

8. TCP/IP oder der Anschluss eines UNIX-Rechners ans INTERNET

=====

|

Netzwerk-Schichtenmodell UNIX:

	TCP	UDP
Application Layer	stream	message
Transport Layer	segment	packet
Internet Layer	datagram	datagram
Network Access Layer	frame	frame

Protokolle:

- ARP - Adress Resolution Protocol (RFC 826)
- RARP - Reverse Adress Resolution Protocol (RFC 903)
- IP - Internet Protocol (Verbindungslos)
 - Internet Adressierungsschema
 - Routing, Multi-homed host
 - Definition von Datagrammen
 - Datentransport
 - Fragmentierung
- ICMP - Internet Control Message Protocol
 - Flow Control, Redirect Routes
 - Checking Remote-Hosts (ping)
 - Erkennung nicht erreichbarer Hosts
- UDP - User Datagram Protocol
 - verbindungsloser Datentransport mit
 - Minimum an Overhead (Source Port,
 - Destination Port, Länge, Checksumme)
- TCP - Transmission Control Protocol
 - verbindungsorientierter Datentransport
 - (Positive Acknowledgment with Re-transmission - PAR)
 - Streamorientiert, Checksummen gesichert,
 - Window for Acknowledgment

Applikationsschicht - Beispiele:

SSH - Filetransport (RFC-4250 bis RFC-4254, TCP)
 TELNET - Remote Login (RFC-854, TCP)
 SMTP - Mail-Transport (RFC-821, TCP)
 RIP - Routing Information Protocoll (RFC-1058, UDP)
 DNS - Domain Name Service (RFC-1058, UDP)
 NFS - Network File System (RFC-1094, UDP, RFC-3530, TCP)

Adressierung (IPv4) RFC-791

IP-Adressen: kkk.nnn.nnn.nnn

Aufteilung der IP-Adressen in Klassen:

Klasse	Kennung	Ber.von kkk	Netzwerkbits	Hostbits
A	0	1-126	7	24
B	10	128-191	14	16
C	110	192-224	21	8
D	111	224-254	reserviert	

Subnetze: Unterteilung der Adressklassen in Teilbereiche für die Verwaltung der Netzwerktopologie. Durch die Netzmaske erfolgt die Bildung von Subnetzen (neue Festlegung der Verteilung von Netzwerkbits und Hostbits in der IP-Adresse unabhängig von der Klasse, zu der eine IP-Adresse gehört).

Netzmaske: 255.255.255.192 (6 Bits für Hosts)
 141.20.20.126 und 141.20.20.12 nicht im selben Subnetz
 141.20.20.12 und 141.20.20.56 im selben Subnetz (141.20.20.0)

Broadcastadresse:

IP-Adresse in Subnetzen, die von allen Rechnern akzeptiert wird. Entweder alle Hostbits == 0 oder alle Hostbits == 1. Beachte Subnetze!!!

z.B.

Netzmaske = 255.255.255.192

Subnetz = 141.20.20.0

Broadcast: 141.20.20.0 oder 141.20.20.63

Subnetz = 141.20.20.192

Broadcast: 141.20.20.192 oder 141.20.20.255

Sollte einheitlich in einem Subnetz sein.

Wo erfolgt eine Namenszuordnung von Subnetzen und Namen?

1. /etc/networks - Aufbau:

<Kommentar>

<Subnetzname> <IP-Adresse> # <Kommentar>

Beispiel: /etc/networks

141.20.39.0	fddi	# FDDI
141.20.33.0	rok	# Malek
141.20.31.0	tech-inf	# Meffert
141.20.29.0	teserv	# terminalserver
141.20.28.0	petri	# Starke, Rödel
141.20.27.0	datbank	# Freytag
141.20.26.0	pra-inf2	# Bothe, Burkhard
141.20.25.0	bildung	# Coy
141.20.23.0	pra-inf1	# Polze, Fischer
141.20.21.0	pools	# Pools
141.20.20.0	zentral	# Zentrale Server, RBG

2. NIS, NIS+ (Network Information Service)

3. LDAP

Woher bekomme ich meine IP-Adresse?

