Zentraleinrichtung Rechenzentrum der TU Berlin
Kommunikationshandbuch
TCP/IP

Wolfgang Ksoll

17. Januar 1992

Inhaltsverzeichnis

1 Einführung

Im Rahmen des Projektes WOTAN [1] [2] hat die Zentraleinrichtung Rechenzentrum (ZRZ) damit begonnen, ihre EDV–Infrastruktur umzustellen. Nicht mehr die Großrechner eines Herstellers stehen im Vordergrund, sondern viele Workstations am Arbeitsplatz, die sich Dienstleistungen von zentralen Servern erbringen lassen. Workstations und Server sind über ein campusweites Netzwerk zunächst auf Ethernetbasis verbunden. Da Rechner vieler Hersteller eingesetzt werden, ist es notwendig die Kommunikation auf dem Netz nach festgelegten Protokollen abzuwickeln und nicht mit Hilfe von Produkten bestimmter Hersteller [2].

Wünschenswert als Protokolle wären die ISO–OSI–Normen, die aber noch nicht weit verbreitet und für alle Rechner verfügbar sind. Deshalb werden zunächst die TCP/IP–Protokolle (Transmission Control Protocol/Internet Protocol) eingesetzt, zumal entsprechende Produkte für alle UNIX–Rechner verfügbar und in der Regel Teil des Betriebssystems sind.

Was gehört zu den TCP/IP[3]–Protokollen?

TELNET	Telnet Protocol Specification	RFC 854
FTP	File Transfer Protocol	RFC 959
SMTP	Simple Mail Transfer Protocol	RFC 821
NFS	Network File System	RFC 1094
X11	X Window System, Version 11	RFC 1013

und die dazugehörigen unterliegenden Protokolle

IP	Internet Protocol	RFC 791
ICMP	Internet Control Message Protocol	RFC 792
TCP	Transmission Control Protocol	RFC 793
UDP	User Datagramm Protocol	RFC 768
ARP	Adress Resolution Protocol	RFC 826

Daneben gibt es Produkte einzelner Hersteller, die zwar auch auf dem Ethernet existieren können, aber nicht einem herstellerübergreifenden Normungsverfahren unterworfen waren. Dazu gehören DECNET und XNS, netzwerkweite Dateisysteme von Apollo, PCS oder Novell oder die BSD–Dienste wie `rlogin, rsh, rcp, rwho`.

Ziel dieser Schrift ist es, den Benutzer des Netzwerkes möglichst schnell mit der Handhabung der Dienste vertraut zu machen, dem Systemadministrator einen Überblick über seine Aufgaben im Netzwerksbereich zu verschaffen und bei auftretenden Störungen erste Hinweise zu geben. Es soll nicht versucht werden, die Handbücher der Software–Hersteller zu ersetzen.

Wenn immer es geht und sinnvoll ist, wird versucht, neben den Produkten der TCP/IP–Welt auch die der ISO–OSI–Welt zu nennen, um bei einer Migration die erworbenen Kenntnisse möglichst lange verwerten zu können.

2 Benutzerdienste

In den TCP/IP–Protokollen sind schon seit vielen Jahren drei wesentliche Benutzerdienste genormt:
- der **interaktive Zugang** zu einem entfernten Rechner mit TELNET
- der **Dateitransfer** zwischen zwei Rechnern mit FTP (File Transfer Protocol)

[1] WOTAN = **WO**rkstations der **T**echnischen Universität **A**m **N**etz

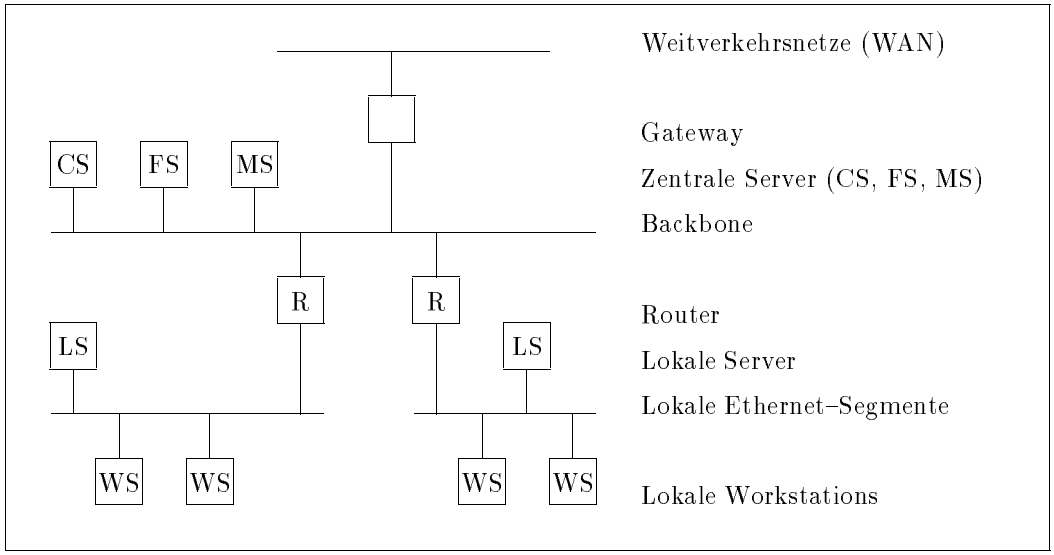
[2] Wotan (alias Odin), germanischer Gott, wohnt mit Frigg (alias Fricka), der Göttin des heimischen Herdes, Thor (alias Donar), dem Gott des Gewitters, und Thyr, dem Rechts– und Kriegsgott, in Walhall. Ihm dienen die Walküren und die Einherier.

[3] RFC = Request For Comment, Normenpapier in der TCP/IP–Welt

- und die **elektronische Post** (E–Mail) mit SMTP (Simple Mail Transfer Protocol).
Diese drei Protokolle wurden ursprünglich für Weitverkehrsverbindungen im US–amerikanischen Internet entwickelt, haben aber mit dem Aufkommen von Workstations eine weite Verbreitung in LAN's (Local Area Networks) gefunden.

Mit dem Einsatz schneller LAN's (10 MBit/s) sind auch weitere Dienste möglich geworden, wie NFS (Network File System) als **verteiltes Dateisystem** und X–Windows als **verteiltes Graphiksystem**.

Bei den weiteren Ausführungen wird das folgende Modell einer verteilten Rechnerlandschaft zugrunde gelegt:



Der Benutzer hat als Schnittstelle zu dem System seine lokale Workstation (Arbeitsplatzrechner). Innerhalb desselben Gebäudes oder derselben Abteilung ist dieser Rechner über ein Ethernet mit anderen Workstations oder lokalen Servern verbunden. Dabei kann z.B. ein lokaler Server den Workstations mittels NFS Plattenplatz zur Verfügung stellen.

Das Hausnetz wird über einen Router an den Backbone angeschlossen, der mehrere Gebäude miteinander verbindet. Dabei kann der Backbone ein anderes Medium als innerhalb des Hauses benutzen, z.B. Glasfasern anstelle eines Koaxialkabels. Neben der Signalumsetzung auf ein anderes Medium hat der Router vor allem die Aufgabe, den lokalen Verkehr auch lokal zu halten und nur Daten durchzulassen, die zu einer Verbindung zwischen verschiedenen lokalen Netzsegmenten gehören.

Am Backbone können dann zentrale Server erreichbar sein, die allen Abteilungen zur Verfügung stehen sollen. Neben einem Computeserver (CS) für rechenintensive Aufgaben stehen u.a. ein Fileserver (FS) mit großer Speicherkapazität und Sicherungsaufgaben sowie ein Mailserver (MS) zur Abwicklung verschiedener E–Mail–Aufgaben bereit.

Ein Gateway (GW) kann eine Protokollumsetzung vornehmen, z.B. von TCP/IP zu den OSI–Protokollen.

Anwendungsbeispiele:

1. Mit `ftp` wird ein FORTRAN–Programm mit Eingabedateien von der lokalen Workstation zum Compute–Server transferiert, wo es dann abgearbeitet werden soll. Mit `telnet` kann es dann gestartet oder in eine Warteschlange gestellt werden. Das Ergebnis des Rechenlaufes wird mit `ftp` oder `mail` zurückgeschickt.

