das das DHCP Labor im netlabor.ch

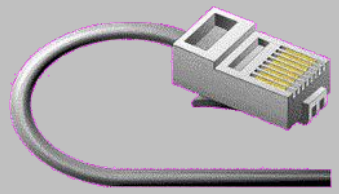


Fragen?

A DHCP packet walks into a bar and asks for a beer.
Bartender says: “here, but I’ll need that back in an hour!

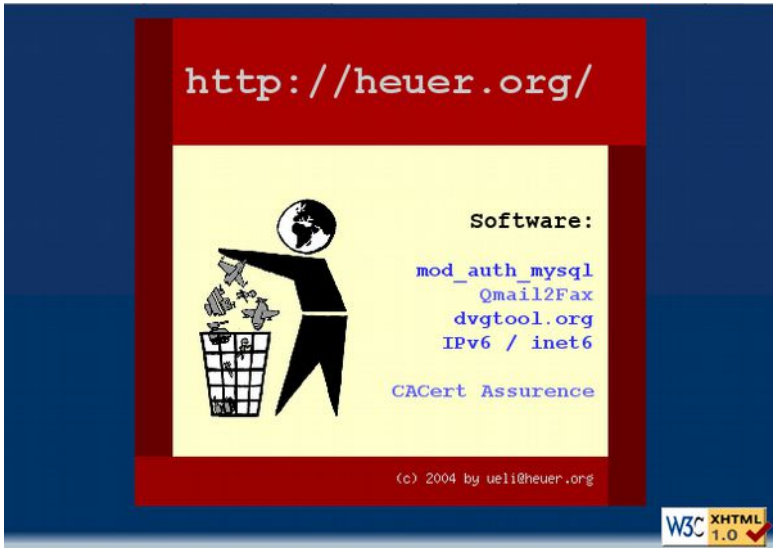
I tried to DISCover some DHCP jokes, but nobody was OFFERing any. When I REQuested some, I wasn't even ACKnowleged.

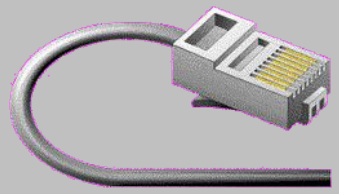
http://hardware.localhost.nl/



Domain Name Service

Ohne die DNS Datenbank wäre das Internet sehr viel unhandlicher:
Anstelle von http://www.heuer.org/ müsste man http://62.48.3.35/ eingeben - und dann wird es erst noch nicht funktionieren :(



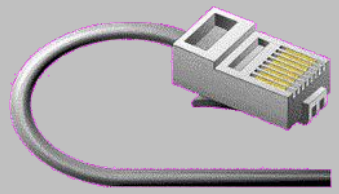


Domain Name Service

Namen sind für uns viel einfacher zu merken als eine IP-Adresse.
Mit der zunehmenden Verbreitung von IPv6 werden Hostnamen noch viel wichtiger.

`ssh guybrush.maillink.ch` kann man sich merken
`ssh 2001:8a8:30:11::2` geht vielleicht noch, aber bei
`ssh 2001:8a8:30:11:e2cb:4eff:fe9d:b99a` ist es definitiv nicht mehr möglich!

Diese Problematik wurde schon sehr früh bei der Entwicklung des Internets erkannt und gelöst

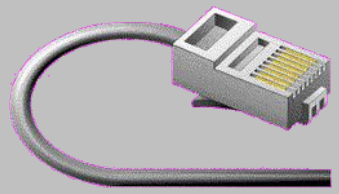


Die Hosts Datei

Als erste Lösung wurde die hosts-Datei verwendet. Diese Datei wurde entweder mit dem Namen durchsucht um die IP-Adressen zu suchen oder mit der IP-Adresse um den Namen des Rechners zu finden.

```
# Adresse           FQDN                      Alias-Namen
127.0.0.1           localhost
::1                 localhost
2001:8a8:30:11::2   guybrush.maillink.ch      guybrush
212.55.196.74       guybrush.maillink.ch      guybrush
192.168.23.1        guybrush.maillink.ch      guybrush

192.168.42.2        toaster.maillink.ch       toaster
192.168.42.23       wally.maillink.ch          wally
```

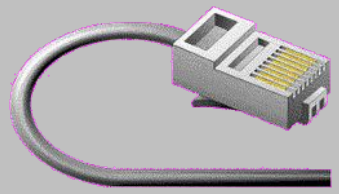


Domain Name Service

Die `hosts`-Datei wurde am Anfang vom Internet von Rechner zu Rechner kopiert. Solange das Netz nur ein paar dutzende Hosts hatte funktioniert das gut.

Als das Netz grösser wurde, war dies sehr umständlich und fehleranfällig:

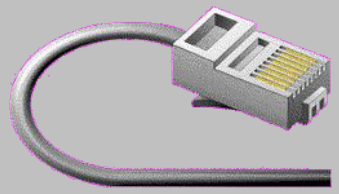
- Der Hostnamen muss eindeutig sein.
- Das File wird mit der Zeit viel zu gross und die Suche im File ist ineffizient.
- Der administrative Aufwand wird zu gross.



Domain Name Service

Um diese Probleme zu lösen wurde eine verteilte, hierarchische Datenbank entwickelt.

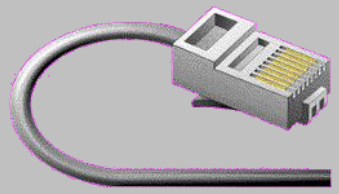
- Durch das hierarchische Konzept, muss nur der Administrator einer Zone sicherstellen, dass der Hostname eindeutig ist.
- Durch die Verteilung der Datenbank, verteilt sich die Last der Abfragen auf viele Servern und die Administration wird einfacher.



Hierarchische Konzept

Damit die Informationen verteilt administriert werden können musste das Konzept. Das hinter der flachen **`hosts`**-Datei zugrunde liegt, modifiziert werden.

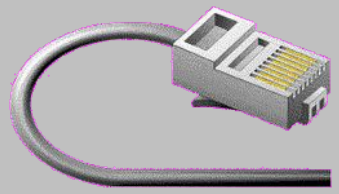
Zum Hostnamen wurden Domain-Namen hinzugefügt.
Die Domain-Namen wurden hierarchisch aufgebaut.



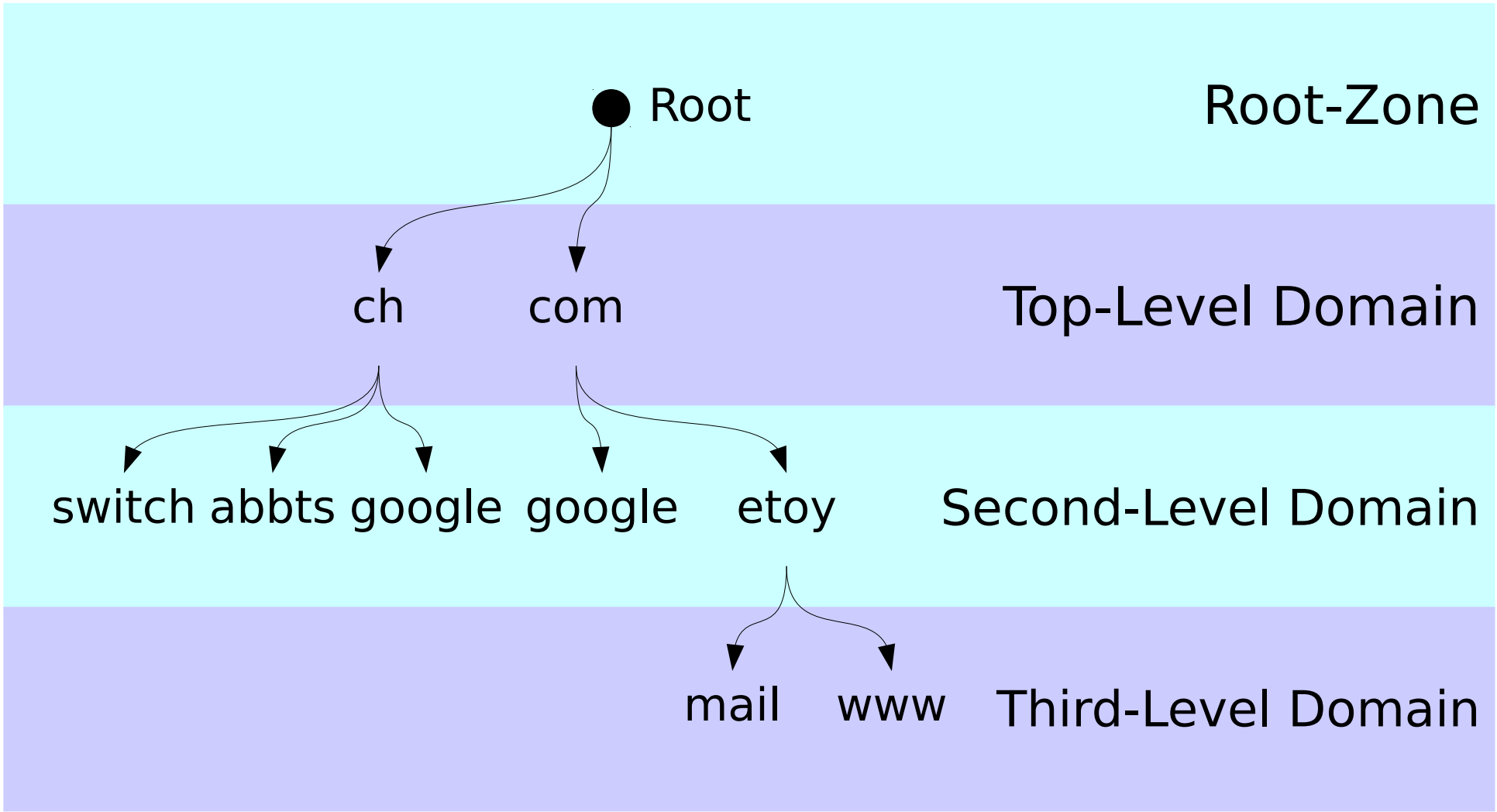
Hierarchische Konzept

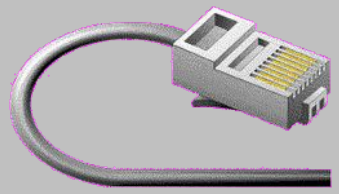
Von der Wurzel (root) aus werden immer weitere Namens teile hintereinander angehängt.