Vom Provider

Übersicht über Deutsche Provider:

de-nic Deutsches Network Information Center

<http://www.nic.de/>

Eigene DNS-Server (mindestens zwei getrennte Server)

offiziell registriert

Wo wird die Zuordnung von IP-Adresse und Hostname hinterlegt?

1. File /etc/hosts - Aufbau:

```
# <Kommentar>
```

```
<IP-Adresse> <Hostname> { <Alias-Name>... } # <Kommentar>
```

2. NIS, NIS+ (Network Information Service) (lokales Netzwerk)

3. LDAP (lokales Netzwerk)

4. DNS (Domain Name Service) (weltweit)

Beispiel /etc/hosts

```
# Sun Host Database
```

```
# If the NIS is running, this file is only consulted when booting
```

```
#
```

```
127.0.0.1      localhost loghost
```

```
#
```

```
141.20.22.152  data
```

```
141.20.20.139  adler      # router
```

```
141.20.22.141  enterprise # timehost
```

Routing:

Transport von eines Pakets von einem Host zu einem anderen, der sich in einem anderen Subnetz befindet, über einen oder mehrere Router.

Problem: Bestimmung der Route - sehr kompliziert

Lösung:

1. statische Routen (lokal)
2. Dynamisches Route (Routingprotokolle)

RIP

OSPF

Besichtigung der aktuellen Routen auf einem Rechner:

```
netstat -nr
```

Flags im Ausgabeprotokoll:

- U - Route aktiv
- H - Host-Route
- G - Netz-Route
- D - Dynamische Route (icmp redirect)

Beispiel Protokoll netstat -nr

Routing tables

Destination	Gateway	Flags	Refcnt	Use	Interface
141.20.29.48	141.20.20.40	UGHD	0	0	le0
141.20.29.40	141.20.20.40	UGHD	0	3	le0
141.20.29.41	141.20.20.40	UGHD	0	0	le0
127.0.0.1	127.0.0.1	UH	8	1376572	lo0
141.20.28.130	141.20.20.1	UGHD	0	0	le0
141.20.29.43	141.20.20.40	UGHD	0	11	le0
141.20.22.141	141.20.20.62	UGH	1	410573	le0
141.20.29.46	141.20.20.40	UGHD	0	27	le0
141.20.29.22	141.20.20.40	UGHD	0	12328	le0
141.20.29.30	141.20.20.40	UGHD	0	98	le0
141.20.29.47	141.20.20.40	UGHD	0	35	le0
141.20.28.128	141.20.20.61	UG	0	15	le0
141.20.20.192	141.20.20.40	UG	0	0	le0
default	141.20.20.1	UG	4	1527177	le0
194.94.235.0	141.20.20.61	UGD	0	131	le0
141.20.20.0	141.20.20.18	U	44	63679589	le0

Weiter wichtige Informationen:

Protokollnummern:

/etc/protocols wird auch über NIS, NIS+ und LDAP verteilt

```

#
# Decimal  Keyword      Protocol      References
# -----  -
# protocol num aliases  # comments
hopopt      0 HOPOPT      # IPv6 Hop-by-Hop Option      [RFC1883]
icmp        1 ICMP       # Internet Control Message     [RFC792]
igmp        2 IGMP       # Internet Group Management     [RFC1112]
ggp         3 GGP        # Gateway-to-Gateway           [RFC823]
ip          4 IP         # IP in IP (encapsulation)     [RFC2003]
st          5 ST         # Stream                         [RFC1190,RFC1819]
tcp         6 TCP        # Transmission Control         [RFC793]
cbt         7 CBT        # CBT                             [Ballardie]
egp         8 EGP        # Exterior Gateway Protocol     [RFC888,DLM1]
igp         9 IGP        # any private interior gateway  [IANA]
#
#                               (used by Cisco for their IGRP)
bbn-rcc-mon 10 BBN-RCC-MON # BBN RCC Monitoring           [SGC]
nvp-ii      11 NVP-II     # Network Voice Protocol       [RFC741,SC3]
pup         12 PUP       # PUP                           [PUP,XEROX]
argus       13 ARGUS     # ARGUS                          [RWS4]
emcon       14 EMCON     # EMCON                          [BN7]
xnet        15 XNET      # Cross Net Debugger            [IEN158,JFH2]
chaos       16 CHAOS     # Chaos                          [NC3]
udp         17 UDP       # User Datagram                  [RFC768,JBP]