2. Für eine Datenbankabfrage (Literaturrecherche) in einer anderen Stadt wird auf der lokalen Workstation `telnet` aufgerufen und eine Verbindung über den Router zum X.29–Gateway aufgebaut. Von dort wird dann der Datenbankrechner im X.25–Netz angewählt.

3. Post aus dem X.400–Netz wird vom Mailserver empfangen und an die lokale Workstation über SMTP weitergeleitet.

4. Die Workstations sind diskless und werden durch ihren lokalen Server mit Plattenkapazität über NFS versorgt.

5. Der lokale Server ist ein leistungsstarker Mini–Rechner, der Berechnungen durchführt und die Ergebnisse auf der Workstation graphisch über X–Windows visualisiert.

2.1 Telnet

Mit dem Programm `telnet` kann der Benutzer eine Verbindung zu einem entfernten Rechner aufbauen und dort interaktiv arbeiten. In der OSI–Welt wird diese Funktionalität durch das VT (Virtuelle Terminal) [4] erbracht.

Grundsätzlich arbeitet `telnet` zeilenorientiert. Aber die meisten UNIX–Implementationen schalten beim Aufruf automatisch die zeichenorientierte Option ein. Jedes Zeichen, das an der Tastatur eingegeben wird, wird sofort zum entfernten Rechner gesendet. Bei UNIX-Anlagen kommt das Zeichen als Echo sofort zurück und wird auf dem Bildschirm dargestellt. `telnet` interpretiert den Zeichenstrom nicht. Was die lokale Workstation z.B. mit Escape-Sequenzen macht, obliegt nicht `telnet` sondern der Terminal–Charakteristik. Diese kann auf UNIX-Anlagen sehr unterschiedlich sein, je nachdem ob man an einer graphikfähigen Console sitzt oder an an einem seriellen Zeichen–Terminal arbeitet. Auf MS–DOS–Rechnern fallen `telnet`–Programm und Emulation einer Terminal–Charakteristik häufig zusammen. Oft wird ein DEC–Terminal VT100 emuliert (wie auch in `xterm` in X–Windows).

2.1.1 Verbindungsaufbau, –abbau, Sitzungssteuerung

Für alle Programme (`telnet, ftp, mail`), sollte sichergestellt sein, daß der Suchpfad das Directory enthält, in dem sie stehen. Oft ist dies das Directory `/usr/ucb` (SunOS, 386/ix), manchmal auch `/usr/bsd`. Bei MS–DOS–Rechnern ist dies sehr unterschiedlich (`C:\NCSA, C:\NFS, C:\WD8003,` usw.).

Ruft man das Programm `telnet` ohne Parameter auf, so befindet man sich im Kommando–Modus. Dann kann von Hand die Verbindung aufgebaut werden (`open`). Zum Verbindungsaufbau muß die IP–Adresse oder der Name des Zielrechners eingegeben werden.

Einfacher ist es, beim Aufruf des Programms gleich den Zielrechner mit Adresse oder Namen anzugeben:

`telnet <ip-addresse>` [5] oder

`telnet <hostname>`

Das Programm versucht dann eine Verbindung zum Zielrechner aufzubauen. Gelingt dies, so meldet sich der Zielrechner mit einer freundlichen Meldung und man kann an ihm arbeiten, als wenn man an einem Terminal direkt an ihm angeschlossen wäre. Gelingt der Aufbau nicht, so landet `telnet` im Kommando-Modus, den man mit `quit` verlassen kann.

[4] Da dieses aber noch nicht abschließend genormt ist, findet man zur Zeit eine Hilfslösung unter dem Namen PAD (Packet–Assembler–Disassembler nach X.29/X.28/X.3) für den interaktiven Zugang zu weit entfernten Rechnern, wenn man nicht TCP/IP verwendet.

[5] Eine IP–Adresse besteht aus vier Dezimalzahlen zwischen 0 und 255, die durch Punkte getrennt werden. Z.B.: 130.149.2.16. IP–Adressen sind weltweit eindeutig und fangen an der TUB mit 130.149 an.

Der Aufruf mit der Adresse eines Rechners gelingt immer, wenn das Netzwerk arbeitet. Der Aufruf mit dem Namen eines Rechners gelingt nur, wenn der Name in eine Adresse umgesetzt werden kann. Dafür muß der Name entweder in der Datei `/etc/hosts` vorhanden sein, oder der Domain–Name–Service kann den Namen in eine Adresse übersetzen oder Yellow-Pages müssen den Dienst erbringen. Yellow Pages ist ein Herstellerprotokoll der Firma Sun zur netzwerkweiten Verwaltung von Benutzern, Rechnernamen, Netzwerksnamen usw., das auch von einigen anderen Herstellern angeboten wird.

Will man während einer `telnet`–Sitzung in den Kommando–Modus, so gelingt dies durch die Eingabe des Fluchtsymbols. In der Regel ist dies `^]` (Ctrl–]). Das Fluchtsymbol kann im Kommando–Modus umgesetzt werden.

Die wichtigsten Kommandos im Kommando–Modus sind:

`close`	schließt eine bestehende Verbindung
`open`	öffnet eine neue Verbindung
`quit`	beendet eine `telnet`–Sitzung
`?`	gibt eine Beschreibung der Befehle

2.1.2 Full–Screen–Anwendungen

Geht man von einer lokalen Workstation mit `telnet` an einen entfernten Rechner und möchte dort eine Full–Screen–Anwendung starten, z.B. den Editor `vi`, so muß die entfernte Anlage über den Terminal–Typ des lokalen Rechners informiert werden. Viele Telnet–Implementationen verhandeln den Terminaltyp automatisch, indem die Umgebungsvariable `TERM` (bei der Bourne-Shell) oder `term` (bei der C-Shell) ausgelesen und dem entfernten Rechner mitgeteilt wird.

Damit dies reibungslos funktioniert, muß die entfernte Anlage den übermittelten Terminal–Typ auch kennen. Das heißt hier, daß auf BSD-Anlagen für das Terminal ein Eintrag in `/etc/termcap` vorhanden sein oder bereitgestellt werden muß oder auf System–V–Anlagen ein `terminfo`–Eintrag. Dies ist bei den wenigsten modernen Terminal–Typen der Fall. Denkbar wäre, daß der Systemadministrator alle möglichen Terminal–Typen auf seiner Anlage nachhält. Aufgrund der Vielfalt von möglichen Einträgen ist aber nicht damit zu rechnen, daß er es wirklich tut.

Eine Ausweichmöglichkeit für den Benutzer ist es, weit verbreitete Terminaltypen zu verwenden. Auf UNIX-Workstations, die X-Windows bereithalten, ist dies durch den Einsatz von `xterm` möglich. `xterm` verhält sich wie ein VT100, das jede UNIX-Anlage kennt. MS–DOS–Benutzer können durch den Einsatz von NCSA–Telnet (Public Domain, kostenlos, s.u.) eine VT100–Emulation benutzen.

Hat `telnet` beim Verbindungsaufbau aus Versehen einen ungewünschten Terminal–Typ übermittelt (z.B. VT102 oder einen exotischen Typ), kann man sich durch den Aufruf von `set` über die aktuelle Einstellung informieren und mit

`TERM=vt100 ; export TERM` (in der Bourne-Shell) oder

`set term=vt100` (in der C-Shell)

die Einstellung korrigieren.

Manchmal ist es hilfreich, mit `stty` weitere Terminaleinstellungen zu überprüfen. Besonders sollte man darauf achten, welche Zeichen als kill-, intr- und erase-character interpretiert werden. System–V–Maschinen verwenden gerne @ und # . Diese Zeichen stehen dann nicht mehr für andere Zwecke zur Verfügung (speziell für Mail und C-Programme). Günstig ist deshalb folgende Setzung:

```
stty kill '^U' intr '^C' erase '^H'
```

Dies bedeutet im Einzelnen: wird `^U` (Ctrl-U) eingegeben, wird die bisherige Kommandozeile weggeworfen. Mit `^C` (Ctrl-C) werden laufende Programme abgebrochen und mit `^H` (Backspace) wird das letzte Zeichen der Kommandozeile gelöscht.

2.2 Ftp — Dateitransfer

Mit dem Programm `ftp` kann man Textdateien oder Binärdateien von einem Rechner zu einem anderen transferieren. Diese Funktionalität wird in der OSI–Welt mit erheblich mehr Möglichkeiten durch **FTAM** (File Transfer, Access and Management) erbracht.