Der komplette Domain Namen (**F**ull **Q**ualified **D**omain **N**ame, FQDN) wird vom Blatt des Baums gegen die Wurzel gebildet und jeder Namen wird durch einen Punkt abgetrennt



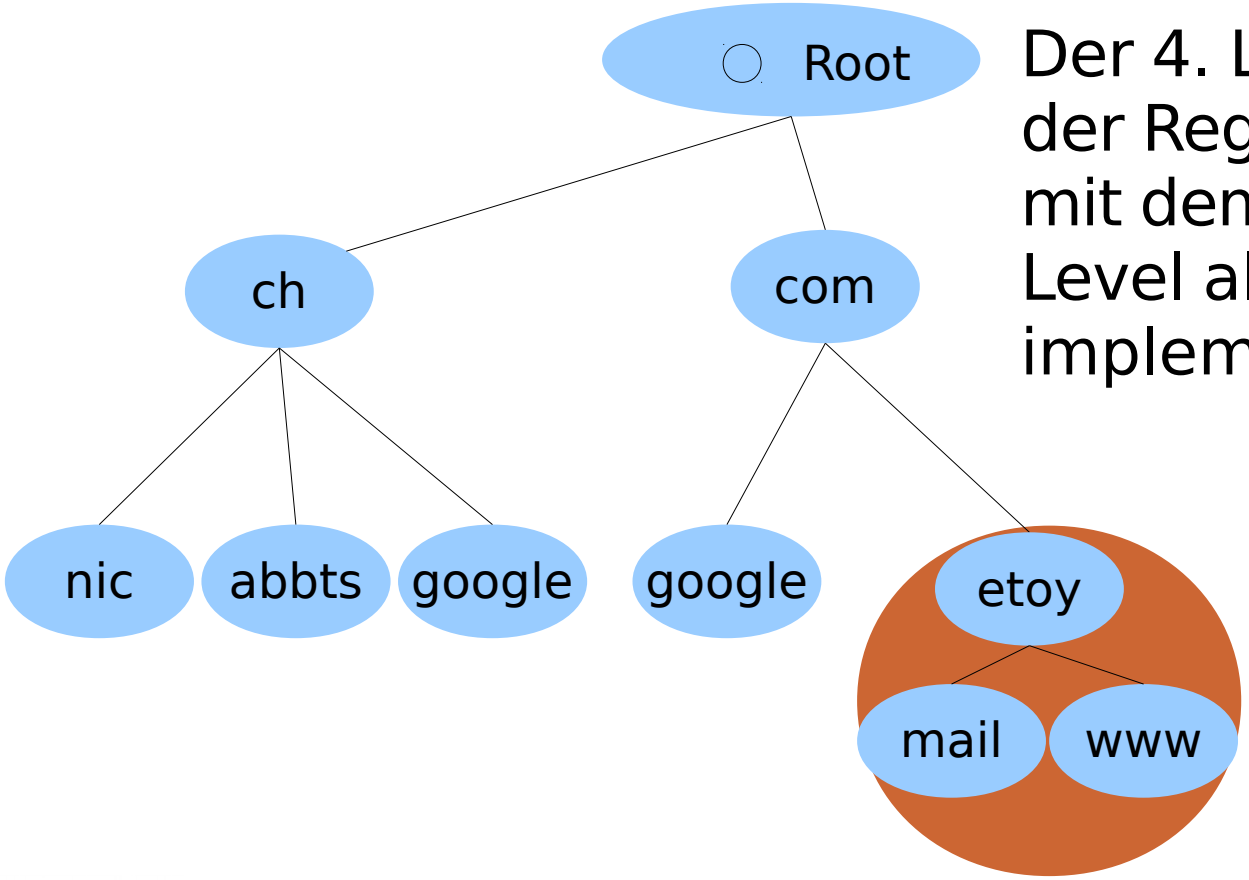
Hierarchische Konzept



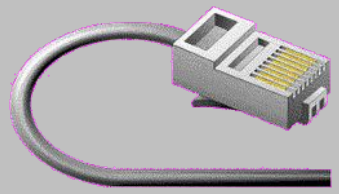


DNS Zonen

DNS Zonen bilden eine administrative Einheit. Für die ersten 3 Level der Domains wird jeweils eine eigene Zone erstellt



Der 4. Level wird in der Regel zusammen mit dem dritten Level als ein Zone implementiert

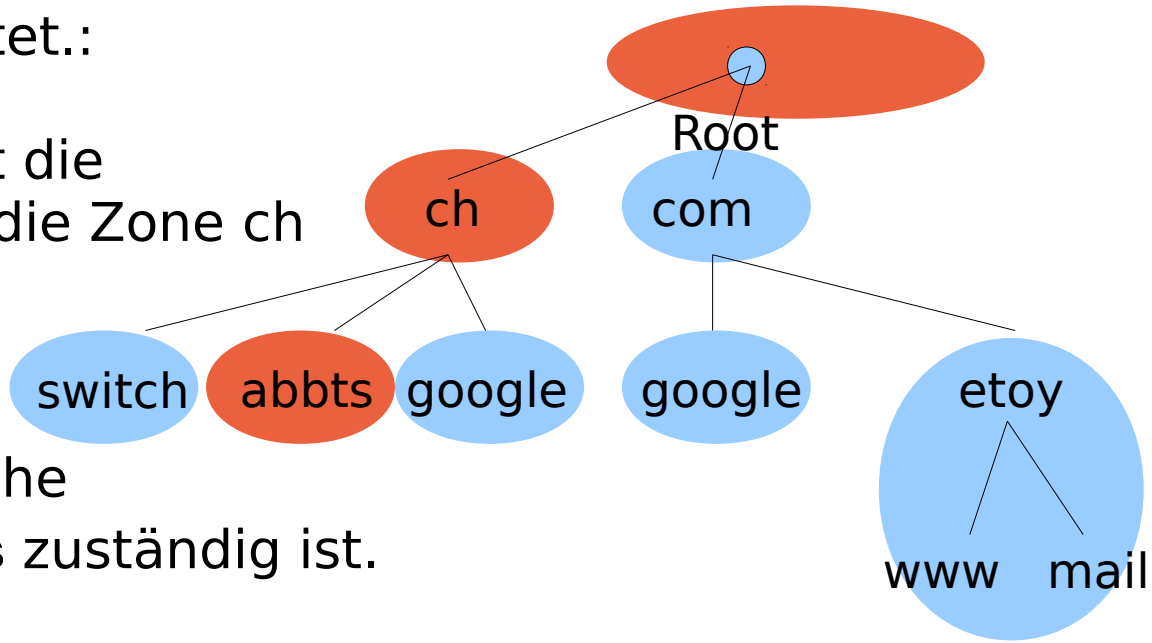


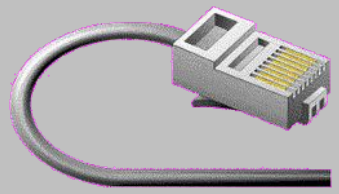
Das Huhn und Ei Problem

Wenn man die IP-Adresse vom `www.abbts.ch` herausfinden will, so muss man zuerst die zuständigen DNS-Server finden. Dies erfolgt indem man von der Wurzel her den Namen abarbeitet.:

Die Root-Zone kennt die DNS-Server, die für die Zone ch zuständig sind.

Die DNS-Server der Zone ch wissen welche DNS-Server für abbts zuständig ist.





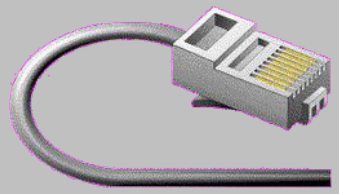
Das Huhn und Ei Problem

Das bedeutet, wenn ein DNS-Server die Root-Server kennt, kann er alle **existierende** Namen auflösen.

Kennt ein DNS-Server die Root_Server nicht, so kann er die meisten Namen nicht auflösen kann.

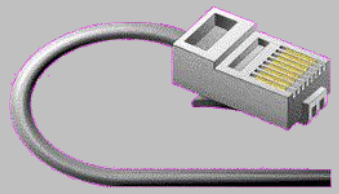
Darum sind bei allen DNS-Server die Root_Server von Hand im sogenannten hint-File konfiguriert.

Der DNS-Server wird die aktuelle Liste der Root-Server nach dem Start bei einem der konfiguriert Root-Server 'abholen'.



Master und Slaves

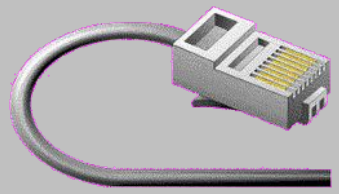
- Da DNS-Server sehr wichtig sind, wird eine Zone in der Regel auf mehreren Servern gehostet.
- Damit der Administrator nicht auf allen Servern die Daten anpassen muss, gibt es die Primary-Name-Server und die Secondary-Name-Server.
- Ein Primary-NameServer besitzen immer die aktuelle Daten der entsprechenden Zone.



Master und Slaves

- Die Secondary-Name-Server überprüfen in regelmässigen Abständen ob aktuellere Daten beim Primary Server bereitstehen und holen sich die Daten gegeben falls und speichern die Daten lokal ab (Pull-Methode).

- Primary-Server können den Secondary-Name-Server eine Meldung senden (Notify), dass neue Daten vorhanden sind, damit diese den Transfer der Daten einleiten (Push-Methode).



Daten im DNS

In den DNS-Zonen können verschiedene Informationen angelegt werden. Die folgenden Typen sind die wichtigsten heute verwendeten Resource Records (RR)

A (IPv4 **A**dresse)

AAAA (IPv6 Adressen (4 mal 32Bit lang, darum AAAA))

PTR (**P**oin**t**e**r**) Werden gebraucht für die Übersetzung von IP-Adressen in FQDN.

MX (**M**ail e**X**change) Gibt an welche Mailserver verwendet werden sollen.

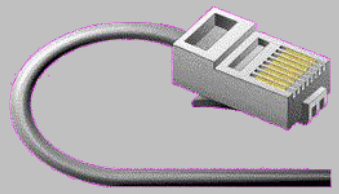
NS (**N**ame **S**erver) Gibt die Nameserver der eigenen Zone an.

SOA (**S**tart **o**f **A**utoritativ) Definiert die Gültigkeitsdaten für eine Zone.

CNAME (**c**anonical **name**) Alias Namen (Vorsicht!)

SRV (**S**e**rv**ice) Wo befindet sich der entsprechende Service.

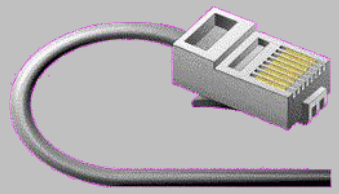
TXT (**T**e**xt**) Freier Text



Daten im DNS

Der MX Record wird verwendet, um die Mailserver einer Domain anzugeben.
Man kann mehrerer Mailserver mit unterschiedlichen Prioritäten konfigurieren. Mails sollen an den Server mit der tiefste Priorität gesendet werden.

```
$host -t mx heuer.org
heuer.org mail is handled by 10 mail.heuer.org.
heuer.org mail is handled by 20 mx2.cyberlink.ch.
```



Reverse Lookups bei IPv4

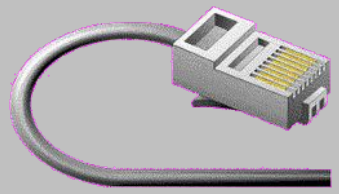
Um von der IP-Adresse auf dem Hostnamen zu kommen ist ein Reverse Lookup notwendig.

Wenn sie beispielsweise wissen wollen, welchem Host die IP 212.55.197.226 zugeordnet ist so passiert folgendes:

Der DNS-Client erstellt eine Anfrage vom Type Pointer für den Host 226.197.55.212.in-addr.arpa.

```
$ host 212.55.197.226
226.197.55.212.in-addr.arpa domain name pointer example.org.
```

Das bedeutet, dass die Reverse Lookups nicht aus dem vorwärts Baum erstellt werden, sondern dass diese separat erstellt werden müssen.