```


Portnummern:

```
/etc/services      wird auch über NIS, NIS+ und LDAP verteilt
# Network services, Internet style
# This file is never consulted when the NIS are running
echo               7/tcp
echo               7/udp
discard           9/tcp          sink null
discard           9/udp          sink null
ftp-data          20/tcp
ftp               21/tcp
telnet            23/tcp
smtp              25/tcp          mail
time              37/tcp          timserver
time              37/udp          timserver
sunrpc            111/udp
sunrpc            111/tcp
tftp              69/udp
finger            79/tcp
# these are NOT officially assigned
exec              512/tcp
login             513/tcp
shell             514/tcp          cmd          # no passwords used
printer           515/tcp          spooler      # line printer spooler
who               513/udp          whod
syslog            514/udp
talk              517/udp
route             520/udp          router routed
timed             525/udp
http              80/tcp          # 4.10.94 G.B. (D.Verworner)
http_proxy        8000/tcp         # 11.10.94 G.B. (D.Verworner)
```

UNIX-Rechner ins Netzwerk einbinden

Anfangsinformationen beschaffen:

IP-Adress: 141.20.22.97
Hostname: xyz1
Subnetzmaske: 255.255.255.192
Netzwerkadresse: 141.20.22.64
Broadcastadresse: 141.20.22.64 oder 141.20.22.127
Routing:
 Defaultrouter: 141.20.22.65
 oder
 Routingdaemonen: gated oder routed
domainname: informatik.hu-berlin.de
DNS-Server: 141.20.20.51

Basis Konfiguration:

Kernkonfiguration mit TCP/IP, SLIP, PPP
(standard)

Interfacekonfiguration:

Netzwerkkarte - ifconfig, ipadm
SLIP - dip
PPP - chat, pppd

Routingkonfiguration:

route-Kommando
routed (/etc/gateways), gated(/etc/gated.conf)

DNS-Konfiguration(außer einige LINUX-Systeme):

/etc/resolv.conf

nameserver 141.20.20.50
domain informatik.hu-berlin.de
search informatik.hu-berlin.de cms.hu-berlin.de

Allgemein gültige Operationen:**1. /etc/hosts:****aktuellen Host hinzufügen****141.20.22.97 xyz1 [<aliasname>] # ein Beispiel Raum 426, PC****2. /etc/networks****aktuelle Netzwerk hinzufügen****testnet 141.20.22.64 [<aliasname>] # ein Beispiel****3. ... n-1.****Betriebssystem spezifische Aktionen****n. /etc/resolv.conf****anlegen****nameserver 141.20.20.50****domain informatik.hu-berlin.de****search informatik.hu-berlin.de cms.hu-berlin.de**

Betriebssystemspezifische Aktionen:

SunOS:

1. /etc/netmasks aktualisieren
Network masks database

only non-default subnet masks need to be defined here

Network netmask
141.20.22.0 255.255.255.192
141.20.22.64 255.255.255.192
141.20.22.128 255.255.255.192
141.20.22.192 255.255.255.192

2. /etc/hosts
141.20.22.97 xyz1

3. /etc/networks
test 141.20.22 blabla

4. /etc/hostname.le0 anlegen
xyz1

5. /etc/defaultrouter
141.20.22.65

Test von Hand:

```
ifconfig le0 141.20.22.97 netmask 255.255.255.192 \  
        broadcast 141.20.22.64 up
```

```
route add default 141.20.22.65 1
```

```
ping 141.20.20.50
```

```
reboot
```

Solaris (bis 10):