Der Aufruf von `ftp` kann wie bei `telnet` mit IP–Adresse oder Zielrechnernamen erfolgen:

`ftp <ip-addresse>` oder

`ftp <hostname>`

Gelingt der Aufbau zum Zielrechner, wird dort ein `ftpd` als Serverprogramm gestartet, das als erstes eine Validierung des Benutzers mit Benutzernamen und Paßwort vornimmt. Werden keine weiteren Angaben gemacht, so nehmen beide Rechner an, es sollen ASCII–7–Bit–Dateien übertragen werden. Dabei werden notwendige Korrekturen bei unterschiedlicher Zeilenende–Behandlung automatisch ausgeführt.

In der Regel stehen folgende Befehle zur Verfügung:

bin	die Dateiübertragung soll Binärdateien betreffen
ascii	es sollen Textdateien übertragen werden
get	eine benannte Datei soll der Server schicken
put	eine benannte Datei soll der Server empfangen
mget	mehrere Dateien soll der Server schicken
mput	mehrere Dateien soll der Server empfangen
prompt	Umschalten, ob bei jeder Datei eine Bestätigung erfolgen soll
cd	auf dem Server soll das Directory gewechselt werden
dir	der Inhalt des Directorys auf dem Server wird angezeigt
lcd	auf dem Client soll das Directory gewechselt werden
?	alle implementierten Befehle werden angezeigt
help	Kurztext für jeden Befehl

Will man die Dateiübertragung automatisieren, weil man sie zum Beispiel in einem Shell–Script aufruft, so kann man im eigenen **HOME**–Directory auf der Maschine, auf der `ftp` aufgerufen wird, eine Datei `.netrc` anlegen, in der für jede Zielmaschine ein Tripel folgender Art existiert:

```
machine Rechnername
login meinAccount
password geheim
```

Dieser automatisierte `ftp` gelingt nur, wenn die Datei `.netrc` lediglich für den Eigentümer lesbar ist (Dateizugriffsrechte: 600 oder `rw-------`). **Diese Arbeitsweise birgt ein Sicherheitsrisiko und verstößt gegen die Philosophie, daß sich auf einer UNIX-Anlage kein Paßwort in Klartext befinden soll**.

2.3 Mail — Elektronische Post

Elekronische Post (E–Mail) basiert bei TCP/IP auf dem SMTP–Protokoll (Simple Mail Transfer Protocol, RFC 821). Mit SMTP können ASCII–Text–Dateien begrenzter Grösse verschickt und empfangen werden. Wie in den OSI–Protokollen (X.400 ff.) wird eine funktionale Trennung zwischen dem Zustelldienst (Message Transfer Agent, MTA) und dem Benutzer–Zugriff (User Agent, UA) vorgenommen. Auf Unix–Anlagen werden diese Funktionen durch die Programme `sendmail` (als MTA) und `mail` (als UA) realisiert. Beide zusammen stellen den überwiegenden Teil eines Message Handling Systems (MHS) dar.

Das Programm `sendmail` arbeitet meist als Dämon, der auf einkommende Post wartet. Ausgehende Post wird sofort zugestellt. Ist die Gegenseite nicht erreichbar, wird der Zustellversuch alle 30 Minuten wiederholt, bis zu drei Tage lang.

Der Benutzer kommuniziert aber über `mail` mit dem System. Der einfache Aufruf:

`mail`

schaut nach, ob neue Post eingegangen ist. Mit

`mail user` (für einen Benutzer am gleichen System) oder

`mail user@host` (für den Benutzer `user` am entfernten Rechner `host`)

versetzt man das Programm in den Eingabemodus. Dann kann über die Tastatur eine Nachricht eingegeben werden, bis das Dateiende mit `^D` (UNIX) angezeigt wird. Vorgefertigte Dateien können mit

`mail user@host < datei`

dem Zustelldienst übergeben werden.

Neben dem AT&T–Programm `mail` gibt es noch einige andere Mail–Programme. Besonders zu nennen ist das `mailx`, das auf manchen Anlagen auch als `/usr/ucb/mail` zu finden ist. Bei interaktivem Aufruf ist es an der Frage nach dem Betreff (`Subject:`) zu erkennen. Es gestattet wesentlich mehr Konfigurationsmöglichkeiten. So können Aliase definiert werden, der Editor der Wahl zum Zusammenstellen von Nachrichten benannt werden und vieles mehr.

Hat man auf mehreren Rechnern Accounts und möchte nur auf einem Rechner einkommende Post abarbeiten, kann man auf allen anderen eine Datei `.forward` im `HOME`–Directory anlegen, in der steht, wohin einkommende Post weitergeleitet werden soll. Es genügt ein einzeiliger Eintrag der Form

`user@host`

in der Datei `.forward`.

2.4 NFS – Network File System

Die grundlegenden TCP/IP–Dienste `telnet`, `ftp` und `mail` fanden dadurch weite Verbreitung, daß sie Bestandteile des BSD–UNIX (Berkeley Software Distribution) der University of California at Berkeley (UCB) sind. Dagegen wurde das Network File System von der Firma Sun Microsystems entwickelt und vertrieben. Es unterscheidet sich von vergleichbaren Produkten anderer Hersteller (z.B. dem Remote File System der Firma AT&T) dadurch, daß es für fast alle Rechner verfügbar ist und daß das Protokoll als RFC 1094 veröffentlicht wurde.

Mit Hilfe des Network File Systems wird ein Directory oder ein Filesystem eines entfernten Rechners über das Netz auf einem lokalen Rechner verfügbar gemacht. Für den Benutzer sieht es so aus, als wenn die Dateien des entfernten Rechners auf der lokalen Festplatte wären.

Datei–Exporteure nennt man NFS–Server, Datei–Importeure NFS–Clients. UNIX–Anlagen können beide Rollen, auch gleichzeitig, einnehmen, während MS–DOS–Rechner als NFS–Clients auftreten.

Mit dem Befehl `mount` kann man nachsehen, ob Dateisysteme auf dem lokalen Rechner residieren oder ob sie durch NFS von einem Server importiert werden. **Nur der Systemadministrator (`root`) kann importieren oder exportieren**.

Dateiattribute vom exportierenden Rechner bleiben auf dem importierenden Rechner erhalten (Zugriffsrechte, Besitzer, Zeitangaben). Dabei können zwei Probleme auftauchen:

1. Die Dateieigentümer werden über die UID identfiziert. Gehört eine Datei auf dem exportierenden Rechner dem Benutzer karl mit der UID 4711, dann muß karl auf dem importierenden Rechner auch die UID 4711 haben, damit er an seine Datei herankommt. Dies erfordert eine konsistente

Benutzerverwaltung, die man entweder durch ein konsequentes Schema oder den Einsatz einer mehrere Rechner umspannenden Benutzerverwaltung, wie z.B. Yellow Pages (YP), erreicht.

2. Wenn die Uhren der beiden beteiligten Rechner nicht synchron laufen, kann es zur Verwirrung kommen. Geht zum Beispiel die Uhr des Exporteurs vor, dann sieht es für den Importeur so aus, als wenn eine gerade angelegte Datei in der Zukunft angelegt wurde, was unter anderem das Programm `/bin/ls` dazu veranlaßt, das Datum des letzten Dateizugriffs in einer anderen Form darzustellen. Diese Probleme kann man umgehen, indem man auf den Rechnern den Zeitdämon `timed` laufen läßt, sodaß die Uhren synchron laufen.

2.5 X–Windows

Das X–Window–System wurde am Massachusetts Institute of Technology (MIT) im Rahmen des Projektes Athena entwickelt. Die Protokoll–Spezifikationen der Version 11 wurden als RFC 1013 veröffentlicht. Seit Januar 1990 liegt das Release 4 vor. Der Quellcode ist kostenlos erhältlich. Für einige UNIX–Rechner sind vollständige Systeme auf dem Distributionsband des MIT [6]. Für andere Rechner muß X–Windows käuflich erworben werden (MS-DOS, 386/ix, Sony, SINIX, etc.).