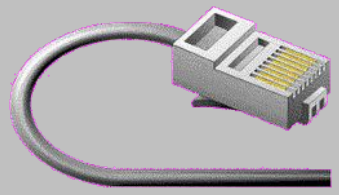


Reverse Lookups bei IPv6

Bei IPv6 erfolgt die reverse DNS-Auflösung analog der Lösung vom IPv4

Anstelle von `in-addr.arpa` wird bei IPv6 die Domain `ip6.arpa` der umgedrehten IP-Adresse angehängt und nach einem PTR (Pointer) gefragt.

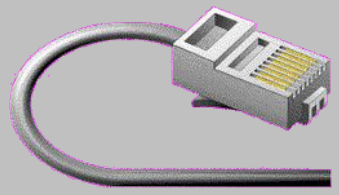
```
$ host ::1
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
0.0.0.0.0.0.ip6.arpa domain name pointer localhost.
```



rekursive und authoritive-only DNS-Server

DNS-Server beantworten die Fragen Aufgrund ihrer lokalen – konfigurierten – Datenbank.
Wenn der DNS-Server nun eine Frage bekommt, die er aufgrund der lokalen Informationen nicht beantworten kann so hat er zwei Möglichkeiten:

- Der DNS-Server kann herausfinden, welcher DNS-Server für die gesuchte Zone zuständig ist und dann die Frage an diesen Server weiterleiten und die Antwort dem Client zurück liefern. (rekursive Server)
  Der DNS-Server wird selber zum Client gegenüber dem zuständigen Server.

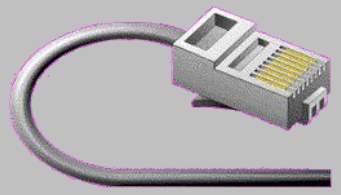


rekursive und authoritive-only DNS-Server (2)

- Der DNS-Server kann dem Client zurückmelden, dass er diese Domain nicht kennt. (authoritive-only)

DNS-Clients von Benutzer-Rechnern und Servern sollten nur rekursive DNS-Server verwenden. d.h. verwenden sie immer nur die DNS-Server, die der Provider angibt - oder sie wissen was sie tun und installieren selber einen rekursiven DNS-Server.

Wenn sie 'fremde' DNS-Server verwenden, kann es sein, dass sich dieser DNS-Server wie ein authoritive-only Server verhält.



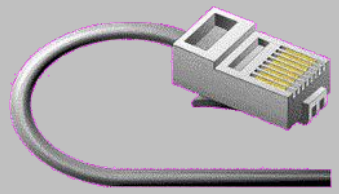
rekursive und authoritive-only DNS-Server (3)

Die rekursiven DNS-Server cachen die Resultate der Abfragen, so dass sie bei einer gleichen Abfrage das Ergebnis direkt liefern können.

Einerseits wird da durch das Ergebnis schneller geliefert und andererseits werden die Root-, TopLevel- und SecondLevel-Server entlastet.

Sowohl positive (Name gefunden) wie auf negative (Name nicht gefunden) Ergebnisse werden im Cache abgelegt.

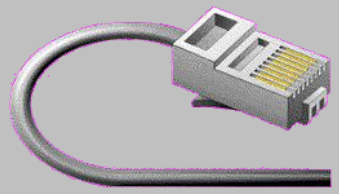
Die DNS-Clients verwenden in der Regel immer zuerst die lokale hosts Datei und danach die DNS-Server um einen Namen aufzulösen.



DNS Clients - Server

DNS-Clients werden in der Regel mit 2 DNS-Server konfiguriert. Oft erfolgt dies automatisch mittels DHCP.

Der 2. DNS-Server wird nur verwendet, wenn der der erste Server ausfällt. Da jedes mal einen Timeout abgewartet werden muss, scheint das Internet "langsam" zu sein.

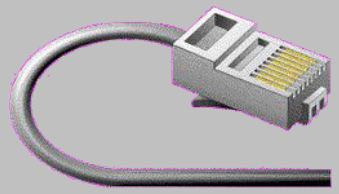


DNS Clients - searchdomain

DNS-Clients können so konfiguriert werden, dass ein relativer Hostname mit verschiedenen Domains versucht wird aufzulösen. (Unixen: `/etc/resolv.conf`, Windows: Netzwerk, TCP/IP Einstellungen)

```
$ host -v monet
Trying "monet.maillink.ch"
Trying "monet.heuer.org"
Trying "monet.cyberlink.ch"
Trying "monet.magnet.ch"
```

Der erste Hostnamen der einen gesuchten Eintrag besitzt wird zurück gegeben.



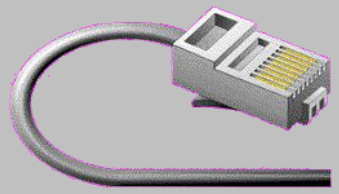
RootServer

Die RootServer sind der Zentrale Punkt, mit denen der ganze DNS-Service steht und fällt.

Weltweit gibt es 13 Root-Server. (Die maximale Anzahl ist aufgrund der minimalem Paket Grösse von 576Byte limitiert. Diese minimale Paket Grösse definiert diemaximale DNS Paketgrösse).

Um trotzdem mehr Root Servers zu verwenden sind einige Server in AnyCast Netzen platziert.

AnyCast-Server sind Server die an verschiedenen Orten im Netz angeschlossen sind. Diese Server besitzen all dieselben IP-Adressen (!).

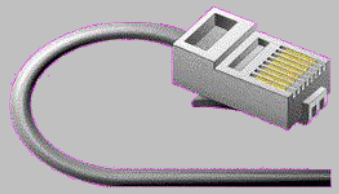


RootServer

Kennen die Root Server eine Top-Level Domain nicht, so können keine Domain-Namen aus dieser Top-Level Domain aufgelöst werden. Der Betreiber der Root Zone hat extrem grossen Einfluss aufs Internet.

Neben den IANA-Root Servern gibt es weitere Root-Server Netzwerke. Viele der alternativen RootServer verwenden zusätzliche Top-Level Domains - was zu Problemen führt wenn eine dieser zusätzlichen Domain offiziell etabliert wird.

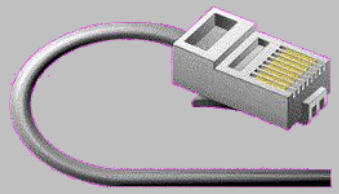
~~ORSN, European Open Root Server Network, betreibt alternative RootServer, die zu 100% kompatibel zu den RootServern von IANA sind.~~



Betreiben eines DNS-Servers

Wenn sie einen eigenen authoritiven DNS-Server betrieben möchten, so müssen sie einige Randbedingungen beachten:

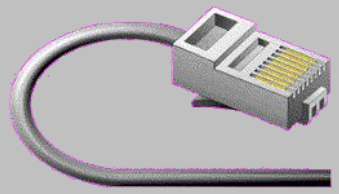
- Die Server müssen rund um die Uhr erreichbar sein.
- Die Server sollten geographisch an unterschiedlichen Orten stehen. Das bedingt, dass es mehrere Server sind, und dass die Server über verschiedene Leitungen am Netz angeschlossen sind.
- Die Server sollten in unterschiedlichen CIDR-Blöcken liegen, wenn möglich sollen die Server über unterschiedliche ISPs angeschlossen sein.



Betreiben eines DNS-Servers (2)

Eine Möglichkeit um diese Anforderungen zu "umgehen" ist der Einsatz eines versteckten Primary-Servers.
In der Zone werden die öffentlichen Server des Providers konfiguriert. Diese Server sind jedoch alles Secondary NameServer, die die Informationen vom ihrem nicht öffentlichen Primary Server abholen.
Dadurch behalten sie die Kontrolle über ihre Daten in den Zonen, müssen aber nicht die DNS-Server selber verwalten.
Verschiedene Anbieter bieten auch global verteilte DNS-Server an.

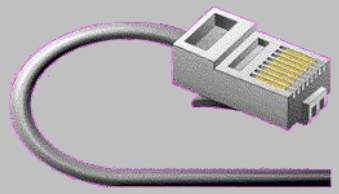
Denken sie immer daran, ohne DNS-Server ist ihre Domain wertlos, denn niemand wird sie finden!



URLs

DNS Software:

Bind	http://www.isc.org/index.pl?/sw/bind/
Powerdns	http://www.powerdns.com/
TinyDns	http://tinydns.org/
RootServer:	http://www.root-servers.org/
RootServer der RIPE:	http://k.root-servers.org/
~~ORSN Network:~~	~~http://european.ch.orsn.net/index.php~~ Archiv: http://orsn.maillink.ch/index.php
Tools:	http://dnstools.com/
IANA	http://www.iana.org/
IANA-Domains	http://www.iana.org/domain-names.htm



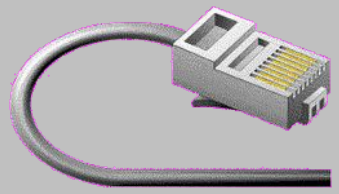
Test Domains

Die Domain-Namen

- **`example.com`**
- **`example.net`** und
- **`example.org`**

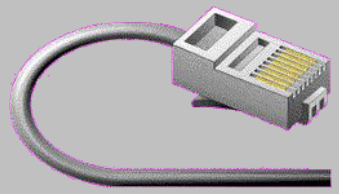
sind gemäss RFC 2606 reserviert.

Sie können diese Domain-Namem für lokale Experimente oder für Dokumentationen verwenden.



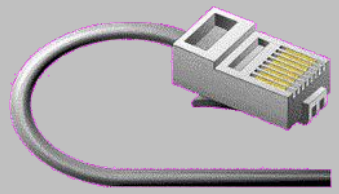
Fragen?

DNS servers must feel sad,
nobody calls them by their name.