1. /etc/inet/ipnodes anlegen (wie /etc/hosts, /etc/inet/ipnodes)

2. /etc/netmasks aktualisieren

```
#ident "@(#)networks 1.4 92/07/14 SMI" /* SVr4.0 1.1 */
# The networks file associates Internet Protocol (IP)
# network numbers with network names. The format of
# this file is:
#     network-name      network-number  nicnames . . .
#
# The loopback network is used only
# for intra-machine communication
#
loopback          127
#
# Internet networks
#
141.20.22.0      255.255.255.0
```

3. /etc/hostname.ce0 anlegen
xyz1

4. /etc/defaultrouter anlegen
141.20.22.65

5. /etc/nsswitch.conf modifizieren

Solaris 11:

alles bisher gelernt (Solaris, Linux, BSD) vergessen.

dladm show-phys - anzeigen aller verfügbaren Interface

ipadm create-ip net1 # Interface net1 verfügbar machen

ipadm create-addr -T static 141.20.20.33/24 net1/v33 # zuweisen

ipadm show-addr # IP-Adressen anzeigen

ipadm show-if # Interfaces anzeigen

zweite Adresse für ein Interface setzen

ipadm create-addr -T static 141.20.20.34/24 net1/v34

Hostname setzen:

svccfg

select /system/identity:node

listprop config

setprop config/nodename="myhost"

setprop config/loopback="myhost"

Achtung!! danach einige Dienste neu starten.

Routen setzen:

route -p add default 141.20.20.1

route -p add 141.20.22.0 -netmask 255.255.254.0 141.20.20.55

ifconfig nur noch zum anschauen!!!

Files zum Anschauen im Directory /etc/inet/

static_route, ipnodes, hosts, netmasks, networks, protocols, services

DEC-UNIX:

```
Script /usr/sbin/netsetup
      /usr/sbin/bindsetup (für /etc/resolv.conf)
von Hand zusätzlich: /etc/svc.conf
```

```
/etc/rc.config editieren:
```

```
HOSTNAME="xyz1"
IFCONFIG_0="141.20.22.97 netmask 255.255.255.192 \
          broadcast 141.20.22.64"
BIND_CONF="YES"
BIND_SERVERTYPE="CLIENT"
```

```
/etc/routes editieren:
```

```
-default 141.20.22.65
```

```
/etc/svc.conf editieren
```

HP-UX:

```
Kernkonfiguration!!!!
```

```
Konfigurationsprogramm sam!
```

```
/etc/netlinkrc modifizieren:
```

```
/etc/ifconfig lan0 inet `hostname` \
          netmask 255.255.255.192 up
```

```
/etc/lanconfig lan0 ether
```

```
/etc/route add default 141.20.22.65 2 > /dev/null
```

AIX:

```
smit
```


LINUX:

lokales Interface aktivieren:

```
/sbin/ifconfig lo 127.0.0.1
```

```
/sbin/route add -net 127.0.0.0
```

Netzwerk aktivieren (Slackware)

editieren von z.B. /etc/rc.d/rc.inet1:

```
IPADDR="141.20.22.95"
```

```
NETMASK="255.255.255.192"
```

```
NETWORK="141.20.22.64"
```

```
BROADCAST="141.20.22.64"
```

```
GATEWAY="141.20.22.65"
```

```
/sbin/ifconfig eth0 ${IPADDR} broadcast ${BROADCAST} netmask ${NETMASK}
```

```
/sbin/route add -net ${NETWORK} netmask ${NETMASK}
```

```
/sbin/route add default gw ${GATEWAY} metric 2
```

SuSE(11.1 ohne Network Manager):