X–Windows wurde entwickelt, um auf Arbeitsplatzrechnern unterschiedlicher Hersteller dem Benutzer eine einheitliche graphische Oberfläche bieten zu können. Anders als andere Window–Systeme, wie z.B. MS-Windows, SunView oder der Oberfläche des Macintosh, wurde X–Windows netzwerkfähig konzipiert, sodaß dem Benutzer alle Ressourcen im Netzwerk unter einer graphischen Oberfläche zur Verfügung stehen. Gegenüber anderen netzwerkfähigen Windowsystemen (z.B. NeWS) hat X–Windows den Vorteil der größeren Verbreitung.

X–Windows bedient sich des Client–Server–Modells. Dabei ist der Server das Programm, das den Bildschirm, die Tastatur und die Maus steuert. Der Server kommuniziert mit seinen Clients über eine Netzverbindung, wobei diese zur Zeit eine TCP/IP– oder DECNET–Verbindung sein kann oder eine Verbindung auf dem selben Rechner. Der Server stellt seinen Clients Ressourcen auf dem Bildschirm zur Verfügung und benachrichtigt sie über Tastatureingaben und Mausaktionen. Die Clients können auf wesentlich leistungsfähigeren Rechnern laufen als der Server. Zum Beispiel kann ein X–Server auf einem MS–DOS–PC oder einem X–Terminal laufen, während der Client, der auf diesem Server graphische Ausgabe machen will, auf einer Cray läuft. Diese Bezeichnungsweise führt manchmal zur Verwirrung.

Die wichtigsten Clients sind der Windowmanager und `xterm`. Der Windowmanager sorgt für das Verschieben, Ikonisieren und andere Operationen mit Fenstern auf dem Bildschirm. Es gibt verschiedene Windowmanager: `uwm`, `wm` und `twm` werden vom MIT mitgegeben. In der Regel machen alle Windowmanager das gleiche, nur ist die Form der Ansprache eine andere.

`xterm` ist ein Terminal–Emulator, der vom MIT entwickelt wurde, um auch ältere Software unter X–Windows weiterbetreiben zu können. Es besteht aus zwei Fenstern. Das eine verhält sich wie ein zeichenorientiertes DEC VT100–Terminal, das andere wie ein Tektronix 4014 Vektor–Graphik–Terminal.

Verhält sich eine X–Windows–Implementierung einigermaßen MIT–kompatibel, so wird das System mit dem Befehl

```
xinit
```

gestartet. Der Suchpfad sollte dann auch auf `/usr/bin/X11` zeigen. Es gibt mehrere Hersteller, die X–Funktionalitäten in ihr eigenes Window–System einbetten (Sun, Apollo, Silicon Graphics, Siemens, u.a.). Dort sind dann die Besonderheiten jeweils extra zu erforschen.

Zumindest der Xserver und ein `xterm` sollten durch den Aufruf von `xinit` gestartet werden. Der Aufruf eines Windowmanager kann dann von Hand oder automatisch in einem Ressourcen–File geschehen.

[6] Server und Clients: SunOS 4.0.3, HP–UX 6.5, DomainOS 10.1, A/UX 1.1, AIX RT–2.2, AIX PS\ 2-1.1, IBM, AOS-4.3 und UTEK 4.0. Clients: NEWS-OS 3.2, UNICOS 5.0.1, Unix System V.3.2 (AT&T 6386 WGS)

Sollen auf dem Server auch Clients anderer Rechner zugelassen sein, so ist dies dem Server mitzuteilen. Mit

`xhost + <host>` wird die Berechtigung erteilt und mit

`xhost - <host>` wird die Berechtigung entzogen.

Einem Client muß man mitteilen, auf welchem Bildschirm er ausgeben soll. Dies geschieht entweder durch Angabe eines Parameters beim Programmaufruf oder durch Setzen einer Umgebungsvariablen `DISPLAY`:

`DISPLAY=host:0.0 ; export DISPLAY` (in der Bourne–Shell) oder

`setenv DISPLAY host:0.0` (in der C–shell).

Dabei muß für `host` ein valider Rechnername oder eine IP–Adresse eingesetzt werden. Hinter dem Doppelpunkt erfolgt hier die Angabe, daß der erste Server und der erste Bildschirm gemeint sind.

3 Administratoraufgaben

3.1 IP–Philosophie

Für die folgenden Ausführungen ist es hilfreich, ein wenig über die Philosophie des Internet Protocols (IP) zu wissen. IP kennt zwei Arten von Kommunikationsbeziehungen: Punkt–zu–Punkt–Verbindungen und Rundrufe an alle (Broadcasts). Jedes IP–Paket trägt die Adresse von Sender und Empfänger mit sich. Es weiß aber nicht, welchen Weg es gehen wird. Den richtigen Weg zu finden, ist Aufgabe der Kommunikationsgeräte, die das Paket verschicken.

Die Entscheidung, an welches Gerät ein Paket weitergereicht wird, fällt anhand der Zieladresse (und nicht über die Angabe eines Weges). IP–Adressen sind 32 Bit lang und werden wegen der besseren Lesbarkeit für den Menschen in einer byteweisen Punktnotation niedergeschrieben, z.B. 130.149.2.16. Sie werden mittels der sogenannten Netzmaske in einen Netz– und einen Hostteil unterteilt. Eine Netzmaske von z.B. 255.255.255.0 besagt, daß die ersten 3 Bytes das Netz benennen und das letzte Byte zur Adressierung des Hosts in diesem Netz verwendet wird. Die grundlegende Frage, die sich jeder Verschicker von IP–Paketen (sei es eine Workstation oder ein Router) stellen muß , ist, ob die Zieladresse im eigenen Netz liegt (dann kann direkt zugestellt werden) oder ob andernfalls ein Gateway gefunden werden kann, das das Zielnetz auf irgendwelchen Wegen erreichen kann.

Zur Verdeutlichung ein Beispiel: Der Rechner mit der IP–Adresse 130.149.2.16 möchte ein Paket an den Rechner 130.73.128.2 loswerden. Die Netzmaske des Absenders sei 255.255.255.0. Da die ersten drei Byte von Sender– und Zieladresse nicht übereinstimmen, sucht der Rechner in seinen Routingtabellen nach einem geeigneten Gateway, das das Zielnetz erreichen kann. Ist kein spezielles Gateway für das Zielnetz benannt (statisch oder dynamisch, s.u.), hat er noch die Möglichkeit das Paket an sein `default`–Gateway zu schicken. Wenn alles richtig konfiguriert ist, sollte das an der TUB in diesem Beispiel der Router mit der Adresse 130.149.2.1 sein, da in jedem Subnetz an der TUB die Hostadresse 1 für den Router am Backbone reserviert ist. Für Broadcasts sind die Adressen 255 und 0 reserviert.

Bevor der Systemadministrator die TCP/IP–Software des Rechners konfiguriert, muß er einige Informationen einholen (beim Netzwerkadministrator der ZRZ), sodaß er untenstehende Tabelle ausfüllen kann:

	Beispiel	eigene Angaben
Rechnername	w100zrz	
Rechner–IP–Adresse	130.149.2.2	
Interfacename	le0	
Netzmaske	255.255.255.0	
Routername	gate2	
Router–IP–Adresse	130.149.2.1	
Domänenname	zrz.tu-berlin.de	
Domain–Name–Server	mailgzrz.tu-berlin.de	
DNS–IP–Adresse	130.149.4.10	
Mailhost	mailgzrz.tu-berlin.de	
Mailhost–IP–Adresse	130.149.4.10	

3.2 Interface Konfigurieren

Bei der Installation eines UNIX–Systems wird dem Rechner ein Knotenname zugeordnet. Auf BSD-Systemen kann dieser Name mit dem Befehl `hostname` abgefragt werden, auf System–V–Anlagen mit `uname -n`. Bei vielen Systemen wird bei der Initialisierung des Netzinterfaces auf diesen symbolischen Namen zurückgegriffen. Dieser Name muß nicht zwingend eindeutig im Netz sein, insbesondere nicht, wenn der Domain Name Service benutzt wird (s.u.).