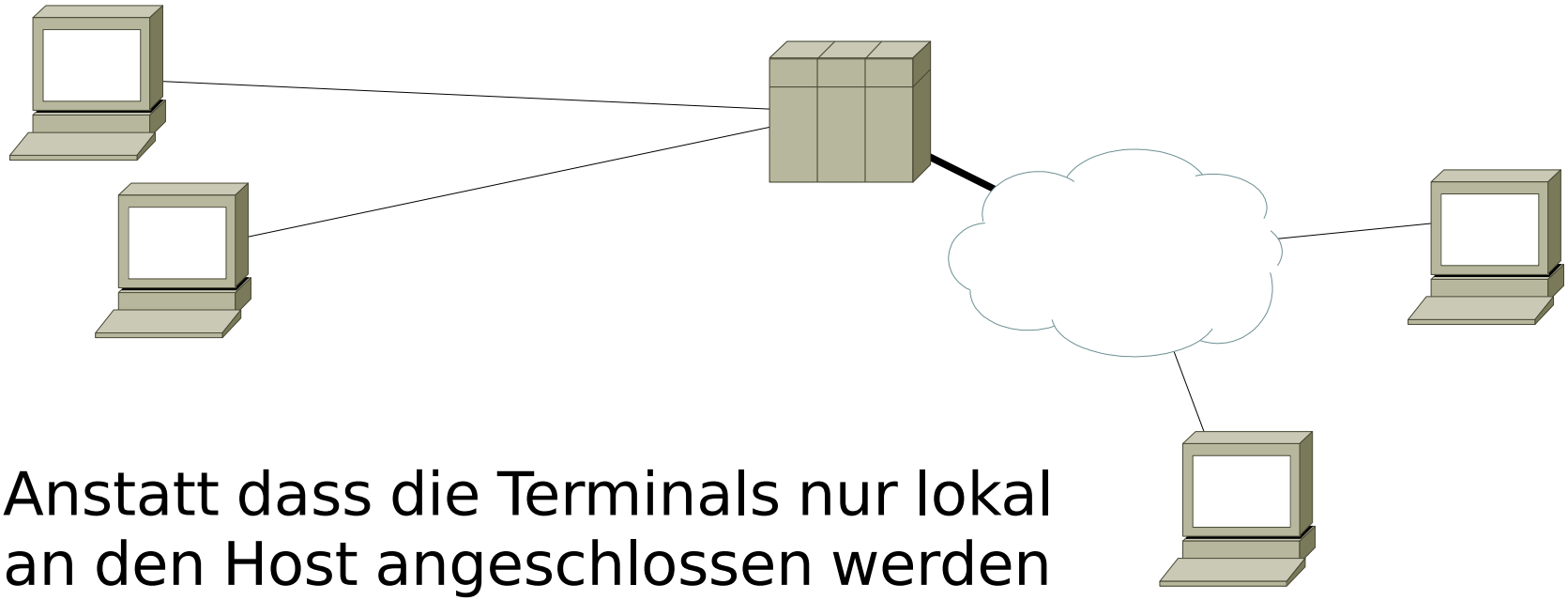


Anwendungen

- Remote Sessions: Telnet / ssh / X11 / RDP
- Mail: SMTP, POP3, IMAP4, (X.400)
- Zeit: NTP
- Überwachung / Konfiguration: SNMP

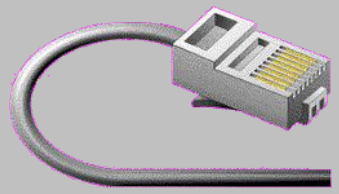


Remote Session



Anstatt dass die Terminals nur lokal an den Host angeschlossen werden können die Terminal von überall her auf den Host zugreifen.

\+ Mehr Anschlüsse sind möglich
\- Sicherheit?



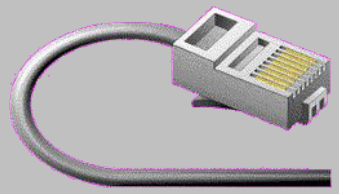
Remote Session

Der Zugriff erfolgt via

- telnet (RFC 854, RFC1572), TCP Port 23
- rlogin (RFC 1282, remote login) TCP Port 513
- rsh (remote shell) TCP Port 514
- ssh (RFC4251), TCP Port 22

bei graphischen Oberflächen:

- X11-Clients können den X-Server direkt via TCP (Port 6000 ...) ansprechen, oft wird der Datenstrom durch einen ssh-Tunnel weitergeleitet
- Windows: verwendet Remote Desktop Protokoll (RDP oder Citrix)



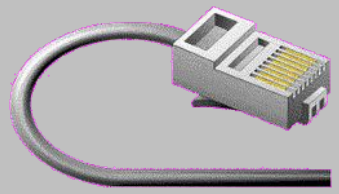
Remote Session telnet

Telnet

Mit **`telnet`** <RemoteHost> <Port> kann die Verbindung zum RemoteHost aufgebaut werden.

Netzwerkgeräte verwendeten häufig Telnet, damit diese administriert werden können.

Telnet überträgt die Ein- und Ausgaben unverschlüsselt übers Netz. Jeder der weiss, wie man Traffic aufzeichnen kann, kann so die Passwörter leicht herausfinden.



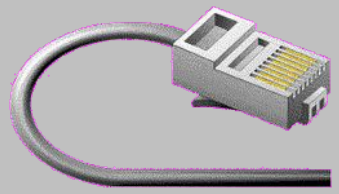
Remote Session telnet

Mit **`telnet`** können grundsätzlich alle TCP-Verbindungen getestet werden. Telnet akzeptiert nach dem Hostnamen/IP-Adresse eine Port Angabe.

```
heuer@guybrush:~$ telnet ::1 22
Trying ::1...
Connected to ::1.
Escape character is '^]'.
SSH-2.0-OpenSSH_5.9p1 Debian-4

Protocol mismatch.
```

Sobald die Angabe "Connected to" erscheint ist der Threeway Handshake fertig und man ist sicher, dass auf den angegeben Port beim Zielhost eine Daemon, Service installiert ist.

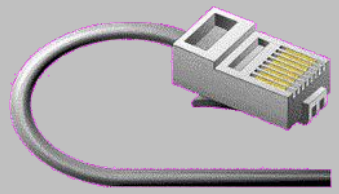


Remote Session telnet

Telnet

`telnet` handelt beim Verbindungsaufbau verschiedene Parameter aus:

```
▷ Frame 2292: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)
▷ Ethernet II, Src: AsustekC_9d:b9:9a (e0:cb:4e:9d:b9:9a), Dst: Cisco_5d:a0:33 (00:1b:0c:5d:a0:33)
▷ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 3
▷ Internet Protocol Version 4, Src: 212.55.196.74 (212.55.196.74), Dst: 195.226.23.14 (195.226.23.
▷ Transmission Control Protocol, Src Port: 39893 (39893), Dst Port: telnet (23), Seq: 1, Ack: 1, Le
▽ Telnet
    Command: Do Suppress Go Ahead
    Command: Will Terminal Type
    Command: Will Negotiate About Window Size
    Command: Will Terminal Speed
    Command: Will Remote Flow Control
    Command: Will Linemode
    Command: Will New Environment Option
    Command: Do Status
    Command: Will X Display Location
```



Remote Session

SSL Verbindungen können mit openssl getestet werden:

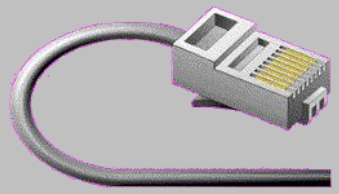
```
openssl s_client -crlf -connect <host:port> [-starttls protpcol]
```

Beispielsweise kann mit

```
openssl s_client -crlf -connect mail.maillink.ch:110 -starttls pop3
```

verschlüsselt auf eine POP3 Mailbox zugegriffen werden.
Die übertragene Daten mittels Wireshark aufgezeichnet sehen wie folgt aus:

```
+OK Dovecot ready.
STLS
+OK Begin TLS negotiation now.
....;...7..Oo3#O.G..S..B.....+....|......".....0.,.(.
$...".!.....k.j.9.8.....2...*.&.......=.5.................…
```

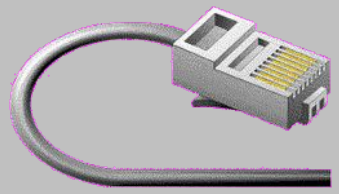


Remote Session ssh

ssh verschlüsselt den Traffic, so dass der grosse Nachteil von telnet behoben ist.

Zusätzlich kann ssh andere Ports innerhalb des verschlüsselten Tunnels transportieren. Damit können andere Protokolle - die selber keine Verschlüsselung kennen - verschlüsselt übers Internet transportiert werden.

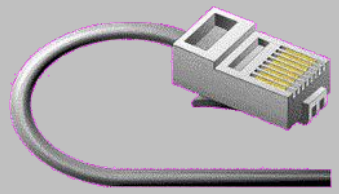
Anleitungen wie man mit ssh Port forwarding einrichtet können im Internet gefunden werden.



Remote Session

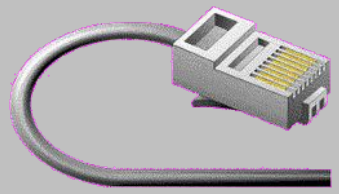
Mit der Verbreitung von graphischen Oberflächen sind neue Remote Session Protokolle entwickelt worden:

X11 kann schon seit jeher über TCP/IP transportiert werden. Dass die Verbindung nicht verschlüsselt ist kann durch ssh und X-Session forwarding verhindert werden.



Remote Session

Microsoft hat für Windows Remote Desktop Protocol entwickelt.
Andere Ansätze sind auch mittels VNC, TeamViewer, Turbomeeting, ...
Mit der Desktop Virtualisierung (DVI) sind diese Protokolle noch mehr optimiert und teileweise in Hardware implementiert worden, beispielsweise PCoIP

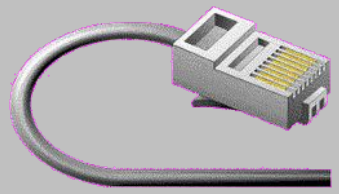


Fragen ?

SSH 1.33 and/or 1.5 protocol jokes are useless.

SSH 1.33 and/or 1.5 protocol jokes are useless.