/etc/sysconfig/network/config

Allgemeine Informationen zur Netzwerkkonfiguration

```
NETWORKMANAGER="no"
DEFAULT_BROADCAST="+"
GLOBAL_POST_UP_EXEC="yes"
GLOBAL_PRE_DOWN_EXEC="yes"
CHECK_DUPLICATE_IP="no"
DEBUG="no"
USE_SYSLOG="yes"
CONNECTION_SHOW_WHEN_IFSTATUS="no"
CONNECTION_CHECK_BEFORE_IFDOWN="no"
CONNECTION_CLOSE_BEFORE_IFDOWN="no"
CONNECTION_UMOUNT_NFS_BEFORE_IFDOWN="no"
CONNECTION_SEND_KILL_SIGNAL="no"
MANDATORY_DEVICES="wlan0 eth0"
WAIT_FOR_INTERFACES="30"
FIREWALL="yes"
LINKLOCAL_INTERFACES="eth*[0-9]|tr*[0-9]|wlan[0-9]|ath[0-9]"
IFPLUGD_OPTIONS="-f -I -b"
NM_ONLINE_TIMEOUT="0"
NETCONFIG_MODULES_ORDER="dns-resolver dns-bind dns-dnsmasq nis ntp-runtime"
NETCONFIG_DNS_POLICY="auto"
NETCONFIG_DNS_FORWARDER="resolver"
NETCONFIG_DNS_STATIC_SEARCHLIST="informatik.hu-berlin.de"
NETCONFIG_DNS_STATIC_SERVERS=""
```

```
NETCONFIG_NTP_POLICY="auto"  
NETCONFIG_NTP_STATIC_SERVERS=""  
NETCONFIG_NIS_POLICY="auto"  
NETCONFIG_NIS_SETDOMAINNAME="yes"  
NETCONFIG_NIS_STATIC_DOMAIN=""  
NETCONFIG_NIS_STATIC_SERVERS=""
```

```
/etc/sysconfig/network/ifcfg-eth0
```

Allgemeine Informationen über die Schnittstelle und die Aktivierung

```
NM_CONTROLLED=no  
BOOTPROTO='static'  
BROADCAST=''  
ETHTOOL_OPTIONS=''  
IPADDR='141.20.20.17/24'  
MTU=''  
NAME='82567LM Gigabit Network Connection'  
NETMASK=''  
NETWORK=''  
REMOTE_IPADDR=''  
STARTMODE='auto'  
USERCONTROL='no'  
IFPLUGD_PRIORITY='20'  
DHCLIENT_PRIMARY_DEVICE="yes"
```

Muster: /etc/sysconfig/network/ifcfg.template

```
/etc/sysconfig/ifroute-eth0
```

Routen, die gesetzt werden sollen, wenn die Schnittstelle (eth0) aktiviert wird

```
141.20.21.0 141.20.21.1 255.255.255.0 eth0
default 141.20.20.1 255.255.255.0 eth0
```

```
/etc/sysconfig/network/routes
```

Routen, die gesetzt werden sollen, wenn kein Routen-File für eine Schnittstell existiert

```
141.20.21.0 141.20.21.1 255.255.255.0 eth0
default 141.20.20.1 255.255.255.0 eth0
```

Aktivieren/Deaktivieren von Schnittstelle

```
/sbin/rcnetwork start eth0
/sbin/rcnetwork stop eth0
```

RedHat

```
/etc/sysconfig/network
/etc/sysconfig/network-scripts/ifcfg-eth0
```

LINUX - Networkmanager

Neuere Programmfamilie zum Verwalten von Netzwerkschnittstellen durch einen Nutzer. Besonders für Notebooks empfehlenswert. Nicht geeignet für Server.

Bestandteile:

NetworkManager - Kommando, Network management daemon
Versucht an Hand der Vorgaben eine Netzwerkverbindung aufzubauen.

/etc/NetworkManager/ - Directory

NetworkManager.conf - Konfigurationsfile für NM

system-connections - Directory mit Verbindungen

VPN - Directory mit VPNs

/var/lib/NetworkManager/ - Directory mit Statusinformationen des NM

Tools zur Verwaltung:

nm-tool - Ausgabe des Zustandes und der Geräte des NM

nm-tool

nm-online - zeigt (manchmal) aktive Verbindung an

nm-online

```
nmcli          - Tool zum Steuern von NM, Kommandozeilen orientiert
nmcli [ OPTIONS ] OBJECT { COMMAND | help }

OPTIONS := { -t[erse]
             -p[retty]
             -m[mode] tabular | multiline
             -f[ields] <field1,field2,...> | all | common
             -e[scape] yes | no
             -v[ersion]
             -h[elp] }
OBJECT := { nm | con | dev }

COMMAND - Kommandos, objektspezifisch
nm: status, enable, sleep, wifi, wwan
    nmcli -p nm status
con: list, status, up, down, delete
    nmcli -p con list
    nmcli con list uuid a6ffcdb6-ee75-4541-85b1-b3f1a077fabe
    nmcli con down uuid a6ffcdb6-ee75-4541-85b1-b3f1a077fabe
dev: status, list, disconnect, wifi
    nmcli dev status
    nmcli -p dev list
```

Kommandos im Detail

ifconfig - Schnittstelle konfigurieren

SunOS:

ifconfig interface

```
[-af] [address [dest_addr]] [u ][down] [netmask mask]
[broadcast broad_addr] [metric n] [trailers|-trailers]
[private|-private] [arp|-arp]
```

ifconfig -a

```
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
      inet 141.20.20.18 netmask fffffffc0 broadcast 141.20.20.0
      ether 8:0:20:1d:8d:30
lo0: flags=49<UP,LOOPBACK,RUNNING>
      inet 127.0.0.1 netmask ff000000
```

Solaris:

```
ifconfig interface [address_family] [address [dest_address] ]
[up] [down] [auto-revarp] [netmask mask] [broadcast address]
[metric n] [mtu n] [trailers|-trailers] [private|-private]
[arp|-arp] [plumb] [unplumb]
```

\$ /sbin/ifconfig -a

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
      inet 127.0.0.1 netmask ff000000
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
      inet 141.20.22.60 netmask fffffffc0 broadcast 141.20.22.63
nf0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 4352
      inet 141.20.39.15 netmask fffffffc0 broadcast 141.20.39.63
```

DEC-UNIX:

```
ifconfig interface_id [address_family] [address [dest_address]]
[parameters]
```

HP-UX:

```
ifconfig interface address_family [address [dest_address]] [parameters]
ifconfig interface [address_family]
```

AIX:

```
ifconfig Interface [AddressFamily [Address [DestinationAddress]]
[ Parameter ... ]]
```

LINUX:

```
ifconfig [interface]
ifconfig interface [aftype] options | address ....
```

options:

up	- Interface aktiv
down	- Interface inaktiv
[-]arp	- ARP-Steuerung
[-]trailers	- Trailers
[-]almulti	- Alle Pakete an Kernel
metric n	- Metrik
mtu N	- Paketlänge
dstaddr addr	- Zieladresse PPP
netmask addr	- Netzmaske
irq addr	- IRQ
broadcast addr	- Broadcastadresse
hw	- Hardwareadresse
multicast	- Multicast


```
# /sbin/ifconfig -a
```

```
eth0  Link encap:Ethernet  Hardware Adresse 00:21:86:FF:74:EC
      inet Adresse:141.20.20.17  Bcast:141.20.20.255  Maske:255.255.255.0
      UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
      RX packets:22548 errors:0 dropped:0 overruns:0 frame:0
      TX packets:3662 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 Sendewarteschlangenlänge:1000
      RX bytes:11452958 (10.9 Mb)  TX bytes:334794 (326.9 Kb)
      Speicher:fc600000-fc620000
```

```
lo    Link encap:Lokale Schleife
      inet Adresse:127.0.0.1  Maske:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:54 errors:0 dropped:0 overruns:0 frame:0
      TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 Sendewarteschlangenlänge:0
      RX bytes:3494 (3.4 Kb)  TX bytes:3494 (3.4 Kb)
```

route - Routingtabelle manipulieren (statische Routen)

SunOS

```
route [-fn] add|delete [host|net] destination [gateway [metric] ]
```

Solaris:

```
route [ -fn ] [-p] add | delete [ host | net ] destination  
[ gateway [ metric ] ]
```

DEC-UNIX:

```
route [-nqv] command [modifiers] [args]  
route add -net default 141.20.39.1
```

HP-UX:

```
/etc/route [-f] [-n] add [net | host] destination gateway [count]  
/etc/route [-f] [-n] delete [net | host] destination gateway [count]  
/etc/route -f [-n]
```

AIX:

```
route [-n ] [-q] [-v] Command [Family] [ [-net|-host]  
Destination [-netmask [Address]] Gateway] [Arguments]
```

LINUX:

```
route [-vn]  
route [-v] add [-net | -host] destination [gw gateway] [metric MMM]  
[netmask NNNN][mss NNNN] [window NNNN] [dev DDD]  
route [-v] del destination
```

Beispiel:

```
/sbin/route -v
```

```
Kernel IP Routentabelle
```

Ziel	Router	Genmask	Flags	Metric	Ref	Use	Iface
141.20.20.0	*	255.255.255.0	U	0	0	0	eth0
link-local	*	255.255.0.0	U	0	0	0	eth0
loopback	*	255.0.0.0	U	0	0	0	lo
default	141.20.20.1	0.0.0.0	UG	0	0	0	eth0

arp - Manipulieren des ARP

```
arp hostname
arp -a [/vmunix] [/dev/kmem]
arp -d hostname
arp -s hostname ether_addr [temp] [pub] [trail]
arp -f filename
```

```
arp [-vn] [-H <Type>] [-i <if>] [-a] [<Hostname>]
arp [-v] [-i <if>] -d <Hostname> [pub][nopub]
arp [-vnD] [-H <Type>] [-i <if>] -f <Dateiname>
arp [-v] [-H <Type>] [-i <if>] -s <Rechnername> <hwaddr> [temp][nopub]
arp [-v] [-H <Type>] [-i <if>] -s <Hostname> <hwaddr> [netmask <nm>] pub
arp [-v] [-H <Type>] [-i <if>] -Ds <hostname> <if> [netmask <nm>] pub
-a Alle Hosts im BSD-Format anzeigen
-s, --set Neuen ARP-Eintrag setzen
-d, --delete Einen bestimmten Eintrag löschen
-v, --verbose Ausführliche Ausgaben
-n, --numeric don't resolve names
-i, --device Netzwerksgerät (z.B. eth0) angeben
-D, --use-device <hwaddr> von gegebenem Gerät lesen
-A, -p, --protocol Routentabelle anzeigen
-f, --file Neue Einträge aus Datei lesen
```

<Type> - Hardwaretype - Standard: ether

möglicher Hardwaretypenn: strip, ether, tr, ax25, netrom,
arcnet, dlci, fddi, hippi, irda, x25

Beispiel:

```
# arp -a
rt2-21 (141.20.20.1) at 8:0:2:4:f2:78
hp832 (141.20.20.50) at 8:0:9:1:9e:f8
sigma (141.20.20.51) at 0:0:f8:21:b9:70
henrik (141.20.20.35) at 0:20:78:c:1:24
wolf (141.20.20.36) at 0:20:78:c:1:25
teserv (141.20.20.40) at 0:0:c0:e1:c9:75
hub22 (141.20.20.9) at 8:0:4e:5:9f:d6
fob (141.20.20.60) at 8:0:20:71:1:fb
rtisdnl (141.20.20.61) at 0:20:a9:0:4c:a6
epsilon (141.20.20.62) at 8:0:2b:3d:de:8b
```

```
ping - Netzwerkverbindung testen
```

```
ping host [timeout]
```

```
ping -s[drvRl] [-t ttl] host [data size] [npackets]
```

```
# ping hp832
```

```
hp832 is alive
```

```
# ping -s hp832
```

```
PING hp832: 56 data bytes
```

```
64 bytes from hp832 (141.20.20.50): icmp_seq=0. time=4. ms
```

```
64 bytes from hp832 (141.20.20.50): icmp_seq=1. time=2. ms
```

```
64 bytes from hp832 (141.20.20.50): icmp_seq=2. time=2. ms
```

```
64 bytes from hp832 (141.20.20.50): icmp_seq=3. time=3. ms
```

```
^C
```

```
----hp832 PING Statistics----
```

```
4 packets transmitted, 4 packets received, 0% packet loss
```

```
round-trip (ms)  min/avg/max = 2/2/4
```

```
#
```

traceroute - Route testen

```
traceroute [-dnrv] [-w wait] [-m max_ttl] [-p port#] \  
          [-q nqueries] [-t tos] [-s src_addr] host [data size]
```