Hat man für einen Rechner eine IP–Adresse, muß dafür gesorgt werden, daß das Ethernet–Interface logisch mit dieser Nummer arbeitet. Bei UNIX-Anlagen wird dies in der Regel mit dem Befehl `ifconfig` während der Boot-Phase erreicht. Der allgemeine Aufbau ist:

```
ifconfig <if> <ipaddr> netmask <nm> broadcast <bc> <up|down>
```

wobei
<if> der Name des Interfaces ist (le0, eth0, wd0, etc.),
<ipaddr> die IP-Adresse (z.B. 130.149.2.16),
<nm> die Netzmaske (z.B. an der TUB 255.255.255.0),
<bc> die Broadcastadresse (z.B. 130.149.2.255 für das Segment 2)
ist und `up` das Interface in Betrieb nimmt.
Beispiel:

```
ifconfig wd0 130.149.2.16 netmask 255.255.255.0 broadcast 130.149.2.255 up
```

3.3 Routing – Wegefindung

Da keine Implementation vom Hersteller auf unser spezielles Subnetting vorkonfiguriert ist, muß auf jeden Fall an der TUB die **Netzmaske** gesetzt werden. Sie wird bei den verschiedenen Implementationen an unterschiedlicher Stelle gesetzt. Hier einige Beispiele:

SunOS 4.0.1	in der Datei `/etc/netmasks` oder `rc.local`
Ultrix V2.1	in `/etc/rc.local` mit dem `ifconfig`–Aufruf
ConvexOS V8.0	in `/etc/rc.local` mit dem `ifconfig`–Aufruf
DomainOS 10.1 (BSD)	in `/etc/rc.local` mit dem `ifconfig`–Aufruf
NEWS-OS 3.9R	in `/etc/rc.local` mit dem `ifconfig`–Aufruf
386/ix 2.0.2 + 2.2	in `/etc/netd.cf` beim `ifconfig`–Aufruf
IRIX SysV 3.2.1	in `/etc/config/ifconfig.1-options`
HP-UX 6.5 und 7.0	in `/etc/netlinkrc` beim `ifconfig`–Aufruf
AIX 1.2 (PS/2)	in `/etc/rc.tcpip` beim `ifconfig`–Aufruf
PC–NFS 3.0.1	mit NFSCONF in den erweiterten Netzwerkfragen
PC/TCP	z.B. mit `IFCONFIG WD8003.SYS SUBNET 8`
NCSA-Telnet 2.2D	in `CONFIG.TEL` (z.B. `netmask=255.255.255.0`)

Ohne richtig gesetzte Netzmaske ist an der TUB kein Verkehr mit den zentralen Servern oder anderen Netzsegmenten möglich!

Das Routing kann statisch oder dynamisch erfolgen. Im Normalfall, wenn z.B. nur ein Router am Netzsegment angeschlossen ist, reicht das statische Routing aus. Jedoch flexibler ist das dynamische Routing.

3.3.1 Statisches Routing

Beim statischen Routing wird der TCP/IP–Software des eigenen Rechners mitgeteilt, über welchen Router welches Netz zu erreichen ist. Man kann sich mit dem Programm `route` eine Tabelle erstellen, in der steht, Netz A erreicht man über Gateway X, Netz B erreicht man über Gateway Y, usw. Einträge in diese Tabelle generiert man mit dem Aufruf:

```
route add net gateway hopcount
```

wobei `net` der Name oder die IP–Adresse des zu erreichenden Netzwerkes ist und `gateway` Namen oder IP–Adresse des Gateways angibt. Zusätzlich kann man noch ein `default`–Gateway benennen an das alle anderen Pakete geschickt werden. Der `hopcount` kann zu 1 gesetzt werden. Für den einfachen Fall, daß nur ein Router existiert, über den der gesamte Fernverkehr abgewickelt wird, wie es für fast alle TU–Institute gegeben ist, reicht daher die alleinige Angabe eines `default`–Gateways, z.B.:

```
route add default 130.149.2.1 1
```

Der Aufruf von `route` erfolgt im entsprechenden `rc`–File während des Bootens. Nachteilig bei diesem Verfahren ist, daß bei einem Ausfall eines Routers und Verfügbarkeit eines zweiten Routers oder bei einer Neukonfiguration des Netzes auf allen Rechnern evtl. die Routing–Informationen geändert werden müssen. Dann ist das dynamische Routing gefragt.

3.3.2 Dynamisches Routing

Beim dynamischen Routing überläßt man die mühselige Arbeit, die neuesten Wegeinformationen nachzuhalten, willfährigen Dämonen, derer es mindestens zwei verschiedene gibt: `routed` und `gated`. Diese lauschen auf dem Netz, was die Router zu erzählen haben, und merken sich alles.

Auf Berkeley–Anlagen (z.B. SunOS) findet man häufig den `routed`. Dieser braucht keine weitere Informationen vorab und wird in `/etc/rc.local` gestartet.

Auf mancher Maschine (z.B. 386/ix) findet man den `gated`, dem in der Datei `/etc/gated.conf` mitgeteilt wird, welche Routing–Protokolle er verstehen soll (zur Zeit nur RIP, und das auch noch `quiet` bei 386/ix). Gestartet wird der `gated` auch in einem `rc`–File, nachdem der Internet–Dämon `inetd`

gestartet wurde. Als Nachteil erhält man einen weiteren Prozeß , der meist überflüssig ist, wenn nur ein Router vorhanden ist.

3.4 Symbolische Adressen

Die Programme `telnet, ftp, mail` und `X-windows` sollten immer sowohl mit einer numerischen IP–Adresse als auch mit einem symbolischen Rechnernamen arbeiten können. Die tieferen Protokoll–Schichten arbeiten immer mit einer IP–Adresse. Deshalb muß sichergestellt werden, wie der Rechner aus der Angabe eines Namens auf eine IP–Adresse schließen kann. Dies kann auf unterschiedlichen Wegen erreicht werden und es lassen sich vier Fälle unterscheiden:

1. Statisch durch Vorhalten einer Tabelle auf jedem Rechner in der Datei /etc/hosts.
2. Dynamisch durch Abfragen einer verteilten Datenbank im Netz mit dem `Domain Name Service`.
3. Dynamisch durch Abfragen einer verteilten Datenbank im Netz mit `Yellow Pages`.
4. Durch eine Mischform der vorgenannten Möglichkeiten.

3.4.1 /etc/hosts

Der einfachste Fall ist das Vorhalten einer kleinen Datei `/etc/hosts`. Als Minimal–Eintrag reicht eine Zeile mit: `127.1.0.0 local localhost`. Eine weitere Zeile mit der IP–Adresse und dem Namen des eigenen Rechners brauchen einige Implementationen, um während des Bootens das Interface mit der richtigen IP–Adresse versehen zu können.

Dann kann man hingehen, und jeden Rechner, mit dem man irgendwann Kommunikation haben möchte, eintragen. Dabei ist es egal, welchen Namen man einer IP–Adresse zuordnet. Mit diesem Verfahren laufen alle Programme, aber für einen Empfänger von Post wird es schwierig sein, Post zurückzuschicken. Deshalb ist es günstig, neben dem Unix–Knotennamen auch den offiziellen Rechnernamen im Hosts–File zu haben. Somit würde ein Eintrag folgendermaßen aussehen können:

`130.149.2.16 w104zrz.zrz.tu-berlin.de w104zrz`

Mit diesem Eintrag wird die IP–Adresse sowohl unter dem offiziellen Rechnernamen, wie er dann weltweit eindeutig ist, als auch unter seinem Unix–Nodename bekannt (siehe `uname -n` bei System V oder `hostname` bei BSD).

Eine solche Datei kann man sich über `anonymen ftp` auf eben diesem Rechner im Directory `pub/zrz` holen (User ftp, Passwort ftp). In dieser Datei sind alle IP–Rechner der TUB verzeichnet, soweit ihre Namen der ZRZ bekannt sind.

Der große Nachteil bei diesem Verfahren ist, daß bei jeder Änderung der Systemadministrator der Anlage die Datei auf den aktuellen Stand bringen muß. Außerdem sind Außenbeziehungen (außerhalb der TUB) dann nur von Hand darstellbar.

Leider gibt es auch Implementationen, die auf diese Datei angewiesen sind, weil sie den Domain–Name–Service nicht beinhalten, z.B. HP–UX 6.5, oder fehlerhaft implementiert haben, wie z.B. Sony NEWS-OS 3.9R.

3.4.2 Domain–Name–Service

Wesentlich angenehmer ist die dynamische Verwaltung der Rechnernamen mit dem Domain–Name–Service (DNS). Die Arbeit, Rechnernamen auf dem aktuellen Stand zu halten, teilen sich weltweit viele Nameserver, sodaß man von einer verteilten Datenbank sprechen kann [17, 18].