http://hardware.localhost.nl/

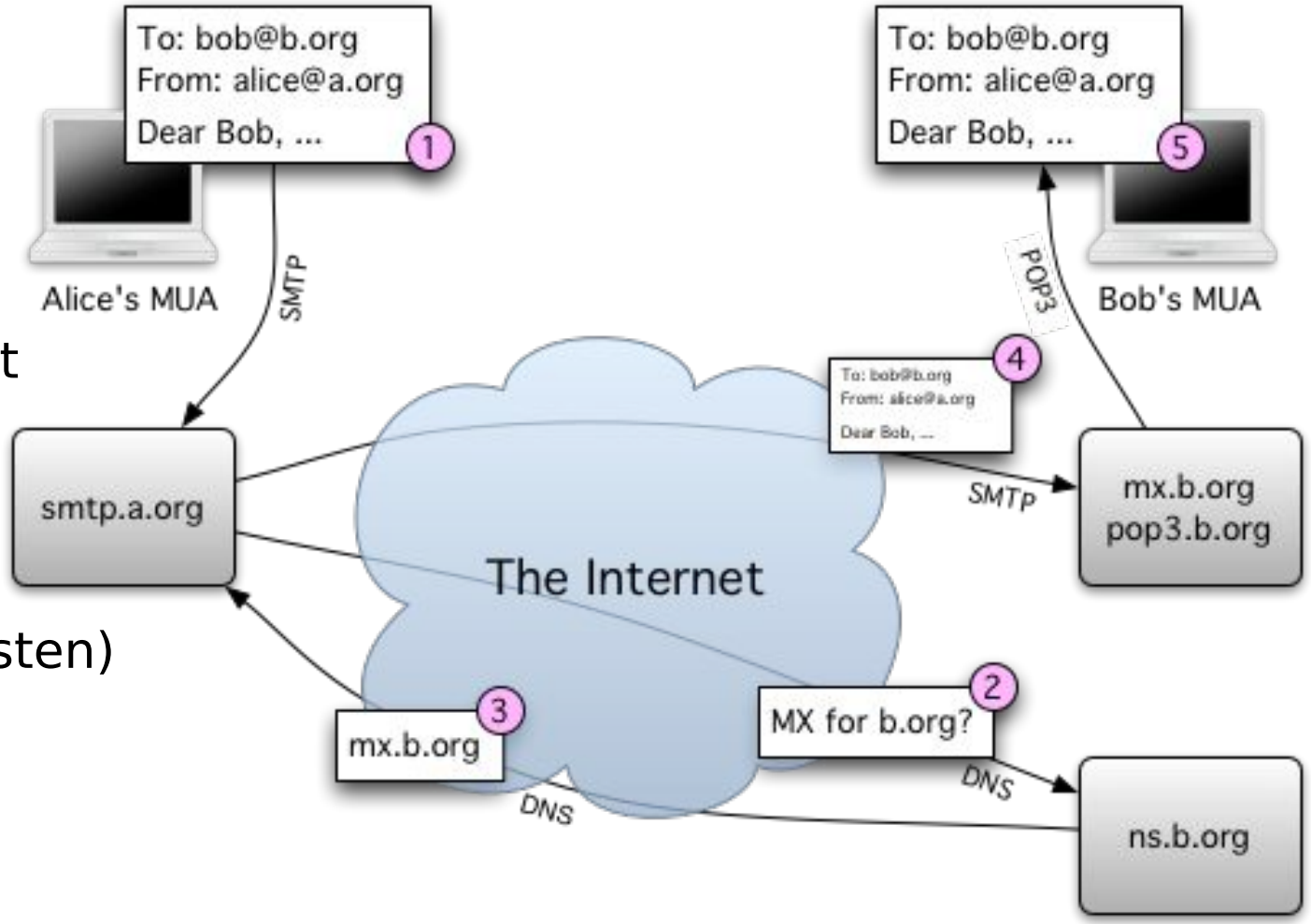


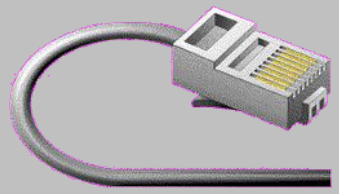
Mail Dienste

MUA:
Mail User Agent
(Mailprogramm)

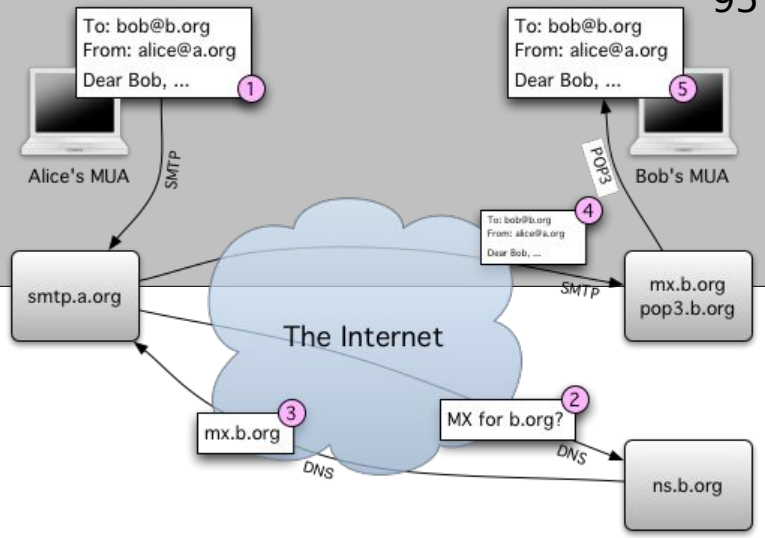
MTA:
Mail Transfer Agent
(Mailserver)

MS:
Mail Storage
(Mailbox / Briefkasten)





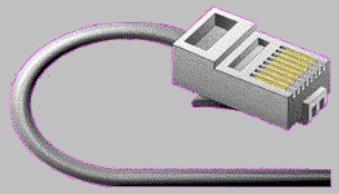
Mail Dienste



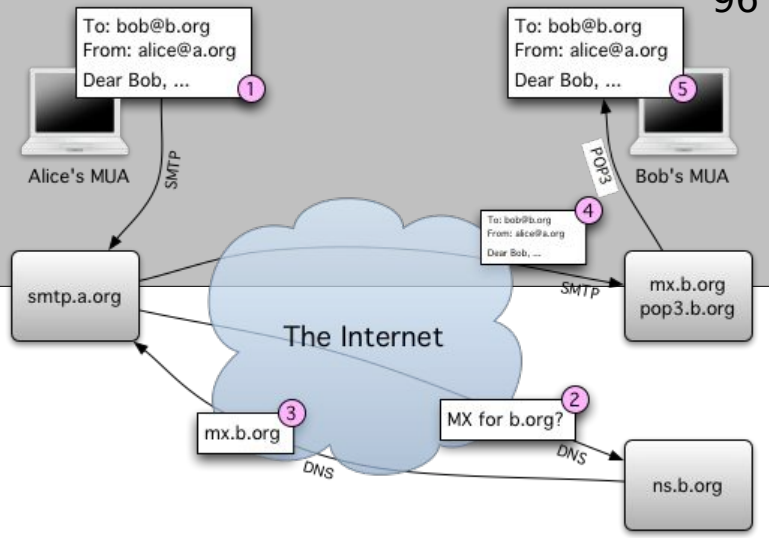
1) User sendet die Mail an den eigenen MTA
(MUA -> MTA, Protokoll: SMTP)

2) der MTA schaut im DNS nach dem MX-Eintrag für die Zieldomain (b.org) (DNS)

3) Aus der Antwort von 2) weiss der MTA welcher andere MTA zuständig ist.

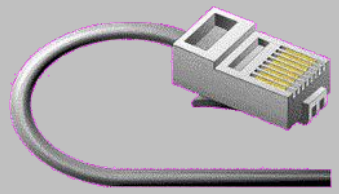


Mail Dienste



4) Der MTA kontaktiert den MTA und überliefert ihm die Mail. Der andere MTA nimmt die Mail entgegen und speichert diese im MailStorage vom User bob@b.org
(MTA -> MTA, Protokoll SMTP)

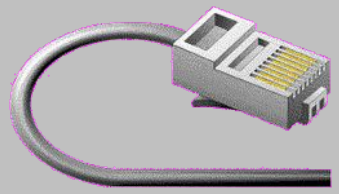
5) Der User bob prüft ob neue Mails in seiner Mailbox (Mailstorage) sind und ruft diese ab.
(MUA -> MS, Protokoll POP3, IMAP4)



Mail Dienste

Beispiel einer SMTP-Session zwischen einem MUA und einem MTA:

SMTP< 220 largo.maillink.ch ESMTP
ESMTP> **EHLO guybrush.maillink.ch**
ESMTP< 250-largo.maillink.ch
ESMTP< 250-STARTTLS
ESMTP< 250-PIPELINING
ESMTP< 250-8BITMIME
ESMTP< 250 AUTH LOGIN PLAIN
ESMTP> **MAIL FROM:<ueli@heuer.org>**
SMTP< 250 ok
SMTP> **RCPT TO:<ueli@heuer.org>**
SMTP< 250 ok
SMTP> **DATA**
SMTP< 354 go ahead
... hier wird die Mail transferiert
SMTP> **.** (EOM)
SMTP< 250 ok 1172662923 qp 8920

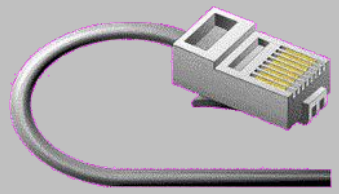


POP3 Session

Beispiel einer POP-Session: (Post Office Protocoll)

RFC 1939

heuer@flunder:~$ **telnet mail.heuer.org 110**
Trying 2001:8a8:30:10::2...
Connected to largo.maillink.ch.
Escape character is '^]'.
+OK <10796.1172664285@pop.maillink.ch>
user ueli@heuer.org
+OK
pass <password>
+OK
list
+OK
1 4017
2 9059
3 2826
.
retr 1
+OK
Received: (qmail 13954 invoked by uid 600); 30 Nov 2006 23:25:09 -0000
....
.
quit
+OK

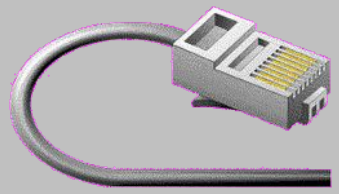


IMAP Session

Beispiel einer IMAP-Session: (Internet Message Access Protocol)

RFC 3501

```
heuer@flunder:~$ telnet mail.heuer.org 143
Trying 2001:8a8:30:10::2...
Connected to largo.maillink.ch.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE ....
0 LOGIN ueli@heuer.org <PASSWORD>
0 OK LOGIN Ok.
1 STATUS "INBOX" (MESSAGES)
* STATUS "INBOX" (MESSAGES 377)
1 OK STATUS Completed.
2 STATUS "INBOX" (MESSAGES)
* STATUS "INBOX" (MESSAGES 378)
2 OK STATUS Completed.
3 LOGOUT
* BYE Courier-IMAP server shutting down
3 OK LOGOUT completed
```



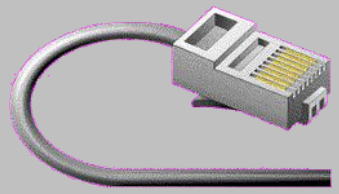
MAIL Probleme

Mail hat heute verschiedene Probleme:

MailClients versenden Mails wie MailServer an den konfigurierten Mailserver. Die Mailclients verwenden dazu das SMTP. Würden alle MailClients die Mails an einem anderen Port abliefern (submission RFC2476), könnte man jeden User zwingen sich per SMTP-AUTH mit Usernamen und Passwort zu authentisieren und zu autorisieren. Die Folge wäre sehr viel weniger 'zombie'-Mailer und damit weniger Viren- und Spam-Mails.

RBLs und Gray-listing können helfen diese abzuweh-ren

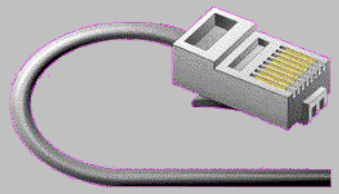
RBL: Realtime Black Listen



MAIL Probleme

Mails sind offen. Die Mails liegen unverschlüsselt auf den MailStorage und jeder der Administrator der zugriff auf den Rechner hat, kann diese lesen. → Mails mit PGP oder S-MIME verschlüsselt versenden.

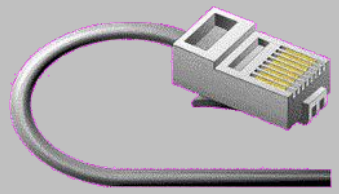
Beim Transfer von MTA zu MTA können – müssen aber nicht - die Kanäle verschlüsselt sein. Auch gibt es Möglichkeiten die IMAP und POP Dienste entsprechend zu verschlüsseln.



MAIL Probleme

Sie sind nicht sicher, ob die Mail auch wirklich von der Person stammen die im Mail angegeben sind? → Da jeder Mails mit beliebigen Absendern versenden kann, ist nicht sicher, ob die Mail wirklich von dieser Person stammt!
Nur Mails signieren und den entsprechenden Schlüssel sicher austauschen hilft hier weiter.