```
# traceroute bruno
```

```
traceroute to bruno (141.20.28.11), 30 hops max, 40 byte packets
```

```
1  rt2-21 (141.20.20.1)  2 ms  2 ms  2 ms  
2  rtk-01 (141.20.39.1)  3 ms  2 ms  3 ms  
3  bruno (141.20.28.11)  3 ms  2 ms  2 ms
```

```
# traceroute amsel
```

```
traceroute to amsel (141.20.22.60), 30 hops max, 40 byte packets
```

```
1  rt2-21 (141.20.20.1)  2 ms  2 ms  2 ms  
2  amsell (141.20.39.15)  2 ms *  2 ms
```

```
#
```

Standard-Netzwerkapplikationen:

Portmapper: rpcbind
 portmapper

Internet-Superserver
 inetd
 xinetd

Portmapper:

portmapper - alte Fassung
rpcbind - neue Fassung

Wird für NIS und NFS benötigt, realisiert Remote Procedure Call (RPC)

Internet-Superserver - inetd (alte Fassung)

Konfigurationsfile: /etc/inetd.conf

```
# If you make changes to this file, either reboot
# your machine or send the inetd a HUP signal:
# "kill -HUP <pid of inetd>".
# The inetd will re-read this file whenever it gets that signal.
# <service> <socket> <proto> <flags> <user> <server> <args>
#  name          type          path
echo      stream  tcp  nowait  root    internal
echo      dgram   udp  wait    root    internal
time     stream  tcp  nowait  root    internal
time     dgram   udp  wait    root    internal
ftp      stream  tcp  nowait  root    /usr/sbin/tcpd  wu.ftpd
telnet   stream  tcp  nowait  root    /usr/sbin/tcpd  in.telnetd
nntp     stream  tcp  nowait  root    /usr/sbin/tcpd  in.nntp
smtp     stream  tcp  nowait  root    /usr/sbin/tcpd  sendmail -v
shell    stream  tcp  nowait  root    /usr/sbin/tcpd  in.rshd -L
login    stream  tcp  nowait  root    /usr/sbin/tcpd  in.rlogind
exec     stream  tcp  nowait  root    /usr/sbin/tcpd  in.rexecd
talk     dgram   udp  wait    root    /usr/sbin/tcpd  in.talkd
pop3     stream  tcp  nowait  root    /usr/sbin/tcpd  in.pop3d
tftp     dgram   udp  wait    nobody   /usr/sbin/tcpd  in.tftpd
bootps   dgram   udp  wait    root    /usr/sbin/in.bootpd  in.bootpd
finger   stream  tcp  nowait  nobody  /usr/sbin/tcpd  in.fingerd -w
netstat  stream  tcp  nowait  root    /usr/sbin/tcpd  /bin/netstat -a
auth     stream  tcp  nowait  root    /usr/sbin/in.identd  in.identd
rstatd/1-3  dgram  rpc/udp  wait  root    /usr/sbin/tcpd  rpc.rstatd
```

```
walld/1 dgram rpc/udp wait root /usr/sbin/tcpd rpc.rwalld
```

Internet-Superserver - xinetd (neue Fassung)

Funktion wie inetd, Konfigurationsfiles unter:

```
/etc/xinetd.d/
```

für jeden Dienst ein File, z.B.

```
/etc/xinetd/rsync
```

```
# default: off
# description: rsync file transfer daemon
service rsync
{
    socket_type      = stream
    protocol        = tcp
    wait            = no
    user            = root
    server          = /usr/sbin/rsyncd
    server_args     = --daemon
    disable         = yes
}
```