Allgemein ist zu sagen daß die Verwaltung nicht wie die unteren Protokollschichten von der Netzstruktur abhängt. So gehören z.B. die Rechner der Domäne `zrz.tu-berlin.de` unterschiedlichen Subnetzen

an (1 – 10). Der Namensraum ist eine hierarchische Struktur, die von rechts nach links gelesen wird. So ist `de` ein Kennzeichen für die Top–Level–Domäne Deutschland, die zentral von der Universität Dortmund verwaltet wird. Andere Top–Level–Domänen sind zum Beispiel `edu` (Bildungseinrichtungen in den USA) oder `oz` (Australien).

Die Subdomäne `tu-berlin.de` wird von der ZRZ verwaltet, die ihre eigenen Rechner in der Domäne `zrz.tu-berlin.de` gruppiert. Aus Rechnername und Domänenname zusammen ergibt sich der offizielle Rechnername oder auch Full Qualified Domain Name. Da es sich bei der Errichtung einer Domäne wegen der weltweiten Bedeutung um eine wichtige Maßnahme handelt, müssen solche Aktivitäten mit dem Netzwerkadministrator (in der ZRZ: Herr Kasielke) koordiniert werden. Die ZRZ bietet an, entweder weitere Subdomänen auf ihren zentralen Nameservern einzurichten oder beim Einrichten von Subdomänen zu helfen.

Die Arbeitsweise des DNS sei kurz skizziert: Die Client–Programme (`telnet, ftp` usw.) fragen über das Netz ihren Domain–Name–Server, welche IP–Adresse der Rechner mit dem Namen x.y.z hat. Führt die Anfrage zum Erfolg, steht danach die IP–Adresse des Zielrechners zur Verfügung. Ist dem Domain–Name–Service kein Rechner unter dem Namen bekannt, verhalten sich die Implementationen unterschiedlich. 386/ix schaut bei einem Scheitern der Anfrage noch in `/etc/hosts` nach und Ultrix kann sogar eine Suchreihenfolge vorgegeben werden (in der Datei `/etc/svcorder`), in der DNS, /etc/hosts und Yellow Pages befragt werden sollen.

Die ZRZ empfiehlt den Domain–Name–Service, wo immer es geht, zu benutzen. Der Nameserver für die Domänen `tu-berlin.de` und `zrz.tu-berlin.de` ist auf dem Rechner `mailgzrz.tu-berlin.de` (130.149.4.10) zu erreichen.

Der Domain-Name–Service wird in der Regel durch einen Eintrag in der Datei `/etc/resolv.conf` aktiviert. Es reichen zwei einfache Zeilen mit der Angabe der eigenen Domäne und der IP–Adresse des Nameservers, z.B.:

```
domain tu-berlin.de
nameserver 130.149.4.10
nameserver 130.149.5.4
```

3.4.3 Yellow Pages — Network Information Service (NIS)

Auch Yellow Pages stellt eine verteilte Datenbank dar. Es gibt einen Yellow–Pages–Master, an den die Sklaven der gleichen Yellow–Pages–Domäne Anfragen richten. Yellow Pages verwaltet neben Rechnernamen und Netznamen auch Benutzer, Gruppen und anderes.

Obwohl Yellow Pages in der Benutzerverwaltung erhebliche Vorteile bringt, ist es in der Rechnernamensverwaltung schwach. Die Yellow Pages Rechner–Tabellen werden aus der Datei `/etc/hosts` abgeleitet, vertragen sich aber dort nicht mit DNS–Namen. Anfragen werden bei der Standard–Distribution von SunOS 4.0.3 nur an Yellow Pages gerichtet. Eine Abfrage des DNS ist nur mit einem zusätzlich zu beschaffenden Bug–Fix möglich. Günstiger verhält sich Ultrix. Ein weiterer Seiteneffekt ist, daß die Yellow–Pages–Domäne mit der Domäne des DNS übereinstimmen muß. Will man also beides benutzen, muß man neben der Yellow–Pages–Domäne (die dann nach dem DNS–Schema benannt wird) auch für die Einrichtung eines Namensteilraumes sorgen.

3.4.4 Einige Beispiele

Die Erfahrungen mit einzelnen Implementationen seien hier mitgeteilt:

- HP-UX 6.5:
  - `/etc/hosts:` kann verwendet werden.

  - DNS: nicht vorhanden.
  - YP: nicht vorhanden.

- HP-UX 7.0:

  - `/etc/hosts:` kann verwendet werden.
  - DNS: kann über `/etc/resolv.conf` aktiviert werden.
  - YP: vorhanden, Zusammenspiel mit DNS nicht bekannt.

- Interactive 386/ix 2.0.2 und 2.2:

  - `/etc/hosts:` kann verwendet werden.
  - DNS: kann über `/etc/resolv.conf` aktiviert werden.
  - YP: vorhanden, Zusammenspiel mit DNS nicht bekannt.

- SunOS 4.0.3 und 4.1:

  - `/etc/hosts:` kann verwendet werden.
  - DNS: kann mit der Standard–Distribution nicht über `/etc/resolv.conf` aktiviert werden. Soll DNS verwendet werden, ohne daß YP läuft, muß eine andere `libc.a` bereitgestellt werden.
  - YP: wenn YP läuft, wird nur dort nachgeschaut, nicht aber in `/etc/hosts`. Wird der YP–Master–Server mit `ypserv -i` gestartet, wird nach YP auch im DNS nachgeschaut.

- Ultrix 2.1:

  - `/etc/hosts:` kann verwendet werden.
  - DNS: kann über `/etc/resolv.conf` aktiviert werden.
  - YP: Vorhanden, aber nicht dominant. Suchreihenfolge kann in `/etc/svcorder` festgelegt werden.

- AIX 1.2:

  - `/etc/hosts:` kann verwendet werden.
  - DNS: kann über `/etc/resolv.conf` aktiviert werden.
  - YP: vorhanden, Zusammenspiel mit DNS nicht bekannt.

- Sony NEWS 3.9R:

  - `/etc/hosts:` kann verwendet werden.
  - DNS: kann nicht über `/etc/resolv.conf` aktiviert werden. Der Fehler in der Implementation soll im Release 4.0 (Ende 1990) behoben werden.
  - YP: vorhanden. DNS kann nicht befragt werden.

- Silicon Graphics IRIX:
  - `/etc/hosts:` kann verwendet werden.
  - DNS: kann über `/usr/etc/resolv.conf` aktiviert werden. Wenn DNS aktiviert ist, kann die Console nicht mehr benutzt werden!
  - YP: Wenn YP läuft, können die anderen beiden Dienste nicht befragt werden.
- PC–NFS 3.0.1:
  - `/etc/hosts:` Unix–Datei kann auf dem PC verwendet werden als datei `C:\NFS\HOST`
  - DNS: nicht vorhanden (nur indirekt über YP ansprechbar).
  - YP: kann in Anspruch genommen werden, dann ist Host–Datei unwirksam.
- NCSA–Telnet 2.2D:
  - `/etc/hosts:` kann nicht verwendet werden. In anderer Notation können aber Hosts in der Datei `CONFIG.TEL` eingetragen werden.
  - DNS: kann verwendet werden, ein Hostname kann sogar in mehreren Domänen gesucht werden.
  - YP: wird nicht unterstützt.
- PC/TCP:
  - `/etc/hosts:` Die UNIX–Datei kann verwendet werden.
  - DNS: kann verwendet werden.
  - YP: wird nicht unterstützt.

3.5 Mail

In der Regel werden UNIX–Anlagen vom Hersteller so vorkonfiguriert, daß das Programm `sendmail` als Dämon im Hintergrund lauert, daß es endlich Post zustellen darf. Mit `ps` sieht man dann einen Eintrag wie z.B. `/usr/lib/sendmail -bd -q30`. In regelmäßigen Abständen wird nachgeschaut, ob für noch nicht zugestellte Post noch ein weiterer Zustellversuch gemacht werden soll, oder ob der Absender auf der lokalen Maschine benachrichtigt werden soll, daß der Empfänger nicht zu erreichen ist.

Der Systemaministrator sollte sich durch einen Test mit lokaler Post von der Lauffähigkeit von `mail` überzeugen. Aus Sicherheitsgründen sollte er darauf achten, daß das Directory `/usr/spool/mqueue`, in dem noch nicht zugestellte Post zwischengelagert wird, nicht für alle lesbar ist.