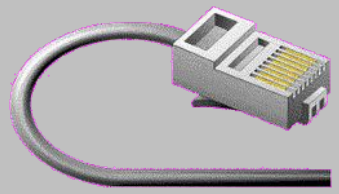
Mails von bestimmten Mail-Adressen können von der Strafverfolgungs-Behörde auf Anfrage in Echtzeit überwacht werden. Der Inhaber der Mail-Adresse wird dabei logischerweise nicht informiert!



MAIL Server selber aufsetzen

Einen eigenen Mail-Server selber aufzusetzen ist nur dann zu empfehlen, wenn sie genau wissen was sie tun. Wenn die Installation nicht 100% abgesichert ist, wird es keine lange Zeitspanne dauern, bis ihr Mailserver missbraucht wird.

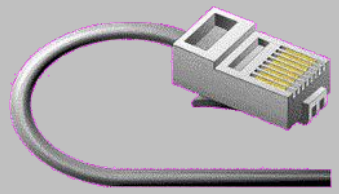
Die Folge ist, das sie mit ihre IP-Adresse umgehend auf verschiedenen Blacklisten landen - und sie dann ihre eigenen Mails nicht mehr versenden können. Gleichzeitig wird ihr guter Ruf kein guter Ruf mehr sein wird.



Fragen ?

You need to tell a authorized POP3 joke before you can have a SMTP laugh.

http://hardware.localhost.nl/

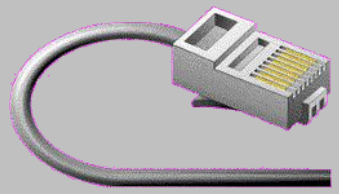


NTP

Jeder Rechner hat eine eigene Uhr, die mal gestellt wurde und mehr oder weniger genau läuft.

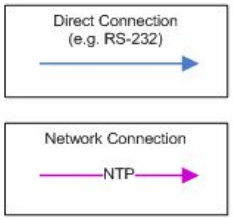
Das ist ein Problem, wenn über mehrere Rechner kommuniziert wird. Stimmen die Uhren der Rechner nicht genau überein, erschwert das den Vergleich von Logfiles, ...

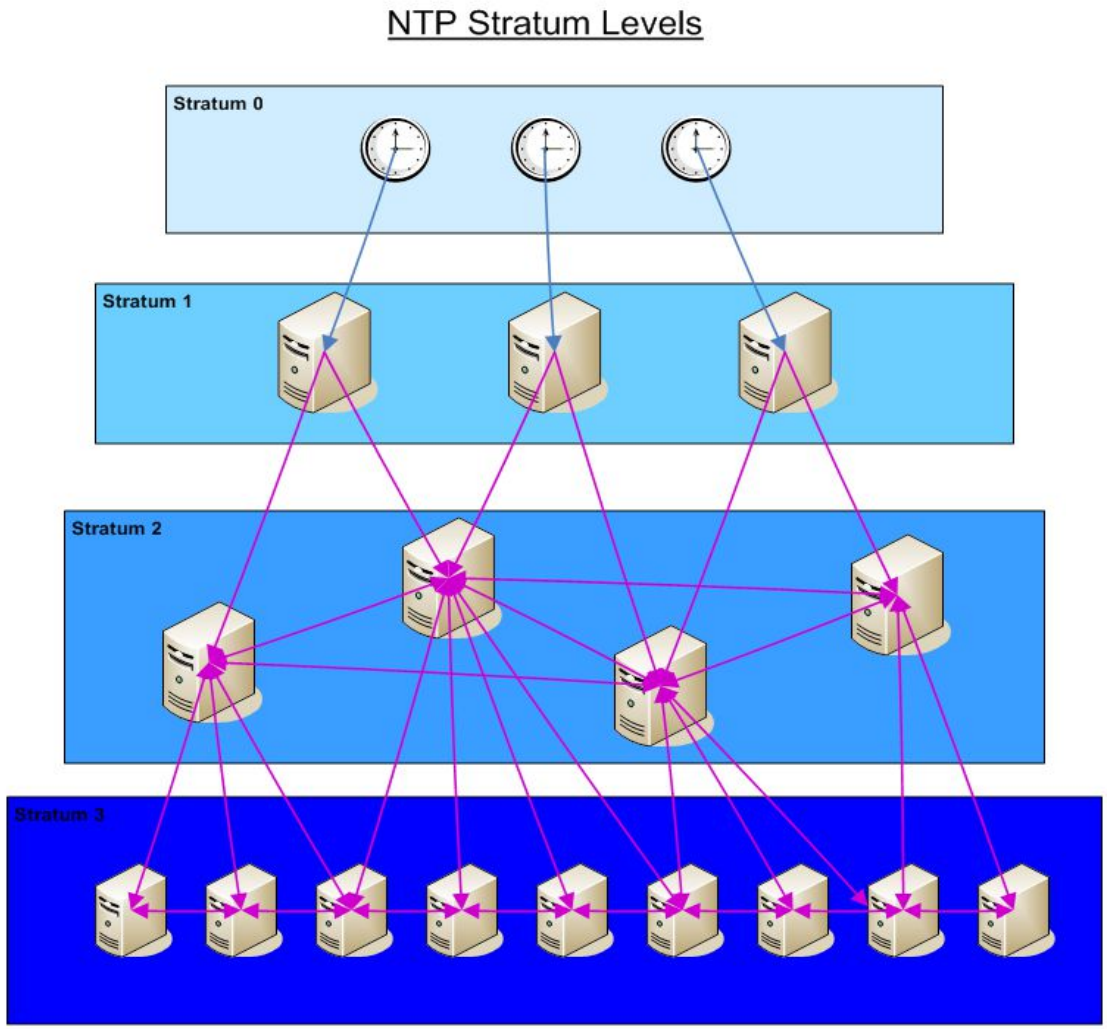
Um die Zeit zu synchronisieren gibt es das Network Time Protokoll (NTP) RFC 1305

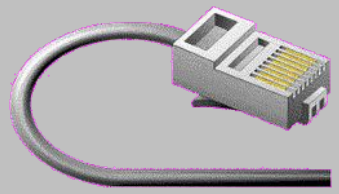


NTP

NTP-Server können die Zeit von anderen NTP-Servern oder von Funk-, GPS-, Atom-, ...-Uhren bekommen.



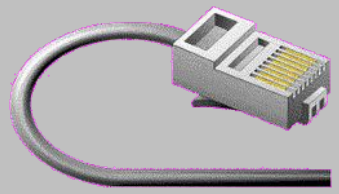




NTP

Um den eigenen Rechner zu synchronisieren brauchen sie keine eigene Atom-Uhr, es reicht wenn sie ihren NTP-Client auf den Pool **<TDL-CODE>.pool.ntp.org** (ch.pool.ntp.org) synchronisieren.

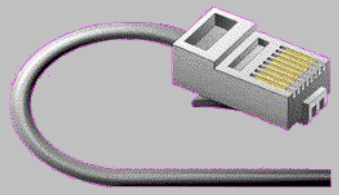
Passende Clients sind eignetlich im Betriebssystem integriert. Clients sind auf der NTP-Homepage http://www.ntp.org/ aufgelistet.



Fragen ?

I made an NTP joke once. The timing was perfect.

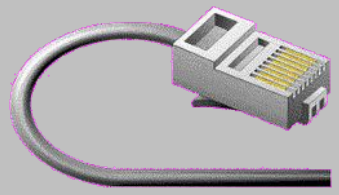
http://hardware.localhost.nl/



Simple Network Management Protocol (SNMP)

Simple Network Management Protocol SNMP

- Lesen und schreiben von Parametern von netz-werkfähigen Geräten



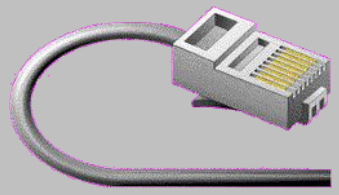
Simple Network Management Protocol (SNMP)

Mittels SNMP können die verschiedensten Werte eines Gerätes abgefragt werden.

Praktisch jedes Gerät, das sich remote managen lässt, hat einen SNMP-Agent integriert.

Bei entsprechender Konfiguration kann das Gerät auch via SNMP konfiguriert werden.

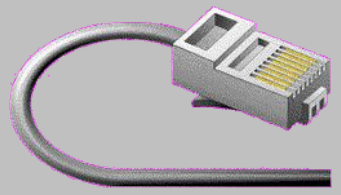
Die Geräte werden mittels GET-, GET-NEXT-Requests, GET-Response und SET Request abgefragt.



Simple Network Management Protocol (SNMP)

Die Geräte können TRAPs versenden, um auf spezielle Bedingungen aufmerksam zu machen. (beispielsweise Temperatur zu hoch, Interface X ist ausgeschaltet worden, ...)

INFORM sind Meldungen, die wie TRAPs vom Gerät versendet wird, INFORM-Meldungen müssen – im Gegensatz zu den TRAPs – vom Empfänger bestätigt werden.

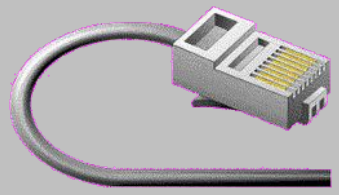


Simple Network Management Protocol (SNMP)

Welche Parameter wie abgefragt bzw gesetzt werden können ist in den Management Information Base (MIB) beschrieben.

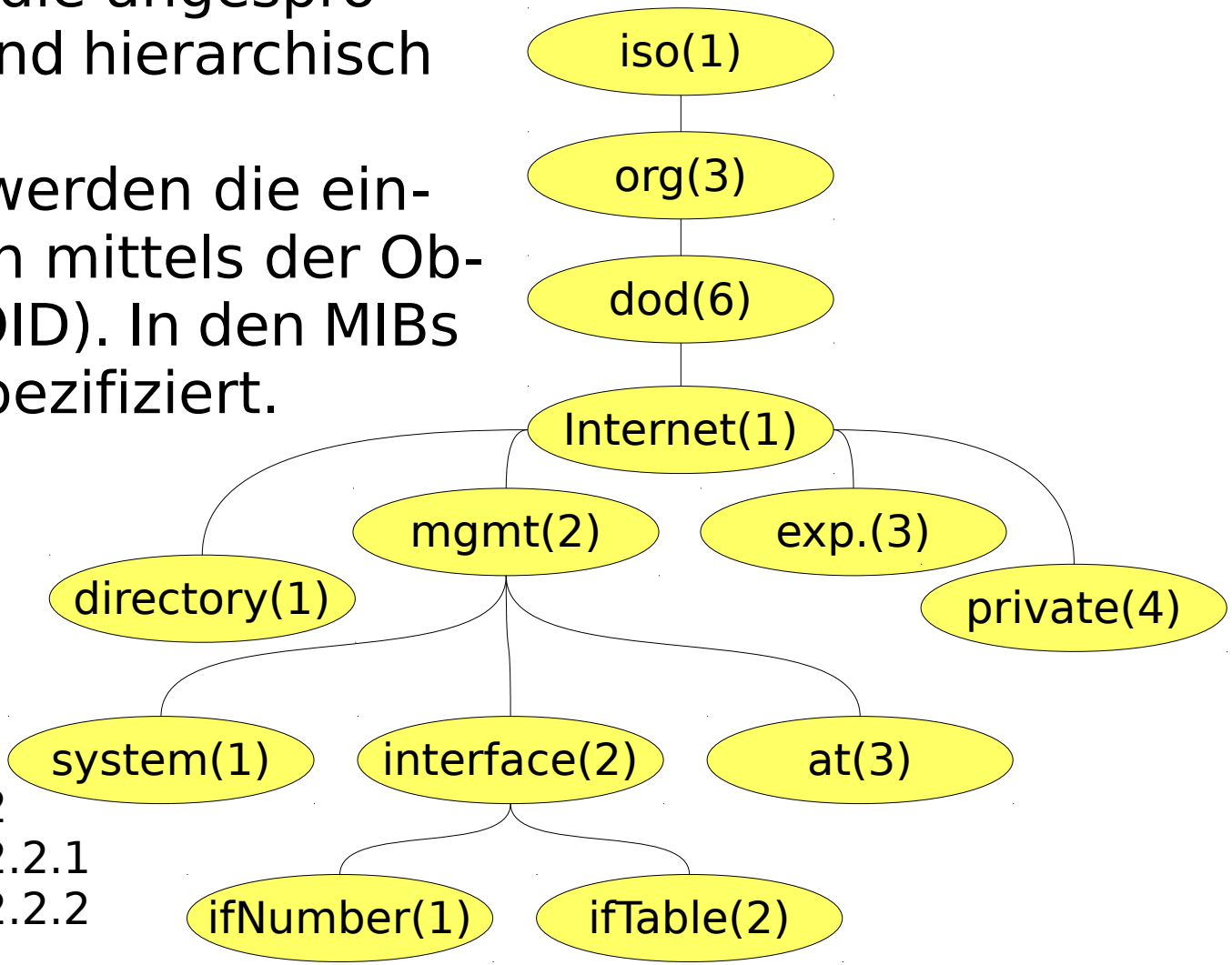
In den MIB-Files wird jede Variable genau beschrieben:

```
sysUpTime OBJECT-TYPE
        SYNTAX       TimeTicks
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
          "The time (in hundredths of a second) since the
           network management portion of the system was
           last reinitialized."
        ::= { system 3 }
```

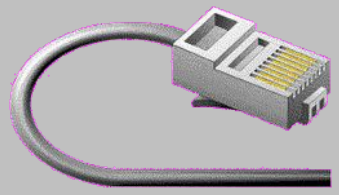


Simple Network Management Protocol (SNMP)

Die Parameter, die angesprochen werden, sind hierarchisch organisiert.
Angesprochen werden die einzelnen Variablen mittels der Object Identifier (OID). In den MIBs sind die OIDs spezifiziert.

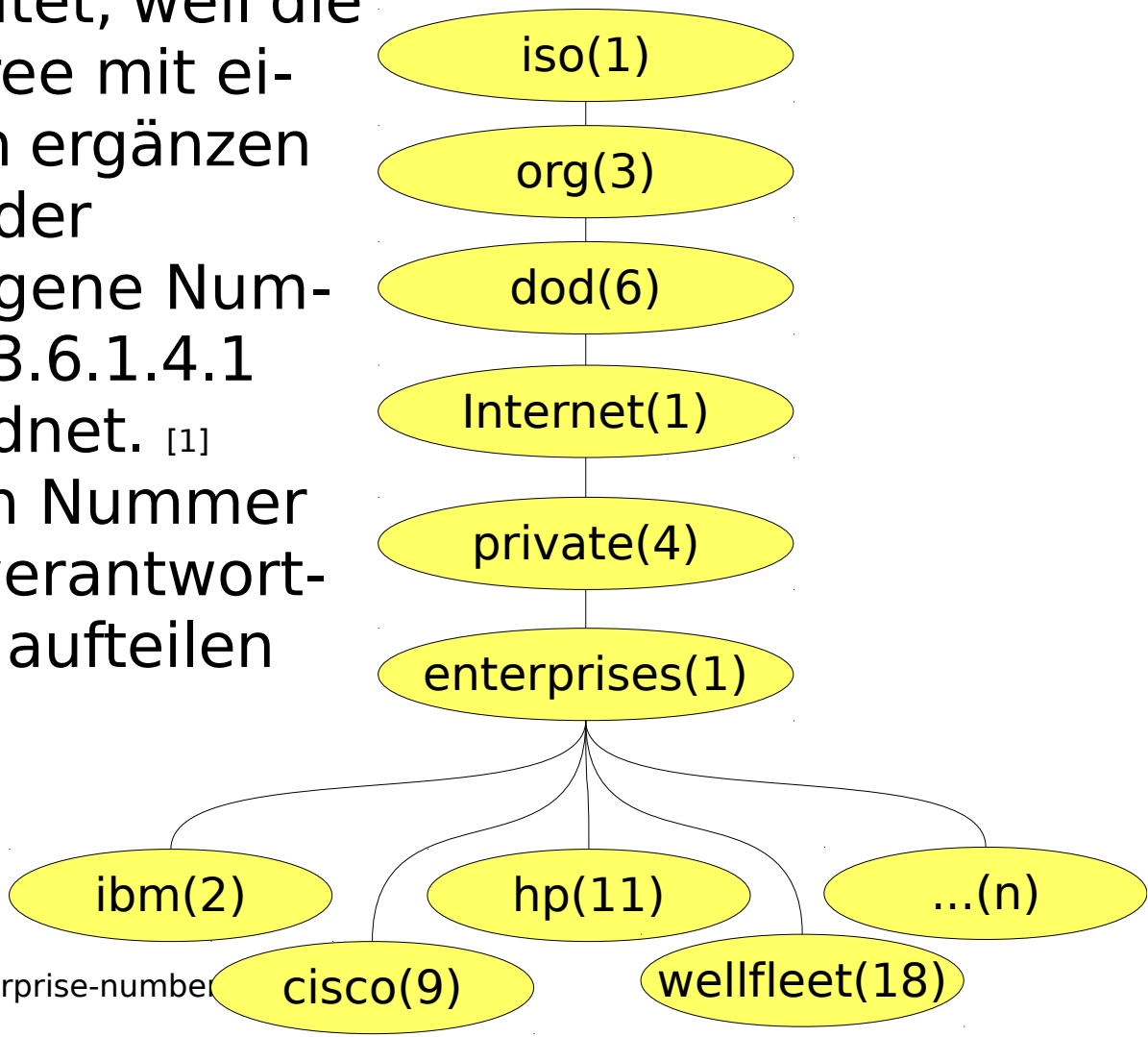


Internet = 1.3.6
Interface = 1.3.6.2.2
IfNumber = 1.3.6.1.2.2.1
ifTable = 1.3.6.1.2.2.2



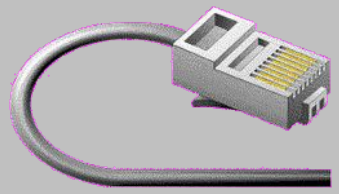
Simple Network Management Protocol (SNMP)

SNMP ist sehr verbreitet, weil die Hersteller den MIB-Tree mit eigenen Erweiterungen ergänzen können. Dazu wird jeder Firma/Projekt eine eigene Nummer unter der OID 1.3.6.1.4.1 (enterprises) zugeordnet. [1]
Unter der zugeteilten Nummer ist die Firma selber verantwortlich wie sie ihre OIDs aufteilen wollen.



Zuständig für die Vergabe ist IANA

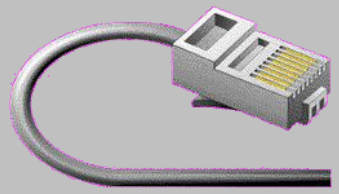
[1] http://www.iana.org/assignments/enterprise-numbe



SNMP Authentication

SNMP V1 und V2c verwenden eine Community (eine Art Passwort) um sich aus zuweisen

SNMPV3 hat eine starke Authentisierung eingebaut. Die Pakete werden bei SNMPv3 mit einem Hash-Wert gesichert.



Abfragen

Mit den Programmen snmpwalk / snmpget können ganze Bäume bzw. einzelne Werte abgefragt werden:

Unter UNIX [1]:

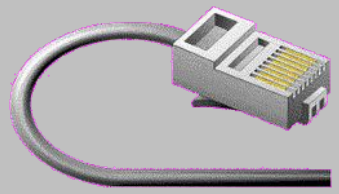
snmpwalk -c <communtiy> -v <version> <host> [OID]

snmpget -c <communtiy> -v <version> <host> [OID]

Mit dem Programm snmpset könnten Werte gesetzt werden. Das geht nur, wenn die OID-Variable schreibbar und die Write-Community eingerichtet ist.

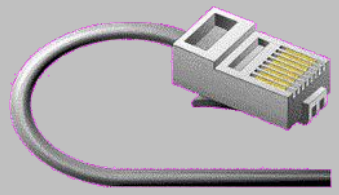
snmpset -c <communtiy> -v <version> <host> OID Type Value

[1] Auf der Net-SNMP Hompage (http://net-snmp.sourceforge.net/) kann unter 'Download' auch ein Windows CLI Version heruntergeladen werden.



Abfragen (Beispiel)

```
$ snmpwalk -cpublic -v2c 212.55.196.65 ifDescr
IF-MIB::ifDescr.1 = STRING: FastEthernet0/0
IF-MIB::ifDescr.2 = STRING: FastEthernet0/1
IF-MIB::ifDescr.3 = STRING: Serial0/0/0
IF-MIB::ifDescr.4 = STRING: Serial0/0/1
IF-MIB::ifDescr.5 = STRING: Null0
IF-MIB::ifDescr.6 = STRING: Loopback0
IF-MIB::ifDescr.7 = STRING: FastEthernet0/0.1
IF-MIB::ifDescr.8 = STRING: FastEthernet0/0.3
IF-MIB::ifDescr.9 = STRING: FastEthernet0/0.100
IF-MIB::ifDescr.10 = STRING: FastEthernet0/0.500
IF-MIB::ifDescr.11 = STRING: Multilink1
IF-MIB::ifDescr.12 = STRING: Multilink1-mpls layer
$
```



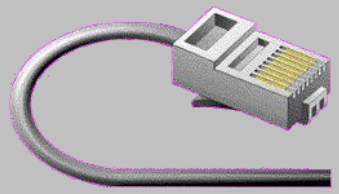
Simple Network Management Protocol (SNMP)

SNMP Anwendungen:

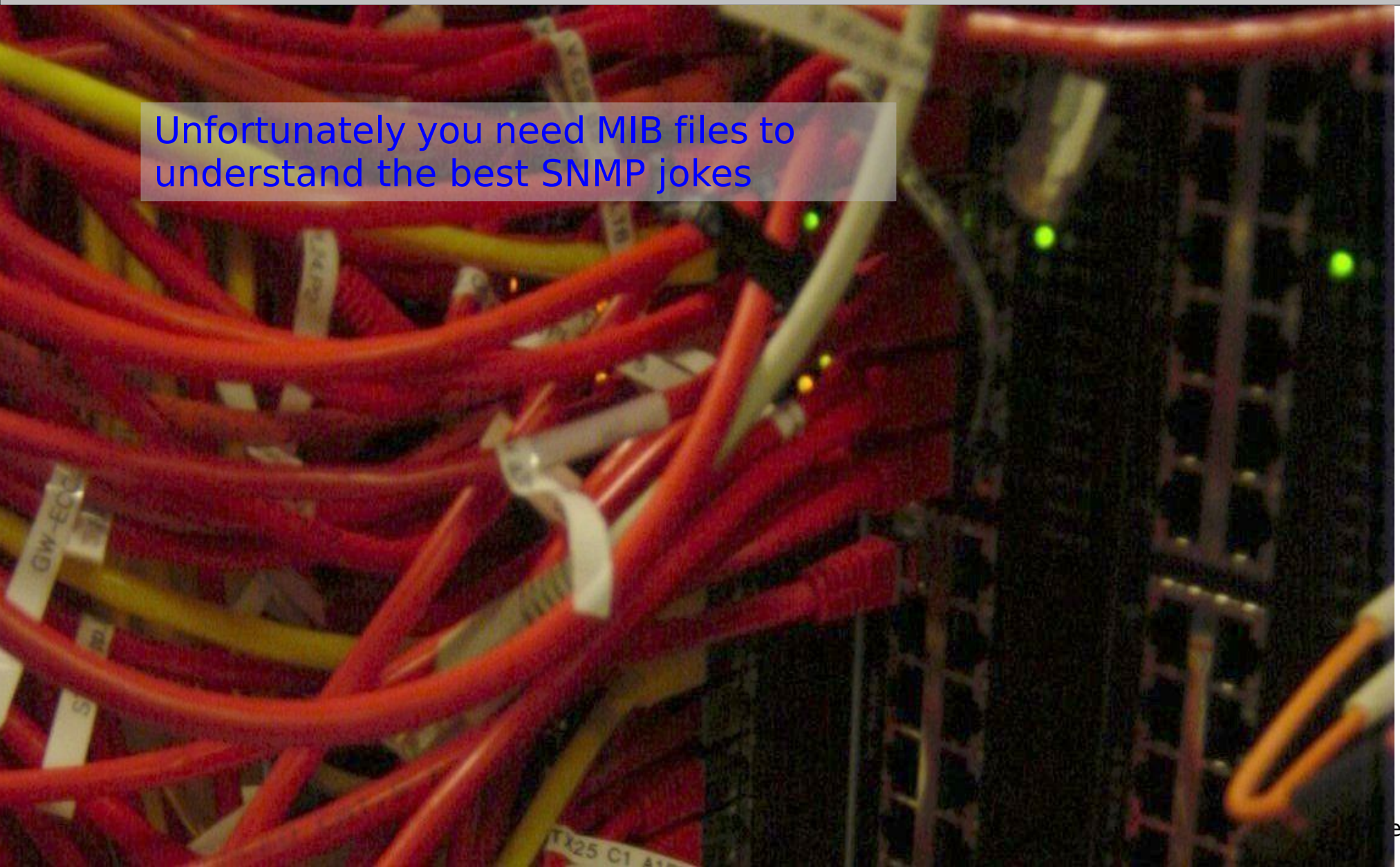
net-snmp: snmpget, snmpwalk
MRTG, (RRD)

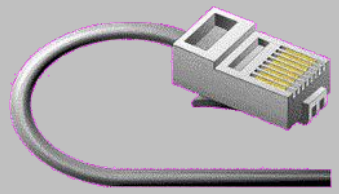
Network Management Systeme

HP Openview, OpenNMS, Netsight, Cacti, Nagios, DVG, ...



Fragen ?

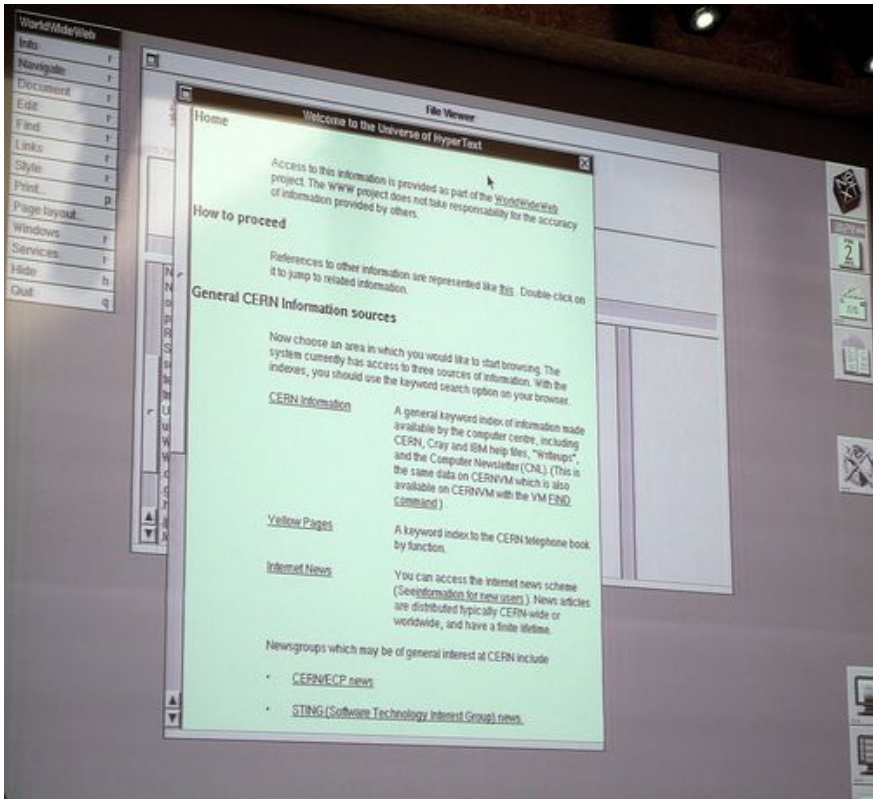

Unfortunately you need MIB files to
understand the best SNMP jokes

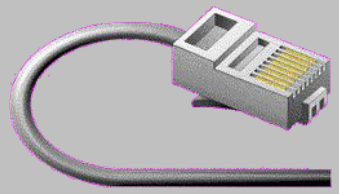


Hyper Text Transfer Protocol

Das Protokoll wurde 1989 von Tim Berners-Lee am CERN zusammen mit der URL-Syntax und HTML-Regeln entwickelt, wodurch das World Wide Web geboren wurde.

Gopher wurde fast gleichzeitig entwickelt

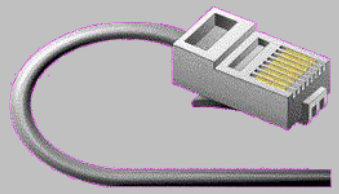




gopher://....

Praktisch gleichzeitig wurde **Gopher** (RFC 1436 a distributed document search and retrieval protocol) als Web ähnliches Protokoll an der University of Minnesota definiert.

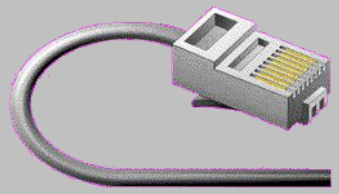
- Gopher verwendet den TCP-Port 70
- Gopher kann verschiedene Datei Typen transferieren.
- Gopher Clients können jeweils nur einen einzigen Type darstellen.
- Im Browser können Gopher Seiten mit der URL gopher://<Host>/<URI> angesehen werden.
  Beispielsweise: gopher://quux.org/



http://....

- **http** verwendet den TCP-Port 80
- **https** verwendet den TCP-Port 443
  Pro Zertifikat ist eine IP notwendig.

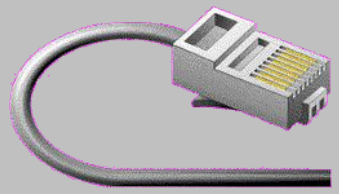
- **http** ist ein Client/Server Protokoll.
  Client ist der Webbrowser. Der Client sendet Request
  Server der Web/Server. Der Server sendet Response
- **http** ist ein stateless Protokoll
- **http** kann zusätzliche Informationen im Header mitsenden. Durch die Angabe des Content-Type können beliebige Datei-Formate Transferiert werden.



http://....

Ein Request für die Seite https://guybrush.maillink.ch/
Der Request wird vom Browser gesendet:

```
GET / HTTP/1.1
Host: guybrush.maillink.ch
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; ...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en,de;q=0.7,de-ch;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: dvg_remember=ueli%3B74ed81ec2e6e9b6edb7e0ca64f4b833d;
```

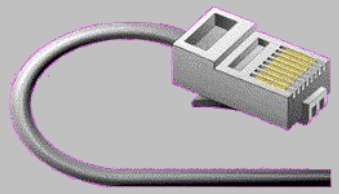


http://....

Die Antwort wird vom Server gesendet:

```
HTTP/1.x 401 Authorization Required
Date: Mon, 05 Oct 2009 08:38:24 GMT
Server: Apache/2.2.6 (Unix) mod_ssl/2.2.6 OpenSSL/0.9.8k DAV/2
WWW-Authenticate: Basic realm="MyCastle"
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
Content-Language: en
```

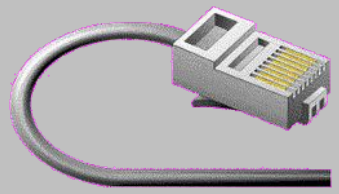
Diese Seite ist Passwort geschützt und es wurde kein (oder ein falcher) Username/Passwort gesendet



http://....

Der Broser sendet eine neue Anfrage, dieses mal mit Username/Passwort

```
GET / HTTP/1.1
Host: guybrush.maillink.ch
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; ...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en,de;q=0.7,de-ch;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: dvg_remember=ueli%3B74baddatad81ec2e6e9b6edb7e0ca64f4b833d;
Authorization: Basic dWVsaTpmbaddatabGxtYQRyOA==
```

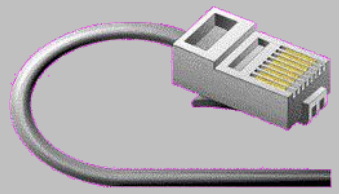


http://....

Die Antwort wird vom Server gesendet:

```
HTTP/1.x 200 OK
Date: Mon, 05 Oct 2009 08:38:27 GMT
Server: Apache/2.2.6 (Unix) mod_ssl/2.2.6 OpenSSL/0.9.8k DAV/2
Last-Modified: Mon, 20 Jul 2009 05:44:24 GMT
Content-Length: 3753
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1
```

Die Seite wird nun im Anhang von diesem Header gesendet.



http://....

Die wichtigsten HTTP Server-Codes:

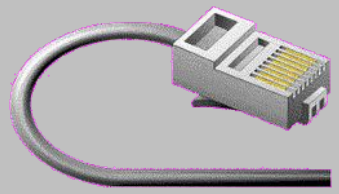
200: OK

301: Moved Permanently
304: Not modified

400: Bad request
401: unauthorized
403: forbidden
404: Not Found

500: Internal Server Error
501: Not Implemented

Siehe auch: `http://de.wikipedia.org/wiki/HTTP-Statuscode`



Fragen ?

Someone was telling me a HTTP 304 joke but I heard it before

http://hardware.localhost.nl/