Eines der schönsten Beispiele abstrakter Kunst ist das Konfigurationsfile `sendmail.cf`. Im Vergleich zu der darin verwendeten Notation sind Hexdumps als milde Prosa zu bezeichnen. Da die Syntax nicht unmittelbar einleuchtend ist, wird davor gewarnt, `sendmail.cf` ohne Not zu ändern. Allerdings sind nur die Rewriting–Rules im hinteren Teil schwer zu verstehen. Leichte Anpassungsarbeiten im vorderen Teil sind durchaus machbar, zumal in der Regel nur drei Punkte zu beachten sind:

1. Für die meisten Workstations ist nur der Ethernet–Anschluß für E–Mail interessant, sodaß alle Hinweise für UUCP übergangen werden können (außer daß die Post standardmäßig auch über das Ethernet geschickt wird anstatt über serielle Leitungen).

2. Ein offizieller Hostname ist zu setzen ($j), der sich aus Nodename ($w) und Domainname ($m) zusammensetzt, sodaß man häufig eine Zeile findet der Form: `Dj$w.$m`. Bei YP–Rechnern kann es vorkommen, daß die YP–Domaine automatisch angehängt wird, sodaß `Dj$w` ausreicht.

3. Ein Mailforwarder ist zu benennen (z.B. `mailgzrz.tu-berlin.de`), an den die Workstation die Post schicken kann mit der Bitte um Weiterleitung, wenn die Workstation nicht alleine (mit `/etc/hosts`, DNS oder YP) herausbekommt, wie sie den Zielrechner erreichen soll.

Sind Änderungen an `sendmail.cf` gemacht worden, so muß `sendmail` neu gestartet werden, um die Änderungen wirksam zu machen.

3.6 Exportieren von Filesystemen mit NFS

Der Systemadministrator bestimmt, welche Filesysteme eines lokalen Rechners welchen anderen Rechnern im Netz zur Verfügung gestellt werden. Dies tut er, indem er einen Eintrag in der Datei `/etc/exports` vornimmt. Es werden das Filesystem, das exportiert werden soll, die Rechner, die diese importieren dürfen, und eventuell Optionen angegeben.

Zu den Optionen gehört, ob ein Filesystem nur gelesen werden darf (read only) oder ob fremde Rechner auf der lokalen Festplatte schreiben dürfen. Es wird immer an einen Rechner und nicht an bestimmte Benutzer exportiert. Eine Ausnahme bildet der Benutzer `root`. Fremde Superuser dürfen nur auf der lokalen Platte schreiben, wenn ihnen das explizit erlaubt wurde. (Allerdings erlauben nicht alle Implementationen diesen `root-access`). Wenn ein fremder Superuser auf der lokalen Platte schreiben will, wird seine User–ID (UID) von 0 zu -2 transformiert, sodaß sichergestellt wird, daß nicht Unbefugte mit höchsten Privilegien Dateien verändern dürfen.

Prinzipiell gilt, daß die Zugriffsrechte auf Dateien auch bei NFS über die UID eingeräumt werden. Daraus ergibt sich, daß die Benutzerverwaltungen auf Exporteur und Importeur gleich sein müssen, wenn Dateieigentümer konsistent sein sollen. Bei Read–Only–Dateien, die nicht gegen unbefugtes Lesen geschützt werden brauchen, oder beim Export von Swap–Space für plattenlose Workstations spielt das keine Rolle.

Ist der Eintrag in die Datei `/etc/exports` erfolgt, so muß bei manchen Implementationen (z.B. SunOS, Ultrix, IRIX, etc.) noch die Änderung wirksam gemacht werden, indem `exportfs -a` aufgerufen wird.

Mit `showmount -e` kann dann nachgesehen werden, was der Rechner tatsächlich gerade exportieren möchte.

Mit `showmount` allein, kann man feststellen, welche Rechner gerade von dem Exportangebot Gebrauch machen.

Mit `showmount -a` wird zudem noch angegeben, welches Filesystem von fremden Rechnern importiert wird.

3.7 Mounten und unmounten von Filesystemen mit NFS

Auch das Importieren von Filesystemen darf nur vom Superuser vorgenommen werden. Zunächst kann mit dem Befehl `showmount -e <host>` festgestellt werden, welche Filesysteme zum Import zur Verfügung stehen.
Dann kann mit

```
mount -f NFS otherhost:/usr/expo /usr/impo
```

das fremde Filesystem `/usr/expo` des Rechners `otherhost` unter dem Namen `/usr/impo` verfügbar gemacht werden, als ob es auf der eigenen Festplatte stünde. Das Directory `/usr/impo` muß vorher auf dem lokalen Rechner als Mountpoint vorhanden sein und sollte leer sein.
Mit

```
umount /usr/impo
```

kann der Vorgang dann rückgängig gemacht werden.

Sollen externe Filesysteme immer automatisch nach dem Einschalten des Rechners gemountet werden, so ist ein entsprechender Eintrag in die Datei `/etc/fstab` vorzunehmen.

3.8 Konfigurieren von X–Windows

Wird das X–Windowsystem von den MIT–Quellen generiert, ist nur sehr wenig zu konfigurieren. Auch wenn Hersteller reine X–Windows Implementationen liefern, sind sie oft fertig vorkonfiguriert. Lediglich bei gemischten Systemen ist ein genaues Studium der Handbücher erforderlich.

Das MIT liefert seine Versionen in drei Directories aus:

1. `/usr/bin/X11` (Clients: xterm, xinit, etc.)
2. `/usr/lib/X11` (Fonts u.a.)
3. `/usr/include/X11` (Header-Dateien für die X–Lib)

Wenn auf dem Rechner eigene X–Windows–Programme entwickelt werden sollen, dann sollte zumindest die X–Lib nicht fehlen. In der Regel ist dies die Datei `/usr/lib/libX11.a`.

Der Systemadministrator sollte prüfen, ob mit `xinit` der Server und wenigstens ein `xterm` gestartet werden.

Dann sollte geprüft werden, welcher Windowmanager standardmäßig läuft, und ob vielleicht ein angenehmerer zur Verfügung steht, z.B. Tom's Window Manager `twm`. Bei diesem sollte dann für die Benutzer ein getestetes Ressourcen–File `.twmrc` zur Verfügung gestellt werden. Windowmanager sind Geschmackssache und es läßt sich darüber trefflich streiten.

Das MIT hat im Release 11.3 die folgenden Clients als harten Kern ausgeliefert:

`bitmap, uwm, x10tox11, xbiff, xcalc, xclipboard, xclock, xdm, xdpyinfo, xedit, xev, xfd, xhost, xinit, xkill, xload, xlogo, xlsfonts, xlswins, xmag, xman, xmh, xmodmap, xpr, xprop, xpseudoroot, xrdb, xrefresh, xset, xsetroot, xterm, xwd, xwininfo, xwud`.

Anhand der obigen Aufzählung kann geprüft werden, ob die Implementierung einigermaßen vollständig ist.

Die Anwesenheit einiger X–Applikationen, die z.T. über die Standard–Auslieferung hinausgehen, sollte genauer geprüft werden:

- `xterm` — ein Terminal–Emulator (VT100 und TEK4014)
- `xman` — ein Manual–Browser
- `xdvi` — ein TEX–Previewer für dvi–Files

Die Benutzer sollten darauf hingewiesen werden, das Directory `/usr/bin/X11` in ihren Suchpfad mitzuübernehmen.

4 Troubleshooting

4.1 Hardware–Probleme

- Sind alle Rechner (Workstation, Repeater, Bridge, Router) eingeschaltet, die an der Kommunikation beteiligt sein sollen?
- Sind die Verbindungsskabel (Koaxial, Ethernet, etc) in Ordnung?
- Sind an beiden Enden des Segments die Endwiderstände (50 Ohm) angebracht?
- Sind die T–Stücke an den Adaptern (Cheapernet) ordentlich aufgesetzt?
- Schließen alle Bajonett–Verschlüsse richtig?

4.2 Software–Probleme

- Arbeitet der Ethernet–Adapter? Abfrage mit `ifconfig <interface>` und `netstat -r`. Einträge in rc.local und /etc/hosts überprüfen.
- Stimmen die IP–Adresse, die Netzmaske, der Broadcast, die Routing–Informationen?
- Ist das Interface up? Abfrage mit `ifconfig <ifname>`. Der Name des Interfaces (le0, eth0, se0, wd0, etc.) läßt sich durch den Aufruf von `netstat -r` herausbekommen.
- Läßt sich der eigene Rechner über `telnet` ansprechen (inner loop)? Wenn nicht, überprüfen, ob Rechner logisch schon am Netz hängen soll (telinit 3, /etc/inittab isdefault).
- Ist die TCP/IP–Software im aktuellen Kernel eingebunden? (Bei den Rechnern, wo die TCP/IP–Software gesondert geliefert wird, z.B. SCO–UNIX, Interactive 386/ix oder NFS bei HP-UX).
- Ist der Zielrechner oder der nächstliegende Router zu erreichen? Abfrage mit `ping <hostname>`.

5 Sicherheit

Oft werden Unix–Anlagen von den Herstellern so vorkonfiguriert, daß man in einer kleinen Abteilung sehr offen mit anderen Systemen kommunizieren kann. Dies ist aber in einem großen Netzwerk nicht vertretbar. Deshalb seien hier einige Anmerkungen gemacht, die vor einem Rechnerzugang durch Unbefugte und vor Datenverlust schützen sollen.

- Man kann davon ausgehen, daß das Netzwerk weltweit erreichbar ist. Dies kann über einen Internet–Anschluß , über X.25 oder Wählanschlüsse geschehen. Daher sollte kein Account ohne Paßwort sein.
- Das Paßwort sollte von Zeit zu Zeit gewechselt werden, da von jedem Netzknoten der Verkehr auf dem eigenen Ethernet–Segment mitgehört werden kann. Insbesondere beim `ftp` wird das Paßwort in einem Paket über das Netz geschickt — in Klartext, unverschlüsselt.
- Es sollte kein Account guest oder Gast existieren, da dieser sehr leicht erraten werden kann.
- Der Finger–Dämon (`fingerd`) sollte nicht laufen. Denn er erzählt nicht nur in der eigenen Abteilung, wer in dieser Abteilung arbeiten darf, sondern tut dies weltweit.

- Die Datei `/etc/hosts.equiv` sollte mit besonderer Sorgfalt geführt werden, wenn sie überhaupt existieren muß . Insbesondere die Firmen Sun und Sony konfigurieren sie bei der Auslieferung so, daß ein unerlaubtes Eindringen in den Rechner erleichtert wird (durch ein + in der letzten Zeile).
- Die Datei `.rhosts` im Home–Directory für `rlogin, rsh, rcp` sollte mit besonderer Sorgfalt geführt werden. Insbesondere verrät sie, auf welchen Anlagen für den Benutzer auch Accounts eingerichtet sind.
- Der Remote–Who–Dämon `rwhod` sollte mit Bewußtsein aktiviert werden. Auch er erzählt auf dem ganzen Subnetz, wer wann und wo arbeitet.
- Ist der Time–Dämon `timed` aktiviert, kann es passieren, daß die Uhr eines Rechners von einer fremden Anlage gestellt wird (was auch sein eigentlicher Zweck ist: verschiedene Uhren zu synchronisieren). Allerdings kann das zu unliebsamen Überraschungen bei `cron`-Jobs und Dateiattributen führen.
- Beim NCSA–Telnet ist in der Datei `config.tel` darauf zu achten, ob während einer Telnet–Sitzung `ftp` erlaubt sein soll. Dann ist eventuell der Passwort–Mechanismus mit einzukonfigurieren. Anderenfalls können Fremde auf der DOS–Platte Dateien verändern.
- NFS–Dateien sollten als read–only oder nur an bestimmte Rechner exportiert werden. Dateien, die an viele Rechner exportiert werden, deren Administration man nicht kennt, sollten Root gehören, damit es keine Zugriffsprobleme mit gleichen UID's gibt.
- Kann der Rechner auch arbeiten, wenn das Netz ausfällt?
- NFS–Dateien sollten mit der Option `soft` importiert oder gemountet werden. Fällt der Netzwerk–Server aus, hängt auch die lokale Workstation, wenn mit der Option `hard` gemountet wurde.

6 NOS/BE–Migrationshilfen

6.1 BETERM

Für den **interaktiven Zugang** von einem UNIX–Rechner zu den NOS/BE–Anlagen stellt die ZRZ die Eigenentwicklung BETERM zur Verfügung. Damit ist es möglich, bequem und zeilenorientiert unter NOS/BE zu arbeiten. Insbesondere wird ein Arbeiten mit dem Editor TUBE ermöglicht. BETERM kann auf unterschiedlichen UNIX–Anlagen installiert werden. Einzelheiten sind von Frau Engelke, Tel. 314–24687, zu erfahren.

6.2 WO2BE und BE2WO

Für den **Dateitransfer von Textdateien** zwischen einem ftp–Rechner und einem NOS/BE–Rechner stellt die ZRZ auf den Cybern die Programme WO2BE (von Wotan nach NOS/BE) und BE2WO (von NOS/BE nach Wotan) zur Verfügung. Einzelheiten sind über DOC, SYSTEM, WO2BE und DOC, WOTAN, BE2WO sowie bei Herrn H. Schulz, Tel. 314–24383, zu erfahren.

7 Informationsquellen und Beratung

- Allgemeine Beratung:
  ZRZ–Beratung: E–N, Raum 003, Tel. 314–25253

- IP–Adressen, Domänen:
  ZRZ–Netzwerkadministrator: Herr Kasielke, E–N, Raum K047, Tel. 314–23733.
- E–Mail—Elektronische Post:
  X.400, SMTP, Bitnet: Herr Elsner, E–N, Raum K045, Tel. 314–23897.
- Public–Domain–Software:
  NCSA–Telnet 2.2D, Kermit 2.31 für MS–DOS.
  X–Windows 11.3 und 11.4 vom MIT für UNIX.
  Herr Ksoll, E–N, Raum K046, Tel. 314–24355.

Literatur

[1] Comer, Douglas: *Internetworking with TCP/IP — Principles, Protocols, and Architecture*. Prentice–Hall, 1988.

[2] Gülker, R., Gürtler, H., Kasielke, D.: *WOTAN. Version C*. Berlin, Mai 1988. (Unveröffentlicht).

[3] Libes, D.:*Choosing a name for your computer*. Communications of the ACM. November 1989, Seite 1289-91.

[4] Santifaller, Michael: *TCP/IP und NFS in Theorie und Praxis*. Addison–Wesley, 1990.

[5] Postel, J.:*RFC 768. User Datagramm Protocol (UDP)*. Sep. 1980.

[6] Postel, J.:*RFC 791. Internet Protocol (IP)*. Sep. 1981.

[7] Postel, J.:*RFC 792. Internet Control Message Protocol*. Sep. 1981.

[8] Postel, J.:*RFC 793. Transmission Control Protocol (TCP)*. Sep. 1981.

[9] Postel, J.:*RFC 821. Simple Mail Transfer Protocol*. Aug. 1982.

[10] Crocker, D.H.:*RFC 822. Standard for the Format of ARPA Internet Text Messages*. Aug. 1982.

[11] Plummer, D.:*RFC 826. An Ethernet Adress Resolution Protocol (ARP)*. Nov. 1982.

[12] Postel, J.:*RFC 854 Telnet Protocol Specification*. May 1983.

[13] Braden, R., Postel, J.:*RFC 1009. Requirements for Internet Gateways*. Jun. 1987.

[14] Scheiffer, R.W.:*RFC1013. X Window System Protocol, Version 11*. June 1987.

[15] Stahl, M.:*RFC 1032. Domain Administrators Guide*. Nov. 1987.

[16] Lottor, M.:*RFC 1033. Domain Administrators Operations Guide*. Nov. 1987.

[17] Mockapetris, P.:*RFC 1034. Domain Names — Concept and Facilities*. Nov. 1987.

[18] Mockapetris, P.:*RFC 1035. Domain Names — Implementation and Specification*. Nov. 1983.

[19] Sun Microsystems Inc.:*RFC 1094. NFS: Network File System Specification*. March 1989.

[20] Braden, R.:*RFC 1122. Requirements for Internet Hosts — Communication Layers*. Oct. 1989.

[21] Braden, R.:*RFC 1123. Requirements for Internet Hosts — Application and Support*. Oct. 1989.